Outlier Detection Under Interval Uncertainty: Algorithmic Solvability and Computational Complexity

Vladik Kreinovich[1], Luc Longpré[1], Praveen Patangay[1], Scott Ferson[2], and Lev Ginzburg[2]

[1] Computer Science Department, University of Texas at El Paso
El Paso, TX 79968, USA
{vladik, longpre, praveen}@cs.utep.edu
[2] Applied Biomathematics, 100 North Country Road
Setauket, NY 11733, USA
{scott, lev}@ramas.com

Abstract. In many application areas, it is important to detect outliers. Traditional engineering approach to outlier detection is that we start with some "normal" values $x_1, \ldots, x_n$, compute the sample average E, the sample standard variation σ, and then mark a value x as an outlier if x is outside the k_0-sigma interval $[E - k_0 \cdot \sigma, E + k_0 \cdot \sigma]$ (for some pre-selected parameter k_0). In real life, we often have only interval ranges $[\underline{x}_i, \overline{x}_i]$ for the normal values $x_1, \ldots, x_n$. In this case, we only have intervals of possible values for the bounds $E - k_0 \cdot \sigma$ and $E + k_0 \cdot \sigma$. We can therefore identify outliers as values that are outside all k_0-sigma intervals. In this paper, we analyze the computational complexity of these outlier detection problems, and provide efficient algorithms that solve some of these problems (under reasonable conditions).

1 Introduction

In many application areas, it is important to detect *outliers*, i.e., unusual, abnormal values. In medicine, unusual values may indicate disease (see, e.g., [7]); in geophysics, abnormal values may indicate a mineral deposit or an erroneous measurement result (see, e.g., [5], [9], [13], [16]); in structural integrity testing, abnormal values may indicate faults in a structure (see, e.g., [2], [6], [7], [10], [11], [17]), etc.

Traditional engineering approach to outlier detection (see, e.g., [1], [12], [15]) is as follows:

- first, we collect measurement results $x_1, \ldots, x_n$ corresponding to normal situations;
- then, we compute the sample average $E \stackrel{\text{def}}{=} \dfrac{x_1 + \ldots + x_n}{n}$ of these normal values and the (sample) standard deviation $\sigma = \sqrt{V}$, where $V \stackrel{\text{def}}{=} \dfrac{(x_1 - E)^2 + \ldots + (x_n - E)^2}{n}$;

– finally, a new measurement result x is classified as an outlier if it is outside the interval $[L, U]$ (i.e., if either $x < L$ or $x > U$), where $L \stackrel{\text{def}}{=} E - k_0 \cdot \sigma$, $U \stackrel{\text{def}}{=} E + k_0 \cdot \sigma$, and $k_0 > 1$ is some pre-selected value (most frequently, $k_0 = 2$, 3, or 6).

In some practical situations, we only have intervals $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$ of possible values of x_i. This happens, for example, if instead of observing the actual value x_i of the random variable, we observe the value $\widetilde{x}_i$ measured by an instrument with a known upper bound Δ_i on the measurement error; then, the actual (unknown) value is within the interval $\mathbf{x}_i = [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$. For different values $x_i \in \mathbf{x}_i$, we get different bounds L and U. Possible values of L form an interval – we will denote it by $\mathbf{L} \stackrel{\text{def}}{=} [\underline{L}, \overline{L}]$; possible values of U form an interval $\mathbf{U} = [\underline{U}, \overline{U}]$.

How do we now detect outliers? There are two possible approaches to this question: we can detect *possible* outliers and we can detect *guaranteed* outliers:

– a value x is a possible outlier if it is located outside one of the possible k_0-sigma intervals $[L, U]$ (but is may be inside some other possible interval $[L, U]$);
– a value x is a guaranteed outlier if it is located outside all possible k_0-sigma intervals $[L, U]$.

Which approach is more reasonable depends on a possible situation:

– if our main objective is not to miss an outlier, e.g., in structural integrity tests, when we do not want to risk launching a spaceship with a faulty part, it is reasonable to look for possible outliers;
– if we want to make sure that the value x is an outlier, e.g., if we are planning a surgery and we want to make sure that there is a micro-calcification before we start cutting the patient, then we would rather look for guaranteed outliers.

The two approaches can be described in terms of the endpoints of the intervals $\mathbf{L}$ and $\mathbf{U}$:

A value x guaranteed to be normal – i.e., it is not a possible outlier – if x belongs to the *intersection* of all possible intervals $[L, U]$; the intersection corresponds to the case when L is the largest and U is the smallest, i.e., this intersection is the interval $[\overline{L}, \underline{U}]$. So, if $x > \underline{U}$ or $x < \overline{L}$, then x is a possible outlier, else it is guaranteed to be a normal value.

If a value x is inside *one* of the possible intervals $[L, U]$, then it can still be normal; the only case when we are sure that the value x is an outlier is when x is outside *all* possible intervals $[L, U]$, i.e., is the value x does not belong to the *union* of all possible intervals $[L, U]$ of normal values; this union is equal to the interval $[\underline{L}, \overline{U}]$. So, if $x > \overline{U}$ or $x < \underline{L}$, then x is a guaranteed outlier, else it can be a normal value.

In real life, the situation may be slightly more complicated because, as we have mentioned, measurements often come with interval inaccuracy; so, instead of the exact value x of the measured quantity, we get an interval $\mathbf{x} = [\underline{x}, \overline{x}]$ of possible values of this quantity.

In this case, we have a slightly more complex criterion for outlier detection:

- the actual (unknown) value of the measured quantity is a possible outlier if some value x from the interval $[\underline{x}, \overline{x}]$ is a possible outlier, i.e., is outside the intersection $[\overline{L}, \underline{U}]$; thus, the value is a possible outlier if one of the two inequalities hold: $\underline{x} < \overline{L}$ or $\underline{U} < \overline{x}$.
- the actual (unknown) value of the measured quantity is guaranteed to be an outlier if all possible values x from the interval $[\underline{x}, \overline{x}]$ are guaranteed to be outliers (i.e., are outside the union $[\underline{L}, \overline{U}]$); thus, the value is a guaranteed outlier if one of the two inequalities hold: $\overline{x} < \underline{L}$ or $\overline{U} < \underline{x}$.

Thus:

- to detect possible outliers, we must be able to compute the values $\overline{L}$ and $\underline{U}$;
- to detect guaranteed outliers, we must be able to compute the values $\underline{L}$ and $\overline{U}$.

In this paper, we consider the problem of computing these bounds.

2 What Was Known Before

As we discussed in the introduction, to detect outliers under interval uncertainty, we must be able to compute the range $\mathbf{L} = [\underline{L}, \overline{L}]$ of possible values of $L = E - k_0 \cdot \sigma$ and the range $\mathbf{U} = [\underline{U}, \overline{U}]$ of possible values of $U = E + k_0 \cdot \sigma$.

In [3, 4], we have shown how to compute the intervals $\mathbf{E} = [\underline{E}, \overline{E}]$ and $[\underline{\sigma}, \overline{\sigma}]$ of possible values for E and σ. In principle, we can use the general ideas of interval computations to combine these intervals and conclude, e.g., that L always belongs to the interval $\mathbf{E} - k_0 \cdot [\underline{\sigma}, \overline{\sigma}]$. However, as often happens in interval computations, the resulting interval for L is *wider* than the actual range – wider because the values E and σ are computed based on the same inputs $x_1, \ldots, x_n$ and cannot, therefore, change independently.

We mark a value x as an outlier if it is outside the interval $[L, U]$. Thus, if, instead of the actual ranges for L and U, we use wider intervals, we may miss some outliers. It is therefore important to compute the *exact* ranges for L and U. In this paper, we show how to compute these exact ranges.

3 Detecting Possible Outliers

To find possible outliers, we must know the values $\underline{U}$ and $\overline{L}$. In this section, we design *feasible* algorithms for computing the exact lower bound $\underline{U}$ of the function U and the exact upper bound $\overline{L}$ of the function L. Specifically, our algorithms are *quadratic-time*, i.e., require $O(n^2)$ computational steps (arithmetic operations or comparisons) for n interval data points $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$.

The algorithms $\underline{\mathcal{A}}_U$ for computing $\underline{U}$ and $\overline{\mathcal{A}}_L$ for computing $\overline{L}$ are as follows:

- In both algorithms, first, we sort all $2n$ values $\underline{x}_i$, $\overline{x}_i$ into a sequence $x_{(1)} \leq x_{(2)} \leq \ldots \leq x_{(2n)}$; take $x_{(0)} = -\infty$ and $x_{(2n+1)} = +\infty$. Thus, the real line is divided into $2n+1$ zones $(x_{(0)}, x_{(1)}]$, $[x_{(1)}, x_{(2)}]$, $\ldots$, $[x_{(2n-1)}, x_{(2n)}]$, $[x_{(2n)}, x_{(2n+1)})$.

– For each of these zones $[x_{(k)}, x_{(k+1)}]$, $k = 0, 1, \ldots, 2n$, we compute the values

$$e_k \stackrel{\text{def}}{=} \sum_{i: \underline{x}_i \geq x_{(k+1)}} \underline{x}_i + \sum_{j: \overline{x}_j \leq x_{(k)}} \overline{x}_j, \quad (1)$$

$$m_k \stackrel{\text{def}}{=} \sum_{i: \underline{x}_i \geq x_{(k+1)}} (\underline{x}_i)^2 + \sum_{j: \overline{x}_j \leq x_{(k)}} (\overline{x}_j)^2, \quad (2)$$

and $n_k =$ the total number of such i's and j's. Then, we solve the quadratic equation

$$A - B \cdot \mu + C \cdot \mu^2 = 0, \quad (3)$$

where

$$A \stackrel{\text{def}}{=} e_k^2 \cdot (1 + \alpha^2) - \alpha^2 \cdot m_k \cdot n; \quad \alpha \stackrel{\text{def}}{=} 1/k_0, \quad (4)$$

$$B \stackrel{\text{def}}{=} 2 \cdot e_k \cdot \left((1 + \alpha^2) \cdot n_k - \alpha^2 \cdot n\right); \quad C \stackrel{\text{def}}{=} n_k \cdot \left((1 + \alpha^2) \cdot n_k - \alpha^2 \cdot n\right). \quad (5)$$

For computing $\underline{U}$, we select only those solutions for which $\mu \cdot n_k \leq e_k$ and $\mu \in [x_{(k)}, x_{(k+1)}]$; for computing $\overline{U}$, we select only those solutions for which $\mu \cdot n_k \geq e_k$ and $\mu \in [x_{(k)}, x_{(k+1)}]$. For each selected solution, we compute the values of

$$E_k = \frac{e_k}{n} + \frac{n - n_k}{n} \cdot \mu, \quad M_k = \frac{m_k}{n} + \frac{n - n_k}{n} \cdot \mu^2, \quad (6)$$

and, correspondingly,

$$U_k = E_k + k_0 \cdot \sqrt{M_k - (E_k)^2} \ \text{ or } \ L_k = E_k - k_0 \cdot \sqrt{M_k - (E_k)^2} \quad (7)$$

– Finally, if we are computing $\underline{U}$, we return the smallest of the values U_k; if we are computing $\overline{L}$, we return the smallest of the values L_k.

Theorem 3.1. *The algorithms $\underline{\mathcal{A}}_U$ and $\overline{\mathcal{A}}_L$ always compute $\underline{U}$ and $\overline{L}$ in quadratic time.*

Comment. The main idea of this proof is given in the last (Proofs) section. The detailed proofs are given in http://www.cs.utep.edu/vladik/2003/tr03-10c.ps.gz and in http://www.cs.utep.edu/vladik/2003/tr03-10c.pdf

4 In General, Detecting Guaranteed Outliers is NP-Hard

As we have mentioned in Section 1, to be able to detect guaranteed outliers, we must be able to compute the values $\underline{L}$ and $\overline{U}$. In general, this is an NP-hard problem:

Theorem 4.1. *For every $k_0 > 1$, computing the upper endpoint $\overline{U}$ of the interval $[\underline{U}, \overline{U}]$ of possible values of $U = E + k_0 \cdot \sigma$ is NP-hard.*

Theorem 4.2. *For every $k_0 > 1$, computing the lower endpoint $\underline{L}$ of the interval $[\underline{L}, \overline{L}]$ of possible values of $L = E - k_0 \cdot \sigma$ is NP-hard.*

Comment. For interval data, the NP-hardness of computing the upper bound for σ was proven in [3] and [4]. The general overview of NP-hardness of computational problems in interval context is given in [8].

5 How Can We Actually Detect Guaranteed Outliers?

How can we actually compute these values? First, we will show that if $1 + (1/k_0)^2 < n$ (which is true, e.g., if $k_0 > 1$ and $n \geq 2$), then the maximum of U (correspondingly, the minimum of L) is always attained at some combination of endpoints of the intervals $\mathbf{x}_i$; thus, in principle, to determine the values $\overline{U}$ and $\underline{L}$, it is sufficient to try all 2^n combinations of values $\underline{x}_i$ and $\overline{x}_i$:

Theorem 5.1. *If $1+(1/k_0)^2 < n$, then the maximum of the function U and the minimum of the function L on the box $\mathbf{x}_1 \times \ldots \times \mathbf{x}_n$ are attained at its vertices, i.e., when for every i, either $x_i = \underline{x}_i$ or $x_i = \overline{x}_i$.*

NP-hard means, crudely speaking, that there are no general ways for solving all particular cases of this problem (i.e., computing $\overline{V}$) in reasonable time.

However, we show that there are algorithms for computing $\overline{U}$ and $\underline{L}$ for many reasonable situations. Namely, we propose efficient algorithms that compute $\overline{U}$ and $\underline{L}$ for the case when all the interval midpoints ("measured values") $\widetilde{x}_i \stackrel{\text{def}}{=} (\underline{x}_i + \overline{x}_i)/2$ are definitely different from each other, in the sense that the "narrowed" intervals

$$\left[\widetilde{x}_i - \frac{1+\alpha^2}{n} \cdot \Delta_i, \widetilde{x}_i + \frac{1+\alpha^2}{n} \cdot \Delta_i\right] \tag{8}$$

– where $\alpha = 1/k_0$ and $\Delta_i \stackrel{\text{def}}{=} (\underline{x}_i - \overline{x}_i)/2$ is the interval's half-width – do not intersect with each other.

The algorithms $\overline{\mathcal{A}}_U$ and $\underline{\mathcal{A}}_L$ are as follows:

- In both algorithms, first, we sort all $2n$ endpoints of the narrowed intervals $\widetilde{x}_i - \frac{1+\alpha^2}{n} \cdot \Delta_i$ and $\widetilde{x}_i + \frac{1+\alpha^2}{n} \cdot \Delta_i$ into a sequence $x_{(1)} \leq x_{(2)} \leq \ldots \leq x_{(2n)}$. This enables us to divide the real line into $2n+1$ segments ("small intervals") $[x_{(i)}, x_{(i+1)}]$, where we denoted $x_{(0)} \stackrel{\text{def}}{=} -\infty$ and $x_{(2n+1)} \stackrel{\text{def}}{=} +\infty$.
- For each of small intervals $[x_{(i)}, x_{(i+1)}]$, we do the following: for each j from 1 to n, we pick the following value of x_j:
  - if $x_{(i+1)} < \widetilde{x}_j - \frac{1+\alpha^2}{n} \cdot \Delta_j$, then we pick $x_j = \overline{x}_j$;
  - if $x_{(i+1)} > \widetilde{x}_j + \frac{1+\alpha^2}{n} \cdot \Delta_j$, then we pick $x_j = \underline{x}_j$;
  - for all other j, we consider both possible values $x_j = \overline{x}_j$ and $x_j = \underline{x}_j$.

  As a result, we get one or several sequences of x_j for each small interval.
- To compute $\overline{U}$, for each of the sequences x_j, we check whether, for the selected values $x_1, \ldots, x_n$, the value of $E - \alpha \cdot \sigma$ is indeed within the corresponding small interval, and if it is, compute the value $U = E + k_0 \cdot \sigma$. Finally, we return the largest of the computed values U as $\overline{U}$.

– To compute $\underline{L}$, for each of the sequences x_j, we check whether, for the selected values $x_1, \ldots, x_n$, the value of $E + \alpha \cdot \sigma$ is indeed within the corresponding small interval, and if it is, compute the value $L = E - k_0 \cdot \sigma$. Finally, we return the smallest of the computed values L as $\underline{L}$.

Theorem 5.2. *Let $1/n + 1/k_0^2 < 1$. The algorithms $\overline{\mathcal{A}}_U$ and $\underline{\mathcal{A}}_L$ compute $\overline{U}$ and $\underline{L}$ in quadratic time for all the cases in which the "narrowed" intervals do not intersect with each other.*

These algorithms also work when, for some fixed C, no more than C "narrowed" intervals can have a common point:

Theorem 5.3. *Let $1+(1/k_0)^2 < n$. For every positive integer C, the algorithms $\overline{\mathcal{A}}_U$ and $\underline{\mathcal{A}}_L$ compute $\overline{U}$ and $\underline{L}$ in quadratic time for all the cases in which no more than C "narrowed" intervals can have a common point.*

The corresponding computation times are quadratic in n but grow exponentially with C. So, when C grows, this algorithm requires more and more computation time. It is worth mentioning that the examples on which we prove NP-hardness correspond to the case when $n/2$ out of n narrowed intervals have a common point.

6 Proofs: Main Idea

Our proof of Theorem 2.1 is based on the fact that when the function $U(x_1, \ldots, x_n)$ attains its smallest possible value at some point $(x_1^{\rm opt}, \ldots, x_n^{\rm opt})$, then, for every i, the corresponding function of one variable

$$U_i(x_i) \stackrel{\rm def}{=} U(x_1^{\rm opt}, \ldots, x_{i-1}^{\rm opt}, x_i, x_{i+1}^{\rm opt}, \ldots, x_n^{\rm opt}) \tag{9}$$

– the function that is obtained from $U(x_1, \ldots, x_n)$ by fixing the values of all the variables except for x_i – also attains its minimum at the value $x_i = x_i^{\rm opt}$.

A differentiable function of one variable attains its minimum on a closed interval either at one of its endpoints or at an internal point in which its first derivative is equal to 0.

This first derivative is equal to 0 when $\sigma + k_0 \cdot (x_i - E) = 0$, i.e., when $x_i = E - \alpha \cdot \sigma$, where $\alpha = 1/k_0$. Thus, for the optimal values $x_1, \ldots, x_n$ for which U attains its minimum, for every i, we have either $x_i = \underline{x}_i$, or $x_i = \overline{x}_i$, or $x_i = E - \alpha \cdot \sigma$.

We then show that if the open interval $(\underline{x}_i, \overline{x}_i)$ contains the value $E - \alpha \cdot \sigma$, then the minimum of the function cannot be attained at points $\overline{x}_i$ or $\underline{x}_i$ and therefore, has to be attained at the value $x_i = E - \alpha \cdot \sigma$.

We also show that:

– when $E - \alpha \cdot \sigma \le \underline{x}_i$, the minimum cannot be attained for $x_i = \overline{x}_i$ and therefore, it is attained when $x_i = \underline{x}_i$;

– when $\overline{x}_i \le E - \alpha \cdot \sigma$, the minimum cannot be attained for $x_i = \underline{x}_i$ and therefore, it is attained when $x_i = \overline{x}_i$.

Due to what we have proven, once we know how the value $\mu \stackrel{\rm def}{=} E - \alpha \cdot \sigma$ is located with respect to all the intervals $[\underline{x}_i, \overline{x}_i]$, we can find the optimal values of x_i. Hence, to find the minimum, we need to analyze how the endpoints $\underline{x}_i$ and $\overline{x}_i$ divide the real line, and consider all the resulting sub-intervals.

Conclusions

In many application areas, it is important to detect outliers. Traditional engineering approach to outlier detection is that we start with some "normal" values $x_1, \ldots, x_n$, compute the sample average E, the sample standard variation σ, and then mark a value x as an outlier if x is outside the k_0-sigma interval $[E - k_0 \cdot \sigma, E + k_0 \cdot \sigma]$ (for some pre-selected parameter k_0).

In real life, we often have only interval ranges $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$ for the normal values $x_1, \ldots, x_n$. For different values $x_i \in \mathbf{x}_i$, we get different values of $L \stackrel{\rm def}{=} E - k_0 \cdot \sigma$ and $U \stackrel{\rm def}{=} E + k_0 \cdot \sigma$ – and thus, different k_0-sigma intervals $[L, U]$. We can therefore identify *guaranteed* outliers as values that are outside *all* k_0-sigma intervals, and *possible* outliers as values that are outside *some* k_0-sigma intervals. To detect guaranteed and possible outliers, we must therefore be able to compute the *range* $\mathbf{L} = [\underline{L}, \overline{L}]$ of possible values of L and the range $\mathbf{U} = [\underline{U}, \overline{U}]$ of possible values of U.

In our previous papers [3, 4], we have shown how to compute the intervals $\mathbf{E} = [\underline{E}, \overline{E}]$ and $[\underline{\sigma}, \overline{\sigma}]$ of possible values for E and σ. In principle, we can combine these intervals and conclude, e.g., that L always belongs to the interval $\mathbf{E} - k_0 \cdot [\underline{\sigma}, \overline{\sigma}]$. However, the resulting interval for L is *wider* than the actual range – wider because the values E and σ are computed based on the same inputs $x_1, \ldots, x_n$ and are, therefore, not independent from each other.

If, instead of the actual ranges for L and U, we use wider intervals, we may miss some outliers. It is therefore important to compute the *exact* ranges for L and U.

In this paper, we showed that computing these ranges is, in general, NP-hard, and we provided efficient algorithms that compute these ranges under reasonable conditions.

Acknowledgments

This work was supported in part by NASA grants NCC5-209 and grant NCC2-1232, by the Air Force Office of Scientific Research grant F49620-00-1-0365, by NSF grants CDA-9522207, EAR-0112968, EAR-0225670, and 9710940 Mexico/Conacyt, by IEEE/ACM SC2001 and SC2002 Minority Serving Institutions Participation Grants, by Small Business Innovation Research grant 9R44CA81741 to Applied Biomathematics from the National Cancer Institute (NCI), a component of the National Institutes of Health (NIH), and by a grant from Sandia National Laboratories as part of the Department of Energy Accelerated Strategic Computing Initiative (ASCI).

References

1. Devore, J., Peck, R.: Statistics: the Exploration and Analysis of Data, Duxbury, Pacific Grove, California (1999)
2. Ferregut, C., Osegueda, R.A., and Nuñez, A. (eds.): Proceedings of the International Workshop on Intelligent NDE Sciences for Aging and Futuristic Aircraft, El Paso, TX, September 30–October 2, 1997
3. Ferson, S., Ginzburg, L., Kreinovich, V., Longpré, L., Aviles, M.: Computing Variance for Interval Data is NP-Hard. ACM SIGACT News **33**, No. 2 (2002) 108–118
4. Ferson, S., Ginzburg, L., Kreinovich, V., Aviles, M.: Exact Bounds on Sample Variance of Interval Data. In: Extended Abstracts of the 2002 SIAM Workshop on Validated Computing, Toronto, Canada (2002) 67–69
5. Goodchild, M., Gopal, S.: Accuracy of Spatial Databases, Taylor & Francis, London (1989)
6. Gros, X.E.: NDT Data Fusion, J. Wiley, London (1997)
7. Kosheleva, O., Cabrera, S., Osegueda, R., Nazarian, S., George, D.L., George, M.J., Kreinovich, V., Worden, K.: Case study of non-linear inverse problems: mammography and non-destructive evaluation. In: Mohamad-Djafari, A. (ed.): Bayesian Inference for Inverse Problems, Proceedings of the SPIE/International Society for Optical Engineering, vol. 3459, San Diego, CA (1998) 128–135.
8. Kreinovich, V., Lakeyev, A., Rohn, J., Kahl, P.: Computational complexity and feasibility of data processing and interval computations, Kluwer, Dordrecht (1997)
9. McCain, M., William, C.: Integrating Quality Assurance into the GIS Project Life Cycle. Proceedings of the 1998 ESRI Users Conference http://www.dogcreek.com/html/documents.html
10. Osegueda, R., Kreinovich, V., Potluri, L., Aló, R.: Non-Destructive Testing of Aerospace Structures: Granularity and Data Mining Approach. Proceedings of FUZZ-IEEE'2002, Honolulu, Hawaii, May 12–17 (2002) vol. 1, 685–689
11. Osegueda, R.A., Seelam, S.R., Holguin, A.C., Kreinovich, V., Tao, C.-W.: Statistical and Dempster-Shafer Techniques in Testing Structural Integrity of Aerospace Structures. International Journal of Uncertainty, Fuzziness, Knowledge-Based Systems (IJUFKS) **9** (2001) 749–758
12. Rabinovich, S.: Measurement Errors: Theory and Practice, American Institute of Physics, New York (1993)
13. Scott, L.: Identification of GIS Attribute Error Using Exploratory Data Analysis. Professional Geographer **46** (1994) 378–386
14. Vavasis, S.A.: Nonlinear optimization: complexity issues, Oxford University Press, N.Y. (1991)
15. Wadsworth, H.M. Jr. (ed.): Handbook of statistical methods for engineers and scientists, McGraw-Hill Publishing Co., N.Y. (1990)
16. Wen, Q., Gates, A.Q., Beck, J., Kreinovich, V., Keller, G.R.: Towards automatic detection of erroneous measurement results in a gravity database. Proceedings of the 2001 IEEE Systems, Man, and Cybernetics Conference, Tucson, Arizona, October 7–10 (2001) 2170–2175
17. Worden, K., Osegueda, R., Ferregut, C., Nazarian, S., George, D.L., George, M.J., Kreinovich, V., Kosheleva, O., and Cabrera, C.: Interval Methods in Non-Destructive Testing of Material Structures. Reliable Computing **7** (2001) 341–352