Combining Interval and Probabilistic Uncertainty in Engineering Applications

Andrew Pownuk

Computational Science Program
University of Texas at El Paso
El Paso, Texas 79968, USA
ampownuk@utep.edu

Part I
Introduction

1. Need for Data Processing

- One of the main objectives of science is to predict future values y of physical quantities:
  - in meteorology, we need to predict future weather;
  - in airplane control, we need to predict the location and the velocity of the plane under current control.
- To make this prediction:
  - we need to know the relation $y = f(x_1, \ldots, x_n)$ between y and related quantities $x_1, \ldots, x_n$;
  - then, we measure or estimate $x_1, \ldots, x_n$;
  - finally, we use the results $\widetilde{x}_i$ of measurement (or estimation) to compute an estimate

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

- This computation of $\widetilde{y}$ is an important case of *data processing*.

2. Need to Take Uncertainty Into Account When Processing Data

- Measurement are never absolutely accurate: in general,

$$\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i \neq 0.$$

- As a result, the estimate $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ is, in general, different from the ideal value $y = f(x_1, \ldots, x_n)$.
- To estimate the accuracy $\Delta y \stackrel{\text{def}}{=} \widetilde{y} - y$, we need to have some information about the measurement errors Δx_i.
- Traditional engineering approach assumes that we know the probability distribution of each Δx_i.
- Often, $\Delta x_i \sim N(0, \sigma_i)$, and different Δx_i are assumed to be independent.
- In such situations, our goal is to find the probability distribution for Δy.

3. Cases of Interval and Fuzzy Uncertainty

- Often, we only know the upper bound Δ_i: $|\Delta x_i| \le \Delta_i$.
- Then, the only information about the x_i is that

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

- Different $x_i \in \mathbf{x}_i$ lead, in general, to different

$$y = f(x_1, \ldots, x_n).$$

- We want to find the range $\mathbf{y}$ of possible values of y:

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

- To gauge the accuracy of expert estimates, it is reasonable to use fuzzy techniques, i.e., to describe:
  - for each possible value x_i,
  - the degree $\mu_i(x_i)$ to which x_i is possible.

4. Measurement and Estimation Inaccuracies Are Usually Small

- In many practical situations, the measurement and estimation inaccuracies Δx_i are relatively small.
- Then, we can safely ignore terms which are quadratic (or of higher order) in terms of Δx_i:

$$\Delta y = \widetilde{y} - y = f(\widetilde{x}_1, \ldots, \widetilde{x}_n) - f(\widetilde{x}_1 - \Delta x_1, \ldots, \widetilde{x}_n - \Delta x_n) = \sum_{i=1}^{n} c_i \cdot \Delta x_i, \text{ where } c_i = \frac{\partial f}{\partial x_i}.$$

- If needed, the derivative can be estimated by numerical differentiation

$$c_i \approx \frac{f(\widetilde{x}_1, \ldots, \widetilde{x}_{i-1}, \widetilde{x}_i + h, \widetilde{x}_{i+1}, \ldots, \widetilde{x}_n) - \widetilde{y}}{h}.$$

5. Case of Interval Uncertainty

- Let us consider the case when $\Delta y = \sum_{i=1}^{n} c_i \cdot \Delta x_i$.
- In this case, $\mathbf{y} = [\widetilde{y} - \Delta, \widetilde{y} + \Delta]$, where $\Delta = \sum_{i=1}^{n} |c_i| \cdot \Delta_i$.
- Sometimes, we have explicit expressions or efficient algorithms for the partial derivatives c_i.
- Often, however, we use proprietary software in our computations.
- Then, we cannot use differentiation formulas, but we can use numerical differentiation.
- *Problem:* We need $n + 1$ calls to f, to compute $\widetilde{y}$ and n values c_i.
- When f is time-consuming and n is large, this takes too long.

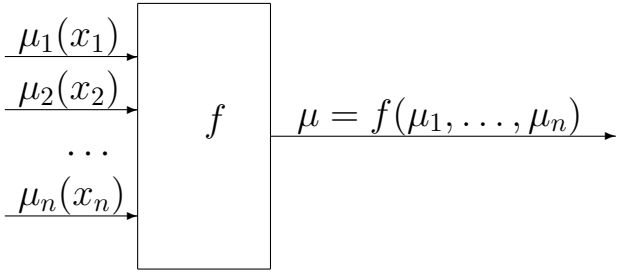
6. A Faster Method: Cauchy-Based Monte-Carlo

- *Idea:* use Cauchy distribution $\rho_\Delta(x) = \dfrac{\Delta}{\pi} \cdot \dfrac{1}{1 + x^2/\Delta^2}$.
- *Why:* when $\Delta x_i \sim \rho_{\Delta_i}(x)$ are indep., then $\Delta y = \sum\limits_{i=1}^{n} c_i \cdot \Delta x_i \sim \rho_\Delta(x)$, with $\Delta = \sum\limits_{i=1}^{n} |c_i| \cdot \Delta_i$.
- Thus, we simulate $\Delta x_i^{(k)} \sim \rho_{\Delta_i}(x)$; then, $\Delta y^{(k)} \stackrel{\text{def}}{=} \widetilde{y} - f(\widetilde{x}_1 - \Delta x_1^{(k)}, \ldots) \sim \rho_\Delta(x)$.
- Maximum Likelihood method can estimate Δ: $\prod\limits_{k=1}^{N} \rho_\Delta(\Delta y^{(k)}) \to \max$, so $\sum\limits_{k=1}^{N} \dfrac{1}{1 + (\Delta y^{(k)})^2/\Delta^2} = \dfrac{N}{2}$.
- To find Δ from this equation, we can use, e.g., the bisection method for $\underline{\Delta} = 0$ and $\overline{\Delta} = \max\limits_{1 \le k \le N} |\Delta y^{(k)}|$.

7. Monte-Carlo: Successes and Limitations

- *Fact:* for Monte-Carlo, accuracy is $\varepsilon \sim 1/\sqrt{N}$.
- *Good news:* the number N of calls to f depends only the desired accuracy ε.
- *Example:* to find Δ with accuracy 20% and certainty 95%, we need $N = 200$ iterations.
- *Limitation:* this method is *not realistic*; indeed:
  - we know that Δx_i is *inside* $[-\Delta_i, \Delta_i]$, but
  - Cauchy-distributed variable has a high probability to be *outside* this interval.
- *Natural question:* is it a limitation of our method, or of a problem itself?

8. Fuzzy Case: A Problem



- *Given:* an algorithm $y = f(x_1, \ldots, x_n)$ and n fuzzy numbers $\mu_i(x_i)$.
- *Compute:* $\mu(y) = \max\limits_{x_1,\ldots,x_n : f(x_1,\ldots,x_n)=y} \min(\mu_1(x_1), \ldots, \mu_n(x_n))$.
- *Motivation:* y is a possible value of $Y \leftrightarrow \exists x_1, \ldots, x_n$ s.t. each x_i is a possible value of X_i and $f(x_1, \ldots, x_n) = y$.
- *Details:* "and" is min, $\exists$ ("or") is max, hence

$$\mu(y) = \max_{x_1,\ldots,x_n} \min(\mu_1(x_1), \ldots, \mu_n(x_n), t(f(x_1, \ldots, x_n) = y)),$$

where $t(\text{true}) = 1$ and $t(\text{false}) = 0$.

9. Fuzzy Case: Reduction to Interval Computations

- *Given:* an algorithm $y = f(x_1, \ldots, x_n)$ and n fuzzy numbers X_i described by membership functions $\mu_i(x_i)$.
- *Compute:* $Y = f(X_1, \ldots, X_n)$, where Y is defined by Zadeh's extension principle:

$$\mu(y) = \max_{x_1,\ldots,x_n : f(x_1,\ldots,x_n)=y} \min(\mu_1(x_1), \ldots, \mu_n(x_n)).$$

- *Idea:* represent each X_i by its α-cuts

$$X_i(\alpha) = \{x_i : \mu_i(x_i) \geq \alpha\}.$$

- *Advantage:* for continuous f, for every α, we have

$$Y(\alpha) = f(X_1(\alpha), \ldots, X_n(\alpha)).$$

- *Resulting algorithm:* for $\alpha = 0, 0.1, 0.2, \ldots, 1$ apply interval computations techniques to compute $Y(\alpha)$.

10. Open Problems

- In engineering applications, we want methods for estimating uncertainty which are:
  - *accurate* – this is most important in most engineering applications;
  - *fast*: this is important in some engineering applications where we need real-time computations,
  - *understandable* to engineers – otherwise, engineers will be reluctant to use them, and
  - sufficiently *general* – so that they can be applied in all kinds of situations.
- It is thus desirable to design more accurate, faster, more understandable, and/or more general methods.

11. What We Do in This Thesis

- First, we show how to make the current methods more accurate.
- Then, we show how to make these methods faster.
- After that, we show how to make these methods more understandable to engineers.
- Finally, we analyze how to make these methods more general.
- We also describe remaining open problems and our plan for future work.

Part II
How to Get More Accurate Estimates – by Properly Taking Model Inaccuracy into Account

12. Linearization-Based Algorithm: Reminder

- *We know:* an algorithm $f(x_1, \ldots, x_n)$ and values $\widetilde{y}_i$ and Δ_i.
- *We need to find:* the range of values $f(x_1, \ldots, x_n)$ when $x_i \in [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$.
- *Algorithm:*

  1) first, we compute $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$;
  2) then, for each i from 1 to n, we compute

$$y_i = f(\widetilde{x}_1, \ldots, \widetilde{x}_{i-1}, \widetilde{x}_i + \Delta_i, \widetilde{x}_{i+1}, \ldots, \widetilde{x}_n);$$

  3) after that, we compute $\overline{y} = \widetilde{y} + \sum_{i=1}^{n} |y_i - \widetilde{y}|$ and $\underline{y} = \widetilde{y} - \sum_{i=1}^{n} |y_i - \widetilde{y}|$.

13. Taking Model Inaccuracy into Account

- We rarely know the exact dependence $y = f(x_1, \ldots, x_n)$.
- We have an approx. model $F(x_1, \ldots, x_n)$ w/known accuracy ε: $|F(x_1, \ldots, x_n) - f(x_1, \ldots, x_n)| \le \varepsilon$.
- *We know:* an algorithm $F(x_1, \ldots, x_n)$, accuracy ε, values $\widetilde{x}_i$ and Δ_i.
- *Find:* the range $\{f(x_1, \ldots, x_n) : x_i \in [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]\}$.
- If we use the approximate model in our estimate, we get $\overline{Y} = \widetilde{Y} + \sum_{i=1}^{n} |Y_i - \widetilde{Y}|$.
- Here, $|\widetilde{Y} - \widetilde{y}| \le \varepsilon$ and $|Y_i - y_i| \le \varepsilon$, so $|\overline{y} - \overline{Y}| \le (2n+1) \cdot \varepsilon$.
- Thus, we arrive at the following algorithm.

14. Resulting Algorithm

- *We know:* an algorithm $F(x_1, \ldots, x_n)$, accuracy ε, values $\widetilde{x}_i$ and Δ_i.
- *Find:* the range $\{f(x_1, \ldots, x_n) : x_i \in [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]\}$.
- *Algorithm:*

  1) compute $\widetilde{Y} = Y(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ and

  $$Y_i = F(\widetilde{x}_1, \ldots, \widetilde{x}_{i-1}, \widetilde{x}_i + \Delta_i, \widetilde{x}_{i+1}, \ldots, \widetilde{x}_n).$$

  2) compute $\overline{B} = \widetilde{Y} + \sum_{i=1}^{n} |Y_i - \widetilde{Y}| + (2n+1) \cdot \varepsilon$ and

  $$\underline{B} = \widetilde{Y} - \sum_{i=1}^{n} |Y_i - \widetilde{Y}| - (2n+1) \cdot \varepsilon.$$

- *Problem:* when n is large, then, even for reasonably small inaccuracy ε, the value $(2n+1) \cdot \varepsilon$ is large.
- *What we do:* we show how we can get better estimates for $\overline{y}$.

15. How to Get Better Estimates: Idea

- One possible source of model inaccuracy is discretization (e.g., FEM).
- When we select a different combination of parameters, we get an *unrelated* value of inaccuracy.
- So, let's consider approx. errors $\Delta y \stackrel{\text{def}}{=} F(x_1, \ldots, x_n) - f(x_1, \ldots, x_n)$ as *independent* random variables.
- What is a probability distribution for these random variables? We know that $\Delta y \in [-\varepsilon, \varepsilon]$.
- We do not have any reason to assume that some values from this interval are more probable than others.
- So, it is reasonable to assume that all the values are equally probable: a uniform distribution.
- For this uniform distribution, the mean is 0, and the standard deviation is $\sigma = \dfrac{\varepsilon}{\sqrt{3}}$.

16. How to Get a Better Estimate for $\widetilde{y}$

- In our main algorithm, we apply the computational model F to $n+1$ different tuples.
- Let's also compute $M \stackrel{\text{def}}{=} F(\widetilde{x}_1 - \Delta_1, \ldots, \widetilde{x}_n - \Delta_n)$.
- In linearized case, $\widetilde{y} + \sum_{i=1}^{n} y_i + m = (n+2) \cdot \widetilde{y}$, so $\widetilde{y} = \frac{1}{n+2} \cdot \left(\widetilde{y} + \sum_{i=1}^{n} y_i + m \right)$, and we can estimate $\widetilde{y}$ as

$$\widetilde{Y}_{\text{new}} = \frac{1}{n+2} \cdot \left(\widetilde{Y} + \sum_{i=1}^{n} Y_i + m \right).$$

- Here, $\Delta\widetilde{y}_{\text{new}} = \frac{1}{n+2} \cdot \left(\Delta\widetilde{y} + \sum_{i=1}^{n} \Delta y_i + \Delta m \right)$, so its variance is $\sigma^2\left[\widetilde{Y}_{\text{new}}\right] = \frac{\varepsilon^2}{3 \cdot (n+2)} \ll \frac{\varepsilon^2}{3} = \sigma^2\left[\widetilde{Y}\right]$.

17. Let Us Use $\widetilde{Y}_{\text{new}}$ When Estimating $\overline{y}$

- Let us compute $\overline{Y}_{\text{new}} = \widetilde{Y}_{\text{new}} + \sum_{i=1}^{n} |Y_i - \widetilde{Y}_{\text{new}}|$.
- Here, when $s_i \in \{-1, 1\}$ are the signs of $y_i - \widetilde{y}$, we get:

$$\overline{y} = \widetilde{y} + \sum_{i=1}^{n} s_i \cdot (y_i - \widetilde{y}) = \left(1 - \sum_{i=1}^{n} s_i\right) \cdot \widetilde{y} + \sum_{i=1}^{n} s_i \cdot y_i.$$

- Thus, $\Delta\overline{y}_{\text{new}} = \left(1 - \sum_{i=1}^{n} s_i\right) \cdot \Delta\widetilde{y}_{\text{new}} + \sum_{i=1}^{n} s_i \cdot \Delta y_i$, so

$$\sigma^2 = \left(1 - \sum_{i=1}^{n} s_i\right)^2 \cdot \frac{\varepsilon^2}{3 \cdot (n+2)} + \sum_{i=1}^{n} \frac{\varepsilon^2}{3}.$$

- Here, $|s_i| \le 1$, so $\left|1 - \sum_{i=1}^{n} s_i\right| \le n + 1$, and

$$\sigma^2 \le \frac{\varepsilon^2}{3} \cdot (2n+1).$$

18. Using $\widetilde{Y}_{\text{new}}$ (cont-d)

- We have $\Delta\overline{y}_{\text{new}} = \left(1 - \sum_{i=1}^{n} s_i\right) \cdot \Delta\widetilde{y}_{\text{new}} + \sum_{i=1}^{n} s_i \cdot \Delta y_i$.
- Due to the Central Limit Theorem, $\Delta\overline{y}_{\text{new}}$ is $\approx$ normal.
- We know that $\sigma^2 \le \dfrac{\varepsilon^2}{3} \cdot (2n+1)$.
- Thus, with certainty depending on k_0, we have

  $$\overline{y} \le \overline{Y}_{\text{new}} + k_0 \cdot \sigma \le \overline{Y}_{\text{new}} + k_0 \cdot \frac{\varepsilon}{\sqrt{3}} \cdot \sqrt{2n+1}:$$

  - with certainty 95% for $k_0 = 2$,
  - with certainty 99.9% for $k_0 = 3$, etc.
- Here, inaccuracy grows as $\sqrt{2n+1}$.
- This is much better than in the traditional approach, where it grows $\sim 2n+1$.

19. Resulting Algorithm

- *We know:* $F(x_1, \ldots, x_n)$, ε, $\widetilde{x}_i$ and Δ_i.
- *We want:* to find the range of $f(x_1, \ldots, x_n)$ when $x_i \in [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$.
- *Algorithm:*

1) compute $\widetilde{Y} = F(\widetilde{x}_1, \ldots, \widetilde{x}_n)$,

$$M = F(\widetilde{x}_1 - \Delta_1, \ldots, \widetilde{x}_n - \Delta_n), \text{ and}$$

$$Y_i = F(\widetilde{x}_1, \ldots, \widetilde{x}_{i-1}, \widetilde{x}_i + \Delta_i, \widetilde{x}_{i+1}, \ldots, \widetilde{x}_n);$$

2) compute $\widetilde{Y}_{\text{new}} = \dfrac{1}{n+2} \cdot \left(\widetilde{Y} + \sum_{i=1}^{n} Y_i + M \right)$,

$$\overline{b} = \widetilde{Y}_{\text{new}} + \sum_{i=1}^{n} \left| Y_i - \widetilde{Y}_{\text{new}} \right| + k_0 \cdot \sqrt{2n+1} \cdot \frac{\varepsilon}{\sqrt{3}};$$

$$\underline{b} = \widetilde{Y}_{\text{new}} - \sum_{i=1}^{n} \left| Y_i - \widetilde{Y}_{\text{new}} \right| - k_0 \cdot \sqrt{2n+1} \cdot \frac{\varepsilon}{\sqrt{3}}.$$

20. A Similar Improvement Is Possible for the Cauchy Method

- In the Cauchy method, we compute $\widetilde{Y}$ and the values

$$Y^{(k)} = F(\widetilde{x}_1 + \eta_1^{(k)}, \ldots, \widetilde{x}_n + \eta_n^{(k)}).$$

- We can then compute the improved estimate for $\widetilde{y}$, as:

$$\widetilde{Y}_{\text{new}} = \frac{1}{N+1} \cdot \left(\widetilde{Y} + \sum_{k=1}^{N} Y^{(k)} \right).$$

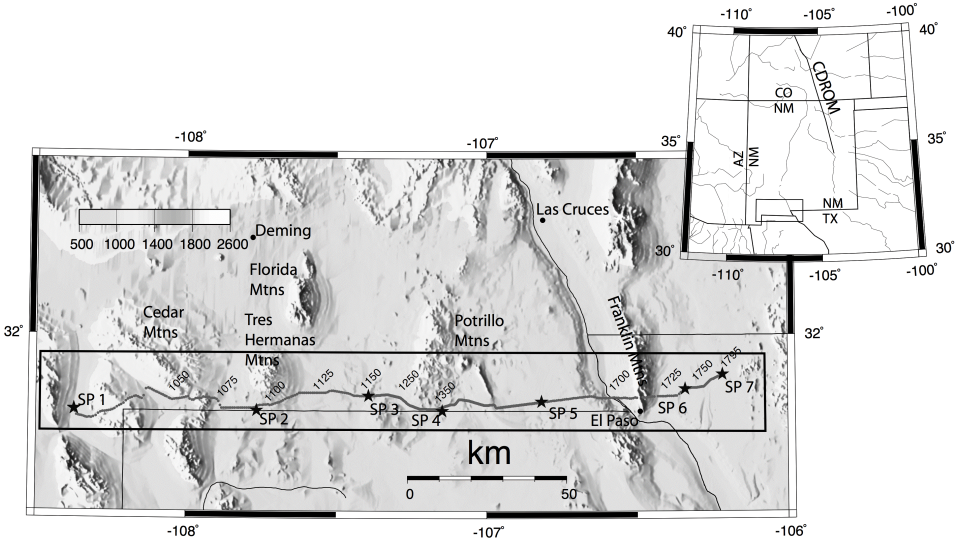
- We can now use this improved estimate when estimating the differences $\Delta y^{(k)}$: namely, we compute

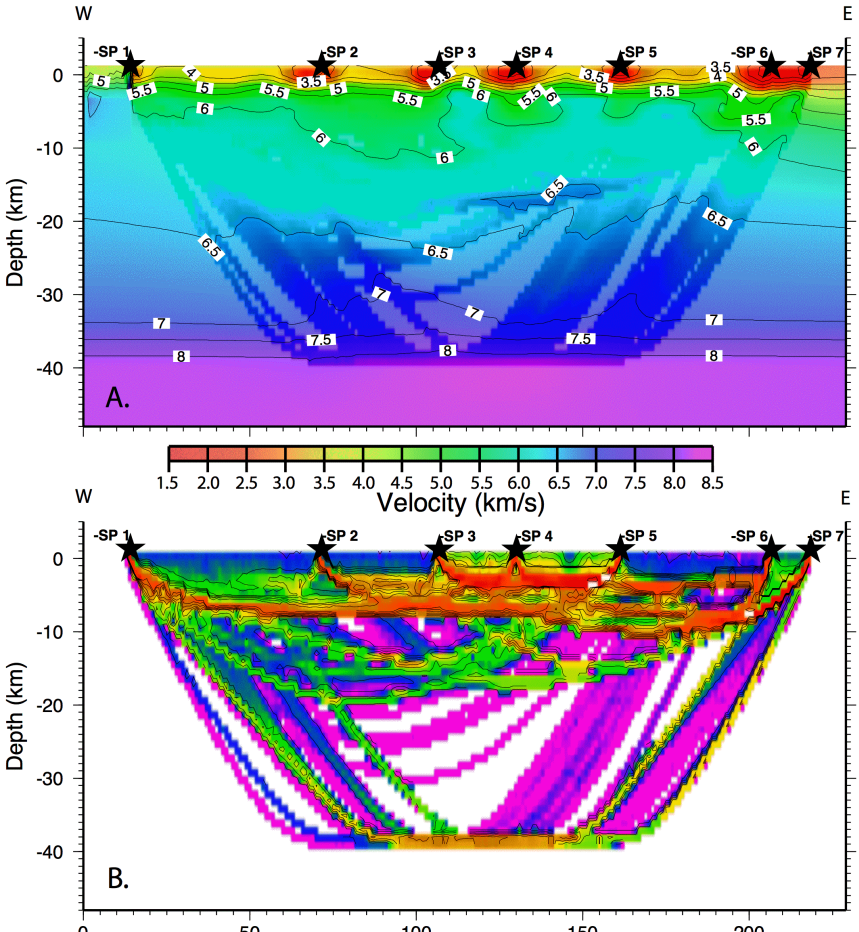
$$Y^{(k)} - \widetilde{Y}_{\text{new}}.$$

21. Experimental Testing: Seismic Inverse Problem in Geophysics

- *Problem:* reconstruct the velocity of sound v_i at different spatial locations and at different depths.
- *What we know:* the travel-times t_j of a seismic signal from the set-up explosion to seismic stations.
- *Hole's iterative algorithm:*
  - we start with geology-motivated values $v_i^{(1)}$;
  - based on the current approximation $v_i^{(k)}$, we estimate the travel times $t_j^{(k)}$;
  - update v_i: $\dfrac{1}{v_i^{(k+1)}} = \dfrac{1}{v_i^{(k)}} + \dfrac{1}{n_i} \cdot \sum_j \dfrac{t_j - t_j^{(k)}}{L_j}$.
- Using $\widetilde{Y}_{\text{new}}$ decreased the inaccuracy σ, on average, by 15%; σ increased only in one case (only by 7%).

22. Case Study: Seismic Inverse Problem in the Geosciences





23. Can We Further Improve the Accuracy?

- The inaccuracy $Y \neq y$ is caused by using elements of finite size h.
- This inaccuracy is proportional to h.
- If we decrease h to h', we thus need $K \stackrel{\text{def}}{=} \dfrac{h^3}{(h')^3}$ more cells, so we need K times more computations.
- Thus, the accuracy decreases as $\sqrt[3]{K}$.
- *New idea:* select K small vectors $\left(\Delta_1^{(k)}, \ldots, \Delta_n^{(k)}\right)$ which add up to 0, and estimate $\widetilde{y}$ as

$$Y_K(x_1, \ldots, x_n) = \frac{1}{K} \cdot \sum_{k=1}^{K} F\left(x_1 + \Delta_1^{(k)}, \ldots, x_n + \Delta_n^{(k)}\right).$$

- Averaging K independent random errors decreases the inaccuracy by a factor of $\sqrt{K}$, much faster than $\sqrt[3]{K}$.

Part III

How to Speed Up Computations – by Processing Different Types of Uncertainty Separately

24. Cases for Which a Speed-Up Is Possible

- Sometimes, all membership functions are "of the same type": $\mu(z) = \mu_0(k \cdot z)$ for some symmetric $\mu_0(z)$.
- *Example:* for triangular functions,

$$\mu_0(z) = \max(1 - |z|, 0).$$

- In this case, $\mu(z) \geq \alpha$ is equivalent to $\mu_0(k \cdot z) \geq \alpha$, so ${}^{\alpha}\Delta_0 = k \cdot {}^{\alpha}\Delta$ and ${}^{0}\Delta_0 = k \cdot {}^{0}\Delta$.
- Thus, ${}^{\alpha}\Delta = f(\alpha) \cdot {}^{0}\Delta$, where $f(\alpha) \stackrel{\text{def}}{=} \dfrac{{}^{\alpha}\Delta_0}{{}^{0}\Delta_0}$.
- For example, for a triangular membership function, we have $f(\alpha) = 1 - \alpha$.
- So, if we know the type μ_0 (hence $f(\alpha)$), and we know the 0-cut, we can compute all α-cuts as ${}^{\alpha}\Delta = f(\alpha) \cdot {}^{0}\Delta$.
- So, if $\mu_i(\Delta x_i)$ are of the same type, then for all α, we have ${}^{\alpha}\Delta_i = f(\alpha) \cdot {}^{0}\Delta_i$

25. When a Speed-Up Is Possible (cont-d)

- We know that ${}^{\alpha}\Delta = \sum_{i=1}^{n} |c_i| \cdot {}^{\alpha}\Delta_i$.
- For ${}^{\alpha}\Delta_i = f(\alpha) \cdot {}^{0}\Delta_i$, we get

$$ {}^{\alpha}\Delta = \sum_{i=1}^{n} |c_i| \cdot f(\alpha) \cdot {}^{0}\Delta_i. $$

- So, ${}^{\alpha}\Delta = f(\alpha) \cdot \sum_{i=1}^{n} |c_i| \cdot {}^{0}\Delta_i = f(\alpha) \cdot {}^{0}\Delta$.
- Thus, if all $\mu(x)$ are of the same type $\mu_0(z)$, there is no need to compute ${}^{\alpha}\Delta$ eleven times:
  - it is sufficient to compute ${}^{0}\Delta$;
  - to find all other values ${}^{\alpha}\Delta$, we simply multiply ${}^{0}\Delta$ by the factors $f(\alpha)$ corresponding to $\mu_0(z)$.

26. A More General Case

- A more general case is:
  - when we have a list of T different types of uncertainty – i.e., types of membership functions, and
  - each approximation error Δx_i consists of $\le T$ components of the corresponding type t:

$$\Delta x_i = \sum_{t=1}^{T} \Delta x_{i,t}.$$

- For example:
  - type $t = 1$ may correspond to intervals (which are, of course, a particular case of fuzzy uncertainty),
  - type $t = 2$ may correspond to triangular membership functions, etc.

27. How This Case Is Processed Now

- *First stage:*
  - we use the known membership functions $\mu_{i,t}(\Delta x_{i,t})$
  - to find the memberships functions $\mu_i(\Delta x_i)$ that correspond to the sum Δx_i.
- *Second stage:* we use $\mu_i(\Delta x_i)$ to compute the desired membership function $\mu(\Delta y)$.
- *Problem:* on the second stage, we apply the above formula eleven times:

$$^{\alpha}\Delta = \sum_{i=1}^{n} |c_i| \cdot {}^{\alpha}\Delta_i.$$

28. Main Idea

- We have $\Delta y = \sum_{i=1}^{n} c_i \cdot \Delta x_i$, where

$$\Delta x_i = \sum_{t=1}^{T} \Delta x_{i,t}.$$

- Thus, $\Delta y = \sum_{i=1}^{n} c_i \cdot \left(\sum_{t=1}^{T} \Delta x_{i,t} \right)$.
- Grouping together all the terms corr. to type t, we get $\Delta y = \sum_{t=1}^{T} \Delta y_t$, where $\Delta y_t \stackrel{\text{def}}{=} \sum_{i=1}^{n} c_i \cdot \Delta x_{i,t}$.
- For each t, we are combining membership functions of the same type, so it is enough to compute ${}^0\Delta_t$.
- Then, we add the resulting membership functions – by adding the corresponding α-cuts.

29. Resulting Algorithm

- Let $[-{}^0\Delta_{i,t}, {}^0\Delta_{i,t}]$ be 0-cuts of the membership functions $\mu_{i,t}(\Delta x_{i,t})$.
- Based on these 0-cuts, we compute, for each type t, the values ${}^0\Delta_t = \sum_{i=1}^{n} |c_i| \cdot {}^0\Delta_{i,t}$.
- Then, for $\alpha = 0$, $\alpha = 0.1$, ..., and for $\alpha = 1.0$, we compute the values ${}^\alpha\Delta_t = f_t(\alpha) \cdot {}^0\Delta_t$.
- Finally, we add up α-cuts corresponding to different types t, to come up with the expression ${}^\alpha\Delta = \sum_{t=1}^{T} {}^\alpha\Delta_t$.
- *Comment.* We can combine the last two steps into a single step: ${}^\alpha\Delta = \sum_{t=1}^{T} f_t(\alpha) \cdot {}^0\Delta_t$.

30. The New Algorithm Is Much Faster

- The original algorithm computed the above formula eleven times:

$$^{\alpha}\Delta = \sum_{i=1}^{n} |c_i| \cdot {}^{\alpha}\Delta_i.$$

- The new algorithm uses the corresponding formula T times, i.e., as many times as there are types.
- All the other computations are much faster, since they do not grow with the input size n.
- Thus, if the number T of different types is smaller than eleven, the new methods is much faster.
- *Example:* for $T = 2$ types (e.g., intervals and triangular $\mu(x)$), we get a $\dfrac{11}{2} = 5.5$ times speedup.

31. Conclusions and Future Work

- We can therefore conclude that sometimes, it is beneficial to process different types of uncertainty separately.
- Namely, it is beneficial when we have ten or fewer different types of uncertainty.
- The fewer types of uncertainty we have, the faster the resulting algorithm.
- We plan to test this idea of several actual data processing examples.
- We also plan to extend this idea to other types of uncertainty, in particular, to probabilistic uncertainty.

Part IV

Towards a Better Understandability of Uncertainty-Estimating Algorithms: Explaining the Need for Non-Realistic Monte-Carlo Simulations

32. Formulation of the Problem: Reminder

- *Good news:* Cauchy-based Monte-Carlo method is an efficient way of estimating interval uncertainty.
- *Limitation:* this method is *not realistic*; indeed:
  - we know that Δx_i is *inside* $[-\Delta_i, \Delta_i]$, but
  - Cauchy-distributed variable has a high probability to be *outside* this interval.
- *Natural question:* is it a limitation of our method, or of a problem itself?
- *Our answer:* for interval uncertainty, a realistic Monte-Carlo method is not possible.

33. Proof: Case of Independent Variables

- It is sufficient to prove that we cannot get the correct estimate for *one* specific function

  $$f(x_1, \ldots, x_n) = x_1 + \ldots + x_n, \text{ when } \Delta y = \Delta x_1 + \ldots + \Delta x_n.$$

- When each variables Δx_i is in the interval $[-\delta, \delta]$, then the range of Δy is $[-\Delta, \Delta]$, where $\Delta = n \cdot \delta$.
- In Monte-Carlo, $\Delta y^{(k)} = \Delta x_1^{(k)} + \ldots + \Delta x_n^{(k)}$.
- $\Delta_i^{(k)}$ are i.i.d. Due to the Central Limit Theorem, when $n \to \infty$, the distribution of the sum tends to Gaussian.
- For a normal distribution, with very high confidence, $\Delta y \in [\mu - k \cdot \sigma, \mu + k \cdot \sigma]$.
- Here, $\sigma \sim \sqrt{n}$, so this interval has width $w \sim \sqrt{n}$.
- However, the actual range of Δy is $\sim n \gg w$. Q.E.D.

34. General Case

- Let's take $f(x_1, \ldots, x_n) = s_1 \cdot x_1 + \ldots + s_n \cdot x_n$, where $s_i \in \{-1, 1\}$.
- Then, $\Delta = \sum_{i=1}^{n} |c_i| \cdot \Delta_i = n \cdot \delta$.
- Let $\varepsilon > 0$, $\delta > 0$, and $p \in (0, 1)$. We consider probability distributions P on the set of all vectors

$$(\Delta x_1 \ldots, \Delta x_n) \in [-\delta, \delta] \times \ldots \times [-\delta, \delta].$$

- We say that P is a (p, ε)-*realistic Monte-Carlo estimation (MCE)* if for all $s_i \in \{-1, 1\}$, we have

$$\text{Prob}(s_1 \cdot \Delta x_1 + \ldots + s_n \cdot \Delta x_n \geq n \cdot \delta \cdot (1 - \varepsilon)) \geq p.$$

- **Result.** *If for every n, we have a (p_n, ε)-realistic MCE, then $p_n \leq \beta \cdot n \cdot c^n$ for some $\beta > 0$ and $c < 1$.*
- For probability p_n, we need $1/p_n \sim c^{-n}$ simulations – more than $n + 1$ for numerical differentiation.

35. Why Cauchy Distribution: Formulation of the Problem

- We want to find a family of probability distributions with the following property:
  - when independent $X_1, \ldots, X_n$ have distributions from this family with parameters $\Delta_1, \ldots, \Delta_n$,
  - then each $Y = c_1 \cdot X_1 + \ldots + c_n \cdot X_n \sim \Delta \cdot X$, where X corr. to parameter 1, and $\Delta = \sum_{i=1}^{n} |c_i| \cdot \Delta_i$.
- In particular, for $\Delta_1 = \ldots = \Delta_n = 1$, the desired property of this probability distribution is as follows:
  - if we have n independent identically distributed random variables $X_1, \ldots, X_n$,
  - then each $Y = c_1 \cdot X_1 + \ldots + c_n \cdot X_n$ has the same distribution as $\Delta \cdot X_i$, where $\Delta = \sum_{i=1}^{n} |c_i|$.

36. Analysis of the Problem

- For $n = 1$ and $c_1 = -1$, the desired property says that $-X \sim X$, the distribution is even.
- A usual way to describe a probability distribution is to use a probability density function $\rho(x)$.
- Often, it is convenient to use its Fourier transform – the *characteristic function* $\chi_X(\omega) \stackrel{\text{def}}{=} E[\exp(\mathrm{i} \cdot \omega \cdot X)]$.
- When X_i are independent, then for $S = X_1 + X_2$:

$$\chi_S(\omega) = E[\exp(\mathrm{i} \cdot \omega \cdot S)] = E[\exp(\mathrm{i} \cdot \omega \cdot (X_1 + X_2)] =$$
$$E[\exp(\mathrm{i} \cdot \omega \cdot X_1 + \mathrm{i} \cdot \omega \cdot X_2)] =$$
$$E[\exp(\mathrm{i} \cdot \omega \cdot X_1) \cdot \exp(\mathrm{i} \cdot \omega \cdot X_2)].$$

- Since X_1 and X_2 are independent,

$$\chi_S(\omega) = E[\exp(\mathrm{i} \cdot \omega \cdot X_1)] \cdot E[\exp(\mathrm{i} \cdot \omega \cdot X_2)] = \chi_{X_1}(\omega) \cdot \chi_{X_2}(\omega).$$

37. Analysis of the Problem (cont-d)

- Similarly, for $Y = \sum_{i=1}^{n} c_i \cdot X_i$, we have

$$\chi_Y(\omega) = E[\exp(\mathrm{i}\cdot\omega\cdot Y)] = E\left[\exp\left(\mathrm{i}\cdot\omega\cdot\sum_{i=1}^{n} c_i\cdot X_i\right)\right] =$$

$$E\left[\prod_{i=1}^{n}\exp\left(\mathrm{i}\cdot\omega\cdot c_i\cdot X_i\right)\right] = \prod_{i=1}^{n}\chi_X(\omega\cdot c_i).$$

- The desired property is $Y \sim \Delta\cdot X$, so

$$\prod_{i=1}^{n}\chi_X(\omega\cdot c_i) = \chi_{\Delta\cdot X}(\omega) = E[\exp(\mathrm{i}\cdot\omega\cdot(\Delta\cdot X))]\chi_X(\omega\cdot\Delta),$$

so $\chi_X(c_1\cdot\omega)\cdot\ldots\cdot\chi_X(c_n\cdot\omega) = \chi_X((|c_1|+\ldots+|c_n|)\cdot\omega)$.

- In particular, for $n = 1$, $c_1 = -1$, we get $\chi_X(-\omega) = \chi_X(\omega)$, so $\chi_X(\omega)$ should be an even function.

38. Analysis of the Problem (cont-d)

- Reminder:

$$\chi_X(c_1 \cdot \omega) \cdot \ldots \cdot \chi_X(c_n \cdot \omega) = \chi_X((|c_1| + \ldots + |c_n|) \cdot \omega).$$

- For $n = 2$, $c_1 > 0$, $c_2 > 0$, and $\omega = 1$, we get

$$\chi_X(c_1 + c_2) = \chi_X(c_1) \cdot \chi_X(c_2).$$

- The characteristic function should be measurable.
- *Known:* the only measurable functions with this property are $\chi_X(\omega) = \exp(-k \cdot \omega)$ for some k.
- Due to evenness, for a general ω, we get $\chi_X(\omega) = \exp(-k \cdot |\omega|)$.
- By applying the inverse Fourier transform, we conclude that X is Cauchy distributed.
- *Conclusion:* so, only Cauchy distribution works.

Part V
How General Can We Go: What Is Computable and What Is Not

39. Need to Take Uncertainty Into Account When Processing Data: Reminder

- In practice, we are often interested in a quantity y which is difficult to measure directly.
- *Examples:* distance to a star, amount of oil in the well, tomorrow's weather.
- *Solution:* find easier-to-measure quantities $x_1, \ldots, x_n$ related to y by a known dependence $y = f(x_1, \ldots, x_n)$.
- Then, we measure x_i and use measurement results $\widetilde{x}_i$ to compute an estimate $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$.
- Measurements are never absolutely accurate, so even if the model f is exact, $\widetilde{x}_i \neq x_i$ leads to $\Delta y \stackrel{\text{def}}{=} \widetilde{y} - y \neq 0$.
- It is important to use information about measurement errors $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i$ to estimate the accuracy Δy.

40. We Often Have Imprecise Probabilities

- *Usual assumption:* we know the probabilities for Δx_i.
- To find them, we measure the same quantities:
  - with our measuring instrument (MI) and
  - with a much more accurate MI, with $\widetilde{x}_i^{\rm st} \approx x_i$.
- In two important cases, this does not work:
  - state-of-the-art measurements, and
  - measurements on the shop floor.
- Then, we have partial information about probabilities.
- Often, all we know is an upper bound $|\Delta x_i| \le \Delta_i$.
- Then, we only know that $x_i \in [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$ and

$$y \in [\underline{y}, \overline{y}] \stackrel{\rm def}{=} \{f(x_1, \ldots, x_n) : x_i \in [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]\}.$$

- Computing $[\underline{y}, \overline{y}]$ is known as *interval computation.*

41. How Do We Describe Imprecise Probabilities?

- *Ultimate goal of most estimates:* to make decisions.
- *Known:* a rational decision-maker maximizes expected *utility* $E[u(y)]$.
- For smooth $u(y)$, $y \approx \widetilde{y}$ implies that

$$u(y) = u(\widetilde{x}) + (y - \widetilde{y}) \cdot u'(\widetilde{y}) + \frac{1}{2} \cdot (y - \widetilde{y})^2 \cdot u''(\widetilde{y}).$$

- So, to find $E[u(y)]$, we must know moments $E[(y-\widetilde{y})^k]$.
- Often, $u(x)$ abruptly changes: e.g., when pollution level exceeds y_0; then $E[u(y)] \sim F(y) \stackrel{\text{def}}{=} \text{Prob}(y \le y_0)$.
- From $F(y)$, we can estimate moments, so $F(x)$ is enough.
- Imprecise probabilities mean that we know $F(y)$, we only know bounds (*p-box*) $\underline{F}(y) \le F(y) \le \overline{F}(y)$.

42. What Is Computable?

- Computations with p-boxes are practically important.
- It is thus desirable to come up with efficient algorithms which are as general as possible.
- It is known that too general problems are often *not* computable.
- To avoid wasting time, it is therefore important to find out what *can* be computed.
- At first glance, this question sounds straightforward:
  - to describe a cdf, we can consider a computable function $F(x)$;
  - to describe a p-box, we consider a computable *function interval* $[\underline{F}(x), \overline{F}(x)]$.
- Often, we can do that, but we will show that sometimes, we need to go *beyond* function intervals.

43. Reminder: What Is Computable?

- A real number x corresponds to a value of a physical quantity.
- We can measure x with higher and higher accuracy.
- So, x is called *computable* if there is an algorithm, that, given k, produces a rational r_k s.t. $|x - r_k| \leq 2^{-k}$.
- A *computable function* computes $f(x)$ from x.
- We can only use approximations to x.
- So, an algorithm for computing a function can, given k, request a 2^{-k}-approximation to x.
- Most usual functions are thus computable.
- *Exception:* step-function $f(x) = 0$ for $x < 0$ and $f(x) = 1$ for $x \geq 0$.
- Indeed, no matter how accurately we know $x \approx 0$, from $r_k = 0$, we cannot tell whether $x < 0$ or $x \geq 0$.

44. Consequences for Representing a cdf $F(x)$

- We would like to represent a general probability distribution by its cdf $F(x)$.
- From the purely mathematical viewpoint, this is indeed the most general representation.
- At first glance, it makes sense to consider computable functions $F(x)$.
- For many distributions, e.g., for Gaussian, $F(x)$ is computable.
- However, when $x = 0$ with probability 1, the cdf $F(x)$ is exactly the step-function.
- And we already know that the step-function is not computable.
- Thus, we need to find an alternative way to represent cdf's – beyond computable functions.

45. Back to the Drawing Board

- Each value $F(x)$ is the probability that $X \le x$.
- We cannot empirically find exact probabilities p.
- We can only estimate *frequencies* f based on a sample of size N.
- For large N, the difference $d \stackrel{\text{def}}{=} p - f$ is asymptotically normal, with $\mu = 0$ and $\sigma = \sqrt{\dfrac{p \cdot (1-p)}{N}}$.
- Situations when $|d - \mu| < 6\sigma$ are negligibly rare, so we conclude that $|f - p| \le 6\sigma$.
- For large N, we can get $6\sigma \le \delta$ for any accuracy $\delta > 0$.
- We get a sample $X_1, \ldots, X_N$.
- We don't know the exact values X_i, only measured values $\widetilde{X}_i$ s.t. $|\widetilde{X}_i - X_i| \le \varepsilon$ for some accuracy ε.
- So, what we have is a frequency $f = \text{Freq}(\widetilde{X}_i \le x)$.

46. Resulting Definition

- Here, $X_i \le x - \varepsilon \Rightarrow \widetilde{X}_i \le x \Rightarrow X_i \le x + \varepsilon$, so

$$\text{Freq}(X_i \le x - \varepsilon) \le f = \text{Freq}(\widetilde{X}_i \le x) \le \text{Freq}(X_i \le x + \varepsilon).$$

- Frequencies are δ-close to probabilities, so we arrive at the following:
- *For every x, $\varepsilon > 0$, and $\delta > 0$, we get a rational number f such that $F(x - \varepsilon) - \delta \le f \le F(x + \varepsilon) + \delta$.*
- This is how we define a computable cdf $F(x)$.
- In the computer, to describe a distribution on an interval $[\underline{T}, \overline{T}]$:
  - we select a grid $x_1 = \underline{T}$, $x_2 = \underline{T} + \varepsilon$, ..., and
  - we store the corr. frequencies f_i with accuracy δ.
- A class of possible distribution is represented, for each ε and δ, by a finite list of such approximations.

47. First Equivalent Definition

- *Original:* $\forall x\, \forall \varepsilon_{>0}\, \forall \delta_{>0}$, we get a rational f such that
$$F(x-\varepsilon) - \delta \le f \le F(x+\varepsilon) + \delta.$$
- *Equivalent:* $\forall x\, \forall \varepsilon_{>0}\, \forall \delta_{>0}$, we get a rational f which is δ-close to $F(x')$ for some x' such that $|x' - x| \le \varepsilon$.
- *Proof of equivalence:*
  - We know that $F(x+\varepsilon) - F(x+\varepsilon/3) \to 0$ as $\varepsilon \to 0$.
  - So, for $\varepsilon = 2^{-k}$, $k = 1, 2, \ldots$, we take f and f' s.t.
$$F(x+\varepsilon/3) - \delta/4 \le f \le F(x + (2/3)\cdot\varepsilon) + \delta/4$$
$$F(x + (2/3)\cdot\varepsilon) - \delta/4 \le f' \le F(x+\varepsilon) + \delta/4.$$
  - We stop when f and f' are sufficiently close:
$$|f - f'| \le \delta.$$
  - Thus, we get the desired f.

48. Second Equivalent Definition

- We start with pairs (x_1, f_1), (x_2, f_2), ...
- When $f_{i+1} - f_i > \delta$, we add intermediate pairs
$$(x_i, f_i + \delta), (x_i, f_i + 2\delta), \ldots, (x_i, f_{i+1}).$$
- The resulting set of pairs is (ε, δ)-close to the graph $\{(x, y) : F(x-0) \le y \le F(x)\}$ in Hausdorff metric d_H.
- (x, y) and (x', y') are (ε, δ)-*close* if $|x - x'| \le \varepsilon$ and $|y - y'| \le \delta$.
- The sets S and S' are (ε, δ)-close if:
  - for every $s \in S$, there is a (ε, δ)-close point $s' \in S'$;
  - for every $s' \in S'$, there is a (ε, δ)-close point $s \in S$.
- Compacts with metric d_H form a computable compact.
- So, $F(x)$ is a monotonic computable object in this compact.

49. What Can Be Computed: A Positive Result for the 1D Case

- *Reminder:* we are interested in $F(x)$ and $E_{F(x)}[u(x)]$ for smooth $u(x)$.
- *Reminder:* estimate for $F(x)$ is part of the definition.
- *Question:* computing $E_{F(x)}[u(x)]$ for smooth $u(x)$.
- *Our result:* there is an algorithm that:
  - given a computable cdf $F(x)$,
  - given a computable function $u(x)$, and
  - given accuracy $\delta > 0$,
  - computes $E_{F(x)}[u(x)]$ with accuracy δ.
- For computable classes $\mathcal{F}$ of cdfs, a similar algorithm computes the range of possible values

$$[\underline{u}, \overline{u}] \stackrel{\text{def}}{=} \{E_{F(x)}[u(x)] : F(x) \in \mathcal{F}\}.$$

50. Proof: Main Idea

- Computable functions are computably continuous: for every $\delta > 0$, we can compute $\varepsilon > 0$ s.t.

$$|x - x'| \le \varepsilon \Rightarrow |f(x) - f(x')| \le \delta.$$

- We select ε corr. to $\delta/4$, and take a grid with step $\varepsilon/4$.
- For each x_i, the value f_i is $(\delta/4)$-close to $F(x_i')$ for some x_i' which is $(\varepsilon/4)$-close to x_i.
- The function $u(x)$ is $(\delta/2)$-close to a piece-wise constant function $u'(x) = u(x_i)$ for $x \in [x_i', x_{i+1}']$.
- Thus, $|E[u(x)] - E[u'(x)]| \le \delta/2$.
- Here, $E[u'(x)] = \sum_i u(x_i) \cdot (F(x_{i+1}') - F(x_i'))$.
- Here, $F(x_i')$ is close to f_i and $F(x_{i+1}')$ is close to f_{i+1}.
- Thus, $E[u'(x)]$ (and hence, $E[u(x)]$) is computably close to a computable sum $\sum_i u(x_i) \cdot (f_{i+1} - f_i)$.

51. What to Do in a Multi-D Case?

- For each $g(x)$, y, $\varepsilon > 0$, and $\delta > 0$, we can find a frequency f such that:

$$|P(g(x) \le y') - f| \le \varepsilon \text{ for some } y' \text{ s.t. } |y - y'| \le \delta.$$

- We select an ε-net $x_1, \ldots, x_n$ for X. Then,

$$X = \bigcup_i B_\varepsilon(x_i), \text{ where } B_\varepsilon(x) \stackrel{\text{def}}{=} \{x' : d(x, x') \le \varepsilon\}.$$

- We select f_1 which is close to $P(B_{\varepsilon'}(x_1))$ for all ε' from some interval $[\underline{\varepsilon}, \overline{\varepsilon}]$ which is close to ε.
- We then select f_2 which is close to $P(B_{\varepsilon'}(x_1) \cup B_{\varepsilon'}(x_2))$ for all ε' from some subinterval of $[\underline{\varepsilon}, \overline{\varepsilon}]$, etc.
- Then, we get approximations to probabilities of the sets $B_\varepsilon(x_i) - (B_\varepsilon(x_1) \cup \ldots \cup B_\varepsilon(x_{i-1}))$.
- This lets us compute the desired values $E[u(x)]$.

Part VI
Conclusions and Future Work

52. Conclusions

- In many practical application, we process measurement results and expert estimates.
- Measurements and expert estimates are never absolutely accurate.
- Their result are slightly different from the actual (unknown) values of the corresponding quantities.
- It is therefore desirable to analyze how measurement inaccuracy affects the results of data processing.
- There exist numerous methods for estimating the accuracy of the results of data processing.
- These methods cover different models of inaccuracy: probabilistic, interval, and fuzzy.

53. Conclusions (cont-d)

- To be useful in engineering applications, the uncertainty methods should satisfy the following objectives.
- They should provide accurate estimate for the resulting uncertainty.
- They should not take too much computation time.
- They should be understandable to engineers.
- They should be sufficiently general to cover all kinds of uncertainty.

54. Conclusions (final)

- In this thesis, on several case studies, we show how we can achieve these four objectives.
- We show that we can get more accurate estimates by properly taking model inaccuracy into account.
- We show that we can speed up computations by processing different types of uncertainty differently.
- We show that we can make uncertainty-estimating algorithms more understandable.
- We also analyze how general uncertainty-estimating algorithms can be.

55. Future Work

- In our future work, we plan to continue working in these four directions.
- In particular, we plan to extend our speed-up algorithms from fuzzy to probabilistic uncertainty.
- One of the main reasons for estimation and data processing is to make decisions; we thus plan to analyze:
  - how the corresponding uncertainty affects decision making, and
  - what is the best way to make decisions under different types of uncertainty.
- We plan to apply these algorithms to practical engineering problems, e.g., pavement compaction.

56. Acknowledgments

I want to express my gratitude to my committee members:

- Vladik Kreinovich, Chair
- Jack Chessa,
- Aaron Velasco, and
- Piotr Wojciechowski.

Part VII
Proofs

57. Proof of the Main Result from Part 4

- Let us pick some $\alpha \in (0,1)$.
- Let us denote, by m, the number of indices i or which $s_i \cdot \Delta x_i > \alpha \cdot \delta$.
- If we have $s_1 \cdot \Delta x_1 + \ldots + s_n \cdot \Delta x_n \geq n \cdot \delta \cdot (1-\varepsilon)$, then:
  - for $n-m$ indices, we have $s_i \cdot \Delta x_i \leq \alpha \cdot \delta$ and
  - for the other m indices, we have $s_i \cdot \Delta x_i \leq \delta$.
- Thus, $n \cdot \delta \cdot (1-\varepsilon) \leq \sum_{i=1}^{n} s_i \cdot \Delta x_i \leq m \cdot \delta + (n-m) \cdot \alpha \cdot \delta$.
- Dividing this inequality by δ, we get
$$n \cdot (1-\varepsilon) \leq m + (n-m) \cdot \alpha.$$
- So, $n \cdot (1-\alpha-\varepsilon) \leq m \cdot (1-\alpha)$ and $m \geq n \cdot \dfrac{1-\alpha-\varepsilon}{1-\alpha}$.
- So, we have at least $n \cdot \dfrac{1-\alpha-\varepsilon}{1-\alpha}$ indices for which Δx_i has the same sign as s_i (and for which $|\Delta x_i| > \alpha \cdot \delta$).

58. Proof from Part 4 (cont-d)

- So, for Δx_i corr. to $(s_1, \ldots, s_n)$, at most $n \cdot \dfrac{\varepsilon}{1-\alpha-\varepsilon}$ indices have a different sign than s_i.
- It is possible that the same tuple Δx can serve two tuples $s \neq s'$. In this case:
  - going from s_i to $\text{sign}(\Delta x_i)$ changes at most $n \cdot \dfrac{\varepsilon}{1-\alpha-\varepsilon}$ signs, and
  - going from $\text{sign}(\Delta x_i)$ to s'_i also changes at most $n \cdot \dfrac{\varepsilon}{1-\alpha-\varepsilon}$ signs.
- Thus, between the tuples s and s', at most $2 \cdot \dfrac{\varepsilon}{1-\alpha-\varepsilon}$ signs are different.
- In other words, for the Hamming distance $d(s, s') \stackrel{\text{def}}{=} \#\{i : s_i \neq s'_i\}$, we have $d(s, s') \leq 2 \cdot n \cdot \dfrac{\varepsilon}{1-\alpha-\varepsilon}$.

59. Proof from Part 4 (cont-d)

- Thus, if $d(s, s') > 2 \cdot n \cdot \dfrac{\varepsilon}{1-\alpha-\varepsilon}$, then no tuples $(\Delta x_1, \ldots, \Delta x_n)$ can serve both sign tuples s and s'.
- In this case, the two sets of tuples Δx do not intersect:
  - tuples s.t. $s_1 \cdot \Delta x_1 + \ldots + s_n \cdot \Delta x_n \geq n \cdot \delta \cdot (1-\varepsilon)$;
  - tuples s.t. $s'_1 \cdot \Delta x_1 + \ldots + s'_n \cdot \Delta x_n \geq n \cdot \delta \cdot (1-\varepsilon)$.
- Let's take take M sign tuples $s^{(1)}, \ldots, s^{(M)}$ for which $d(s^{(i)}, s^{(j)}) > 2 \cdot \dfrac{\varepsilon}{1-\alpha-\varepsilon}$ for all $i \neq j$.
- Then the probability P that Δx serves one of these sign tuples is $\geq M \cdot p$.
- Since $P \leq 1$, we have $p \leq \dfrac{1}{M}$; so:
  - to prove that p_n is exponentially decreasing,
  - it is sufficient to find the sign tuples whose number M is exponentially increasing.

60. Proof from Part 4 (cont-d)

- Let us denote $\beta \stackrel{\text{def}}{=} \dfrac{\varepsilon}{1-\alpha-\varepsilon}$.
- Then, for each sign tuple s, the number t of all sign tuples s' for which $d(s, s') \leq \beta \cdot n$ is equal to the sum of:
  - the number of tuples $\binom{n}{0}$ that differ from s in 0 places,
  - the number of tuples $\binom{n}{1}$ that differ from s in 1 place, ...,
  - the number of tuples $\binom{n}{\beta \cdot n}$ that differ from s in $\beta \cdot n$ places,
- Thus, $t = \binom{n}{0} + \binom{n}{1} + \ldots + \binom{n}{n \cdot \beta}$.

61. Proof from Part 4 (cont-d)

- When $\beta < 0.5$ and $\beta \cdot n < \dfrac{n}{2}$, the number of combinations $\binom{n}{k}$ increases with k, so $t \leq \beta \cdot n \cdot \binom{n}{\beta \cdot n}$.
- Here, $\binom{a}{b} = \dfrac{a!}{b! \cdot (a-b)!}$. Since $n! \sim \left(\dfrac{n}{e}\right)^n$, we have

$$t \leq \beta \cdot n \cdot \left(\frac{1}{\beta^\beta \cdot (1-\beta)^{1-\beta}}\right)^n.$$

- Here, $\gamma \stackrel{\text{def}}{=} \dfrac{1}{\beta^\beta \cdot (1-\beta)^{1-\beta}} = \exp(S)$, where $S \stackrel{\text{def}}{=} -\beta \cdot \ln(\beta) - (1-\beta) \cdot \ln(1-\beta)$ is Shannon's entropy.
- It is known that S attains its largest value when $\beta = 0.5$, in which case $S = \ln(2)$ and $\gamma = \exp(S) = 2$.
- When $\beta < 0.5$, we have $S < \ln(2)$, thus, $\gamma < 2$, and $t \leq \beta \cdot n \cdot \gamma^n$ for some $\gamma < 2$.

62. Proof from Part 4 (cont-d)

- Let us now construct the desired collection of sign tuples $s^{(1)}, \ldots, s^{(M)}$.
  - We start with some sign tuple $s^{(1)}$, e.g., $s^{(1)} = (1, \ldots, 1)$.
  - Then, we dismiss $t \le \gamma^n$ tuples which are $\le \beta$-close to s, and select one of the remaining tuples as $s^{(2)}$.
  - We then dismiss $t \le \gamma^n$ tuples which are $\le \beta$-close to $s^{(2)}$.
  - Among the remaining tuples, we select the tuple $s^{(3)}$, etc.
- Once we have selected M tuples, we have thus dismissed $t \cdot M \le \beta \cdot n \cdot \gamma^n \cdot M$ sign tuples.
- So, as long as this number is smaller than the overall number 2^n of sign tuples, we can continue selecting.

63. Proof from Part 4 (conclusions)

- Our procedure ends when we have selected M tuples for which $\beta \cdot n \cdot \gamma^n \cdot M \geq 2^n$.
- Thus, we have selected $M \geq \left(\frac{2}{\gamma}\right)^n \cdot \frac{1}{\beta \cdot n}$ tuples.
- So, we have indeed selected exponentially many tuples.
- Hence, $p_n \leq \frac{1}{M} \leq \beta \cdot n \cdot \left(\frac{\gamma}{2}\right)^n$, i.e.,

$$p_n \leq \beta \cdot n \cdot c^n, \text{ where } c \stackrel{\text{def}}{=} \frac{\gamma}{2} < 1.$$

- So, the probability p_n is indeed exponentially decreasing. The main result is proven.