Locating Local Extrema under Interval Uncertainty: Multi-D Case

Karen Villaverde[1] and Vladik Kreinovich[2]

[1]Department of Computer Science
New Mexico State University
Las Cruces, NM 88003, USA
kvillave@cs.nmsu.edu

[2]Department of Computer Science
University of Texas at El Paso
El Paso, TX 79968, USA
vladik@utep.edu

1. The Problem of Locating Local Extrema Is Important

- In *spectral analysis*, chemical species are identified by locating local maxima of the spectra.
- In *radioastronomy*, radiosources are identified as local maxima of the measured brightness.
- *Elementary particles* are identified as local maxima of scattering y as a function of energy t.
- Different *clusters* correspond to local maxima of the probability density function.

2. The Problem of Locating Local Extrema: Precise Formulation

In each of these applications, the following problem arises:

- we know that a physical quantity y is a function of one or several ($m \ge 1$) other physical quantities $t_1, \ldots, t_m$:

$$y = f(t_1, \ldots, t_m);$$

- we have n situations, $i = 1, \ldots, n$, in each of which we know the values of all m quantities: $v_i = (t_{i1}, \ldots, t_{im})$;
- in each of these n situations, we have measured the values $y_1 = f(v_1), \ldots, y_n = f(v_n)$ of the quantity y;
- based on this information, we want to locate the local maxima and/or the local minima of the function f.

3. Need to Take into Account Interval Uncertainty

- Observed values $y_i = f(v_i)$ come from measurements, and measurements are never absolutely accurate.
- The measurement results $\widetilde{y}_i$ are, in general, different from the actual (unknown) values y_i.
- In some cases, we know the probabilities of different values of the measurement error $\Delta y_i \stackrel{\text{def}}{=} \widetilde{y}_i - y_i$.
- In many practical cases, however, we only know the upper bound $\varepsilon > 0$ on the measurement error:

$$|\Delta y_i| < \varepsilon.$$

- Then, the only information that we have about y_i is that y_i belongs to the interval $(\widetilde{y}_i - \varepsilon, \widetilde{y}_i + \varepsilon)$.
- We thus need to locate the local maxima and local minima of a function under such interval uncertainty.

4. Need for Guaranteed Results

- Due to measurement uncertainty, the actual observed values fluctuate.
- Hence, the f-n corresponding to the actual measurement results usually has many local maxima (minima).
- Most of these local maxima and minima:
  - are caused by the measurement errors and
  - do not have any physical significance.
- We only want to keep those local maxima and minima which reflect the actual dependence,
- In other words, we want to keep only those local extrema that guaranteed to correspond to:
  - source components,
  - chemical substances,
  - etc.

5. Case of Fuzzy Uncertainty

- Often:
  - in addition (or instead) the guaranteed bound ε for the measurement error Δy_i,
  - an expert can provide bounds that contain Δy_i with a certain degree of confidence.
- Usually, we know several such bounding intervals corresponding to different degrees of confidence.
- Such a nested family of intervals is equivalent to a *fuzzy set* (to be more precise, to its α-cuts).
- From the algorithmic viewpoint, fuzzy uncertainty can be thus reduced to interval uncertainty.
- Because of this reduction, we will be concentrating on the algorithms for solving the interval problem.

6. Locating Local Extrema Under Interval Uncertainty: What Is Known

- A feasible (polynomial-time) algorithm is known for $m = 1$, when there is only one input variable t_1.
- In many practical applications, we need to solve a similar problem in a situation when we have several inputs

$$t_1, \ldots, t_m, \quad m > 1.$$

- For example, in locating components of a radioastronomical source, we start with a 2-D intensity function.
- In clustering, we also need to consider local maxima of functions of several variables, etc.
- In this talk, we describe a polynomial-time algorithm that solves the problem for case of several variables.

7. Definitions: Locations and Connectedness

- Let G be a finite undirected graph; its vertices will be called *locations*.
- If the vertices $x, y \in G$ are connected by an edge, we will call them *neighbors* and denote it by $x \sim y$.
- We say that a function $f : G \to \mathbb{R}$ has a *local minimum* at location x if $f(x) \leq f(y)$ for all neighbors y of x.
- We say that a function $f : G \to \mathbb{R}$ has a *local maximum* at location x if $f(x) \leq f(y)$ for all neighbors y of x.
- Let $S \subseteq G$. We say that $x, y \in S$ are *S-connected* if there exists a *S-connecting sequence*

$$x_0 = x \in S, x_1 \in S, \ldots, x_{m-1} \in S, x_m = y \in S \text{ s.t. } \forall i\,(x_i \sim x_{i+1}).$$

- We say that a subset $S \subseteq G$ is *connected* if every two locations $x, y \in S$ are S-connected.

8. Definitions: Measurement Results

- Let G be a graph. By a *measurement result*, we mean a pair $\mathbf{f} = \langle f_0, \varepsilon \rangle$, where:
  - $f_0 : G \to \mathbb{R}$ is a rational-valued function whose values $f_0(x)$ are called *measured values*;
  - $\varepsilon > 0$ is a rational number called *measurement accuracy*.
- A measurement result will also be called an *interval-valued function* and denoted by

$$\mathbf{f}(x) = (f_0(x) - \varepsilon, f_0(x) + \varepsilon).$$

- We say that a function $f : G \to \mathbb{R}$ is *consistent* with $\mathbf{f}(x) = (f_0(x) - \varepsilon, f_0(x) + \varepsilon)$ if $f(x) \in (f_0(x) - \varepsilon, f_0(x) + \varepsilon)$ for every location x.
- We will denote this consistency by $f \in \mathbf{f}$.

9. Definitions: Locating Local Minimum

- Let G be a graph, and let $\mathbf{f}$ be an interval-valued function on this graph.
- We say that a connected set S is a *local minimum set* of $\mathbf{f}$ if the following properties are satisfied:
  - every function $f \in \mathbf{f}$ attains a local minimum at some location $x \in S$;
  - each location $x_m \in S$ at which $f \in \mathbf{f}$ attains its smallest value on S is a local minimum of f on G;
  - for $S' \subset S$, $S' \neq S$, there is a function $f \in \mathbf{f}$ that does not have any local minimum on S'.
- For $S = \{x_0\}$, for every $f \in \mathbf{f}$, the value $f(x_0)$ is smaller than or equal to the value at all neighbors $y \sim x_0$.
- When $S = \{x_1, x_2, \ldots\}$, different $f \in \mathbf{f}$ may attain local minimum at different locations $x_i \in S$.

10. Definitions: Locating Local Maximum

- Let G be a graph, and let $\mathbf{f}$ be an interval-valued function on this graph.
- We say that a connected set S is a *local maximum set* of $\mathbf{f}$ if the following properties are satisfied:
  - every function $f \in \mathbf{f}$ attains a local maximum at some location $x \in S$;
  - each location $x_m \in S$ at which $f \in \mathbf{f}$ attains its largest value on S is a local maximum of f on G;
  - for $S' \subset S$, $S' \neq S$, there is a function $f \in \mathbf{f}$ that does not have any local maximum on S'.
- For $S = \{x_0\}$, for every $f \in \mathbf{f}$, the value $f(x_0)$ is larger than or equal to the value at all neighbors $y \sim x_0$.
- When $S = \{x_1, x_2, \ldots\}$, different $f \in \mathbf{f}$ may attain local maximum at different locations $x_i \in S$.

11. First Result

- *There exists a polynomial-time algorithm that:*
  - *given an interval-valued function* $\mathbf{f}$ *on a graph* G,
  - *returns all its local minimum sets.*
- **Algorithm:**
  - By trying all locations $x \in G$, we can find all local minima x_ℓ of the function $f_0(x)$.
  - For each such local minimum x_ℓ, we again try all locations $x \in G$ and find the set
$$S_\ell = \{x : f_0(x) < f_0(x_\ell) + 2\varepsilon\}.$$
  - From this set, we select the subset S'_ℓ consisting of all locations $x \in S_\ell$ which are S_ℓ-connected to x_ℓ.
  - If $\forall x \in S'_\ell\,(f_0(x) \ge f_0(x_\ell))$, then S'_ℓ is returned as one of the desired local minimum sets S.

12. Second Result

- *There exists a polynomial-time algorithm that:*
  - *given an interval-valued function* **f** *on a graph* G,
  - *returns all its local maximum sets.*
- **Algorithm:**
  - By trying all locations $x \in G$, we can find all local maxima x_ℓ of the function $f_0(x)$.
  - For each such local maximum x_ℓ, we again try all locations $x \in G$ and find the set

$$S_\ell = \{x : f_0(x) > f_0(x_\ell) - 2\varepsilon\}.$$

  - From this set, we select the subset S'_ℓ consisting of all locations $x \in S_\ell$ which are S_ℓ-connected to x_ℓ.
  - If $\forall x \in S'_\ell\,(f_0(x) \le f_0(x_\ell))$, then S'_ℓ is returned as one of the desired local maximum sets S.

13. Proof: Reducing Maxima to Minima

- One can easily check that:
  - local maxima sets of an interval function $\langle f_0, \varepsilon\rangle = (f_0(x) - \varepsilon, f_0(x) + \varepsilon)$
  - are exactly local minimum sets of the interval function

$$\langle -f_0, \varepsilon\rangle = (-f_0(x) - \varepsilon, -f_0(x) + \varepsilon).$$

- Because of this reduction, it is sufficient to prove the result about the local minimum sets.
- For this case, we need to prove:
  - that the algorithm is indeed polynomial-time,
  - that every set generated by this algorithm is indeed a local minimum set, and
  - that every local minimum set appears in the list of sets generated by our algorithm.

14. Proof that the Algorithm is Polynomial-Time

- Stage 1 – finding local minima – means comparing each $f(x)$ with all neighboring values; time $T_1 \le n \cdot n = n^2$.
- Stage 2: for each of $\le n$ local min, we form $S_\ell = \{x : f_0(x) > f_0(x_\ell) - 2\varepsilon\}$ in time $\le n$; so $T_2 \le n \cdot n = n^2$.
- Stage 3: connectedness is a transitive closure of $\sim$, so we can use $O(n^3)$ iterative *wave algorithm*:
  - initially, we mark x_ℓ;
  - mark all unmarked neighbors of marked locations.
- Thus, total time T_3 of Stage 3 is $T_3 \le n \cdot n^3 = O(n^4)$.
- Stage 4: checking $\forall x \in S'_\ell\,(f_0(x) \ge f_0(x_\ell))$ is straightforward: $O(n)$ time for each of $\le n$ local minima x_ℓ.
- Total time: $O(n^2) + O(n^2) + O(n^4) + O(n^2) = O(n^4)$, i.e., polynomial time.

15. Acknowledgment

This work was supported in part:

- by the National Science Foundation grants HRD-0734825 (Cyber-ShARE Center of Excellence) and DUE-0926721,
- by Grant 1 T36 GM078000-01 from the National Institutes of Health, and
- by a grant from New Mexico State University.