3-1-2006

Computing Variance under Interval Uncertainty: A New Algorithm and its Potential Application to Privacy in Statistical Databases

Richard Aló

Mohsen Beheshti

Gang Xiang

Comments:
Updated version UTEP-CS-06-10a.
Published in the *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMU'06*, Paris, France, July 2-7, 2006, pp. 810-816.

Computing Variance under Interval Uncertainty: A New Algorithm and Its Potential Application to Privacy in Statistical Databases

Richard Aló
Center for Computational Sciences & Advanced Distributed Simulation
Univ. of Houston-Downtown
Houston, TX 77002, USA
RAlo@uh.edu

Mohsen Beheshti
Computer Science
California State University, Dominguez Hills
Carson, CA 90747, USA
mbeheshti@csudh.edu

Gang Xiang
Dept. Computer Science
Univ. of Texas at El Paso
El Paso, TX 79968, USA
gxiang@utep.edu

Abstract

Computation of population mean $E = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i$ and population variance $V = \frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - E)^2$ is an important first step in statistical analysis. In many practical situations, we do not know the exact values of the sample quantities x_i, we only know the intervals $[\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$ that contain the actual (unknown) values of x_i. Different values of x_i from these intervals lead, in general, to different value of population variance. It is therefore desirable to compute the range $\mathbf{V} = [\underline{V}, \overline{V}]$ of possible values of V.

This problem of computing population variance under interval uncertainty is, in general, NP-hard. It is known that in some reasonable cases, there exist feasible algorithms for computing the interval $\mathbf{V}$: e.g., such algorithms are known for the case when for some constant c, any collection of more than c "narrowed" intervals $\left[\widetilde{x}_i - \frac{\Delta_i}{n}, \widetilde{x}_i + \frac{\Delta_i}{n}\right]$ has no common intersection, and for the case when none of the two narrowed intervals are subsets of each other.

In this paper, we provide a new polynomial time algorithm for computing population variance under interval uncertainty, an algorithm that is applicable to all situations where previously, feasible algorithms were known.

Keywords: Population Variance, Interval Uncertainty, Feasible Algorithms, Privacy in Statistical Databases

1 Formulation of the Problem

Computing population mean and variance is important. When we have n results $x_1, \ldots, x_n$ of repeated measurement of the same quantity at different moments of time and/or at different location, then the statistical analysis of these values usually starts with the computation of their population mean

$$E = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i$$

and population variance

$$V = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i^2 - E^2;$$

see, e.g., [13].

Interval uncertainty: practically important situation. Measurements are never 100% accurate. As a result, in many practical situations, for each i, instead of knowing the exact values of x_i, we only know the approximate value $\widetilde{x}_i$ and the measurement accuracy Δ_i. Based on this knowledge, we can conclude that the actual (unknown) value of the quantity x_i belongs to the interval $\mathbf{x}_i = [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$. For example, if

we measure the length x of a specimen with a ruler whose accuracy Δ is 1 cm, and the measured value $\widetilde{x}$ of the length is 15 cm, this means that the actual (unknown) value of the length x can be any number from the interval $[15-1, 15+1] = [14, 16]$.

Another example of interval uncertainty is when we use observations instead of measurements. For example, in a biological experiment, we make daily observation to find out when the seed germinates. If on the 5-th day, the seed did not germinate, and on the 6-th day it germinated, then the only thing we know about the exact moment t of germination is that t is between 5 and 6, i.e., that t belongs to the interval $[5, 6]$.

Yet another example of interval uncertainty comes from the need for privacy protection in statistical databases. To protect privacy, instead of recording the exact values of the desired quantity, e.g., salary, age, etc., we only store the range of possible values of this quantity. For example, instead of asking a respondent for his or her exact age, we only ask whether this age is from 10 to 20, from 20 to 30, etc. Similarly, instead of asking for the exact salary, we only ask whether the salary is between 0 and 10 K, between 10 and 20 K, etc. Once we have this database and we want to perform a statistical analysis of this data, we have to reply on the fact that we only know the intervals that contain the actual values: e.g., we only know that the age is between 30 and 40, and the salary is between 30 and 40 K.

Comment on representing interval uncertainty. A large number of intervals come from measurements. In such situations, the resulting intervals have the form $[\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$. However, in other situations – for example, for observations and for statistical databases – the intervals are described by their endpoints, i.e., these intervals are given as $[\underline{x}_i, \overline{x}_i]$. To simplify our analysis, we will reduce these intervals to the measurement-type form; we can always do it if we take $\widetilde{x}_i = \dfrac{\underline{x}_i + \overline{x}_i}{2}$ and $\Delta_i = \dfrac{\overline{x}_i - \overline{x}_i}{2}$. It is easy to check that in this case, $\widetilde{x}_i - \Delta_i = \underline{x}_i$ and $\widetilde{x}_i + \Delta_i = \overline{x}_i$, i.e., that $[\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$ is indeed the new representation of the original interval $[\underline{x}_i, \overline{x}_i]$.

It is necessary to compute population mean and population variance under interval uncertainty. In the situations of interval uncertainty, we only know the intervals $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ of possible values of x_i. For different possible values $x_i \in \mathbf{x}_i$, we get different values of E and V. In such situations, it is desirable to find the ranges $\mathbf{E}$ and $\mathbf{V}$ of possible values of E and V.

Since both E and V are continuous functions of all the variables x_i, the resulted ranges of E and V are also intervals. So, in the situations of interval uncertainty, our objective is to compute the intervals $\mathbf{E}$ and $\mathbf{V}$ of possible values of E and V:

$$\mathbf{E} = [\underline{E}, \overline{E}] = \left\{ \frac{1}{n} \cdot \sum_{i=1}^{n} x_i : x_i \in \mathbf{x}_i, i = 1, \ldots, n \right\},$$

$$\mathbf{V} = [\underline{V}, \overline{V}] = \left\{ \frac{1}{n} \cdot \sum_{i=1}^{n} x_i^2 - E^2 : x_i \in \mathbf{x}_i, i = 1, \ldots, n \right\}.$$

The practical importance of the problem of computing population variance under interval uncertainty was emphasized, e.g., in [11, 12] on the example of processing geophysical data and in [3] on the example of processing environmental data.

What is known: computing population mean under interval uncertainty. Population mean is an increasing function of all its variables. Thus, its largest value is attained when all x_i are the largest, and its smallest value is attained when all x_i attain the smallest possible values. In other words, the interval $[\underline{E}, \overline{E}]$ for the population mean can be computed as follows:

$$\underline{E} = \frac{1}{n} \cdot \sum_{i=1}^{n} \underline{x}_i, \quad \overline{E} = \frac{1}{n} \cdot \sum_{i=1}^{n} \overline{x}_i$$

What is known: computing population variance under interval uncertainty. In contrast to the population mean, computing the range $\mathbf{V}$ of the population variance is NP-hard. To be more precise, it is possible to compute the lower endpoint $\underline{V}$ in time $O(n \cdot \log(n))$ [6], but computing $\overline{V}$ is, in general, an NP-hard problem [4, 5].

NP-hardness means that there is no general algorithm for computing $\overline{V}$ in all possible cases. As we have shown in [8], there are practically useful cases when a feasible algorithm for computing $\overline{V}$ is possible.

First case: case of narrow intervals. It is the case when the intervals $\mathbf{x}_i$ do not intersect with each other.

The practical meaning of this case is that the measuring instruments are so accurate that, in spite of the measurement errors $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i \neq 0$, we can always distinguish different measurement results from each other.

Second case: case of slightly wider narrow intervals. In this case, two intervals $\mathbf{x}_i$ and $\mathbf{x}_j$ may have a common point, but for some integer $c > 1$, any collection of $c+1$ intervals $\mathbf{x}_i$ does not have any common point.

This case is more general case than the first case, and its practical meaning is that the measuring instruments are somewhat less accurate, but still out of any $c+1$ measurement results, we should be able to distinguish at least some of them even after imprecise measurements.

Third case: case of even wider narrow intervals. In this case, we require the $c+1$ no-intersection property not for the original intervals $\mathbf{x}_i$, but for the *narrowed intervals* $\left[\widetilde{x}_i - \dfrac{\Delta_i}{n}, \widetilde{x}_i + \dfrac{\Delta_i}{n}\right]$.

Fourth case: case of a single measuring instrument. In principle, it is possible that some measurements are performed with a high-accurate measuring instrument – that leads to narrower intervals, and some measurements are performed with a low-accuracy measuring instrument – which leads to wider intervals. In such situations, it is possible that one of the resulting intervals $\mathbf{x}_i$ is a proper subinterval of the (interior of the) other – in the sense that $[\underline{x}_i, \overline{x}_i] \subseteq (\underline{x}_j, \overline{x}_j)$. In many real-life situations, all the measurements are performed by the same measurement instrument; in such situations, the above inclusion is not possible.

We can therefore formally define a sequence of intervals $\mathbf{x}_i$ as *coming from a single measuring instrument* if none of these intervals are proper subsets of the interior of the others, i.e., if $[\underline{x}_i, \overline{x}_i] \not\subseteq (\underline{x}_j, \overline{x}_j)$ for all i and j.

Fifth case: case of same accuracy measurement. It is the case when all non-degenerate intervals $\mathbf{x}_i$'s have the same half-width Δ_i. In practice, this case describes the situation when all the measurements are performed with exactly the same accuracy.

One can easily check that this case is a subcase of the case of single measuring instrument.

Sixth case: case of several measuring instruments In this case, for some integer m, the data intervals $\mathbf{x}_i$ can be divided into m subgroups such that within each of these subgroups, none of the intervals are proper subsets of each other – and we know which interval belongs to which subgroup.

In practice, this case corresponds to the situation when several (m) measuring instruments are used, and we know which measurement was performed with which instrument.

Seventh case: privacy case. This is the case when we have a fixed partition of the real line, and all intervals come from this partition. In precise terms, we fix values $x_{(1)} < x_{(2)} < \ldots < x_{(m)}$, and we are only allowing intervals of the type $[x_{(k)}, x_{(k+1)}]$.

In practice, as we have mentioned earlier, this corresponds to the data processing of the values from a statistical database, where, to protect privacy, we only store the ranges of the corresponding values.

Eighth case: case of non-detects. This is the case when each data interval $\mathbf{x}_i$, is either non-degenerate, or has the form $[0, DL_i]$ for some

value $DL_i > 0$.

In practice, this case corresponds to the situations when the only source of uncertainty is detection limits. In such situations, every measurement result is either exact or is a *non-detect*, i.e., an interval $[0, DL_i]$ for an appropriate detection limit DL_i. (Different sensors can have different detection limits.)

From the mathematical viewpoint, this case is a particular subcase of the single measuring instrument case.

Recent case. In [2], it was shown that a feasible algorithm for computing $\overline{V}$ is possible in the case when none of the *narrowed* intervals are proper subintervals of one another. This is clearly a stronger requirement that the case of a single measuring instrument.

Our new result. In this paper, we provide a polynomial time algorithm for computing $\overline{V}$ in the most general case, which covers all above-described cases as subcases.

2 Analysis of the Problem

One can easily check that all the cases for which an $O(n \cdot \log(n))$ algorithm is known are particular cases of one of the following two cases:

- the recent case – when no two narrowed intervals are proper subsets of each other, and
- the third case – when every collections of $> c$ narrowed intervals $\mathbf{X}_i$ has an empty intersection: $\mathbf{X}_{i_1} \cap \ldots \cap \mathbf{X}_{i_{c+1}} = \emptyset$.

There is also the case of $m > 1$ measuring instruments, for which an $O(n^m)$ algorithm is possible. Thus, to provide the most general case, we must describe the case that includes these three situations as subcases.

3 Formulation of the New Result

We will consider the case described by two parameters $m \geq 1$ and $c \geq 1$. In this case, we can divide the intervals $\mathbf{x}_i$ into m subclasses such that:

- the first $m-1$ subclasses have the property that within each subclass, no two narrowed intervals are proper subsets of each other;
- the last class either has the same property, or it has the property that every collections of $> c$ narrowed intervals from this class has an empty intersection.

Our algorithm will require time $O(n \cdot \log(n))$ when $m = 1$ and time $O(n^m)$ when $m > 1$.

One can easily check that all three above cases are indeed particular cases of the above situation. Namely, the recent case and the third case correspond to $m = 1$ – and moreover, $m = 1$ consists of exactly these two cases.

4 Towards the New Algorithm

The new algorithm builds on the known algorithms for the corresponding three subcases.

Algorithm for the recent case: main idea. The algorithm for the recent case is based on the following idea. Since no two narrowed intervals $\mathbf{X}_i = [\underline{X}_i, \overline{X}_i]$ are proper subsets of one another, we can sort them in lexicographic order – i.e., in the order where $\mathbf{X}_i \leq \mathbf{X}_j$ if and only if either $\underline{X}_i < \underline{X}_j$ or $(\underline{X}_i = \underline{X}_j$ and $\overline{X}_i \leq \overline{X}_j)$. After the sorting, it can then be shown that the maximum of V is attained at one of sequences of the type $(\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$. Thus, $\overline{V}$ is equal to the largest of the values V corresponding to $n+1$ such sequences.

For each value k, for the corresponding sequence, computing $V = M_2 - E$, where $M_2 = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i^2$, requires linear time. Actually, we only need to spend this time to compute E, M_2, and V for $k = 0$. After we computed M_2 and E for some k, going from the case k to the case $k+1$ means replacing only one term in E and in M_2 ($\overline{x}_{k+1}$ by $\underline{x}_{k+1}$), and thus, requires

constant number of steps:

$$E \to E + \frac{1}{n} \cdot (\underline{x}_{k+1} - \overline{x}_{k+1});$$

$$M_2 \to M_2 + \frac{1}{n} \cdot ((\underline{x}_{k+1})^2 - (\overline{x}_{k+1})^2).$$

So, after sorting (which requires time

$$O(n \cdot \log(n));$$

see, e.g., [1]), we only need linear time to compute V for all $n+1$ candidates for an optimal sequence – and after that, linear time to find the largest of these n values as $\overline{V}$.

Algorithm for the third case: main idea. The algorithm for the third case is based on the following idea. First, we sort all the endpoints of the narrowed intervals – and thus divide the real line into zones $[x_{(k)}, x_{(k+1)}]$. The actual (unknown) mean E must belong to one of these $2n+1$ zones. If E belongs to the k-th zone, then in the optimal sequence $(x_1, \ldots, x_n)$, for every i, we should have:

- if $\overline{x}_i \le x_{(k)}$, then $x_i = \underline{x}_i$;
- if $x_{(k+1)} \le \underline{x}_i$, then $x_i = \overline{x}_i$;
- in all other cases, both $x_i = \underline{x}_i$ and $x_i = \overline{x}_i$ are possible.

Due to the condition describing the third case, for each zone, there are no more than c such "un-decided" intervals. So, for each zone, we have to consider, at most, 2^c (constant number) of possible sequences.

For each of these sequences, computing requires linear time. Transition from each zone to the next one may involve changing several terms, but since eventually, every endpoint changes only at one zone, we also have a linear overtime in addition to $O(n \cdot \log(n))$ time for sorting.

Algorithm for the case of m measuring instruments. In this case, we can prove that for each subset corresponding to a measuring instrument, the optimal sequence has the form $(\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$. Thus, to find a (global) optimal sequence, it is sufficient to consider all possible combinations of the indices $k_1, \ldots, k_m$ corresponding to m subsequences.

We start, e.g., with the case $k_1 = \ldots = k_m = 0$ for which we need linear time to compute E, M_2, and V. Similarly to the recent case, once we know the value for each subsequence, moving to the next subsequence $k_1, \ldots, k_{i-1}, k_i + 1, k_{i+1}, \ldots, k_n$ requires a constant number of steps. Thus, to cover all $\le n^m$ subsequences, we need a total of $O(n^m)$ steps. Overall, we need $O(n \cdot \log(n))$ steps for the original sorting, $O(n)$ steps for computing the initial value of V, and then $O(n^m)$ steps to compute the values of V corresponding to all possible combinations $(k_1, \ldots, k_m)$ – and to find the largest of these values, i.e., $\overline{V}$. For $m \ge 2$, the resulting overall computation time is thus $O(n^m)$.

Main idea behind the new algorithm. One can show that the optimization selection ideas behind these three algorithms, in effect, do not change if, instead of considering all n intervals, we only consider a subset of the intervals.

For example, the arguments similar to the ones presented in [2] show that if a *subsequence* of the original sequence of intervals has the property that no two narrowed subintervals from this subsequence are proper intervals of one another, then for this subsequence, the maximum value of V is attained at one of the sequences of the type $(\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$.

The only (minor) difference is with the third case, since for this case, we are no longer talking about a zone that contains the mean E – just one of the zones.

Thus, we arrive at the following algorithm.

Algorithm. For $m = 1$, depending on the situation, we can use either the algorithm for the recent case or the algorithm for the third case. So, to describe the algorithm, it is sufficient to consider the case when $m \ge 2$.

By definition, the set of intervals can be di-

vided into m groups. Within each group, we perform the appropriate sorting. Then, we know that the value $\overline{V}$ is attained when for each of the $m-1$ subgroups, we have a sequence of the type $(\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$ for an appropriate $k = k_j$, and for the last group, a sequence for which:

- if $\overline{x}_i \le x_{(k)}$, then $x_i = \underline{x}_i$;
- if $x_{(k+1)} \le \underline{x}_i$, then $x_i = \overline{x}_i$;

for some parameter $k = k_m$.

Similar to the third case, for each combination $(k_1, \ldots, k_{m-1})$, checking all possible values of k_m requires time $O(n)$. Thus, for all $\le n^{m-1}$ possible combinations $(k_1, \ldots, k_{m-1})$, we need to spend $O(n)$ time – to the total of $O(n^m)$.

5 Often, We Do Not Need to Know Which Interval Belongs to Which Subgroup

In our description of the new algorithm, we assumed that the original set of n intervals can be divided into m subsets, and that we know which interval belongs to which subset. It turns out that in the case when all m subsets have a no-proper-subset (nps) property, there is no need to explicitly describe the corresponding m subsets – it is sufficient to know that it is, in principle, possible to subdivide the original set of n intervals into m subsets with this property.

This possibility can be, in turn, described as follows. Based on the original intervals $\mathbf{x}_i$, we can form the following directed graph:

- its vertices are the original intervals, and
- an edge $\mathbf{x}_i \to \mathbf{x}_j$ is going from the interval $[\underline{x}_i, \overline{x}_i]$ to the interval $[\underline{x}_j, \overline{x}_j]$ if and only the i-th interval is a proper subset of the (interior of the) j-th one, i.e., if and only if $[\underline{x}_i, \overline{x}_i] \subseteq (\underline{x}_j, \overline{x}_j)$.

It is easy to see that this graph is acyclic – so each chain has at most n elements in it. By the *height* h of this graph, we mean the largest length of a chain $\mathbf{x}_{i_1} \to \mathbf{x}_{i_2} \to \ldots \to \mathbf{x}_{i_h}$ from this graph.

The following statement describes the relation between the height of the graph and the number of subgroups:

- if intervals can be divided into m subgroups with the no-proper-subset (nps) property, then the height of the corresponding graph is $\le m$;
- vice versa, if the height of the corresponding graph is m, then we can (efficiently) divide the original intervals into m subgroups with the no-proper-subset property.

Indeed, if we can divide intervals into m nps subgroups, then we cannot have a chain of length $> m$: otherwise, at least two intervals from this chain will be in the same subgroups – and since every two elements from a chain are proper subsets of each other, this would violate the nps property.

Vice versa, if we have a graph of height m, then we can do the following:

- We take all elements which are not dominated by anyone else as the first subgroup. It is easy to see that this group has a nps property.
- After deleting elements from the first group, we can again consider those who are not dominated by anyone in the remaining graph – these will form the second subgroup.
- etc.

One can check that each interval $\mathbf{x}_i$ will be assigned to the group whose number k is the largest length of the chain leading to $\mathbf{x}_i$. Since the height of the graph is m, we will thus subdivide all n original intervals into to m subgroups with the nps property.

The statement has been proven.

Acknowledgements

This work was supported in part by NASA under cooperative agreement NCC5-209, NSF

grants EAR-0225670 and DMS-0532645, Star Award from the University of Texas System, and Texas Department of Transportation grant No. 0-5453

References

[1] Th. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 2001.

[2] E. Dantsin, V. Kreinovich, A. Wolpert, and G. Xiang, "Population Variance under Interval Uncertainty: A New Algorithm", *Reliable Computing*, 2006, Vol. 12, No. 4, pp. 273–280.

[3] S. Ferson, *RAMAS Risk Calc 4.0: Risk Assessment with Uncertain Numbers*, CRC Press, Boca Raton, Florida, 2002.

[4] S. Ferson, L. Ginzburg, V. Kreinovich, L. Longpré, and M. Aviles, "Computing Variance for Interval Data is NP-Hard", *ACM SIGACT News*, 2002, Vol. 33, pp. 108–118.

[5] S. Ferson, L. Ginzburg, V. Kreinovich, L. Longpré, and M. Aviles, "Exact Bounds on Finite Populations of Interval Data", *Reliable Computing*, 2005, Vol. 11, No. 3, pp. 207–233.

[6] L. Granvilliers, V. Kreinovich, and N. Müller, "Novel Approaches to Numerical Software with Result Verfication", In: R. Alt, A. Frommer, R. B. Kearfott, and W. Luther, eds., *Numerical Software with Result Verification*, Proceedings of the International Dagstuhl Seminar, Dagstuhl Castle, Germany, January 19–24, 2003, Springer Lectures Notes in Computer Science, 2004, Vol. 2991, pp. 274–305.

[7] R. B. Kearfott and V. Kreinovich (eds.), *Applications of Interval Computations*, Kluwer, Dordrecht, 1996.

[8] V. Kreinovich, G. Xiang, S. A. Starks, L. Longpré, M. Ceberio, R. Araiza, J. Beck, R. Kandathi, A. Nayak, R. Torres, and J. Hajagos, "Towards combining probabilistic and interval uncertainty in engineering calculations: algorithms for computing statistics under interval uncertainty, and their computational complexity, *Reliable Computing* (to appear).

[9] V. P. Kuznetsov, *Interval statistical models*, Moscow, Radio i Svyaz Publ., 1991 (in Russian).

[10] R. E. Moore, *Methods and Applications of Interval Analysis*, SIAM, Philadelphia, 1979.

[11] P. Nivlet, F. Fournier, and J. Royer, "A new methodology to account for uncertainties in 4-D seismic interpretation", *Proceedings of the 71st Annual International Meeting of the Society of Exploratory Geophysics SEG'2001*, San Antonio, Texas, September 9–14, 2001, pp. 1644–1647.

[12] P. Nivlet, F. Fournier, and J. Royer, "Propagating interval uncertainties in supervised pattern recognition for reservoir characterization", *Proceedings of the 2001 Society of Petroleum Engineers Annual Conference SPE'2001*, New Orleans, Louisiana, September 30 – October 3, 2001, paper SPE-71327.

[13] S. Rabinovich, *Measurement Errors: Theory and Practice*, American Institute of Physics, New York, 1993.

[14] P. Walley, *Statistical reasoning with imprecise probabilities*, Chapman and Hall, N.Y., 1991.

[15] G. W. Walster, "Philosophy and practicalities of interval arithmetic", In: R. E. Moore (ed.), *Reliability in Computing*, 1988, pp. 307–323.

[16] G. Xiang, "Fast algorithm for computing the upper endpoint of sample variance for interval data: case of sufficiently accurate measurements", *Reliable Computing*, 2006, Vol. 12, No. 1, pp. 59–64.