7-1-2006

Fast Computation of Exact Ranges of Symmetric Convex and Concave Functions under Interval Uncertainty

Gang Xiang

Comments:
UTEP-CS-06-33

Fast Computation of Exact Ranges of Symmetric Convex and Concave Functions under Interval Uncertainty

Gang Xiang
Department of Computer Science
The University of Texas at El Paso
El Paso, TX 79968, USA
email gxiang@utep.edu

Abstract

Many statistical characteristics $y = f(x_1, \ldots, x_n)$ are continuous, symmetric, and either concave or convex; examples include population variance
$V = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i^2 - E^2$ $\left(\text{where } E = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i\right)$, Shannon's entropy $S = -\sum_{i=1}^{n} p_i \cdot \log(p_i)$, and many other characteristics. In practice, often, we often only know the intervals $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$ that contain the (unknown) actual inputs x_i. Since different values $x_i \in \mathbf{x}_i$ lead, in general, to different values of $f(x_1, \ldots, x_n)$, we need to find the range $\mathbf{y} \stackrel{\text{def}}{=} \{f(x_1, \ldots, x_n) : x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x_n}\}$, i.e., the maximum and the minimum of $f(x_1, \ldots, x_n)$ over the box $\mathbf{x}_1 \times \ldots \times \mathbf{x}_n$. It is known that for convex functions, there exists a feasible (polynomial-time) algorithm for computing its minimum, but computing its maximum is, in general, NP-hard. It is therefore desirable to find feasible algorithms that compute the maximum in practically reasonable situations. For variance and (negative) entropy, such algorithms are known for the case when the inputs satisfy the following *subset property*: $[\underline{x}_i, \overline{x}_i] \not\subset (\underline{x}_j, \overline{x}_j)$ for all i and j. In this paper, we show that these algorithms can be extended to the case of general symmetric convex characteristics.

1 Introduction

Many statistical characteristics $y = f(x_1, \ldots, x_n)$ are continuous, symmetric – in the sense that

$$f(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_{j-1}, x_j, x_{j+1}, \ldots, x_n) =$$

$$f(x_1, \ldots, x_{i-1}, x_j, x_{i+1}, \ldots, x_{j-1}, x_i, x_{j+1}, \ldots, x_n)$$

for all $x_1, \ldots, x_n$, i, and j – and convex, i.e.,

$$f(\alpha \cdot x_1 + (1-\alpha) \cdot x'_1, \ldots, \alpha \cdot x_n + (1-\alpha) \cdot x'_n) \leq$$

$$\alpha \cdot f(x_1, \ldots, x_n) + (1-\alpha) \cdot f(x'_1, \ldots, x'_n)$$

for all $x_1, \ldots, x_n$ and for all $\alpha \in [0,1]$. Some characteristics are *concave*, i.e.,

$$f(\alpha \cdot x_1 + (1-\alpha) \cdot x'_1, \ldots, \alpha \cdot x_n + (1-\alpha) \cdot x'_n) \geq$$

$$\alpha \cdot f(x_1, \ldots, x_n) + (1-\alpha) \cdot f(x'_1, \ldots, x'_n)$$

for all $x_1, \ldots, x_n$ and for all $\alpha \in [0,1]$.

Examples of such characteristics include population variance

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

$\left(\text{where } E = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i\right)$, Shannon's entropy $S = -\sum_{i=1}^{n} p_i \cdot \log(p_i)$, and many others. Many important physical quantities outside statistics are also convex (or concave); see, e.g., [1, 2, 13, 16]. Some of these convex or concave characteristics are also symmetric.

In practice, often, we often only know the intervals $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$ that contain the (unknown) actual inputs x_i; see, e.g., [12]. Since different values $x_i \in \mathbf{x}_i$ lead, in general, to different values of $f(x_1, \ldots, x_n)$, we need to find the range

$$\mathbf{y} = [\underline{y}, \overline{y}] \stackrel{\text{def}}{=} \{f(x_1, \ldots, x_n) : x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x_n}\},$$

i.e., the maximum $\underline{y}$ and the minimum $\overline{y}$ of $f(x_1, \ldots, x_n)$ over the box $\mathbf{x}_1 \times \ldots \times \mathbf{x}_n$; practical examples of such situations are given, e.g., in [10, 11, 14, 17]. It is known that for convex functions, there exists a feasible (polynomial-time) algorithm for computing its minimum (see, e.g., [2, 15]), but computing its maximum is, in general, NP-hard [15]; it is even NP-hard for population variance [5, 6]. It is therefore desirable to find feasible algorithms that solve the maximum in practically reasonable situations.

For variance [4, 8, 9] and entropy [7], such algorithms are known for the case when the inputs satisfy the following *subset property*:

Definition 1 *We say that intervals* $\mathbf{x}_1 = [\underline{x}_1, \overline{x}_1]$, …, $\mathbf{x}_n = [\underline{x}_n, \overline{x}_n]$ *satisfy the* subset property *if* $[\underline{x}_i, \overline{x}_i] \not\subset (\underline{x}_j, \overline{x}_j)$ *for all* i *and* j.

This property makes sense, e.g., in the measurement situations when the only information that we have about the (unknown) value x_i of the measured quantity is that it cannot differ from the measured value $\widetilde{x}_i$ by more than the bound Δ_i provided by the manufacturer. In this case, possible values of x_i

form an interval $[\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$. A measuring instrument can have different accuracies on different parts of the scale, so values Δ_i corresponding to different measurement $\widetilde{x}_i$ can be different; however, when we measure all the values $x_1, \ldots, x_n$ with the same measuring instrument, we cannot have two intervals contained in each other.

In this paper, we show that these feasible algorithms can be extended to the case of general symmetric convex characteristics, such as the higher even central moments $\sum\limits_{i=1}^{n}(x_i - E)^{2d}$, where $d = 2, 3, \ldots$ and generalized entropy functions $\sum\limits_{i=1}^{n} g(p_i)$.

In this generalization, we consider two cases: the case (similar to population variance) where there is no constraint on the possible values of x_i, and the case (similar to entropy) where there is a constraint $\sum\limits_{i=1}^{n} x_i = c$ on the values $x_i \in \mathbf{x}_i$; in the entropy case, the variables are probabilities p_i and the constraint is $\sum\limits_{i=1}^{n} p_i = 1$.

2 Computing $\overline{y}$

In this section, we solve the following problem: we know a symmetric convex function $f(x_1, \ldots, x_n)$ (given as an algorithm or, equivalently, as a computer program), and we know the intervals $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$ that satisfy the above subset property. Our objective is to compute the value

$$\overline{y} = \max\{f(x_1, \ldots, x_n) : x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n\}.$$

A function $f(x_1, \ldots, x_n)$ is concave if and only if the function $f'(x_1, \ldots, x_n) \stackrel{\text{def}}{=} -f(x_1, \ldots, x_n)$ is convex. Thus, if we know how to compute the upper bound $\overline{y}$ for symmetric convex functions, we can compute the exact lower bound $\underline{y}$ for symmetric concave functions, by first computing the exact upper bound $\overline{y}'$ for the function $f'(x_1, \ldots, x_n)$, and then computing $\underline{y} = -\overline{y}'$.

Our algorithm (and algorithms presented in the following sections) calls the function f several times. So, we can gauge the computation time required by our algorithm by counting how many times it calls f.

Theorem 1 *There exists an algorithm that computes the maximum $\overline{y}$ of a given symmetric convex function $f(x_1, \ldots, x_n)$ over a given box $\mathbf{x}_1 \times \ldots \times \mathbf{x}_n$ for all the cases in which the intervals $\mathbf{x}_i$ satisfy the subset property; this algorithm:*

- *calls the function f $n + 1$ times, and*
- *uses $O(n \cdot \log(n))$ computational steps in addition to these calls.*

Comment. So, if the algorithm for computing the function f is feasible, i.e., requires a polynomial time $t \le P(n)$ for some polynomial $P(n)$, then computing $\overline{y}$ requires time $\le P(n) \cdot (n + 1) + O(n \cdot \log(n))$ – i.e., also a polynomial time.

Algorithm. First, let us first describe the algorithm; it consists of the following steps.

- First, we sort n intervals $\mathbf{x}_i$ in lexicographic order

$$\mathbf{x}_1 \leq_{\text{lex}} \mathbf{x}_2 \leq_{\text{lex}} \ldots \leq_{\text{lex}} \mathbf{x}_n,$$

  where $[\underline{a}, \overline{a}] \leq_{\text{lex}} [\underline{b}, \overline{b}]$ if and only if either $\underline{a} < \overline{b}$, or $\underline{a} = \underline{b}$ and $\overline{a} \leq \overline{b}$.

- Second, for each k from 0 to n, we compute

$$F_k \stackrel{\text{def}}{=} f(\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n).$$

- Finally, we return the largest of $n+1$ values F_k as $\overline{y}$.

Computational complexity. The above algorithm requires $O(n \cdot \log(n))$ steps for sorting (see, e.g., [3]), $n+1$ calls to f (to compute $n+1$ values $F_0, \ldots, F_n$), and $O(n)$ steps to find the largest of these $n+1$ values. Thus, the overall computational complexity of all the additional computational steps (beyond the calls to f) is $O(n \cdot \log(n)) + O(n) = O(n \cdot \log(n))$.

Proof. Let us prove that this algorithm indeed computes $\overline{y}$. Let us assume that the intervals $\mathbf{x}_i$ are already sorted in lexicographic order.

1°. Let us first prove that in this order, the lower endpoints and the upper endpoints are also sorted, i.e., $\underline{x}_i \leq \underline{x}_{i+1}$ and $\overline{x}_i \leq \overline{x}_{i+1}$ for all i.

Indeed, the inequality $\underline{x}_i \leq \underline{x}_{i+1}$ directly follows from $\mathbf{x}_i \leq_{\text{lex}} \mathbf{x}_{i+1}$ and the definition of the lexicographic order. To prove that $\overline{x}_i \leq \overline{x}_{i+1}$, we will consider two cases:

- the case when $\underline{x}_i = \underline{x}_{i+1}$, and
- the case when $\underline{x}_i < \underline{x}_{i+1}$.

In the first case, the desired inequality $\overline{x}_i \leq \overline{x}_{i+1}$ follows from $\mathbf{x}_i \leq_{\text{lex}} \mathbf{x}_{i+1}$ and the definition of the lexicographic order. In the second case, this inequality follows from the subset property: indeed, if this inequality was not true, we would then have $\overline{x}_i > \overline{x}_{i+1}$ and hence, $[\underline{x}_{i+1}, \overline{x}_{i+1}] \subset (\underline{x}_i, \overline{x}_i)$ – contrary to the subset property.

The statement is proven.

2°. To prove that the algorithm is correct, we must prove that the maximum of the function f is attained at one of the points $x = (\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$ and, thus, that $\max f = F_k$ for some $k = 0, 1, \ldots, n$.

Indeed, since f is a continuous function on a bounded closed set $\mathbf{x}_1 \times \ldots \times \mathbf{x}_n \in \mathbb{R}^n$, its maximum is always attained at some point $x = (x_1, \ldots, x_n)$. Since f is a convex function on a convex polyhedron $\mathbf{x}_1 \times \ldots \times \mathbf{x}_n$, the maximum of the function f is attained at one of the vertices of this polyhedron [2, 13, 16], i.e., in this case, when each variable x_i attains either the values $\underline{x}_i$ or the value $\overline{x}_i$.

There may be several such vertices x at which the maximum is attained. The coordinates x_i of the maximizing point x may start with a sequence of lower bounds $\underline{x}_1, \ldots, \underline{x}_k$, or its first element x_1 may be an upper bound $x_1 = \overline{x}_1$ – in which case we can say that we have an initial sequence of lower bounds of length 0.

Out of all maximizing vertices x, we choose a one with the largest length of the starting sequence of lower bounds. We will denote this length by k; this means that the chosen point s has the form $s = (\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots)$, i.e., it starts with k lower bounds and then has an upper bound at the $(k+1)$-st place.

Let us prove, by reduction to a contradiction, that for this point s, all the components s_l for $l > k+1$ are upper bounds, i.e., that this point s has the desired form $x = (\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$.

Indeed, let us assume that for some $l > k+1$, the component s_l of the chosen point is a lower bound: $s_l = \underline{x}_l$. We will then show that there exists another point s' at which f also attains its maximum and which has a longer starting sequence of lower bounds – which contradicts to our choice of s. We will construct this new point s' in two steps; on each step, we will keep all the components of s unchanged, except for the $(k+1)$-st and l-th components.

Our procedure for producing s' will depend on the relation between the half-widths of the intervals $\mathbf{x}_{k+1}$ and $\mathbf{x}_l$. Let us denote the half-width of each interval $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$ by $\Delta_i \stackrel{\text{def}}{=} \dfrac{\overline{x}_i - \underline{x}_i}{2}$.

2.1°. Let us first consider the case when $\Delta_{k+1} \le \Delta_l$.

2.1.1°. Let us first prove that the maximum of f is also attained at an auxiliary point

$$s'' = (\underline{x}_1, \ldots, \underline{x}_k, \underline{x}_{k+1}, s_{k+2}, \ldots, s_{l-1}, \underline{x}_l + 2\Delta_{k+1}, s_{l+1}, \ldots, s_n)$$

(which differs from the original point s by only $(k+1)$-st and l-th components).

Since $\Delta_{k+1} \le \Delta_l$, we conclude that $\underline{x}_i + 2\Delta_{k+1} \le \overline{x}_l$ and therefore, that the point s'' belongs to the original box $\mathbf{x}_1 \times \ldots \times \mathbf{x}_n$.

Indeed, since the function f is symmetric, we conclude that $f(s'') = f(z'')$, where $z'' = (\underline{x}_1, \ldots, \underline{x}_k, \underline{x}_l + 2\Delta_{k+1}, s_{k+2}, \ldots, s_{l-1}, \underline{x}_{k+1}, s_{l+1}, \ldots, s_n)$ is a point in which the $(k+1)$-st and l-th components of the point s'' have been swapped. (To avoid confusion, it should be mentioned that the point z'' is a purely auxiliary point, it does not necessarily belong to the original box $\mathbf{x}_1 \times \ldots \times \mathbf{x}_n$.)

The chosen point s can be represented as a convex combination of the points s'' and z''. Indeed, since $k+1 < l$, then, due to Part 1 of this proof, we have $\underline{x}_{k+1} \le \underline{x}_l$. If we add $2\Delta_{k+1}$ to both sides of this inequality, we conclude that $\overline{x}_{k+1} \le \underline{x}_l + 2\Delta_{k+1}$, hence, that

$$\underline{x}_{k+1} \le \overline{x}_{k+1} \le \underline{x}_l + 2\Delta_{k+1}.$$

This means, in turn, that there exists a real value $\alpha \in [0,1]$ for which $\overline{x}_{k+1}$ is a convex combination of $\underline{x}_{k+1}$ and $\overline{x}_{k+1} \le \underline{x}_l + 2\Delta_{k+1}$:

$$\overline{x}_{k+1} = \alpha \cdot \underline{x}_{k+1} + (1-\alpha)\cdot(\underline{x}_l + 2\Delta_{k+1}).$$

Subtracting both sides of this inequality from

$$\overline{x}_{k+1} + \underline{x}_l = \underline{x}_{k+1} + (\underline{x}_l + 2\Delta_{k+1}),$$

we conclude that

$$\underline{x}_l = \alpha \cdot (\underline{x}_l + 2\Delta_{k+1}) + (1-\alpha)\cdot \underline{x}_{k+1}.$$

Since all other components of s, s'', and z'' are identical, we get

$$s = \alpha \cdot s'' + (1-\alpha)\cdot z''.$$

The function f is convex, hence $f(s) \le \alpha \cdot f(s'') + (1-\alpha)\cdot f(z'')$. We already know that (due to symmetry) $f(z'') = f(s'')$, so $\alpha \cdot f(s'') + (1-\alpha)\cdot f(z'') = f(s'')$ and hence, $f(s) \le f(s'')$.

By our selection of the point s, the value $f(s)$ is the largest value of f in the box: $f(s) = \max\limits_x f(x)$. Since s'' belongs to the same box, we thus have $f(s'') \le f(s)$ hence $f(s'') = f(s) = \max\limits_x f(x)$, i.e., the function f indeed attains its maximum at the point s''.

2.1.2°. Let us now prove that the maximum of f is also attained at an auxiliary point $s' = (\underline{x}_1, \ldots, \underline{x}_k, \underline{x}_{k+1}, s_{k+2}, \ldots)$. This will contradict to our assumption that k is the largest length of the starting sequence of lower bounds in an optimal vector x.

Similarly to the previous part of the proof, since $\Delta_{k+1} \le \Delta_l$, we have $\underline{x}_l \le \underline{x}_l + 2\Delta_{k+1} \le \overline{x}_l$. Thus,

$$\underline{x}_l + 2\Delta_{k+1} = \beta \cdot \underline{x}_l + (1-\beta)\cdot \overline{x}_l$$

for some $\beta \in [0,1]$, and therefore, $s'' = \beta \cdot s^- + (1-\beta)\cdot s^+$, where

$$s^- \stackrel{\text{def}}{=} (\underline{x}_1, \ldots, \underline{x}_k, \underline{x}_{k+1}, s_{k+2}, \ldots, s_{l-1}, \underline{x}_l, s_{l+1}, \ldots, s_n);$$

$$s^+ \stackrel{\text{def}}{=} (\underline{x}_1, \ldots, \underline{x}_k, \underline{x}_{k+1}, s_{k+2}, \ldots, s_{l-1}, \overline{x}_l, s_{l+1}, \ldots, s_n).$$

Since the function f is convex, we have $f(s'') \le \beta \cdot f(s^-) + (1-\beta)\cdot f(s^+)$.

Since the function f attains its maximum at the point s'', we have $f(x) \le f(s'')$ for all x from the box, in particular, we have $f(s^-) \le f(s'')$ and $f(s^+) \le f(s'')$. If we had $f(s^-) < f(s'')$ and $f(s^+) < f(s'')$, then we would be able to conclude that $f(s'') \le \beta \cdot f(s^-) + (1-\beta)\cdot f(s^+) < f(s'')$ and $f(s'') < f(s'')$ – a contradiction. Thus, at least one of the values $f(s^-)$ and $f(s^+)$ must be equal to the maximum value $f(s'')$. We can thus take $s' \in \{s^-, s^+\}$ to be equal to the point for which $f(s') = f(s'')$.

Each of the points s^- and s^+ has a starting sequence of lower bounds of length $\geq k+1$, which contradicts to our selection of the sequence s.

2.2°. To complete our proof, we must show that the same contradiction appears when $\Delta_{k+1} > \Delta_l$.

$2.2.1^\circ$. Let us first prove that the maximum of f is also attained at an auxiliary point

$$s'' = (\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1} - 2\Delta_l, s_{k+2}, \ldots, s_{l-1}, \overline{x}_l, s_{l+1}, \ldots, s_n)$$

(which differs from the original point s by only $(k+1)$-st and l-th components).

Since $\Delta_{k+1} > \Delta_l$, we conclude that $\overline{x}_{k+1} - 2\Delta_l > \underline{x}_{k+1}$ and therefore, that the point s'' belongs to the original box $\mathbf{x}_1 \times \ldots \times \mathbf{x}_n$.

Indeed, since the function f is symmetric, we conclude that $f(s'') = f(z'')$, where $z'' = (\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_l, s_{k+2}, \ldots, s_{l-1}, \overline{x}_{k+1} - 2\Delta_l, s_{l+1}, \ldots, s_n)$ is a point in which the $(k+1)$-st and l-th components of the point s'' have been swapped. (Similarly to Part 2.1.1 of this proof, the point z'' is a purely auxiliary point, it does not necessarily belong to the original box $\mathbf{x}_1 \times \ldots \times \mathbf{x}_n$.)

The chosen point s can be represented as a convex combination of the points s'' and z''. Indeed, since $k+1 < l$, then, due to Part 1 of this proof, we have $\overline{x}_{k+1} \leq \overline{x}_l$. Thus, $\overline{x}_{k+1} - 2\Delta_l \leq \overline{x}_{k+1} \leq \overline{x}_l$. This means, in turn, that there exists a real value $\alpha \in [0,1]$ for which $\overline{x}_{k+1}$ is a convex combination of $\overline{x}_{k+1} - 2\Delta_l$ and $\overline{x}_l$:

$$\overline{x}_{k+1} = \alpha \cdot (\overline{x}_{k+1} - 2\Delta_l) + (1-\alpha) \cdot \overline{x}_l.$$

Subtracting both sides of this inequality from

$$\overline{x}_{k+1} + \underline{x}_l = (\overline{x}_{k+1} - 2\Delta_l) + \overline{x}_l,$$

we conclude that

$$\underline{x}_l = \alpha \cdot \overline{x}_l + (1-\alpha) \cdot (\overline{x}_{k+1} - 2\Delta_l).$$

Since all other components of s, s'', and z'' are identical, we get

$$s = \alpha \cdot s'' + (1-\alpha) \cdot z''.$$

The function f is convex, hence $f(s) \leq \alpha \cdot f(s'') + (1-\alpha) \cdot f(z'')$. We already know that (due to symmetry) $f(z'') = f(s'')$, so $\alpha \cdot f(s'') + (1-\alpha) \cdot f(z'') = f(s'')$ and hence, $f(s) \leq f(s'')$.

By our selection of the point s, the value $f(s)$ is the largest value of f in the box: $f(s) = \max\limits_x f(x)$. Since s'' belongs to the same box, we thus have $f(s'') \leq f(s)$ hence $f(s'') = f(s) = \max\limits_x f(x)$, i.e., the function f indeed attains its maximum at the point s''.

$2.2.2^\circ$. Let us now prove that the maximum of f is also attained at an auxiliary point $s' = (\underline{x}_1, \ldots, \underline{x}_k, \underline{x}_{k+1}, s_{k+2}, \ldots)$. This will contradict to our assumption that k is the largest length of the starting sequence of lower bounds in an optimal vector x.

Similarly to the previous part of the proof, since $\Delta_{k+1} > \Delta_l$, we have $\underline{x}_{k+1} < \overline{x}_{k+1} - 2\Delta_l < \overline{x}_{k+1}$. Thus,

$$\overline{x}_{k+1} - 2\Delta_l = \beta \cdot \underline{x}_{k+1} + (1-\beta) \cdot \overline{x}_{k+1}$$

for some $\beta \in (0,1)$, and therefore, $s'' = \beta \cdot s^- + (1-\beta) \cdot s^+$, where

$$s^- \stackrel{\text{def}}{=} (\underline{x}_1, \ldots, \underline{x}_k, \underline{x}_{k+1}, s_{k+2}, \ldots, s_{l-1}, \underline{x}_l, s_{l+1}, \ldots, s_n);$$

$$s^+ \stackrel{\text{def}}{=} (\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, s_{k+2}, \ldots, s_{l-1}, \underline{x}_l, s_{l+1}, \ldots, s_n).$$

Since the function f is convex, we have $f(s'') \le \beta \cdot f(s^-) + (1-\beta) \cdot f(s^+)$.

Since the function f attains its maximum at the point s'', we have $f(x) \le f(s'')$ for all x from the box, in particular, we have $f(s^-) \le f(s'')$ and $f(s^+) \le f(s'')$. If we had $f(s^-) < f(s'')$, then, since $\beta > 0$, we would be able to conclude that $f(s'') \le \beta \cdot f(s^-) + (1-\beta) \cdot f(s^+) < f(s'')$ and $f(s'') < f(s'')$ – a contradiction. Thus, $f(s^-)$ must be equal to the maximum value $f(s'')$. We can thus take $s' = s^-$.

The point $s' = s^-$ has a starting sequence of lower bounds of length $\ge k+1$, which contradicts to our selection of the sequence s.

In both cases $\Delta_{k+1} \le \Delta_l$ and $\Delta_{k+1} > \Delta_l$, we get a contradiction. Thus, our assumption – that in the maximizing vector s, the longest sequence of lower bounds $\underline{x}_1, \ldots, \underline{x}_k$ can be followed eventually by one of the lower bounds $\underline{x}_l$, $l > k+1$ – is false. So, the optimizing sequence s with the longest possible starting sequence of lower bounds has the desired form $s = (\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$. Hence, to find the maximum, it is sufficient to test all $n+1$ such sequences.

The theorem is proven.

Computing $\overline{y}$ for easy-to-revise functions. For the variance V, the above algorithm requires n calls to computing V. Since computation of V takes linear time $O(n)$, this means that the above algorithm requires time $(n+1) \cdot O(n) + O(n \cdot \log(n)) = O(n^2)$.

In the case of variance, we can actually compute its upper bound faster, in time $O(n \cdot \log(n))$. This can be done if we take into consideration that we only need to compute variance for the points $s^{(k)} = (\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$ for $k = 0, 1, \ldots, n$. Computing $V(s^{(0)})$ requires linear time; when we go from $s^{(k)}$ to $s^{(k+1)}$, we only change a single component of the point s. If we keep the values $M \stackrel{\text{def}}{=} \frac{1}{n} \cdot \sum_{i=1}^{n} x_i^2$ and $E = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i$, then updating each of these two values requires a constant number $O(1)$ of arithmetic operations (independent on n), and then computing $V = M - E^2$ also requires a constant number of operations. Thus, overall, we need $O(n)$ time to compute $V(s^{(0)})$ and time $n \cdot O(1) = O(n)$ to compute n values $V(s^{(1)})$, ..., $V(s^{(n)})$. So, overall, we need time $O(n) + O(n) + O(n \cdot \log(n)) = O(n \cdot \log(n))$.

This idea can be applied to any situation in which the symmetric convex function $f(x_1, \ldots, x_n)$ is easy-to-revise in the following precise sense.

Definition 2 *We say that a function* $f(x_1, \ldots, x_n)$ *is* easy-to-revise *if there exist auxiliary functions* $f_1, \ldots, f_m$ *and the following two algorithms:*

- *an algorithm for computing* f *that first computes the values of the auxiliary functions* $f_1, \ldots, f_m$ *and then uses these values to compute the value of* f*, and*
- *an algorithm* Δf *that revises the values of each auxiliary function* f_j *when only one of the components* x_i *changes, and then computes the new value of* f *based on the updated values of* $f_1, \ldots, f_m$.

Of course, this definition makes practical sense only when a revision algorithm Δf is faster than the algorithm for computing f. For example, for the variance, computing requires time $O(n)$, but an update requires a much smaller time $O(1)$.

The auxiliary functions do not have to be different from f: e.g., the mean E is also easy-to-revise, with $f_1 = f$.

For such functions, we get the following result:

Theorem 2 *There exists an algorithm that computes the maximum* $\overline{y}$ *of a given symmetric convex easy-to-revise function* $f(x_1, \ldots, x_n)$ *over a given box* $\mathbf{x}_1 \times \ldots \times \mathbf{x}_n$ *for all the cases in which the intervals* $\mathbf{x}_i$ *satisfy the subset property; this algorithm:*

- *calls the function* f *once,*
- *calls the revision algorithm* Δf n *times, and*
- *uses* $O(n \cdot \log(n))$ *computational steps in addition to these calls.*

Proof. We have already proven, in Theorem 1, that the maximum of f is attained at one of the points $s^{(k)} = (\underline{x}_1, \ldots, \underline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$ for $k = 0, 1, \ldots, n$. So, to find the desired maximum $\overline{y}$, we first apply the algorithm f and compute the values of the auxiliary functions f_j and of f for $s^{(0)}$. Then, for each k from 1 to n, we apply the algorithm Δf to revise the values of f_j from $x = s^{(k-1)}$ to $x = s^{(k)}$, and compute $f(s^{(k)})$.

Thus, we use one call of f to compute $f(s^{(0)})$, n calls to Δf to compute n values $f(s^{(1)})$, ..., $f(s^{(n)})$, and $O(n \cdot \log(n))$ additional steps to sort the interval and to find the largest of the $n + 1$ values $f(s^{(k)})$. The theorem is proven.

3 Computing $\overline{y}$ under the constraint $\sum\limits_{i=1}^{n} x_i = c$

In this section, we solve the following problem: we know a symmetric convex function $f(x_1, \ldots, x_n)$ (given as an algorithm or, equivalently, as a computer program), we know the intervals $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$ that satisfy the above subset property, and we know the number c for which we should have $\sum_{i=1}^{n} x_i = c$. Our objective is to compute the value

$$\overline{y} = \max\left\{f(x_1, \ldots, x_n) : x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n, \sum_{i=1}^{n} x_i = c\right\}.$$

An example of such a problem is computing the maximum of negative entropy $-S$ under interval uncertainty about probabilities p_i; in this case, $c = 1$.

Similarly to the previous section, we can use the algorithms for solving this problem to compute the exact lower bound $\underline{y}$ of a symmetric concave function $f(x_1, \ldots, x_n)$ under the constraint $\sum x_i = c$. Namely, we first compute the exact upper bound $\overline{y}'$ of the function $f'(x_1, \ldots, x_n) = -f(x_1, \ldots, x_n)$ under the constraint $\sum x_i = c$, and then compute $\underline{y} = -\overline{y}'$. In particular, we can thus find the exact lower bound for the entropy function under interval uncertainty.

Theorem 3 *There exists an algorithm that computes the maximum $\overline{y}$ of a given symmetric convex function over a given box $\mathbf{x}_1 \times \ldots \times \mathbf{x}_n$ under the constraint $\sum_{i=1}^{n} x_i = c$, for all the cases in which the intervals $\mathbf{x}_i$ satisfy the subset property; this algorithm:*

- *calls the function f once, and*
- *uses $O(n)$ computational steps in addition to these calls.*

Comment. So, if the algorithm for computing the function f is feasible, i.e., requires a polynomial time $t \le P(n)$ for some polynomial $P(n)$, then computing $\overline{y}$ requires time $\le P(n) + O(n)$ – i.e., also a polynomial time. In particular, for Shannon's entropy, $P(n) = n$, so we can compute its exact lower bound in linear time.

Algorithm. The proposed algorithm is iterative. At each iteration of this algorithm, we have three sets:

- the set I^- of all the indices i from 1 to n for we already know that for the optimal vector x, we have $x_i = \underline{x}_i$;
- the set I^+ of all the indices j for which we already know that for the optimal vector x, we have $x_j = \overline{x}_j$;
- the set $I = \{1, \ldots, n\} - I^- - I^+$ of the indices i for which we are still undecided.

In the beginning, $I^- = I^+ = \emptyset$ and $I = \{1, \ldots, n\}$. At each iteration, we also update the values of two auxiliary quantities $E^- \stackrel{\text{def}}{=} \sum_{i \in I^-} \underline{x}_i$ and $E^+ \stackrel{\text{def}}{=} \sum_{j \in I^+} \overline{x}_j$. In principle, we could compute these values by computing these sums, but to speed up computations, on each iteration, we update these two auxiliary values in a way that is faster than re-computing the corresponding two sums. Initially, since $I^- = I^+ = \emptyset$, we take $E^- = E^+ = 0$.

At each iteration, we do the following:

- first, we compute the median m of the set I (median in terms of sorting by $\widetilde{x}_i = \dfrac{\underline{x}_i + \overline{x}_i}{2}$);
- then, by analyzing the elements of the undecided set I one by one, we divide them into two subsets $X^- = \{i : \widetilde{x}_i \le \widetilde{x}_m\}$ and $X^+ = \{j : \widetilde{x}_j > \widetilde{x}_m\}$;
- we compute $e^- = E^- + \sum\limits_{i \in X^-} \underline{x}_i$ and $e^+ = E^+ + \sum\limits_{i \in X^+} \overline{x}_i$;
- if $e^- + e^+ > c$, then we replace I^- with $I^- \cup X^-$, E^- with e^-, and I with X^+;
- if $e^- + e^+ + 2\Delta_m < c$, then we replace I^+ with $I^+ \cup X^+$, E^+ with e^+, and I with X^-;
- finally, if $e^- + e^+ \le c \le e^- + e^+ + 2\Delta_m$, then we replace I^- with $I^- \cup (X^- - \{m\})$, I^+ with $I^+ \cup X^+$, I with $\{m\}$, E^- with $e^- - \underline{p}_m$, and E^+ with e^+.

At each iteration, the set of undecided indices is divided in half. Iterations continue until we have only one undecided index $I = \{k\}$, after which we return, as $\overline{y}$, the value of the function $f(x_1, \ldots, x_n)$ for the vector x for which $x_i = \underline{x}_i$ for $i \in I^-$, $x_j = \overline{x}_j$ for $j \in I^+$, and $x_k = c - E^- - E^+$ for the remaining value k.

Comment. If some intervals $\mathbf{x}_i$ are degenerate, i.e., $\mathbf{x}_i = [x_i, x_i]$, then we need the following modifications to the above algorithm:

- first, as the initial set I, we take the set of all indices corresponding to non-degenerate intervals;
- second, we pre-compute the sum e of all the exactly known values x_i (corresponding to degenerate intervals);
- third, on each iteration, instead of comparing c with the sum $e^- + e^+$, we compare c with the sum $e^- + e^+ + e$.

Computational complexity. At each iteration, computing median requires linear time; see, e.g., [3]. All other operations with I require time t linear in the number of elements $|I|$ of I: $t \le C \cdot |I|$ for some C. We start with the set I of size n; on the next iteration, we have a set of size $n/2$, then $n/4$, etc. Thus, the overall computation time is $\le C \cdot (n + n/2 + n/4 + \ldots) \le C \cdot 2n$, i.e., linear in n.

Proof. Let us prove that this algorithm indeed computes $\overline{y}$.

1°. Due to the subset property, as we have shown in the proof of Theorem 1, we can sort the intervals $[\underline{x}_i, \overline{x}_i]$ in lexicographic order, in which case their lower endpoints $\underline{x}_i$, their upper endpoints $\overline{x}_i$, and their midpoints $\widetilde{x}_i$ are also sorted: $\underline{x}_i \le \underline{x}_{i+1}$, $\overline{x}_i \le \overline{x}_{i+1}$, and $\widetilde{x}_i \le \widetilde{x}_{i+1}$. Let us thus assume that the intervals are thus sorted.

Since f is a continuous function on a bounded closed set

$$(\mathbf{x}_1 \times \ldots \times \mathbf{x}_n) \cap \left\{ x : \sum_{i=1}^{n} x_i = c \right\}, \tag{1}$$

its maximum is always attained at some point $x = (x_1, \ldots, x_n)$. Since f is a convex function on a convex polyhedron (1), the maximum of the function f is attained at one of the vertices of this polyhedron. Points on the polyhedron are determined by the equality $\sum\limits_{i=1}^{n} x_i = c$ and by $2n$ inequalities $\underline{x}_i \le x_i$ and $x_i \le \overline{x}_i$, $i = 1, 2, \ldots, n$. A vertex is where n of the original equalities are inequalities are equalities. Since we already have one original equality, this means that we must have $n-1$ inequalities turn to equalities, i.e., that for at least $n-1$ values x_i, we must have $x_i = \underline{x}_i$ or $x_i = \overline{x}_i$.

So, the maximum $\overline{y}$ is attained when for $n-1$ out of n variables x_i, we have either $x_i = \underline{x}_i$ or $x_i = \overline{x}_i$.

2°. To prove that the algorithm is correct, we must prove that the maximum of the function f is attained at one of the points $x = (\underline{x}_1, \ldots, \underline{x}_{k-1}, x_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$; here, the value $x_k \in [\underline{x}_k, \overline{x}_k]$ can be uniquely determined from the condition $\sum\limits_{i=1}^{n} x_i = c$, as $c - \sum\limits_{i=1}^{k-1} \underline{x}_i - \sum\limits_{i=k+1}^{n} \overline{x}_i$.

Indeed, we have already proven that the maximum is attained at one of the vectors $x = (x_1, \ldots, x_n)$ at which $\ge n-1$ values x_i take endpoint values $\underline{x}_i$ and $\overline{x}_i$.

There may be several such vertices x at which the maximum is attained. Out of all such maximizing sequences, we choose a one with the largest length of the starting sequence of lower bounds. Let us denote this largest length by $k-1$.

If there are several such sequences, with the same (largest) length of the starting sequence of lower bounds, then for each of these sequences x, we can find the largest value l for which $x_l \ne \overline{x}_l$. We will then select the sequence for which this index l is the smallest possible. In other words, we have $s_i = \overline{x}_i$ for all $i > l$.

Let us denote the selected sequence by

$$s = (\underline{x}_1, \ldots, \underline{x}_{k-1}, s_k, \ldots, s_l, \overline{x}_{l+1}, \ldots, \overline{x}_n).$$

We will prove, by reduction to a contradiction, that this sequence s has the desired form $s = (\underline{x}_1, \ldots, \underline{x}_{k-1}, x_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$.

Indeed, let us assume that the selected sequence s does not have this form. This means that $s_t \neq \overline{x}_t$ for some $t \geq k+1$. We have already denoted the largest index with this property by l. This means that $s_l \neq \overline{s}_l$ for this value l, and $s_u = \overline{x}_u$ for all $u > l$.

Since $s_l \in [\underline{x}_l, \overline{x}_l]$, we conclude that $s_l < \overline{s}_l$.

According to our choice of the sequence $s = (\underline{x}_1, \ldots, \underline{x}_{k-1}, s_k, \ldots)$, the value $k-1$ is the largest possible length of the starting sequence of lower bounds. This means that the k-th element s_k of this sequence is not a lower bound: $s_k \neq \underline{x}_k$. Since $s_k \in \mathbf{x}_k$, we thus have $s_k > \underline{x}_k$.

Similarly to the proof of Theorem 1, we can conclude that since $f(x_1, \ldots, x_n)$ is a symmetric convex function and s is maximizing sequence, with $f(s) = \overline{y}$, we get $f(s') = \overline{y}$ for

$$s' = (\underline{x}_1, \ldots, \underline{x}_{k-1}, s_k - \Delta, s_{k+1}, \ldots, s_{l-1}, s_l + \Delta, \overline{x}_{l+1}, \ldots, \overline{x}_n)$$

for every $\Delta > 0$ for which $s_k - \Delta \geq \underline{x}_k$ and $s_l + \Delta \leq \overline{x}_l$.

We will construct the vector s' corresponding to the largest possible Δ. The largest value Δ for which the first inequality holds is $s_k - \underline{x}_k$, the larger value Δ for which the second inequality holds is $\overline{x}_l - s_l$. Thus, the largest possible value Δ for which both inequalities hold is $\Delta = \min(s_k - \underline{x}_k, \overline{x}_l - s_l)$. Let us consider two possible cases.

If $s_k - \underline{x}_k \leq \overline{x}_l - s_l$, then we take $\Delta = s_k - \underline{x}_k$, and conclude that $f(s') = \overline{y}$, where

$$s' = (\underline{x}_1, \ldots, \underline{x}_{k-1}, \underline{x}_k, s_{k+1}, \ldots, s_{l-1}, s_l + \Delta, \overline{x}_{l+1}, \ldots, \overline{x}_n).$$

In other words, we would then get a maximizing sequence s' that starts with k lower bounds. This would contradict to the our selection of $k-1$ as the largest length of the starting sequence of lower bounds in a maximizing sequence.

If $s_k - \underline{x}_k > \overline{x}_l - s_l$, then we take $\Delta = \overline{x}_l - s_l$ and conclude that $f(s') = \overline{y}$ for

$$s' = (\underline{x}_1, \ldots, \underline{x}_{k-1}, s_k - \Delta, s_{k+1}, \ldots, s_{l-1}, \overline{x}_l, \overline{x}_{l+1}, \ldots, \overline{x}_n).$$

In this maximizing sequence, the largest non-upper bound is $< l$ – which contradicts to our selection of s as the sequence for which the index of the largest non-upper bound is the smallest possible.

So, we get a contradiction in both cases. This proves that the selected sequence s indeed has the desired form.

3°. For the resulting vector $x = (\underline{x}_1, \ldots, \underline{x}_{k-1}, x_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)$, with $\underline{x}_k \leq x_k \leq \overline{x}_k$, the condition $\sum\limits_{i=1}^{n} x_i = c$ implies that $\Sigma_k \leq c \leq \Sigma_{k-1}$, where $\Sigma_k \stackrel{\text{def}}{=} \sum\limits_{i=1}^{k} \underline{x}_i + \sum\limits_{j=k+1}^{n} \overline{x}_j$. When we go from Σ_k to Σ_{k+1}, we replace a larger value $\overline{x}_{k+1}$ with a smaller value $\underline{x}_{k+1}$, hence $\Sigma_k > \Sigma_{k+1}$. Thus, there has to be exactly one $k_{\max}$ for which $\Sigma_k \leq c \leq \Sigma_{k-1}$.

So, if we have $\Sigma_m > c$, this means that the value $k_{\max}$ corresponding to the maximum of f is $> m$; hence for all the indices $\leq m$, we already know that in the optimal vector x, $x_i = \underline{x}_i$. Thus, these indices can be added to the set I^-.

If Σ_{m-1} $(= \Sigma_m + 2\Delta_m) < c$, this means that the value $k_{\min}$ corresponding to the maximum of f is $< m$; hence for all the indices $\geq m$, we already know that in the optimal vector x, $x_j = \overline{x}_j$. Thus, these indices can be added to the set I^+.

Finally, if $\Sigma_m \leq c \leq \Sigma_{m-1}$, then this m is where the maximum of the function f is attained.

The algorithm has been justified.

Acknowledgments. The author is thankful to Martine Ceberio, Vladik Kreinovich, and David Novick for their encouragement and invaluable editorial help.

References

[1] Berberian, S. (1974). Lectures in functional analysis and operator theory, Springer, New York-Heidelberg-Berlin.

[2] Boyd, S., and Vandenberghe, L. (2004). Convex Optimization, Cambridge University Press.

[3] Cormen, T.H., Leiserson, C.E., Rivest, R.L., and Stein, S. (2001). Introduction to Algorithms, MIT Press, Cambridge, Massachusetts.

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

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

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

[7] Klir, G., Xiang, G., and Kosheleva, O. (2006). Estimating Information Amount under Interval Uncertainty: Algorithmic Solvability and Computational Complexity, unpublished report, Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMU'06, Paris, France, July 2–7, 2006, pp. 840–847.

[8] Kreinovich, V., and Longpre, L. (2003). Computational complexity and feasibility of data processing and interval computations, with extension to cases when we have partial information about probabilities, In: V. Brattka, M. Schroeder, K. Weihrauch, and N. Zhong, editors, Proc. Conf. on Computability and Complexity in Analysis CCA'2003, Cincinnati, Ohio, USA, August 28–30, 2003, pp. 19–54.

[9] Kreinovich, V., Xiang, G., Starks, S. A., Longpré, L., Ceberio, M., Araiza, R., Beck, J., Kandathi, R., Nayak, A., Torres, R., and Hajagos, J. (2006). 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.

[10] Nivlet, P., Fournier, F., and Royer, J. (2001a). A new methodology to account for uncertainties in 4-D seismic interpretation, Proc. 71st Annual Int'l Meeting of Soc.of Exploratory Geophysics SEG'2001, San Antonio, TX, September 9–14, 2001, pp. 1644–1647.

[11] Nivlet, P., Fournier, F., and Royer, J. (2001b). Propagating interval uncertainties in supervised pattern recognition for reservoir characterization, Proc. 2001 Society of Petroleum Engineers Annual Conf. SPE'2001, New Orleans, LA, September 30-October 3, paper SPE-71327.

[12] Rabinovich, S. (2005). Measurement Errors and Uncertainties: Theory and Practice, Springer-Verlag, New York, 2005.

[13] Roberts, A.W., and Varberg, D.E. (1973). Convex functions, Academic Press, New York and London.

[14] Shmulevich, I., and Zhang, W. (2002). Binary analysis and optimization-based normalization of gene expression data, Bioinformatics, Vol. 18, No. 4, pp. 555–565.

[15] Vavasis, S.A. (1991). Nonlinear Optimization: Complexity Issues. Oxford Science, New York, 1991.

[16] Webster, R. (1994). Convexity, Oxford University Press, Oxford, New York, Tokyo.

[17] Zhang, W., Shmulevich, I., and Astola, J. (2004). Microarray Quality Control, Wiley, Hoboken, New Jersey.