Choosing Sample Sizes for Statistical Measures on Interval-Valued Data

Josie McCulloch†‡, Zack Ellerby‡, Christian Wagner‡
‡ LUCID Research Group, Computer Science, University of Nottingham, UK
†School of Geography, University of Leeds, UK
j.mcculloch@leeds.ac.uk, zack.ellerby; christian.wagner@nottingham.ac.uk

***Abstract*—Intervals have frequently been used in the literature to represent uncertainty in data, from eliciting uncertain judgements from experts to representing uncertainty in sensor measurements. This widespread use of intervals has led to research on interval statistics to help understand the data. However, even seemingly trivial statistics (such as variance) cannot be calculated on interval-valued data using the same approach as for point data without incurring substantial loss of precision to a level which can make results close to useless. This loss of precision makes it challenging for decision makers to appropriately interpret interval-valued data using familiar statistics. Although there exist several approaches to computing statistics such as variance, these are all developed for specific properties of the data, and there is no general-case method. In addition, there are many statistical measures for which no efficient and accurate method exist. For such cases, we can use a Monte Carlo sampling approach to generate approximate statistics. While sampling does not generally produce exact solutions, it can provide a useful and efficient approximation to a desired degree of accuracy given sufficient computational resources. In this paper, we focus on the application of Monte Carlo sampling to generate statistics for interval-valued data. Specifically, we explore the optimum sample size required to calculate statistics on interval-valued data for a given degree of accuracy desired. We compare different sizes of data and different sampling methods to demonstrate how these affect the choice of an optimum sample size.**

I. Introduction

Intervals have frequently been used in the literature to represent uncertainty in data. For example, when eliciting expert judgements, an expert may find it difficult to give a precise response but find it easier or more appropriate to give a range. Intervals have been used most commonly to represent uncertainty in engineering [1]. No sensor is perfect and all sensors are limited in their precision. For example, consider a sonar sensor that has an error of 0.1cm. If the sensor measures an object 10cm away, the correct distance of the object is not 10cm with absolute certainty, but is in fact somewhere in range $[9.9, 10.1]$cm. This interval is typically treated as a uniform distribution as we cannot be certain that the centre is any more likely to be correct than any other value.

The widespread use of intervals for representing uncertainty has led to research on statistics for interval-valued data to better understand said data. As the data are intervals, any statistic is also represented as an interval to reflect the uncertainty of the statistic; for example, the variance of interval-valued data will be an interval $V = [\underline{V}, \overline{V}]$. However, it has been found that if methods for calculating statistics on point data are directly applied to interval-valued data, the result often has excessive width [2]; that is, the interval-valued statistic contains the correct answer but is excessively wide, providing a poor quality estimate of the actual uncertainty. For example, given data $X = \{[57.0, 63.0], [37.0, 43.0]\}$, the true variance of X is $[98.0, 338.0]$; note that the exact variance can be calculated for a small set of data using a brute force approach [3]. If the formula for variance on point data is applied directly to the intervals, the result is $[32.0, 512.0]$. While the former accurate result is a subset of the interval arithmetic result, the latter is so wide that in practice it is effectively useless.

Excessive width is a common problem in interval computations. Using interval arithmetic, any operation on two intervals works on the assumption that the intervals are independent. However, while sometimes it may be true that they are independent, this is not always the case. For example, consider two intervals $x = [0, 4]$ and $y = [0, 4]$. If x and y are independent, $x - y = [-4, 4]$ (we subtract every value in y from every value in x). If we calculate $x - x$ then, intuitively, the result should be 0 because it is an operation on identical variables. However, we generally do not know if the intervals operated on are dependent or otherwise, so we always treat $x - x$ as if it were $x - y$. As a result of this, when intervals are related, the result of an operation may be too wide [2]. This excessive width in the statistical result is undesirable because decision makers may be reluctant to make a decision when the analysis of the data contains considerably more uncertainty than if a more accurate method was used. Therefore, it is necessary to find a more accurate method than a straightforward interval arithmetic approach.

It is possible to compute the exact statistic for a small number of intervals, but the computational complexity of this calculation increases exponentially as the number of intervals increases, and quickly becoming infeasible [4], [5]. In recent years, efficient methods have been developed to accurately calculate a variety of interval statistics [6], including variance [7] and covariance [8]. However, accurate and efficient methods do not as of yet exist for all interval statistics, as the research area is still growing. For example, while many algorithms have been developed to compute the upper bound of the variance of intervals ($\overline{V}$), there is no general method for solving $\overline{V}$ for all cases. Specifically, efficient methods are developed based

This work was supported by EPSRCs EP/P011918/1 grant.

on the properties of the intervals. Such properties include if the intervals are all narrow [9], if there is no partial overlap (i.e., any two intervals are either disjoint or identical) [10], [11], or there is only a small degree of overlap where *small* is predefined [9]. A different method is required to efficiently compute $\overline{V}$ in each of these cases, and there is no common method to satisfy all of these criteria.

A Monte Carlo approach could be applicable to intervals with differing properties, i.e. disjoint vs overlapping, of identical vs varying widths. Therefore, a Monte Carlo approach may be promising as a general approach to interval statistics where an efficient method does not yet exist [12]. One potential method for using a Monte Carlo approach on interval-valued data involves reducing the intervals to random samples of point data. For example, Fig. 1 shows two intervals and a random sample of points within those intervals. A statistical test can be used on these points without needing to rely on interval arithmetic.

When using samples of points to analyse interval data, it is important that a sufficiently large sample size is used. If too few point samples are used, then these samples are less likely to find a good approximation of the correct result. However, if enough samples are used, a useful approximation of the interval statistic can be obtained. As more samples are taken, the accuracy of the result will increase, but so will the computational time to calculate the result. If the sample size is too large, the calculation may take an unreasonable amount of time for only a small gain in result accuracy.

We wish to find the *optimum* sample size, at which sufficient accuracy in the result is achieved without taking excessive computational time. It is expected that the optimum sample size will increase exponentially with the total number of intervals. That is, if a given problem needs n samples to achieve a certain level of accuracy for a single variable, a total of n^k samples would be needed to achieve the same sample density for k variables [13].

This paper aims to assess how a Monte Carlo approach can be used to calculate descriptive statistics on intervals. We do so by investigating the optimum sample size and sampling method required for calculating the variance of different sets of intervals. We explore how the most favourable sample size is affected by the total number of intervals in the data and the method of sampling used. We aim to find the smallest sample size that gives a good accurate result, such that increasing the sample size is unsuitable when considering both the small increase in accuracy and large increase in computational time that it brings. Using these results, we aim to demonstrate how a Monte Carlo approach can be effectively applied to calculate interval statistics for data where a specific-case algorithm does not exist.

II. Methods

To inform the optimum sample size for a Monte Carlo approach to interval statistics, we focus on the problem of calculating variance. Multiple methods for calculating variance on interval-valued data have been published, which take into account different properties of the data, such as narrow or wide intervals and overlapping or disjoint intervals [2], [9], [10], [11]. By focusing on the problem of measuring variance, we have a ground truth result that our Monte Carlo approximations can be compared against. Specifically, we can observe how close the Monte Carlo approximations reach the correct result for different sample sizes. These results can then be used to help choose an appropriate sample size for calculating an interval statistic for which an efficient, exact method is not known. This is important as there is no general method to solving the upper-bound of a statistic on interval-valued data in a feasible amount of time [2].

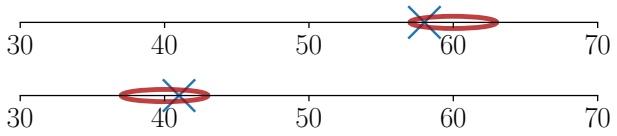


Fig. 1. An example of two intervals and a sample of points within the intervals

Let X be a set of k unordered intervals $X = \{x_1, ..., x_k\}$ where $x_i = [\underline{x_i}, \overline{x_i}]$. In Monte Carlo sampling, for each sample, we reduce X from a set of intervals to a set of points $X' = \{x'_1, ..., x'_k\}$ where x'_i is a pseudo-randomly chosen value satisfying the constraint $\underline{x_i} \leq x'_i \leq \overline{x_i}$. The variance of X' will be a point value.

Each time we generate a new sample of points X', the variance for this given sample will be different. We calculate the variance for a total of n different samples, resulting in n different measurements of variance $\{v_1, ..., v_n\}$ in respect to the original intervals. We can then obtain an interval-valued variance on X using the smallest and largest variances across the samples; i.e. $V = [\underline{V}, \overline{V}] = [\min(v_1, ..., v_n), \max(v_1, ..., v_n)]$.

In this paper, we test how the result V differs for different values of n. We start with a small sample $n = 10$ and calculate the result V. As the sample X is pseudo-randomly chosen, different runs of the test where $n = 10$ will produce different results. We therefore run 100 tests, keeping the number of samples n constant, to observe how much the result may differ for a single sample size. We note the minimum, mean and maximum result across the 100 tests for the given sample size.

For a given set of data, we test a large range of sample sizes, where in each new experiment the value of n is double the previous value; i.e. $n \in \{10, 20, ..., 655360, 1310720\}$. We consider two different methods of generating the sample point data. The first method is to use pseudo-random numbers with which no effort is made to ensure an even distribution of samples in the design. We achieve this using the *rand* function in Octave. In the second method, we use maximin Latin Hypercube Sampling (maximin-LHS) to ensure an even distribution of point values are sampled across the data. We achieve this using the *stk_sampling_maximinLHS* function in the STK toolbox for Octave. The maximin-LHS method optimises the sample space by choosing a design (i.e. a selection of samples) that maximises the distances between all possible pairs, whilst minimising the number of pairs separated

by a given distance. It is generally expected that the maximin-LHS design is more likely to find a better approximation with fewer samples than non-optimised sampling. However, an LHS-approach will also be more computationally expensive.

In the next section, we analyse the effects of sample size and the two different methods on three different synthetic data sets containing a total of 2, 3 and 10 intervals.

III. Results

In this section, we demonstrate variance measured on three different examples of data using different sample sizes and sampling methods. To maintain consistency throughout the examples, the mean of the interval midpoints in each of the three example sets is 50, and the width of each interval is 6. Therefore, the interval mean of each example is $[47.0, 53.0]$. In addition, the variance of the interval midpoints in each example is kept constant at 10. Note, however, that while the variance of the midpoints is constant across the examples, the variance of the full intervals differs as the total number of intervals in each example differs.

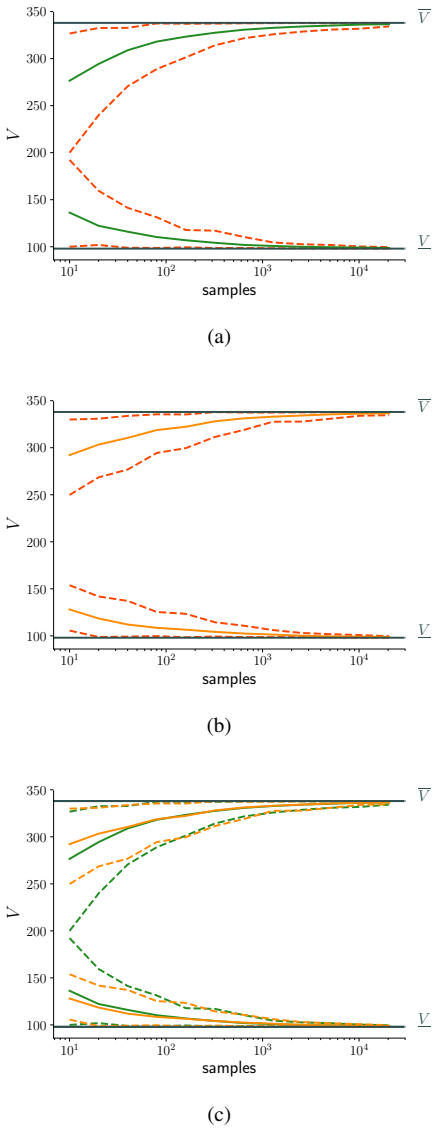
Example 1 (Two intervals): Let X be a set of two intervals as follows:

$$X = \{[57.0, 63.0], [37.0, 43.0]\}$$

The exact variance of X is $[98.0, 338.0]$. Fig. 2 shows the approximated results of the lower-bound and upper-bound of the variance of X using different sample sizes and the two different methods of generating random numbers. Each sample size is tested 100 times. The average result is shown with a solid line, and the minimum and maximum results are shown with red dashed lines. The exact lower and upper bounds of variance are shown with a black solid horizontal line. Note that while we tested up to a sample size of $n = 1310720$, the figure only shows results up to $n = 20480$ because at higher samples there is no visual difference in the results at this resolution. The results show that as the sample size increases, the approximated result using the Monte Carlo method gets closer to the exact result.

It is generally expected that the optimised sample points (using maximin-LHS) should perform better than non-optimised samples (using rand) for small sample sizes. Fig. 2(c) shows the results of both sampling methods together. We can see that maximin-LHS appears to perform better than rand at small sample sizes; that is, it gets closer to the exact variance. Comparing the results from the 100 runs of the two methods for each sample size using the Mann-Whitney U test, we find there is only a significant difference when the sample size was 10. For higher sample sizes, there was no statistically significant difference in the estimated bounds of the variance between the two methods. As it is unlikely that a sample size of 10 will give a result with a desired degree of accuracy, these results suggest it is unnecessary to choose an optimised sample design over one that is not optimised.

To further analyse the difference between the sampling methods, we can measure how well the sample designs fill the sample space. We use the $\phi_p(S)$ criterion function [14],



(a)

(b)

(c)

Fig. 2. The lower-bound ($\underline{V}$) and upper-bound ($\overline{V}$) of the variance calculated using (a) *rand*, (b) *rand_maximinlhs* and (c) both across different sample sizes for example 1 (two intervals).

[15] to rank a sampling design S. The smaller the value of ϕ_p the better the space-filling properties of S. This is given as

$$\phi_p(S) = \Big(\sum_{1 \le i < j \le m} d_{ij}^{-p} \Big)^{1/p} \tag{1}$$

where d_{ij} is the distance between sample points i and j in the sorted design S. In this paper, we use $p = 50$.

In Fig. 3, we show ϕ_p for the two methods of sampling where the sample size is 10, 160 and 5120. We compare ϕ_p against the error in variance (as a percentage of how far the estimated variance is from the actual variance). Note that the errors for $\underline{V}$ and $\overline{V}$ are shown together in the figure. For each sample size, maximin-LHS produced a better selection of samples than rand according to ϕ_p. However, there is no

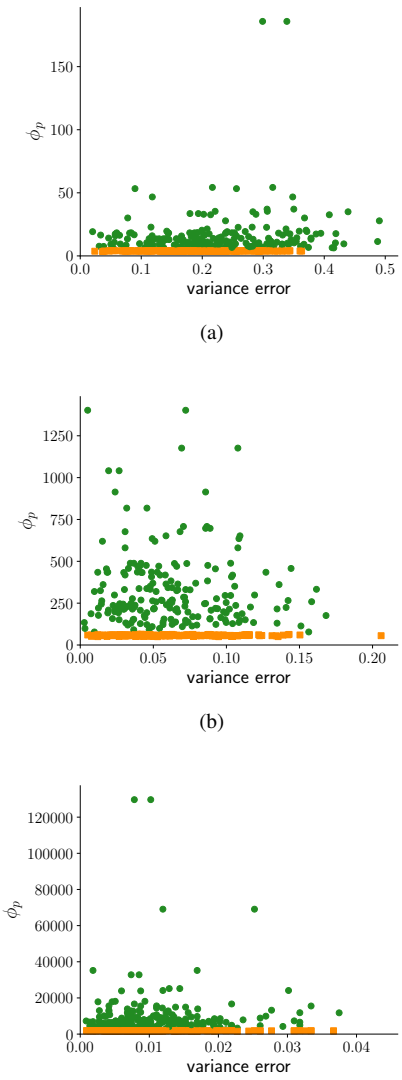


Fig. 3. The error in variance and the space filling criterion ϕ_p (see eq. 1) of each sample result for (a) 10 samples, (b) 160 samples and (c) 5120 samples for example 1 (two intervals) using rand (green circles) and maximin-LHS (orange squares).

clear correlation between a good sample design and a low error in variance. This may be because even a poor sample design can find a good approximation of variance by chance. It is also clear from the figure that using maximin-LHS gave a smaller maximum error compared to rand when the sample size was 10, but there is no clear difference in the maximum error between the two methods when the sample size is 160 or 5120. Specifically, there is no correlation between how well the sample design fills the space and the size of error in the approximated variance.

Note that for both methods, as the sample size increases ϕ_p increases. This is because we calculate the sum of distances between sample points and, therefore, as more sample points are used the sum of their distances must increase. This is why the scales of ϕ_p are different within each plot in Fig. 3.

We aim to find the smallest sample size that maximises result accuracy and minimises computational time. We first investigate if the accuracy of the results significantly increases as we double the sample size. We want to know if results from a given sample size get significantly closer to the exact result when the sample size is doubled. We therefore compare consecutive sample sizes within the list $n = \{10, 20, ..., 655360, 1310720\}$ to test, for example, if using 20 samples provides a significant improvement in result accuracy compared to 10 samples, and if 1,310,720 samples provides an improvement over 655,360 samples. We test this using the Mann-Whitney U test. We find that in all tests, the results significantly improved when the sample size was doubled. However, subjectively, the improvement in accuracy may be considered *small* at high sample sizes. For example, using rand, when the sample size was at least 163,840 we obtained an error of no more than 0.002% in $\underline{V}$ and no more than 0.01% in $\overline{V}$.

We next demonstrate the effect of increasing the number of intervals.

Example 2 (Three intervals): Let X be a set of three intervals as follows:

$$X = \{[60.0, 66.0], [36.0, 42.0], [44.0, 50.0]\}$$

The exact variance of these intervals is $[81.333, 241.335]$. Fig. 4 shows the results of the lower-bound and upper-bound of the variance of X using different samples sizes and the two different methods of generating random numbers. As with example 1, we find that as the number of samples increases, the approximated results approach the exact results. Comparing the 100 test results of a given sample size n against the results for $2n$, we find a statistically significant improvement for all n up to 1,310,720. Using rand at a sample size of 655,360, we obtained an error of no more than 0.01% in $\underline{V}$ and $\overline{V}$. At lower sample sizes, the error was larger. Compared to the previous example, we have needed to use more samples to achieve the same degree of accuracy. In the previous example, a sample size 163,840 was sufficient to obtain an error no higher than 0.01%.

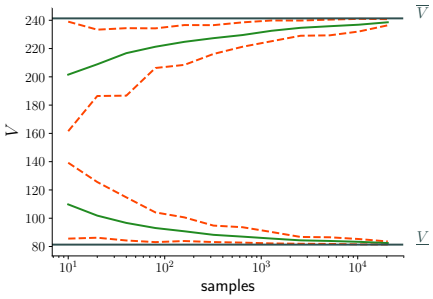
While there is some visual difference between the two sampling methods for small sample sizes (see Fig. 4(c)) this is not statistically significant.

Example 3 (Ten intervals): Let X be a set of ten intervals as follows:

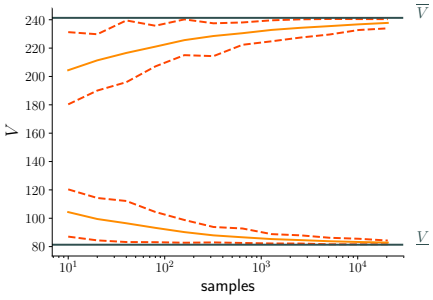
$$\begin{aligned} X = \{&[47.0, 53.0], [58.0, 64.0], [47.0, 53.0], [55.0, 61.0], \\ &[48.0, 54.0], [20.0, 26.0], [49.0, 55.0], [48.0, 54.0], \\ &[43.0, 49.0], [56.0, 62.0]\} \end{aligned}$$

The exact variance of X is $[77.0, 164.0]$.

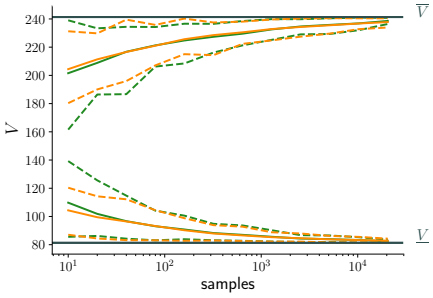
Fig. 5 shows the Monte Carlo results of the lower-bound and upper-bound of the variance of X using different samples sizes and the two different methods of generating random numbers. It is clear that the approximated results are much further from the exact results compared to examples 1 and 2 where there



(a)

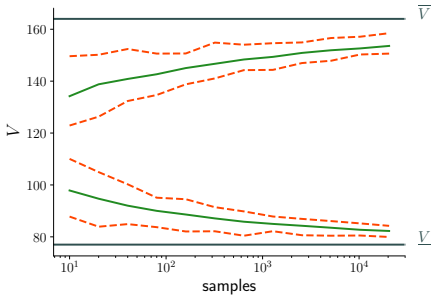


(b)

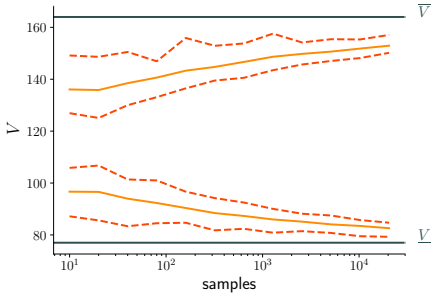


(c)

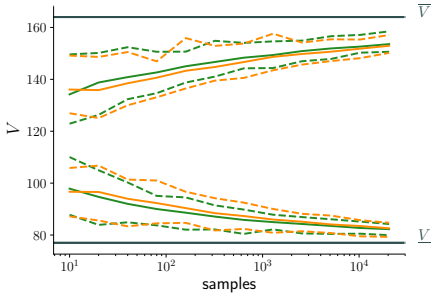
Fig. 4. The lower-bound ($\underline{V}$) and upper-bound ($\overline{V}$) of the variance calculated using (a) *rand*, (b) *rand_maximinlhs* and (c) both across different sample sizes for example 2 (three intervals).



(a)



(b)



(c)

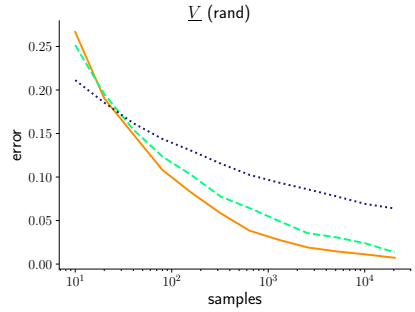
Fig. 5. The lower-bound ($\underline{V}$) and upper-bound ($\overline{V}$) of the variance calculated using (a) *rand*, (b) *rand_maximinlhs* and (c) both across different sample sizes for example 3 (10 intervals).

were fewer intervals. Also, looking at both sampling methods together in Fig. 5(c), there is no noticeable difference — both methods appear to perform equally poor. Note that a much higher sample size than that shown is required to obtain a result that is as accurate as achieved in examples 1 and 2.

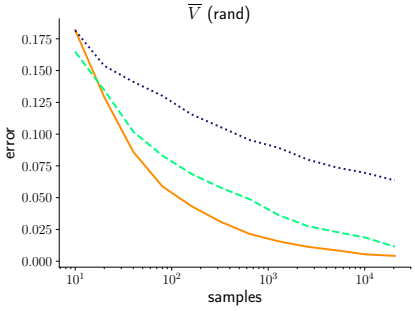
Next, we compare the results across the three examples. Fig. 6 shows how close the approximated variance reaches the actual variance (as a relative percentage of error) across each of the three examples. It is clear that, for a given sample size, a smaller error is achieved when measuring fewer intervals. It is therefore also clear that to achieve a given degree of accuracy, the sample size must significantly increase as the number of intervals increases.

Next, we consider the computation time to calculate variance for a given sample size. It is obvious that as we increase the sample size, the computational time will also increase. How much it increases will depend on the method of sampling used. If arbitrary pseudo-random samples are used, the increase will be linear. If a method that ensures an even sampling of the space is taken (e.g. LHS methods) the computation time is expected to grow as the number of samples increases — the more samples there are the longer it will take to ensure they cover the sample space effectively.

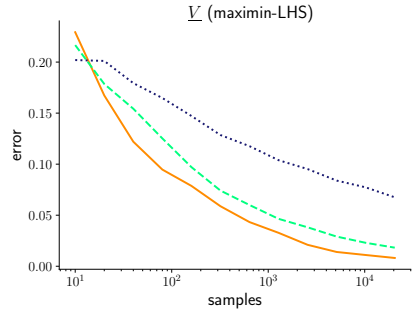
Fig. 7 shows the average time in seconds taken to calculate variance for a given sample size for data containing two, three and ten intervals (from examples 1, 2 and 3), using the two methods of generating pseudo-random numbers (rand and maximin-LHS). Using rand, as the sample size



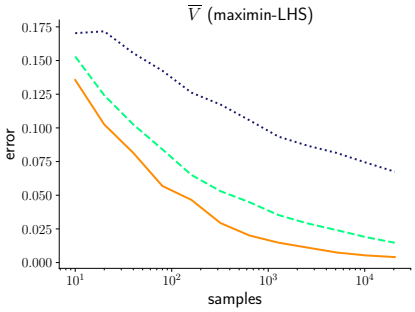
(a)



(b)

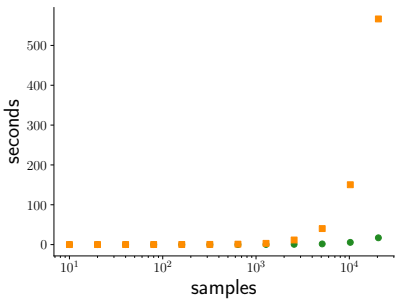


(c)

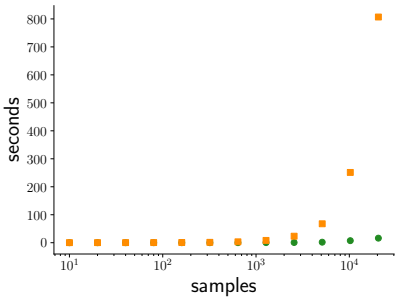


(d)

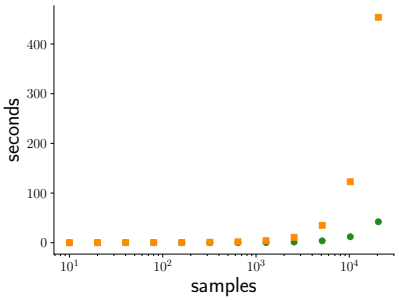
Fig. 6. The relative error in variance approximation for examples 1 (solid), 2 (dashed) and 3 (dotted) for (a) $\underline{V}$ using rand; (b) $\overline{V}$ using rand; (c) $\underline{V}$ using maximin-LHS; (d) $\overline{V}$ using maximin-LHS.



(a)



(b)



(c)

Fig. 7. The average time (in seconds) taken to calculate variance with a given sample size using the rand (green circles) and maximin-LHS (orange squares) methods for (a) example 1, (b) example 2 and (c) example 3.

doubles, the computational time also doubles, as expected (note, however, that it is much faster than maximin-LHS and therefore this does not show in the figure). Using maximin-LHS, the computational time to select samples is noticeably increased compared to rand, increasing exponentially as the sample size increases. As the number of intervals increases, the computational time also increases, but this has much less of an effect compared to increasing sample size.

IV. Conclusions

Using straightforward methods of calculating statistics on intervals leads to excessive width in the result. While it is possible to measure the statistic with perfect accuracy, the problem is NP-hard, and soon takes an infeasible amount of time to compute given enough intervals. There have been

developments in approximating some statistical measures on intervals that have specific properties, but there is no method for the general case, and efficient approximations do not exist for all statistical measures.

In this paper, we explore if Monte Carlo sampling can be used to approximate statistical measures on intervals. While we demonstrate using sampling to calculate variance, this technique can be used to calculate any statistic on intervals. We show that the sample size required increases, as expected, as the total number of intervals increases. We demonstrate that for two intervals approximately 10^4 samples are required for high accuracy, for three intervals at least 10^5 samples are required, and for 10 intervals much more than 10^6 samples (beyond what was tested) are required to achieve the same level of accuracy. The number of samples required for a given degree of accuracy increases exponentially with the number of intervals. This may take an infeasible amount of time to compute as the number of intervals increases. We also show that using a maximin-LHS sample design only provided improved results over using a non-optimised design for low sample sizes, at which accuracy was poor for both methods. We therefore envisage little practical benefit to this design.

These results demonstrate that a Monte Carlo method alone is not sufficient to calculate an accurate approximation of certain descriptive statistics—specifically the variance—on a large number of intervals ($k > 10$). There is a need for sampling to be combined with more complex optimisation methods, beyond optimising the design of the sample space.

REFERENCES

[1] G. Xiang, A. Pownuk, O. Kosheleva, and S. A. Starks, "Von Mises Failure Criterion in Mechanics of Materials: How to Efficiently Use it Under Interval and Fuzzy Uncertainty," in *NAFIPS 2007-2007 Annual Meeting of the North American Fuzzy Information Processing Society*. IEEE, 2007, pp. 570–575.

[2] H. T. Nguyen, V. Kreinovich, B. Wu, and G. Xiang, *Computing statistics under interval and fuzzy uncertainty*. Springer, 2012.

[3] S. Ferson, V. Kreinovich, J. Hajagos, W. Oberkampf, and L. Ginzburg, "Experimental uncertainty estimation and statistics for data having interval uncertainty," *Sandia National Laboratories, Report SAND2007-0939*, vol. 162, 2007.

[4] V. Kreinovich, A. V. Lakeyev, J. Rohn, and P. Kahl, *Computational complexity and feasibility of data processing and interval computations*. Springer Science & Business Media, 2013, vol. 10.

[5] S. A. Vavasis, *Nonlinear optimization: complexity issues*. Oxford University Press, Inc., 1991.

[6] V. Kreinovich, L. Longpré, S. A. Starks, G. Xiang, J. Beck, R. Kandathi, A. Nayak, S. Ferson, and J. Hajagos, "Interval versions of statistical techniques with applications to environmental analysis, bioinformatics, and privacy in statistical databases," *Journal of Computational and Applied Mathematics*, vol. 199, no. 2, pp. 418–423, 2007.

[7] V. Kreinovich, G. Xiang, S. A. Starks, L. Longpré, M. Ceberio, R. Araiza, J. Beck, R. Kandathi, A. Nayak, R. Torres *et al.*, "Towards combining probabilistic and interval uncertainty in engineering calculations: algorithms for computing statistics under interval uncertainty, and their computational complexity," *Reliable Computing*, vol. 12, no. 6, pp. 471–501, 2006.

[8] M. Kishida, "On computations of variance, covariance and correlation for interval data," *Mechanical Systems and Signal Processing*, vol. 84, pp. 462–468, 2017.

[9] S. Ferson, L. Ginzburg, V. Kreinovich, L. Longpré, and M. Aviles, "Computing variance for interval data is np-hard," *ACM SIGACT News*, vol. 33, no. 2, pp. 108–118, 2002.

[10] L. Longpré, G. Xiang, V. Kreinovich, and E. Freudenthal, "Interval approach to preserving privacy in statistical databases: Related challenges and algorithms of computational statistics," in *International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security*. Springer, 2007, pp. 346–361.

[11] G. Xiang and V. Kreinovich, "Estimating variance under interval and fuzzy uncertainty: case of hierarchical estimation," in *International Fuzzy Systems Association World Congress*. Springer, 2007, pp. 3–12.

[12] J. Helton, J. Johnson, W. Oberkampf, and C. B. Storlie, "A sampling-based computational strategy for the representation of epistemic uncertainty in model predictions with evidence theory," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 37-40, pp. 3980–3998, 2007.

[13] A. Forrester, A. Sobester, and A. Keane, *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008.

[14] M. D. Morris and T. J. Mitchell, "Exploratory designs for computational experiments," *Journal of statistical planning and inference*, vol. 43, no. 3, pp. 381–402, 1995.

[15] L. Pronzato and W. G. Müller, "Design of computer experiments: space filling and beyond," *Statistics and Computing*, vol. 22, no. 3, pp. 681–701, 2012.