Probabilities, Intervals, What Next? Algorithmic Problems Related to Uncertainty in Data Processing

Vladik Kreinovich

Computer Science Department

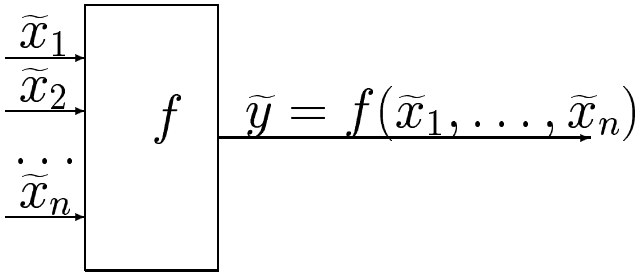
University of Texas at El Paso

El Paso, TX 79968, USA

vladik@cs.utep.edu

Why Data Processing

- *Indirect measurements:* way to measure y that are are impossible or difficult to measure directly.
- *Examples:* distance to a star, the amount of oil in a given well.
- *Idea:* $y = f(x_1, \ldots, x_n)$

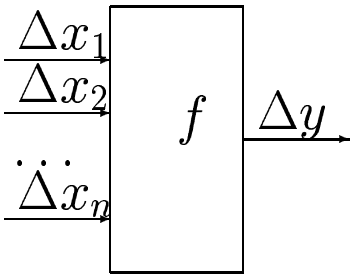


- *Problem:* measurements are never 100% accurate: $\widetilde{x}_i \neq x_i$ ($\Delta x_i \neq 0$) hence

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

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

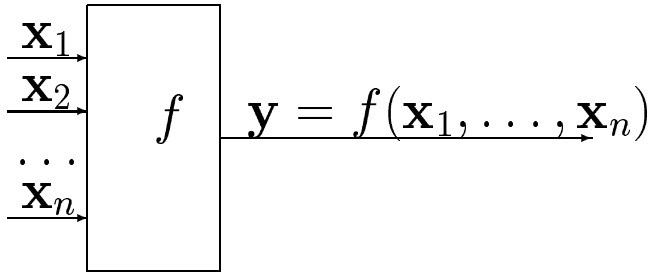
Why Interval Computations: Reminder



- *Traditional approach:* we know probability distribution for Δx_i (usually Gaussian).
- *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

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

Interval Computations: What? How?



- *What:*

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

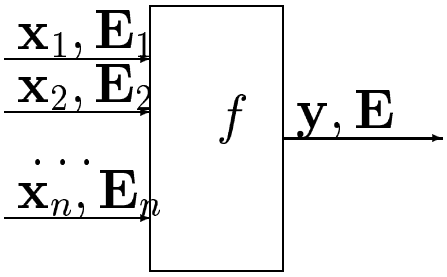
- *How* (straightforward interval computations):
  - parse f into elementary operations $+$, $-$, $\cdot$, $1/x$, min, max;
  - replace each operation by the corresponding operation of interval arithmetic:

$$[\underline{x}_1, \overline{x}_1] + [\underline{x}_2, \overline{x}_1] = [\underline{x}_1 + \underline{x}_2, \overline{x}_1 + \overline{x}_2];$$

$$[\underline{x}_1, \overline{x}_1] - [\underline{x}_2, \overline{x}_1] = [\underline{x}_1 - \overline{x}_2, \overline{x}_1 - \underline{x}_2].$$

Adding Moments: Step One

- So far, we have considered two cases:
  - *statistical case:* we know Prob(Δx_i);
  - *interval case:* we know nothing about Prob(Δx_i).
- *Possible:* we have *partial* information about Prob(Δx_i).
- *Example:* we know moments.
- *Simplest case:* we know $E_i \stackrel{\text{def}}{=} E[x_i]$ (or rather $\mathbf{E}_i$).
- *Problem:*



- *Solution:* parse to $+$, $-$, $\cdot$, $1/x$, max, min.

Problem: Formulation, Cases

- *Given:*
  - $[\underline{x}_1, \overline{x}_1]$, $[\underline{E}_1, \overline{E}_1]$,
  - $[\underline{x}_2, \overline{x}_2]$, $[\underline{E}_2, \overline{E}_2]$,
  - an operation $y = x_1 \odot x_2$ ($\odot = +, -, \cdot, 1/x, \max, \min$).
- *Find:* exact bounds on $[\underline{y}, \overline{y}]$ and $[\underline{E}, \overline{E}]$.
- *Comment:* bounds on $[\underline{y}, \overline{y}]$ same.
- *Cases:*
  - we have no info about correlation between x_i;
  - we know that x_i are independent;
  - we know that x_i are maximally $+$ correlated:

$$\exists t \text{ s.t. } x_1(t) \uparrow \ \& \, x_2(t) \uparrow;$$

  - we know that x_i are maximally $-$ correlated:

$$\exists t \text{ s.t. } x_1(t) \uparrow \ \& \, x_2(t) \downarrow .$$

Formulation of the problem in Precise Terms

- *Given:* values $\underline{x}_1$, $\overline{x}_1$, $\underline{x}_2$, $\overline{x}_2$, $\underline{E}_1$, $\overline{E}_1$, $\underline{E}_2$, $\overline{E}_2$, and operation $\odot$.
- *Find:* the values

$$\underline{E} \stackrel{\text{def}}{=} \min\{E(x_1 \odot x_2) \mid \text{ all distributions of } (x_1, x_2)$$

$$\text{for which } x_1 \in [\underline{x}_1, \overline{x}_1], x_2 \in [\underline{x}_2, \overline{x}_2],$$

$$E[x_1] \in [\underline{E}_1, \overline{E}_1], E[x_2] \in [\underline{E}_2, \overline{E}_2]\}$$

and

$$\overline{E} \stackrel{\text{def}}{=} \max\{E(x_1 \odot x_2) \mid \text{ all distributions of } (x_1, x_2)$$

$$\text{for which } x_1 \in [\underline{x}_1, \overline{x}_1], x_2 \in [\underline{x}_2, \overline{x}_2],$$

$$E[x_1] \in [\underline{E}_1, \overline{E}_1], E[x_2] \in [\underline{E}_2, \overline{E}_2]\}$$

(plus restrictions on the correlation).

Simplest Cases: +, − (All 4 Cases), and Product of Independent x_i

- *Addition:* we know that

$$E[x_1 + x_2] = E[x_1] + E[x_2],$$

so

$$[\underline{E}, \overline{E}] = [\underline{E}_1 + \underline{E}_2, \overline{E}_1 + \overline{E}_2]$$

(in all 4 cases).

- *Subtraction:* similarly,

$$E[x_1 - x_2] = E[x_1] - E[x_2],$$

so

$$[\underline{E}, \overline{E}] = [\underline{E}_1 - \overline{E}_2, \overline{E}_1 - \underline{E}_2].$$

(in all 4 cases).

- *Product, independent x_i:*

here, $E[x_1 \cdot x_2] = E[x_1] \cdot E[x_2]$, hence

$$\mathbf{E} = \mathbf{E}_1 \cdot \mathbf{E}_2.$$

Product – Case When We Have No Info About Correlation: Theorem

Theorem. For multiplication $y = x_1 \cdot x_2$, when we have no information about the correlation,

$$\underline{E} = \max(p_1 + p_2 - 1, 0) \cdot \overline{x}_1 \cdot \overline{x}_2 +$$

$$\min(p_1, 1 - p_2) \cdot \overline{x}_1 \cdot \underline{x}_2 +$$

$$\min(1 - p_1, p_2) \cdot \underline{x}_1 \cdot \overline{x}_2 +$$

$$\max(1 - p_1 - p_2, 0) \cdot \underline{x}_1 \cdot \underline{x}_2;$$

and

$$\overline{E} = \min(p_1, p_2) \cdot \overline{x}_1 \cdot \overline{x}_2 +$$

$$\max(p_1 - p_2, 0) \cdot \overline{x}_1 \cdot \underline{x}_2 +$$

$$\max(p_2 - p_1, 0) \cdot \underline{x}_1 \cdot \overline{x}_2 +$$

$$\min(1 - p_1, 1 - p_2) \cdot \underline{x}_1 \cdot \underline{x}_2,$$

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

Meaning of the Theorem

- What are p_i: if we only allow values $\underline{x}_i$ and $\overline{x}_i$, then p_i is $p[\overline{x}_i]$ for which average is E_i; then $p[\underline{x}_i] = 1 - p_i$.
- If we know $p(A)$ and $p(B)$, then $p(A \,\&\, B)$ can take any values:
  - from $\underline{p}(A \,\&\, B) \stackrel{\text{def}}{=} \max(p(A) + p(B) - 1, 0)$
  - to $\overline{p}(A \,\&\, B) \stackrel{\text{def}}{=} \min(p(A), p(B))$;
- Hence,

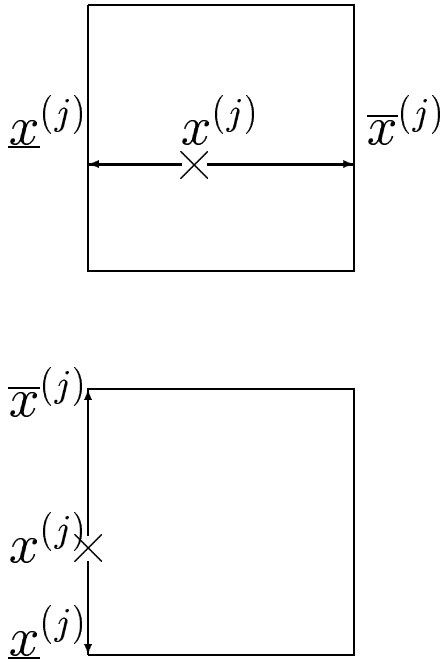
$$\underline{E} = \underline{p}[\overline{x}_1 \,\&\, \overline{x}_2] \cdot \overline{x}_1 \cdot \overline{x}_2 + \overline{p}[\overline{x}_1 \,\&\, \underline{x}_2] \cdot \overline{x}_1 \cdot \underline{x}_2 +$$

$$\overline{p}[\underline{x}_1 \,\&\, \overline{x}_2] \cdot \underline{x}_1 \cdot \overline{x}_2 + \underline{p}[\underline{x}_1 \,\&\, \underline{x}_2] \cdot \underline{x}_1 \cdot \underline{x}_2;$$

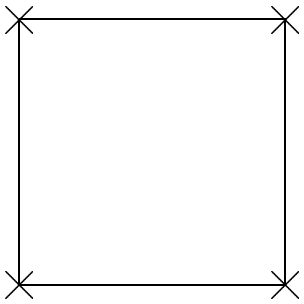
$$\overline{E} = \overline{p}[\overline{x}_1 \,\&\, \overline{x}_2] \cdot \overline{x}_1 \cdot \overline{x}_2 + \underline{p}[\overline{x}_1 \,\&\, \underline{x}_2] \cdot \overline{x}_1 \cdot \underline{x}_2 +$$

$$\underline{p}[\underline{x}_1 \,\&\, \overline{x}_2] \cdot \underline{x}_1 \cdot \overline{x}_2 + \overline{p}[\underline{x}_1 \,\&\, \underline{x}_2] \cdot \underline{x}_1 \cdot \underline{x}_2.$$

Proof: Main Idea



Thus, instead of considering all possible distributions, it is sufficient to consider only distributions for which $x_1 \in \{\underline{x}_1, \overline{x}_1\}$ and $x_2 \in \{\underline{x}_2, \overline{x}_2\}$:



Further Results

- Similar results are given:
  - correlation cases;
  - for the case when we have non-degenerate intervals $\mathbf{E}_i$.
  - for other elementary arithmetic operations ($1/x$, min, max);
- Similar ideas can be used:
  - for more general operations;
  - for the case when we know 2nd moments in addition to the 1st moments.

Formulation of the Auxiliary Problem

- We have n measurement results $x_1, \ldots, x_n$,
- Traditional data processing techniques: compute population parameters, e.g.,

$$\mu = \frac{x_1 + \ldots + x_n}{n},$$

$$\sigma^2 = \frac{(x_1 - \mu)^2 + \ldots + (x_n - \mu)^2}{n} \ (\text{or } \sigma = \sqrt{\sigma^2}).$$

- Often, we only have intervals $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$.
- *Example:* for measurements, $\mathbf{x}_i = [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$.
- We need $\mathbf{y} = \{f(x_1, \ldots, x_n) \,|\, x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n\}$.
- What are $[\underline{\mu}, \overline{\mu}]$ and $[\underline{\sigma^2}, \overline{\sigma^2}]$?
- For $[\underline{\mu}, \overline{\mu}]$, the answer is easy.
- When $\cap_{i=1}^{n} \mathbf{x}_i \neq \emptyset$, we have $\underline{\sigma^2} = 0$; else $\underline{\sigma^2} > 0$.
- *Problem* (Walster): what is the total set $[\underline{\sigma^2}, \overline{\sigma^2}]$ of possible values of σ^2?

For this Problem, Straightforward Interval Computations Sometimes Lead to Excess Width

- *Reminder:*
  - parse the function $f(x_1, \ldots, x_n)$, and
  - replace each elementary operation by the corr. operation of interval arithmetic.
- *Example:* for $\mathbf{x}_1 = \mathbf{x}_2 = [0, 1]$.
- *Actual range:* since $\sigma^2 = (x_1 - x_2)^2/4$, the actual range is $[\underline{\sigma^2}, \overline{\sigma^2}] = [0, 0.25]$.
- *Estimate:* $[\underline{\mu}, \overline{\mu}] = [0, 1]$, hence
$$\frac{(\mathbf{x}_1 - [\underline{\mu}, \overline{\mu}])^2 + (\mathbf{x}_2 - [\underline{\mu}, \overline{\mu}])^2}{2} = [0, 1] \supset [0, 0.25].$$
- *Comment:* other formulas also lead to excess width.
- *Explanation:* in each formula for σ^2, each variable occurs several times.

Centered Form Sometimes Leads to Excess Width

- *Reminder:*

$$f(\mathbf{x}_1, \ldots, \mathbf{x}_n) \subseteq f(\widetilde{x}_1, \ldots, \widetilde{x}_n) + \sum_{i=1}^{n} \frac{\partial f}{\partial x_i}(\mathbf{x}_1, \ldots, \mathbf{x}_n) \cdot [-\Delta_i, \Delta_i],$$

  where:

  - $\widetilde{x}_i = (\underline{x}_i + \overline{x}_i)/2$ is the interval's midpoint and
  - $\Delta_i = (\underline{x}_i - \overline{x}_i)/2$ is its half-width.

- *Not perfect* (similar to Hertling):

  - it produces an interval centered at $f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$;
  - when all intervals $\mathbf{x}_i$ are equal, all midpoints $\widetilde{x}_i$ are the same;
  - hence the population variance $f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ is 0;
  - so, the estimate's lower bound is < 0, but $\sigma^2 \geq 0$.

First Result: Computing $\underline{\sigma^2}$

The following algorithm always compute $\underline{\sigma^2}$ in $O(n^2)$:

- First, we sort all $2n$ values $\underline{x}_i$, $\overline{x}_i$ into a sequence $x_{(1)} \le x_{(2)} \le \ldots \le x_{(2n)}$.
- Second, we compute $\underline{\mu}$ and $\overline{\mu}$ and select all "small intervals" $[x_{(k)}, x_{(k+1)}]$ that intersect with $[\underline{\mu}, \overline{\mu}]$.
- For each of the selected small intervals $[x_{(k)}, x_{(k+1)}]$, we compute the ratio $r_k = S_k/N_k$, where

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

  and N_k is the total number of such i's and j's.
- If $r_k \in [x_{(k)}, x_{(k+1)}]$, then we compute

$$\sigma'^2_k \stackrel{\text{def}}{=} \frac{1}{n} \cdot \left(\sum_{i:\underline{x}_i \ge x_{(k+1)}} (\underline{x}_i - r_k)^2 + \sum_{j:\overline{x}_j \le x_{(k)}} (\overline{x}_j - r_k)^2 \right).$$

  If $N_k = 0$, we take $\sigma'^2_k \stackrel{\text{def}}{=} 0$.
- Finally, we return the smallest of the values σ'^2_k as $\underline{\sigma^2}$.

Example

- Input: $\mathbf{x}_1 = [2.1, 2.6]$, $\mathbf{x}_2 = [2.0, 2.1]$, $\mathbf{x}_3 = [2.2, 2.9]$, $\mathbf{x}_4 = [2.5, 2.7]$, and $\mathbf{x}_5 = [2.4, 2.8]$.
- "small intervals": $[x_{(1)}, x_{(2)}] = [2.0, 2.1]$, $[2.1, 2.1]$, $[2.1, 2.2]$, $[2.2, 2.4]$, $[2.4, 2.5]$, $[2.5, 2.6]$, $[2.6, 2.7]$, $[2.7, 2.8]$, and $[2.8, 2.9]$.
- Population average $[\underline{\mu}, \overline{\mu}] = [2.24, 2.62]$, so we keep $[2.2, 2.4]$, $[2.4, 2.5]$, $[2.5, 2.6]$, $[2.6, 2.7]$. For these intervals:
  - $S_4 = 7.0$, $N_4 = 3$, so $r_4 = 2.333\ldots$;
  - $S_5 = 4.6$, $N_5 = 2$, so $r_5 = 2.3$;
  - $S_6 = 2.1$, $N_6 = 1$, so $r_6 = 2.1$;
  - $S_7 = 4.7$, $N_7 = 2$, so $r_7 = 2.35$.
- Only r_4 lies within the corresponding small interval.
- Here, $\sigma'^2_4 = 0.017333\ldots$, so $\underline{\sigma^2} = 0.017333\ldots$

Second Result: Computing $\overline{\sigma^2}$ is NP-Hard

- **Theorem.** *Computing $\overline{\sigma^2}$ is NP-hard.*
- *Comments:*
  - NP-hard means, crudely speaking, that there are no general ways for solving *all* particular cases of this problem in reasonable time.
  - NP-hardness of computing the range of a quadratic function was proven by Vavasis (1991).
  - By using peeling, we can compute $\overline{\sigma^2}$ in exponential time $O(2^n)$.
- *Natural question:* maybe the difficulty comes from the requirement that the range be computed exactly?
- **Theorem.** *For every $\varepsilon > 0$, the problem of computing $\overline{\sigma^2}$ with accuracy ε is NP-hard.*

Third Result:

A Feasible Algorithm

that Computes $\overline{\sigma^2}$

in Many Practical Situations

- *Case:* all midpoints ("measured values")

$$\widetilde{x}_i = \frac{\underline{x}_i + \overline{x}_i}{2}$$

  of the intervals

$$\mathbf{x}_i = [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$$

  are definitely different from each other.

- *Namely:* the "narrowed" intervals

$$\left[\widetilde{x}_i - \frac{\Delta_i}{n}, \widetilde{x}_i + \frac{\Delta_i}{n}\right]$$

  do not intersect with each other.

- In this case, there exists an algorithm computes $\overline{\sigma^2}$ in quadratic time.

Algorithm

- Sort $2n$ endpoints of narrowed intervals into $x_{(1)} \le x_{(2)} \le \ldots \le x_{(2n)}$.
- Thus, IR is divided into $2n+2$ segments ("small intervals") $[x_{(k)}, x_{(k+1)}]$.
- Select only "small intervals" $[x_{(k)}, x_{(k+1)}]$ that intersect with $[\underline{\mu}, \overline{\mu}]$; for each, pick x_i as follows:
  - if $x_{(k+1)} < \widetilde{x}_i - \Delta_i/n$, then we pick $x_i = \overline{x}_i$;
  - if $x_{(k)} > \widetilde{x}_i + \Delta_i/n$, then we pick $x_i = \underline{x}_i$;
  - for all other i, we consider both possible values $x_i = \overline{x}_i$ and $x_i = \underline{x}_i$.
- For each of the sequences x_i, we check whether the average E is indeed within this small interval, and if it is, compute the population variance.
- The largest of these population variances is $\overline{\sigma^2}$.

Third Result (cont-d)

- *Question:* what if two "narrowed" intervals have a common point?
- *Case:* let us fix k and consider all cases C_k in which no more than k "narrowed" intervals can have a common point.
- *Result:* $\forall k$, the above algorithm $\overline{\mathcal{A}}$ computes $\overline{\sigma^2}$ in quadratic time for all problems $\in C_k$.
- *Comments:*
  - Computation time t is quadratic in n.
  - However, t is exponential in k.
  - So, when $k \uparrow$, the algorithm $\overline{\mathcal{A}}$ requires more and more computation time.
  - In our proof of NP-hardness, we use the case when all n narrowed intervals have a common point.

Population Mean, Population Variance: What Next?

- *Population covariance*

$$C = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu_x) \cdot (y_i - \mu_y).$$

- *Result:* both computing $\overline{C}$ and computing $\underline{C}$ are NP-hard problems.
- *Population correlation*

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

- *Result:* both computing $\overline{\rho}$ and computing $\underline{\rho}$ are NP-hard problems.
- *Open problem:* design feasible algorithms that work in many practical cases.
- *Median:* feasible (since it is monotonic in x_i).
- *Open problem:* analyze other population parameters from this viewpoint.

Bounds for Sample Variance: Variant of the First Problem

- *We know:*
  - measurement results $\widetilde{x}_1, \ldots, \widetilde{x}_n$;
  - the accuracies Δ_i of each measurement;
  - hence, that the actual values x_i are within

$$\mathbf{x}_i \stackrel{\text{def}}{=} [\underline{x}_i, \overline{x}_i] = [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i].$$

  - that x_i are normally distributed, w/CDF $F_0\left(\frac{x-a}{\sigma}\right)$.
- *Question:* what are the possible values of a and σ?
- *Main idea:* Kolmogorov-Smirnov (KS) inequality implies (with probability $p \geq p_0$) that

$$|F(x) - F_{\text{sample}}(x)| \leq \Delta,$$

where $F_{\text{sample}}(x) = \frac{i}{n}$ for $x_{(i)} \leq x < x_{(i+1)}$.

Bounds for Sample Variance: Solution

- Due to KS, for every i, for some $x_i \in [\underline{x}_i, \overline{x}_i]$:

$$\frac{i}{n} - \Delta \le F_0\left(\frac{x_{(i)} - a}{\sigma}\right) \le \frac{i}{n} + \Delta.$$

- So,

$$\frac{l(x_i')}{n} - \Delta \le F_0\left(\frac{x_i' - a}{\sigma}\right) \le \frac{u(x_i')}{n} + \Delta,$$

where $l(x)$ is # of k s.t. $\overline{x}_k \le x$, $u(i)$ is # of k s.t. $\underline{x}_k \le x$, and $x_i' = \underline{x}_i$ or $x_i' = \overline{x}_i$.

- Hence,

$$F_0^{-1}\left(\frac{l(x_i')}{n} - \Delta\right) \le \frac{x_i' - a}{\sigma} \le \left(\frac{u(x_i')}{n} + \Delta\right).$$

- We get a system of linear inequalities for a and σ:

$$\sigma \cdot F_0^{-1}\left(\frac{l(x_i)}{n} - \Delta\right) \le x_i - a \le \sigma \cdot F_0^{-1}\left(\frac{u(x_i)}{n} + \Delta\right).$$

- So, we can use linear programming to find bounds on a and σ.

Detecting Outliers Is Important

- 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 to Outlier Detection

- First, we collect measurement results $x_1, \ldots, x_n$ corresponding to normal situations.
- Then, we compute the sample average

$$E \stackrel{\text{def}}{=} \frac{x_1 + \ldots + x_n}{n}$$

and the (sample) standard deviation $\sigma = \sqrt{V}$, where

$$V \stackrel{\text{def}}{=} \frac{(x_1 - E)^2 + \ldots + (x_n - E)^2}{n};$$

- 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.

Outlier Detection Under Fuzzy Uncertainty: A Problem

- *Traditional (crisp) approach:*
  - arbitrarily select k_0;
  - x is an outlier if $x \notin [E - k_0 \cdot \sigma, E + k_0 \cdot \sigma]$;
  - x is not an outlier if $x \notin [E - k_0 \cdot \sigma, E + k_0 \cdot \sigma]$.
- *More reasonable approach* – degree of outlier-ness:
  - if x is an outlier for a large k_0 (e.g., for $k_0 = 10$), this degree is large;
  - if x is an outlier only for small k_0 (e.g., for $k_0 = 2$), this degree is small.
- *Definition:* largest k_0 s.t. $x \notin [E - k_0 \cdot \sigma, E + k_0 \cdot \sigma]$.
- *Equivalent definition:* $r = |x - E|/\sigma$.

Outlier Detection Under Interval Uncertainty: A Problem

- In some practical situations, we only have intervals $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$.
- *Example:* value $\widetilde{x}_i$ measured by an instrument with measurement error $\leq \Delta_i$; then $x_i \in [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$.
- 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.

Outlier Detection Reformulated in Terms of Ranges

- Let $[\underline{L}, \overline{L}]$ and $[\underline{U}, \overline{U}]$ be ranges of L and U.
- A value x is *not* a possible outlier if $x \in \cap[L, U]$, i.e., if $x \in [\overline{L}, \underline{U}]$.
- Thus, a value x *is* a possible outlier if $x \notin [\overline{L}, \underline{U}]$.
- A value x is *not* a guaranteed outlier if $x \in \cup[L, U]$, i.e., if $x \in [\underline{L}, \overline{U}]$.
- Thus, a value x *is* a guaranteed outlier if $x \notin [\underline{L}, \overline{U}]$.
- In real life, we often have an interval $\mathbf{x}$ for x. Then:
  - x is a possible outlier if $\mathbf{x} \not\subseteq [\overline{L}, \underline{U}]$;
  - x is a guaranteed outlier if $\mathbf{x} \cap [\underline{L}, \overline{U}] = \emptyset$.
- *Conclusion:* to detect outliers, we must know the ranges of L and U.

What Was Known Before and Why It Is Not Enough

- *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:* previously, we have shown 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 will do:* exactly this.

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.

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 $\mu \cdot n_k \le e_k$.
- $E_k \stackrel{\text{def}}{=} \dfrac{e_k}{n} + \dfrac{n - n_k}{n} \cdot \mu$, $M_k \stackrel{\text{def}}{=} \dfrac{m_k}{n} + \dfrac{n - n_k}{n} \cdot \mu^2$, $U_k \stackrel{\text{def}}{=} E_k + k_0 \cdot \sqrt{M_k - (E_k)^2}$.
- $\underline{U}$ is the smallest of these values U_k.

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 $x_{(0)} \stackrel{\text{def}}{=} -\infty$, $x_{(2n+1)} \stackrel{\text{def}}{=} +\infty$.
- 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 $\mu \cdot n_k \ge e_k$.

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

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.

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 $n \geq 2$.
- *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.

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 $\overline{U}$.

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 $\underline{L}$.

Computational Complexity

- *1st result:* for the case when $\leq C$ narrowed intervals can have a common point, the above algorithm $\overline{\mathcal{A}}_U$ always computes $\overline{U}$ in quadratic time.
- *2nd result:* for the case when $\leq C$ narrowed intervals can have a common point, the above algorithm $\underline{\mathcal{A}}_L$ always computes $\underline{L}$ in quadratic time.
- *Comment:* the corresponding computation times are quadratic in n but grow exponentially with C.
- *Corollary:* when C grows, this algorithm requires more and more computation time.
- *Comment:* in the examples on which we prove NP-hardness, $n/2$ out of n narrowed intervals have a common point.

Fuzzy-Related Problem: Computing Degree of Outlier-Ness

- *Reminder:* this degree r is the largest k_0 s.t.

$$x \notin [E - k_0 \cdot \sigma, E + k_0 \cdot \sigma].$$

- *Equivalent definition:* $r = |x - E|/\sigma$.
- *Problem:*
  - in some practical situations, we only have intervals $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$ of possible values of x_i;
  - for different values $x_i \in \mathbf{x}_i$, we get different values of the degree r.
- *Computational problem:* compute the range $[\underline{r}, \overline{r}]$ of possible values of outlier-ness r,

Reduction to a Simpler Problem

- *Step 1: Motivation.* The value of $r = |x - E|/\sigma$ does not change if we take $x' = 0$ and $x'_i = x_i - x$.
- *Step 1: Reduction.* W.l.o.g, we take $x = 0$, and consider $r = |E|/\sigma$.
- *Step 2: Motivation.* The lower bound of r is when the ratio $1/r^2 = V/E^2$ is the largest, and vice versa.
- *Step 2: Reduction.* It is sufficient to find the interval of possible values of V/E^2.
- *Step 3: Motivation.* Since $V = M - E^2$, where

$$M \stackrel{\text{def}}{=} \frac{x_1^2 + \ldots + x_n^2}{n},$$

we have $V/E^2 = M/E^2 - 1$.

- *Step 3: Reduction.* It is sufficient to find the interval of possible values of $R \stackrel{\text{def}}{=} M/E^2$.

Computing $\underline{R}$: $O(n^2)$ Algorithm

- If all the original intervals have a common point, then the smallest value of V is 0, and $\underline{R} = 1$.
- Else, 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.
- If $\lambda_k \stackrel{\text{def}}{=} m_k/e_k$ is in the zone, we compute $R_k = M_k/E_k^2$, where

$$E_k \stackrel{\text{def}}{=} \frac{e_k + \lambda_k \cdot n_k}{n}; \quad M_k \stackrel{\text{def}}{=} \frac{m_k + \lambda_k^2 \cdot n_k}{n}.$$

- $\underline{R}$ is the smallest of these values R_k.

Computing $\overline{R}$.

- *Case:* No more than C "narrowed" intervals $[x_i^-, x_i^+]$ have a common point, where:

$$x_i^- \stackrel{\text{def}}{=} \frac{\widetilde{x}_i}{1 + \dfrac{\Delta_i}{\underline{E} \cdot n}}; \quad x_i^+ \stackrel{\text{def}}{=} \frac{\widetilde{x}_i}{1 - \dfrac{\Delta_i}{\underline{E} \cdot n}},$$

$$\underline{E} \stackrel{\text{def}}{=} \frac{\underline{x}_1 + \ldots + \underline{x}_n}{n}.$$

- *Algorithm:* sort $2n$ values $\underline{x}_i$ and $\overline{x}_i$ into a sequence.
- For each zone $[x_{(k)}, x_{(k+1)}]$, and for each x_i, we take:
  - $x_i = \underline{x}_i$ if $x_i^+ \le x_{(k)}$;
  - $x_i = \overline{x}_i$ if $x_i^- \ge x_{(k+1)}$;
  - both values $x_i = \underline{x}_i$ and $x_i = \overline{x}_i$ otherwise.
- For each resulting combination, we:
  - compute E and M; check if $M/E \in [x_{(k)}, x_{(k+1)}]$;
  - if yes, we compute M/E^2.
- The largest of thus computed M/E^2 is $\overline{R}$.

Conclusions

- In many applications, it's important to detect outliers.
- Traditional idea: $x \notin [E - k_0 \cdot \sigma, E + k_0 \cdot \sigma]$.
- We often have only interval ranges $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$.
- For different values $x_i \in \mathbf{x}_i$, we get different k_0-sigma intervals $[L, U]$.
- x a *guaranteed* outlier if outside *all* k_0-sigma intervals.
- x a *possible* outlier if outside *some* k_0-sigma interval.
- To detect guaranteed and possible outliers, we must thus be able to compute the *ranges* $\mathbf{L} = [\underline{L}, \overline{L}]$ and $\mathbf{U} = [\underline{U}, \overline{U}]$.
- We show that computing these ranges is, in general, NP-hard.
- We also provide efficient algorithms that compute these ranges under reasonable conditions.

Quantum Computing

Why? Because It Is Inevitable

- *General problem:* we need faster computers, faster algorithms.
- *Main restriction on computer speed:* communication speed is bounded by speed of light.
- *Conclusion:* to make computers faster, we must build smaller processing elements.
- *Current elements:* nm size, $10^1 - 10^3$ atoms.
- *Future elements:* size of an atom or a molecule.
- *Problem:* we need to take quantum effects into consideration.
- *Example:* uncertainty principle leads to quantum noise.
- *Lemonade out of lemon:* it turns out quantum effects can also speed up computations.

Quantum Computing: Why Theoretical Successes? and Why not Yet Practical Gain?

- *Classical physics:* bit – 0 or 1.
- *Quantum physics:* superposition principle

  $\alpha \cdot |0\rangle + \beta \cdot |1\rangle$, $|\alpha|^2 + |\beta|^2 = 1$ – qubit.
- *Consequences:*
  - a single (qu)bit can store a lot of information;
  - a (qu)bit gate can process a lot of information.
- *Example:* $O(\sqrt{N})$ search in an un-sorted list (Grover).
- *Example:* factoring in polynomial time (Shor).
- *In practice:* only quantum cryptography.
- *Why:* known quantum algorithms need lots of qubits.

Quantum Algorithms That We Use

- *Grover's quantum search:* finds an element a_i $(1 \le i \le N)$ with given property P in time $O(\sqrt{N})$.
- *Important:* no need to have a_i *a priori*.
- *Quantum counting* (Brassard): counts the number of a_i's that satisfy P in time $O(\sqrt{N})$.
- *Quantum minimization* (Dürr et al.): finds i for which $a_i = \min_j a_j$. Main idea–bisection:
  - Let $a_i \in [-M, M] \cap Z$; then $\min \in [-M, M]$.
  - Once we know $\min \in [\underline{M}, \overline{M}]$, we use quantum search with $P(a_i) \equiv a_i < m \stackrel{\text{def}}{=} (\underline{M} + \overline{M})/2$.
  - If yes, $\min \in [\underline{M}, m]$, else $\min \in [m, \overline{M}]$.
  - We stop when there is only one integer in $[\underline{M}, \overline{M}]$.
- *Quantum mean* (Grover): accuracy $1/M$ after M iterations (non-quantum Monte-Carlo: $1/\sqrt{M}$).

Quantum Algorithm for Probabilistic Analysis

- *Problem:*
  - *Given:* a data processing algorithm $f(x_1, \ldots, x_n)$, the mean values $\widetilde{x}_i$ and st. dev. σ_i of the inputs x_i.
  - *Compute:* st. dev. σ of the result $y = f(x_1, \ldots, x_n)$ of data processing.
- *Idea:* σ^2 is a mean of $(y - \widetilde{y})^2$, where $y = f(x_1, \ldots, x_n)$, and x_i is normal with mean $\widetilde{x}_i$ and st. dev. σ_i.
- *Speed-up:*
  - For accuracy $\varepsilon = 20\%$, Monte-Carlo needs $1/\varepsilon^2 \approx 25$ iterations.
  - For the same ε, quantum needs $1/\varepsilon \approx 5$ iterations.
  - Computing f may take a long time, so this 5 times speed-up is essential.

Quantum Algorithm for Interval Computations

- *Problem:*
  - *Given:* a data processing algorithm $f(x_1, \ldots, x_n)$, measured values $\widetilde{x}_i$, bounds Δ_i on $\widetilde{x}_i - x_i$.
  - *Compute:* interval $[\widetilde{y} - \Delta, \widetilde{y} + \Delta]$ of possible values of the result $y = f(x_1, \ldots, x_n)$ of data processing.
- *Idea:* use Cauchy distribution

$$\rho(x) \sim 1/(1 + (x - a)^2/\Delta^2).$$

- *When Δ_i are small:* if x_i are Cauchy-distributed with mean x_i and parameter Δ_i, then y is Cauchy distributed with parameter Δ.
- *Trick:* since $\sigma = \infty$, we compute st. dev. of $\mathrm{atan}((y - \widetilde{y})^2)$ – that depends on Δ.
- *Result:* drastic speed-up.

Quantum Algorithms for Interval Computations: General Case

- *NP-hard:* in general, computing interval range is NP-hard.
- *What it means:* there is no faster method that exhaustive search.
- *Specifically:* no faster method than testing all $N = (D/\delta)^n$ points on an n-dimensional grid, where:

  D is the domain's linear size, δ is the grid step,

  and finding the smallest and the largest of f's.
- *Quantum computing:* finds min and max in $\sqrt{N} = (D/\delta)^{n/2}$ calls to f.
- *Advantage:* double dim of the problem for which we are able to compute the desired uncertainty.

What If We Have Several Different Types of Uncertainty

- *Traditional engineering approach:* compute

$$E = \frac{x_1 + \ldots + x_n}{n},$$

$$V = \frac{(x_1 - E)^2 + \ldots + (x_n - E)^2}{n}.$$

- *Problem:* often, we only have intervals $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i]$ of possible values of x_i.
- *Hence:* we have $\mathbf{E} = [\underline{E}, \overline{E}]$ and $\mathbf{V} = [\underline{V}, \overline{V}]$.
- *How this problem is solved now:* E is monotonic, so $[\underline{E}, \overline{E}]$ is computed as:

$$\underline{E} = \frac{\underline{x}_1 + \ldots + \underline{x}_n}{n}; \quad \overline{E} = \frac{\overline{x}_1 + \ldots + \overline{x}_n}{n}.$$

  Computing $\mathbf{V}$*:* We can compute $\underline{V}$ (and, under reasonable assumptions, $\overline{V}$) in $O(n^2)$ time.
- *Remaining problem:* for large datasets, $O(n^2)$ is too long.

Quantum Speed-Up for $\underline{V}$

- Sort $2n$ values $\underline{x}_i$, $\overline{x}_i$ into $x_{(1)} \le x_{(2)} \le \ldots \le x_{(2n)}$.
- If $[x_{(k)}, x_{(k+1)}] \cap [\underline{E}, \overline{E}] \neq \emptyset$, compute $r_k = S_k / N_k$, where

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

  and N_k is the total number of such is and js.
- If $r_k \in [x_{(k)}, x_{(k+1)}]$, then we compute V_k' as

$$\frac{1}{n} \cdot \left(\sum_{i: \underline{x}_i \ge x_{(k+1)}} (\underline{x}_i - r_k)^2 + \sum_{j: \overline{x}_j \le x_{(k)}} (\overline{x}_j - r_k)^2 \right).$$

- Finally, we return the smallest of the values V_k' as $\underline{V}$.
- *Quantum speed-up:*
  - r_k is a mean, so Monte-Carlo needs $O(1)$ time;
  - similarly, we can compute V_k' in constant time;
  - computing $\min V_k'$ requires $O(\sqrt{n})$ steps.
- *Result:* $O(\sqrt{n})$ steps after sorting;

  overall, $O(n \cdot \log(n)) \ll O(n^2)$.

Quantum Speed-Up for $\overline{V}$

- *Case:* no more than C intervals $[\widetilde{x}_i - \Delta_i/n, \widetilde{x}_i + \Delta_i/n]$ have a common point.
- Sort values $\widetilde{x}_i - \Delta_i/n$, $\widetilde{x}_i + \Delta_i/n$, $-\infty$, and $+\infty$ into a sequence $x_{(0)} \le x_{(1)} \le \ldots \le x_{(2n+1)}$.
- If $[x_{(k)}, x_{(k+1)}] \cap [\underline{E}, \overline{E}] \neq \emptyset$, then for each i, we select:
  - if $x_{(k+1)} < \widetilde{x}_i - \Delta_i/n$, then $x_i = \overline{x}_i$;
  - if $x_{(k)} > \widetilde{x}_i + \Delta_i/n$, then $x_i = \underline{x}_i$;
  - for all other i, we consider both possible values $x_i = \overline{x}_i$ and $x_i = \underline{x}_i$.
- For each resulting sequyence of x_i, we check whether $E \in [x_{(k)}, x_{(k+1)}]$; if yes, we compute V.
- The largest of these V is $\overline{V}$.
- *Quantum speed-up:* computing E and V takes $O(1)$ time; selecting the largest V takes $O(\sqrt{n})$ time.

What If Not Enough Qubits?

- *Situation:* not enough qubits for quantum algorithms.
- *Problem:* can we use smaller quantum registers to get a partial speedup?
- *Reminder:* Grover's algorithm need $q = \log_2(N)$ qubits to search $2^q = N$ records in time $\sqrt{N}$.
- *In practice:* only $q' = \alpha \cdot \log_2(N)$ qubits ($\alpha < 1$).
- *Main idea:* by using q' qubits, we can search $2^{q'} = N^\alpha$ in time $N^{\alpha/2}$. So:
  - divide N records into $N^{1-\alpha}$ groups of N^α;
  - use q' qubits to search these groups one by one.
- *Overall time:* each search takes $N^{\alpha/2}$, so we need

$$t(N) = N^{1-\alpha} \cdot N^{\alpha/2} = N^{1-\alpha/2} \ll N.$$

- *Comment:* when $\alpha \to 1$, we have $t(N) \to \sqrt{N}$.

Acknowledgments

This work was supported in part:

- by NASA grants NCC5-209 and NCC2-1232;
- by NSF grants CDA-9522207, EAR-0112968, EAR-0225670, and 9710940 Mexico/Conacyt;
- by the Future Aerospace Science and Technology Program (FAST) Center for Structural Integrity of Aerospace Systems,
- FAST Center was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, grants F49620-95-1-0518 and F49620-00-1-0365;
- by the Air Force Research Laboratory, Air Force Material Command, USAF, grant F33615-99-C-5211.