How to Take Into Account Model Inaccuracy When Estimating the Uncertainty of the Result of Data Processing

Vladik Kreinovich, Olga Kosheleva,
Andrzej Pownuk, and Rodrigo Romero

Cyber-ShARE Center
University of Texas at El Paso
El Paso, Texas 79968, USA
vladik@utep.edu, olgak@utep.edu
ampownuk@utep.edu, raromero2@utep.edu

1. Bounds on Unwanted Processes: An Important Part of Engineering Specifications

- An engineering system performs certain tasks.
- However, it also generates undesirable side effects: noise, vibration, heat, stress, etc.
- The size q of each such effect should not exceed a certain pre-defined threshold t.
- It is therefore important to check that $q \le t$ in all possible situations.
- Let $p_1, \ldots, p_n$ be parameters that describe different situations: wind speed, load, Young module.
- For each of these parameters, we know the interval of possible values $[\underline{p}_i, \overline{p}_i] = [\widetilde{p}_i - \Delta_i, \widetilde{p}_i + \Delta_i]$.

2. Bounds on Unwanted Processes (cont-d)

- We want to make sure that $q \le t$ for *all* possible combinations of $p_i \in [\underline{p}_i, \overline{p}_i]$.
- Even if we consider extreme cases, when $p_i = \underline{p}_i$ or $p_i = \overline{p}_i$, we get 2^n cases.
- For large n, it is not feasible to physically check all these cases.
- Thus, we need to rely on computer simulations.

3. Formulation of the Problem

- There exist techniques for checking that $q \le t$ for all $p_i \in [\underline{p}_i, \overline{p}_i]$.
- However, these techniques assume that we have an *exact* model of the system.
- In many cases, we only have an *approximate* description information of the system.
- We show that in such cases, the existing techniques overestimate uncertainty.
- We also show that a proper modification of these techniques drastically decreases this overestimation.

4. How to Check Specifications When We Have an Exact Model of a System: Reminder

- Let us assume that we know the exact dependence $q = q(p_1, \ldots, p_n)$.
- Usually, deviations $\Delta p_i = p_i - \widetilde{p}_i$ from nominal values $\widetilde{p}_i$ are reasonably small.
- In such situations, we can linearize the dependence:

$$q(p_1, \ldots, p_n) = \widetilde{q} + \sum_{i=1}^{n} c_i \cdot \Delta p_i, \text{ where}$$

$$\widetilde{q} \stackrel{\text{def}}{=} q(\widetilde{x}_1, \ldots, \widetilde{x}_n) \text{ and } c_i \stackrel{\text{def}}{=} \frac{\partial q}{\partial p_i}.$$

- The largest value $\overline{q}$ is attained when $\Delta p_i = \pm \Delta_i$:

$$\overline{q} = \widetilde{q} + \sum_{i=1}^{n} |c_i| \cdot \Delta_i.$$

5. What If We Have an Exact Model (cont-d)

- Here, $\overline{q} = \widetilde{q} + \sum\limits_{i=1}^{n} |c_i| \cdot \Delta_i$.
- When the expression for $q(p_i)$ is implicit, we cannot explicitly compute c_i.
- In this case, we can use numerical differentiation

$$c_i = \frac{q(\widetilde{p}_1, \ldots, \widetilde{p}_{i-1}, \widetilde{p}_i + h_i, \widetilde{p}_{i+1}, \ldots, \widetilde{p}_n) - \widetilde{q}}{h_i}.$$

- Then, for $h_i = \Delta_i$, we get $\overline{q} = \widetilde{q} + \sum\limits_{i=1}^{n} |q_i - \widetilde{q}|$, where

$$q_i \stackrel{\text{def}}{=} q(\widetilde{p}_1, \ldots, \widetilde{p}_{i-1}, \widetilde{p}_i + \Delta_i, \widetilde{p}_{i+1}, \ldots, \widetilde{p}_n).$$

6. Resulting Algorithm

- *We know:* an algorithm $q(p_1, \ldots, p_n)$, a threshold t, and values $\widetilde{p}_i$ and Δ_i.
- *We need to check:* whether $q(p_1, \ldots, p_n) \le t$ for all $p_i \in [\widetilde{p}_i - \Delta_i, \widetilde{p}_i + \Delta_i]$.
- *Algorithm:*

  1) first, we compute $\widetilde{q} = q(\widetilde{p}_1, \ldots, \widetilde{p}_n)$;

  2) then, for each i from 1 to n, we compute

$$q_i = q(\widetilde{p}_1, \ldots, \widetilde{p}_{i-1}, \widetilde{p}_i + \Delta_i, \widetilde{p}_{i+1}, \ldots, \widetilde{p}_n);$$

  3) after that, we compute $\overline{q} = \widetilde{q} + \sum\limits_{i=1}^{n} |q_i - \widetilde{q}|$;

  4) finally, we check whether $\overline{q} \le t$.

7. Possibility of a Further Speed-Up

- The above algorithm requires $n+1$ calls to the program that computes q.
- In many practical situations, this is too long.
- We can speed up computations if we Cauchy distribution $\rho(x)=\dfrac{1}{\pi\cdot\Delta}\cdot\dfrac{1}{1+\left(\dfrac{x}{\Delta}\right)^2}$.
- If η_i are independent Cauchy distributed with parameters Δ_i, then $\sum\limits_{i=1}^n c_i\cdot\eta_i$ is also Cauchy distributed, with

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

- Thus, we can find Δ by using the following algorithm.

8. Faster Algorithm

- *Algorithm:*

  1) first, for $k = 1, \ldots, N$, we simulate $\eta_i^{(k)}$ Cauchy-distributed with parameters Δ_i;

  2) for each k, we estimate $\Delta y^{(k)} = \sum\limits_{i=1}^{n} c_i \cdot \eta_i^{(k)}$ as

  $$\Delta y^{(k)} = q(\widetilde{p}_1 + \eta_1^{(k)}, \ldots, \widetilde{p}_n + \eta_n^{(k)}) - \widetilde{y};$$

  3) based on N values $\Delta y^{(1)}$, $\ldots$, $\Delta y^{(N)}$ which are Cauchy-distributed with parameter Δ, we find Δ;

  4) finally, we compute $\overline{q} = \widetilde{q} + \Delta$.

- In this algorithm, we need $N + 1$ computations of q.
- The accuracy depends only on the sample size N and *not* on the number of inputs n.
- *Example:* $N = 100$ leads to 20% accuracy.
- So, for $n \gg 200$, this method is much faster.

9. For Many Practical Problems, We Can Achieve an Even Faster Speed-Up

- Often, once we have $\widetilde{q} = q(\widetilde{p}_1, \ldots, \widetilde{p}_n)$, we can compute $q(\widetilde{p}_1 + \eta_1, \ldots, \widetilde{p}_n + \eta_n)$ faster than by applying q.
- For example, often, $q(p_1, \ldots, p_n)$ comes from solving a system of nonlinear equations

$$F_j(q_1, \ldots, q_k, p_1, \ldots, p_n) = 0.$$

- Since $\Delta p_i = \eta_i \ll p_i$, we can linearize, solve the resulting easy-to-solve linear system, and get

$$\Delta q = q(\widetilde{p}_1 + \eta_1, \ldots, \widetilde{p}_n + \eta_n) - \widetilde{q}.$$

- A similar simplifying linearization is possible when q comes from solving a system of nonlinear diff. eqs.
- This idea – known as *local sensitivity analysis* – is successfully used in many practical applications.

10. Taking Model Inaccuracy into Account

- We rarely know the exact dependence $q(p_1, \ldots, p_n)$.
- Usually, we have an approximate model $Q(p_1, \ldots, p_n)$ with known accuracy ε:

$$|Q(p_1, \ldots, p_n) - q(p_1, \ldots, p_n)| \leq \varepsilon.$$

- *We know:* an algorithm $Q(p_1, \ldots, p_n)$, accuracy ε, threshold t, values $\widetilde{p}_i$ and Δ_i.
- *We want:* to check whether $q(p_1, \ldots, p_n) \leq t$ for all $p_i \in [\widetilde{p}_i - \Delta_i, \widetilde{p}_i + \Delta_i]$.
- If we use this approximate model in our estimate, we get $\overline{Q} = \widetilde{Q} + \sum\limits_{i=1}^{n} |Q_i - \widetilde{Q}|$.
- Here, $|\widetilde{Q} - \widetilde{q}| \leq \varepsilon$ and $|Q_i - q_i| \leq \varepsilon$, so $|\overline{q} - \overline{Q}| \leq (2n+1) \cdot \varepsilon$.
- Thus, we arrive at the following algorithm.

11. Resulting Algorithm

- *We know:* an algorithm $Q(p_1, \ldots, p_n)$, accuracy ε, threshold t, values $\widetilde{p}_i$ and Δ_i.
- *We want:* to check whether $q(p_1, \ldots, p_n) \le t$ for all $p_i \in [\widetilde{p}_i - \Delta_i, \widetilde{p}_i + \Delta_i]$.
- *Algorithm:*

  1) compute $\widetilde{Q} = Q(\widetilde{p}_1, \ldots, \widetilde{p}_n)$ and

  $$Q_i = Q(\widetilde{p}_1, \ldots, \widetilde{p}_{i-1}, \widetilde{p}_i + \Delta_i, \widetilde{p}_{i+1}, \ldots, \widetilde{p}_n).$$

  2) compute $B = \widetilde{Q} + \sum\limits_{i=1}^{n} |Q_i - \widetilde{Q}| + (2n+1) \cdot \varepsilon$;

  3) check whether $B \le t$.
- *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 $\widetilde{q}$.

12. 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 $\Delta q \stackrel{\text{def}}{=} Q(p_1, \ldots, p_n) - q(p_1, \ldots, p_n)$ as *independent* random variables.
- What is a probability distribution for these random variables? We know that $\Delta q \in [-\varepsilon, \varepsilon]$.
- 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 $\sigma = \dfrac{\varepsilon}{\sqrt{3}}$.

13. How to Get a Better Estimate for $\widetilde{q}$

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

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

- Here, $\Delta\widetilde{q}_{\text{new}} = \dfrac{1}{n+2} \cdot \left(\Delta\widetilde{q} + \sum\limits_{i=1}^n \Delta q_i + \Delta m\right)$, so its variance is $\sigma^2\left[\widetilde{Q}_{\text{new}}\right] = \dfrac{\varepsilon^2}{3 \cdot (n+2)} \ll \dfrac{\varepsilon^2}{3} = \sigma^2\left[\widetilde{Q}\right]$.

14. Let Us Use $\widetilde{Q}_{\text{new}}$ When Estimating $\overline{q}$

- Let us compute $\overline{Q}_{\text{new}} = \widetilde{Q}_{\text{new}} + \sum\limits_{i=1}^{n} |Q_i - \widetilde{Q}_{\text{new}}|$.
- Here, when $s_i \in \{-1, 1\}$ are the signs of $q_i - \widetilde{q}$, we get:

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

- Thus, $\Delta\overline{q}_{\text{new}} = \left(1 - \sum\limits_{i=1}^{n} s_i\right) \cdot \Delta\widetilde{q}_{\text{new}} + \sum\limits_{i=1}^{n} s_i \cdot \Delta q_i$, 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 $\left|1 - \sum\limits_{i=1}^{n} s_i\right| \le n + 1$, and

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

15. Using $\widetilde{Q}_{\text{new}}$ (cont-d)

- We have $\Delta\overline{q}_{\text{new}} = \left(1 - \sum\limits_{i=1}^{n} s_i\right) \cdot \Delta\widetilde{q}_{\text{new}} + \sum\limits_{i=1}^{n} s_i \cdot \Delta q_i.$
- Due to the Central Limit Theorem, $\Delta\overline{q}_{\text{new}}$ 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

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

  - 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 $\sim 2n+1$.

16. Resulting Algorithm

- *We know:* $Q(p_1,\ldots,p_n)$, ε, t, $\widetilde{p}_i$ and Δ_i.
- *We want:* to check that $q(p_1,\ldots,p_n)\le t$ for all $p_i\in[\widetilde{p}_i-\Delta_i,\widetilde{p}_i+\Delta_i]$.
- *Algorithm:*

1) compute $\widetilde{Q}=Q(\widetilde{p}_1,\ldots,\widetilde{p}_n)$,

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

$$Q_i=Q(\widetilde{p}_1,\ldots,\widetilde{p}_{i-1},\widetilde{p}_i+\Delta_i,\widetilde{p}_{i+1},\ldots,\widetilde{p}_n);$$

2) compute $\widetilde{Q}_{\rm new}=\dfrac{1}{n+2}\cdot\left(\widetilde{Q}+\sum\limits_{i=1}^n Q_i+M\right)$ and

$$b=\widetilde{Q}_{\rm new}+\sum_{i=1}^n\left|Q_i-\widetilde{Q}_{\rm new}\right|+k_0\cdot\sqrt{2n+1}\cdot\frac{\varepsilon}{\sqrt{3}};$$

3) check whether $b\le t$.

17. A Similar Improvement Is Possible for the Cauchy Method

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

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

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

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

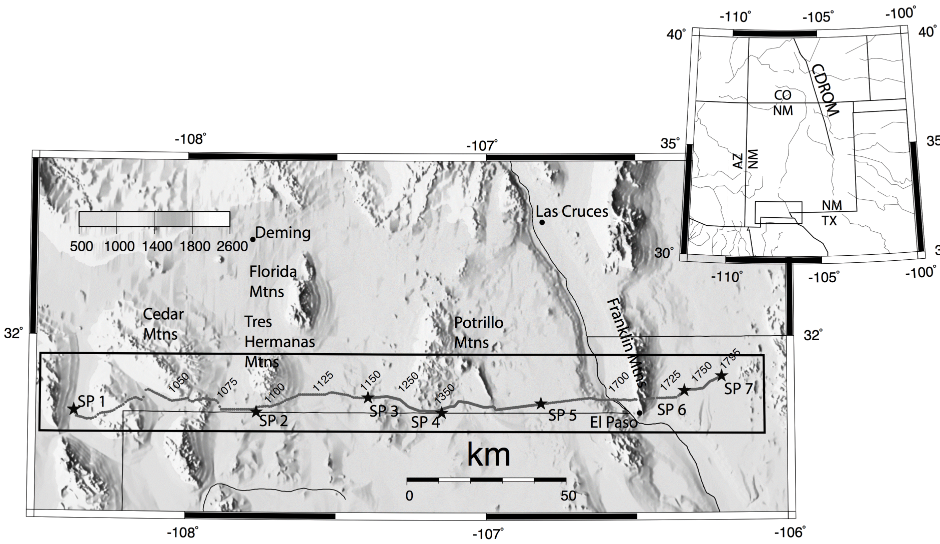
- We can now use this improved estimate when estimating the differences $\Delta y^{(k)}$: namely, we compute

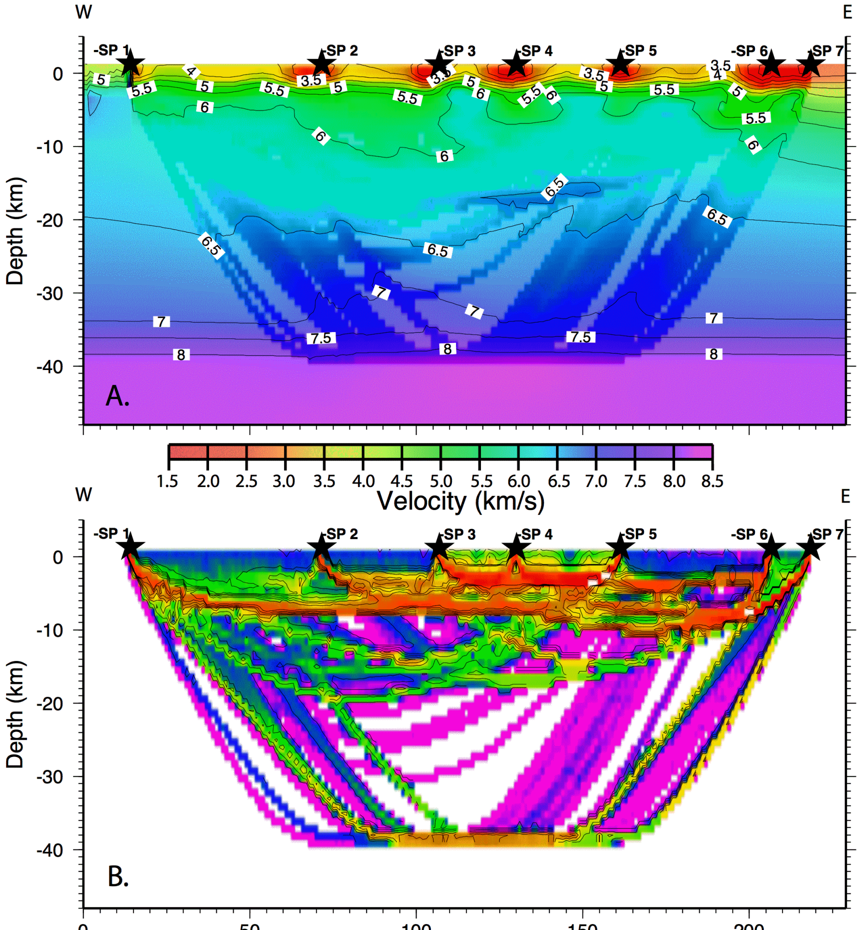
$$Y^{(k)} - \widetilde{Q}_{\text{new}}.$$

18. 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 $\widetilde{Q}_{\text{new}}$ decreased the inaccuracy σ, on average, by 15%; σ increased only in one case (only by 7%).

19. Case Study: Seismic Inverse Problem in the Geosciences




W
E
-SP 1
SP 2
SP 3
SP 4
SP 5
-SP 6
SP 7
Depth (km)
0
-10
-20
-30
-40
A.
1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5
Velocity (km/s)
B.
0
50
100
150
200

20. Can We Further Improve the Accuracy?

- The inaccuracy $Q \neq q$ 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 $\widetilde{q}$ as

$$Q_K(p_1, \ldots, p_n) = \frac{1}{K} \cdot \sum_{k=1}^{K} Q\left(p_1 + \Delta_1^{(k)}, \ldots, p_n + \Delta_n^{(k)}\right).$$

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

21. Acknowledgments

This work was supported in part by the National Science Foundation grants:

- HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence), and
- DUE-0926721.

Home Page
Title Page
Page 23 of 23
Go Back
Full Screen
Close
Quit