Estimating Correlation under Interval Uncertainty

Ali Jalal-Kamali and Vladik Kreinovich
Department of Computer Science
University of Texas at El Paso
500 W. University
El Paso, TX 79968, USA
ajalalkamali@miners.utep.edu
vladik@utep.edu

Abstract

In many engineering situations, we are interested in finding the correlation ρ between different quantities x and y based on the values x_i and y_i of these quantities measured in different situations i. Measurements are never absolutely accurate; it is therefore necessary to take this inaccuracy into account when estimating the correlation ρ. Sometimes, we know the probabilities of different values of measurement errors, but in many cases, we only know the upper bounds Δ_{xi} and Δ_{yi} on the corresponding measurement errors. In such situations, after we get the measurement results $\widetilde{x}_i$ and $\widetilde{y}_i$, the only information that we have about the actual (unknown) values x_i and y_i is that they belong to the corresponding intervals $[\widetilde{x}_i - \Delta_{xi}, \widetilde{x}_i + \Delta_{xi}]$ and $[\widetilde{y}_i - \Delta_{yi}, \widetilde{y}_i + \Delta_{yi}]$. Different values from these intervals lead, in general, to different values of the correlation ρ. It is therefore desirable to find the range $\left[\underline{\rho}, \overline{\rho}\right]$ of possible values of the correlation when x_i and y_i take values from the corresponding intervals. In general, the problem of computing this range is NP-hard. In this paper, we provide a feasible (= polynomial-time) algorithm for computing at least one of the endpoints of this interval: for computing $\overline{\rho}$ when $\overline{\rho} > 0$ and for computing $\underline{\rho}$ when $\underline{\rho} < 0$.
Keywords: imprecise probabilities, correlation, interval uncertainty

1 Introduction

Need for correlation. In engineering, we design systems for real-world applications. To make sure that the system functions correctly, we need to take into account all possible situations in which these systems will function. Each such situation can be characterized by the values of different quantities. To describe which combinations of these values are more probable and which are less

probable, it is necessary to know which quantities are independent and which are correlated – positively or negatively.

To estimate the correlation between the quantities x and y, we repeatedly measure the values x_i and y_i of both quantities in different situations i. The correlation ρ is then estimated as the ratio

$$\rho = \frac{C}{\sigma_x \cdot \sigma_y},$$

of the covariance C to the product of standard deviations $\sigma_x = \sqrt{V_x}$ and $\sigma_y = \sqrt{V_y}$. Covariance and standard deviations, in their turn, are defined as follows:

$$C = \frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - E_x) \cdot (y_i - E_y) = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i \cdot y_i - E_x \cdot E_y,$$

$$V_x = \frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - E_x)^2, \quad V_y = \frac{1}{n} \cdot \sum_{i=1}^{n} (y_i - E_y)^2,$$

and the means E_x and E_y are estimates as follows:

$$E_x = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i, \quad E_y = \frac{1}{n} \cdot \sum_{i=1}^{n} y_i.$$

Comment. In the above formulas, we use the estimates for C, V_x, and V_y which are known to be biased. Usually, correlation is defined by using unbiased definitions

$$C = \frac{1}{n-1} \cdot \sum_{i=1}^{n} (x_i - E_x) \cdot (y_i - E_y) = \frac{1}{n-1} \cdot \sum_{i=1}^{n} x_i \cdot y_i - E_x \cdot E_y,$$

$$V_x = \frac{1}{n-1} \cdot \sum_{i=1}^{n} (x_i - E_x)^2, \quad V_y = \frac{1}{n-1} \cdot \sum_{i=1}^{n} (y_i - E_y)^2.$$

One can easily check that the resulting expression for ρ is the same whether we use biased or unbiased estimates; we use biased estimates because they make the computations slightly simpler.

Known facts about correlation: brief reminder. It is known that the value of this correlation coefficient ρ is always between -1 and 1. The correlation is equal to 1 if and only if the values are positively linearly dependent, i.e., when for some coefficient $k_x > 0$, we have $y_i = E_y + k_x \cdot (x_i - E_x)$ for every i. The correlation is equal to -1 if and only if the values are negatively linearly dependent, i.e., when for some coefficient $k_x < 0$, we have $y_i = E_y + k_x \cdot (x_i - E_x)$ for every i.

Need to take into account interval uncertainty. The values x_i and y_i used to estimate correlation come from measurements, and measurements are never absolutely accurate: the measurement results $\widetilde{x}_i$ and $\widetilde{y}_i$ are, in general, different from the actual (unknown) values x_i and y_i of the corresponding quantities. As a result, the value $\widetilde{\rho}$ estimated based on these measurement results is, in general, different from the ideal value ρ which we would get if we could use the actual values x_i and y_i. It is therefore desirable to determine how accurate is the resulting estimate.

Sometimes, we know the probabilities of different values of measurement errors $\widetilde{x}_i - x_i$ and $\widetilde{y}_i - y_i$. However, in many cases, we do not know these probabilities, we only know the upper bounds Δ_{xi} and Δ_{yi} on the corresponding measurement errors: $|\widetilde{x}_i - x_i| \le \Delta_{xi}$ and $|\widetilde{y}_i - y_i| \le \Delta_{yi}$; see, e.g., [15]. In this case, the only information that we have about the actual values x_i and y_i is that they belong to the corresponding intervals $[\underline{x}_i, \overline{x}_i] = [\widetilde{x}_i - \Delta_{xi}, \widetilde{x}_i + \Delta_{xi}]$ and $\left[\underline{y}_i, \overline{y}_i\right] = [\widetilde{y}_i - \Delta_{yi}, \widetilde{y}_i + \Delta_{yi}]$. Different values $x_i \in [\underline{x}_i, \overline{x}_i]$ and $y_i \in \left[\underline{y}_i, \overline{y}_i\right]$ lead, in general, to different values of the covariance. It is therefore desirable to find the range of all possible values of the covariance ρ:

$$\left[\underline{\rho}, \overline{\rho}\right] = \left\{\rho(x_1, \ldots, x_n, y_1, \ldots, y_n) : x_i \in [\underline{x}_i, \overline{x}_i], y_i \in \left[\underline{y}_i, \overline{y}_i\right]\right\}.$$

The problem of computing the range of correlation under interval uncertainty is a particular case of the general problem of *interval computations* (see, e.g., [8, 12]): computing the range of a given function $f(x_1, \ldots, x_n)$ under the interval uncertainty $x_1 \in [\underline{x}_1, \overline{x}_1], \ldots, x_n \in [\underline{x}_n, \overline{x}_n]$. Interval computations – in particular, interval computations of statistical characteristics – have many applications, in particular, engineering applications; see, e.g., [2, 7, 8, 10, 11, 12, 13, 14, 16].

For example, if we perform a statistical analysis of the measurement results, then, for each statistical characteristic $S(x_1, \ldots, x_n)$, we need to find its range

$$\mathbf{S} = \{S(x_1, \ldots, x_n) : x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n\}.$$

For the mean E_x, the situation is simple: the mean is an increasing function of all its variables. So, its smallest value $\underline{E}_x$ is attained when each of the variables x_i attains its smallest value $\underline{x}_i$, and its largest value $\overline{E}_x$ is attained when each of the variables attains its largest value $\overline{x}_i$:

$$\underline{E}_x = \frac{1}{n} \cdot \sum_{i=1}^{n} \underline{x}_i, \quad \overline{E}_x = \frac{1}{n} \cdot \sum_{i=1}^{n} \overline{x}_i.$$

Estimating correlation under interval uncertainty is NP-hard. In contrast to the mean – which is always monotonic – variance, covariance, and correlation are sometimes non-monotonic. It turns out that, in general, computing the values of these characteristics under interval uncertainty is NP-hard [3, 4, 13, 14]. This means, crudely speaking, that unless P=NP (which most computable scientists believe to be wrong), no feasible (i.e., no polynomial-time)

algorithm is possible that would always compute the range of the corresponding characteristic under interval uncertainty.

The problem of estimating correlation under interval uncertainty is formulated and analyzed in [16]; in that paper, this problem is formulated and solved as an optimization problem. For reasonably small n, the corresponding optimization algorithms work well [16]. However, since the problem is NP-hard, the computation time becomes infeasible when n is large.

What we do in this paper. We show that while we cannot have an efficient algorithm for computing both bounds $\underline{\rho}$ and $\overline{\rho}$, we can effectively compute (at least) one of the bounds. Specifically, we show that we can compute $\overline{\rho}$ when $\overline{\rho} > 0$ and we can compute $\underline{\rho}$ when $\underline{\rho} < 0$. This means that, in the case of a non-degenerate interval $\left[\underline{\rho}, \overline{\rho}\right]$ (i.e., $\underline{\rho} < \overline{\rho}$):

- when $\overline{\rho} \le 0$, we compute the lower endpoint $\underline{\rho}$;
- when $0 \le \underline{\rho}$, we compute the upper endpoint $\overline{\rho}$;
- in all remaining cases, when $\underline{\rho} < 0 < \overline{\rho}$, we compute both lower endpoint $\underline{\rho}$ and $\overline{\rho}$.

2 Main Result and the Corresponding Algorithm

Main result. *There exists a polynomial-time algorithm that, given n pairs of intervals $[\underline{x}_i, \overline{x}_i]$ and $\left[\underline{y}_i, \overline{y}_i\right]$, computes (at least) one of the endpoints of the interval $\left[\underline{\rho}, \overline{\rho}\right]$ of possible values of the correlation ρ:*

- *it computes $\overline{\rho}$ if $\overline{\rho} > 0$, and*
- *it computes $\underline{\rho}$ if $\underline{\rho} < 0$.*

Reducing minimum to maximum. When we change the sign of y_i, the correlation changes sign as well:

$$\rho(x_1, \ldots, x_n, -y_1, \ldots, -y_n) = -\rho(x_1, \ldots, x_n, y_1, \ldots, y_n).$$

Since the function $z \to -z$ is decreasing, its smallest value is attained when z is the largest, and its largest value is attained when z is the smallest. Thus, if z goes from $\underline{z}$ to $\overline{z}$, the range of $-z$ is $[-\overline{z}, -\underline{z}]$. So, for the endpoints of the ranges, we get

$$\overline{\rho}\left([\underline{x}_1, \overline{x}_1], \ldots, [\underline{x}_n, \overline{x}_n], -\left[\underline{y}_1, \overline{y}_1\right], \ldots, -\left[\underline{y}_n, \overline{y}_n\right]\right) =$$

$$-\underline{\rho}\left([\underline{x}_1, \overline{x}_1], \ldots, [\underline{x}_n, \overline{x}_n], \left[\underline{y}_1, \overline{y}_1\right], \ldots, \left[\underline{y}_n, \overline{y}_n\right]\right),$$

where

$$-\left[\underline{y}_i, \overline{y}_i\right] = \left\{-y_i : y_i \in \left[\underline{y}_i, \overline{y}_i\right]\right\} = \left[-\overline{y}_i, -\underline{y}_i\right].$$

So, if we know how to compute the largest value $\overline{\rho}$ when this value is positive, we can then compute the smallest value $\underline{\rho}$ when this value is negative, as

$$\underline{\rho}\left([\underline{x}_1,\overline{x}_1],\ldots,[\underline{x}_n,\overline{x}_n],\left[\underline{y}_1,\overline{y}_1\right],\ldots,\left[\underline{y}_n,\overline{y}_n\right]\right)=$$

$$-\overline{\rho}\left([\underline{x}_1,\overline{x}_1],\ldots,[\underline{x}_n,\overline{x}_n],\left[-\overline{y}_1,-\underline{y}_1\right],\ldots,\left[-\overline{y}_n,-\underline{y}_n\right]\right).$$

Because of this reduction, in the following text, we will concentrate on computing the largest value $\overline{\rho}$.

Algorithm: preliminary definitions. To describe the algorithm, we need to introduce an auxiliary notion of a 4-tuple. This notion is related to the vertices of the input boxes $[\underline{x}_i,\overline{x}_i]\times\left[\underline{y}_i,\overline{y}_i\right]$.

For each i from 1 to n, the corresponding box $[\underline{x}_i,\overline{x}_i]\times\left[\underline{y}_i,\overline{y}_i\right]$ has four vertices: $\left(\underline{x}_i,\underline{y}_i\right)$, $(\underline{x}_i,\overline{y}_i)$, $\left(\overline{x}_i,\underline{y}_i\right)$, and $(\overline{x}_i,\overline{y}_i)$. So, totally, we have $4n$ vertices.

In our algorithm, we will combine these vertices with "signs". A sign is defined as one of the three possible symbols: $+$, $-$, and 0. A *4-tuple* is then defined as a tuple consisting of two vertices and two signs. In this algorithm, we will consider all possible 4-tuples.

Out of $4n$ vertices, we can form $\dfrac{4n\cdot(4n-1)}{2}$ pairs; there are $9=3^2$ possible combinations of signs: $(-,-)$, $(-,0)$, $(-,+)$, $(0,-)$, $(0,0)$, $(0,+)$, $(+,-)$, $(+,0)$, and $(+,+)$. So, totally, we will consider $9\cdot\dfrac{4n\cdot(4n-1)}{2}$ different 4-tuples.

Algorithm: general structure. Our algorithm consists of three stages:

- on the first (preliminary) stage, we consider all 4-tuples one by one, and perform computations on each of these 4-tuples;
- on the second (main) stage, we consider, one by one, all possible *pairs* of 4-tuples, and perform computations on each of these pairs; for some of these pairs, we compute a correlation value;
- on the third stage, we compute the largest of all the correlation values computed on the second stage, and return this largest correlation value as $\overline{\rho}$.

These stages, in their turn, consist of steps. Let us describe these stages and corresponding steps one by one.

Algorithm: first stage. As we have mentioned, on the first (preliminary) stage, we consider all 4-tuples one by one, and perform computations on each of these 4-tuples.

By definition, a 4-tuple consists of two vertices and two signs. For this 4-tuple, we perform the following three steps:

- on the first step, we create a point which is close to the first vertex;
- on the second step, we create a point which is close to the second vertex;
- in the third operation, we form a straight line.

Let us describe these three steps one by one.

First step. On the first step, if the first sign is 0, we select the first vertex itself as the desired point-close-to-the-first-vertex.

If the first sign is not 0, we create the desired point by moving the first vertex "slightly" along the x axis in the direction determined by the first sign, i.e.:

- slightly increase x if the sign is $+$ and
- slightly decrease x if the sign is $-$.

Here, "slightly" means that the change is smaller than the smallest difference between distinct values x_i and y_i.

Second step. On the second step, if the first sign is 0, we select the second vertex itself as the desired point-close-to-the-second-vertex.

If the second sign is not 0, we create the desired point by moving the second vertex "slightly" along the x axis in the direction determined by the second sign, i.e.:

- slightly increase x if the sign is $+$ and
- slightly decrease x if the sign is $-$.

Here, "slightly" means the same as for the first operation.

Third step. As a result of performing the first two steps, we have created two points on the (x, y) plane:

- a point close to the first vertex, and
- a point close to the second vertex.

On the third step, we form a straight line going through these two created points.

Comment. The purpose of the small shifts is to make sure that the resulting lines represent all possible location of the vertices with respect to the two vertices:

- the line corresponding to $(0, 0)$ passes through the two vertices;
- the line corresponding to $(+, +)$ passes to the right of both vertices;

- the line corresponding to $(-,+)$ passes to the left of the first vertex and to the right of the second vertex, etc.

This is necessary to make sure that we consider all possible locations of the corresponding lines.

Algorithm: second stage. As we mentioned, on the second stage, we consider, one by one, all possible *pairs* of 4-tuples, and perform computations on each of these pairs.

Let us consider a pair of 4-tuples. For each of these 4-tuples, on the first stage (to be more precise, on the third step of the first stage), we constructed a straight line. In the description of the algorithm,

- the line constructed based on the first 4-tuple will be called the *representative x-line*, and
- the line constructed based on the second 4-tuple will be called the *representative y-line*.

If the representative x-line is vertical and/or the representative y-line is horizontal, we dismiss the corresponding pair of 4-tuples. (The justification for this dismissal is given in the proof.)

Comment. These lines are called "representative" because they will serve as "representatives" on the "actual" x- and y-lines that we will be using – representatives in the following sense:

- each actual x-line has the same relation to each of the n boxes as the corresponding representative x-line, and
- each actual y-line has the same relation to each of the n boxes as the corresponding representative y-line.

0-th step of the second stage. Once we have computed the representative x-line and the representative y-line, we first check whether these two lines coincide. Let us consider the two possible results of this checking one by one.

What we do when the representative lines coincide. If the representative x-line and the representative y-line coincide, we check whether this representative x-line (which, in this case, is the same line as the representative y-line) intersects with each of the n boxes.

- If the representative x-lines *does not intersect* with at least one of the n input boxes, then we end the analysis of this pair of 4-tuples without returning any correlation value; after that:
  - if *have not yet completed* the analysis of all pairs of 4-tuples, then we start analyzing the next tuple;

- if we *have* already *completed* the analysis of all pairs of 4-tuples, the we move on to the third stage of our algorithm.

- If the representative x-line *intersects* with each of n input boxes, then we stop the algorithm and return the value $\overline{\rho} = 1$.

Comment. In the last case, when we compute the correlation value equal to 1, this will clearly be the largest possible value of the correlation – since correlation cannot be larger than 1. Thus, if we know that one of the correlations is 1, there is no need to continue our second-stage analysis and then, on the third stage, compute the largest of the correlation values – we already know that this largest correlation value is 1.

What we do when the representative lines differ: terminology. Let us now consider the case when the representative x-line is different from the representative y-line.

When we constructed the representative lines, we dismissed the cases when the representative x-line is vertical and/or when the representative y-line is horizontal. In all remaining cases, the representative x-line divides the plane into two semi-planes:

- the points *above* this line, i.e., the points (x, y) for which the y coordinate is larger than the y-value of the point on the x-line with the same x coordinate, and

- the points *below* this line, i.e., the points (x, y) for which the y coordinate is smaller than the y-value of the point on the x-line with the same x coordinate.

The representative y-line similarly divides the plane into two semi-planes:

- the points to the *right* of this line, i.e., the points (x, y) for which the x coordinate is larger than the x-value of the point on the x-line with the same y coordinate, and

- the points to the *left* of this line, i.e., the points (x, y) for which the x coordinate is smaller than the x-value of the point on the y-line with the same y coordinate.

What we do when the representative lines differ: notations. For each pair of the 4-tuples for which the representative x-line differs from the representative y-line, we will use four to-be-determined parameters E_x, E_y, k_x, and k_y. The meaning of this parameters is related to the optimal values $x_1, \ldots, x_n, y_1, \ldots, y_n$, i.e., to the values x_i and y_i for which the correlation attains its largest possible value:

- the parameter E_x is the average of the optimal values x_i;

- the parameter E_y is the average of the optimal values y_i;
- the parameter k_x is the ratio $\dfrac{C}{V_x}$, where C is the correlation of the values x_i and y_i and V_x is the sample variance of the values x_i; and
- the parameter k_y is the ratio $\dfrac{C}{V_y}$, where C is the correlation of the values x_i and y_i and V_y is the sample variance of the values y_i.

In precise terms:

$$E_x = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i; \quad E_y = \frac{1}{n} \cdot \sum_{i=1}^{n} y_i;$$

$$\frac{1}{n} \cdot \sum_{i=1}^{n} x_i \cdot y_i - E_x \cdot E_y = k_x \cdot \left(\frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - E_x)^2 \right);$$

$$\frac{1}{n} \cdot \sum_{i=1}^{n} x_i \cdot y_i - E_x \cdot E_y = k_y \cdot \left(\frac{1}{n} \cdot \sum_{i=1}^{n} (y_i - E_y)^2 \right).$$

What we do when the representative lines differ: general procedure. The corresponding procedure consists of the following steps:

- on the first step, under the assumption that the "actual" x- and y-lines are in the same relation with all n boxes as the representative x- and y-lines, we form the expressions for the optimal points x_i and y_i in terms of the parameters E_x, E_y, k_x, and k_y;
- on the second step, we use these expressions to form equations for determining the parameters E_x, E_y, k_x, and k_y, and use these equations to determine the values of these four parameters; in general, we may have several possible solution vectors (E_x, E_y, k_x, k_y);
- finally, on the third step, for each solution vector, we substitute the computed values of these parameters and check whether the resulting actual x- and y-lines are indeed in the same relation to all the boxes as the corresponding representative x- and y-lines:
  - if they are in the *same relation*, we compute the correlation between the values x_i and y_i add this correlation to the list of correlation values (to be considered on the third stage);
  - if they are *not in the same relation*, do not compute any correlation, and simply move on to the next solution.

Once we have exhausted all the solutions, we either go to the next pair of 4-tuples or – if we have already exhausted all the pairs of 4-tuples – to the third stage of the algorithm.

Let us describe these three steps in detail.

First step. On the first step, we start by forming the *actual* lines as follows:

- the actual x-line has the form $y = E_y + k_x \cdot (x - E_x)$, and
- the actual y-line has the form $x = E_x + k_y \cdot (y - E_y)$,

where E_x, E_y, k_x, and k_y are to-be-determined real numbers.

The values of these numbers will be determined later; these values will be selected in such a way that:

- the actual x-line has the same relation to each of the n boxes as the representative x-line, and
- the actual y-line has the same relation to each of the n boxes as the corresponding representative y-line.

Once the actual lines have been determined, then for each box $[\underline{x}_i, \overline{x}_i] \times \left[\underline{y}_i, \overline{y}_i\right]$, based on the location of this box in comparison to the representative x- and y-lines, we select the values x_i and y_i as follows:

- If the whole box is above the representative x-line, we take $x_i = \overline{x}_i$. On the resulting segment $\{\overline{x}_i\} \times \left[\underline{y}_i, \overline{y}_i\right]$, we select the point which is the closest to the actual y-line; namely:
  - if the whole segment is to the right of the representative y-line, we select $y_i = \underline{y}_i$;
  - if the whole segment is to left of the representative y-line, we select $y_i = \overline{y}_i$;
  - if the segment intersects with the representative y-line, we select the value y_i corresponding to the intersection point between the segment and the actual y-line.
- If the whole box is below the representative x-line, we take $x_i = \underline{x}_i$. On the resulting segment $\{\underline{x}_i\} \times \left[\underline{y}_i, \overline{y}_i\right]$, we select the point which is the closest to the actual y-line; namely:
  - if the whole segment is to the right of the representative y-line, we select $y_i = \underline{y}_i$;
  - if the whole segment is to left of the representative y-line, we select $y_i = \overline{y}_i$;
  - if the segment intersects with the representative y-line, we select the value y_i corresponding to the intersection point between the segment and the actual y-line.
- If the whole box is to the right of the representative y-line, we take $y_i = \underline{y}_i$. On the resulting segment $[\underline{x}_i, \overline{x}_i] \times \left\{\underline{y}_i\right\}$, we select the point which is the closest to the actual x-line; namely:

- if the whole segment is above the representative x-line, we select $x_i = \underline{x}_i$;
- if the whole segment is below the representative x-line, we select $x_i = \overline{x}_i$;
- if the segment intersects with the representative x-line, we select the value x_i corresponding to the intersection point between this segment and the actual x-line.

- If the whole box is to the left of the representative y-line, we take $y_i = \overline{y}_i$. On the resulting segment $[\underline{x}_i, \overline{x}_i] \times \{\overline{y}_i\}$, we select the point which is the closest to the actual x-line; namely:
  - if the whole segment is above the representative x-line, we select $x_i = \underline{x}_i$;
  - if the whole segment is below the representative x-line, we select $x_i = \overline{x}_i$;
  - if the segment intersects with the representative x-line, we select the value x_i corresponding to the intersection point between the segment and the actual x-line.

- The only remaining case is when the box contains the intersection point (E_x, E_y) of the actual x- and y-lines.

Comment. For each pair of lines, for each i, according to our algorithm, as the appropriate value of x_i, we make one of the following four selections:

- sometimes, we select a known value $\underline{x}_i$;
- sometimes, we select a know value $\overline{x}_i$;
- sometimes, we select the value $x_i = E_x$ (which is not a priori known, it is one of the four variables that we need to determine), and
- sometimes, we select a value x_i that lies on the x-line $y = E_y + k_x \cdot (x_i - E_x)$, i.e., a value $x_i = E_x + K_x \cdot (y_i - E_y)$, where $K_x \stackrel{\text{def}}{=} \dfrac{1}{k_x} = \dfrac{V_x}{C}$.

In general, each expression x_i is a linear combination of a constant and the unknowns E_x, K_x, and $K_x \cdot E_y$. According to the algorithm, for each i, it takes a finite number of computational steps to check the corresponding conditions and, based on the results of this checking, to find the appropriate value x_i.

Similarly, for each i, as the appropriate value of y_i, we make one of the following four selections:

- sometimes, we select a known value $\underline{y}_i$;
- sometimes, we select a know value $\overline{y}_i$;

- sometimes, we select the value $y_i = E_y$ (which is not a priori known, it is one of the four variables that we need to determine), and
- sometimes, we select a value y_i that lies on the y-line $x = E_x + k_y \cdot (y_i - E_y)$, i.e., a value $y_i = E_y + K_y \cdot (x_i - E_x)$, where $K_y \stackrel{\text{def}}{=} \dfrac{1}{k_y} = \dfrac{V_y}{C}$.

In general, each expression y_i is a linear combination of a constant and the unknowns E_y, K_y, and $K_y \cdot E_x$.

Second step. On the fist step, for each i, we get an explicit expression of the values x_i and y_i in terms of the four parameters E_x, E_y, k_x and k_y (the parameters that describe the actual x- and y- lines): $x_i = x_i(E_x, E_y, k_x, k_y)$ and $y_i = y_i(E_x, E_y, k_x, k_y)$.

Now, we substitute these expressions for $x_i = x_i(E_x, E_y, k_x, k_y)$ and $y_i = y_i(E_x, E_y, k_x, k_y)$ into the formulas that define E_x, E_y, k_x, and k_y in terms of x_i and y_i, As a result, we get a system of four equations with four unknowns E_x, E_y, k_x and k_y:

$$E_x = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i(E_x, E_y, k_x, k_y); \quad E_y = \frac{1}{n} \cdot \sum_{i=1}^{n} y_i(E_x, E_y, k_x, k_y);$$

$$\frac{1}{n} \cdot \sum_{i=1}^{n} x_i(E_x, E_y, k_x, k_y) \cdot y_i(E_x, E_y, k_x, k_y) - E_x \cdot E_y = k_x \cdot \left(\frac{1}{n} \cdot \sum_{i=1}^{n} (x_i(E_x, E_y, k_x, k_y) - E_x)^2 \right);$$

$$\frac{1}{n} \cdot \sum_{i=1}^{n} x_i(E_x, E_y, k_x, k_y) \cdot y_i(E_x, E_y, k_x, k_y) - E_x \cdot E_y = k_y \cdot \left(\frac{1}{n} \cdot \sum_{i=1}^{n} (y_i(E_x, E_y, k_x, k_y) - E_y)^2 \right).$$

Substituting the above expressions for x_i and y_i into the four equations for the unknowns E_x, E_y, K_x, and K_y, we conclude that:

- the equation $E_x = \dfrac{1}{n} \cdot \sum_{i=1}^{n} x_i$ is transformed into equating a linear combination of E_x, K_x, and $K_x \cdot E_y$, to zero;
- the equation $E_y = \dfrac{1}{n} \cdot \sum_{i=1}^{n} y_i$ is transformed into equating a linear combination of E_y, K_y, and $K_y \cdot E_x$, to zero;

- the equation $V_x = K_x \cdot C$, i.e.,

$$K_x \cdot \left(\frac{1}{n} \cdot \sum_{i=1}^{n} x_i \cdot y_i - E_x \cdot E_y \right) = \frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - E_x)^2$$

  is transformed into equating a linear combination of terms of order ≤ 4 in terms of the unknowns;

- we also get a similar transformation for the equation $V_y \cdot K_y \cdot C$.

As a result, to find the four unknown E_x, E_y, K_x, and K_y, we get a system of four polynomial equations of order ≤ 4.

Efficient algorithms for solving systems of polynomial equations are known; see, e.g., [1, 5, 6]. The amount of computation time which is needed to solve this system does not depend on the size n of the original sample, so in terms of dependence on this size, we need $O(1)$ time. We use the known algorithms to solve the above system, and get one or several possible solutions, i.e., possible combinations of the parameters E_x, E_y, k_x, and k_y.

Third step. On the third step, for each combination of values E_x, E_y, k_x, and k_y that we obtained on the second step, we substitute these values into the formulas for the x- and y-lines, and check whether the resulting lines are indeed in the same relation to all the boxes (i.e., to all $4n$ vertices) as the corresponding representative x- and y-lines:

In other words, we check, for each of $4n$ vertices,

- whether this vertex is above, below, or on the actual x-line if and only if it is, correspondingly, above, below, or on the corresponding representative x-line, and

- whether this vertex is to the left, to the right, or on the actual y-line if and only if it is, correspondingly, to the left, to the right, or on the corresponding representative y-line.

If at least one vertex is in a different relation, we dismiss this solution. Otherwise, we compute the value of the correlation ρ based on the corresponding values $x_i(E_x, E_y, k_x, k_y)$ and $y_i(E_x, E_y, k_x, k_y)$.

Algorithm: third stage. Once we have analyzed all possible pairs of 4-tuples on the second stage of our algorithm, we move on to the final (third) stage.

On this third stage, we compute the largest of all the correlation values ρ produced on the second stage, and return this largest value as the desired value $\overline{\rho}$. The largest of all the values ρ corresponding to all possible pairs of tuples is then returned as the desired value $\overline{\rho}$.

3 Proof of the Main Result

Proof that the above algorithm is polynomial-time. Before we prove that the algorithm is correct, let us first prove that it is indeed a polynomial time algorithm.

We have $4n$ possible vertices, so we have $O(n^2)$ possible pairs of vertices – and thus, $O(n^2)$ possible 4-tuples. Thus, we have $O(n^2)$ possible representative x-lines, and we also have $O(n^2)$ representative y-lines. In our algorithms, we consider pairs consisting of a representative x-line and a representative y-line. Since we have $O(n^2)$ x-lines and we have $O(n^2)$ y-lines, we therefore have $O(n^2) \cdot O(n^2) = O(n^4)$ possible pairs consisting of a representative x-line and a representative y-line.

For each pair of lines, we perform the following computations:

- First, we need a constant number of steps to find the expression for each of n values x_i and each of n values y_i in terms of the parameters E_x, E_y, K_x, and K_y. So, we need $O(n)$ steps to find these expressions for all i.
- Then, we need linear time $O(n)$ to form the corresponding systems of four equations with four unknowns and constant time $O(1)$ to solve this system.
- Once this system is solved, and we know the corresponding values E_x, E_y, k_x, and k_y, we need:
  - linear time $O(n)$ to check whether each of $4n = O(n)$ vertices is in the right position with respect to the corresponding lines, and,
  - if needed, linear time $O(n)$ to compute the corresponding value of the correlation ρ – by using the above explicit formula describing how the correlation ρ depends on x_i and y_i.

Totally, for each pair of lines, we need

$$O(n) + O(n) + O(1) + O(n) + O(n) = O(n)$$

computational steps.

We need $O(n)$ steps for each of $O(n^4)$ pairs of lines. Thus, the total computation time of this algorithm is $O(n^4) \cdot O(n) = O(n^5)$ – which is indeed polynomial in the size n of the problem.

Case when the representative x-line coincides with the representative y-line. If this common line intersects with all n boxes $[\underline{x}_i, \overline{x}_i] \times \left[\underline{y}_i, \overline{y}_i\right]$, then, for each box, we can select values x_i and y_i for which the corresponding point (x_i, y_i) belongs to this line. Then, all selected values (x_i, y_i) follow the same linear dependence $y_i = E_y + k_x \cdot (x_i - E_x)$ (as described by the common lines). Therefore, for this selection, the correlation is 1. Since $\rho \le 1$, this means that in this case, $\overline{\rho} = 1$.

Remaining cases. Let us now prove that our algorithm is correct for all other cases, when the x- and the y-lines are different.

When a differentiable function attains maximum on the interval: known facts from calculus. A function $f(x)$ defined on an interval $[\underline{x}, \overline{x}]$ attains its maximum either at one of its endpoints, or in some internal point of the interval. If a differentiable function attains is maximum at a point $x \in (a, b)$, then its derivative at this point is 0: $\dfrac{df}{dx} = 0$.

If it attains its maximum at the point $x = \underline{x}$, then we cannot have $\dfrac{df}{dx} > 0$, because then, for some point $x + \Delta x \in [\underline{x}, \overline{x}]$, we would have a larger value of $f(x)$. Thus, in this case, we must have $\dfrac{df}{dx} \le 0$.

Similarly, if a function $f(x)$ attains its maximum at the point $x = \overline{x}$, then we must have $\dfrac{df}{dx} \ge 0$.

Comment. In this proof, we only use the relation between the maximum and the first derivatives. It is known that we can distinguish between maxima and minima (and saddle points) is we also use second derivatives. It would great to see if taking second derivatives into account could lead to faster algorithms.

Computing the corresponding derivatives. We are interested in the values x_i and y_i for which the correlation ρ attains maximum. To use the above facts, let us find the partial derivatives of ρ with respect to x_i and y_i.

The correlation is defined as the ratio of the covariance C and the product of the standard deviations σ_x and σ_y. These quantities, in their turn, are described in terms of V_x, V_y, E_x, and E_y. To compute the corresponding partial derivative, let us first compute the partial derivatives of E_x and E_y, then of V_x, V_y, and C, and then finally, of the correlation ρ.

Based on the above expression for E_x, we conclude that $\dfrac{\partial E_x}{\partial x_i} = \dfrac{1}{n}$ and similarly $\dfrac{\partial E_y}{\partial y_i} = \dfrac{1}{n}$. Since the variance V_x can be described in an equivalent form $V_x = \dfrac{1}{n} \cdot \sum\limits_{i=1}^{n} x_i^2 - E_x^2$, we get

$$\frac{\partial V_x}{\partial x_i} = \frac{2}{n} \cdot x_i - 2 \cdot E_x \cdot \frac{\partial E_x}{\partial x_i} = \frac{2}{n} \cdot (x_i - E_x).$$

Similarly,

$$\frac{\partial V_y}{\partial y_i} = \frac{2}{n} \cdot (y_i - E_y).$$

Now, since $\sigma_x = \sqrt{V_x}$, we have

$$\frac{\partial \sigma_x}{\partial x_i} = \frac{1}{2} \cdot \frac{1}{\sqrt{V_x}} \cdot \frac{\partial V_x}{\partial x_i} = \frac{1}{2} \cdot \frac{1}{\sigma_x} \cdot \frac{\partial V_x}{\partial x_i}.$$

Substituting the above formula for the derivative of V_x, we get $\dfrac{\partial \sigma_x}{\partial x_i} = \dfrac{x_i - E_x}{n \cdot \sigma_x}$ and similarly, $\dfrac{\partial \sigma_y}{\partial y_i} = \dfrac{y_i - E_y}{n \cdot \sigma_y}$.

Now, since $C = \dfrac{1}{n} \cdot \sum_{i=1}^{n} x_i \cdot y_i - E_x \cdot E_y$, we get

$$\frac{\partial C}{\partial x_i} = \frac{1}{n} \cdot y_i - \frac{\partial E_x}{\partial x_i} \cdot E_y = \frac{y_i - E_y}{n}.$$

Thus, for $\rho = \dfrac{C}{\sigma_x \cdot \sigma_y}$, since σ_y does not depend on x_i, we get

$$\frac{\partial \rho}{\partial x_i} = \frac{1}{\sigma_y} \cdot \frac{\partial}{\partial x_i}\left(\frac{C}{\sigma_x}\right) = \frac{1}{\sigma_y} \cdot \frac{\dfrac{\partial C}{\partial x_i} \cdot \sigma_x - C \cdot \dfrac{\partial \sigma_x}{\partial x_i}}{\sigma_x^2} =$$

$$\frac{1}{\sigma_y \cdot \sigma_x^2 \cdot n} \cdot \left[(y_i - E_y) \cdot \sigma_x - C \cdot \frac{x_i - E_x}{\sigma_x}\right].$$

Since the standard deviations are always non-negative, the sign of this derivative coincides with the sign of the value $(y_i - E_y) \cdot \sigma_x - C \cdot \dfrac{x_i - E_x}{\sigma_x}$. Dividing this expression by a positive value σ_x, we conclude that the sign of the derivative $\dfrac{\partial \rho}{\partial x_i}$ coincides with the sign of the expression $(y_i - E_y) - k_x \cdot (x_i - E_x)$, where we denoted $k_x \stackrel{\text{def}}{=} \dfrac{C}{V_x}$.

Similarly, the sign of the derivative $\dfrac{\partial \rho}{\partial y_i}$ coincides with the sign of the expression $(x_i - E_x) - k_y \cdot (y_i - E_y)$, where we denoted $k_y \stackrel{\text{def}}{=} \dfrac{C}{V_y}$.

It is worth mentioning since the standard deviations and variances are non-negative, the sign of both coefficients $k_x = \dfrac{C}{V_x}$ and $k_y = \dfrac{C}{V_y}$ coincides with the sign of the correlation $\rho = \dfrac{C}{\sigma_x \cdot \sigma_y}$.

Let us apply the known facts from calculus to this situation. Let x_i and y_i be the values from the corresponding boxes for which the correlation ρ attains its largest possible value $\overline{\rho} > 0$. Then, according to the above facts from calculus, we have one of the three possible situations:

- $x_i \in (\underline{x}_i, \overline{x}_i)$ and $\dfrac{\partial \rho}{\partial x_i} = 0$, i.e., $y_i = E_y + k_x \cdot (x_i - E_x)$;
- $x_i = \underline{x}_i$ and $\dfrac{\partial \rho}{\partial x_i} \le 0$, i.e., $y_i \le E_y + k_x \cdot (x_i - E_x)$;

- $x_i = \overline{x}_i$ and $\dfrac{\partial \rho}{\partial x_i} \geq 0$, i.e., $y_i \geq E_y + k_x \cdot (x_i - E_x)$.

Here, k_x has the same sign as the correlation, so $k_x > 0$. Let us now consider possible locations of the box $[\underline{x}_i, \overline{x}_i] \times \left[\underline{y}_i, \overline{y}_i\right]$ with respect to the x-line $y_i = E_y + k_x \cdot (x_i - E_x)$.

1°. The first case is when the whole box $[\underline{x}_i, \overline{x}_i] \times \left[\underline{y}_i, \overline{y}_i\right]$ is above the x-line $y_i = E_y + k_x \cdot (x_i - E_x)$, i.e., when $y_i > E_y + k_x \cdot (x_i - E_x)$ for all $y_i \in \left[\underline{y}_i, \overline{y}_i\right]$ and $x_i \in [\underline{x}_i, \overline{x}_i]$. In this case, we cannot have $x_i \in (\underline{x}_i, \overline{x}_i)$ and $x_i = \underline{x}_i$, so we must have $x_i = \overline{x}_i$.

On the segment $x_i = \overline{x}_i$, we can apply the same argument about the dependence on y_i and conclude that we can have one of the three possible situations:

- $y_i \in (\underline{y}_i, \overline{y}_i)$ and $\dfrac{\partial \rho}{\partial y_i} = 0$, i.e., $x_i = E_x + k_y \cdot (y_i - E_y)$;
- $y_i = \underline{y}_i$ and $\dfrac{\partial \rho}{\partial y_i} \leq 0$, i.e., $x_i \leq E_x + k_y \cdot (y_i - E_y)$;
- $y_i = \overline{y}_i$ and $\dfrac{\partial \rho}{\partial y_i} \geq 0$, i.e., $x_i \geq E_x + k_y \cdot (y_i - E_y)$.

Here, k_y has the same sign as the correlation, so $k_y > 0$. Let us now consider possible locations of the segment $\{\overline{x}_i\} \times \left[\underline{y}_i, \overline{y}_i\right]$ in relation to the y-line $x_i = E_x + k_y \cdot (y_i - E_y)$.

1.1°. The first subcase is when the whole segment is to the left of the y-line, i.e., when $x_i < E_x + k_y \cdot (y_i - E_y)$ for all $y_i \in \left[\underline{y}_i, \overline{y}_i\right]$. In this case, we cannot have $y_i \in (\underline{y}_i, \overline{y}_i)$ and we cannot have $y_i = \overline{y}_i$, so we must have $y_i = \underline{y}_i$.

1.2°. The second subcase is when the whole segment is to the right of the y-line, i.e., when $x_i > E_x + k_y \cdot (y_i - E_y)$ for all $y_i \in \left[\underline{y}_i, \overline{y}_i\right]$. In this case, we cannot have $y_i \in (\underline{y}_i, \overline{y}_i)$ and we cannot have $y_i = \underline{y}_i$, so we must have $y_i = \overline{y}_i$.

1.3°. The third subcase is when the segment intersects the y-line, i.e., when $x_i = E_x + k_y \cdot (y_i' - E_y)$ for some $y_i' \in \left[\underline{y}_i, \overline{y}_i\right]$. As we have mentioned, there are three possibilities for the value y_i at which the correlation attains its maximum: the value for which $x_i = E_x + k_y \cdot (y_i - E_y)$, the value $\underline{y}_i$, and the value $\overline{y}_i$.

1.3.1°. In the first case (when $x_i = E_x + k_y \cdot (y_i - E_y)$), since $k_y > 0$, there is only one value $y_i = y_i'$.

1.3.2°. If $\underline{y}_i \neq y_i'$, then $\underline{y}_i < y_i'$, and thus,

$$E_x + k_y \cdot (\underline{y}_i - E_y) < E_x + k_y \cdot (y_i' - E_y) = x_i.$$

Thus, we have $x_i > E_x + k_y \cdot (\underline{y}_i - E_y)$, so we cannot have $x_i \leq E_x + k_y \cdot (\underline{y}_i - E_y)$, and therefore, the maximum cannot be attained for $y_i = \underline{y}_i$.

1.3.3°. If $\overline{y}_i \neq y_i'$, then $y_i' < \overline{y}_i$, and thus,

$$x_i = E_x + k_y \cdot (y_i' - E_y) < E_x + k_y \cdot (\overline{y}_i - E_y) = x_i.$$

Thus, we have $x_i < E_x + k_y \cdot (\overline{y}_i - E_y)$, so we cannot have $x_i \le E_x + k_y \cdot (\overline{y}_i - E_y)$, and therefore, the maximum cannot be attained for $y_i = \overline{y}_i$.

1.3.4°. Therefore, in this third subcase, maximum can only be attained at the point on the y-line.

2°. The second case is when the whole box $[\underline{x}_i, \overline{x}_i] \times \left[\underline{y}_i, \overline{y}_i\right]$ is below the x-line $y_i = E_y + k_x \cdot (x_i - E_x)$, i.e., when $y_i < E_y + k_x \cdot (x_i - E_x)$ for all $y_i \in \left[\underline{y}_i, \overline{y}_i\right]$ and $x_i \in [\underline{x}_i, \overline{x}_i]$. In this case, we cannot have $x_i \in (\underline{x}_i, \overline{x}_i)$ and we cannot have $x_i = \overline{x}_i$, so we must have $x_i = \underline{x}_i$.

On the segment $x_i = \underline{x}_i$, we can apply the same argument about the dependence on y_i as in Part 1 of this proof and come with the same conclusions.

3°. Same arguments apply if the whole box is fully to the left or to the right of the y-line. In this case, we have $y_i = \overline{y}_i$ or $y_i = \underline{y}_i$.

4°. The only remaining case is when the box intersects both with the x-line and with the y-line. In this case, similar to Part 1.3 of this proof, we conclude that the point (x_i, y_i) corresponding to the optimal tuple belongs both to the x-line and to the y-line. Thus, this point coincides with the intersection of these two lines.

In general, the x-line has the form $y - E_y = k_x \cdot (x - E_x)$. The y-line has the form $x - E_x = k_y \cdot (y - E_y)$, i.e., equivalently, $y - E_y = \dfrac{1}{k_y} \cdot (x - E_x)$. Both lines pass through the same point (E_x, E_y), but their slopes are, in general, different: k_x for the x-line and $\dfrac{1}{k_y}$ for the y-line. Thus, these lines coincide if and only if $k_x = \dfrac{1}{k_y}$, i.e., if and only if $k_x \cdot k_y = 1$.

In general, $\rho \le 1$. Here, $\rho = \dfrac{C}{\sigma_x \cdot \sigma_y} = \dfrac{C}{\sqrt{V_x} \cdot \sqrt{V_y}}$; thus, $\rho = \sqrt{k_x \cdot k_y}$, so $k_x \cdot k_y \le 1$. If $k_x \cdot k_y < 1$, then $k_x \cdot k_y \neq 1$ and thus, the x-line and the y-line are different. So, the intersection of these two lines is a single point (E_x, E_y). If $k_x \cdot k_y = 1$, this means that $\rho = 1$, and all the points (x_i, y_i) are on the same straight line – this is the case we have considered above.

5°. We enumerated all the cases described in the algorithm and showed that in all these cases, we should produce exactly the values x_i and y_i described in the algorithm. Thus, we have justified the algorithm – provided that we enumerate all possible locations of the vertices with respect to x- and y-lines.

To complete the proof, we need to show that all possible locations are captured by what we called representative x- and y-lines. Indeed, let us start with any x-line, and let us show that there exists a representative x-line that has

exactly the same location with respect to all the vertices – i.e., that each vertex is above, below, or on the representative x-line if and only if this vertex is, correspondingly, above, below, or on the actual x-line.

Let us take the actual x-line. If it contains one of the vertices, mark this vertex. If the original x-line does not contain any of the vertices, let us move the line (parallel to itself) along the x-axis – until the line hits a vertex. Then, we move the line back by a small amount, and we mark this almost-vertex point.

Once the marked vertex is fixed, we check if the line contains another vertex. If it does, we mark that vertex, and so we have the desired representative x-line. If it does not, we rotate the line around the already marked vertex (or almost-vertex) until the line starts containing another vertex. We similarly move the line back by a small amount, and we get the desired representative x-line that is in exactly same relation to all the vertices as the actual x-line.

We can perform the same procedure with the y-line. Correctness is proven.

4 Conclusion

In many practical situations, it is important to find the correlation between the two quantities x and y. Usually, the correlation is estimated based on the sample values $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$. Traditional statistical methods assume that we know the exact values of x_i and y_i. In practice, the sample values of x and y come from measurements; measurements are never absolutely accurate, so the measured values $\widetilde{x}_i$ and $\widetilde{y}_i$ are, in general, different from the actual (unknown) values x_i and y_i of the corresponding quantities. Often, the only information that we have about the measurement inaccuracy $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i$ and $\Delta y_i \stackrel{\text{def}}{=} \widetilde{y}_i - y_i$ of each measurement is the upper bound (Δ_{xi} or Δ_{yi}) for which $|\Delta x_i| \le \Delta_{xi}$ and $|\Delta y_i| \le \Delta_{yi}$. In this case, we only know the intervals $[\widetilde{x}_i - \Delta_{xi}, \widetilde{x}_i + \Delta_{xi}]$ and $[\widetilde{y}_i - \Delta_{yi}, \widetilde{y}_i + \Delta_{yi}]$ of possible values of each sample point x_i and y_i. Different values x_i and y_i from these intervals lead, in general, to different values of the correlation ρ. It is therefore desirable to find the range $\left[\underline{\rho}, \overline{\rho}\right]$ of possible values of ρ. In general, the problem of computing both endpoints of this range is known to be NP-hard. In this paper, we describe a feasible algorithms for computing one of the endpoints – the one that corresponds to the largest absolute value $|\rho|$.

Acknowledgments. This work was supported in part by the National Science Foundation grants HRD-0734825 (Cyber-ShARE Center of Excellence) and DUE-0926721 and by Grant 1 T36 GM078000-01 from the National Institutes of Health. The authors are thankful to the anonymous referees for valuable suggestions.

References

[1] S. Basu, R. Pollack, and M.-F. Roy, *Algorithms in Real Algebraic Geometry*, Springer-Verlag, Berlin, 2006.

[2] C. Ferregut, F. J. Campos, and V. Kreinovich, "Reducing over-conservative expert failure rate estimates in the presence of limited data: a new probabilistic/fuzzy approach", *Proceedings of the 30th Annual Conference of the North American Fuzzy Information Processing Society NAFIPS'2011*, El Paso, Texas, March 18–20, 2011.

[3] 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.

[4] S. Ferson, L. Ginzburg, V. Kreinovich, L. Longpré, and M. Aviles, "Exact Bounds on Finite Populations of Interval Data", *Reliable Computing*, vol. 11, no. 3, pp. 207–233, 2005.

[5] D. Yu. Grigor'ev and N. N. Vorobjov, Jr., "Finding real solutions of systems of algebraic inequalities in subexponential time", *Soviet Math. Dokl.*, vol. 32, no. 1, pp. 316–320, 1985.

[6] D. Yu. Grigor'ev and N. N. Vorobjov, Jr., "Solving systems of polynomial inequalities in subexponential time", *Journal of Symbolic Computation*, vol. 5, no. 1-2, pp. 37–64, 1988.

[7] C. Jacob, D. Dubois, J. Cardoso, M. Ceberio, and V. Kreinovich, "Estimating Probability of Failure of a Complex System Based on Partial Information about Subsystems and Components, with Potential Applications to Aircraft Maintenance", *Proceedings of the International Workshop on Soft Computing Applications and Knowledge Discovery SCAKD'2011*, Moscow, Russia, June 25, 2011, pp. 30–41.

[8] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*, Springer-Verlag, London, 2001.

[9] V. Kreinovich, "Reliability Analysis for Aerospace Applications: Reducing Over-Conservative Expert Estimates in the Presence of Limited Data", In: Sergei O. Kuznetsov and Dominik Slezak (eds.), *Expert and Industry Sessions of the 13th International Conference on Rough Sets, Fuzzy Sets and Granular Computing RSFDGrC'2011 and the 4th International Conference on Pattern Recognition and Machine Intelligence PReMI'2011*, Moscow, Russia, June 25–30, 2011.

[10] 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: al-

gorithms for computing statistics under interval uncertainty, and their computational complexity", *Reliable Computing*, vol. 12, no. 6, pp. 471–501, 2006.

[11] 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.

[12] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*, SIAM Press, Philadelphia, Pennsylvania, 2009.

[13] H. T. Nguyen, V. Kreinovich, B. Wu, and G. Xiang, *Computing Statistics under Interval and Fuzzy Uncertainty*, Springer Verlag, to appear.

[14] R. Osegueda, V. Kreinovich, L. Potluri, and R. Aló, "Non-destructive testing of aerospace structures: granularity and data mining approach". *Proc. FUZZ-IEEE'2002*, Honolulu, Hawaii, May 12–17, 2002, vol. 1, pp. 685–689.

[15] S. Rabinovich, *Measurement Errors and Uncertainties: Theory and Practice*, Springer Verlag, New York, 2005.

[16] F. Tonon and C. L. Pettit, "Toward a definition and understanding of correlation for variables constrained by random relations", *International Journal of General Systems*, 2010, Vol. 39, No. 6, pp. 577–604.