Towards Efficient Prediction of Decisions under Interval Uncertainty

Van Nam Huynh[1], Vladik Kreinovich[2]
Yoshiteru Nakamori[1], and Hung T. Nguyen[3]

[1]Japan Advanced Institute of
Science and Technology (JAIST)
Tatsunokuchi, Ishikawa 923-1292, Japan

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

[3]New Mexico State University
Las Cruces, NM 88003, USA

1. Outline

- *Decision making:* in many practical situations, users select between n alternatives $a_1, \ldots, a_n$.
- *Situation:* often, we only know bounds $\underline{v}_i$, $\overline{v}_i$ on the actual (unknown) utility v_i: $\underline{v}_i \le v_i \le \overline{v}_i$.
- *Reasonable assumption:* v_i are independent and uniformly distributed on $[\underline{v}_i, \overline{v}_i]$.
- *Objective:* to estimate, for each i, the probability p_i that the alternative a_i will be selected.
- *What we do:* we provide efficient algorithms for computing these probabilities.

2. Making a Decision when We Know the Exact Values of the Maximized Quantity

- *General situation:* select an alternative with the largest possible value of a certain quantity v.
- *Case of exact values:* for n alternatives $a_1, \ldots, a_n$, we know the exact values $v_1, \ldots, v_n$ of v.
- *Solution:* select a_i for which $v_i \to \max$.
- *How to predict this decision:* compute the index i_n of the largest value v_i.
- *Algorithm:* process v_i one by one and keep track of the index of largest-so-far value.
- *Computation time:* $O(n)$.

3. Decisions under Interval Uncertainty: Formulation of the Problem

- *Typical situation:* we only know the bounds $\underline{v}_i$ and $\overline{v}_i$ for the (unknown) actual value v_i.
- *Interval reformulation:* our only information about v_i is that v_i belongs to the *interval* $[\underline{v}_i, \overline{v}_i]$.
- *Difficulty:* when intervals $[\underline{v}_i, \overline{v}_i]$ and $[\underline{v}_j, \overline{v}_j]$ intersect, it is possible that $v_i > v_j$ and it is possible that $v_j > v_i$.
- *Conclusion:* some decision makers will prefer v_i, some may prefer v_j.
- *Exact predictions* are thus impossible.
- *Reasonable idea:* predict the probability p_i of selecting v_i.

4. Decision Making under Interval Uncertainty: Main Idea and Related Computational Problem

- *Idea:* we assume that for all $i \neq j$:
  - v_i is uniformly distributed on $[\underline{v}_i, \overline{v}_i]$, and
  - v_i and v_j are independent random variables.
- *Consistent* with MaxEnt: $-\int \rho(v)\cdot\log(\rho(v))\,dv \to \max$
- *How to compute p_i:* p_i is the probability that, under this distribution, v_i is the largest of n values $v_1, \ldots, v_n$.
- *Case of $n = 2$:* there are explicit formulas.
- *Problem:* how to compute for larger n?
- *First idea:* $p_i = V_i/V$, where $V = (\overline{v}_1 - \underline{v}_1)\cdot\ldots\cdot(\overline{v}_n - \underline{v}_n)$, and V_i is the volume of the set $\{v : v_i = \max v_j\}$.
- *How to compute volumes:* use a grid of linear size ε.
- *Problem:* huge computation time $(1/\varepsilon)^n$ even for the medium-size values $n \approx 100$.

5. Monte-Carlo Simulations as a Way to Approximate the Desired Decision Probabilities

- *Idea:*
  - select a number N;
  - then, N times, simulate each v_i as a uniformly distributed random variable;
  - estimate p_i as N_i/N, where N_i is the number of simulations in which v_i was the largest value.
- *Accuracy* of this method is $\varepsilon \approx 1/\sqrt{N}$.
- *Conclusion:* to achieve desired accuracy ε, we need $N \approx \dfrac{1}{\varepsilon^2}$ simulations.
- *Example:* to achieve a 10% accuracy, we need $N \approx 100$ simulations.
- *Limitation:* for high accuracy, we need a large number of simulations.

6. Efficient $O(n^2)$ Time Algorithm for Exact Computation of Decision Probabilities: Main Idea

- *Two steps:*
  - describe the conditional probability

$$p_1(v_1) \stackrel{\text{def}}{=} \text{Prob}(v_1 \text{ is the largest} : v_1 \text{ is actual});$$

  - use the complete probability formula to compute p_1:

$$p_1 \stackrel{\text{def}}{=} \text{Prob}(v_1 \text{ is the largest}) = \int p_1(v_1) \cdot \rho_1(v_1)\, dv_1.$$

- *Observation:* since v_1 is uniform on $[\underline{v}_1, \overline{v}_1]$, we have

$$p_1 = \frac{1}{\overline{v}_1 - \underline{v}_1} \cdot \int p_1(v_1)\, dv_1.$$

- *Remaining problem:* find the expression for $p_1(v_1)$.

7. Deriving an Expression for $p_1(v_1)$

- *Find:* the probability $p_1(v_1)$ that given v_1 is the largest.
- *Fact:* the fact that v_1 is the largest means that $v_2 \le v_1$, $v_3 \le v_1$, etc.
- *We assumed:* that all the variables v_i are independent,
- *Conclusion:* $p_1(v_1)$ is the product of the probability that $v_2 \le v_1$, the probability that $v_3 \le v_1$, etc.
- *Fact:* for each i, the probability that $v_i < v_1$ is:
  - if $\overline{v}_i \le v_1$, then $v_i \le v_1$ with probability 1;
  - if $v_1 < \underline{v}_i$, then $v_i \le v_1$ with probability 0;
  - else, since v_i is uniform, $p_i = \dfrac{v_1 - \underline{v}_i}{\overline{v}_i - \underline{v}_i}$.
- *Result:* if $v_1 \ge \underline{v}_i$ for all i, we have $p_1(v_1) = \prod\limits_{i: v_1 \le \overline{v}_i} \dfrac{v_1 - \underline{v}_i}{\overline{v}_i - \underline{v}_i}$.
- *Otherwise:* $p_1(v_1) = 0$.

8. Towards an $O(n^2)$ Time Algorithm

- *Reminder:* $p_1(v_1) = \prod\limits_{i: v_1 \le \overline{v}_i} \dfrac{v_1 - \underline{v}_i}{\overline{v}_i - \underline{v}_i}$.
- *Fact:* $p_1(v_1)$ depends on whether $v_1 > \overline{v}_i$ or $v_1 \le \underline{v}_i$.
- *Idea:* sort the endpoints $\underline{v}_i$ and $\overline{v}_i$ into an increasing sequence $v_{(1)} \le v_{(2)} \le \ldots \le v_{(2n)}$.
- *Conclusion:* in each of the zones $z_j = [v_{(j)}, v_{(j+1)})$, we get $p_1(v_1) = a_0 + a_1 \cdot v_1 + \ldots + a_k \cdot v_1^k$.
- *Fact:* $\int p_1(v_1)\,dv_1 = a_0 \cdot v_1 + \dfrac{a_2}{2} \cdot v_1^2 + \ldots + \dfrac{a_k}{k+1} \cdot v_1^{k+1}$.
- *Computation time:* for each of $2n+1$ zones, we need:
  - n products to compute $p_1(v_1)$,
  - n steps to integrate over this zone, and
  - one addition to add to the previous integral.
- *Resulting time:* $(2n+1) \cdot O(n) = O(n^2)$.

9. Resulting Algorithm

- First, sort endpoints $\underline{v}_i$ and $\overline{v}_i$: $v_{(1)} \le v_{(2)} \le \ldots \le v_{(2n)}$.
- This divides the real line into $2n+1$ zones $z_0 = (-\infty, v_{(1)})$, $z_1 = [v_{(1)}, v_{(2)}), \ldots, z_j = [v_{(j)}, v_{(j+1)}), \ldots, z_{2n} = [v_{(2n)}, \infty)$.
- For the zones z_j for which $v_{(j)} < \underline{v}_1$, $v_{(j+1)} > \overline{v}_1$, or $v_{(j+1)} < \underline{v}_i$ for some i, the integral p_{1j} is equal to 0.
- For every other zone, find $p_1(v_1) = \prod\limits_{i: v_{(j+1)} \le \overline{v}_i} \dfrac{v_1 - \underline{v}_i}{\overline{v}_i - \underline{v}_i}$ as a polynomial of v_1.
- Compute an expression for $P_{1j}(v_1) = \int p_{1j}(v_1)\, dv_1$.
- The desired integral p_{1j} can then be computed as the difference $P_{1j}(v_{(j+1)}) - P_{1j}(v_{(j)})$.
- Finally, compute $p_1 = \dfrac{1}{\overline{v}_1 - \underline{v}_1} \cdot \sum\limits_{j=0}^{2n} p_{1j}$.

10. Discussion

- *Fact:* the above algorithm is based on the following three assumptions:
  - that we have a finite set of alternatives,
  - that decision makers know the exact values of v_i, and
  - that the distributions are uniform.
- *Natural question:* what will happen if we do not make these assumptions?

11. First Observation: What If We Have Infinitely Many Alternatives

- *Motivation:* an alternative is often characterized by a continuous real-valued parameter(s) a on a range $[\underline{a}, \overline{a}]$.
- *Description:* for every a, we have an interval $[\underline{v}(a), \overline{v}(a)]$.
- *Assumption:* $v(a)$ are independent and uniform.
- *New assumption:* both $\underline{v}(a)$ and $\overline{v}(a)$ continuously depend on a.
- *Additional difficulty:* maximum may not be attained.
- *Solution:* look for ε-optima: $v(a) \geq \max\limits_{b} v(b) - \varepsilon$.
- *Theorem:* an ε-optimal alternative corresponding to the random $v(a)$ is ε-optimal for $\overline{v}(a)$.
- *Comment:* $\overline{v}(a)$ is the "optimistic" function.
- *Discussion:* to avoid this conclusion, we must relax assumptions about independence and uniformity.

12. Second Observation: What If Decision Makers Know the Values with Interval Uncertainty

- *Previous assumption:* decision makers (DMs) know the exact values v_i.
- *More realistic situation* (Sevastjanov): decision makers know v_i with some accuracy $\delta > 0$: $|\widetilde{v}_i - v_i| \le \delta$;
  - if $\widetilde{v}_i - \delta > \widetilde{v}_j + \delta$, then $v_i > v_j$ so DM selects i;
  - if $\widetilde{v}_j - \delta > \widetilde{v}_i + \delta$, then $v_j > v_i$ so DM selects j;
  - else, could be $v_i > v_j$ or $v_j > v_i$; DM selects i or j.
- *We know:* the intervals $[\underline{v}_i, \overline{v}_i]$.
- *Find:* the probability p_i^- that a DM will *necessarily* select i.
- *Find:* the probability p_i^+ that a DM *may* select i.
- *Our result:* we modify the above algorithm so that it computes both p_i^- and p_i^+.

13. Third Observation: What If the Distributions are not Uniform (Case $n = 2$)

- Only *uniform* ρ is shift-invariant $\rho(v_1 + a_1, v_2 + a_2) = \rho(v_1, v_2)$ & conditionally scale-invariant

$$\rho(\lambda_1 \cdot v_1, \lambda_2 \cdot v_2) = a(\lambda_1, \lambda_2) \cdot \rho(v_1, v_2).$$

- *Discussion:* both v_i are same quantity, so we should have same shift and scaling.
- *Find:* all symmetric functions $\rho(v_1, v_2) = \rho(v_2, v_1)$ for which $\rho(v_1 + a, v_2 + a) = \rho(v_1, v_2)$ and

$$\rho(\lambda \cdot v_1, \lambda \cdot v_2) = a(\lambda) \cdot \rho(v_1, v_2).$$

- *Solution:* $\rho(v_1, v_2) = \varepsilon \cdot \delta(v_1 - v_2) + A \cdot |v_1 - v_2|^{-\alpha}$.
- *Discussion:* when $\varepsilon \neq 0$, $\text{Prob}(v_1 = v_2) > 0$.
- *Computations:* we can use Monte-Carlo simulations.
- *Remaining open problem:* $n > 2$.

14. Acknowledgments

This work was supported in part by:

- by National Science Foundation grants HRD-0734825, EAR-0225670, and EIA-0080940,
- by Texas Department of Transportation grant No. 0-5453, and
- by the Japan Advanced Institute of Science and Technology (JAIST) International Joint Research Grant 2006–08.