

Efficient evaluation of the sample variance of an interval-valued dataset

Michal Černý

Department of Econometrics, University of Economics Prague, Winston Churchill Square 4,
13067 Prague, Czech Republic

E-mail: cernym@vse.cz

Abstract. Given a set of interval-valued data, a general problem is to compute bounds for a particular statistic, such as sample mean or variance, variation coefficient or entropy. It is well known that computation of the upper bound of sample variance is an NP-hard problem. Here we consider a variant of an algorithm by Ferson et al., which is exponential in the worst case, and investigate its behavior under a natural probabilistic model. A simulation study shows that the undesirable case, which forces the algorithm to work in exponential time (and which appears in the proof of NP-hardness), occurs very rarely in an environment when the interval data are generated by probabilistic processes which are natural from a statistical viewpoint. The main finding is that the algorithm is practically very efficient and that the NP-hardness result usually “does not matter too much”.

1. Introduction

Let x_1, \dots, x_n denote a set of one-dimensional data. Often we face the problem that the dataset cannot be observed exactly. What we observe instead of x_1, \dots, x_n is a family of intervals $[\underline{x}_i, \bar{x}_i]$, $i = 1, \dots, n$, such that it is guaranteed that

$$\underline{x}_i \leq x_i \leq \bar{x}_i, \quad i = 1, \dots, n. \quad (1)$$

We often meet interval datasets in practice. Example include:

- A real-valued data point x is affected by rounding to the nearest integer \tilde{x} ; then we cannot observe the true value x , but only the interval $[\tilde{x} - \frac{1}{2}, \tilde{x} + \frac{1}{2}] \ni x$. The same problem occurs in scientific computing whenever we represent rational numbers by data types of fixed size.
- Another situation where we need to handle datasets of the form (1) is when data suffer from imprecision. Measurement devices sometimes guarantee the measurement precision in the form $\tilde{x} \pm \Delta$; that is, when the device reports the value \tilde{x} , then we only know that the true value x lies in the interval $[\tilde{x} - \Delta, \tilde{x} + \Delta]$.
- Further examples are encountered in econometrics, where we often need to work with interval predictions of future values of economic quantities (such as inflation or interest rates) or expert estimates, which are also often of interval nature.
- Another example is when we observe only daily mins/maxs of a continuous random process (which is the case of financial data). Let $x(t)$ be a trajectory of a random process with time $t \geq 0$, representing the price of an asset, say. We are often reported only daily minimum \underline{x} and maximum \bar{x} . Then we only know that the value $x(t)$ lies in $[\underline{x}, \bar{x}]$.



- In general, data subject to categorization, censoring or dicrestization can be usually interpreted as intervals of possible values.
- Another example is natural in physics or chemistry: “constants” are rarely constant, e.g. the gravity acceleration constant often should be treated as an interval of possible values (since the true value is not constant, depending on the position on the Earth).

2. The main question

We are interested in computation of a particular characteristic of the dataset x_1, \dots, x_n , such as sample mean or variance. But this cannot be done directly since we observe only the intervals satisfying (1). Let $S(x_1, \dots, x_n)$ be a statistic (or, generally, a continuous function). We need to replace S by another statistic $S^*(\underline{x}_1, \bar{x}_1, \dots, \underline{x}_n, \bar{x}_n)$ giving us similar information to S .

Example. Let x_1, \dots, x_n be sampled from $N(\mu, \sigma^2)$, where the parameters μ and σ^2 are unknown and are to be estimated. The assumption of a particular distribution allows us to write down the likelihood function and construct the max-likelihood estimator

$$(\hat{\mu}, \hat{\sigma}) = \arg \max_{\substack{\mu \in \mathbb{R} \\ \sigma \geq 0}} \prod_{i=1}^n \left[\Phi \left(\frac{\bar{x}_i - \mu}{\sigma} \right) - \Phi \left(\frac{\underline{x}_i - \mu}{\sigma} \right) \right],$$

where Φ stands for the cumulative distribution function of $N(0, 1)$. □

Another approach, complementary to the max-likelihood construction of the Example, is the possibilistic approach; for details see e.g. [1]. This approach does not rely on the assumption of a particular form of the underlying distribution. It simply considers an interval of *all possible* values of the statistic S , regardless of the underlying distribution (which is often unknown). So, the possibilistic version of the statistic S is the interval $[\underline{S}, \bar{S}]$, where

$$\bar{S} = \max\{S(x_1, \dots, x_n) : \underline{x}_i \leq x_i \leq \bar{x}_i, i = 1, \dots, n\}, \quad (2)$$

$$\underline{S} = \min\{S(x_1, \dots, x_n) : \underline{x}_i \leq x_i \leq \bar{x}_i, i = 1, \dots, n\}. \quad (3)$$

Now the assumption (1) implies that $S \in [\underline{S}, \bar{S}]$. For example, when S is the sample mean $\hat{\mu} = \frac{1}{n} \sum_{i=1}^n x_i$, then we get the interval

$$[\underline{\hat{\mu}}, \bar{\hat{\mu}}] = \left[\frac{1}{n} \sum_{i=1}^n \underline{x}_i, \frac{1}{n} \sum_{i=1}^n \bar{x}_i \right].$$

In this text we will restrict ourselves to the case when S is the sample variance:

$$S(x_1, \dots, x_n) = \frac{1}{n-1} \sum_{i=1}^n \left(x_i - \frac{1}{n} \sum_{j=1}^n x_j \right)^2.$$

It is easily seen that computation of \underline{S} is easy, since the optimization problem (3) reduces to convex quadratic minimization, which is an efficiently solvable problem (see e.g. Vavasis [7]). But now we arrive at the main problem, which has been proved in [2], [4]:

Theorem 1. *For every fixed $\delta \geq 0$ it is NP-hard to compute a value Σ such that $|\bar{S} - \Sigma| \leq \delta$.* □

Theorem 1 tells us that the value \bar{S} cannot be computed efficiently, even if we do not insist on an exact value but we are allowed to compute it only approximately with a prescribed absolute error δ . So we can expect only algorithms working in time $\approx 2^n$ or worse. The good news is

that we can indeed do it in time 2^n . The geometry of the optimization problem (2) shows that we are to maximize a convex function over the box $[\underline{x}_1, \bar{x}_1] \times \cdots \times [\underline{x}_n, \bar{x}_n]$. It follows that a maximizer is in a vertex. Thus the 2^n algorithm works as follows:

$$\bar{S} = \max\{S(x_1, \dots, x_n) : x_i \in \{\underline{x}_i, \bar{x}_i\}, i = 1, \dots, n\}. \quad (4)$$

However, when n is larger than 50 (say), the approach (4) is useless. So the main question is: what to do then?

3. A practically useful approach

The situation with computation of $\bar{S} = \max\{\frac{1}{n-1} \sum_{i=1}^n (x_i - \frac{1}{n} \sum_{j=1}^n x_j)^2 : \underline{x}_i \leq x_i \leq \bar{x}_i, i = 1, \dots, n\}$ is not as bad as it might look from Theorem 1.

For an interval $\mathbf{x} = [\underline{x}_i, \bar{x}_i]$ we define its center point and radius

$$x^C = \frac{1}{2}(\bar{x} + \underline{x}), \quad x^\Delta = \frac{1}{2}(\bar{x} - \underline{x}).$$

and denote $\mathbf{x} = [x^C \pm x^\Delta]$.

Now the interval dataset $[\underline{x}_1, \bar{x}_1], \dots, [\underline{x}_n, \bar{x}_n]$, can be denoted as $\mathbf{x}_1 = [x_1^C \pm x_1^\Delta], \dots, \mathbf{x}_n = [x_n^C \pm x_n^\Delta]$.

Ferson et al. [4] proved the following result:

Theorem 2. *Let $k \in \{2, \dots, n\}$ be a number such that for every $I \subseteq \{1, \dots, n\}$ such that $|I| = k$ we have*

$$\bigcap_{i \in I} [x_i^C \pm \frac{1}{n} x_i^\Delta] = \emptyset. \quad (5)$$

Then \bar{S} can be computed in time $O(n^2 2^k)$. □

The computation time $O(n^2 2^k)$ can be expected to be much better than 2^n in (4), since for datasets appearing in practice it will often be $k \ll n$. However, there are “extremal” examples: when all intervals $\mathbf{x}_1, \dots, \mathbf{x}_n$ share a common point (which can happen, for example, when $x_1^C = x_2^C = \dots = x_n^C$), then the condition (5) is satisfied with no k .

So, when analyzing a particular dataset, a good approach is to find

$$k^* = \min \left\{ k : \bigcap_{i \in I} [x_i^C \pm \frac{1}{n} x_i^\Delta] = \emptyset, I \subseteq \{1, \dots, n\}, |I| = k \right\}$$

and assess whether the computation time $n^2 2^{k^*}$ is acceptable; often it will be the case.

Remark. Observe that computation of k^* is an easily solvable problem: it suffices to sort the $(2n)$ -tuple of numbers $a_1 := x_1^C - \frac{1}{n} x^\Delta, a_2 := x_1^C + \frac{1}{n} x^\Delta, \dots, a_{2n-1} := x_n^C - \frac{1}{n} x^\Delta, a_{2n} := x_n^C + \frac{1}{n} x^\Delta$ into the form $a_{\pi(1)} \leq a_{\pi(2)} \leq \dots \leq a_{\pi(2n)}$, where π is a permutation, and for each interval $[a_{\pi(i)}, a_{\pi(i+1)}]$ to check in how many intervals $[x_i^C \pm \frac{1}{n} x_i^\Delta]$ it belongs.

The crucial parameter affecting computational complexity is k^* . The main message of this paper is: though the algorithm of Theorem 2 is exponential in the worst case, in practice we usually find out that k^* is reasonably low, so that the factor 2^{k^*} “does not matter too much”. This is what we will do in the next section.

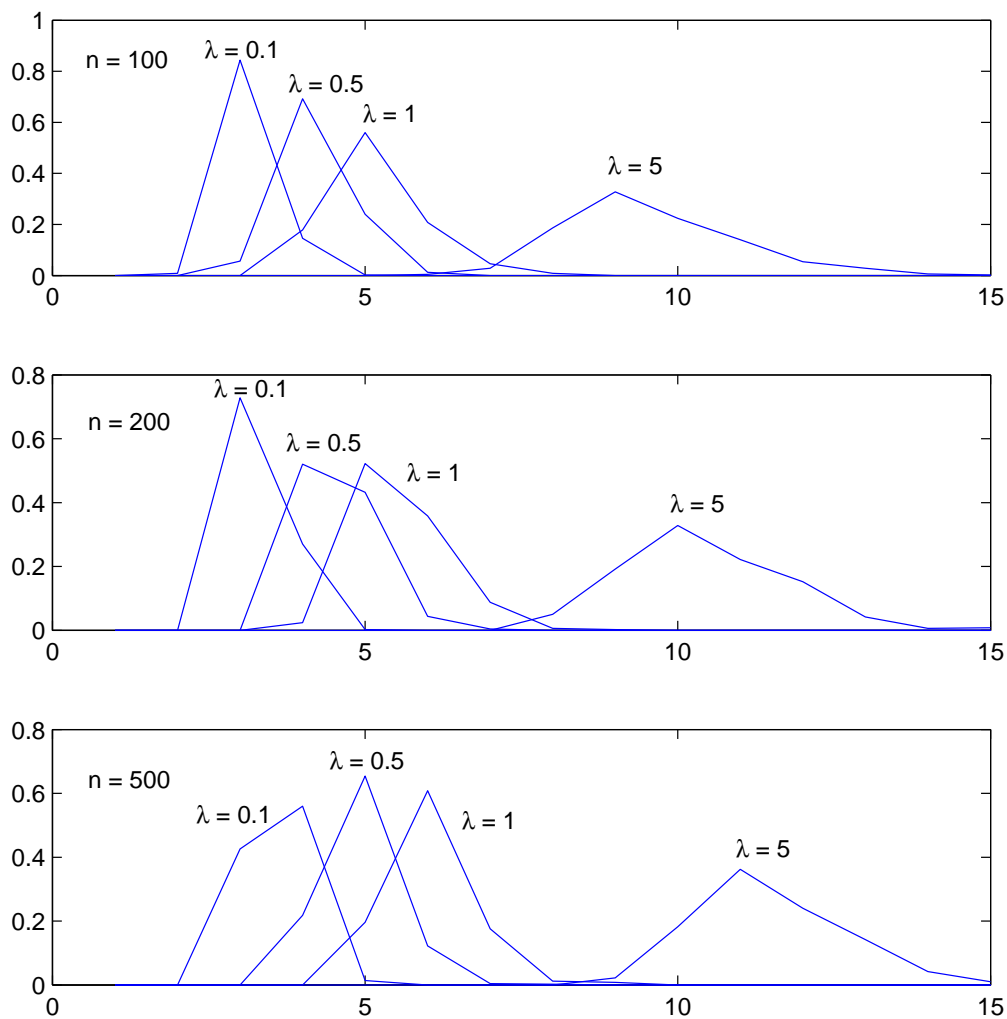


Figure 1. Distribution of k^* for $n \in \{100, 200, 500\}$ and $\lambda \in \{0.1, 0.5, 1, 5\}$.

4. A simulation

We will illustrate how k^* behaves in a quite natural model: we select

$$x_i^C \sim N(0, \sigma^2), \quad x_i^\Delta \sim \text{Exp}(\kappa), \quad i = 1, \dots, n. \quad (6)$$

It seems reasonable to measure k^* as a function of n and

$$\lambda = \frac{\text{var}(x_i^\Delta)}{\text{var}(x_i^C)} = \frac{\kappa}{\sigma}.$$

Remark. Though we select the data generating process in the particular form (6), further simulations (not presented here) show similar behavior also for other distributions with bounded variance.

The simulated distributions of k^* for $n = 100, 200, 500$ and $\lambda = 0.1, 0.5, 1, 5$ are depicted in Figure 1. It is not surprising that the average value of k^* grows with both n and λ . But the

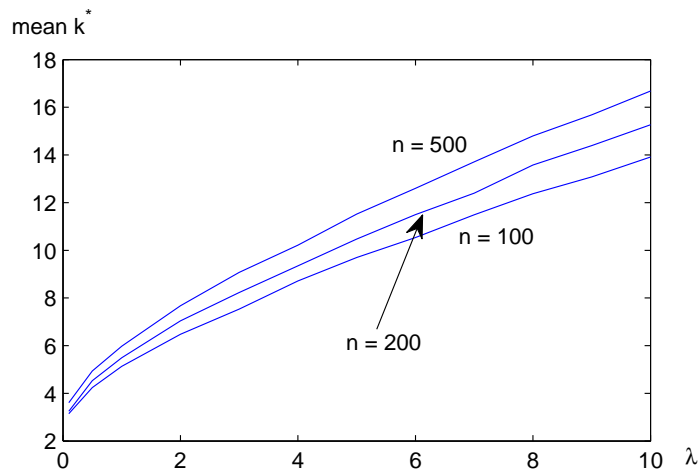


Figure 2. Estimated mean value of k^* for $n \in \{100, 200, 500\}$ and $\lambda \in [0.1, 10]$.

main message is that in the selected setup we encounter values of k^* at most 13 (say), and $2^{13} = 8192$, which is still an acceptable computation time.

We also plot the (simulated) mean value k^* as a function of λ in Figure 2.

5. Further results

As far the author is aware, the algorithm of Theorem 2 is the best practically useful tool for computation of \bar{S} . However, there are two more results which are complementary: the algorithm by Dantsin et al. [3], which computes \bar{S} in polynomial time under the condition that no interval $[x_i^C \pm \frac{1}{n}x_i^A]$ is a proper subinterval of another interval $[x_j^C \pm \frac{1}{n}x_j^A]$. Many datasets appearing in practice fulfill this condition.

And finally we should mention the pseudopolynomial algorithm of Černý and Hladík [2], which is applicable under the condition that all numbers $\underline{x}_1, \bar{x}_1, \dots, \underline{x}_n, \bar{x}_n$ are integers. Then, the pseudopolynomial algorithm works in time $O(n^3M^3)$, where

$$M = \max\{|\underline{x}_1|, |\bar{x}_1|, \dots, |\underline{x}_n|, |\bar{x}_n|\}.$$

So, this algorithm is polynomial when M can be bounded by a polynomial in n . From a practical viewpoint, the pseudopolynomial algorithm is fast when the dataset does not contain excessively large numbers.

6. Conclusions

Though the problem of computation of \bar{S} is NP-hard and inapproximable in general, we have seen that with a very natural probabilistic data-generating model (6) we almost do not encounter a hard instance which would require an excessively high computation time. This is good news for practice since the problem of computation of \bar{S} seems to be practically feasible, at least in many cases. The most tempting question for further research is: given a probabilistic data-generating model, how to estimate the probability that k^* attains a high value? This question makes a bridge to graph theory. It can be reformulated as follows. Let \mathbf{x}_i , $i = 1, \dots, n$, be random intervals. Consider the interval graph G determined by the dataset; that is, let \mathbf{x}_i , $i = 1, \dots, n$, be the vertices, and \mathbf{x}_i and \mathbf{x}_j ($i \neq j$) are connected with an edge iff $\mathbf{x}_i \cap \mathbf{x}_j \neq \emptyset$. How could we bound the probability that G has a large clique (under some reasonable assumptions, say that the random centers and random radii of the intervals are iid with finite variance)? If an estimate would show that the probability is reasonably small, we would have a proof that the

algorithm of Theorem 2 works “fast” on average, even though in the worst case it still can be exponential.

Acknowledgment

The work was supported by the Czech Science Foundation under Project No. P402/12/G097.

References

- [1] Černý M, Antoch J and Hladík M 2013 *Inform. Sci.* **244** 23–47
- [2] Černý M and Hladík M 2014 *Comp. Stat. Data Anal.* **80** 26–43
- [3] Dantsin E, Kreinovich V, Wolper A and Xiang G 2006 *Reliab. Comput.* **12** 273–80
- [4] Ferson S, Ginzburg L, Kreinovich V, Longpré L and Aviles M 2005 *Reliab. Comput.* **11** 207–33
- [5] Horowitz J L, Manski C F, Ponomareva C F and Stoye J 2003 *Reliab. Comput.* **9** 419–40
- [6] Kreinovich V, Longpré L, Patangay P, Ferson S and Ginzburg L 2005 *Reliab. Comput.* **11** 59–76
- [7] Vavasis S A 1991 *Nonlinear Optimization: Complexity Issues* (Oxford: Oxford University Press)