Towards A Neural-Based Understanding of the Cauchy Deviate Method for Processing Interval and Fuzzy Uncertainty

Vladik Kreinovich[1] and Hung T. Nguyen[2]

[1]Department of Computer Science
University of Texas, El Paso, TX 79968, USA, vladik@utep.edu
[2]Department of Mathematical Sciences, New Mexico State University,
Las Cruces, NM 88003, USA, hunguyen@nmsu.edu

1. Practical Need for Uncertainty Propagation

- *Practical problem:* we are often interested in the quantity y which is difficult to measure directly.
- *Solution:*
  - estimate easier-to-measure quantities $x_1, \ldots, x_n$ which are related to y by a known algorithm $y = f(x_1, \ldots, x_n)$;
  - compute $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ based on the estimates $\widetilde{x}_i$.
- *Fact:* estimates are never absolutely accurate: $\widetilde{x}_i \neq x_i$.
- *Consequence:* the estimate $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ is different from the actual value $y = f(x_1, \ldots, x_n)$.
- *Problem:* estimate the uncertainty $\Delta y \stackrel{\text{def}}{=} \widetilde{y} - y$.

2. Propagation of Probabilistic Uncertainty

- *Fact:* often, we know the probabilities of different values of Δx_i.
- *Example:* Δx_i are independent normally distributed with mean 0 and known st. dev. σ_i.
- *Monte-Carlo approach:*
  - For $k = 1, \ldots, N$ times, we:
    - simulate the values $\Delta x_i^{(k)}$ according to the known probability distributions for x_i;
    - find $x_i^{(k)} = \widetilde{x}_i - \Delta x_i^{(k)}$;
    - find $y^{(k)} = f(x_1^{(k)}, \ldots, x_n^{(k)})$;
    - estimate $\Delta y^{(k)} = y^{(k)} - \widetilde{y}$.
  - Based on the sample $\Delta y^{(1)}, \ldots, \Delta y^{(N)}$, we estimate the statistical characteristics of Δy.

3. Propagation of Interval Uncertainty

- *In practice:* we often do not know the probabilities.
- *What we know:* the upper bounds Δ_i on the measurement errors Δx_i: $|\Delta x_i| \le \Delta_i$.
- *Enter intervals:* once we know $\widetilde{x}_i$, we conclude that the actual (unknown) x_i is in the interval

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

- *Problem:* find the range $\mathbf{y} = [\underline{y}, \overline{y}]$ of possible values of y when $x_i \in \mathbf{x}_i$:

$$\mathbf{y} = f(\mathbf{x}_1, \ldots, \mathbf{x}_n) \stackrel{\text{def}}{=}$$

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

- *Fact:* this *interval computation* problem is, in general, NP-hard.

4. Propagation of Fuzzy Uncertainty

- In many practical situations, the estimates $\widetilde{x}_i$ come from experts.
- Experts often describe the inaccuracy of their estimates by natural language terms like "approximately 0.1".
- A natural way to formalize such terms is to use membership functions $\mu_i(x_i)$.
- For each α, we can determine the α-*cut*

$$\mathbf{x}_i(\alpha) = \{x_i \,|\, \mu_i(x_i) \geq \alpha\}.$$

- Natural idea: find $\mu(y)$ for which, for each α,

$$\mathbf{y}(\alpha) = f(\mathbf{x}_1(\alpha), \ldots, \mathbf{x}_1(\alpha)).$$

- So, the problem of propagating fuzzy uncertainty can be reduced to several interval propagation problems.

5. Need for Faster Algorithms for Uncertainty Propagation

- For propagating probabilistic uncertainty, there are efficient algorithms such as Monte-Carlo simulations.
- In contrast, the problems of propagating interval and fuzzy uncertainty are computationally difficult.
- It is therefore desirable to design faster algorithms for propagating interval and fuzzy uncertainty.
- The problem of propagating fuzzy uncertainty can be reduced to the interval case.
- Hence, we mainly concentrate on faster algorithms for propagating interval uncertainty.

6. Linearization

- In many practical situations, the errors Δx_i are small, so we can ignore quadratic terms:

$$\Delta y = \widetilde{y} - y = f(\widetilde{x}_1, \ldots, \widetilde{x}_n) - f(x_1, \ldots, x_n) =$$
$$f(\widetilde{x}_1, \ldots, \widetilde{x}_n) - f(\widetilde{x}_1 - \Delta x_1, \ldots, \widetilde{x}_n - \Delta x_n) \approx$$
$$c_1 \cdot \Delta x_1 + \ldots + c_n \cdot \Delta x_n,$$

  where $c_i \stackrel{\text{def}}{=} \dfrac{\partial f}{\partial x_i}(\widetilde{x}_1, \ldots, \widetilde{x}_n)$.

- For a linear function, the largest Δy is obtained when each term $c_i \cdot \Delta x_i$ is the largest:

$$\Delta = |c_1| \cdot \Delta_1 + \ldots + |c_n| \cdot \Delta_n.$$

- Due to the linearization assumption, we can estimate each partial derivative c_i as

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

7. Linearization: Algorithm

To compute the range $\mathbf{y}$ of y, we do the following.

- First, we apply the algorithm f to the original estimates $\widetilde{x}_1, \ldots, \widetilde{x}_n$, resulting in the value $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$.
- Second, for all i from 1 to n,
  - we compute $f(\widetilde{x}_1, \ldots, \widetilde{x}_{i-1}, \widetilde{x}_i + h_i, \widetilde{x}_{i+1}, \ldots, \widetilde{x}_n)$ for some small h_i and then
  - we compute

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

- Finally, we compute $\Delta = |c_1| \cdot \Delta_1 + \ldots + |c_n| \cdot \Delta_n$ and the desired range $\mathbf{y} = [\widetilde{y} - \Delta, \widetilde{y} + \Delta]$.
- *Problem:* we need $n+1$ calls to f, and this is often too long.

8. Cauchy Deviate Method: Idea

- For large n, we can further reduce the number of calls to f if we Cauchy distributions, w/pdf

$$\rho(z) = \frac{\Delta}{\pi \cdot (z^2 + \Delta^2)}.$$

- Known property of Cauchy transforms:
  - if $z_1, \ldots, z_n$ are independent Cauchy random variables w/parameters $\Delta_1, \ldots, \Delta_n$,
  - then $z = c_1 \cdot z_1 + \ldots + c_n \cdot z_n$ is also Cauchy distributed, w/parameter

$$\Delta = |c_1| \cdot \Delta_1 + \ldots + |c_n| \cdot \Delta_n.$$

- This is exactly what we need to estimate interval uncertainty!

9. Cauchy Deviate Method: Towards Implementation

- To implement the Cauchy idea, we must answer the following questions:
  - how to simulate the Cauchy distribution; and
  - how to estimate the parameter Δ of this distribution from a finite sample.
- Simulation can be based on the functional transformation of uniformly distributed sample values:

$$\delta_i = \Delta_i \cdot \tan(\pi \cdot (r_i - 0.5)), \text{ where } r_i \sim U([0,1]).$$

- To estimate Δ, we can apply the Maximum Likelihood Method $\rho(\delta^{(1)}) \cdot \rho(\delta^{(2)}) \cdot \ldots \cdot \rho(\delta^{(N)}) \to \max$, i.e., solve

$$\frac{1}{1 + \left(\frac{\delta^{(1)}}{\Delta}\right)^2} + \ldots + \frac{1}{1 + \left(\frac{\delta^{(N)}}{\Delta}\right)^2} = \frac{N}{2}.$$

10. Cauchy Deviates Method: Algorithm

- Apply f to $\widetilde{x}_i$; we get $\widetilde{y} := f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$.
- For $k = 1, 2, \ldots, N$, repeat the following:
  - use the standard RNG to draw $r_i^{(k)} \sim U([0,1])$, $i = 1, 2, \ldots, n$;
  - compute Cauchy distributed values $c_i^{(k)} := \tan(\pi \cdot (r_i^{(k)} - 0.5))$;
  - compute $K := \max_i |c_i^{(k)}|$ and normalized errors $\delta_i^{(k)} := \Delta_i \cdot c_i^{(k)}/K$;
  - compute the simulated "actual values" $x_i^{(k)} := \widetilde{x}_i - \delta_i^{(k)}$;
  - compute simulated errors of indirect measurement: $\delta^{(k)} := K \cdot \left(\widetilde{y} - f\left(x_1^{(k)}, \ldots, x_n^{(k)}\right)\right)$;
- Compute Δ by applying the bisection method to solve the Maximum Likelihood equation.

11. Important Comment

- To avoid confusion, we should emphasize that:
  - in contrast to the Monte-Carlo solution for the probabilistic case,
  - the use of Cauchy distribution in the interval case is a computational trick,
  - it is *not* a truthful simulation of the actual measurement error Δx_i.
- Indeed:
  - we know that the actual value of Δx_i is always inside the interval $[-\Delta_i, \Delta_i]$, but
  - a Cauchy distributed random attains values outside this interval as well.

12. Cauchy Deviate Method: Need for Intuitive Explanation

- *Fact:* the Cauchy deviate method is mathematically valid.
- *Problem:* this method is somewhat counterintuitive:
  - we want to analyze errors which are located *instead* a given interval $[-\Delta, \Delta]$, but
  - this analysis use Cauchy simulated errors which are located *outside* this interval.
- It is therefore desirable to come up with an intuitive explanation for this technique.
- In this talk, we show that such an explanation can be obtained from neural networks.

13. Werbos's Idea: Use Neurons

- *Traditionally:* neural networks are used to simulate a deterministic dependence.
- *Paul Werbos* suggested that the same neural networks can be used to describe stochastic dependencies as well.
- *How:* as one of the inputs, we take a random number $r \sim U([0,1])$.
- *Simplest case:* a single neuron.
- *In this case:* we apply the activation (input-output) function $f(y)$ to the random number r.
- *What we do:* let us analyze the resulting distribution of $f(r)$.
- *Question:* which $f(y)$ should we use?

14. We Must Choose a Family of Functions, Not a Single Function

- *Changing units:* if $f \in F$, then $k \cdot f \in F$.
- *Conclusion:* in mathematical terms, we choose a *family* F of functions f.
- *Changing starting point:* if $f \in F$, then $f + c \in F$.
- *Non-linear changes:* since NN are useful in non-linear case, we consider $f(y) \to g(f(y))$ for non-linear $g \in G$.
- *Natural requirement:* G is closed under composition and depends on finitely many parameters.
- *Result:* any finite-D group G containing all linear f-s has fractional-linear ones.
- *Conclusion:* $F = \{g(f(x)) : g \in G\}$.

15. Which Family is the Best?

- *Optimality criterion* is not necessary numerical:
  - we can choose F with smallest approximation error,
  - among such F, the fastest to compute.
- *General idea:* a partial (pre-)order.
- *Shift-invariance:* if $F > G$, then $T_a(F) > T_a(G)$, where $T_a(F) = \{f(x+a) \mid f \in F\}$.
- *Finality:*
  - if several families are optimal w.r.t. some criterion,
  - we can use this non-uniqueness to select the one with some additional good qualities;
  - in effect, we this change a criterion to a new one in which the optimal family is unique;
  - thus, in the *final* criterion, there is only one optimal family.

16. Main Result

Theorem.

- *Let a* F *be optimal in the sense of some optimality criterion that is final and shift-invariant.*
- *Then* $f \in F$ *has the form* $a + b \cdot s_0(K \cdot y + l)$ *for some* a, b, K *and* l, *where* $s_0(y)$ *is*
  - *either a linear or fractional-linear function,*
  - *or* $s_0(y) = \exp(y)$,
  - *or the logistic function* $s_0(y) = 1/(1 + \exp(-y))$,
  - *or* $s_0(y) = \tan(y)$.

Comments.

- The logistic function is indeed the most popular activation in NN, but others are also used.
- $\tan(r)$ leads to the desired Cauchy distribution.

17. Acknowledgments

This work was supported in part:

- by NSF grant HRD-0734825 and
- by Grant 1 T36 GM078000-01 from the National Institutes of Health.