Fitting a Normal Distribution to Interval and Fuzzy Data

Gang Xiang and Vladik Kreinovich
Department of Computer Science
University of Texas at El Paso
El Paso, TX 79968, USA
{gxiang,vladik}@utep.edu

Scott Ferson
Applied Biomathematics
100 North Country Road
Setauket, NY 11733, USA
scott@ramas.com

***Abstract*—In traditional statistical analysis, if we know that the distribution is normal, then the most popular way to estimate its mean a and standard deviation σ from the data sample $x_1, \ldots, x_n$ is to equate a and σ to the arithmetic mean and sample standard deviation of this sample. After this equation, we get the cumulative distribution function $F(x) = \Phi\left(\frac{x-a}{\sigma}\right)$ of the desired distribution.**

In many practical situations, we only know intervals $[\underline{x}_i, \overline{x}_i]$ that contain the actual (unknown) values of x_i or, more generally, a fuzzy number that describes x_i. Different values of x_i lead, in general, to different values of $F(x)$. In this paper, we show how to compute, for every x, the resulting interval $[\underline{F}(x), \overline{F}(x)]$ of possible values of $F(x)$ – or the corresponding fuzzy numbers.

I. Introduction

Formulation of the problem. In many real-life situations, the actual distribution is normal (Gaussian). It is known that a normal distribution is uniquely determined by its mean a and its standard deviation σ. Usually, a cumulative distribution function corresponding to the distribution (cdf) with 0 mean and standard deviation 1 is denoted by $\Phi(x)$. In terms of this function $\Phi(x)$, the cdf $F(x)$ of a general normal distribution has the form

$$F(x) = \Phi\left(\frac{x-a}{\sigma}\right). \quad (1)$$

To find the cdf, we must therefore estimate the (unknown) parameters a and σ from the (known) sample values $x_1, \ldots, x_n$. In traditional statistical data processing, one of the most widely used methods for estimating a and σ is the *method of moments*, when we find the mean and variance of the data, i.e., the values

$$a = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i, \quad \sigma^2 = \frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - a)^2 = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i^2 - a^2, \quad (2)$$

and consider the normal distribution with these values a and σ as "fitted" to the data $x_1, \ldots, x_n$; see, e.g., [12], [13].

Case of interval uncertainty. In practice, instead of the exact values x_i of the sample, we often only know the intervals $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$ of possible values of x_i. Different values of $x_i \in \mathbf{x}_i$ lead, in general, to different values of a and σ and thus, for every x, to different values of the cdf $F(x) = \Phi\left(\frac{x-a}{\sigma}\right)$. It is therefore desirable to find the interval $[\underline{F}(x), \overline{F}(x)]$ of possible values of the cdf, i.e., in terms of [3], to find a *p-box* that contains all possible cumulative distribution functions.

Case of fuzzy uncertainty. Often, knowledge comes in terms of uncertain expert estimates. In the fuzzy case, to describe this uncertainty, for each value of estimation error Δx_i, we describe the degree $\mu_i(\Delta x_i)$ to which this value is possible.

For each degree of certainty α, we can determine the set of values of Δx_i that are possible with at least this degree of certainty – the *α-cut* $\{x \mid \mu(x) \geq \alpha\}$ of the original fuzzy set. In most cases, this α-cut is an interval.

Vice versa, if we know α-cuts for every α, then, for each object x, we can determine the degree of possibility that x belongs to the original fuzzy set [1], [2], [5], [9], [10], [11]. A fuzzy set can be thus viewed as a nested family of its α-cuts.

A *fuzzy number* can be defined as a fuzzy set for which all α-cuts are intervals.

So, if instead of a (crisp) interval $\mathbf{x}_i$ of possible values of the measured quantity, we have a fuzzy number $\mu_i(x)$ of possible values, then we can view this information as a family of nested intervals $\mathbf{x}_i(\alpha)$ – α-cuts of the given fuzzy sets.

From the computational viewpoint, processing fuzzy uncertainty reduces to processing of interval uncertainty. We have already mentioned that if instead of a (crisp) interval $\mathbf{x}_i$ of possible values of the measured quantity, we have a fuzzy number $\mu_i(x)$ of possible values, then we can view this information as a family of nested intervals $\mathbf{x}_i(\alpha)$ – α-cuts of the given fuzzy sets.

Our objective is then to compute the fuzzy number corresponding to this the desired value $y = f(x_1, \ldots, x_n)$. In this case, for each level α, to compute the α-cut of this fuzzy number, we can apply interval computations to the α-cuts $\mathbf{x}_i(\alpha)$ of the corresponding fuzzy sets. The resulting nested intervals form the desired fuzzy set for y.

So, e.g., if we want to describe 10 different levels of uncertainty, then we must solve 10 interval computation problems. In other cases, we know *fuzzy numbers* which describe x_i. In such situations, it is desirable, for every x, to find the corresponding fuzzy number $\mathbf{F}(x)$.

Thus, from the computational viewpoint, it is sufficient to produce an efficient algorithm for the interval case.

Computing the fuzzy number can be reduced to computing, for different values α, the corresponding α-cut intervals based on the α-cuts of the fuzzy sets X_i.

What is known. Since the value of $F(x)$ is determined by the values of the mean a and of the standard deviation σ, it is reasonable to first analyze the intervals of possible values for a and for σ. For a, the interval of possible values is easy to describe: since the average is an increasing function of all its variables, its minimum is attained when all x_i takes their smallest values $\underline{x}_i$, and the maximum is attained when all its variables take their largest values $\overline{x}_i$; as a result, we get the interval $[\underline{a}, \overline{a}]$, where

$$\underline{a} = \frac{1}{n} \cdot \sum_{i=1}^{n} \underline{x}_i, \quad \overline{a} = \frac{1}{n} \cdot \sum_{i=1}^{n} \overline{x}_i. \tag{3}$$

For standard deviation, the problem of computing the corresponding interval $[\underline{\sigma}, \overline{\sigma}]$ is, in general, NP-hard. Crudely speaking, this means that unless it turns out that P=NP (which few computer scientists believe), every algorithm that computes this interval exactly in all cases requires time which grows at least exponentially with n. Actually, exponential time is sufficient: we can compute the upper endpoint $\overline{\sigma}$ if we consider all 2^n possible combinations of the values $\underline{x}_i$ and $\overline{x}_i$, i.e., all the corners of the n-dimensional box $[\underline{x}_1, \overline{x}_1] \times \ldots \times [\underline{x}_n, \overline{x}_n]$.

In some practically important cases, there exist efficient algorithms whose running time grows only polynomially with n. For example, such algorithms are possible in the "no-nesting" case when no two intervals $[\underline{x}_i, \overline{x}_i]$ and $[\underline{x}_j, \overline{x}_j]$ $(i \neq j)$ are proper subset of one another in the sense that $[\underline{x}_i, \overline{x}_i] \not\subseteq (\underline{x}_j, \overline{x}_j)$. For an overview of known results, see, e.g., [7], [8].

In principle, we can use the resulting bounds $[\underline{a}, \overline{a}]$ on a and $[\underline{\sigma}, \overline{\sigma}]$ on σ to produce bounds on the ratio $\frac{x-a}{\sigma}$ and thus, on the desired cumulative distribution function (cdf) $F(x)$. However, the values a and σ are dependent in the sense that not all combinations of $a \in [\underline{a}, \overline{a}]$ and $\sigma \in [\underline{\sigma}, \overline{\sigma}]$ are possible; as a result, these bounds contain excess width – a typical situation when computations with intervals ignore dependence (see, e.g., [4]).

How can we compute the exact bounds on $F(x)$? The closest to this are the algorithms from [6] which produce bounds for the absolute value $\frac{|x-a|}{\sigma}$ of the desired ratio.

What we plan to do. In this paper, we show how to compute the desired p-box, i.e., the exact bounds for the normal $F(x)$ under interval data.

II. Algorithms for Computing the p-Box in the General Case: Motivations and Description

Reducing the problem to computing the ratio. The above cdf $\Phi(x)$ of a "standard" normal distribution is a strictly increasing function. Thus, for every x, the interval $[\underline{F}(x), \overline{F}(x)]$ of possible values of the the quantity $F(x) = \Phi\left(\frac{x-a}{\sigma}\right)$ can be computed as $[\Phi(\underline{r}(x)), \Phi(\overline{r}(x))]$, where $[\underline{r}(x), \overline{r}(x)]$ is the interval of possible values of the ratio $r \stackrel{\text{def}}{=} \frac{x-a}{\sigma}$. Thus, to compute the desired p-box, it is sufficient to compute this interval $[\underline{r}(x), \overline{r}(x)]$.

Comment. To make the following text easier to read, we will write $\underline{r}$ instead of $\underline{r}(x)$ and $\overline{r}$ instead of $\overline{r}(x)$. A reader should keep in mind, however, that for different x, generally, we get different bounds $\underline{r}$ and $\overline{r}$.

The need to consider the signs: informal explanation. We have already mentioned that we know how to compute the bounds on the absolute value $|r|$ of the ratio r; see, e.g., [6]. The absolute value can be equal either to the ratio itself or to $-r$. Here:

- If $r \geq 0$ and $|r| = r$, then, e.g., the maximum of $|r|$ is the same as the maximum of r.
- On the other hand, if $r < 0$ and $|r| = -r$, then the maximum of the absolute value may correspond to the minimum of r.

So, to apply the results and techniques from [6] to our problem, we must first analyze the signs of the corresponding extreme values $\underline{r}$ and $\overline{r}$.

Signs of the bounds $\underline{r}$ and $\overline{r}$. In view of the above, it is reasonable to first find out the signs of the bounds $\underline{r}$ and $\overline{r}$ of the desired interval.

Proposition 1.

- *For $x \leq \underline{a}$, we have $\underline{r} \leq \overline{r} \leq 0$.*
- *For $\underline{a} < x < \overline{a}$, we have $\underline{r} < 0 < \overline{r}$.*
- *For $x \geq \underline{a}$, we have $0 \leq \underline{r} \leq \overline{r}$.*

Comment. For reader's convenience, all the proofs are placed in the special Appendix.

General idea: using basic facts about derivatives. Let us fix the value x. For this x, each tuple $(x_1, \ldots, x_n)$ from the box $B \stackrel{\text{def}}{=} \mathbf{x}_1 \times \ldots\ldots \mathbf{x}_n$ leads, in general, to a different value of the ratio r. The ratio is a continuous function of $(x_1, \ldots, x_n)$; thus, both its smallest and its largest values are attained at some tuple. (To be more precise, we first have to add $-\infty$ and $+\infty$ to the set of possible values of r to take care of the possibility that $\sigma = 0$.)

Let $(x_1^M, \ldots, x_n^M)$ be a tuple at which the ratio r attains its largest possible value. If we fix all the values except one x_i, then we conclude that the corresponding function $r(x_1^M, \ldots, x_{i-1}^M, x_i, x_{i+1}^M, \ldots, x_n^M)$ also attains its maximum for $x_i = x_i^M$. There are three possible cases:

- If the ratio r attains its maximum at $x_i \in (\underline{x}_i, \overline{x}_i)$, then, according to calculus, we should have $\frac{\partial r}{\partial x_i} = 0$ at this point.
- If r attains its maximum at $x_i = \underline{x}_i$, then the derivative $\frac{\partial r}{\partial x_i}$ at this point cannot be positive – else we would have even larger values for $x_i > \underline{x}_i$; thus, we should have $\frac{\partial r}{\partial x_i} \leq 0$.

- Similarly, if r attains its maximum at $x_i = \overline{x}_i$, then at this point, $\dfrac{\partial r}{\partial x_i} \geq 0$.

For the point $(x_1^m, \ldots, x_n^m)$ at which the ratio r attains its *minimum*, we similarly have three cases for each i:

- If the ratio r attains its minimum for $x_i \in (\underline{x}_i, \overline{x}_i)$, then $\dfrac{\partial r}{\partial x_i} = 0$.
- If r attains its minimum at $x_i = \underline{x}_i$, then $\dfrac{\partial r}{\partial x_i} \geq 0$.
- Finally, if r attains its minimum at $x_i = \overline{x}_i$, then at this point, $\dfrac{\partial r}{\partial x_i} \leq 0$.

The corresponding analysis leads to the following algorithm. In this algorithm, we assumed that the value x is given. If we need to find the range $[\underline{F}(x), \overline{F}(x)]$ for several different values x, we need to repeat this algorithm for each of these values x.

Algorithm A_1. In this algorithm, we consider all possible partitions of the set of indices $\{1, \ldots, n\}$ into three disjoint subsets I^-, I^+, and I_0. For each subdivision we set $x_i = \underline{x}_i$ for $i \in I^-$ and $x_i = \overline{x}_i$ for $i \in I^+$. When $I_0 \neq \emptyset$, we set the values x_i for $i \in I_0$ as follows:

- We compute the values
$$\widetilde{a} = \sum_{i \in I^-} \underline{x}_i + \sum_{j \in I^+} \overline{x}_j, \quad \widetilde{m} = \sum_{i \in I^-} (\underline{x}_i)^2 + \sum_{j \in I^+} (\overline{x}_j)^2.$$
- We find the value a from the quadratic equation
$$\widetilde{m} + \frac{1}{N_0} \cdot (n \cdot a - \widetilde{a})^2 = n \cdot \left(a - \frac{n \cdot a - \widetilde{a}}{N_0} \right) \cdot (x - a) + n \cdot a^2,$$
where N_0 is the number of elements in the set I_0, and then compute $a_0 = \dfrac{n \cdot a - \widetilde{a}}{N_0}$.
- If this quadratic equation does not have any real solutions, or if it does but the corresponding value a_0 does not belong to all intervals $\mathbf{x}_i$ with $i \in I_0$, then we skip this partition and go to the next one.
- For each solution a_0 that belongs to all the intervals $\mathbf{x}_i$, $i \in I_0$, we set $x_i = a_0$ for $i \in I_0$ and compute $\sigma = \sqrt{(a - a_0) \cdot (x - a)}$ and $r = \dfrac{x - a}{\sigma}$.

The smallest of these values r is returned as $\underline{r}$, and the largest is returned as $\overline{r}$. Then, we compute the desired p-box as $[\Phi(\underline{r}), \Phi(\overline{r})]$.

Proposition 2. *The above algorithm always computes the exact range $[\underline{F}(x), \overline{F}(x)]$ of the normal cdf under interval uncertainty.*

Comments.

- For each of n indices i, we have 3 choices: we can assign this index to I^-, to I^+, and to I_0. For a single index, we have 3 possible assignments; for two indices, we have $3 \cdot 3 = 2^2$ possible assignments; in general, for n indices, we have 3^n possible assignments. Thus, this algorithm requires an exponential number of computational steps which grows with n as 3^n.
- In this algorithm, the values x_i at which the minimum and the maximum of r are assigned depend, in general, on the value x at which we estimate $F(x)$. So, in general, to find the range $[\underline{F}(x), \overline{F}(x)]$ at N_p points x, we have to repeat this algorithm N_p times.

Some bounds can be computed faster. It turns out that some of the bounds can be computed in polynomial time, namely, the upper bound $\overline{r}$ for $x \geq \underline{a}$ and the lower bound $\underline{r}$ for $x \leq \overline{a}$.

Algorithm A_2. To find $\overline{r}$ for $x \geq \underline{a}$, we do the following:

- First, we sort all $2n$ values $\underline{x}_i$ and $\overline{x}_i$ into a non-decreasing sequence $x_{(1)} \leq x_{(2)} \leq \ldots \leq x_{(2n)}$. Thus, we subdivide the real line into $2n+1$ zones $[x_{(0)}, x_{(1)}]$, $[x_{(1)}, x_{(2)}]$, $\ldots$, $[x_{(2n-1)}, x_{(2n)}]$, $[x_{(2n)}, x_{(2n+1)}]$, where we denoted $x_0 \stackrel{\text{def}}{=} -\infty$ and $x_{(2n+1)} \stackrel{\text{def}}{=} +\infty$.
- For each zone $[x_{(k)}, x_{(k+1)}]$, we partition indices $i = 1, \ldots, n$ into three sets
$$I^- = \{i : x_{(k+1)} \leq \underline{x}_i\}, \quad I^+ = \{i \notin I^- : x_{(k)} \geq \overline{x}_i\},$$
$$I_0 = \{1, \ldots, n\} - I^- - I^+.$$
Based on this partition, we compute $\widetilde{a}$, $\widetilde{m}$, a, and a_0 as in Algorithm A_1. For each value a_0 which is within the zone, we compute $\sigma = \sqrt{(a - a_0) \cdot (x - a)}$ and $r = \dfrac{x - a}{\sigma}$.
- The largest of the resulting values r is the desired $\overline{r}$.

Comment. To find $\underline{r}$ for $x \leq \overline{a}$, we perform the same computations, with the only difference that at the end, instead of finding the *largest* of the resulting values r, we find the *smallest* of these values.

Proposition 3. *The above algorithms always computes the exact bound $\overline{r}$ for $x \geq \underline{a}$ and $\underline{r}$ for $x \leq \overline{a}$, and they require quadratic time $O(n^2)$.*

III. Efficient Algorithm for the No-Nesting Case

Let us show that in a no-nesting case, when no two intervals are nested, i.e., when $[\underline{x}_i, \overline{x}_i] \not\subseteq (\underline{x}_j, \overline{x}_j)$ for all $i \neq j$. In this case, we can compute the remaining bounds $\underline{r}$ for $x > \overline{a}$ and $\overline{r}$ for $x < \underline{a}$ also in polynomial time.

It is known that intervals which satisfy the no-nesting property can be ordered in "lexicographic" order, i.e., the order in which $[\underline{x}_i, \overline{x}_i] < [\underline{x}_j, \overline{x}_j]$ if and only if either $\underline{x}_i < \underline{x}_j$ or ($\underline{x}_i = \underline{x}_j$ and $\overline{x}_i \leq \overline{x}_j$); see, e.g., [7], [8]. With respect to this order, both sequences $\underline{x}_i$ and $\overline{x}_i$ become monotonic: $\underline{x}_1 \leq \underline{x}_2 \leq \ldots \leq \underline{x}_n$ and $\overline{x}_1 \leq \overline{x}_2 \leq \ldots \leq \overline{x}_n$. We have used this order in our previous algorithms [7], [8], and we will use it here as well.

Algorithm A_3. To find $\underline{r}$ for $x > \overline{a}$, we do the following:

- First, we sort all n intervals $[\underline{x}_i, \overline{x}_i]$ in the lexicographic order. As a result, we get two monotonic sequences
$$\underline{x}_1 \leq \underline{x}_2 \leq \ldots \leq \underline{x}_n, \quad \overline{x}_1 \leq \overline{x}_2 \leq \ldots \leq \overline{x}_n.$$

- For every n^- from 1 to n, we consequently compute $\sum_{i=1}^{n^-} \underline{x}_i$ and $\widetilde{m} = \sum_{i=1}^{n^-} (\underline{x}_i)^2$: we start with 0 and we consequently add, correspondingly, $\underline{x}_i$ or $(\underline{x}_i)^2$.
- For every n^+ from 1 to $n+1$, we consequently compute $\sum_{i=n^+ +1}^{n} \overline{x}_i$ and $\widetilde{m} = \sum_{i=n^+ +1}^{n} (\overline{x}_i)^2$: we start with 0 for $n^+ = n+1$ and then we take $n^+ = n, n-1, \ldots, 1$ by consequently adding, correspondingly, $\overline{x}_i$ or $(\overline{x}_i)^2$.
- For every two natural numbers n^- and n^+ for which $0 \le n^- < n^+ \le n+1$, we do the following:
  - We compute the values $N_0 = n - n^- - (n+1-n^+)$ and
$$\widetilde{a} = \sum_{i=1}^{n^-} \underline{x}_i + \sum_{j=n^+}^{n} \overline{x}_j, \quad \widetilde{m} = \sum_{i=1}^{n^-} (\underline{x}_i)^2 + \sum_{j=n^+}^{n} (\overline{x}_j)^2.$$
  - We find the value a from the same quadratic equation
$$\widetilde{m} + \frac{1}{N_0} \cdot (n \cdot a - \widetilde{a})^2 =$$
$$n \cdot \left(a - \frac{n \cdot a - \widetilde{a}}{N_0} \right) \cdot (x - a) + n \cdot a^2,$$
as in Algorithm A_1 (with $N_0 = n^+ - n^- - 1$), and then compute $a_0 = \dfrac{n \cdot a - \widetilde{a}}{N_0}$.
  - If this quadratic equation does not have any real solutions, or if it does but the corresponding value a_0 does not belong to the interval $[\underline{x}_{n^+-1}, \overline{x}_{n^-+1}]$, then we skip this partition and go to the next one.
  - For each solution a_0 which belongs to the interval $[\underline{x}_{n^+-1}, \overline{x}_{n^-+1}]$, we compute $\sigma = \sqrt{(a - a_0) \cdot (x - a)}$ and $r = \dfrac{x - a}{\sigma}$.
- The smallest of the resulting values r is the desired $\overline{r}$.

Comments. To find $\overline{r}$ for $x < \underline{a}$, we perform the same computations, with the only difference that at the end, instead of finding the *smallest* of the resulting values r, we find the *largest* of these values.

Proposition 4. *The above algorithms always computes the exact bound $\underline{r}$ for $x > \overline{a}$ and $\overline{r}$ for $x < \underline{a}$, and they require quadratic time $O(n^2)$.*

ACKNOWLEDGMENTS

This work was supported in part by NSF grants EAR-0225670 and DMS-0532645 and by Texas Department of Transportation grant No. 0-5453.

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

REFERENCES

[1] G. Bojadziev and M. Bojadziev, *Fuzzy sets, fuzzy logic, applications*, World Scientific, Singapore, 1995.

[2] D. Dubois and H. Prade, "Operations on fuzzy numbers", *International Journal of Systems Science*, 1978, Vol. 9, pp. 613–626.

[3] S. Ferson, *RAMAS Risk Calc 4.0*, CRC Press, Boca Raton, Florida, 2002.

[4] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis*, Springer, London, 2001.

[5] G. Klir and B. Yuan, *Fuzzy sets and fuzzy logic*, Prentice Hall, New Jersey, 1995.

[6] V. Kreinovich, L. Longpré, P. Patangay, S. Ferson, and L. Ginzburg, "Outlier Detection Under Interval Uncertainty: Algorithmic Solvability and Computational Complexity", *Reliable Computing*, 2005, Vol. 11, No. 1, pp. 59–76.

[7] 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*, 2007, Vol. 199, No. 2, pp. 418–423.

[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*, 2006, Vol. 12, No. 6, pp. 471–501.

[9] R. E. Moore and W. A. Lodwick, "Interval analysis and fuzzy set theory", *Fuzzy Sets and Systems*, 2003, Vol. 135, No. 1, pp. 5–9.

[10] H. T. Nguyen and V. Kreinovich, "Nested intervals and sets: concepts, relations to fuzzy sets, and applications", In: R. B. Kearfott and V. Kreinovich (eds), *Applications of interval computations*, Kluwer, Dordrecht, pp. 245–290.

[11] H. T. Nguyen and E. A. Walker, *A First Course in Fuzzy Logic*, CRC Press, Boca Raton, Florida, 2006.

[12] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, Chapman & Hall/CRC, Boca Raton, Florida, 2004.

[13] H. M. Wadsworth, Jr. (editor), *Handbook of statistical methods for engineers and scientists*, McGraw-Hill Publishing Co., N.Y., 1990.

APPENDIX A: PROOF OF PROPOSITION 1

We know that the mean a can take any values from the interval $[\underline{a}, \overline{a}]$. When $x \le \underline{a}$, this means that the value $x - a$ is always non-positive. Since the standard deviation σ is always non-negative, the ratio $\dfrac{x-a}{\sigma}$ is also non-positive. Therefore, both the smallest and the largest values of this ratio are non-positive: $\underline{r} \le 0$ and $\overline{r} \ge 0$.

Similarly, when $x \ge \overline{a}$, we have $x - a \ge 0$, hence the ratio r is non-negative and its bounds are also non-negative.

When $\underline{a} < x < \overline{a}$, the difference $x - a$ can take both positive values (e.g., when $x = \overline{a}$) and negative values (e.g., when $x = \underline{a}$). Thus, the ratio r can also be both positive and negative. Hence, the largest possible value of this ratio is positive, and the smallest possible value of this ratio is negative.

APPENDIX B: PROOF OF PROPOSITION 2

1°. Within each interval $[\underline{x}_i, \overline{x}_i]$, the value x_i corresponding to the optimal tuple can be either at the left endpoint $\underline{x}_i$ or at the right endpoint $\overline{x}_i$, or inside the interval $(\underline{x}_i, \overline{x}_i)$. Let us denote the set of all indices for which $x_i = \underline{x}_i$ by I^-, the set of all indices for which $x_i = \overline{x}_i$ by I^+, and the set of all remaining indices by I_0.

2°. According to the arguments described before the formulation of this proposition, for every i, either $x_i = \underline{x}_i$ or $x_i = \overline{x}_i$, or $\dfrac{\partial r}{\partial x_i} = 0$. Let us therefore describe an explicit formula for this derivative.

2.1°. Since r is defined in terms of a and σ, let us first find the formulas for the derivatives of a and σ.

Since $a = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i$, we have $\frac{\partial a}{\partial x_i} = \frac{1}{n}$. Since $\sigma = \sqrt{V}$, where

$$V \stackrel{\text{def}}{=} \frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - a)^2 = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i^2 - a^2,$$

we have

$$\frac{\partial \sigma}{\partial x_i} = \frac{1}{2\sigma} \cdot \frac{\partial V}{\partial x_i}.$$

Here,

$$\frac{\partial V}{\partial x_i} = \frac{2}{n} \cdot x_i - 2a \cdot \frac{\partial a}{\partial x_i} = \frac{2}{n} \cdot (x_i - a).$$

Therefore, we have

$$\frac{\partial \sigma}{\partial x_i} = \frac{1}{n \cdot \sigma} \cdot (x_i - a).$$

2.2°. Now, we are ready to compute the desired derivative. Here,

$$\frac{\partial r}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\frac{x - a}{\sigma} \right) = \frac{-\frac{\partial a}{\partial x_i} \cdot \sigma - (x - a) \cdot \frac{\partial \sigma}{\partial x_i}}{\sigma^2}.$$

In view of the analysis that preceded the formulation of this proposition, we are only interested in the sign of the derivative $\frac{\partial r}{\partial x_i}$. Since the denominator σ^2 of the expression describing this derivative is always non-negative, this sign coincides with the sign of the numerator

$$N_i \stackrel{\text{def}}{=} -\frac{\partial a}{\partial x_i} \cdot \sigma - (x - a) \cdot \frac{\partial \sigma}{\partial x_i}.$$

Substituting the above expressions for $\frac{\partial a}{\partial x_i}$ and $\frac{\partial \sigma}{\partial x_i}$ into this formula, we conclude that

$$N_i = -\frac{1}{n} \cdot \sigma - (x - a) \cdot \frac{1}{n \cdot \sigma} \cdot (x_i - a).$$

We can simplify this expression even further if we multiply it by $n \cdot \sigma$ – which also does not change the signs. Thus, the sign of the desired derivative $\frac{\partial r}{\partial x_i}$ coincides with the sign of the product $p_i \stackrel{\text{def}}{=} n \cdot \sigma \cdot N_i$, which is equal to

$$p_i = -(x_i - a) \cdot (x - a) - \sigma^2.$$

3°. The only possibility for x_i to be inside the interval $(\underline{x}_i, \overline{x}_i)$ is to have $p_i = 0$. Dividing both sides by $x - a$, we conclude that $x_i = a_0$, where we denoted

$$a_0 \stackrel{\text{def}}{=} a - \frac{\sigma^2}{x - a}.$$

Thus, all the values x_i with $i \in I_0$ have exactly the same value a_0.

Once we know the partition into the sets I^-, I^+, and I_0, we also know the values x_i for $i \in I^-$ and $i \in I^+$. To find the values x_i for $i \in I_0$, we need to find the value a_0.

4°. By definition of the sample mean a, the sum of all n values x_i is equal to $n \cdot a$, i.e.,

$$\sum_{i \in I^-} \underline{x}_i + \sum_{i \in I^+} \overline{x}_i + N_0 \cdot a_0 = n \cdot a.$$

The sum of the first two sums is what we denoted by $\widetilde{a}$; so, we conclude that $\widetilde{a} + N_0 \cdot a_0 = n \cdot a$ and hence, that

$$a_0 = \frac{n \cdot a - \widetilde{a}}{N_0}. \tag{4}$$

Since $a_0 = a - \frac{\sigma^2}{x - a}$, we conclude that

$$\sigma^2 = (a - a_0) \cdot (x - a) = \left(a - \frac{n \cdot a - \widetilde{a}}{N_0} \right) \cdot (x - a). \tag{5}$$

By definition of the sample variance, we have $\sum_{i=1}^{n} x_i^2 = n \cdot a^2 + n \cdot \sigma^2$, i.e.,

$$\sum_{i \in I^-} (\underline{x}_i)^2 + \sum_{i \in I^+} (\overline{x}_i)^2 + N_0 \cdot a_0^2 = n \cdot a^2 + n \cdot \sigma^2.$$

The sum of the first two sums is what we denoted by $\widetilde{m}$; so, we conclude that $\widetilde{m} + N_0 \cdot a_0^2 = n \cdot a^2 + n \cdot \sigma^2$. Substituting the expressions (4) and (5) for a_0 and σ^2 into this formula, we get the quadratic equation given in the algorithm.

So, the optimal solution is indeed among those processed by the algorithm. The proposition is proven.

Appendix C: Proof of Proposition 3

1°. We have already proven that the sign of the desired derivative $\frac{\partial r}{\partial x_i}$ coincides with the sign of the product $p_i = -(x_i - a) \cdot (x - a) - \sigma^2$. According to Proposition **??**, when we are looking for $\overline{r}$ and $x \geq \overline{a}$, then $x - a \geq 0$. In this case, the sign of p_i coincides with the sign of the ratio

$$r_i \stackrel{\text{def}}{=} \frac{p_i}{x - a} = a_0 - x_i.$$

So we make the following conclusions:

(i) If the maximum $\overline{r}$ is attained for $x_i \in (\underline{x}_i, \overline{x}_i)$, then (the derivative is 0 hence) $x_i = a_0$.

(ii) If the maximum is attained for $x_i = \underline{x}_i$, then (the derivative is non-positive hence) $\underline{x}_i \geq a_0$.

(iii) Finally, if the maximum is attained for $x_i = \overline{x}_i$, then (the derivative is non-negative hence) $\overline{x}_i \leq a_0$.

Therefore, if $a_0 < \underline{x}_i$, then we cannot have the case (i) when $\underline{x}_i \leq x_i = a_0$ and we cannot have the case (iii) when $\underline{x}_i \leq \overline{x}_i \leq a_0$, so we must have case (ii), i.e., we must have $x_i = \underline{x}_i$.

Similarly, if $a_0 > \overline{x}_i$, then our only option is $x_i = \overline{x}_i$, and if $\underline{x}_i \leq a_0 \leq \overline{x}_i$, then our only option is $x_i = a_0$. Thus, as soon as we know the location of the value a_0 in comparison to the bounds $\underline{x}_i$ and $\overline{x}_i$ – i.e., as soon as we know the zone which contains a_i – we can (almost) uniquely determine the values x_i for all x_i – with the only additional problem that we still need to determine the value a_0. We already described, in Algorithm A_1, how we can find a_0.

The case of $\underline{r}$ for $x \leq \underline{a}$ is handled similarly.

2°. To complete the proof, it is sufficient to show that these algorithms require quadratic time. Indeed, in addition to sorting (time $O(n \cdot \log(n))$), this algorithm requires linear time for each of $2n+1$ zones. So, overall, we need $O(n^2)$ computational steps. The proposition is proven.

Appendix D: Proof of Proposition 4

1°. We have already proven that in the optimal tuple, each value of x_i is either equal to $\underline{x}_i$ or to $\overline{x}_i$, or to a common value a_0. Let us now prove that in the no-nesting case, we can also assume that the optimal sequence x_i is monotonic, i.e., $x_1 \le x_2 \le \ldots \le x_n$.

Indeed, let us assume that in the optimal sequence, we have $x_i > x_j$ for some $i < j$. Here, we have $\underline{x}_i \le x_i \le \overline{x}_i$, $\underline{x}_j \le x_j \le \overline{x}_j$, and, since $i < j$, we also have $\underline{x}_i \le \underline{x}_j$ and $\overline{x}_i \le \overline{x}_j$. Let us show that in this case, $x_i \in \mathbf{x}_j$ and $x_j \in \mathbf{x}_i$.

Indeed, from $x_i \le \overline{x}_i$ and $\overline{x}_i \le \overline{x}_j$, we conclude that $x_i \le \overline{x}_j$. Similarly, from $\underline{x}_j \le x_j$ and $x_j < x_i$, we conclude that $\underline{x}_i < x_j$. Thus, indeed $x_i \in [\underline{x}_i, \overline{x}_i]$. Similarly, we have $x_i \in [\underline{x}_i, \overline{x}_i]$.

Because of these two inclusions, we can "swap" the values x_i and x_j, i.e., produce a new tuple in which $x_i^{\text{new}} = x_j$ and $x_j^{\text{new}} = x_i$. The values of sample mean a and sample standard deviation σ do not change if we simply swap two values. So, for this new tuple, we can the exact same values of a, σ and therefore, the same value of the ratio r. Since the original tuple maximized r, the new tuple is also maximizing r.

By repeating this swapping sufficiently many times, we will get a monotonic optimizing tuple.

2°. Let us now prove that if in the optimal solution, we have $x_i > \underline{x}_i$ and $x_j < \overline{x}_j$, then we should have $x_i \ge x_j$.

Indeed, in this case, for sufficiently small $h > 0$, we can simultaneously do the following:

- decrease x_i by h, i.e., replace it with $x_i^{\text{new}} = x_i - h$, and
- increase x_j by h, i.e., replace it with $x_j^{\text{new}} = x_j + h$,

and still keep both values x_i and x_j within the corresponding intervals $[\underline{x}_i, \overline{x}_i]$ and $[\underline{x}_j, \overline{x}_j]$.

Since the value of r was originally the smallest, it cannot decrease under this replacement. After the replacement, the sum $\sum x_i$ does not change hence the average a does not change, and the value of the numerator $x - a > 0$ does not change either.

The value of $\sigma^2 = \frac{1}{n} \cdot \sum x_i^2 - a^2$ changes since two terms change: x_i^2 to $(x_i - h)^2 = x_i^2 - 2h \cdot x_i + o(h)$ and x_j^2 to $(x_j + h)^2 = x_j^2 + 2h \cdot x_j + o(h)$. Thus, overall, σ^2 is replaced with $\sigma^2 + \frac{2h}{n} \cdot (x_j - x_i) + o(h)$. We cannot have an increase in σ – that would lead to an impossible decrease of r below it smallest value. Thus, the new value of σ^2 cannot be larger than its original value. In other words, we must have

$$\sigma^2 + \frac{2h}{n} \cdot (x_j - x_i) + o(h) \ge \sigma^2.$$

Subtracting σ^2 from both sides and dividing both sides by $h \ge 0$, we conclude that $x_j - x_i + o(1) \le 0$. In the limit $h \to 0$, we get the desired inequality $x_j \le x_i$.

3°. So, in the optimal tuple, every value $x_i = \underline{x}_i$ must precede every value $x_j = \overline{x}_j$, and all the values $x_i = a_0 \in (\underline{x}_i, \overline{x}_i)$ must be in between. Due to monotonicity, we therefore conclude that first we have a sequence of several values $\underline{x}_i$, then several values equal to a_0, and after that, several values equal to $\overline{x}_j$. This is exactly the type of solution we analyze in the algorithm.

For each selection of n^- and n^+, we need to check whether the value a_0 is indeed contained in all the corresponding intermediate intervals $[\underline{x}_i, \overline{x}_i]$ for $i = n^- + 1, \ldots, n^+ - 1$. Since the sequence $\underline{x}_i$ is increasing, it is sufficient to check the inequality $a_0 \ge \underline{x}_i$ only for the largest of these bounds, i.e., for the bound $\underline{x}_{n^+-1}$. Similarly, since the sequence $\overline{x}_i$ is increasing, it is sufficient to check the inequality $a_0 \le \overline{x}_i$ only for the smallest of these bounds, i.e., for the bound $\overline{x}_{n^-+1}$. Thus, the algorithm is justified.

4°. To complete the proof, it is sufficient to show that the algorithm A_3 require quadratic time. Indeed, in addition to sorting (time $O(n \cdot \log(n))$) and linear time for computing the sums $\sum \underline{x}_i$, $\sum (\underline{x}_i)^2$, $\sum \overline{x}_i$, $\sum (\overline{x}_i)^2$, we need a constant time for each of n^2 pairs of indices – i.e., $O(n^2)$ computational steps overall. The proposition is proven.