How to Take into Account Dependence Between the Inputs: From Interval Computations to Constraint-Related Set Computations, with Potential Applications to Nuclear Safety, Bio- and Geosciences

M. Ceberio¹, S. Ferson², V. Kreinovich¹
S. Chopra¹,³, G. Xiang¹, A. Murguia¹,⁴, J. Santillan¹

¹Computer Science, University of Texas, El Paso, TX 79968, USA, contact vladik@utep.edu
²Applied Biomathematics, Setauket, NY 11733, USA
³Lexmark International, Lexington, KY 40550, USA
⁴XIMIS, El Paso, TX 79912, USA
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: $\tilde{x}_i \neq x_i$ ($\Delta x_i \neq 0$) hence

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

What are bounds on $\Delta y \overset{\text{def}}{=} \tilde{y} - y$?
2. Probabilistic and Interval Uncertainty

- **Traditional approach:** we know probability distribution for $\Delta x_i$ (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 $\Delta_i$ on $|\Delta x_i|$ hence

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

3. Interval Computations: A Problem

![Diagram](https://via.placeholder.com/150)

- **Given:**
  - an algorithm \( y = f(x_1, \ldots, x_n) \) that transforms \( n \) real numbers \( x_i \) into a number \( y \);
  - \( n \) intervals \( x_i = [x_i, \bar{x}_i] \).

- **Compute:** the corresponding range of \( y \):

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

- **Fact:** even for quadratic \( f \), the problem of computing the exact range \( y \) 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 \( y \).
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, \( \Delta y \) 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. **General Approach: Interval-Type Step-by-Step Techniques**

- **Problem:** it is difficult to compute the range $y$.
- **Solution:** compute an enclosure $Y$ such that $y \subseteq Y$.
- **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 $[1/\overline{x}_1, 1/\overline{x}_2]$ (if $0 \notin \mathbf{x}_1$).
6. **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*:
  - \( r_1 := [1, 2] - [2, 2] = [-1, 0]; \)
  - \( r_2 := [1, 2] + [2, 2] = [3, 4]; \)
  - \( r_3 := [-1, 0] \cdot [3, 4] = [-4, 0]. \)

- **Actual range:** \( f(x) = [-3, 0]. \)

- **Comment:** this is just a toy example, there are more efficient ways of computing an enclosure \( Y \supseteq y. \)
7. Interval Computations: Analysis

- **Computation time**: $\leq 4$ arithmetic operations per original operation, so $O(T)$, where $T$ is the running time of the original algorithm.

- **Result**: often, enclosure $Y \supseteq y$ with excess width.

- **Reason**: there is a relation between intermediate results, and we ignore it in straightforward interval computations.

- **Alternative**: we can compute the exact range: e.g., Tarksi algorithm for algebraic $f$.

- **Computation time**: can be exponential $O(2^T)$.

- **Summarizing**: we have two algorithms:
  - a fast and efficient $O(T)$ algorithm which often has large excess width;
  - a slow and inefficient (often non-feasible) algorithm with no excess width.

- **It is desirable**: to develop a sequence of feasible algorithms with:
  - longer and longer computation time and
  - smaller and smaller excess width.
8. Interval Computations: Limitations

- **Traditional interval computations:**
  - we know the intervals $x_i$ of possible values of different parameters $x_i$,
  and
  - we assume that an arbitrary combination of these values is possible.

- **In geometric terms:** the set of possible combinations $x = (x_1, \ldots, x_n)$ is a box $x = x_1 \times \ldots \times x_n$.

- **In practice:** we also know additional restrictions on the possible combinations of $x_i$.

- **Example:** in geosciences, in addition to intervals for velocities $v_i$ at different points, we know that $|v_i - v_j| \leq \Delta$ for neighboring points:

- **Example:** in nuclear engineering, experts often state that combinations of extreme values are impossible, we have an ellipsoid, not a box.
9. Similar Situation: Statistics

- Ideally, we should take into account dependence between all the variables.

- In the first approximation, it is often reasonable to consider them independent.

- In the next approximation, we consider pairwise dependencies.

- To get an even better picture, we can consider dependencies between triples, etc.

- As a result, we get a sequence of methods which:
  - require more and more time
  - but at the same time lead to more and more accurate results.
10. Let Us Use a Similar Idea for Interval Uncertainty

- Ideally, we should take the box $x_1 \times \ldots \times x_n$ (or appropriate subset of the box), divide it into smaller boxes, estimate the range over each small box, and combine the results.

- This requires $C^n$ subboxes – i.e., exponential time.

- In straightforward interval computations, we consider only intervals of possible values of $x_i$.

- A natural next approximation is when we consider:
  - sets $x_i$ of possible values of $x_i$, and also
  - sets $x_{ij}$ of possible pairs $(x_i, x_j)$.

- Third approximation: we also consider possible sets of triples, etc.

- As a result, we hope to get a sequence of methods which:
  - require more and more time
  - but at the same time lead to more and more accurate results.
11. How to Represent Sets

- **First idea:** do it in a way cumulative probability distributions (cdf) are represented in RiskCalc package: by discretization.

- In RiskCalc, we:
  - divide the interval \([0, 1]\) of possible values of probability into, say, 10 subintervals of equal width and
  - represent cdf \(F(x)\) by 10 values \(x_1, \ldots, x_{10}\) at which \(F(x_i) = i/10\).

- Similarly, we:
  - divide the box \(x_i \times x_j\) into, say, 10 \(\times\) 10 subboxes and
  - describe the set \(x_{ij}\) by listing all subboxes which contain possible pairs.

- **Comment:**
  - A more efficient idea is to represent this set by a covering paving – in the style of Jaulin et al. – i.e., consider boxes of different sizes starting with larger ones and only decrease the size when necessary.
  - It is also possible (and often efficient) to use ellipsoids.
  - Idea is similar to rough sets.
12. How to Propagate This Uncertainty: A Problem and General Idea

Problem:

- **In the beginning**: we know the intervals \( r_1, \ldots, r_n \) corresponding to the input variables \( r_i = x_i \), and we know the sets \( r_{ij} \) for \( i, j \) from 1 to \( n \).
- **Question**: propagate this information through an intermediate computation step, a step of computing \( r_k = r_a * r_b \) for some arithmetic operation \(*\) and for previous results \( r_a \) and \( r_b \) \((a, b < k)\).
- By the time we come to this step, we know the intervals \( r_i \) and the sets \( r_{ij} \) for \( i, j < k \).
- We want to find the interval \( r_k \) for \( x_k \), and the sets \( r_{ik} \) for \( i < k \).

General idea:

- The range \( r_k \) can be naturally found as \( \{r_a * r_b \mid (r_a, r_b) \in r_{ab}\} \).
- The set \( r_{ak} \) is described as \( \{(r_a, r_a * r_b) \mid (r_a, r_b) \in r_{ab}\} \).
- The set \( r_{bk} \) is described as \( \{(r_b, r_a * r_b) \mid (r_a, r_b) \in r_{ab}\} \).
- For \( i \neq a, b \), the set \( r_{ik} \) is described as
  \[
  \{(r_i, r_a * r_b) \mid (r_i, r_a) \in r_{ia}, (r_i, r_b) \in r_{ib}\}.
  \]
- **Comment.** This is related to join
  \[
  r_{ai} \Join i r_{ib} = \{(r_a, r_i, r_b) \mid (r_a, r_i) \in r_{ai}, (r_i, r_b) \in r_{ib}\}.
  \]
13. **First Example: Computing the Range of** $x - x$

- **Problem:**
  - for $f(x) = x - x$ on $[0, 1]$, the actual range is $[0, 0]$;
  - straightforward interval computations lead to an enclosure $[0, 1] - [0, 1] = [-1, 1]$.

- In straightforward interval computations:
  - we have $r_1 = x$ with interval $r_1 = [0, 1]$;
  - we have $r_2 = x$ with interval $x_2 = [0, 1]$;
  - the variables $r_1$ and $r_2$ are dependent, but we ignore this dependence.

- In the new approach: we have $r_1 = r_2 = [0, 1]$, and we also have $r_{12}$:

  \[ \begin{array}{c}
  \hline
  \end{array} \]

- The resulting set is the exact range $\{0\} = [0, 0]$. 
14. How to Propagate This Uncertainty: Numerical Implementation

- First step: computing \( r_k \):
  - In our representation, the set \( x_{ab} \) consists of small 2-D boxes \( X_a \times X_b \).
  - For each small box \( X_a \times X_b \), we use interval arithmetic to compute the range \( X_a \ast X_b \) of the value \( r_a \ast r_b \) over this box.
  - Then, we take the union (interval hull) of all these ranges.

- Second step: computing \( r_{ik} \):
  - We consider the sets \( r_{ab} \), \( r_{ai} \), and \( r_{bi} \).
  - For each small box \( R_a \times R_b \) from \( r_{ab} \), we:
    * consider all subintervals \( R_i \) for which \( R_a \times R_i \) is in \( r_{ai} \) and \( R_b \times R_i \) is in \( r_{bi} \), and then
    * we add \( (R_a \ast R_b) \times R_i \) to the set \( r_{ki} \).
  - To be more precise:
    * since the interval \( R_a \ast R_b \) may not have bounds of the type \( p/10 \),
    * we may need to expand it to get within bounds of the desired type.

- We repeat these computations step by step until we get the desired estimate for the range of the final result of the computations.
15. First Example: Computing the Range of \( x - x \) (cont-d)

- **Problem:**
  - for \( f(x) = x - x \) on \([0, 1]\), the actual range is \([0, 0]\);
  - straightforward interval computations lead to an enclosure \([0, 1] - [0, 1] = [-1, 1]\).

- In straightforward interval computations:
  - we have \( r_1 = x \) with interval \( r_1 = [0, 1]\);
  - we have \( r_2 = x \) with interval \( x_2 = [0, 1]\);
  - the variables \( r_1 \) and \( r_2 \) are dependent, but we ignore this dependence.

- In the new approach: we have \( r_1 = r_2 = [0, 1] \), and we also have \( r_{12} \):

  \[
  \begin{array}{|c|c|c|c|c|}
  \hline
  \times & \times & \times & \times & \times \\
  \times & \times & \times & \times & \times \\
  \hline
  \end{array}
  \]

  - For each small box, we have \([-0.2, 0.2]\], so the union is \([-0.2, 0.2]\).
  - If we divide into more pieces, we get close to 0.
16. Second Example: Computing the Range of $x - x^2$

- In straightforward interval computations:
  - we have $r_1 = x$ with interval $r_1 = [0, 1]$;
  - we have $r_2 = x^2$ with interval $x_2 = [0, 1]$;
  - the variables $r_1$ and $r_2$ are dependent, but we ignore this dependence and estimate $r_3$ as $[0, 1] - [0, 1] = [-1, 1]$.

- In the new approach: we have $r_1 = r_2 = [0, 1]$, and we also have $r_{12}$:
  - the union of $R_1^2$ is $[0, 1]$, so we have $[0, 0.2]$, $[0.2, 0.4]$, etc.;
  - for $R_1 = [0, 0.2]$, we have $R_1^2 = [0, 0.04]$, so only $[0, 0.2]$ is affected;
  - for $R_1 = [0.2, 0.4]$, we have $R_1^2 = [0.04, 0.16]$, so only $[0, 0.2]$ is affected;
  - for $R_1 = [0.4, 0.6]$, we have $R_1^2 = [0.16, 0.25]$, so $[0, 0.2]$ and $[0.2, 0.4]$ are affected, etc.

  \[
  \begin{array}{|c|c|c|c|}
  \hline
  & & & \times \\ 
  & & \times & \\
  & \times & & \\
  \times & & & \\
  \hline
  \end{array}
  \]

- For each possible pair of small boxes $R_1 \times R_2$, we have $R_1 - R_2 = [-0.2, 0.2]$, $[0, 0.4]$ and $[0.2, 0.6]$, so the union of $R_1 - R_2$ is $r_3 = [-0.2, 0.6]$.

- If we divide into more pieces, we get closer to $[0, 0.25]$. 

17. How to Compute \( r_{ik} \)

- Since \( r_3 = [-0.2, 0.6] \), we divide this range into 5 subintervals \([-0.2, -0.04], [-0.04, 0.12], [0.12, 0.28], [0.28, 0.44], [0.44, 0.6] \).

- For \( R_1 = [0, 0.2] \), the only possible \( R_2 \) is \( [0, 0.2] \), so \( R_1 - R_2 = [-0.2, 0.2] \). This covers \([-0.2, -0.04]\) and \([-0.04, 0.12]\).

- For \( R_1 = [0.2, 0.4] \), the only possible \( R_2 \) is \( [0, 0.2] \), so \( R_1 - R_2 = [0, 0.4] \). This covers \([-0.04, 0.12], [0.12, 0.28], \) and \( [0.28, 0.44] \).

- For \( R_1 = [0.4, 0.6] \), we have two possible \( R_2 \):
  - for \( R_2 = [0, 0.2] \), we have \( R_1 - R_2 = [0.2, 0.6] \); this covers \([0.12, 0.28], [0.28, 0.44], \) and \([0.44, 0.6] \);
  - for \( R_2 = [0.2, 0.4] \), we have \( R_1 - R_2 = [0, 0.4] \); this covers \([-0.04, 0.12], [0.12, 0.28], \) and \([0.28, 0.44] \).

- For \( R_1 = [0.6, 0.8] \), we have \( R_1^2 = [0.36, 0.64] \), so three possible \( R_2 \): \([0.2, 0.4], [0.4, 0.6], \) and \([0.6, 0.8] \), to the total of \([0.2, 0.8] \). Here, \([0.6, 0.8] - [0.2, 0.8] = [-0.2, 0.6] \), so all 5 subintervals are affected.

- For \( R_1 = [0.8, 1.0] \), we have \( R_1^2 = [0.64, 1.0] \), so two possible \( R_2 \): \([0.6, 0.8] \) and \([0.8, 1.0] \), to the total of \([0.6, 1.0] \). Here, \([0.8, 1.0] - [0.6, 1.0] = [-0.2, 0.4] \), so the first 4 subintervals are affected.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
18. **Distributivity: $a \cdot (b + c)$ vs. $a \cdot b + a \cdot c$**

- **Problem:** compute the range of $x_1 \cdot (x_2 + x_3) = x_1 \cdot x_2 + x_1 \cdot x_3$ when $x_1 \in x_1 = [0, 1]$, $x_2 = [1, 1]$, and $x_3 = [-1, -1]$.

- **Actual range:** we have $x_1 \cdot (x_2 + x_3) = 0$ for all possible $x_i$ hence the actual range is $[0, 0]$.

- **Straightforward interval computations:**
  
  - for $x_1 \cdot (x_2 + x_3)$, we get $[0, 1] \cdot [0, 0] = [0, 0]$;
  
  - for $x_1 \cdot x_2 + x_1 \cdot x_3$, we get $[0, 1] \cdot 1 + [0, 1] \cdot (-1) = [0, 1] + [-1, 0] = [-1, 1]$, i.e., excess width.

- **Reason:** we have $r_4 = x_1 \cdot x_2$, $r_5 = x_1 \cdot x_3$, but we ignore the dependence between $r_4$ and $r_5$. 


- **Reminder:** \( r_4 = r_1 \cdot r_2, r_5 = r_1 \cdot r_3, r_6 = r_4 + r_5, \ r_1 = [0, 1], \ r_2 = 1, \ r_3 = -1. \)
- When we get \( r_4 = r_1 \cdot r_2, \) we compute the ranges \( r_{14}, r_{24}, \) and \( r_{34}; \) the only non-trivial range is \( r_{14}: \)

\[ \begin{array}{cccccc} \times & & & & & \\
& \times & & & & \\
& & \times & & & \\
& & & \times & & \\
& & & & \times & \\
\end{array} \]

- For \( r_5 = r_1 \cdot r_3, \) we get \( r_5 = [-1, 0]. \)
- To compute the range \( r_{45}, \) for each possible box \( R_1 \times R_3, \) we:
  - consider all boxes \( R_4 \) for which \( R_4 \times R_1 \) is possible and \( R_4 \times R_3 \) is possible;
  - add \( R_4 \times (R_1 \cdot R_3) \) to the set \( r_{45}. \)
- **Result:**

\[ \begin{array}{cccccc} \times & & & & & \\
& \times & & & & \\
& & \times & & & \\
& & & \times & & \\
& & & & \times & \\
\end{array} \]

- Hence, for \( r_6 = r_4 + r_5, \) we get \([0, -0.2]. \)
- If we divide into more pieces, we get the enclosure closer to 0.
20. Toy Example with Prior Dependence

- **Case study:** find the range of $r_1 - r_2$ when $r_1 = [0, 1]$, $r_2 = [0, 1]$, and $|r_1 - r_2| \leq 0.2$.

- **Actual range:** $[-0.2, 0.2]$.

- **Straightforward interval computations:** $[0, 1] - [0, 1] = [-1, 1]$.

- **New approach:**
  - First, we describe the set $r_{12}$:

  

  ![Diagram of r_{12}]

  - Next, we compute $\{r_1 - r_2 \mid (r_1, r_2) \in r_{12}\}$.

- **Result:** $[-0.2, 0.2]$.
21. Toy Example with Prior Dependence (cont-d)

- **Case study:** find the range of \( r_1 - r_2 \) when \( r_1 = [0, 1], \ r_2 = [0, 1], \) and \( |r_1 - r_2| \leq 0.1. \)

- **Actual range:** \([-0.2, 0.2].\)

- **Straightforward interval computations:** \([0, 1] - [0, 1] = [-1, 1].\)

- **New approach:**
  - First, we describe the constraint in terms of subboxes:
    
    \[
    \begin{array}{cccccccc}
    & & & & \times & \times & \\
    & & \times & \times & \times & \\
    & \times & \times & \times & \\
    \times & \times & \times & \\
    \times & \times & & \\
    & & & & & & \\
    & & & & & & \\
    \end{array}
    \]
  - Next, we compute \( R_1 - R_2 \) for all possible pairs and take the union.

- **Result:** \([-0.6, 0.6].\)

- If we divide into more pieces, we get the enclosure closer to \([-0.2, 0.2].\)
22. Computation Time

- **Straightforward interval computations:**
  - we need to compute $T$ intervals $r_i$, $i = 1, \ldots, T$;
  - so, it requires $O(T)$ steps.

- **New idea:**
  - we need to compute $T^2$ sets $r_{ij}$, $i, j = 1, \ldots, T$;
  - so, it requires $O(T^2)$ steps.

- **Conclusion:**
  - the new method is longer than for straightforward interval computations, but
  - it is still feasible.
23. What Next?

- **Known fact:** the range estimation problem is, in general, NP-hard (even without any dependency between the inputs).

- **Corollary:** our quadratic time method cannot completely avoid excess width.

- To get better estimates, in addition to sets of pairs, we can also consider sets of *triples* $r_{ijk}$.

- This will be a $T^3$ time version of our approach.

- We can also go to *quadruples* etc.

- Similar ideas can be applied to the case when we also have partial information about probabilities.
24. Probabilistic Case: In Brief

- *Traditionally:* expert systems use technique similar to straightforward interval computations.
- We parse $F$ and replace each computation step with corresponding probability operation.
- *Problem:* at each step, we ignore the dependence between the intermediate results $F_j$.
- *Result:* intervals are too wide (and numerical estimates off).
- *Example:* the estimate for $P(A \lor \neg A)$ is not 1.
- *Solution:* similarly to the above algorithm, besides $P(F_j)$, we also compute $P(F_j \& F_i)$ (or $P(F_{j_1} \& \ldots \& F_{j_k})$).
- On each step, use all combinations of $l$ such probabilities to get new estimates.
- *Result:* e.g., $P(A \lor \neg A)$ is estimated as 1.
25. **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.

Many thanks to:

- Luc Jaulin,
- Arnold Neumaier, and
- Bill Walster

for valuable suggestions.
26. When is the New Method Exact?

- Straightforward interval computations are exact for single-use expressions (SUE).
- Our method is exact for \( x - x \), \( x - x^2 \), and \( x_1 \cdot x_2 + x_1 \cdot x_3 \).
- In all these expressions, each variable occurs no more than twice.
- **Hypothesis**: the new method is exact for all “double-use” expressions (DUE).
- **Counterexample**:
  
  - variance is DUE \( V = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i^2 - \left( \frac{1}{n} \cdot \sum_{i=1}^{n} x_i \right)^2 \), but
  
  - computing the range of variance on interval data \( x_i \) is NP-hard.

- **Counterexample to another reasonable hypothesis**: range estimation is NP-hard even for SUE expressions with linear SUE constraints.
- **Open question**: when is the new method exact?