Practical Applications of Interval Computations: an Overview with a Special Emphasis on Applications Involving Probabilities

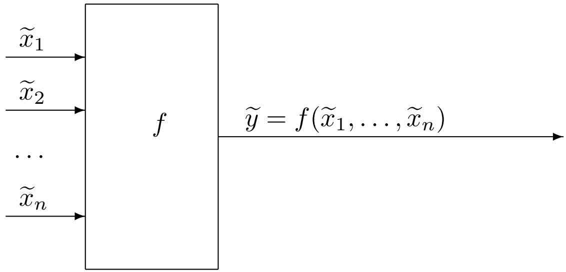
Vladik Kreinovich
Department of Computer Science
University of Texas at El Paso, 500 W. University
El Paso, TX 79968, USA, vladik@utep.edu

http://www.cs.utep.edu/vladik
http://www.cs.utep.edu/interval-comp

1. General Problem of Data Processing under Uncertainty

- *Indirect measurements:* way to measure y that are are difficult (or even impossible) to measure directly.

- *Idea:* $y = f(x_1, \ldots, x_n)$

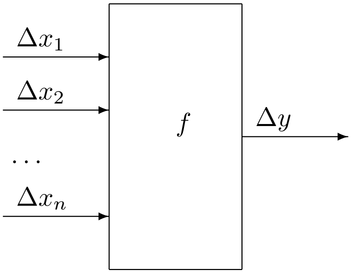


- *Problem:* measurements are never 100% accurate: $\widetilde{x}_i \neq x_i$ ($\Delta x_i \neq 0$) hence

$$\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n) \neq y = f(x_1, \ldots, y_n).$$

What are bounds on $\Delta y \stackrel{\text{def}}{=} \widetilde{y} - y$?

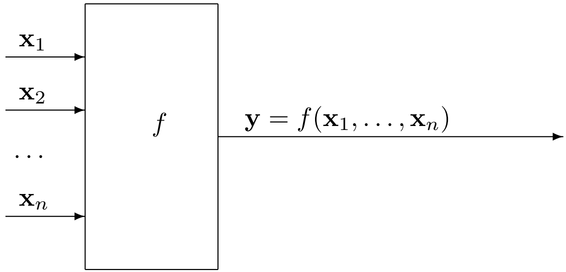
2. Probabilistic and Interval Uncertainty



- *Traditional approach:* we know probability distribution for Δx_i (usually Gaussian).
- *Where it comes from:* calibration using standard MI.
- *Problem:* sometimes we do not know the distribution because no "standard" (more accurate) MI is available. Cases:
  - fundamental science
  - manufacturing
- *Solution:* we know upper bounds Δ_i on $|\Delta x_i|$ hence

$$x_i \in [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i].$$

3. Interval Computations: A Problem



- *Given:*
  - an algorithm $y = f(x_1, \dots, x_n)$ that transforms n real numbers x_i into a number y;
  - n intervals $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$.
- *Compute:* the corresponding range of y:

$$[\underline{y}, \overline{y}] = \{f(x_1, \dots, x_n) \mid x_1 \in [\underline{x}_1, \overline{x}_1], \dots, x_n \in [\underline{x}_n, \overline{x}_n]\}.$$

- *Fact:* even for quadratic f, the problem of computing the exact range $\mathbf{y}$ is NP-hard.
- *Practical challenges:*
  - find classes of problems for which efficient algorithms are possible; and
  - for problems outside these classes, find efficient techniques for *approximating* uncertainty of y.

General Problem of . . .
General Challenge
General Approach: . . .
General Approach: . . .
General Approach: . . .
Successes (cont-d)
Challenges
Additional Challenges
Title Page
◀◀ ▶▶
◀ ▶
Page 4 of 28
Go Back
Full Screen
Close
Quit

4. Why Not Maximum Entropy?

- *Situation:* in many practical applications, it is very difficult to come up with the probabilities.
- *Traditional engineering approach:* use probabilistic techniques.
- *Problem:* many different probability distributions are consistent with the same observations.
- *Solution:* select one of these distributions – e.g., the one with the largest entropy.
- *Example – single variable:* if all we know is that $x \in [\underline{x}, \overline{x}]$, then MaxEnt leads to a uniform distribution on $[\underline{x}, \overline{x}]$.
- *Example – multiple variables:* different variables are independently distributed.
- *Conclusion:* if $\Delta y = \Delta x_1 + \ldots + \Delta x_n$, with $\Delta x_i \in [-\Delta_i, \Delta_i]$, then due to Central Limit Theorem, Δy is almost normal, with $\sigma = \frac{1}{\sqrt{3}} \cdot \sqrt{\sum_{i=1}^{n} \Delta_i^2}$.
- *Why this may be inadequate:* when $\Delta_i = \Delta$, we get $\Delta \sim \sqrt{n}$, but due to correlation, it is possible that $\Delta = n \cdot \Delta_i \sim n \gg \sqrt{n}$.
- *Conclusion:* using a single distribution can be very misleading, especially if we want guaranteed results – e.g., in high-risk application areas such as space exploration or nuclear engineering.

5. Chip Design: Case Study When Intervals Are Not Enough

- *One of the main objectives:* decrease the chip's clock cycle D.
- *Conclusion:* it is therefore important to estimate the clock cycle on the design stage.
- *Formula – idea:* D is the maximum delay over all possible paths $D \stackrel{\text{def}}{=} \max(D_1, \ldots, D_N)$, where D_i is the sum of the delays corresponding to the gates and wires along this path.
- *Formula – details:* each D_i depends on factors $x_1, \ldots, x_n$ – variation caused by the current design practices, environmental design characteristics (e.g., variations in temperature and in in supply voltage), etc. –

$$D_i = a_i + \sum_{j=1}^{n} a_{ij} \cdot x_j, \text{ so } D = \max_i \left(a_i + \sum_{j=1}^{n} a_{ij} \cdot x_j \right).$$

- *Traditional approach to estimating D:* worst-case (interval) analysis.
- *Result:* over-estimation up to 30% above the observed clock time, so chips are over-designed and under-performing.
- *Reason:* factors x_i are independent, so the probability that all these factors are at their worst is extremely small.
- *Challenge:* take into account the probabilistic character of the factor variations.

6. General Approach: Interval-Type Step-by-Step Techniques

- *Problem:*
- *Solution:* compute an enclosure $\mathbf{Y}$ such that $\mathbf{y} \subseteq \mathbf{Y}$.
- *Interval arithmetic:* for arithmetic operations $f(x_1, x_2)$, we have explicit formulas for the range.
- *Examples:* when $x_1 \in \mathbf{x}_1 = [\underline{x}_1, \overline{x}_1]$ and $x_2 \in \mathbf{x}_2 = [\underline{x}_2, \overline{x}_2]$, then:
  - The range $\mathbf{x}_1 + \mathbf{x}_2$ for $x_1 + x_2$ is $[\underline{x}_1 + \underline{x}_2, \overline{x}_1 + \overline{x}_2]$.
  - The range $\mathbf{x}_1 - \mathbf{x}_2$ for $x_1 - x_2$ is $[\underline{x}_1 - \overline{x}_2, \overline{x}_1 - \underline{x}_2]$.
  - The range $\mathbf{x}_1 \cdot \mathbf{x}_2$ for $x_1 \cdot x_2$ is $[\underline{y}, \overline{y}]$, where

$$\underline{y} = \min(\underline{x}_1 \cdot \underline{x}_2, \underline{x}_1 \cdot \overline{x}_2, \overline{x}_1 \cdot \underline{x}_2, \overline{x}_1 \cdot \overline{x}_2);$$

$$\overline{y} = \max(\underline{x}_1 \cdot \underline{x}_2, \underline{x}_1 \cdot \overline{x}_2, \overline{x}_1 \cdot \underline{x}_2, \overline{x}_1 \cdot \overline{x}_2).$$

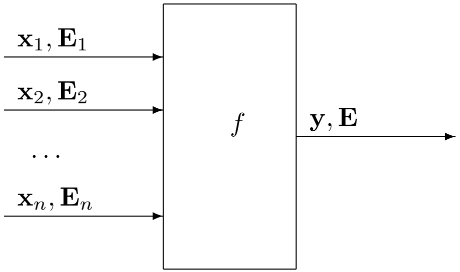
- The range $1/\mathbf{x}_1$ for $1/x_1$ is $[1/\overline{x}_1, 1/\underline{x}_1]$ (if $0 \notin \mathbf{x}_1$).

7. Interval Approach: Example

- *Example:* $f(x) = (x-2)\cdot(x+2)$, $x \in [1,2]$.
- How will the computer compute it?
  - $r_1 := x - 2$;
  - $r_2 := x + 2$;
  - $r_3 := r_1 \cdot r_2$.
- *Main idea:* do the same operations, but with *intervals* instead of *numbers*:
  - $\mathbf{r}_1 := [1,2] - [2,2] = [-1,0]$;
  - $\mathbf{r}_2 := [1,2] + [2,2] = [3,4]$;
  - $\mathbf{r}_3 := [-1,0]\cdot[3,4] = [-4,0]$.
- *Actual range:* $f(\mathbf{x}) = [-3,0]$.
- *Comment:* this is just a toy example, there are more efficient ways of computing an enclosure $\mathbf{Y} \supseteq \mathbf{y}$.

8. Extension of Interval Arithmetic to Probabilistic Case: Successes

- *Objective:* make decisions $E_x[u(x,a)] \to \max a$.
- For smooth $u(x)$, we have $u(x) = u(x_0) + (x - x_0) \cdot u'(x_0) + \ldots$, so we must know moments to estimate $E[u]$.
- For threshold-type $u(x)$, we need cdf $F(x) = \text{Prob}(\xi \le x)$.
- *General solution:* parse to elementary operations $+$, $-$, $\cdot$, $1/x$, max, min.
- Explicit formulas for arithmetic operations known for intervals, for p-boxes $\mathbf{F}(x) = [\underline{F}(x), \overline{F}(x)]$, for intervals + 1st moments $E_i \stackrel{\text{def}}{=} E[x_i]$:



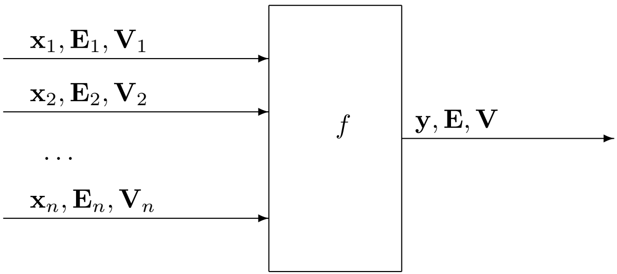
9. Successes (cont-d)

- *Easy cases:* $+$, $-$, product of independent x_i.
- *Example of a non-trivial case:* multiplication $y = x_1 \cdot x_2$, when we have no information about the correlation:
  - $\underline{E} = \max(p_1 + p_2 - 1, 0) \cdot \overline{x}_1 \cdot \overline{x}_2 + \min(p_1, 1 - p_2) \cdot \overline{x}_1 \cdot \underline{x}_2 +$ $\min(1 - p_1, p_2) \cdot \underline{x}_1 \cdot \overline{x}_2 + \max(1 - p_1 - p_2, 0) \cdot \underline{x}_1 \cdot \underline{x}_2;$
  - $\overline{E} = \min(p_1, p_2) \cdot \overline{x}_1 \cdot \overline{x}_2 + \max(p_1 - p_2, 0) \cdot \overline{x}_1 \cdot \underline{x}_2 +$ $\max(p_2 - p_1, 0) \cdot \underline{x}_1 \cdot \overline{x}_2 + \min(1 - p_1, 1 - p_2) \cdot \underline{x}_1 \cdot \underline{x}_2,$

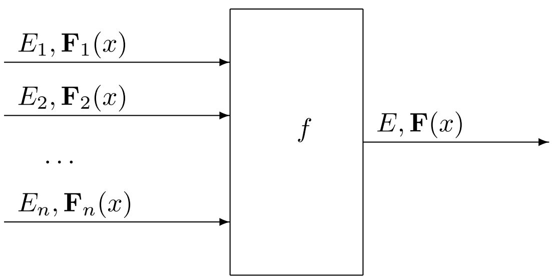
  where $p_i \stackrel{\text{def}}{=} (E_i - \underline{x}_i)/(\overline{x}_i - \underline{x}_i)$.

10. Challenges

- intervals + 2nd moments:



- moments + p-boxes; e.g.:



11. Problem

- *Result of interval-type approach:* over-estimation practically as bad as with interval computations.
- *Good news:* for $D_i = a_i + \sum a_{ij} \cdot x_j$, we use independence of x_i and get reasonable p-boxes.
- *Bad news:* the values D_i depends on same factors, so they are not independent.
- *Analogy:* this is similar to dependence-caused excess width in interval computations.
- *In interval computations:* methods beyond straightforward interval computations – centroid, affine, bisection – decrease excess width.
- *What we have done so far:* extended interval arithmetic to the probabilistic case.
- *What we need:* extend state-of-the-art interval computations techniques to the probabilistic case.

12. Main Idea: Use Moments

- *What we want:* find D_0 s.t. $D \le D_0$ with the probability $\ge 1 - \varepsilon$ (where $\varepsilon > 0$ is a given small probability).
- *Traditional statistical analysis:* compute moments $M_v \stackrel{\text{def}}{=} E[D^v]$, $v = 1, 2, \ldots$
- *From moments to p-boxes – guaranteed:* Chebyshev inequality

$$\text{Prob}(|D - M_1| > k_0 \cdot \sigma) \le 1/k_0^2,$$

  where $\sigma \stackrel{\text{def}}{=} \sqrt{V} = \sqrt{M_2 - M_1^2}$.
- *Example:* for $\varepsilon = 10^{-3}$, we need $D_0 = E + 30\sigma$.
- *Problem:* D is often almost normal, so $D_0 \approx E + 3\sigma$ – excess width.
- *Idea:* higher moments $D_0 = M_1 + k_{2q} \cdot \sigma_{2q}$ with $\sigma_{2q} = C_{2q}^{1/q}$ and $k_{2q} = \varepsilon^{-1/(2q)}$.
- *Example:* for $\varepsilon = 10^{-3}$, $k_2 \approx 30$, $k_4 \approx 5.5$, $k_6 \approx 3$.
- *Central moment:* $C_4 = E[(D - M_1)^4] = M_4 - 4 \cdot M_3 \cdot M_1 + 6 \cdot M_2 \cdot M_1^2 - 3 \cdot M_1^4$.
- *Interval uncertainty:* $D_0 = \overline{M}_1 + k_{2q} \cdot \overline{(C_{2q})}^{1/q}$, where

$$\overline{C}_4 = \overline{M}_4 - 4 \cdot \underline{M}_3 \cdot \underline{M}_1 + 6 \cdot \overline{M}_2 \cdot \overline{M}_1^2 - 3 \cdot \underline{M}_1^4.$$

13. Formulation of the Problem: Convex Case

GIVEN:
- natural numbers n, $k \le n$, and $v \ge 1$;
- a function $y = F(x_1, \ldots, x_n)$ (algorithmically defined) such that for every combination of values $x_{k+1}, \ldots, x_n$, the dependence of y on $x_1, \ldots, x_k$ is convex;
- $n-k$ probability distributions $x_{k+1}, \ldots, x_n$ – e.g., given in the form of cumulative distribution function (cdf) $F_j(x)$, $k+1 \le j \le n$;
- k intervals $\mathbf{x}_1, \ldots, \mathbf{x}_k$, and
- k values $E_1, \ldots, E_k$.

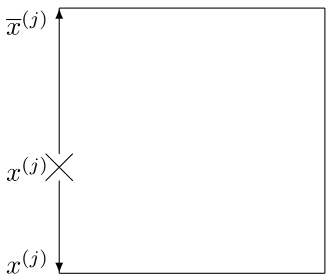
such that for every $x_1 \in [\underline{x}_1, \overline{x}_1], \ldots, x_k \in [\underline{x}_k, \overline{x}_k]$, we have $F(x_1, \ldots, x_n) \ge 0$ with probability 1.

TAKE: all possible joint probability distributions on R^n for which:
- all n random variables are independent;
- for each j from 1 to k, $x_j \in \mathbf{x}_j$ with probability 1 and the mean value of x_j is equal E_j;
- for $j > k$, the variable x_j has a given distribution $F_j(x)$.

FIND: for the variable $y = F(x_1, \ldots, x_n)$, find the set $\mathbf{M}_v = [\underline{M}_v, \overline{M}_v]$ of all possible values of $M_v \stackrel{\text{def}}{=} E[y^v]$ for all such distributions.

14. Result

- The *smallest* possible value $\underline{M}_v$ is attained when for each j from 1 to k, we use a 1-point distribution in which $x_j = E_j$ with probability 1.
- The *largest* possible values $\overline{M}_v$ is attained when for each j from 1 to k, we use a 2-point distribution for x_j, in which:
  - $x_j = \underline{x}_j$ with probability $\underline{p}_j \stackrel{\text{def}}{=} \dfrac{\overline{x}_j - E_j}{\overline{x}_j - \underline{x}_j}$.
  - $x_j = \overline{x}_j$ with probability $\overline{p}_j \stackrel{\text{def}}{=} \dfrac{E_j - \underline{x}_j}{\overline{x}_j - \underline{x}_j}$.
- *Main idea – transfer:* F is convex and $F \geq 0$, hence F^v is convex.



- *Algorithm:* Monte-Carlo simulations.
- *Results:* much smaller excess width.
- *Additional result:* if we also know that each distribution is unimodal.

General Problem of . . .
General Challenge
General Approach: . . .
General Approach: . . .
General Approach: . . .
Successes (cont-d)
Challenges
Additional Challenges
Title Page
Page 15 of 28
Go Back
Full Screen
Close
Quit

15. Case Study: Bioinformatics

- *Practical problem:* find genetic difference between cancer cells and healthy cells.
- *Ideal case:* we directly measure concentration c of the gene in cancer cells and h in healthy cells.
- *In reality:* difficult to separate, so we measure $y_i \approx x_i \cdot c + (1 - x_i) \cdot h$, where x_i is the percentage of cancer cells in i-th sample.
- *Equivalent form:* $a \cdot x_i + h \approx y_i$, where $a \stackrel{\text{def}}{=} c - h$.
- *If we know x_i exactly:* Least Squares Method $\sum\limits_{i=1}^{n} (a \cdot x_i + h - y_i)^2 \to \min\limits_{a,h}$,

  hence $a = \dfrac{C(x,y)}{V(x)}$ and $h = E(y) - a \cdot E(x)$, where $E(x) = \dfrac{1}{n} \cdot \sum\limits_{i=1}^{n} x_i$,

  $$V(x) = \frac{1}{n-1} \cdot \sum_{i=1}^{n} (x_i - E(x))^2, \;\; C(x,y) = \frac{1}{n-1} \cdot \sum_{i=1}^{n} (x_i - E(x)) \cdot (y_i - E(y)).$$
- *Interval uncertainty:* experts manually count x_i, and only provide interval bounds $\mathbf{x}_i$, e.g., $x_i \in [0.7, 0.8]$.
- *Fact:* different $x_i \in \mathbf{x}_i$ lead to different a and h.
- *Problem:* find the range of a and h corresponding to all possible values $x_i \in [\underline{x}_i, \overline{x}_i]$.

16. General Problem

- *General problem:* how to efficiently deduce the statistical information from, e.g., interval data.
- *Example:* we know intervals $\mathbf{x}_1 = [\underline{x}_1, \overline{x}_1], \ldots, \mathbf{x}_n = [\underline{x}_n, \overline{x}_n]$, we want to compute the ranges of possible values of the population mean $E(x) = \dfrac{1}{n}\sum\limits_{i=1}^{n} x_i$, population variance $V = \dfrac{1}{n}\sum\limits_{i=1}^{n}(x_i - E(x))^2$, etc.
- *Difficulty:* in general, this problem is NP-hard even for the variance.
- *Known:*
  - efficient algorithms for $\underline{V}$,
  - efficient algorithms for $\overline{V}$ for reasonable situations,
  - efficient algorithms for $C(x, y)$ when intervals comes from a partition, etc.
- *Bioinformatics case:* we find intervals for $C(x, y)$ and for $V(x)$ and divide.
- *Challenges:* finding the ranges of covariance, correlation, etc., in other situations

17. Case Study: Detecting Outliers

- In many application areas, it is important to detect *outliers*, i.e., unusual, abnormal values.
- In *medicine*, unusual values may indicate disease.
- In *geophysics*, abnormal values may indicate a mineral deposit (or an erroneous measurement result).
- In *structural integrity* testing, abnormal values may indicate faults in a structure.
- *Traditional engineering approach:* a new measurement result x is classified as an outlier if $x \notin [L, U]$, where

$$L \stackrel{\text{def}}{=} E - k_0 \cdot \sigma, \quad U \stackrel{\text{def}}{=} E + k_0 \cdot \sigma,$$

  and $k_0 > 1$ is pre-selected.
- *Comment:* most frequently, $k_0 = 2$, 3, or 6.

18. Outlier Detection Under Interval Uncertainty: A Problem

- In some practical situations, we only have intervals $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$.
- For different values $x_i \in \mathbf{x}_i$, we get different k_0-sigma intervals $[L, U]$.
- A *possible* outlier is a value outside *some* k_0-sigma interval.
- *Example:* structural integrity – not to miss a fault.
- A *guaranteed* outlier is a value outside *all* k_0-sigma intervals.
- *Example:* before a surgery, we want to make sure that there is a micro-calcification.
- A value x is a possible outlier if $x \notin [\overline{L}, \underline{U}]$.
- A value x is a guaranteed outlier if $x \notin [\underline{L}, \overline{U}]$.
- *Conclusion:* to detect outliers, we must know the ranges of $L = E - k_0 \cdot \sigma$ and $U = E + k_0 \cdot \sigma$.

19. Outlier Detection Under Interval Uncertainty: A Solution

- *We need:* to detect outliers, we must compute the ranges of $L = E - k_0 \cdot \sigma$ and $U = E + k_0 \cdot \sigma$.
- *We know:* how to compute the ranges $\mathbf{E}$ and $[\underline{\sigma}, \overline{\sigma}]$ for E and σ.
- *Possibility:* use interval computations to conclude that $L \in \mathbf{E} - k_0 \cdot [\underline{\sigma}, \overline{\sigma}]$ and $L \in \mathbf{E} + k_0 \cdot [\underline{\sigma}, \overline{\sigma}]$.
- *Problem:* the resulting intervals for L and U are *wider* than the actual ranges.
- *Reason:* E and σ use the same inputs $x_1, \ldots, x_n$ and are hence not independent from each other.
- *Practical consequence:* we miss some outliers.
- *Desirable:* compute *exact* ranges for L and U.
- *What we do:* exactly this.
- *Application:* detecting outliers in gravity measurements.

20. Acknowledgments

This work was supported in part:

- by NASA under cooperative agreement NCC5-209,
- by NSF grant EAR-0225670,
- by NIH grant 3T34GM008048-20S1,
- by Army Research Lab grant DATM-05-02-C-0046,
- by Star Award from the University of Texas System,
- by Texas Department of Transportation grant No. 0-5453, and
- by the workshop organizers.

21. Detecting Possible Outliers: Idea

- To detect possible outliers, we need $\overline{L}$ and $\underline{U}$.
- The minimum $\underline{U}$ of a smooth function U on an interval $[\underline{x}_i, \overline{x}_i]$ is attained:
  - either inside, when $\dfrac{\partial U}{\partial x_i} = 0$ – i.e., when

$$x_i = \mu \stackrel{\text{def}}{=} E - \alpha \cdot \sigma \text{ (where } \alpha \stackrel{\text{def}}{=} 1/k_0);$$

  - or at $x_i = \underline{x}_i$, when $\dfrac{\partial U}{\partial x_i} \geq 0$ – i.e., when $\mu \leq \underline{x}_i$;
  - or at $x_i = \overline{x}_i$, when $\dfrac{\partial U}{\partial x_i} \leq 0$ – i.e., when $\overline{x}_i \leq \mu$.
- Thus, once we know how μ is located w.r.t. all the intervals $\mathbf{x}_i$, we can find the optimal values of x_i.
- *Comment.* the value μ can be obtained from the condition $E - \alpha \cdot \sigma = \mu$.
- Hence, to find $\min U$, we analyze how the endpoints $\underline{x}_i$ and $\overline{x}_i$ divide the real line, consider all the resulting sub-intervals, and take the smallest U.

22. Computing $\underline{U}$: Algorithm

- First, sort all $2n$ values $\underline{x}_i$, $\overline{x}_i$ into a sequence $x_{(1)} \le x_{(2)} \le \ldots \le x_{(2n)}$; take $x_{(0)} \stackrel{\text{def}}{=} -\infty$, $x_{(2n+1)} \stackrel{\text{def}}{=} +\infty$.
- For each zone $[x_{(k)}, x_{(k+1)}]$, we compute the values
$$e_k \stackrel{\text{def}}{=} \sum_{i:\underline{x}_i \ge x_{(k+1)}} \underline{x}_i + \sum_{j:\overline{x}_j \le x_{(k)}} \overline{x}_j,$$
$$m_k \stackrel{\text{def}}{=} \sum_{i:\underline{x}_i \ge x_{(k+1)}} (\underline{x}_i)^2 + \sum_{j:\overline{x}_j \le x_{(k)}} (\overline{x}_j)^2,$$
and $n_k =$ the total number of such i's and j's.
- Solve equation $A - B \cdot \mu + C \cdot \mu^2 = 0$, where
$$A \stackrel{\text{def}}{=} e_k^2 \cdot (1 + \alpha^2) - \alpha^2 \cdot m_k \cdot n,$$
$$B \stackrel{\text{def}}{=} 2e_k \cdot \left((1 + \alpha^2) \cdot n_k - \alpha^2 \cdot n\right); \quad C \stackrel{\text{def}}{=} B \cdot \frac{n_k}{2e_k};$$
select $\mu \in$ zone for which $\mu \cdot n_k \le e_k$.
- $E_k \stackrel{\text{def}}{=} \frac{e_k}{n} + \frac{n - n_k}{n} \cdot \mu$, $M_k \stackrel{\text{def}}{=} \frac{m_k}{n} + \frac{n - n_k}{n} \cdot \mu^2$,
$U_k \stackrel{\text{def}}{=} E_k + k_0 \cdot \sqrt{M_k - (E_k)^2}$.
- $\underline{U}$ is the smallest of these values U_k.

23. Computing $\overline{L}$: Algorithm

- First, sort all $2n$ values $\underline{x}_i$, $\overline{x}_i$ into a sequence $x_{(1)} \le x_{(2)} \le \ldots \le x_{(2n)}$; take $x_{(0)} \stackrel{\rm def}{=} -\infty$, $x_{(2n+1)} \stackrel{\rm def}{=} +\infty$.
- For each zone $[x_{(k)}, x_{(k+1)}]$, we compute the values

$$e_k \stackrel{\rm def}{=} \sum_{i:\underline{x}_i \ge x_{(k+1)}} \underline{x}_i + \sum_{j:\overline{x}_j \le x_{(k)}} \overline{x}_j,$$

$$m_k \stackrel{\rm def}{=} \sum_{i:\underline{x}_i \ge x_{(k+1)}} (\underline{x}_i)^2 + \sum_{j:\overline{x}_j \le x_{(k)}} (\overline{x}_j)^2,$$

  and $n_k =$ the total number of such i's and j's.

- Solve equation $A - B \cdot \mu + C \cdot \mu^2 = 0$, where

$$A \stackrel{\rm def}{=} e_k^2 \cdot (1 + \alpha^2) - \alpha^2 \cdot m_k \cdot n,$$

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

  select $\mu \in$ zone for which $\mu \cdot n_k \ge e_k$.

- $E_k \stackrel{\rm def}{=} \dfrac{e_k}{n} + \dfrac{n - n_k}{n} \cdot \mu$, $\;M_k \stackrel{\rm def}{=} \dfrac{m_k}{n} + \dfrac{n - n_k}{n} \cdot \mu^2$,
  $L_k \stackrel{\rm def}{=} E_k - k_0 \cdot \sqrt{M_k - (E_k)^2}$.
- $\overline{L}$ is the largest of these values L_k.

24. Computational Complexity of Outlier Detection

- *Detecting possible outliers:* The above algorithm $\underline{\mathcal{A}}_U$ always computes $\underline{U}$ in quadratic time.
- *Detecting possible outliers:* The above algorithm $\overline{\mathcal{A}}_L$ always computes $\overline{L}$ in quadratic time.
- *Detecting guaranteed outliers:* 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.
- *Detecting guaranteed outliers:* 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 known before.

25. How Can We Actually Detect Guaranteed Outliers?

- *1st result:* if $1+(1/k_0)^2 < n$, then $\max U$ and $\min L$ are attained at endpoints of $\mathbf{x}_i$.
- *Example:* $k_0 > 1$ and $n \ge 2$.
- *Resulting algorithm:* test all 2^n combinations of values $\underline{x}_i$ and $\overline{x}_i$.
- *Important case:* often, measured values $\widetilde{x}_i$ are definitely different from each other, in the sense that the "narrowed" intervals

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

  do not intersect with each other.
- *Slightly more general case:* for some C, no more than C "narrowed" intervals can have a common point.

26. Computing $\overline{U}$

- Sort all endpoints of the narrowed intervals into a sequence $x_{(1)} \le x_{(2)} \le \ldots \le x_{(2n)}$, with $x_{(0)} \stackrel{\text{def}}{=} -\infty$, $x_{(2n+1)} \stackrel{\text{def}}{=} +\infty$.
- For each zone $[x_{(i)}, x_{(i+1)}]$, for each j, pick x_j:
  - if $x_{(i+1)} < \widetilde{x}_j - \dfrac{1+\alpha^2}{n} \cdot \Delta_j$, pick $x_j = \overline{x}_j$;
  - if $x_{(i+1)} > \widetilde{x}_j + \dfrac{1+\alpha^2}{n} \cdot \Delta_j$, pick $x_j = \underline{x}_j$;
  - for all other j, consider both $x_j = \overline{x}_j$ and $x_j = \underline{x}_j$.
- We get $\le 2^C$ sequences of x_j for each zone.
- For each sequence x_j, check whether $E - \alpha \cdot \sigma$ is within the zone.
- If $E - \alpha \cdot \sigma \in$ zone, compute $U \stackrel{\text{def}}{=} E + k_0 \cdot \sigma$.
- Finally, we return the largest of the computed values U as $\overline{U}$.

27. Computing $\underline{L}$

- Sort all endpoints of the narrowed intervals into a sequence $x_{(1)} \le x_{(2)} \le \ldots \le x_{(2n)}$, with $x_{(0)} \stackrel{\text{def}}{=} -\infty$, $x_{(2n+1)} \stackrel{\text{def}}{=} +\infty$.
- For each zone $[x_{(i)}, x_{(i+1)}]$, for each j, pick x_j:
  - if $x_{(i+1)} < \widetilde{x}_j - \dfrac{1+\alpha^2}{n} \cdot \Delta_j$, pick $x_j = \overline{x}_j$;
  - if $x_{(i+1)} > \widetilde{x}_j + \dfrac{1+\alpha^2}{n} \cdot \Delta_j$, pick $x_j = \underline{x}_j$;
  - for all other j, consider both $x_j = \overline{x}_j$ and $x_j = \underline{x}_j$.
- We get $\le 2^C$ sequences of x_j for each zone.
- For each sequence x_j, check whether $E + \alpha \cdot \sigma$ is within the zone.
- If $E + \alpha \cdot \sigma \in$ zone, compute $L \stackrel{\text{def}}{=} E - k_0 \cdot \sigma$.
- Finally, we return the smallest of the computed values L as $\underline{L}$.