

# Fast Algorithm for Computing the Upper Endpoint of Sample Variance for Interval Data: Case of Sufficiently Accurate Measurements

Gang Xiang  
Department of Computer Science  
University of Texas at El Paso  
500 W. University  
El Paso, TX 79968, USA  
gxian@utep.edu

## Abstract

When we have  $n$  results  $x_1, \dots, x_n$  of repeated measurement of the same quantity, the traditional statistical approach usually starts with computing their sample average  $E$  and their sample variance  $V$ . Often, due to the inevitable measurement uncertainty, we do not know the exact values of the quantities, we only know the intervals  $\mathbf{x}_i$  of possible values of  $x_i$ . In such situations, for different possible values  $x_i \in \mathbf{x}_i$ , we get different values of the variance. We must therefore find the range  $\mathbf{V}$  of possible values of  $V$ . It is known that in general, this problem is NP-hard. For the case when the measurements are sufficiently accurate (in some precise sense), it is known that we can compute the interval  $\mathbf{V}$  in quadratic time  $O(n^2)$ . In this paper, we describe a new algorithm for computing  $\mathbf{V}$  that requires time  $O(n \cdot \log(n))$  (which is much faster than  $O(n^2)$ ).

## 1 Introduction

**Computing sample variance is important.** When we have  $n$  results  $x_1, \dots, x_n$  of repeated measurement of the same quantity (at different points, or at different moments of time), the traditional statistical approach usually starts with computing their sample average

$$E = \frac{x_1 + \dots + x_n}{n} \tag{1}$$

and their (sample) variance

$$V = \frac{1}{n} \cdot \sum_{i=1}^n (x_i - E)^2 = \frac{x_1^2 + \dots + x_n^2}{n} - E^2 \quad (2)$$

(or, equivalently, the sample standard deviation  $\sigma = \sqrt{V}$ ); see, e.g., [14].

**Problem: computing sample variance under interval uncertainty.**

Measurements are never 100% accurate, so in reality, the actual value  $x_i$  of the  $i$ -th measured quantity can differ from the measurement result  $\tilde{x}_i$ . Often, we only know the intervals  $\mathbf{x}_1, \dots, \mathbf{x}_n$  of possible values of  $x_i$ . In this case, 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 of possible values of  $E$  and  $V$ .

Since both  $E$  and  $V$  are continuous functions of  $n$  variables  $x_1 \in \mathbf{x}_1, \dots, x_n \in \mathbf{x}_n$ , the range of each of these functions on the box  $\mathbf{x}_1 \times \dots \times \mathbf{x}_n$  is an interval. So, in such situations, our objective is to compute the intervals  $\mathbf{E}$  and  $\mathbf{V}$  of possible values of  $E$  and  $V$ :

$$\mathbf{E} = [\underline{E}, \overline{E}] \stackrel{\text{def}}{=} \left\{ \frac{x_1 + \dots + x_n}{n} \mid x_1 \in \mathbf{x}_1 \ \& \ \dots \ \& \ x_n \in \mathbf{x}_n \right\};$$

$$\mathbf{V} = [\underline{V}, \overline{V}] \stackrel{\text{def}}{=} \left\{ \frac{x_1^2 + \dots + x_n^2}{n} - E^2 \mid x_1 \in \mathbf{x}_1 \ \& \ \dots \ \& \ x_n \in \mathbf{x}_n \right\}.$$

The practical importance of the problem of computing sample variance under interval uncertainty was emphasized, e.g., in [7, 8] on the example of processing geophysical data and in [3] on the example of processing environmental data.

**What is known.** For  $E$ , the straightforward interval computations [9, 10, 11, 13] lead to the exact range:

$$\mathbf{E} = \frac{\mathbf{x}_1 + \dots + \mathbf{x}_n}{n}, \text{ i.e., } \underline{E} = \frac{\underline{x}_1 + \dots + \underline{x}_n}{n}, \text{ and } \overline{E} = \frac{\overline{x}_1 + \dots + \overline{x}_n}{n}.$$

For  $V$ , straightforward interval computations lead to an excess width, and moreover, the problem of computing the range  $\mathbf{V}$  is, in general, NP-hard [5] (this result originally appeared in [4]).

In [5], it was shown that we can compute the lower endpoint  $\underline{V}$  of the desired range in quadratic time  $O(n^2)$ . For the upper bound  $\overline{V}$  of the desired range, in [5], it was proven that we can compute it in quadratic time if the measurements are sufficiently accurate in the sense that different measurement results can still be distinguished from each other – i.e., when intervals  $\mathbf{x}_i$  corresponding to different measurement do not intersect.

Moreover, it was proven that a quadratic time algorithm is possible not only when the original intervals  $[\tilde{x}_i - \Delta_i, \tilde{x}_i + \Delta_i]$  do not intersect, but also in a

more general case when the “narrowed” intervals  $[\tilde{x}_i - \Delta_i/n, \tilde{x}_i + \Delta_i/n]$  do not intersect. In fact, this quadratic time algorithm even works in the case when for some integer  $c < n$ , no sub-collection of greater than  $c$  narrowed intervals of  $\mathbf{x}_i$  has a common intersection [5].

For large  $n$ ,  $n^2$  is still a lot of time; it is therefore desirable to speed up the computations.

Many applications fall into one of two cases: (i) measurement error is constant, either in absolute or as a fraction of  $x$  (e.g., if we the same physical instrument to get all the measurement results), (ii) the real line is partitioned into pre-assigned bins, and one learns the bin into which each observation falls (e.g., income brackets). In either of these cases, we can compute the range of variance in time  $O(n \cdot \log(n))$  [1, 15, 16]. However, if we use different measuring instruments, then the measurement error is no longer constant, so we cannot directly apply the algorithms developed for these special cases.

In [6], it was shown that in the general case of “sufficiently accurate” measurements, we can compute  $\underline{V}$  in time  $O(n \cdot \log(n))$  – which is much faster than  $O(n^2)$ . A natural question is: can we similarly speed up the computation of  $\overline{V}$ ?

**What we are planning to do.** In this paper, we describe a new algorithm for computing  $\mathbf{V}$  that requires time  $O(n \cdot \log(n))$  in the case when for some integer  $c$ , no sub-collection of more than  $c$  narrowed intervals of  $\mathbf{x}_i$  has a common intersection.

## 2 Previously Known Quadratic-Time Algorithm: A Brief Reminder

The *input* to our problem is a finite list of intervals  $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$ . There are two standard ways to represent an interval in the computer:

- first, by describing two real numbers  $\underline{x}_i$  and  $\overline{x}_i$ ;
- second, by describing the midpoint  $\tilde{x}_i \stackrel{\text{def}}{=} (\underline{x}_i + \overline{x}_i)/2$  and the half-width  $\Delta_i \stackrel{\text{def}}{=} (\overline{x}_i - \underline{x}_i)/2$  of this interval.

Once we know the midpoint and the half-width, we can reconstruct the endpoints of the interval as  $\underline{x}_i = \tilde{x}_i - \Delta_i$  and  $\overline{x}_i = \tilde{x}_i + \Delta_i$ .

We have already mentioned that we consider the case when for some given integer  $c$ , no sub-collection of more than  $c$  narrowed intervals  $[\tilde{x}_i - \Delta_i/n, \tilde{x}_i + \Delta_i/n]$  has a common intersection.

For this situation, the following quadratic-time algorithm for computing  $\overline{V}$  was described in [5]:

- First, we sort all  $2n$  endpoints of the narrowed intervals  $\tilde{x}_i - \Delta_i/n$  and  $\tilde{x}_i + \Delta_i/n$  into a sequence  $x_{(1)} \leq x_{(2)} \leq \dots \leq x_{(2n)}$ . This enables us

to divide the real line into  $2n + 1$  zones  $[x_{(k)}, x_{(k+1)}]$ , where we denote  $x_{(0)} \stackrel{\text{def}}{=} -\infty$  and  $x_{(2n+1)} \stackrel{\text{def}}{=} +\infty$ .

- Second, we compute  $\underline{E}$  and  $\overline{E}$  and pick all zones  $[x_{(k)}, x_{(k+1)}]$  that intersect with  $[\underline{E}, \overline{E}]$ .
- For each of remaining zones  $[x_{(k)}, x_{(k+1)}]$ , for each  $i$  from 1 to  $n$ , we pick the following value of  $x_i$ :
  - if  $x_{(k+1)} \leq \tilde{x}_i - \Delta_i/n$ , then we pick  $x_i = \overline{x}_i$ ;
  - if  $\tilde{x}_i + \Delta_i/n \leq x_{(k)}$ , then we pick  $x_i = \underline{x}_i$ ;
  - for all other  $i$ , we consider both possible values  $x_i = \overline{x}_i$  and  $x_i = \underline{x}_i$ .
- As a result, we get one or several sequences of  $x_i$ . For each of these sequences, we check whether the average  $E$  of the selected values  $x_1, \dots, x_n$  is indeed within this zone, and if it is, compute the variance by using the formula (2).
- Finally, we return the largest of the computed variances as  $\overline{V}$ .

The proof that this algorithm requires only  $O(n^2)$  time is based on the fact that for each zone, there are at most  $c$  indices  $i$  for which the  $i$ -th narrowed interval  $[\tilde{x}_i - \Delta_i/n, \tilde{x}_i + \Delta_i/n]$  contains this zone and therefore, at most  $c$  indices for which we had to consider both choices  $\underline{x}_i$  and  $\overline{x}_i$ . As a result, for each zone, there are at most  $2^c$  corresponding sequences  $x_i$ .

### 3 New Algorithm

1°. Let us first sort the lower endpoints  $\tilde{x}_i - \Delta_i/n$  of the narrowed intervals into an increasing sequence. Without losing generality, we can therefore assume that these lower endpoints are ordered in increasing order:

$$\tilde{x}_1 - \Delta_1/n \leq \tilde{x}_2 - \Delta_2/n \leq \dots$$

It is well known that sorting requires time  $O(n \cdot \log(n))$ ; see, e.g., [2].

2°. Then, similar to the previously known algorithm, we sort *all* the endpoints of the narrowed intervals into a sequence  $x_{(1)} \leq x_{(2)} \leq \dots \leq x_{(k)} \leq \dots \leq x_{(2n)}$ . Sorting means that for every  $i$ , we know which element  $k^-(i)$  represents the lower endpoint of the  $i$ -th narrowed interval and which element  $k^+(i)$  represents the upper endpoint of the  $i$ -th narrowed interval.

This sorting also requires  $O(n \cdot \log(n))$  steps.

3°. On the third stage, we produce, for each of the resulting zones  $[x_{(k)}, x_{(k+1)}]$ , the set  $S_k$  of all the indices  $i$  for which the  $i$ -th narrowed interval

$$[\tilde{x}_i - \Delta_i/n, \tilde{x}_i + \Delta_i/n]$$

contains this zone.

As we have mentioned, for each  $i$ , we know the value  $k = k^-(i)$  for which  $\tilde{x}_i - \Delta_i/n = x_{(k)}$ . So, for each  $i$ , we place  $i$  into the set  $S_{k^-(i)}$  corresponding to the zone  $[x_{(k^-(i))}, x_{(k^-(i)+1)}]$ , into the set corresponding to the next zone, etc., until we reach the zone for which the upper endpoint is exactly  $\tilde{x}_i + \Delta_i/n$ .

Here, we need one computational step for each new entry of  $i$  into the set corresponding to a new zone. Therefore, filling in all these sets requires as many steps as there are items in all these sets. For each of  $2n + 1$  zones, as we have mentioned, there are no more than  $c$  items in the corresponding set; therefore, overall, all the sets contain no more than  $c \cdot (2n + 1) = O(n)$  steps. Thus, this stage requires  $O(n)$  time.

4°. On the fourth stage, for all integers  $p$  from 0 to  $n$ , we compute the sums

$$E_p \stackrel{\text{def}}{=} \frac{1}{n} \cdot \sum_{i=1}^p \underline{x}_i + \frac{1}{n} \cdot \sum_{i=p+1}^n \bar{x}_i;$$

$$M_p \stackrel{\text{def}}{=} \frac{1}{n} \cdot \sum_{i=1}^p (\underline{x}_i)^2 + \frac{1}{n} \cdot \sum_{i=p+1}^n (\bar{x}_i)^2.$$

We compute these values sequentially. Once we know  $E_p$  and  $M_p$ , we can compute  $E_{p+1}$  and  $M_{p+1}$  as  $E_{p+1} = E_p + \underline{x}_{p+1} - \bar{x}_{p+1}$  and  $M_{p+1} = M_p + (\underline{x}_{p+1})^2 - (\bar{x}_{p+1})^2$ .

Transition from  $E_p$  and  $M_p$  to  $E_{p+1}$  and  $M_{p+1}$  requires a constant number of computational steps; so overall, we need  $O(n)$  steps to compute all the values  $E_p$  and  $M_p$ .

5°. Finally, for each zone  $k$ , we compute the corresponding values of the variance. For that, we first find the smallest index  $i$  for which  $x_{(k+1)} \leq \tilde{x}_i - \Delta_i/n$ . We will denote this value  $i$  by  $p(k)$ .

Since the values  $\tilde{x}_i - \Delta_i/n$  are sorted, we can find this  $i$  by using bisection [2]. It is known that bisection requires  $O(\log(n))$  steps, so finding such  $p(k)$  for all  $2n + 1$  zones requires  $O(n \cdot \log(n))$  steps.

Once  $i \geq p(k)$ , then  $\tilde{x}_i - \Delta_i/n \geq \tilde{x}_{p(k)} - \Delta_{p(k)}/n \geq x_{(k+1)}$ . So, in accordance with the above justification for the quadratic-time algorithm, we should select  $x_i = \bar{x}_i$ , as in the sums  $E_{p(k)}$  and  $M_{p(k)}$ .

In accordance with the same justification, the only values  $i < p(k)$  for which we may also select  $x_i = \bar{x}_i$  are the values for which the  $i$ -th narrowed intervals contains this zone. These values are listed in the set  $S_k$  of no more than  $c$  such intervals. So, to find all possible values of  $V$ , we can do the following.

We then consider all subsets  $s \subseteq S_k$  of the set  $S_k$ ; there are no more than  $2^c$  such subsets. For each subset  $s$ , we replace, in  $E_{p(k)}$  and  $M_{p(k)}$ , values  $\underline{x}_i$  and  $(\underline{x}_i)^2$  corresponding to all  $i \in s$ , with, correspondingly,  $\bar{x}_i$  and  $(\bar{x}_i)^2$ .

Each replacement means subtracting no more than  $c$  terms and then adding no more than  $c$  terms, so each computation requires no more than  $2c$  steps.

Once we have  $E$  and  $V$  corresponding to the subset  $s$ , we can check whether  $E$  belongs to the analyzed zone and, if yes, compute  $V = M - E^2$ .

For each subset, we need no more than  $2c + 2$  computations, so for all no more than  $2^c$  subsets, we need no more than  $(2c + 2) \cdot 2^c$  computations. For a fixed  $c$ , this value does not depend on  $n$ ; in other words, for each zone, we need  $O(1)$  steps.

To perform this computation for all  $2n + 1$  zones, we need  $(2n + 1) \cdot O(1) = O(n)$  steps.

6°. Finally, we find the largest of the resulting values  $V$  – this will be the desired value  $\bar{V}$ .

Finding the largest of  $O(n)$  values requires  $O(n)$  steps.

Overall, we need

$$O(n \cdot \log(n)) + O(n \cdot \log(n)) + O(n) + O(n) + O(n \cdot \log(n)) + O(n) = O(n \cdot \log(n))$$

steps. Thus, we have proven that our algorithm computes  $\bar{V}$  in  $O(n \cdot \log(n))$  steps.

## Acknowledgments.

This work was supported in part by the Army Research Laboratories grant DATM-05-02-C-0046. The author is thankful to Vladik Kreinovich for his help with editing the text, and to the anonymous referees for valuable suggestions.

## References

- [1] F. A. Cowell, “Grouping Bounds for Inequality Measures under Alternative Informational Assumptions,” *Journal of Econometrics*, 1991, Vol. 48, pp. 1–14.
- [2] Th. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 2001.
- [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 Verification”, In: R. Alt, A. Frommer, R. B. Kearfott, and W. Luther (eds.), *Numerical Software with Result Verification*, Springer Lectures Notes in Computer Science, 2004, Vol. 2991, pp. 274–305.
- [7] 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.
- [8] 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.
- [9] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis: With Examples in Parameter and State Estimation, Robust Control and Robotics*, Springer, London, 2001.
- [10] R. B. Kearfott, *Rigorous Global Search: Continuous Problems*, Kluwer, Dordrecht, 1996.
- [11] R. B. Kearfott and V. Kreinovich (eds.), *Applications of Interval Computations*, Kluwer, Dordrecht (1996)
- [12] R. E. Moore, *Interval Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1966.
- [13] R. E. Moore, *Methods and Applications of Interval Analysis*, SIAM, Philadelphia, 1979.
- [14] S. Rabinovich, *Measurement Errors: Theory and Practice*, American Institute of Physics, New York, 1993.
- [15] S. A. Starks, V. Kreinovich, L. Longpré, 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.
- [16] 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.