New Algorithms for Statistical Analysis of Interval Data

Gang Xiang, Scott A. Starks, Vladik Kreinovich, and Luc Longpré

NASA Pan-American Center for Earth and
Environmental Studies (PACES)
University of Texas, El Paso, TX 79968, USA
vladik@cs.utep.edu

Abstract. It is known that in general, statistical analysis of interval data is an NP-hard problem: even computing the variance of interval data is, in general, NP-hard. Until now, only one case was known for which a feasible algorithm can compute the variance of interval data: the case when all the measurements are accurate enough – so that even after the measurement, we can distinguish between different measured values $\widetilde{x}_i$. In this paper, we describe several new cases in which feasible algorithms are possible – e.g., the case when all the measurements are done by using the same (not necessarily very accurate) measurement instrument – or at least a limited number of different measuring instruments.

1 Introduction

Once we have several results $\widetilde{x}_1, \ldots, \widetilde{x}_n$ of measuring some physical quantity – e.g., the amount of pollution in a lake – traditional statistical data processing starts with computing the sample average $E = E(\widetilde{x}_1, \ldots, \widetilde{x}_n)$, the sample median $M = M(\widetilde{x}_1, \ldots, \widetilde{x}_n)$, and the sample variance $V = V(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ of these results. For example, $E(\widetilde{x}_1, \ldots, x_n) = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} \widetilde{x}_i$.

The values $\widetilde{x}_i$ come from measurements, and measurements are never 100% accurate. In many real-life situations, the only information about the corresponding measurement errors is the upper bound Δ_i on the absolute value of the measurement error. As a result, the only information we have about the actual value x_i of each measured quantity is that x_i belongs to the interval $\mathbf{x}_i \stackrel{\text{def}}{=} [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$.

For interval data, instead of the exact values of E, M, and V, it is desirable to get the intervals $\mathbf{E}$, $\mathbf{M}$, and $\mathbf{V}$ of possible values, intervals formed by all possible values of E (correspondingly, M or V) when each x_i takes values from the interval $\mathbf{x}_i$.

Computing $\mathbf{E} = [\underline{E}, \overline{E}]$ and $\mathbf{M} = [\underline{M}, \overline{M}]$ is straightforward: indeed, both the sample average E and the sample median M are (non-strictly) increasing functions of the variables $x_1, \ldots, x_n$. So, the smallest possible value $\underline{E}$ (correspondingly, $\underline{M}$) is attained when we take the smallest possible values $\underline{x}_1, \ldots, \underline{x}_n$

from the corresponding intervals; similarly, the largest possible value $\overline{E}$ (correspondingly, $\overline{M}$) is attained when we take the largest possible values $\overline{x}_1, \ldots, \overline{x}_n$ from the corresponding intervals. Thus, $\underline{E} = E(\underline{x}_1, \ldots, \underline{x}_n)$, $\overline{E} = E(\overline{x}_1, \ldots, \overline{x}_n)$, $\underline{M} = M(\underline{x}_1, \ldots, \underline{x}_n)$, and $\overline{M} = M(\overline{x}_1, \ldots, \overline{x}_n)$.

On the other hand, computing the exact range $\mathbf{V} = [\underline{V}, \overline{V}]$ of V turns out to be an NP-hard problem; specifically, computing the upper endpoint $\overline{V}$ is NP-hard (see, e.g., [2]).

It is worth mentioning that computing the lower endpoint $\underline{V}$ is feasible; in [3], we show that it can done in time $O(n \cdot \log(n))$.

In the same paper [2] in which we prove that computing $\overline{V}$ is, in general, NP-hard, we also show that in the case when the measuring instruments are accurate enough – so that even after the measurements, we can distinguish between different measured values $\widetilde{x}_i$ (e.g, if the corresponding intervals $\mathbf{x}_i$ do not intersect) – we can compute $\overline{V}$ (hence, $\mathbf{V}$) in feasible time (actually, quadratic time).

In some practical examples, the measurement instruments are indeed very accurate, but in many other practical cases, their accuracy may be much lower – so the algorithm from [2] is not applicable.

In this paper, we describe new practically useful cases when we can compute $\mathbf{V}$ by a feasible (polynomial-time) algorithm.

The first case is when all the measurements are made by the same measuring instrument or by similar measurement instruments. In this case, none of two input intervals $\mathbf{x}_i$ is a proper subset of one another, and as a result, we can find the exact range $\mathbf{V}$ in time $O(n \cdot \log(n))$.

The second case is when instead of a single type of measuring instruments, we use a limited number ($m > 1$) of different types of measuring instruments. It turns out that in this case, we can compute $\mathbf{V}$ in polynomial time $O(n^{m+1})$.

The third case is related to privacy in statistical databases; see details below.

2 First Case: Measurements by Same Measuring Instrument

In the proof that computing variance is NP-hard (given in [2]), we used interval data in which some intervals are proper subintervals of others: $\mathbf{x}_i \subset \mathbf{x}_j$ (and $\mathbf{x}_i \neq \mathbf{x}_j$).

From the practical viewpoint, this situation makes perfect sense: the interval data may contain values measurement by more accurate measuring instruments – that produce narrower intervals $\mathbf{x}_i$ – and by less accurate measurement instruments – that produce wider intervals $\mathbf{x}_j$. When we measure the same value $x_i = x_j$, once with an accurate measurement instrument, and then with a less accurate instrument, then it is quite possible that the wider interval corresponding to the less accurate measurement properly contains the narrower interval corresponding to the more accurate instrument.

Similarly, if we measure close values $x_i \approx x_j$, it is quite possible that the wider interval coming from the less accurate instrument contains the narrower interval coming from the more accurate instrument.

In view of the above analysis, a natural way to avoid such difficult-to-compute situations is to restrict ourselves to situations when all the measurement are done with the same measuring instrument.

For a single measuring instrument, it is not very probable that in two different measurements, we get two intervals in which one is a proper subinterval of the other.

Let us show that in this case, we have a feasible algorithm for computing $\overline{V}$. For each interval $\mathbf{x} = [\underline{x}, \overline{x}]$, we will denote its half-width $(\overline{x} - \underline{x})/2$ by Δ, and its midpoint $(\underline{x} + \overline{x})/2$ by $\widetilde{x}$.

Definition 1. *By an* interval data*, we mean a finite set of intervals* $\mathbf{x}_1, \ldots, \mathbf{x}_n$.

Definition 2. *For n real numbers $x_1, \ldots, x_n$, their* variance $V(x_1, \ldots, x_n)$ *is defined in the standard way – as* $V \stackrel{\text{def}}{=} \dfrac{1}{n} \cdot \displaystyle\sum_{i=1}^{n} x_i^2 - E^2$*, where* $E \stackrel{\text{def}}{=} \dfrac{1}{n} \cdot \displaystyle\sum_{i=1}^{n} x_i$.

Definition 3. *By the* interval variance $\mathbf{V}$ *of the interval data, we mean the interval* $\mathbf{V} \stackrel{\text{def}}{=} \{V(x_1, \ldots, x_n) \mid x_i \in \mathbf{x}_i\}$ *filled by the values* $V(x_1, \ldots, x_n)$ *corresponding to different* $x_i \in \mathbf{x}_i$.

Theorem 1. *There exists an algorithm that computes the variance* $\mathbf{V}$ *of the interval data in time* $O(n \cdot \log(n))$ *for all the cases in which no element of the interval data is a subset of another element.*

Proof. In order to compute the interval $\mathbf{V}$, we must compute both endpoints $\underline{V}$ and $\overline{V}$ of this interval.

The proof of the theorem consists of three parts:

- in Part A, we will mention that the algorithm for computing $\underline{V}$ in time $O(n \cdot \log(n))$ is already known;
- in Part B, we describe a new algorithm for computing $\overline{v}$;
- in Part C, we prove that the new algorithm is correct and has the desired complexity.

A. Algorithm for computing $\underline{V}$ with desired time complexity is already known. The algorithm for computing $\underline{V}$ in time $O(n \cdot \log(n))$ is described in [4].

B. New algorithm for computing $\overline{V}$: description. The proposed algorithm for computing $\overline{V}$ is as follows:

- First, we sort n intervals $\mathbf{x}_i$ in lexicographic order:

$$\mathbf{x}_1 \leq_{\text{lex}} \mathbf{x}_2 \leq_{\text{lex}} \cdots \leq_{\text{lex}} \mathbf{x}_n,$$

  where $[\underline{a}, \overline{a}] \leq_{\text{lex}} [\underline{b}, \overline{b}]$ if and only if either $\underline{a} < \underline{b}$, or $\underline{a} = \underline{b}$ and $\overline{a} \leq \overline{b}$.

– Second, we use bisection to find the value k ($1 \le k \le n$) for which the following two inequalities hold:

$$\widetilde{x}_k + \frac{1}{n} \cdot \sum_{i=1}^{k-1} \Delta_i \le \frac{1}{n} \cdot \sum_{i=k+1}^{n} \Delta_i + \frac{1}{n} \cdot \sum_{i=1}^{n} \widetilde{x}_i; \tag{1}$$

$$\widetilde{x}_{k+1} + \frac{1}{n} \cdot \sum_{i=1}^{k} \Delta_i \ge \frac{1}{n} \cdot \sum_{i=k+2}^{n} \Delta_i + \frac{1}{n} \cdot \sum_{i=1}^{n} \widetilde{x}_i. \tag{2}$$

At each iteration of this bisection, we have an interval $[k^-, k^+]$ that is guaranteed to contain k. In the beginning, $k^- = 1$ and $k^+ = n$. At each stage, we compute the midpoint $k_{\text{mid}} = \lfloor (k^- + k^+)/2 \rfloor$, and check both inequalities (1) and (2) for $k = k_{\text{mid}}$. Then:

- If both inequalities (1) and (2) hold for his k, this means that we have found the desired k.
- If (1) holds but (2) does not hold, this means that the desired value k is larger than k_{mid}, so we keep k^+ and replace k^- with $k_{\text{mid}} + 1$.
- If (2) holds but (1) does not hold, this means that the desired value k is smaller than k_{mid}, so we keep k^- and replace k^+ with $k_{\text{mid}} - 1$.

– Once k is found, we compute

$$V_k \stackrel{\text{def}}{=} \frac{1}{n} \cdot \sum_{i=1}^{k} \underline{x}_i^2 + \frac{1}{n} \cdot \sum_{i=k+1}^{n} \overline{x}_i^2 - \left(\frac{1}{n} \sum_{i=1}^{k} \underline{x}_i + \frac{1}{n} \cdot \sum_{i=k+1}^{n} \overline{x}_i \right)^2. \tag{3}$$

This is the desired value $\overline{V}$.

C. Proof of correctness and complexity. Let us prove that this algorithm indeed produces the correct result and indeed requires time $O(n \cdot \log(n))$.

1°. Let us first prove that if no element of the interval data is a subset of another element, then, after we sort these elements in lexicographic order, both the lower endpoints $\underline{x}_i$ and the upper endpoints $\overline{x}_i$ are sorted in non-decreasing order: $\underline{x}_i \le \underline{x}_{i+1}$ and $\overline{x}_i \le \overline{x}_{i+1}$.

Indeed, by definition of a lexicographic order, we always have $\underline{x}_i \le \underline{x}_{i+1}$. If $\underline{x}_i = \underline{x}_{i+1}$, then, by definition of the lexicographic order, we have $\overline{x}_i \le \overline{x}_{i+1}$. If $\underline{x}_i < \underline{x}_{i+1}$, then we cannot have $\overline{x}_i \ge \overline{x}_{i+1}$ – otherwise, we would have $\mathbf{x}_{i+1} \subset \mathbf{x}_i$ – hence $\overline{x}_i < \overline{x}_{i+1}$. The statement is proven.

It is known that sorting requires time $O(n \cdot \log(n))$; see, e.g., [1].

In the following text, we will assume that the sequence of intervals has been sorted in this manner.

2°. Let us now prove that the desired maximum of the variance V is attained when each variable x_i is at one of the endpoints of the corresponding interval $\mathbf{x}_i$.

Indeed, if the maximum is attained in the interior point of this interval, this would means that in this point, $\partial V/\partial x_i = 0$ and $\partial^2 V/\partial x_i^2 \le 0$. For variance, $\partial V/\partial x_i = (2/n)\cdot(x_i - E)$, so $\partial^2 V/\partial x_i^2 = (2/n)\cdot(1-1/n) > 0$ – hence maximum cannot be inside.

3°. Let us show the maximum is attained at a vector

$$x = (\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n) \tag{4}$$

in which we first have lower endpoints and then upper endpoints.

What we need to prove is that there exists a maximizing vector in which, once we have an upper endpoint, what follows will also be an upper endpoint, i.e., in which we cannot have $x_k = \overline{x}_k > \underline{x}_k$ and $x_{k+1} = \underline{x}_{k+1} < \overline{x}_{k+1}$.

For that, let us start with a maximizing vector in which this property does not hold, i.e., in which $x_k = \overline{x}_k > \underline{x}_k$ and $x_{k+1} = \underline{x}_{k+1} < \overline{x}_{k+1}$ for some k. Based on this vector, we will now construct a different maximizing vector with the desired property. For that, let us consider two cases: $\Delta_k < \Delta_{k+1}$ and $\Delta_k \ge \Delta_{k+1}$, where $\Delta_i \stackrel{\text{def}}{=} (\overline{x}_i - \underline{x}_i)/2$ is the half-width of the interval $\mathbf{x}_i$.

In the first case, let us replace $\overline{x}_k = \underline{x}_k + 2\Delta_k$ with $\underline{x}_k$, and $\underline{x}_{k+1}$ with $\underline{x}_{k+1} + 2\Delta_k$ (since $\Delta_k < \Delta_{k+1}$, this new value is $< \overline{x}_{k+1}$). Here, the average E remains the same, so the only difference between the new value V' of the variance and its old value V comes from the change in terms x_k^2 and x_{k+1}^2. In other words,

$$V' - V = \frac{1}{n}\cdot((\underline{x}_{k+1} + 2\Delta_k)^2 - \underline{x}_{k+1}^2) - \frac{1}{n}\cdot((\underline{x}_k + 2\Delta_k)^2 - \underline{x}_k^2).$$

Opening parentheses and simplifying the resulting expression, we conclude that $V' - V = (4\Delta_k/n)\cdot(\underline{x}_{k+1} - \underline{x}_k)$. Since V is the maximum, we must have $V' - V \le 0$, hence $\underline{x}_{k+1} \le \underline{x}_k$. Due to our ordering, we thus have $\underline{x}_{k+1} = \underline{x}_k$. Since we assumed that $\Delta_k < \Delta_{k+1}$, we have $\overline{x}_k = \underline{x}_k + 2\Delta_k < \overline{x}_{k+1} = \underline{x}_{k+1} + 2\Delta_{k+1}$, hence the interval $\mathbf{x}_k$ is a proper subset of $\mathbf{x}_{k+1}$ – which is impossible.

In the second case, when $\Delta_k \ge \Delta_{k+1}$, let us replace $\overline{x}_k$ with $\overline{x}_k - 2\Delta_{k+1}$ (which is still $\ge \underline{x}_k$), and $\underline{x}_{k+1} = \overline{x}_{k+1} - 2\Delta_{k+1}$ with $\overline{x}_{k+1}$. Here, the average E remains the same, and the only difference between the new value V' of the variance and its old value V comes from the change in terms x_k^2 and x_{k+1}^2, hence

$$V' - V = \frac{1}{n}\cdot(\overline{x}_{k+1}^2 - (\overline{x}_{k+1} - 2\Delta_{k+1})^2) - \frac{1}{n}\cdot(\overline{x}_k^2 - (\overline{x}_k - 2\Delta_{k+1})^2),$$

i.e., $V' - V = (4\Delta_{k+1}/n)\cdot(\overline{x}_{k+1} - \overline{x}_k)$. Since V is the maximum, we must have $V' - V \le 0$, hence $\overline{x}_{k+1} \le \overline{x}_k$. Due to our ordering, we thus have $\overline{x}_{k+1} = \overline{x}_k$. Since we assumed that $\Delta_k \ge \Delta_{k+1}$, we have $\underline{x}_k = \overline{x}_k - 2\Delta_k \ge \underline{x}_{k+1} = \overline{x}_{k+1} - 2\Delta_{k+1}$, i.e., $\mathbf{x}_k \subseteq \mathbf{x}_{k+1}$. Since intervals cannot be proper subsets of each other, we thus have $\mathbf{x}_k = \mathbf{x}_{k+1}$. In this case, we can simply swap the values x_k and x_{k+1}, variance will not change.

If necessary, we can perform this swap for all needed k; as a result, we get the maximizing vector with the desired property.

4°. Due to Part 3 of this proof, the desired value $\overline{V} = \max V$ is the largest of $n+1$ values (3) corresponding to $k = 0, 1, \ldots, n$.

In principle, to compute $\overline{V}$, we can therefore compute each of these values and find the largest of them. Computing each value takes $O(n)$ times, so computing $n+1$ such values would require time $O(n^2)$. Let us show that we can compute $\overline{V}$ faster.

We must find the index k for which V_k is the largest. For the desired k, we have $V_k \ge V_{k-1}$ and $V_k \ge V_{k+1}$. Due to (3), we conclude that

$$V_k - V_{k-1} = \frac{1}{n} \cdot \left(\underline{x}_k^2 - \overline{x}_k^2\right)$$

$$- \left(\frac{1}{n} \cdot \sum_{i=1}^{k} \underline{x}_i + \frac{1}{n} \cdot \sum_{i=k+1}^{n} \overline{x}_i\right)^2 + \left(\frac{1}{n} \cdot \sum_{i=1}^{k-1} \underline{x}_i + \frac{1}{n} \cdot \sum_{i=k}^{n} \overline{x}_i\right)^2. \qquad (5)$$

Each pair of terms in the right-hand side of (5) can be simplified if we use the fact that $a^2 - b^2 = (a-b)\cdot(a+b)$ and use the notations Δ_k and $\widetilde{x}_k \stackrel{\text{def}}{=} (\underline{x}_k + \overline{x}_k)/2$. First, we get $\underline{x}_k^2 - \overline{x}_k^2 = (\underline{x}_k - \overline{x}_k) \cdot (\underline{x}_k + \overline{x}_k) = -4\Delta_k \cdot \widetilde{x}_k$. Second, we get

$$\left(\frac{1}{n} \cdot \sum_{i=1}^{k-1} \underline{x}_i + \frac{1}{n} \cdot \sum_{i=k}^{n} \overline{x}_i\right)^2 - \left(\frac{1}{n} \cdot \sum_{i=1}^{k} \underline{x}_i + \frac{1}{n} \cdot \sum_{i=k+1}^{n} \overline{x}_i\right)^2 =$$

$$\frac{2}{n} \cdot (\overline{x}_k - \underline{x}_k) \cdot \left(\frac{1}{n} \cdot \sum_{i=1}^{k-1} \underline{x}_i + \frac{1}{n} \cdot \widetilde{x}_k + \frac{1}{n} \cdot \sum_{i=k+1}^{n} \overline{x}_i\right).$$

Here, $\overline{x}_k - \underline{x}_k = 2\Delta_k$, hence the formula (5) takes the following form:

$$V_k - V_{k-1} = \frac{4}{n} \cdot \Delta_k \cdot \left(-\widetilde{x}_k + \frac{1}{n} \cdot \sum_{i=1}^{k-1} \underline{x}_i + \frac{1}{n} \cdot \widetilde{x}_k + \frac{1}{n} \cdot \sum_{i=k+1}^{n} \overline{x}_i\right).$$

Since $V_k \ge V_{k-1}$ and $\Delta_k > 0$, we conclude that

$$-\widetilde{x}_k + \frac{1}{n} \cdot \sum_{i=1}^{k-1} \underline{x}_i + \frac{1}{n} \cdot \widetilde{x}_k + \frac{1}{n} \cdot \sum_{i=k+1}^{n} \overline{x}_i \ge 0. \qquad (6)$$

Substituting the expressions $\underline{x}_i = \widetilde{x}_i - \Delta_i$ and $\overline{x}_i = \widetilde{x}_i + \Delta_i$ into the formula (6) and moving all the negative terms to the other side of the inequality, we get the inequality (1). Similarly, the inequality $V_{k+1} \le V_k$ leads to (2).

When k increases, the left-hand side of the inequality (1) increases – because $\widetilde{x}_k$ increases as the average of the two increasing values $\underline{x}_k$ and $\overline{x}_k$, and the sum is increasing. Similarly, the right-hand side of this inequality decreases with k. Thus, if this inequality holds for k, it should also hold for all smaller values, i.e., for $k-1$, $k-2$, etc.

Similarly, in the second desired inequality (2), when k increases, the left-hand side of this inequality increases, while the right-hand side decreases. Thus, if this inequality is true for k, it is also true for $k+1$, $k+2$, ...

If both inequalities (1) and (2) are true for two different values $k < k'$, then they should both be true for all the values intermediate between k and k', i.e., for $k+1, k+2, \ldots, k'-1$. If (1) and and (2) are both true for k and $k+1$, this means that in both cases, we have equality, thus $V_k = V_{k+1}$, so it does not matter which of these values k we take.

Thus, modulo this equality case, there is, in effect, only one k for which both inequalities are true, and this k can be found by the bisection method as described in the above algorithm.

How long does this algorithm take? In the beginning, we only know that k belongs to the interval $[1, n]$ of width $O(n)$. At each stage of the bisection step, we divide the interval (containing k) in half. After I iterations, we decrease the width of this interval by a factor of 2^I. Thus, to find the exact value of k, we must have I for which $O(n)/2^I = 1$, i.e., we need $I = O(\log(n))$ iterations. On each iteration, we need $O(n)$ steps, so we need a total of $O(n \cdot \log(n))$ steps. With $O(n \cdot \log(n))$ steps for sorting, and $O(n)$ for computing the variance, we get a $O(n \cdot \log(n))$ algorithm. □

3 Second Case: Using a Limited Number of Different Types of Measuring Instruments

In this case, the interval data consists of m families of intervals such that within each family, no two intervals are proper subsets of each other.

Similarly to the proof of Theorem 1, we can conclude that if we sort each family in lexicographic order, then, within each family, the maximum of V is attained on one of the sequences (4). Thus, to find the desired maximum $\overline{V}$, it is sufficient to know the value $k_\alpha \le n$ corresponding to each of m families. Overall, there are $\le n^m$ combinations of such values, and for each combination, computing the corresponding value of the variance requires $O(n)$ steps. Thus, overall, we need time $O(n^{m+1})$.

4 Third Case: Privacy in Statistical Databases

When the measurements $\widetilde{x}_i$ correspond to data that we want to keep private, e.g., health parameters of different patients, we do not want statistical programs to have full access to the data – because otherwise, by computing sufficiently many different statistics, we would be able to uniquely reconstruct the actual values $\widetilde{x}_i$. One way to prevent this from happening is to supply the statistical data processing programs not with the exact data, but only with intervals of possible values of this data, intervals corresponding to a fixed partition; see, e.g., [4]. For example, instead of the exact age, we tell the program that a person's age is between 30 and 40.

To implement the above idea, we need to fix a partition, i.e., to fix the values $t_1 < t_2 < \ldots < t_n$. In this case, instead of the actual value of the quantity, we return the partition-related interval $[t_i, t_{i+1}]$ that contains this value.

Privacy-related intervals $[t_i, t_{i+1}]$ satisfy the same property as intervals from the first case: none of them is a proper subset of the other. Thus, we can apply the algorithm described in Section 2 and compute the exact range $\mathbf{V}$ in polynomial time – namely, in time $O(n \cdot \log(n))$.

Acknowledgments. This work was supported in part by NASA grant NCC5-209, by the AFOSR 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 NIH grant 3T34GM008048-20S1.

The authors are thankful to the anonymous referees for the valuable suggestions.

References

1. Cormen Th. H., Leiserson C. E., Rivest R. L., and Stein C.: Introduction to Algorithms, MIT Press, Cambridge, MA, 2001.
2. Ferson, S., Ginzburg, L., Kreinovich, V., Longpré, L., Aviles, M.: Computing Variance for Interval Data is NP-Hard, ACM SIGACT News **33**(2) (2002) 108–118
3. Granvilliers, L., Kreinovich, V., Müller, L.: Novel Approaches to Numerical Software with Result Verification", In: Alt, R., Frommer, A., Kearfott, R. B., Luther, W. (eds.), Numerical software with result verification, Springer Lectures Notes in Computer Science, 2004, Vol. 2991, pp. 274–305.
4. Kreinovich, V., Longpré, L.: Computational complexity and feasibility of data processing and interval computations, with extension to cases when we have partial information about probabilities, In: Brattka, V., Schroeder, M., Weihrauch, K., Zhong, N.: Proc. Conf. on Computability and Complexity in Analysis CCA'2003, Cincinnati, Ohio, USA, August 28–30, 2003, pp. 19–54.