Towards Combining Probabilistic and Interval Uncertainty in Engineering Calculations: Algorithms for Computing Statistics under Interval Uncertainty, and Their Computational Complexity

V. Kreinovich, G. Xiang, S. A. Starks, L. Longpré, M. Ceberio,
R. Araiza, J. Beck, R. Kandathi, A. Nayak and R. Torres
NASA Pan-American Center for Earth and Environmental Studies (PACES), University of Texas, El Paso, TX 79968, USA (`vladik@utep.edu`*)*

J. G. Hajagos
Applied Biomathematics, 100 North Country Road, Setauket, NY 11733, USA, and Dept. of Ecology and Evolution, State University of New York, Stony Brook, NY 11794, USA (`janos@ramas.com`*)*

Abstract. In many engineering applications, we have to combine probabilistic and interval uncertainty. For example, in environmental analysis, we observe a pollution level $x(t)$ in a lake at different moments of time t, and we would like to estimate standard statistical characteristics such as mean, variance, autocorrelation, correlation with other measurements. In environmental measurements, we often only measure the values with interval uncertainty. We must therefore modify the existing statistical algorithms to process such interval data.

In this paper, we provide a survey of algorithms for computing various statistics under interval uncertainty and their computational complexity. The survey includes both known and new algorithms.

Keywords: probabilistic uncertainty, interval uncertainty, engineering calculations, computational complexity

1. Formulation of the Problem

Computing statistics is important. In many engineering applications, we are interested in computing statistics. For example, in environmental analysis, we observe a pollution level $x(t)$ in a lake at different moments of time t, and we would like to estimate standard statistical characteristics such as mean, variance, autocorrelation, correlation with other measurements. For each of these characteristics C, there is an expression $C(x_1, \ldots, x_n)$ that enables us to provide an estimate for C based on the observed values $x_1, \ldots, x_n$. For example, a reasonable statistic for estimating the mean value of a probability distribution is the population average $E(x_1, \ldots, x_n) = \frac{1}{n} \cdot (x_1 + \ldots + x_n)$; a reasonable statistic for estimating the variance V is the population variance $V(x_1,\ldots,x_n) = \frac{1}{n}\cdot\sum\limits_{i=1}^{n}(x_i - E)^2$, where $E \stackrel{\text{def}}{=} \frac{1}{n}\cdot\sum\limits_{i=1}^{n} x_i$.

Comment. The population variance is often computed by using an alternative formula $V = M - E^2$, where $M = \frac{1}{n}\cdot\sum\limits_{i=1}^{n} x_i^2$ is the population second moment.

Comment. In many practical situations, we are interested in an *unbiased* estimate of the population variance

$$V_u(x_1,\ldots,x_n) = \frac{1}{n-1}\cdot\sum_{i=1}^{n}(x_i - E)^2.$$

In this paper, we will describe how to estimate V under interval uncertainty; since $V_u = \frac{n}{n-1}\cdot V$, we can easily transform estimates for V into estimates for V_u.

Interval uncertainty. In environmental measurements, we often only measure the values with interval uncertainty. For example, if we did not detect any pollution, the pollution value v can be anywhere between 0 and the sensor's detection limit DL. In other words, the only information that we have about v is that v belongs to the interval $[0, DL]$; we have no information about the probability of different values from this interval.

Another example: to study the effect of a pollutant on the fish, we check on the fish daily; if a fish was alive on Day 5 but dead on Day 6, then the only information about the lifetime of this fish is that it is somewhere within the interval $[5, 6]$; we have no information about the distribution of different values in this interval.

In non-destructive testing, we look for outliers as indications of possible faults. To detect an outlier, we must know the mean and standard deviation of the normal values – and these values can often only be measured with interval uncertainty (see, e.g., [38, 39]). In other words, often, we know the result $\widetilde{x}$ of measuring the desired characteristic x, and we know the upper bound Δ on the absolute value $|\Delta x|$ of the measurement error $\Delta x \stackrel{\text{def}}{=} \widetilde{x} - x$ (this upper bound is provided by the manufacturer of the measuring instrument), but we have no information about the probability of different values $\Delta x \in [-\Delta, \Delta]$. In such situations, after the measurement, the only information that we have about the true value x of the measured quantity is that this value belongs to interval $[\widetilde{x} - \Delta, \widetilde{x} + \Delta]$.

In geophysics, outliers should be identified as possible locations of minerals; the importance of interval uncertainty for such applications was emphasized in [36, 37]. Detecting outliers is also important in bioinformatics [42].

In bioinformatics and bioengineering applications, we must solve systems of linear equations in which coefficients come from experts and are only known with interval uncertainty; see, e.g., [52].

In biomedical systems, statistical analysis of the data often leads to improvements in medical recommendations; however, to maintain privacy, we do not want to use the exact values of the patient's parameters. Instead, for each parameter, we select fixed values, and for each patient, we only keep the corresponding range. For example, instead of keeping the exact age, we only record whether the age is between 0 and 10, 10 and 20, 20 and 30, etc. We must then perform statistical analysis based on such interval data; see, e.g., [20, 51].

Estimating statistics under interval uncertainty: a problem. In all such cases, instead of the true values $x_1, \ldots, x_n$, we only know the intervals $\mathbf{x}_1 = [\underline{x}_1, \overline{x}_1], \ldots, \mathbf{x}_n = [\underline{x}_n, \overline{x}_n]$ that contain the (unknown) true values of the measured quantities. For different values $x_i \in \mathbf{x}_i$, we get, in general, different values of the corresponding statistical characteristic $C(x_1, \ldots, x_n)$. Since all values $x_i \in \mathbf{x}_i$ are possible, we conclude that all the values $C(x_1, \ldots, x_n)$ corresponding to $x_i \in \mathbf{x}_i$ are possible estimates for the corresponding statistical characteristic. Therefore, for the interval data $\mathbf{x}_1, \ldots, \mathbf{x}_n$, a reasonable estimate for the corresponding statistical characteristic is the range

$$C(\mathbf{x}_1, \ldots, \mathbf{x}_n) \stackrel{\text{def}}{=} \{C(x_1, \ldots, x_n) \mid x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n\}.$$

We must therefore modify the existing statistical algorithms so that they compute, or bound these ranges. This is the problem that we solve in this paper.

This problem is a part of a general problem. The above range estimation problem is a specific problem related to a combination of interval and probabilistic uncertainty. Such problems – and their potential applications – have been described, in a general context, in the monographs [27, 45]; for further developments, see, e.g., [2, 3, 4, 5, 7, 11, 28, 30, 40, 41, 48] and references therein.

2. Analysis of the Problem

Mean. Let us start our discussion with the simplest possible characteristic: the mean. The arithmetic average E is a monotonically increasing function of each of its n variables $x_1, \ldots, x_n$, so its smallest possible value $\underline{E}$ is attained when each value x_i is the smallest possible ($x_i = \underline{x}_i$) and its largest possible value is attained when $x_i = \overline{x}_i$ for all i. In other words, the range $\mathbf{E}$ of E is equal to $[E(\underline{x}_1, \ldots, x_n), E(\overline{x}_1, \ldots, \overline{x}_n)]$. In other words, $\underline{E} = \dfrac{1}{n} \cdot (\underline{x}_1 + \ldots + \underline{x}_n)$ and $\overline{E} = \dfrac{1}{n} \cdot (\overline{x}_1 + \ldots + \overline{x}_n)$.

Variance: computing the exact range is difficult. Another widely used statistic is the variance. In contrast to the mean, the dependence of the variance V on x_i is not monotonic, so the above simple idea does not work. Rather surprisingly, it turns out that the problem of computing the exact range for the variance over interval data is, in general, NP-hard [10, 24] which means, crudely speaking, that the worst-case computation time grows exponentially with n. Moreover, if we want to compute the variance range with a given accuracy ε, the problem is still NP-hard. (For a more detailed description of NP-hardness in relation to interval uncertainty, see, e.g., [19].)

Linearization. From the practical viewpoint, often, we may not need the exact range, we can often use approximate linearization techniques. For example, when the uncertainty comes from measurement errors Δx_i, and these errors are small, we can ignore terms that are quadratic (and of higher order) in Δx_i and get reasonable estimates for the corresponding statistical characteristics. In general, in order to estimate the range of the statistic $C(x_1, \ldots, x_n)$ on the intervals $[\underline{x}_1, \overline{x}_1]$, $\ldots$, $[\underline{x}_n, \overline{x}_n]$, we expand the function C in Taylor series at the midpoint $\widetilde{x}_i \stackrel{\text{def}}{=} (\underline{x}_i + \overline{x}_i)/2$ and keep only linear terms in this expansion. As a result, we replace the original statistic with its linearized version $C_{\text{lin}}(x_1, \ldots, x_n) = C_0 - \sum\limits_{i=1}^{n} C_i \cdot \Delta x_i$, where $C_0 \stackrel{\text{def}}{=} C(\widetilde{x}_1, \ldots, \widetilde{x}_n)$, $C_i \stackrel{\text{def}}{=} \dfrac{\partial C}{\partial x_i}(\widetilde{x}_1, \ldots, \widetilde{x}_n)$, and $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i$. For each i, when $x_i \in [\underline{x}_i, \overline{x}_i]$, the difference Δx_i can take all possible values from $-\Delta_i$ to Δ_i, where $\Delta_i \stackrel{\text{def}}{=} (\overline{x}_i - \underline{x}_i)/2$. Thus, in the linear approximation, we can estimate the range of the characteristic C as $[C_0 - \Delta, C_0 + \Delta]$, where $\Delta \stackrel{\text{def}}{=} \sum\limits_{i=1}^{n} |C_i| \cdot \Delta_i$.

In particular, if we take, as the statistic, the population variance $C = V$, then $C_i = \dfrac{\partial V}{\partial x_i} = \dfrac{2}{n} \cdot (\widetilde{x}_i - \widetilde{E})$, where $\widetilde{E}$ is the average of the midpoints $\widetilde{x}_i$, and $C_0 = \dfrac{1}{n} \cdot \sum\limits_{i=1}^{n} (\widetilde{x}_i - \widetilde{E})^2$ is the variance of the midpoint values $\widetilde{x}_1, \ldots, \widetilde{x}_n$. So, for the variance, $\Delta = \dfrac{2}{n} \cdot \sum\limits_{i=1}^{n} |\widetilde{x}_i - \widetilde{E}| \cdot \Delta_i$.

It is worth mentioning that for the variance, the ignored quadratic term is equal to $\dfrac{1}{n} \cdot \sum\limits_{i=1}^{n} (\Delta x_i)^2 - (\Delta E)^2$, where $\Delta E \stackrel{\text{def}}{=} \dfrac{1}{n} \cdot \sum\limits_{i=1}^{n} \Delta x_i$, and therefore, can be bounded by 0 from below and by $\Delta^{(2)} \stackrel{\text{def}}{=} \dfrac{1}{n} \cdot \sum\limits_{i=1}^{n} \Delta_i^2$ from above. Thus, the interval $[V_0 - \Delta, V_0 + \Delta + \Delta^{(2)}]$ is a guaranteed enclosure for $\mathbf{V}$.

Linearization is not always acceptable. In some cases, linearized estimates are not sufficient: the intervals may be wide so that quadratic terms can no longer be ignored, and/or we may be in a situation where we want to guarantee that, e.g., the variance does not exceed a certain required threshold. In such situations, we need to get the exact range – or at least an enclosure for the exact range.

Since, even for as simple a characteristic as variance, the problem of computing its exact range is NP-hard, we cannot have a feasible-time algorithm that always computes the exact range of these characteristics. Therefore, we must look for the reasonable classes of problems for which such algorithms are possible. Let us analyze what such classes can be.

3. Reasonable Classes of Problems for Which We Can Expect Feasible Algorithms for Statistics of Interval Data

First class: narrow intervals. As we have just mentioned, the computational problems become more complex when we have wider intervals. In other words, when intervals are narrower, the problems are easier. How can we formalize "narrow intervals"? One way to do it is as follows: the true values $x_1, \ldots, x_n$ of the measured quantity are real numbers, so they are usually different. The data intervals $\mathbf{x}_i$ contain these values. When the intervals $\mathbf{x}_i$ surrounding the corresponding points x_i are narrow, these intervals do not intersect. When their widths becomes larger than the distance between the original values, the intervals start intersecting.

Definition. Thus, the ideal case of "narrow intervals" can be described as the case when no two intervals $\mathbf{x}_i$ intersect.

Second class: slightly wider intervals. Slightly wider intervals correspond to the situation when few intervals intersect, i.e., when for some integer K, no set of K intervals has a common intersection.

Third class: single measuring instrument. Since we want to find the exact range $\mathbf{C}$ of a statistic C, it is important not only that intervals are relatively narrow, it is also important that they are approximately of the same size: otherwise, if, say, Δx_i^2 is of the same order as Δx_j, we cannot meaningfully ignore Δx_i^2 and retain Δx_j. In other words, the interval data set should not combine high-accurate measurement results (with narrow intervals) and low-accurate results (with wide intervals): all measurements should have been done by a single measuring instrument (or at least by several measuring instruments of the same type).

How can we describe this mathematically? A clear indication that we have two measuring instruments (MI) of different quality is that one interval is a proper subset of the other one: $[\underline{x}_i, \overline{x}_i] \subseteq (\underline{x}_j, \overline{x}_j)$.

This restriction only refers to inexact measurement results, i.e., to non-degenerate intervals. In additional to such interval values, we may also have values produced by very accurate measurements, so accurate that we can, for all practical purposes, consider these values exactly known. From this viewpoint, when we talk about measurements made by a single measuring instrument, we may allow degenerate intervals (i.e., exact numbers) as well.

As we will see, the absence of such pairs is a useful property that enables us to compute interval statistics faster. We will also see that this absence happens not only for measurements made by a single MI, but also in several other useful practical cases. Since this property is useful, we will give it a name.

Definition. We say that a collection of intervals satisfies a *subset property* if $[\underline{x}_i, \overline{x}_i] \not\subseteq (\underline{x}_j, \overline{x}_j)$ for all i and j for which the intervals $\mathbf{x}_i$ and $\mathbf{x}_j$ are non-degenerate.

Fourth class: same accuracy measurement. In some situations, it is also reasonable to consider a specific subcase of the single MI case when all measurements are performed with exactly the same accuracy.

After each measurement, we get the measurement result $\widetilde{x}_i$, and we conclude that the (unknown) true value x_i of the measured quantity belongs to the interval $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$, where $\underline{x}_i = \widetilde{x}_i - \Delta_i$ and $\overline{x}_i = \widetilde{x}_i + \Delta_i$.

In the above text, we characterized measurement uncertainty in terms of the (absolute) measurement error $\Delta x_i = \widetilde{x}_i - x_i$. In this case, the upper bound Δ_i on the absolute value $|\Delta x_i|$ of the measurement error is a natural measure of the measurement accuracy. In these terms, the case of same accuracy measurements can be described as the case when all these upper bounds coincide: $\Delta_1 = \ldots = \Delta_n$.

We have mentioned that the single MI case covers not only the situation when the intervals come from measurements, but other important situations as well. How can we describe this same accuracy property in the general case, when we are simply given n intervals $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$ that do not necessarily come from measurements?

In the case when the interval $\mathbf{x}_i$ results from a measurement, the value Δ_i is the half-width of the interval. Thus, in general, the case of "same accuracy" measurements can be described as the case in which all non-degenerate intervals $[\underline{x}_i, \overline{x}_i]$ have exactly the same half-width $\Delta_i = \dfrac{1}{2} \cdot (\overline{x}_i - \underline{x}_i)$.

Comment. Sometimes, it is reasonable to describe measurement errors in *relative* terms – as accuracy, say, 5% instead of 0.1 units; (see, e.g., [39]). A *relative measurement error* is defined as the ratio $\delta x_i \stackrel{\text{def}}{=} \Delta x_i / |\widetilde{x}_i|$. Within this description, as a natural measure of the measurement accuracy, we can take the largest possible absolute value $|\delta x_i|$ of the relative error, i.e., the value $\delta_i = \Delta_i / \widetilde{x}_i$.

In such situations, it is reasonable to consider the case when all the measurements are of the same *relative* accuracy, i.e., in which all non-degenerate intervals $[\underline{x}_i, \overline{x}_i]$ have exactly the same ratio $\delta_i = \Delta_i / |\widetilde{x}_i|$ between the half-width and the midpoint. One can easily check that this condition is equivalent to the condition that all non-degenerate intervals have the same ratio $\overline{x}_i / \underline{x}_i$.

This case is yet another subcase of the single MI case; it may be beneficial to check whether any of our algorithms can be simplified when restricted to this subcase.

Fifth class: several MI. After the single MI case, the natural next case is when we have several MI, i.e., when our intervals are divided into several subgroups each of which has the above-described subset property.

Comment. The need to use multiple MI comes from the interval uncertainty. Indeed, as sample size increases, a point of diminishing returns is reached when observations are interval bounds that contain measurement error. For example, when constructing a confidence interval on E, we expect its width to monotonically decrease as a function of sample size. In the presence of interval bounded measurement errors, a point will be reached where increasing sample size has almost no impact. Indeed, in the limit, as $n \to \infty$, the confidence interval will be the width of measurement error intervals.

If, however, it is possible to use multiple measuring instruments to produce multiple intervals for each observation, then these multiple measurements can be intersected to reduce the interval width due to measurement error on each observation; see, e.g., [46, 47]. In this way, effort could be balanced between increasing sample size and increasing the number of measuring instruments used for each observation.

Sixth class: privacy case. Although these definitions are in terms of measurements, they make sense for other sources of interval data as well. For example, for privacy data, intervals either coincide (if the value corresponding to the two patients belongs to the same range) or are different, in which case they can only intersect in one point. Similarly to the above situation, we also allow exact values in addition to ranges; these values correspond, e.g., to the exact records made in the past, records that are already in the public domain.

Definition. We will call interval data with this property – that every two non-degenerate intervals either coincide or intersect at most in one point – *privacy case.*

Comment. For the privacy case, the subset property is satisfied, so algorithms that work for the subset property case work for the privacy case as well.

Comment. Sometimes, in the privacy-motivated situation, we must process interval data in which intervals come from several different "granulation" schemes. For example, to find the average salary in North America, we may combine US interval records in which the salary is from 0 to 10,000 US dollars, from 10,000 to 20,000, etc., with the Canadian interval records in which the ranges are between 0 to 10,000 Canadian dollars, 10,000 to 20,000 Canadian dollars, etc. When we transform these records to a single unit, we get two different families of intervals, each of which satisfies the subset property. Thus, to handle such situations, we can use algorithms develop for the several MI case.

Seventh class: non-detects. Similarly, if the only source of interval uncertainty is detection limits, i.e., if every measurement result is either an exact value or a *non-detect*, i.e., an interval $[0, DL_i]$ for some real number DL_i (with possibly different detection limits for different sensors), then the resulting non-degenerate intervals also satisfy the subset property. Thus, algorithms that work for the subset property case work for this "non-detects" case as well.

Also, an algorithm that works for the general privacy case also works for the non-detects case when all sensors have the same detection limit DL.

Let us now describe the known algorithms for statistics of interval data.

4. Results

4.1. Variance: Lower Bound

Known result: in brief. The lower bound $\underline{V}$ can be always computed in time $O(n \cdot \log(n))$ [14].

Main idea behind this result. The algorithm for computing $\underline{V}$ is based on the fact that when a function V attains a minimum on an interval $[\underline{x}_i, \overline{x}_i]$, then either $\dfrac{\partial V}{\partial x_i} = 0$, or the minimum is attained at the left endpoint $x_i = \underline{x}_i$ – then $\dfrac{\partial V}{\partial x_i} > 0$, or the minimum is attained at the right endpoint $x_i = \overline{x}_i$ and $\dfrac{\partial V}{\partial x_i} < 0$. Since the partial derivative is equal to $(2/n) \cdot (x_i - E)$, we conclude that either $x_i = E$, or $x_i = \underline{x}_i > E$, or $x_i = \underline{x}_i < E$. Thus, if we know where E is located in relation to all the endpoints, we can uniquely determine the corresponding minimizing value x_i for every i: if $\overline{x}_i \le E$ then $x_i = \overline{x}_i$; if $x_i \le \underline{x}_i$, then $x_i = \underline{x}_i$; otherwise, $x_i = E$. The corresponding value E can be found from the condition that E is the average of all the selected values x_i.

So, to find the smallest value of V, we can sort all $2n$ bounds $\underline{x}_i, \overline{x}_i$ into a sequence $x_{(1)} \le x_{(2)} \le \ldots$; then, for each zone $[x_{(k)}, x_{(k+1)}]$, we compute the corresponding values x_i, find their variance V_k, and then compute the smallest of these variances V_k.

As we have mentioned, the corresponding value E can be found from the condition that E is the average of all the selected values x_i. If E is in the zone $[x_{(k)}, x_{(k+1)}]$, then we know all the values x_i, so $n \cdot E$ should be equal to the sum of these values:

$$n \cdot E = \sum_{i : \underline{x}_i \ge x_{(k+1)}} \underline{x}_i + (n - N_k) \cdot E + \sum_{j : \overline{x}_j \le x_{(k)}} \overline{x}_j,$$

where by N_k, we denoted the total number of such i's for which $\underline{x}_i \geq x_{(k+1)}$ and j's for which $\underline{x}_j \leq x_{(k)}$.

Subtracting $(n - N_k) \cdot E$ from both sides of this equality, we conclude that $N_k \cdot E = S_k$, where

$$S_k \stackrel{\text{def}}{=} \sum_{i:\underline{x}_i \geq x_{(k+1)}} \underline{x}_i + \sum_{j:\overline{x}_j \leq x_{(k)}} \overline{x}_j.$$

If $N_k = 0$, this means that $x_i = E$ for all i, so $V = 0$. If $N_k \neq 0$, then $E = S_k / N_k$.

Once E is computed, we can now compute the corresponding variance V_k as $M_k - E^2$, where M_k is the second population moment:

$$M_k = \frac{1}{n} \cdot \sum_{i:\underline{x}_i \geq x_{(k+1)}} (\underline{x}_i)^2 + \frac{n - N_k}{n} \cdot E^2 + \frac{1}{n} \cdot \sum_{j:\overline{x}_j \leq x_{(k)}} (\overline{x}_j)^2,$$

i.e., $V_k = M'_k - \dfrac{N_k}{n} \cdot E^2$, where

$$M'_k \stackrel{\text{def}}{=} \frac{1}{n} \cdot \left(\sum_{i:\underline{x}_i \geq x_{(k+1)}} (\underline{x}_i)^2 + \sum_{j:\overline{x}_j \leq x_{(k)}} (\overline{x}_j)^2 \right).$$

How many steps do we need for this algorithm? Sorting requires $O(n \cdot \log(n))$ steps. Computing the initial values of S_k, N_k, and M'_k requires linear time, i.e., $O(n)$ steps.

For each k, the values S_k, N_k, and M'_k differ from the previous value by only one or two terms – namely, e.g., the values i for which $\underline{x}_i \geq x_{(k)}$ but $\underline{x}_i < x_{(k+1)}$. In other words, the only change is for i for which $x_{(k)} \leq \underline{x}_i < x_{(k+1)}$. Since $x_{(k)}$ is the ordering of all lower and upper bounds, this means that $x_{(k)} = \underline{x}_i$.

Similarly, the only change in the second sum is the term for which $\overline{x}_j = x_{(k)}$.

So, each of these values $S_k, \ldots$, can be computed from the previous values $S_{k-1}, \ldots$ in a constant number of steps. Thus, the overall number of steps for computing them is linear in n. The smallest of the values V_k is the desired $\underline{V}$. Thus, we can compute $\underline{V}$ in $O(n \cdot \log(n)) + O(n) = O(n \cdot \log(n))$ steps.

Comment. If two bounds happen to coincide, then for the corresponding k, we may have a difference of several values between S_k and S_{k-1}. However, each of the $2n$ bounds can occur only once in this change, so the overall number of terms is still $O(n)$.

How good is this algorithm? Since even simple sorting requires at least $O(n \cdot \log(n))$ steps, algorithms like this, that compute a bound of a statistical interval characteristic in $O(n \cdot \log(n))$ steps, can be considered a "golden standard" for such algorithms.

4.2. Variance: Upper Bound

General case. We have already mentioned that computing $\overline{V}$ is, in general, an NP-hard problem.

A new NP-hardness result. In the original proof of NP-hardness, we have $\widetilde{x}_1 = \ldots = \widetilde{x}_n = 0$, i.e., all measurement results are the same, only accuracies Δ_i are different. What if all the measurement results are different? We can show that in this case, computing $\overline{V}$ is still an NP-hard problem: namely, for every n-tuple of real numbers $\widetilde{x}_1, \ldots, \widetilde{x}_n$, the problem of computing $\overline{V}$ for intervals $\mathbf{x}_i = [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$ is still NP-hard.

To prove this result, it is sufficient to consider $\Delta_i = N \cdot \Delta_i^{(0)}$, where $\Delta_i^{(0)}$ are the values used in the original proof and N is a large integer (that will be selected later). In this case, we can describe $\Delta x_i = \widetilde{x}_i - x_i$ as $N \cdot \Delta x_i^{(0)}$, where $\Delta_i^{(0)} \in [-\Delta_i^{(0)}, \Delta_i^{(0)}]$. For large N, the difference between the variance corresponding to the values $x_i = \widetilde{x}_i + N \cdot \Delta x_i^{(0)}$ and N^2 times the variance of the values $\Delta x_i^{(0)}$ is bounded by a term proportional to N (and the coefficient at N can be easily bounded). Thus, the difference between $\overline{V}$ and $N^2 \cdot \overline{V}^{(0)}$ is bounded by $C \cdot N$ for some known constant C. Hence, by computing $\overline{V}$ for sufficiently large N, we can compute $\overline{V}^{(0)}$ with a given accuracy $\varepsilon > 0$, and we already know that computing $\overline{V}^{(0)}$ with given accuracy is NP-hard. This reduction proves that our new problem is also NP-hard.

How to compute the upper bound: general case. It is known that the maximum of a quadratic function on an interval is always attained at one of the endpoints. Thus, in principle, we can always compute the upper bound $\overline{V}$ in time 2^n: namely, it is sufficient to compute the variance V for all 2^n possible vectors $x = (x_1^{\varepsilon_1}, \ldots, x_n^{\varepsilon_n})$, where $\varepsilon_i \in \{-,+\}$, $x_i^- = \underline{x}_i$ and $x_i^+ = \overline{x}_i$ – then the largest of these 2^n values is the desired value $\overline{V}$.

Cases of narrow intervals and slightly wider intervals. For $\overline{V}$, we can provide an analysis of the derivatives which is similar to the analysis provided for $\underline{V}$. For $\overline{V}$, to this analysis, we can add the fact that the second derivative of V is ≥ 0, so there cannot be a maximum inside the interval $[\underline{x}_i, \overline{x}_i]$.

So, when $\overline{x}_i \leq E$, we take $x_i = \underline{x}_i$; when $E \leq \underline{x}_i$, we take $x_i = \overline{x}_i$; otherwise, we must consider both possibilities $x_i = \underline{x}_i$ and $x_i = \overline{x}_i$.

When intervals do not intersect, we thus end up with an $O(n \cdot \log(n))$ algorithm for computing $\overline{V}$. It turns out that a $O(n \cdot \log(n))$ algorithm is possible not only when the original intervals $[\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$ do not intersect, but also in a more general case when the "narrowed" intervals $[\widetilde{x}_i - \Delta_i/n, \widetilde{x}_i + \Delta_i/n]$ do not intersect. In fact, a $O(n \cdot \log(n))$ algorithm is even possible in the case when for some integer $K < n$, no sub-collection of greater than K narrowed intervals of $\mathbf{x}_i$ has a common intersection [50].

Case of the subset property. For the case of the subset property, we can sort the intervals in lexicographic order: $\mathbf{x}_i \leq \mathbf{x}_j$ if and only if $\underline{x}_i < \underline{x}_j$ or ($\underline{x}_i = \underline{x}_j$ and $\overline{x}_i \leq \overline{x}_j$).

It can be proven that the maximum of V is always attained if for some k, the first k values x_i are equal to $\underline{x}_i$ and the next $n-k$ values x_i are equal to $\overline{x}_i$. This result is proven by reduction to a contradiction: if in the maximizing vector $x = (x_1, \ldots, x_n)$, some $\overline{x}_i$ is preceding some $\underline{x}_j$, $i < j$, then we can increase V while keeping E intact – which is in contradiction with the assumption that the vector x was maximizing. Specifically, to increase V, we can do the following: if $\Delta_i \leq \Delta_j$, we replace $\overline{x}_i$ with $\underline{x}_i = \overline{x}_i - 2\Delta_i$ and $\underline{x}_j$ with $\underline{x}_j + 2\Delta_i$; otherwise, we replace $\underline{x}_j$ with $\overline{x}_j = \underline{x}_j + 2\Delta_j$ and $\overline{x}_i$ with $\overline{x}_i - 2\Delta_j$.

As a result, we arrive at the following algorithm: first, we sort the intervals $[\underline{x}_i, \overline{x}_i]$ in lexicographic order; then, for $k = 0, 1, \ldots, n$, compute the value $V = M - E^2$ for the corresponding vectors $x^{(k)} = (\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$. When we go from a vector $x^{(k)}$ to the vector $x^{(k+1)}$, only one term changes in the vector x, so only one term changes in each of the sums E and M.

How good is this algorithm? Sorting takes $O(n \cdot \log(n))$ time; computing the initial values of E and M requires linear time $O(n)$. For each k, computing the new values of E and M requires a constant number of steps, so overall, computing all n values of E, M (and hence V) requires linear time. Thus, the overall time of this algorithm is $O(n \cdot \log(n))$.

Comment. In our proof, we used a technique of replacing two values in such a way that their sum (and hence, the overall average) remain unchanged. According to [13], this technique, called a *transfer*, was first introduced by Robert Muirhead in 1903. The transfer technique is actively used both in *mathematics*, where it is one of the main tools in proving inequalities (see, e.g., [15]), and in *economics*, where it has been used by several major economists (e.g., by Hugh Dalton) as a basis of their economic theories and results.

Case of several MI. In case of several MI, we can similarly prove that if we sort the intervals corresponding to each MI in lexicographic order, then the maximum of V is attained when from intervals corresponding to each MI, the values x_i corresponding to this MI form a sequence $(\underline{x}_1, \ldots, \underline{x}_{k_j}, \overline{x}_{k_j+1}, \ldots, \overline{x}_{n_j})$, where n_j is the total number of intervals corresponding to the j-th MI.

Thus, to find the maximum of V, we must find the values $k_1, \ldots, k_m$ corresponding to m MIs. For these values, $V = M - E^2$, where $M = \sum M_j$ and $E = \sum E_j$, where we denoted by E_j and M_j, the averages of, correspondingly, x_i and x_i^2, taken by using only results of j-th MI.

For each MI j, we can compute all $n_j + 1$ possible values E_j and M_j in linear time.

There are $\le n^m$ combinations of k_is; for each combination, we need m additions to compute $E = \sum E_j$, m additions to compute $M = \sum M_j$, and a constant number of operations to compute $V = M - E^2$. Thus, overall, we need time $O(n^m)$.

Cases of privacy and non-detects. Since these two cases are a particular case of the subset property case, and for the subset property case, we have an $O(n \cdot \log(n))$ algorithm, this same algorithm can be applied to these two cases as well.

Case when only some intervals are non-degenerate. Sometimes, most of the data is accurate, so among n intervals, only $d \ll n$ are non-degenerate intervals. For example, we can have many accurate values and d non-detects.

In this situation, to find the extrema of V, we only need to find x_i for d non-degenerate intervals; thus, we only need to consider $2d$ zones formed by their endpoints.

To compute $\underline{V}$, we need time $O(d \cdot \log(d))$ to sort $2d$ endpoints, time $O(n)$ to produce the initial values of S_k, N_k, and M'_k, and then time $O(d)$ to compute all the other values S_k, N_k, and M'_k – and the corresponding values V_k. Since $d \le n$, the resulting time is $O(d \cdot \log(d) + n)$.

To compute $\overline{V}$, in the general case, we only have to consider possible combinations of d endpoints, so the overall time is $n + 2^d$ instead of 2^n.

For the case of feasible algorithms, similarly to computing $\underline{V}$, for computing $\overline{V}$ for classes 1, 2, and 3, we need time $O(d \cdot \log(d) + n)$ (hence we need the same time for classes 4, 6, and 7).

For class 5 (with m MI), we need $O(d \cdot \log(d))$ steps for sorting, $O(n)$ steps for the original arrangement, and d^m steps to compute the values for all d^m possible combinations of k_j. For $m \geq 2$, we have $d \cdot \log(d) \leq d^m$, hence the overall time is $O(n + d^m)$.

4.3. Covariance

What is covariance. When we two different measurement results x_i and y_i for each measurement i, then an important statistical characteristic is the covariance $C_{xy} = \frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - E_x) \cdot (y_i - E_y)$, where $E_x = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i$ and $E_y = \frac{1}{n} \cdot \sum_{i=1}^{n} y_i$ are the corresponding population averages. The covariance can also be described as $C_{xy} = M_{xy} - E_x \cdot E_y$, where $M_{xy} \stackrel{\text{def}}{=} \frac{1}{n} \cdot \sum_{i=1}^{n} x_i \cdot y_i$ is the second mixed moment.

General case. In general, computing the range of the covariance C_{xy} based on given intervals $\mathbf{x}_i$ and $\mathbf{y}_i$ is NP-hard [38].

Cases of narrow intervals and slightly wider intervals. When boxes $\mathbf{x}_i \times \mathbf{y}_i$ do not intersect – or if $\geq K$ boxes cannot have a common point – we can compute the range in feasible time [1].

The main idea behind the corresponding algorithm is to consider the derivatives of C_{xy} relative to x_i and y_i. Then, once we know where the point (E_x, E_y) is in relation to x_i and y_i, we can uniquely determine the optimizing values x_i and y_i – except for the boxes $\mathbf{x}_i \times \mathbf{y}_i$ that contain (E_x, E_y).

The bounds $\underline{x}_i$ and $\overline{x}_i$ divide the x axis into $2n+2$ intervals; similarly, the y-bounds divide the y-axis into $2n + 2$ intervals. Combining these intervals, we get $O(n^2)$ zones.

Due to the limited intersection property, for each of these zones, we have finitely many ($\leq K$) indices i for which the corresponding box intersects with the zone. For each such box, we may have two different combinations: $(\underline{x}_i, \underline{y}_i)$ or $(\overline{x}_i, \overline{y}_i)$ for $\overline{C}_{xy}$, $(\underline{x}_i, \overline{y}_i)$ or $(\overline{x}_i, \underline{y}_i)$ for $\underline{C}_{xy}$. Thus, we have finitely many ($\leq 2^K$) possible combinations of (x_i, y_i) corresponding to each zone.

When we move from a zone to the next one, each sum M_{xy}, E_x, and E_y changes by a single term. Thus, for each zone, we need to perform finitely many steps to update these values and to find the corresponding values of the covariances. Thus, to cover all $O(n^2)$ zones, we need $O(n^2)$ time.

Case of same accuracy measurements. Another polynomial-time case is when all the measurements are exactly of the same accuracy, i.e., when all non-degenerate x-intervals have the same half-width Δ_x, and all non-degenerate y-intervals have the same half-width Δ_y.

In this case, e.g., for $\overline{C}_{xy}$, if we have at least two boxes i and j intersecting with the same zone, and we have $(x_i, y_i) = (\underline{x}_i, \underline{y}_i)$ and $(x_j, y_j) = (\overline{x}_j, \overline{y}_j)$, then we can swap i and j assignments – i.e., make $(x'_i, y'_i) = (\overline{x}_i, \overline{y}_i)$ and $(x'_j, y'_j) = (\underline{x}_j, \underline{y}_j)$ – without changing E_x and E_y. In this case, the only change in C_{xy} comes from replacing $x_i \cdot y_i + x_j \cdot y_j$. It is easy to see that the new value C is larger than the old value if and only if $z_i > z_j$, where $z_i \stackrel{\text{def}}{=} \widetilde{x}_i \cdot \Delta_y + \widetilde{y}_i \cdot \Delta_x$.

Thus, in the true maximum, whenever we assign $(\underline{x}_i, \underline{y}_i)$ to some i and $(\overline{x}_i, \overline{y}_j)$ to some j, we must have $z_i \le z_j$.

So, to get the largest value of C_{xy}, we must: sort the indices by z_i, select a threshold t, and assign $(\underline{x}_i, \underline{y}_i)$ to all the boxes with $z_i \le t$ and $(\overline{x}_j, \overline{y}_j)$ to all the boxes j with $z_j > t$.

If $n_k \le n$ denotes the overall number of all the boxes that intersect with k-th zone, then we have $n_k + 1$ possible choices of thresholds, hence $n_k + 1$ such assignments.

For each of $O(n^2)$ zones, we test $\le n$ assignments; to the total of $O(n^3)$. Computing each assignment from the previous one requires a constant number of steps, so overall, we need time $O(n^3)$.

Privacy case. In the privacy case, all boxes $\mathbf{x}_i \times \mathbf{y}_i$ are either identical or non-intersecting, so the only case when a box intersects with a zone is when the box coincides with this zone.

For each zone k, there may be many (n_k) such boxes, but since they are all identical, what matters for our estimates is how many of them are assigned one of the possible (x_i, y_i) combinations and how many the other one. There are only n_k+1 such assignments: 0 to first combination and n_k to second, 1 to first and $n_k - 1$ to second, etc. Thus, the overall number of all combinations for all the zones k is $\sum\limits_k n_k + \sum\limits_k 1$, where $\sum n_k = n$ and $\sum\limits_k 1$ is the overall number of zones, i.e., $O(n^2)$.

For the original combination of x_i and y_i, we need $O(n)$ steps. Moving from one combination to another means changing only one term in each sum M_{xy}, E_x, E_y, thus, computing each combination requires a constant number of steps – to the total of $O(n^2)$.

Thus, in the privacy case, we can compute both $\underline{C}_{xy}$ and $\overline{C}_{xy}$ in time $O(n^2) + O(n) = O(n^2)$.

Case when only some intervals are non-degenerate. For *narrow intervals*, if $n - d$ measurement results (x_i, y_i) are exact numbers and only

d are non-point boxes, then we only need $O(d^2)$ zones. So, we need $O(d \cdot \log(d))$ time for sorting, $O(n)$ to compute the initial values of M_{xy}, E_x, and E_y, and $O(d^2)$ time to compute the values C_{xy} for all the zones – to the total of $O(n + d^2)$.

In the case of *same accuracy measurements*, if only d boxes are non-degenerate, we need sorting (which takes time $O(d \cdot \log(d))$), initial computation (which takes time $O(n)$), and checking all $O(d^3)$ assignments – so the overall time is $O(n + d^3)$.

In the *privacy* case, similarly to the case of narrow intervals, we have $O(d^2)$ zones, so we also need time $O(n + d^2)$.

4.4. Population Moments

For population moments $\dfrac{1}{n} \cdot \sum\limits_{i=1}^{n} x_i^q$, known interval bounds on x^q leads to exact range.

4.5. Central Moments of Even Order

Definition. A population central moment is defined as
$M_q = \dfrac{1}{n} \cdot \sum\limits_{i=1}^{n} (x_i - E)^q$.

Computing $\underline{M}_q$. For even q, we have $\dfrac{\partial M_q}{\partial x_i} = \dfrac{q}{n} \cdot (x_i - E)^{q-1} - q \cdot M_{q-1}$, so $\dfrac{\partial M_q}{\partial x_i} \geq 0 \leftrightarrow x_i \geq \lambda \stackrel{\text{def}}{=} E + (q \cdot M_{q-1})^{1/(q-1)}$.

Thus, once we know where λ is located w.r.t. the endpoints, we can find all x_i – and find λ from the condition that this value λ is equal to the values $E + (q \cdot M_{q-1})^{1/(q-1)}$ computed based on the resulting sample.

The value M_{q-1} can be computed based on the corresponding population moments up to order $q - 1$. Once we know the moments for one zone, we recomputing the moment for the next zone requires constant time.

Thus, to find $\underline{M}_q$, we must sort the endpoints (which takes time $O(n \cdot \log(n))$), compute the original values of the moments (time $O(n)$), and then compute the moments for all zones (time $O(n)$)) – overall time is $O(n \cdot \log(n))$.

Computing $\overline{M}_q$ for narrow and slightly wider intervals. In this case, a similar analysis of partial derivatives leads to an $O(n \cdot \log(n))$ algorithm.

Computing $\overline{M}_q$ *the subset property case: idea.* In this case, similarly to the variance, we can prove that the maximum is always attained at one of the vectors $x = (\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$.

The following proof works not only for M_q, but also for a generalized central moment $M_\psi \stackrel{\text{def}}{=} \frac{1}{n} \cdot \sum_{i=1}^{n} \psi(x_i - E)$, where $E = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i$ and $\psi(x) \geq 0$ is an (arbitrary) convex function for which $\psi(0) = 0$ and $\psi''(x) > 0$ for all $x \neq 0$.

Let us first show that the maximum cannot be attained inside an interval $[\underline{x}_i, \overline{x}_i]$.

Indeed, in this case, at the maximizing point, the first derivative

$$\frac{\partial M_\psi}{\partial x_i} = \frac{1}{n} \cdot \psi'(x_i - E) - \frac{1}{n^2} \cdot \sum_{j=1}^{n} \psi'(x_j - E)$$

should be equal to 0, and the second derivative

$$\frac{\partial^2 M_\psi}{\partial x_i^2} = \frac{1}{n} \cdot \psi''(x_i - E) \cdot \left(1 - \frac{2}{n}\right) + \frac{1}{n^3} \cdot \sum_{j=1}^{n} \psi''(x_j - E)$$

is non-positive. Since the function $\psi(x)$ is convex, we have $\psi''(x) \geq 0$, so this second derivative is a sum of non-negative terms, and the only case when it is non-negative is when all these terms are 0s, i.e., when $x_j = E$ for all j. In this case, $M_\psi = 0$ which, for non-degenerate intervals, is clearly not the largest possible value of M_ψ.

So, for every i, the maximum of M_ψ is attained either when $x_i = \underline{x}_i$ or when $x_i = \overline{x}_i$.

Similarly to the proof for the variance, we will now prove that the maximum is always attained for one of the vectors $(\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$.

To prove this, we need to show that if $x_i = \overline{x}_i$ and $x_j = \underline{x}_j$ for some $i < j$ (and $\underline{x}_i \leq \underline{x}_j$), then the change described in that proof, while keeping the average E intact, increases the value of M_ψ. Without losing generality, we can consider the case $\Delta_i \leq \Delta_j$. In this case, the fact that M_ψ increase after the above-described change is equivalent to: $\psi(\underline{x}_i + 2\Delta_i - E) + \psi(\underline{x}_j - E) \leq \psi(\underline{x}_i - E) + \psi(\underline{x}_j + 2\Delta_i - E)$, i.e., that $\psi(\underline{x}_i + 2\Delta_i - E) - \psi(\underline{x}_i - E) \leq \psi(\underline{x}_j + 2\Delta_j - E) - \psi(\underline{x}_j - E)$. Since $\underline{x}_i \leq \underline{x}_j$ and $\underline{x}_i - E \leq \underline{x}_j - E$, this can be proven if we show that for every $\Delta > 0$ (and, in particular, for $\Delta = 2\Delta_i$), the function $\psi(x + \Delta) - \psi(x)$ is increasing. Indeed, the derivative of this function is equal to $\psi'(x+\Delta) - \psi'(x)$, and since $\psi''(x) \geq 0$, we do have $\psi'(x+\Delta) \geq \psi'(x)$.

Computing $\overline{M}_q$ the subset property case: algorithm. In view of the above result, to find $\overline{M}_\psi$, it is sufficient to check all n vectors of the type $(\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$, which, as we have shown, requires $O(n \cdot \log(n))$ steps. For m MIs, we similarly need $O(n^m)$ steps.

Case when only $d < n$ intervals are non-degenerate. If only d out of n intervals are non-degenerate, then we need $O(n+2^d)$ time instead of $O(2^n)$, and $O(n + d \cdot \log(d))$ instead of $O(n \cdot \log(n))$.

4.6. Central Moments of Odd Order

For odd q, the formula for the derivative has the same form $\dfrac{\partial M_q}{\partial x_i} = \dfrac{q}{n} \cdot (x_i - E)^{q-1} - q \cdot M_{q-1}$, but due to the fact q is odd, it leads to a more complex description of the condition $\dfrac{\partial M_q}{\partial x_i} \geq 0$: it is equivalent to $x_i \geq \lambda^+$ or $x_i \leq \lambda^-$, where $\lambda^\pm \stackrel{\text{def}}{=} E \pm (q \cdot M_{q-1})^{1/(q-1)}$.

Thus, to find all the values x_i, instead of knowing a single zone where λ lies, we now need to know *two* zones: a zone containing λ^- and a zone containing λ^+. There are $O(n^2)$ such pairs of zones, and each needs to be tried. So, for odd q, if $\leq K$ intervals do not intersect, we can compute both $\underline{M}_q$ and $\overline{M}_q$ in time $O(n^2)$.

If only d out of n intervals are non-degenerate, then we need $O(n + 2^d)$ time instead of $O(2^n)$, and $O(n+d^2)$ instead of $O(n^2)$.

4.7. Confidence Intervals and Outliers

What we are going to compute. Traditionally, in statistics, we fix a value k_0 (e.g., 2 or 3) and claim that every value x outside the k_0-sigma interval $[L, U]$, where $L \stackrel{\text{def}}{=} E - k_0 \cdot \sigma$, $U \stackrel{\text{def}}{=} E + k_0 \cdot \sigma$ (and $\sigma \stackrel{\text{def}}{=} \sqrt{V}$), is an outlier.

Thus, to detect outliers based on interval data, we must know the ranges of L and U.

The values L and U can also be viewed as bound for the confidence intervals, so by checking outliers, we thus estimate the confidence interval as well.

Comment. Previously, we have mainly considered *descriptive* statistics – statistics used to describe the sample.

L and U are examples of *inferential* statistics – statistics that are used to make conclusions about the data (in this case, whether a given data point is an outlier).

Why cannot we simply combine the intervals for E and $\sigma = \sqrt{V}$. In principle, we can use the general ideas of interval computations to combine these intervals and conclude, e.g., that U always belongs to the interval $\mathbf{E} + k_0 \cdot [\underline{\sigma}, \overline{\sigma}]$. However, as often happens in interval computations, the resulting interval for U is *wider* than the actual range – wider because the values E and σ are computed based on the same inputs $x_1, \ldots, x_n$ and cannot, therefore, change independently.

As an example that we may lose precision by combining intervals for E and σ, let us consider the case when $\mathbf{x}_1 = \mathbf{x}_2 = [0,1]$ and $k_0 = 2$. In this case, the range $\mathbf{E}$ of $E = (x_1 + x_2)/2$ is equal to $[0,1]$, where the largest value 1 is attained only if $x_1 = x_2 = 1$. For the variance, we have $V = ((x_1 - E)^2 + (x_2 - E)^2)/2 = (x_1 - x_2)^2/4$; so, the range $\mathbf{V}$ of V is $[0, 0.25]$ and, correspondingly, the range for $\sigma = \sqrt{V}$ is $[0, 0.5]$. The largest value $\sigma = 0.5$ is only attained in two cases: when $x_1 = 0$ and $x_2 = 1$, and when $x_1 = 1$ and $x_2 = 0$. When we simply combine the intervals, we conclude that $U \in [0,1] + 2 \cdot [0, 0.5] = [0, 2]$. However, it is easy to see that U cannot be equal to 2:

– The only way for U to be equal to 2 is when both E and σ attain their largest values: $E = 1$ and $\sigma = 0.5$.

– However, the only pair on which the mean E attains its largest value 1 is $x_1 = x_2 = 1$, and for this pair, $\sigma = 0$.

So, in this case, the actual range of U is narrower than the result $[0,2]$ of combining intervals for E and σ.

Computing $\underline{U}$ and $\overline{L}$: general case. There is a feasible algorithm for computing $\underline{U}$ and $\overline{L}$; see, e.g., [23, 24, 26].

The idea of such an algorithm is similar to the idea of an algorithm for computing $\underline{V}$. It comes from the fact that the minimum of a differentiable function of x_i on an interval $[\underline{x}_i, \overline{x}_i]$ is attained either inside this interval or at one of the endpoints. If the minimum is attained inside, the derivative $\dfrac{\partial U}{\partial x_i}$ is equal to 0; if it is attained at $x_i = \underline{x}_i$, then $\dfrac{\partial U}{\partial x_i} \geq 0$; finally, if it is attained at $x_i = \overline{x}_i$, then $\dfrac{\partial U}{\partial x_i} \leq 0$. For our function, $\dfrac{\partial U}{\partial x_i} = \dfrac{1}{n} + k_0 \cdot \dfrac{x_i - E}{\sigma \cdot n}$; thus, $\dfrac{\partial U}{\partial x_i} = 0$ if and only if $x_i = \lambda \stackrel{\text{def}}{=} E - \alpha \cdot \sigma$; similarly, the non-positiveness and non-negativeness of the derivative can be described by comparing x_i with λ. So, either $x_i \in (\underline{x}_i, \overline{x}_i)$ and $x_i = \lambda$, or $x_i = \underline{x}_i$ and $x_i = \underline{x}_i \geq \lambda$, or $x_i = \overline{x}_i$ and $x_i = \overline{x}_i \leq \lambda$.

Hence, if we know how the value λ is located with respect to all the intervals $[\underline{x}_i, \overline{x}_i]$, we can find the optimal values of x_i: if $\overline{x}_i \leq \lambda$, then minimum cannot be attained inside or at the lower endpoint, so it is attained when $x_i = \overline{x}_i$; if $\lambda \le \underline{x}_i$, then, similarly, the minimum is attained when $x_i = \underline{x}_i$; if $\underline{x}_i < \lambda < \overline{x}_i$, then the minimum is attained when $x_i = \lambda$. So, to find the minimum, we will analyze how the endpoints $\underline{x}_i$ and $\overline{x}_i$ divide the real line, and consider all the resulting zones.

Let the corresponding zone $[x_{(k)}, x_{(k+1)}]$ be fixed. For the i's for which $\lambda \notin (\underline{x}_i, \overline{x}_i)$, the values x_i that correspond to the minimal sample variance are uniquely determined by the above formulas.

For the i's for which $\lambda \in (\underline{x}_i, \overline{x}_i)$, the selected value x_i should be equal to the same value λ. To determine this λ, we will use the fact that, by definition, $\lambda = E - \alpha \cdot \sigma$, where E and σ are computed by using the same value of λ. This equation is equivalent to $E - \lambda \ge 0$ and $\alpha^2 \cdot \sigma^2 = (\lambda - E)^2$. Substituting the above values of x_i into the formula for the mean E and for the standard deviation σ, we get the quadratic equation for λ. So, for each zone, we can uniquely determine the values x_i that may correspond to a minimum of U.

For the actual minimum, the value λ is inside one of these zone, so the smallest of the values U_k is indeed the desired minimum.

The resulting algorithms $\underline{\mathcal{A}}_U$ for computing $\underline{U}$ and $\overline{\mathcal{A}}_L$ for computing $\overline{L}$ are as follows [24]. First, we sort all $2n$ values $\underline{x}_i$, $\overline{x}_i$ into a sequence $x_{(1)} \le x_{(2)} \le \ldots \le x_{(2n)}$; take $x_{(0)} = -\infty$ and $x_{(2n+1)} = +\infty$. For each of these zones $[x_{(k)}, x_{(k+1)}]$, $k = 0, 1, \ldots, 2n$, we compute the values

$$e_k \stackrel{\text{def}}{=} \sum_{i:\underline{x}_i \ge x_{(k+1)}} \underline{x}_i + \sum_{j:\overline{x}_j \le x_{(k)}} \overline{x}_j,$$

$$m_k \stackrel{\text{def}}{=} \sum_{i:\underline{x}_i \ge x_{(k+1)}} (\underline{x}_i)^2 + \sum_{j:\overline{x}_j \le x_{(k)}} (\overline{x}_j)^2,$$

and $n_k =$ the total number of such i's and j's. Then, we solve the quadratic equation $A_k - B_k \cdot \lambda + C_k \cdot \lambda^2 = 0$, where

$$A_k \stackrel{\text{def}}{=} e_k^2 \cdot (1+\alpha^2) - \alpha^2 \cdot m_k \cdot n; \quad \alpha \stackrel{\text{def}}{=} 1/k_0,$$

$$B_k \stackrel{\text{def}}{=} 2 \cdot e_k \cdot \Big((1+\alpha^2) \cdot n_k - \alpha^2 \cdot n\Big);$$

$$C_k \stackrel{\text{def}}{=} n_k \cdot \Big((1+\alpha^2) \cdot n_k - \alpha^2 \cdot n\Big).$$

For computing $\underline{U}$, we select only those solutions for which $\lambda \cdot n_k \le e_k$ and $\lambda \in [x_{(k)}, x_{(k+1)}]$; for computing $\overline{L}$, we select only those solutions for which $\lambda \cdot n_k \ge e_k$ and $\lambda \in [x_{(k)}, x_{(k+1)}]$. For each selected solution, we compute the values of

$$E_k = \frac{e_k}{n} + \frac{n - n_k}{n} \cdot \lambda, \quad M_k = \frac{m_k}{n} + \frac{n - n_k}{n} \cdot \lambda^2,$$

$$U_k = E_k + k_0 \cdot \sqrt{M_k - (E_k)^2} \text{ or } L_k = E_k - k_0 \cdot \sqrt{M_k - (E_k)^2}.$$

Finally, if we are computing $\underline{U}$, we return the smallest of the values U_k; if we are computing $\overline{L}$, we return the smallest of the values L_k.

In these algorithms, sorting requires $O(n \cdot \log(n))$ steps (see, e.g., [6]). The initial computation of all the quantities requires linear time, and for each zone, we need a constant time to update all the quantities, compute λ, and then compute the corresponding value U. Thus, the algorithm requires $O(n \cdot \log(n))$ time.

Computing $\overline{U}$ and $\underline{L}$: general case. It is known that in general, computing $\overline{U}$ and $\underline{L}$ is NP-hard [23, 24, 26].

If $1 + (1/k_0)^2 \le n$ (which is true, e.g., if $k_0 > 1$ and $n \ge 2$), then the corresponding maximum and minimum are always attained at the endpoints of the intervals $[\underline{x}_i, \overline{x}_i]$; so, to compute $\overline{U}$ and $\underline{L}$, it is sufficient to consider all 2^n combinations of such endpoints.

Computing $\overline{U}$ and $\underline{L}$: cases of narrow intervals and slightly wider intervals. For computing $\overline{U}$ and $\underline{L}$, a feasible algorithm is possible not only when the original intervals $[\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$ do not intersect, but also in a more general case when $1/n + 1/k_0^2 < 1$ and the "narrowed" intervals $\left[\widetilde{x}_i - \dfrac{1+\alpha^2}{n} \cdot \Delta_i, \widetilde{x}_i + \dfrac{1+\alpha^2}{n} \cdot \Delta_i\right]$ do not intersect – where $\alpha = 1/k_0$ and $\Delta_i \stackrel{\text{def}}{=} (\overline{x}_i - \underline{x}_i)/2$ is the interval's half-width.

In fact, a feasible algorithm is even possible in the case when for some integer $K < n$, no sub-collection of greater than K narrowed intervals of $\mathbf{x}_i$ has a common intersection [24].

The algorithms presented in [24] require quadratic time, but we can use the arguments like in the above description of $\overline{V}$ (see also [50]) and perform both algorithms in time $O(n \cdot \log(n))$.

Computing $\overline{U}$ and $\underline{L}$: subset property case. For the subset property case, similarly to variance, we can prove that the maximum of U and the minimum of L are attained at one of the vectors $(\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$; actually, practically the same proof works, because increasing V without changing E increases $U = E + k_0 \cdot \sqrt{V}$ as well.

Thus, in this case, first, we sort the intervals $[\underline{x}_i, \overline{x}_i]$ in lexicographic order; then, for $k = 0, 1, \ldots, n$, we compute the values $V = M - E^2$ and L and U for the corresponding vectors $x^{(k)} = (\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$. When we go from a vector $x^{(k)}$ to the vector $x^{(k+1)}$, only one term changes in the vector x, so only one terms changes in each of the sums E and M.

Sorting takes $O(n \cdot \log(n))$ time. Computing the initial values of E and M requires linear time $O(n)$. For each k, computing the new values of E and M requires a constant number of steps, so overall, computing all n values of E, M (and hence V) requires linear time.

Thus, the overall time of this algorithm is $O(n \cdot \log(n))$.

Computing $\overline{U}$ and $\underline{L}$: case of several MIs. In this case, similar to computing $\overline{V}$, we can perform all the computations in time $O(n^m)$.

Case when only some intervals $d < n$ are non-degenerate. In this case, we only need to sort the endpoints of non-degenerate intervals, and, correspondingly, only consider $2d$ zones. Thus, here, the computational complexity is the same as for the case of computing $\overline{V}$ (see table below).

4.8. Degree of Outlier-Ness

What is the degree of outlier-ness. For every x, we can also determine the "degree of outlier-ness" r as the smallest k_0 for which x is no longer inside the interval $[E - k_0 \cdot \sigma, E + k_0 \cdot \sigma]$, i.e., as $|x - E|/\sigma$.

This ratio is another example of inferential statistic actively used in statistics.

Simplification of the problem. First, it turns out that the value of r does not change if, instead of the original variables x_i with values from intervals $\mathbf{x}_i$, we consider new variables $x'_i \stackrel{\text{def}}{=} x_i - x$ and a new value $x' = 0$. Indeed, in this case, $E' = E - x$ hence $E' - x' = E - x$, and the standard deviation σ does not change if we simply shift all the values x_i. Thus, without losing generality, we can assume that $x = 0$, and we are therefore interested in the ratio $|E|/\sigma$.

Second, the lower bound of the ratio r is attained when the reverse ratio $1/r = \sigma/|E|$ is the largest, and vice versa. Thus, to find the interval of possible values for $|E|/\sigma$, it is necessary and sufficient to find the interval of possible values of $\sigma/|E|$. Computing this interval is, in its turn, equivalent to computing the interval for the square V/E^2 of the reverse ratio $1/r$.

Finally, since $V = M - E^2$, where $M \stackrel{\text{def}}{=} \dfrac{x_1^2 + \ldots + x_n^2}{n}$ is the second moment, we have $V/E^2 = M/E^2 - 1$, so computing the sharp bounds for V/E^2 is equivalent to computing the sharp bounds for the ratio $R \stackrel{\text{def}}{=} M/E^2$.

Computing $\underline{R}$. For every i, the location of the minimum on the interval $[\underline{x}_i, \overline{x}_i]$ depends on the values of the derivative

$$\frac{\partial R}{\partial x_i} = \frac{2}{n \cdot E^2} \cdot \left(x_i - \frac{M}{E} \right).$$

Thus, once we know where $\lambda \stackrel{\text{def}}{=} M/E$ is located in comparison with the endpoints, we can uniquely determine all the values x_i – and the value λ can be determined from the condition that the ratio M/E is exactly equal to λ.

Thus, we arrive at the algorithm presented in [24]. Similar to $\underline{V}$, sorting requires time $O(n \cdot \log(n))$; computing the initial values of the corresponding sums requires $O(n)$ steps; finally, updating each of these values requires a constant number of steps, so the overall time is linear. Thus, we can compute $\underline{R}$ in $O(n \cdot \log(n))$ steps.

Computing $\overline{R}$: general case. In principle, we can have $\overline{R} = +\infty$ – e.g., if $0 \in [\underline{E}, \overline{E}]$. If $0 \notin [\underline{E}, \overline{E}]$ – e.g., if $\underline{E} > 0$ – then we can guarantee that $\overline{R} < +\infty$. In this case, we can bound $\overline{R}$ by the ratio $\overline{M}/\underline{E}^2$.

When $\overline{R} < n$, the maximum $\overline{R}$ is always attained at the endpoints [24], so we can compute $\overline{R}$ by testing all 2^n combinations of $\underline{x}_i$ and $\overline{x}_i$.

Computing $\overline{R}$: cases of narrow intervals and slightly wider intervals. For computing $\overline{U}$ and $\underline{R}$, a feasible algorithm is possible not only when the original intervals $[\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$ do not intersect, but also in a more general case when $\overline{R} < n$ and the "narrowed" intervals $[x_i^-, x_i^+]$ do not intersect, where $x_i^- \stackrel{\text{def}}{=} \dfrac{\widetilde{x}_i}{1 + \dfrac{\Delta_i}{\underline{E} \cdot n}}$ and $x_i^+ \stackrel{\text{def}}{=} \dfrac{\widetilde{x}_i}{1 - \dfrac{\Delta_i}{\underline{E} \cdot n}}$. In fact, a feasible algorithm is even possible in the case when for some integer $K < n$, no sub-collection of greater than K narrowed intervals of $\mathbf{x}_i$ has a common intersection [24].

So, first, we sort all $2n$ values $\underline{x}_i$, $\overline{x}_i$ into a sequence $x_{(1)} \le x_{(2)} \le \ldots \le x_{(2n)}$, take $x_{(0)} = -\infty$ and $x_{(2n+1)} = +\infty$, and thus divide the real line into $2n+1$ zones $(x_{(0)}, x_{(1)}]$, $[x_{(1)}, x_{(2)}]$, $\ldots$, $[x_{(2n-1)}, x_{(2n)}]$, $[x_{(2n)}, x_{(2n+1)})$. For each of these zones $[x_{(k)}, x_{(k+1)}]$, $k = 0, 1, \ldots, 2n$, and for each variable x_i, we take:

- $x_i = \underline{x}_i$ if $x_i^+ \le x_{(k)}$;
- $x_i = \overline{x}_i$ if $x_i^- \ge x_{(k+1)}$;
- both values $x_i = \underline{x}_i$ and $x_i = \overline{x}_i$ otherwise.

For each of the resulting tuples $(x_1, \ldots, x_n)$, we compute E, M, and $\lambda = M/E$, and check if λ is within the zone; if it is, we compute $R_k = M/E^2$.

The largest of these computed values R_k is the desired upper endpoint $\overline{R}$.

This algorithm, if implemented along the lines of our algorithm for $\overline{V}$, can perform in time $O(n \cdot \log(n))$.

Computing $\overline{R}$: cases of the subset property and of several MIs. For the subset property case, similarly to computing $\overline{V}$, we can prove that the maximum is attained on one of the vectors $x = (\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$. Thus, in this case, we can compute $\overline{R}$ in $O(n \cdot \log(n))$ steps.

Similarly, for the case of m MIs, we can compute $\overline{R}$ in time $O(n^m)$.

Computing $\overline{R}$: other cases. Other cases are handled similarly to the case of computing bound for L and U, so we have similar complexity estimates.

Comment. We have described interval versions of several statistics of the type $C = f(M, E)$: namely, $V = M - E^2$, $L = E - k_0 \cdot \sqrt{V}$, $U = E + k_0 \cdot \sqrt{V}$, and $R = M/E^2$. In all these cases, f is an increasing function of M, hence $\dfrac{\partial f}{\partial x_i} = \dfrac{\partial f}{\partial M} \cdot \dfrac{2x_i}{n} + \dfrac{\partial f}{\partial E} \cdot \dfrac{1}{n}$ has the same sign as $x_i - \lambda$ for some constant $\lambda \stackrel{\text{def}}{=} -\dfrac{1}{2} \cdot \left(\dfrac{\partial f}{\partial E} \bigg/ \dfrac{\partial f}{\partial M} \right)$. Thus, for other statistics of the type $f(M, E)$, it may be possible to repeat arguments similar to the ones given for V, L, U, and R, and derive similar algorithms and similar computational complexity results.

4.9. Summary

The above results are summarized in the following table. In this table, the first row corresponds to a general case, other rows correspond to different classes of problems:

class number	class description
0	general case
1	narrow intervals: no intersection
2	slightly wider intervals $\le K$ intervals intersect
3	subset property – no interval is a "proper" subset of the other e.g., single measuring instrument (MI)
4	same accuracy measurements: all intervals have the same half-width
5	several (m) measuring instruments: intervals form m groups, with subset property in each group
6	privacy case: intervals same or non-intersecting
7	non-detects case: only non-degenerate intervals are $[0, DL_i]$

#	E	V, L, U, R, M_{2p}	C_{xy}	M_{2p+1}
0	$O(n)$	NP-hard	NP-hard	?
1	$O(n)$	$O(n \cdot \log(n))$	$O(n^2)$	$O(n^2)$
2	$O(n)$	$O(n \cdot \log(n))$	$O(n^2)$	$O(n^2)$
3	$O(n)$	$O(n \cdot \log(n))$	?	?
4	$O(n)$	$O(n \cdot \log(n))$	$O(n^3)$	?
5	$O(n)$	$O(n^m)$	?	?
6	$O(n)$	$O(n \cdot \log(n))$	$O(n^2)$	?
7	$O(n)$	$O(n \cdot \log(n))$	?	?

The case when only d out of n data points are intervals is summarized in the following table:

#	E	V, L, U, R, M_{2p}	C_{xy}	M_{2p+1}
0	$O(n)$	NP-hard	NP-hard	?
1	$O(n)$	$O(n + d \cdot \log(d))$	$O(n + d^2)$	$O(n + d^2)$
2	$O(n)$	$O(n + d \cdot \log(d))$	$O(n + d^2)$	$O(n + d^2)$
3	$O(n)$	$O(n + d \cdot \log(d))$	?	?
4	$O(n)$	$O(n + d \cdot \log(d))$	$O(n + d^3)$	?
5	$O(n)$	$O(n + d^m)$	?	?
6	$O(n)$	$O(n + d \cdot \log(d))$	$O(n + d^2)$	?
7	$O(n)$	$O(n + d \cdot \log(d))$	?	?

4.10. Other Statistical Characteristics

Weighted mean and weighted average. In the above text, we considered the case when we only know the upper bound Δ_i on the overall measurement error. In some real-life situations (see, e.g., [39]), we know the standard deviation σ_i of the random error component and the bound Δ_i on the absolute value of the systematic error component. If we had no systematic errors, then we would able to estimate the mean E by solving the corresponding Least Squares problem $\sum \sigma_i^{-2} \cdot (x_i - E)^2 \to \min\limits_E$, i.e., as $E_w = \sum\limits_{i=1}^n p_i \cdot x_i$, where $p_i \stackrel{\text{def}}{=} \dfrac{\sigma_i^{-2}}{\sum\limits_{j=1}^n \sigma_j^{-2}}$. In this case, the variance can be estimated as $V_w = \sum\limits_{i=1}^n p_i \cdot (x_i - E_w)^2 = \sum\limits_{i=1}^n p_i \cdot x_i^2 - E_w^2$. Due to the presence of systematic errors, the true values x_i may be anywhere within the intervals $[\underline{x}_i, \overline{x}_i] \stackrel{\text{def}}{=} [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$. Thus, we arrive at the problem of estimating the range of the above expressions for weighted mean and weighted variance on the interval data $[\underline{x}_i, \overline{x}_i]$.

The expression for the mean is monotonic, so, similar to the average, we substitute the values $\underline{x}_i$ to get $\underline{E}_w$ and the values $\overline{x}_i$ to get $\overline{E}_w$.

For the weighted variance, the derivative is equal to $2p_i \cdot (x_i - E_w)$, and the second derivative is always ≥ 0, so, similarly to the above proof for the non-weighted variance, we conclude that the minimum is always attained at a vector $(\overline{x}_1, \ldots, \overline{x}_k, E_w, \ldots, E_w, \underline{x}_{k+l}, \ldots, \overline{x}_n)$. So, by considering $2n+2$ zones, we can find $\underline{V}_w$ in time $O(n \cdot \log(n))$.

For $\overline{V}_w$, we can prove that the maximum is always attained at values $x_i = \underline{x}_i$ or $x_i = \overline{x}_i$, so we can always find it in time $O(2^n)$. If no more than K intervals intersect, then, similarly to the non-weighted variance, we can compute $\overline{V}_w$ in time $O(n \cdot \log(n))$.

Robust estimates for the mean. Arithmetic average is vulnerable to outliers: if one of the values is accidentally mis-read as 10^6 times larger than the others, the average is ruined. Several techniques have been proposed to make estimates robust; see, e.g., [16]. The best known estimate of this type is the median; there are also more general *L-estimates* of the type $\sum\limits_{i=1}^{n} w_i \cdot x_{(i)}$, where $w_1 \geq 0, \ldots, w_n \geq 0$ are given constants, and $x_{(i)}$ is the i-th value in the ordering of $x_1, \ldots, x_n$ in increasing order. Other techniques include *M-estimates*, i.e., estimates a for which $\sum\limits_{i=1}^{n} \psi(|x_i - a|) \to \max\limits_a$ for some non-decreasing function $\psi(x)$.

Each of these statistics C is a (non-strictly) increasing function of each of the variables x_i. Thus, similarly to the average, $\mathbf{C} = [C(\underline{x}_1, \ldots, \underline{x}_n), C(\overline{x}_1, \ldots, \overline{x}_n)]$.

Robust estimates for the generalized central moments. When we discussed central moments, we considered generalized central moments $M_\psi = \frac{1}{n} \cdot \sum\limits_{i=1}^{n} \psi(x_i - E)$ for an appropriate convex function $\psi(x)$. In that description, we assumed that E is the usual average.

It is also possible to consider the case when E is not the average, but the value for which $\sum\limits_{i=1}^{n} \psi(x_i - E) \to \min\limits_E$. In this case, the robust estimate for the generalized central moment takes the form

$$M_\psi^{\mathrm{rob}} = \min_E \left(\frac{1}{n} \cdot \sum_{i=1}^{n} \psi(x_i - E) \right).$$

Since the function $\psi(x)$ is convex, the expression $\sum\limits_{i=1}^{n} \psi(x_i - E)$ is also convex, so it only attains its maximum at the vertices of the convex box $\mathbf{x}_1 \times \ldots \times \mathbf{x}_b$, i.e., when for every i, either $x_i = \underline{x}_i$ or $x_i = \overline{x}_i$. For the subset property case, the same proof as for the average E enables us to conclude that the maximum of the new generalized central moment is also always attained at one of n vectors $(\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$, and thus, that this maximum can be computed in time $O(n^2)$. For m MIs, we need time $O(n^{m+1})$.

Correlation. For correlation, we only know that in general, the problem of computing the exact range is NP-hard [9].

4.11. From the 1-D Ranges of E and V to the 2-D Range of (E, V)

Formulation of the Problem In the above text, we have described how, given the interval data $\mathbf{x}_1, \ldots, \mathbf{x}_n$, we can compute the exact range $\mathbf{E}$ of the population mean E and the exact range $\mathbf{V}$ of the population variance V. The fact that the range is exact means the following:

- first, that for every $x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n$, the values E and V belong to the corresponding intervals $\mathbf{E}$ and $\mathbf{V}$;

- second, that for every value $E \in \mathbf{E}$, there exist values $x_i \in \mathbf{x}_i$ for which the population mean is equal to E, and that for every value $V \in \mathbf{V}$, there exist values $x_i \in \mathbf{x}_i$ for which the population variance is equal to V.

Based on the computed ranges $\mathbf{E}$ and $\mathbf{V}$, we can conclude that for every $x_i \in \mathbf{x}_i$, the pair (E, V) belongs to the box $\mathbf{E} \times \mathbf{V}$. However, not all pairs (E, V) from this box are possible. For example, the only way to get $E = \underline{E}$ is to use $x_1 = \underline{x}_1, \ldots, x_n = \underline{x}_n$; in this case, the population variance can take only one value $V(\underline{x}_1, \ldots, \underline{x}_n) = \dfrac{1}{n} \cdot \sum\limits_{i=1}^{n} (\underline{x}_i - \underline{E})^2$.

It is therefore desirable to describe not only the ranges $\mathbf{E}$ and $\mathbf{V}$ of E and V, but also the range of possible values of the pairs (E, V). In other words, for each $E \in \mathbf{E}$, we want to know the range $\mathbf{V}(E) = [\underline{V}(E), \overline{V}(E)]$ of possible values of the population variance V under the condition that the population mean is equal to E. Let us describe how we can compute the dependence of the range $\mathbf{V}(E)$ on the given value of E.

Computing $\underline{V}(E)$. Let us show that we can compute the dependence $\underline{V}(E)$ in time $O(n \cdot \log(n))$. The corresponding algorithm is similar to the one that we used to compute $\underline{V}$.

Indeed, since $V = M - E^2$, minimizing V under fixed E is equivalent to minimizing $M = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i^2$. Let $(x_1, \ldots, x_n)$ be a tuple that minimizes M.

Let us first show that if $x_i < \overline{x}_i$ and $x_j > \underline{x}_j$, then $x_i \geq x_j$. Indeed, in this case, if $\varepsilon > 0$ is sufficiently small, we can replace x_i with $x_i + \varepsilon$, and x_j with $x_j - \varepsilon$. After this replacement, the population average E does not change, and M is replaced with $M + \frac{2}{n} \cdot (x_i - x_j) \cdot \varepsilon + O(\varepsilon^2)$. Since M was at its minimum, the change in M cannot be negative, hence $x_i - x_j \geq 0$ and $x_i \geq x_j$.

From this property, we can conclude that if x_i and x_j are inside the corresponding intervals, i.e., $\underline{x}_i < x_i < \overline{x}_i$ and $\underline{x}_j < x_j < \overline{x}_j$, then $x_i = x_j$. Indeed, from the fact that $x_i < \overline{x}_i$ and $x_j > \underline{x}_j$, we conclude that $x_i \geq x_j$, and similarly, from the fact that if $x_j < \overline{x}_j$ and $x_i > \underline{x}_i$, we conclude that $x_j \geq x_i$; therefore, $x_i = x_j$. Thus, all internal values of x_i coincide. Let us denote the common internal value of x_i by λ.

Similarly, we conclude that in the minimizing tuples $x = (x_1, \ldots, x_n)$, every value $x_i = \underline{x}_i$ is not smaller than every internal value $x_i = \lambda$, and λ is not smaller than every value $x_j = \overline{x}_j$. Thus, if we know λ, we can uniquely determine all n values x_i: $x_i = \overline{x}_i$ when $\overline{x}_i \leq \lambda$, $x_i = \underline{x}_i$ when $\lambda \leq \underline{x}_i$, and $x_i = \lambda$ when $\underline{x}_i < \lambda < \overline{x}_i$. (If no x_i attains its internal value, then, as λ, we can take any value separating $\overline{x}_i$ from $\underline{x}_j$.) Hence, after we sort the endpoints $\underline{x}_i$ and $\overline{x}_i$ into an increasing sequence $x_{(1)} \leq x_{(2)} \leq \ldots$, then, once we know to which zone $[x_{(k)}, x_{(k+1)}]$ the value λ belongs, we get N_k values $x_i = \lambda$ and $n - N_k$ known values x_i. Similarly to the case of $\underline{V}$, for $\lambda \in [x_{(k)}, x_{(k+1)}]$, we thus get E as a linear function of λ and the corresponding value $\underline{M}(E)$ – and hence, $\underline{V}(E)$ – as an explicit quadratic function of λ.

We can use the expression for E to describe λ as a linear function of E; substituting this expression into the formula for $\underline{V}(E)$, we get the coefficients of the quadratic expression that describes the dependence of $\underline{V}(E)$ on E for all E from the corresponding interval $[E_k, E_{k+1}]$, where $E_k \stackrel{\text{def}}{=} \sum_{i=1}^{k} \overline{x}_i + \sum_{j=k+1}^{n} \underline{x}_j$.

Similar to the case of $\underline{V}$, sorting requires $O(n \cdot \log(n))$ steps; the initial computation of the necessary expressions E_k, M_k', and N_k requires $O(n)$ steps, and the transition from each k to the next requires a constant number of steps. Thus, overall, we can determine the piece-wise quadratic dependence of $\underline{V}(E)$ on E in $O(n \cdot \log(n))$ steps.

Computing $\overline{V}(E)$: general case. For $\overline{V}$, we can similarly conclude that in the maximizing tuple $x = (x_1, \ldots, x_n)$, all the internal values of x_i

coincide ($x_i = \lambda$), all the values $x_i = \underline{x}_i$ are smaller than or equal to λ, and all the values $x_i = \overline{x}_i$ are larger than or equal to λ.

In addition, we can prove that in the maximizing sequence, there can be at most one internal value. Indeed, if we have two internal values $x_i = x_j = \lambda$, then, for sufficiently small $\varepsilon > 0$, we can replace $x_i = \lambda$ with $x_i = \lambda + \varepsilon$, and $x_j = \lambda$ with $x_j = \lambda - \varepsilon$. After this replacement, the population average E does not change, and M is replaced by a larger value $M + \frac{2}{n} \cdot \varepsilon^2$. Since M was at its maximum, this cannot happen, so in the maximizing tuple, there is indeed at most one internal value x_i; all the other values x_j are equal to either $\underline{x}_j$ or to $\overline{x}_j$.

In the general case, to find $\overline{V}(E)$, we can, therefore, test all the values i from 1 to n; for each i, we try all 2^{n-1} combinations of $\underline{x}_j$ and $\overline{x}_j$. For each such combination, E is a linear function of λ and V is a quadratic function of λ. Similarly to the case of $\underline{V}(E)$, we can find the linear dependence of λ on E and hence, the quadratic dependence of V on E. The actual dependence $\overline{V}(E)$ is thus the maximum of the corresponding $2^{n-1} \cdot n = O(2^n)$ quadratic dependences. So, in general, we can find the dependence of $\overline{V}(E)$ on E in $O(2^n)$ steps.

Computing $\overline{V}(E)$: cases of narrow and almost narrow intervals. From the above properties of the maximizing sequence x_i, it follows that $x_i = \underline{x}_i$ when $\overline{x}_i \le \lambda$, $x_i = \overline{x}_i$ when $\lambda \le \underline{x}_i$, and for the remaining case when $\underline{x}_i < \lambda < \overline{x}_i$, we can have 3 possibilities: $x_i = \underline{x}_i$, $x_i = \overline{x}_i$, and $x_i = \lambda$ (where the equality $x_i = \lambda$ is possible for at most one value i).

So, once we fix the zone that contains λ, we can uniquely determine the values x_i for all the intervals $\mathbf{x}_i$ except for the intervals that contain this zone. In the case of almost narrow intervals, for each zone, there are at most K such intervals, so we have $\le K \cdot 2^K = O(1)$ possible assignments. Thus, we can describe $\overline{V}(E)$ as the maximum of $O(n)$ dependences corresponding to all these assignments.

Similarly to the case of $\overline{V}$, computing all the coefficients of all these $O(n)$ dependences requires $O(n \cdot \log(n))$ time.

Computing $\overline{V}(E)$: subset property case. In the subset property case, we can sort the intervals $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$ in lexicographic order so that both their lower endpoints $\underline{x}_i$ and upper endpoints $\overline{x}_i$ become sorted: $\underline{x}_1 \le \underline{x}_2 \le \ldots$ and $\overline{x}_1 \le \overline{x}_2 \le \ldots$

Let us first show that, for every E, we can choose a monotonic maximizing sequence x_i. Indeed, if we have a maximizing sequence $(x_1, \ldots, x_n)$ in which $x_i > x_j$ for some $i < j$, then we have $x_j < x_i \le \overline{x}_i$ and $x_j \ge \underline{x}_j \ge \underline{x}_i$ hence $x_j \in \mathbf{x}_i$, and similarly, $x_i \in \mathbf{x}_j$. Thus, we can swap the values x_i and x_j (i.e., take $x_i^{\text{new}} = x_j$ and $x_j^{\text{new}} = x_i$)

and get a new sequence with exactly the same values of E, M, and V. After repeating such a swap as many times as necessary, we will get a maximizing sequence that is monotonic in the sense that $x_1 \le x_2 \le \ldots \le x_n$.

In view of the already proven properties of maximizing sequences, and in view of the fact that the sequences $\underline{x}_i$ and $\overline{x}_i$ are also monotonic, we conclude that the maximizing sequence has the following form: $(\underline{x}_1, \ldots, \underline{x}_{k_1}, \lambda, \overline{x}_{k+1}, \ldots, \overline{x}_n)$.

For each zone $\lambda \in [x_{(k)}, x_{(k+1)}]$, we thus get E as a linear function of λ and the corresponding value $\underline{V}$ as an explicit quadratic function of λ. So, we can get the coefficients of the quadratic expression that describes the dependence of $\overline{V}(E)$ on E for all E from the corresponding interval $[E'_k, E'_{k+1}]$, where $E'_k \stackrel{\text{def}}{=} \sum\limits_{i=1}^{k} \underline{x}_i + \sum\limits_{j=k+1}^{n} \overline{x}_j$.

Similar to the case of $\overline{V}$, we can thus determine the piece-wise quadratic dependence of $\overline{V}(E)$ on E in $O(n \cdot \log(n))$ steps.

5. Additional Issues

On-line data processing. In the above text, we implicitly assumed that before we start computing the statistics, we have all the measurement results. In real life, we often continue measurements after we started the computations. Traditional estimates for mean and variance can be easily modified with the arrival of the new measurement result x_{n+1}: $E' = (n \cdot E + x_{n+1})/(n+1)$ and $V' = M' - (E')^2$, where $M' = (n \cdot M + x_{n+1}^2)/(n+1)$ and $M = V + E^2$. For the interval mean, we can have a similar adjustment. However, for other statistics, the above algorithms for processing interval data require that we start computation from scratch. Is it possible to modify these algorithms to adjust them to on-line data processing? The only statistic for which such an adjustment is known is the variance, for which an algorithm proposed in [25, 49] requires only $O(n)$ steps to incorporate a new interval data point.

In this algorithm, we store the sorting corresponding to the zones and we store auxiliary results corresponding to each zone (finitely many results for each zone). So, if only d out of n intervals are non-degenerate, we only need $O(d)$ steps to incorporate a new data point.

Fuzzy data. Often, in addition to (or instead of) the guaranteed bounds, an expert can provide bounds that contain x_i with a certain degree of confidence. Often, we know several such bounding intervals corresponding to different degrees of confidence. Such a nested family of intervals is also called a *fuzzy set*, because it turns out to be equivalent to a more traditional definition of fuzzy set [32, 33] (if a traditional fuzzy set is given, then different intervals from the nested family can be viewed as α-cuts corresponding to different levels of uncertainty α).

To provide statistical analysis of fuzzy-valued data, we can therefore, for each level α, apply the above interval-valued techniques to the corresponding α-cuts [29, 34].

Can we detect when the algorithms designed for the several MI case are applicable? For the several MI case, we know how many MIs there are and which measurements are made with which MI. In other words, the measurement results are labeled by the corresponding MI, and this labeling is used in the algorithms.

Sometimes, the intervals come not from measurements but, e.g., from experts. In some such cases, we can still divide the resulting intervals into a small number of sub-families each of which has a subset property. In such cases, instead of the time-consuming general-case algorithms, we can use more efficient algorithms designed for the case of several MIs.

To be able to apply these efficient algorithms, we must be able, given a family of intervals and a small integer m, to check whether this family can be subdivided into m families that have the subset property.

For $m = 2$, we can check whether this subdivision is possible as follows. Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be the intervals that we want to subdivide. We will analyze these intervals one by one and, in the process of analyzing, assign each interval to one of the two families.

Without losing generality, we assign the first interval $\mathbf{x}_1$ to Family 1. When intervals $\mathbf{x}_1, \ldots, \mathbf{x}_k$ are already assigned to different families, we check whether the next interval $\mathbf{x}_{k+1}$ is in subset relation with the already assigned intervals $\mathbf{x}_1, \ldots, \mathbf{x}_k$. If $\mathbf{x}_{k+1}$ is in subset relation with an interval from the first family and with an interval from the second family, then the subdivision into 2 families is impossible, so we stop. Otherwise, if $\mathbf{x}_{k+1}$ is in subset relation with one of the intervals assigned to the first family, we assign it to the second family, else we assign it to the first family. If the algorithm did not stop, this means that we have the desired subdivision, so we can apply the algorithms developed for the case of several MIs.

For $m > 2$, checking may not be easy. Indeed, we can construct a graph in which vertices are intervals, and vertices are connected if they are in a subset relation. Our objective is to assign a class to each vertex so that connected vertices cannot be of the same class. This is exactly the coloring problem that is known to be NP-hard [12].

Parallelization. In the general case, the problem of computing the range $\mathbf{C}$ of a statistic C on interval data $\mathbf{x}_i$ requires too much computation time. One way to speed up computations is to use parallel computations.

If we have a potentially unlimited number of parallel processors, then, for the mean, the addition can be done in time $O(\log(n))$ [17]. In $O(n \cdot \log(n))$ algorithms for computing $\underline{V}$ and $\overline{V}$, we can perform sorting in time $O(\log(n))$, then compute V_k for each zone in parallel, and find the largest of the n resulting values V_k in parallel (in time $O(\log(n))$). The sum that constitutes the variance can also be computed in parallel in time $O(\log(n))$, so overall, we need $O(\log(n))$ time.

Similarly, we can transform polynomial algorithms for computing the bounds for covariance, outlier statistics (L, U, and R), and moments into $O(\log(n))$ parallel algorithms.

In the general case, to find $\overline{V}$ and other difficult-to-compute bounds, we must compute the largest of the $N \stackrel{\text{def}}{=} 2^n$ values corresponding to 2^n possible combinations of $\underline{x}_i$ and $\overline{x}_i$. This maximum can be computed in time $O(\log(N)) = O(n)$. This does not mean, of course, that we can always physically compute $\overline{V}$ in linear time: communication time grows exponentially with n; see, e.g., [31].

It is desirable to also analyze the case when we have a limited number of processors $p \ll n$.

Quantum algorithms. Another way to speed up computations is to use quantum computing. In [29, 21], we describe how quantum algorithms can speed up the computation of $\mathbf{C}$.

Acknowledgements

This work was supported in part by NASA under cooperative agreement NCC5-209, by the Future Aerospace Science and Technology Program (FAST) Center for Structural Integrity of Aerospace Systems, effort sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant F49620-00-1-0365, by NSF grants EAR-0112968, EAR-0225670, and EIA-0321328, by the Army Research Laboratories grant DATM-05-02-C-0046, and by a research grant from Sandia National Laboratories as part of the Department of Energy Accelerated Strategic Computing Initiative (ASCI).

The authors are greatly thankful to all the participants of the International Workshop on Reliable Engineering Computing REC'04 (Savannah, Georgia, September 15–17, 2004) for valuable comments, and to the anonymous referees for their very useful suggestions.

References

1. J. Beck, V. Kreinovich, and B. Wu, Interval-Valued and Fuzzy-Valued Random Variables: From Computing Sample Variances to Computing Sample Covariances, In: M. Lopez, M. A. Gil, P. Grzegorzewski, O. Hrynewicz, and J. Lawry, editor, *Soft Methodology and Random Information Systems*, Springer-Verlag, Berlin-Heidelberg, 2004, pp. 85–92.
2. D. Berleant, Automatically verified arithmetic with both intervals and probability density functions, *Interval Computations*, 1993, (2):48–70.
3. D. Berleant, Automatically verified arithmetic on probability distributions and intervals, In: R. B. Kearfott and V. Kreinovich, editors, *Applications of Interval Computations*, Kluwer, Dordrecht, 1996.
4. D. Berleant and C. Goodman-Strauss, Bounding the results of arithmetic operations on random variables of unknown dependency using intervals, *Reliable Computing*, 1998, 4(2):147–165.
5. D. Berleant, L. Xie, and J. Zhang, Statool: A Tool for Distribution Envelope Determination (DEnv), an Interval-Based Algorithm for Arithmetic on Random Variables, *Reliable Computing*, 2003, 9(2):91–108.
6. Th. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 2001.
7. S. Ferson, *RAMAS Risk Calc 4.0: Risk Assessment with Uncertain Numbers*, CRC Press, Boca Raton, Florida, 2002.
8. S. Ferson, L. Ginzburg, V. Kreinovich, and M. Aviles, Exact Bounds on Sample Variance of Interval Data, *Extended Abstracts of the 2002 SIAM Workshop on Validated Computing*, Toronto, Canada, 2002, pp. 67–69
9. S. Ferson, L. Ginzburg, V. Kreinovich, L. Longpré, and M. Aviles, Exact Bounds on Finite Populations of Interval Data, *Reliable Computing*, 2005, 11(3):207–233.
10. S. Ferson, L. Ginzburg, V. Kreinovich, L. Longpré, and M. Aviles, Computing Variance for Interval Data is NP-Hard, *ACM SIGACT News*, 2002, 33(2):108–118.
11. S. Ferson, D. Myers, and D. Berleant, *Distribution-free risk analysis: I. Range, mean, and variance*, Applied Biomathematics, Technical Report, 2001.
12. M. E. Garey and D. S. Johnson, *Computers and intractability: a guide to the theory of NP-completeness*, Freeman, San Francisco, 1979.
13. D. J. H. Garling, A book review, *American Mathematical Monthly*, 2005, 112(6):575–579.
14. L. Granvilliers, V. Kreinovich, and N. Müller, Novel Approaches to Numerical Software with Result Verification, In: R. Alt, A. Frommer, R. B. Kearfott, and W. Luther (eds.), *Numerical Software with Result Verification*, (International Dagstuhl Seminar, Dagstuhl Castle, Germany, January 19–24, 2003), Springer Lectures Notes in Computer Science, 2004, Vol. 2991, pp. 274–305.
15. G. H. Hardy, J. E. Littlewood, and G. Pólya, *Inequalities*, Cambridge University Press, 1988.
16. P. J. Huber, *Robust statistics*, Wiley, New York, 2004.
17. J. Jájá, *An Introduction to Parallel Algorithms*, Addison-Wesley, Reading, MA, 1992.
18. V. Kreinovich, Probabilities, Intervals, What Next? Optimization Problems Related to Extension of Interval Computations to Situations with Partial Information about Probabilities, *Journal of Global Optimization*, 2004, 29(3):265–280.

19. V. Kreinovich, A. Lakeyev, J. Rohn, and P. Kahl, *Computational complexity and feasibility of data processing and interval computations*, Kluwer, Dordrecht, 1997.
20. V. Kreinovich and L. Longpré, Computational complexity and feasibility of data processing and interval computations, with extension to cases when we have partial information about probabilities, In: V. Brattka, M. Schroeder, K. Weihrauch, and N. Zhong, editors, *Proc. Conf. on Computability and Complexity in Analysis CCA'2003*, Cincinnati, Ohio, USA, August 28–30, 2003, pp. 19–54.
21. V. Kreinovich and L. Longpré, Fast Quantum Algorithms for Handling Probabilistic and Interval Uncertainty, *Mathematical Logic Quarterly*, 2004, 50(4/5):507–518.
22. V. Kreinovich, L. Longpré, S. Ferson, and L. Ginzburg, *Computing Higher Central Moments for Interval Data*, University of Texas at El Paso, Department of Computer Science, Technical Report UTEP-CS-03-14b, 2004, http://www.cs.utep.edu/vladik/2003/tr03-14b.pdf
23. V. Kreinovich, L. Longpré, P. Patangay, S. Ferson, and L. Ginzburg, Outlier Detection Under Interval Uncertainty: Algorithmic Solvability and Computational Complexity, In: I. Lirkov, S. Margenov, J. Wasniewski, and P. Yalamov, editors, *Large-Scale Scientific Computing*, Proceedings of the 4-th International Conference LSSC'2003, Sozopol, Bulgaria, June 4–8, 2003, Springer Lecture Notes in Computer Science, 2004, Vol. 2907, pp. 238–245
24. V. Kreinovich, L. Longpré, P. Patangay, S. Ferson, and L. Ginzburg, Outlier Detection Under Interval Uncertainty: Algorithmic Solvability and Computational Complexity, Reliable Computing, 2005, 11(1):59–76.
25. V. Kreinovich, H. T. Nguyen, and B. Wu, On-Line Algorithms for Computing Mean and Variance of Interval Data, and Their Use in Intelligent Systems, *Information Sciences* (in press).
26. V. Kreinovich, P. Patangay, L. Longpré, S. A. Starks, C. Campos, S. Ferson, and L. Ginzburg, Outlier Detection Under Interval and Fuzzy Uncertainty: Algorithmic Solvability and Computational Complexity, *Proceedings of the 22nd International Conference of the North American Fuzzy Information Processing Society NAFIPS'2003*, Chicago, Illinois, July 24–26, 2003, pp. 401–406.
27. V. P. Kuznetsov, *Interval Statistical Models*, Radio i Svyaz, Moscow, 1991 (in Russian).
28. W. A. Lodwick and K. D. Jamison, Estimating and Validating the Cumulative Distribution of a Function of Random Variables: Toward the Development of Distribution Arithmetic, *Reliable Computing*, 2003, 9(2):127–141.
29. M. Martinez, L. Longpré, V. Kreinovich, S. A. Starks, and H. T. Nguyen, Fast Quantum Algorithms for Handling Probabilistic, Interval, and Fuzzy Uncertainty, *Proceedings of the 22nd International Conference of the North American Fuzzy Information Processing Society NAFIPS'2003*, Chicago, Illinois, July 24–26, 2003, pp. 395–400.
30. R. E. Moore and W. A. Lodwick, Interval Analysis and Fuzzy Set Theory, *Fuzzy Sets and Systems*, 2003, 135(1):5–9.
31. D. Morgenstein and V. Kreinovich, Which algorithms are feasible and which are not depends on the geometry of space-time, *Geombinatorics*, 1995, 4(3):80–97.
32. H. T. Nguyen and V. Kreinovich, Nested Intervals and Sets: Concepts, Relations to Fuzzy Sets, and Applications, In: R. B. Kearfott and V. Kreinovich, editors, *Applications of Interval Computations*, Kluwer, Dordrecht, 1996, pp. 245–290

33. H. T. Nguyen and E. A. Walker, *First Course in Fuzzy Logic*, CRC Press, Boca Raton, Florida, 1999.
34. H. T. Nguyen, T. Wang, and V. Kreinovich, Towards Foundations of Processing Imprecise Data: From Traditional Statistical Techniques of Processing Crisp Data to Statistical Processing of Fuzzy Data, In: Y. Liu, G. Chen, M. Ying, and K.-Y. Cai, editors, *Proceedings of the International Conference on Fuzzy Information Processing: Theories and Applications FIP'2003*, Beijing, China, March 1–4, 2003, Vol. II, pp. 895–900.
35. H. T. Nguyen, B. Wu, and V. Kreinovich, Shadows of Fuzzy Sets – A Natural Approach Towards Describing 2-D and Multi-D Fuzzy Uncertainty in Linguistic Terms, *Proc. 9th IEEE Int'l Conference on Fuzzy Systems FUZZ-IEEE'2000*, San Antonio, Texas, May 7–10, 2000, Vol. 1, pp. 340–345.
36. P. Nivlet, F. Fournier, and J. Royer, A new methodology to account for uncertainties in 4-D seismic interpretation, *Proc. 71st Annual Int'l Meeting of Soc. of Exploratory Geophysics SEG'2001*, San Antonio, TX, September 9–14, 2001, 1644–1647.
37. P. Nivlet, F. Fournier, and J. Royer, Propagating interval uncertainties in supervised pattern recognition for reservoir characterization, *Proc. 2001 Society of Petroleum Engineers Annual Conf. SPE'2001*, New Orleans, LA, September 30–October 3, 2001, paper SPE-71327.
38. R. Osegueda, V. Kreinovich, L. Potluri, R. Aló, Non-Destructive Testing of Aerospace Structures: Granularity and Data Mining Approach, *Proc. FUZZ-IEEE'2002*, Honolulu, HI, May 12–17, 2002, Vol. 1, pp. 685–689
39. S. Rabinovich, *Measurement Errors: Theory and Practice*, American Institute of Physics, New York, 1993.
40. H. Regan, S. Ferson, and D. Berleant, Equivalence of five methods for bounding uncertainty, *Journal of Approximate Reasoning*, 2004, 36(1):1–30.
41. N. C. Rowe, Absolute bounds on the mean and standard deviation of transformed data for constant-sign-derivative transformations, *SIAM Journal of Scientific Statistical Computing*, 1988, 9:1098–1113.
42. I. Shmulevich and W. Zhang, Binary analysis and optimization-based normalization of gene expression data, *Bioinformatics*, 2002, 18(4):555–565.
43. S. A. Starks, V. Kreinovich, L. Longpre, M. Ceberio, G. Xiang, R. Araiza, J. Beck, R. Kandathi, A. Nayak, and R. Torres, "Towards combining probabilistic and interval uncertainty in engineering calculations", *Proceedings of the Workshop on Reliable Engineering Computing*, Savannah, Georgia, September 15–17, 2004, pp. 193–213.
44. H. M. Wadsworth, Jr., editor, *Handbook of statistical methods for engineers and scientists*, McGraw-Hill Publishing Co., N.Y., 1990.
45. P. Walley, *Statistical Reasoning with Imprecise Probabilities*, Chapman & Hall, N.Y., 1991.
46. G. W. Walster, Philosophy and practicalities of interval arithmetic, In: *Reliability in Computing*, Academic Press, N.Y., 1988, pp. 309–323.
47. G. W. Walster and V. Kreinovich, For unknown-but-bounded errors, interval estimates are often better than averaging, *ACM SIGNUM Newsletter*, 1996, 31(2)6–19.
48. R. Williamson and T. Downs, Probabilistic arithmetic I: numerical methods for calculating convolutions and dependency bounds, *International Journal of Approximate Reasoning*, 1990, 4:89–158.
49. B. Wu, H. T. Nguyen, and V. Kreinovich, Real-Time Algorithms for Statistical Analysis of Interval Data, *Proceedings of the International Conference on*

Information Technology InTech'03, Chiang Mai, Thailand, December 17–19, 2003, pp. 483–490.

50. G. Xiang, Fast algorithm for computing the upper endpoint of sample variance for interval data: case of sufficiently accurate measurements, *Reliable Computing* (in press).
51. G. Xiang, S. A. Starks, V. Kreinovich, and L. Longpré, New Algorithms for Statistical Analysis of Interval Data, *Proceedings of the Workshop on State-of-the-Art in Scientific Computing PARA'04*, Lyngby, Denmark, June 20–23, 2004, Vol. 1, pp. 123–129.
52. W. Zhang, I. Shmulevich, and J. Astola, *Microarray Quality Control*, Wiley, Hoboken, New Jersey, 2004.