8-2007

Towards Efficient Prediction of Decisions under Interval Uncertainty

Van Nam Huynh

Vladik Kreinovich
The University of Texas at El Paso, vladik@utep.edu

Yoshiteru Nakamori

Hung T. Nguyen

Comments:
Technical Report: UTEP-CS-07-31a
Published in: Roman Wyrzykowski, Jack Gondarra, Konrad Karczewski, and Jerzy Wasniewski (eds.), *Parallel Processing and Applied Mathematics*, Proceedings of the Seventh International Conference on Parallel Processing and Applied Mathematics PPAM'2007, Gdansk, Poland, September 9-12, 2007, Springer Lecture Notes in Computer Science, 2008, Vol. 4967, pp. 1372-1381.

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, huynh@jaist.ac.jp
[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,
hunguyen@nsmu.edu

Abstract. In many practical situations, users select between n alternatives $a_1, \ldots, a_n$, and the only information that we have about the utilities v_i of these alternatives are bounds $\underline{v}_i \le v_i \le \overline{v}_i$. In such situations, it is reasonable to assume that the values v_i are independent and uniformly distributed on the corresponding intervals $[\underline{v}_i, \overline{v}_i]$. Under this assumption, we would like to estimate, for each i, the probability p_i that the alternative a_i will be selected. In this paper, we provide efficient algorithms for computing these probabilities.

1 Decisions under Interval Uncertainty: Formulation of the Problem

Making a decision when we know the exact values of the maximized quantity. Let us assume that we want to select an alternative with the largest possible value of a certain quantity. If for n alternatives $a_1, \ldots, a_n$, we know the exact values $v_1, \ldots, v_n$ of the corresponding quantity, then the decision maker will select the alternative a_i for which the corresponding value v_i is the largest.

How to predict this decision. When we know the values $v_1, \ldots, v_n$, then predicting a decision means computing the index i_n of the largest value v_i.

This can be done in time $O(n)$, by the following iterative process. At each iteration k ($k = 1, \ldots, n$), i_k will be index of the largest of the first k values $v_1, \ldots, v_k$. In the first iteration $k = 1$, we naturally take $i_1 = 1$. Once we got i_k, on the next $(k+1)$-st iteration, we compare the largest-so-far value v_{i_k} with the new value v_{k+1}. If $v_{k+1} > v_{i_k}$, then we take $i_{k+1} = k + 1$ as the new index, otherwise we take keep the old index, i.e., take $i_{k+1} = i_k$.

Predicting decisions under interval uncertainty: a problem. In many practical situations, we do not know the exact values of the desired quantity. In many such situations, we only know the bounds $\underline{v}_i$ and $\overline{v}_i$ for the (unknown) actual

value v_i, i.e., our only information about v_i is that v_i belongs to the *interval* $[\underline{v}_i, \overline{v}_i]$.

If we only know the intervals $[\underline{v}_i, \overline{v}_i]$ of possible values of v_i, and these intervals share several common points, then it may be that, e.g., v_1 is the largest and it may be that v_2 is the largest. Thus, some decision makers will prefer v_1, some may prefer v_2, etc. In this case, we cannot exactly predict which selection will be made – but we can hopefully predict the probability p_i of selecting v_i.

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

Idea. A natural idea for computing the probability p_i is as follows. For each i, we assume that the (unknown) actual value v_i is uniformly distributed in the corresponding interval $[\underline{v}_i, \overline{v}_i]$, and that different values v_i are independent random variables. Then, the desired probability p_i is the probability that, under this distribution, v_i is the largest of n values $v_1, \ldots, v_n$.

Comment. The above assumptions about the probability distributions correspond, e.g., to the Maximum Entropy (MaxEnt) approach (see, e.g., [4]), in which among all possible distributions $\rho(v_1, \ldots, v_n)$ on the given box $[\underline{v}_1, \overline{v}_1] \times \ldots \times [\underline{v}_n, \overline{v}_n]$, we select the one with the largest value of the entropy

$$-\int \rho(v_1, \ldots, v_n) \cdot \log(\rho(v_1, \ldots, v_n))\, dv_1 \ldots dv_n.$$

This MaxEnt distribution is uniform on the box, which is equivalent to assuming that all values v_i are independent and uniformly distributed.

For $n = 2$, there are explicit formulas for computing p_i. For the case $n = 2$ of two alternatives, p_1 is the probability that $v_1 > v_2$. There exist explicit formulas for this probability; see, e.g., [3, 6, 8–12]. So, for $n = 2$, we have an efficient algorithm for computing the desired probabilities p_1 and p_2.

Problem: how to compute p_i for large n? The case of $n = 2$ is a toy example. In most practical decision problems, we have a large number of alternatives – sometimes so large that we need high performance parallel computers to handle these problems. How can we then compute the corresponding probabilities p_i?

Since the distribution is uniform, the desired probability p_i is equal to the ratio V_i/V, where $V = (\overline{v}_1 - \underline{v}_1) \cdot \ldots \cdot (\overline{v}_n - \underline{v}_n)$ is the (n-dimensional) volume of the box, and V_i is the volume of the part of which box for which v_i is larger than the values of all other values v_j.

In principle, we can compute the volume V_i by computing the corresponding n-dimensional integral. However, computing n-dimensional integrals with a given accuracy $\varepsilon > 0$ means that we have to consider a grid of size $\sim \varepsilon$ along each axis – i.e., consider $\sim \dfrac{1}{\varepsilon}$ points along each axis and $\sim \dfrac{1}{\varepsilon^n}$ points overall.

For large n, this computation time is too high to be practically useful. It is therefore desirable to come up with more efficient algorithms for computing p_i.

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

Idea. A natural idea is to use Monte-Carlo simulations; see, e.g., [7]. Specifically, we select a number N, and then N times, we simulate each v_i as a uniformly distributed random variable. After that, we take N_i/N as an estimate for p_i, where N_i is the number of simulations in which v_i was the largest value.

It is known that the accuracy of the Monte-Carlo simulation is $1/\sqrt{N}$. So, to get 10% accuracy in computing p_i, it is sufficient to take $N \approx 100$ simulations.

Limitations. The main limitation of this approach is that if we want accurate estimates, with accuracy $\varepsilon \ll 1$, we need a large number of simulations $N \approx \dfrac{1}{\varepsilon^2}$. This number is not impossible (as for direct integration) but still large. It is therefore desirable to design an algorithm for computing p_i exactly.

4 Efficient Algorithm for Exact Computation of Decision Probabilities

Let us describe an efficient ($O(n^2)$) algorithm for computing p_i. Without losing generality, we can assume that $i = 1$, i.e., that we need to compute the probability p_1 that v_1 is the largest of n values v_i. The outline of this section is as follows:

- First, we will describe the main idea behind this algorithm.
- Then, we will show how this idea translates into an actual $O(n^2)$ algorithm.
- Finally, we will explicitly describe the resulting algorithm.

Main idea. Our idea is to first describe, for each given v_1, the conditional probability $p_1(v_1)$ that this v_1 is the largest – under the condition that v_1 is the actual value. Then, due to the Bayes formula, the overall probability p_1 that v_1 is the largest can be obtained by integrating this conditional probability $p_1(v_1)$ times the probability density of v_1:

$$\text{Prob}(v_1 \text{ is the largest}) = \int \text{Prob}(v_1 \text{ is the largest} \,|\, v_1 \text{ is actual}) \cdot \rho_1(v_1)\, dv_1.$$

The distribution of v_1 is uniform on the interval $[\underline{v}_1, \overline{v}_1]$, hence

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

How can we describe the expression for $p_1(v_1)$? Once v_1 is fixed, the fact that v_1 is the largest means that $v_2 \le v_1$, $v_3 \le v_1$, etc. Since all the variables v_i are independent, this probability is equal to the product of $n-1$ probabilities: the probability that $v_2 \le v_1$, the probability that $v_3 \le v_1$, etc.

For each i, the probability that $v_i < v_1$ can be determined as follows:

- If $\overline{v}_i \le v_1$, then $v_i \le v_1$ with probability 1. This probability does not change the product and can thus simply be omitted.
- If $v_1 < \underline{v}_i$, this means that $v_i \le v_1$ cannot happen at all. The resulting probability is 0, so such terms can be completely ignored.
- Finally, if $\underline{v}_i \le v_1 < \overline{v}_i$, then, since the distribution of v_i is uniform on the interval $[\underline{v}_i, \overline{v}_i]$, the probability that $v_i \le v_1$ is equal to $\dfrac{v_1 - \underline{v}_i}{\overline{v}_i - \underline{v}_i}$.

Thus, the conditional probability $p_1(v_1)$ is equal to

$$p_1(v_1) = \prod_{i: v_1 \le \overline{v}_i} \frac{v_1 - \underline{v}_i}{\overline{v}_i - \underline{v}_i},$$

if $v_1 \ge \underline{v}_i$ for all i, and to 0 otherwise.

Transforming this idea into the actual algorithm. As we see, the expression for $p_1(v_1)$ depends on the relation between v_1 and the endpoints $\underline{v}_i$ and $\overline{v}_i$ of the intervals $[\underline{v}_i, \overline{v}_i]$. So, if we sort these endpoints into an increasing sequence $v_{(1)} \le v_{(2)} \le \ldots \le v_{(2n)}$, then, in each of the resulting $2n+1$ zones $z_0 = (-\infty, v_{(1)})$, $z_1 = [v_{(1)}, v_{(2)})$, $\ldots$, $z_j = [v_{(j)}, v_{(j+1)})$, $\ldots$, $z_{2n} = [v_{(2n)}, \infty)$, we will have the same analytical expression for $p_1(v_1)$.

For each zone, the corresponding expression is a product of $\le n$ linear terms. Multiplying these terms one by one, we get a polynomial of degree $\le n$ in $\le n$ computational steps.

The integral $\int p_1(v_1)\, dv_1$ can be computed as the sum of integrals p_{1j} over all the zones z_j, $j = 0, \ldots, 2n$. An integral of a polynomial $a_0 + a_1 \cdot v_1 + \ldots + a_k \cdot v_1^k$ is equal to $a_0 \cdot v_1 + \dfrac{a_2}{2} \cdot v_1^2 + \ldots + \dfrac{a_k}{k+1} \cdot v_1^{k+1}$, i.e., it can be also computed coefficient-by-coefficient in linear time. Since we have $2n$ zones, we thus need $(2n+1) \cdot O(n) = O(n^2)$ time to compute all $2n+1$ sub-integrals, and then $2n = O(n)$ operations to add them and get $\int p_1(v_1)\, dv_1$. Dividing this integral by $\overline{v}_1 - \underline{v}_1$, we get p_1. Thus, overall, we indeed need quadratic time.

Resulting algorithm. At the first step of this algorithm, we order all $2n$ endpoints $\underline{v}_i$ and $\overline{v}_i$ into an increasing sequence $v_{(1)} \le v_{(2)} \le \ldots \le v_{(2n)}$. As a result, we divide 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, we form the expression

$$p_1(v_1) = \prod_{i: v_{(j+1)} \le \overline{v}_i} \frac{v_1 - \underline{v}_i}{\overline{v}_i - \underline{v}_i}.$$

This expression is a product of $\le n$ linear functions of the unknown v_1. By multiplying by these functions one by one, we get an explicit expression for a polynomial in v_1. By processing the coefficients of this polynomial one by one,

we can provide the explicit analytical expression for the (indefinite) integral $P_{1j}(v_1)$ of this polynomial. The desired integral p_{1j} can then be computed as the difference $P_{1j}(v_{(j+1)}) - P_{1j}(v_{(j)})$.

Finally, the desired probability p_1 is computed as

$$p_1 = \frac{1}{\overline{v}_1 - \underline{v}_1} \cdot \sum_{j=0}^{2n} p_{1j}.$$

Comments.

- The idea of dividing the real line into zones corresponding to sorted endpoints of the given intervals comes from another situation where we need to combine probabilities and intervals: namely, from the algorithms for algorithms for computing population variance under interval uncertainty [2].
- The above algorithm is based on the 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. In the following sections, we consider discuss what will happen if we do not make these assumptions.

5 First Observation: What If We Have Infinitely Many Alternatives

Formulation of the problem. In many practical problem, we have infinitely many alternatives. For example, an alternative is often characterized by a continuous real-valued parameter a on a range $[\underline{a}, \overline{a}]$ (or by several such parameters). In such situations, for every a, we have an interval $[\underline{v}(a), \overline{v}(a)]$ of possible values of $v(a)$. For example, we may know the approximate values $\widetilde{v}(a)$, and we know the bound $\Delta(a) > 0$ on the approximation error; in this case, the (unknown) actual value $v(a)$ belongs to the interval $[\widetilde{v}(a) - \Delta(a), \widetilde{v}(a) + \Delta(a)]$.

It is usually reasonable to assume that both $\underline{v}(a)$ and $\overline{v}(a)$ are continuous functions of a. Again, we assume that the values $v(a)$ corresponding to different a are independent random variables uniformly distributed on the corresponding intervals $[\underline{v}(a), \overline{v}(a)]$. If a decision makers selects the action with the largest possible value of a, what is the probability of selecting different values of a?

A minor complication here is that since there are infinitely many possible alternatives a, the maximum may not necessarily be attained. In this case, it is reasonable to fix some small value ε and select an alternative $a(\varepsilon)$ for which $v(a(\varepsilon)) \geq \max\limits_{a} v(a) - \varepsilon$. We will call such alternative *ε-optimal*.

A somewhat unexpected solution. Our result is that for every $\varepsilon > 0$, an ε-optimal alternative corresponding to the random values $v(a)$ is ε-optimal for the function $\overline{v}(a)$.

In other words, with probability 1, the decision maker will select the solution that maximizes the "optimistic" value $\overline{v}(a)$.

Proof. Before we start discussing this result, let us first prove it. It is sufficient to prove that $\max\limits_a v(a) = \max\limits_a \overline{v}(a)$. Indeed, from the fact that $v(a) \le \overline{v}(a)$, we conclude that $\max\limits_a v(a) \le \max\limits_a \overline{v}(a)$. Let us now pick any number $\varepsilon' > 0$ and show that $\max\limits_a v(a) \ge \max\limits_a \overline{v}(a) - \varepsilon'$; then in the limit $\varepsilon' \to 0$ we will get $\max\limits_a v(a) \le \max\limits_a \overline{v}(a)$ and hence, $\max\limits_a v(a) = \max\limits_a \overline{v}(a)$.

Indeed, let a_m be a value at which the continuous function $\overline{v}(a)$ attains its maximum. Since $\overline{v}(a)$ is continuous, there exists a value δ such that $|a_m - a'| \le \delta$ implies that $|\overline{v}(a') - \overline{v}(a_m)| \le \varepsilon'/2$, i.e., that $\overline{v}(a') \ge \overline{v}(a_m) - \varepsilon'/2 = \max\limits_a \overline{v}(a) - \varepsilon'/2$. Let us prove that we cannot have $\max\limits_a v(a) < \max\limits_a \overline{v}(a) - \varepsilon'$. Indeed, that would imply that $v(a') < \overline{v}(a_m) - \varepsilon'$ for all (infinitely many) values a' for which $|a' - a_m| \le \delta$. This means that for all such a', we have $v(a') \notin [\overline{v}(a) - \varepsilon'/2, \overline{v}(a)]$ – because for values from that subinterval, we have $v(a) \ge \overline{v}(a) - \varepsilon'/2 \ge (\overline{v}(a_m) - \varepsilon'/2) - \varepsilon'/2 = \overline{v}(a_m) - \varepsilon'$. The probability of being not in this interval is proportional to $1 - (\varepsilon'/2)(\overline{v}(a) - \underline{v}(a))$ and is hence $\le 1 - (\varepsilon'/2)/W$, where $W \stackrel{\text{def}}{=} \max\limits_a (\overline{v}(a) - \underline{v}(a))$. There are infinitely many such values a', and all variables $v(a')$ are independent; thus, the probability that $v(a') < \overline{v}(a_m) - \varepsilon'$ for all a' does not exceed $(1 - (\varepsilon'/2)/W)^n$ for every n. When $n \to \infty$, we conclude that this probability is 0. Thus, with probability 1, we have some value a' for which $v(a') \ge \overline{v}(a_m) - \varepsilon'$. The statement is proven.

Discussion. The above counter-intuitive result follows from the assumption that the values v_i are independent and uniformly distributed. So, to avoid this conclusion, we must relax this assumption; in the last section of this paper, we will start analyzing what will happen if relax this assumption.

6 Second Observation: What If Decision Makers Also Only Know the Values of the Desired Quantity with Interval Uncertainty

Formulation of the problem. In the previous text, we assumed that the decision makers know the exact values v_i of the desired quantity, and make their decisions based on these exact values. Based on this assumption, we considered the situation when we only know the intervals $[\underline{v}_i, \overline{v}_i]$ for v_i, and we estimated the probability p_i that for randomly selected values $v_1 \in [\underline{v}_1, \overline{v}_1]$, ..., $v_n \in [\underline{v}_n, \overline{v}_n]$, a decision maker will select the alternative v_i.

In practice, decision makers may also know the values v_i only approximately. How does this approximate character affect the decisions?

Previous work. For the case of $n = 2$ alternatives, the case when decision makers know v_i with accuracy $\delta > 0$ was considered in [11]; a case of general interval bounds was analyzed in [8–10].

What we plan to do. In this section, we consider the simplest case of accuracy δ, and we show how to modify the above algorithms to account for this uncertainty.

What happens when decision makers only know the values v_i with accuracy δ: our assumption. When the decision maker knows the exact values of v_1 and v_2, then the decision is straightforward:

- if $v_1 = v_2$, then both alternative are equally attractable, so any of them can be selected;
- if $v_1 > v_2$, then the first alternative a_1 is better, so it will be selected;
- if $v_1 < v_2$, then the second alternative is better, so a_2 will be selected.

If we only know the approximate values v_1 and v_2, values which are only correct within an accuracy δ, then we also have three options:

- It is possible that $v_1 - \delta > v_2 + \delta$ (i.e., equivalently, $v_1 - v_2 > \varepsilon$, where $\varepsilon \stackrel{\text{def}}{=} 2\delta$). In this case, every value from the interval $[v_1 - \delta, v_1 + \delta]$ is larger than every value from the interval $[v_2 - \delta, v_2 + \delta]$. Thus, we are sure that the alternative a_1 is larger, and we select it.
- It is also possible that $v_1 + \delta < v_2 - \delta$ (i.e., equivalently, $v_1 - v_2 < -\varepsilon$). In this case, every value from the interval $[v_1 - \delta, v_1 + \delta]$ is smaller than every value from the interval $[v_2 - \delta, v_2 + \delta]$. Thus, we are sure that the alternative a_2 is larger, and we select it.
- It is also possible that the values v_1 and v_2 are so close that we cannot tell whether a_1 is larger or a_2 is better; this case corresponds to $|v_1 - v_2| \le \varepsilon$.

Following [11], we assume that in the third case, both alternatives a_1 and a_2 are equally attractable, so any of them can be selected.

What we would like to estimate. Under the above assumption, if the values v_1 and v_2 are close, then both a_1 and a_2 may be selected as the best – and we cannot predict which of them will be selected.

So, for every i, instead of a single probability p_i that the alternative a_i will be selected, we have two different probabilities:

- the probability p_i^+ that a_i *may* be selected, and
- the probability p_i^- that a_i *will necessarily* be selected.

Depending on the decision makers' choice, the actual selection probability p_i can take any value from the interval $[p_i^-, p_i^+]$.

How to estimate p_i^- and p_i^+. According to the above description:

- p_i^- is the probability that $v_j < v_i - \varepsilon$ for all $j \ne i$, and
- p_i^+ is the probability that $v_j \le v_i + \varepsilon$ for all $j \ne i$.

The Monte-Carlo algorithm can be easily modified to compute p_i^- or p_i^+: namely, after we perform N simulations, we can estimate p_i^- as N_i^-/N and p_i^+ as N_i^+/N, where

- N_i^- is the number of simulations in which $v_j < v_i - \varepsilon$ for all $j \ne i$, and
- N_i^+ is the number of simulations in which $v_j \le v_i + \varepsilon$ for all $j \ne i$.

The exact algorithm can be modified as follows:

Towards an algorithm for computing p_i^-. For each i, the probability that $v_i+\varepsilon < v_1$ can be determined as follows:

- If $\overline{v}_i + \varepsilon < v_1$, then $v_i + \varepsilon < v_1$ with probability 1.
- If $v_1 \le \underline{v}_i + \varepsilon$, this means that $v_i + \varepsilon < v_1$ cannot happen at all; the resulting probability is 0.
- Finally, if $\underline{v}_i + \varepsilon \le v_1 \le \overline{v}_i + \varepsilon$, then, since the distribution of v_i is uniform on the interval $[\underline{v}_i + \varepsilon, \overline{v}_i + \varepsilon]$, the probability that $\underline{v}_i + \varepsilon < v_1$ is equal to $\dfrac{v_1 - (\underline{v}_i + \varepsilon)}{\overline{v}_i - \underline{v}_i}$.

Thus, we arrive at the following algorithm.

Algorithm for the exact computation of p_1^-. At the first step of this algorithm, we order those values $\underline{v}_i+\varepsilon$ and $\overline{v}_i+\varepsilon$ ($i \ne 1$) which are inside the interval $[\underline{v}_1, \overline{v}_1]$ into an increasing sequence $v_{(1)} \le v_{(2)} \le \ldots \le v_{(k)}$ ($k \le 2n-2$). As a result, we divide the interval $[\underline{v}_1, \overline{v}_1]$ into $k+1$ zones $z_0 = [\underline{v}_1, v_{(1)})$, $z_1 = [v_{(1)}, v_{(2)})$, $\ldots$, $z_j = [v_{(j)}, v_{(j+1)})$, $\ldots$, $z_k = [v_{(k)}, \overline{v}_1]$.

For the zones z_j for which $v_{(j+1)} \le \underline{v}_i + \varepsilon$ for some i, we set $p_{1j}^- = 0$.

For every other zone, we form the expression

$$p_1^-(v_1) = \prod_{i: v_{(j+1)} \le \overline{v}_i + \varepsilon} \frac{v_1 - \varepsilon - \underline{v}_i}{\overline{v}_i - \underline{v}_i}.$$

This expression is a product of $\le n$ linear functions of the unknown v_1. By multiplying by these functions one by one, we get an explicit expression for a polynomial in v_1. By processing the coefficients of this polynomial one by one, we can provide the explicit analytical expression for the (indefinite) integral $P_{1j}^-(v_1)$ of this polynomial. The desired integral p_{1j}^- can then be computed as the difference $P_{1j}^-(v_{(j+1)}) - P_{1j}^-(v_{(j)})$.

Finally, the desired probability p_1^- is computed as

$$p_1^- = \frac{1}{\overline{v}_1 - \underline{v}_1} \cdot \sum_{j=0}^{k} p_{1j}^-.$$

Algorithm for the exact computation of p_1^+. At the first step of this algorithm, we order those values $\underline{v}_i-\varepsilon$ and $\overline{v}_i-\varepsilon$ ($i \ne 1$) which are inside the interval $[\underline{v}_1, \overline{v}_1]$ into an increasing sequence $v_{(1)} \le v_{(2)} \le \ldots \le v_{(k)}$ ($k \le 2n-2$). As a result, we divide the interval $[\underline{v}_1, \overline{v}_1]$ into $k+1$ zones $z_0 = [\underline{v}_1, v_{(1)})$, $z_1 = [v_{(1)}, v_{(2)})$, $\ldots$, $z_j = [v_{(j)}, v_{(j+1)})$, $\ldots$, $z_k = [v_{(k)}, \overline{v}_1]$.

For the zones z_j for which $v_{(j+1)} < \underline{v}_i - \varepsilon$ for some i, we set $p_{1j} = 0$.

For every other zone, we form the expression

$$p_1^+(v_1) = \prod_{i: v_{(j+1)} \le \overline{v}_i - \varepsilon} \frac{v_1 + \varepsilon - \underline{v}_i}{\overline{v}_i - \underline{v}_i}.$$

This expression is a product of $\le n$ linear functions of the unknown v_1. By multiplying by these functions one by one, we get an explicit expression for a polynomial in v_1. By processing the coefficients of this polynomial one by one, we can provide the explicit analytical expression for the (indefinite) integral $P_{1j}^+(v_1)$ of this polynomial. The desired integral p_{1j}^+ can then be computed as the difference $P_{1j}^+(v_{(j+1)}) - P_{1j}^+(v_{(j)})$.

Finally, the desired probability p_1^+ is computed as

$$p_1^+ = \frac{1}{\overline{v}_1 - \underline{v}_1} \cdot \sum_{j=0}^{k} p_{1j}^+.$$

7 Third Observation: What If the Distributions are not Uniform

Formulation of the problem. For the case of two alternatives, the uniform distribution can be justified by the requirement that the distribution be invariant relative to arbitrary shifts $v_1 \to v_1 + a_1$, $v_2 \to v_2 + a_2$ and conditionally invariant with respect to re-scalings $v_1 \to \lambda_1 \cdot v_1$, $v_2 \to \lambda_2 \cdot v_2$; see, e.g., [6]. To be more precise, the corresponding (generalized) probability density function $\rho(v_1, v_2)$ is invariant relative to shift $\rho(v_1 + a_1, v_2 + a_2) = \rho(v_1, v_2)$ and conditionally invariant with respect to re-scalings: $\rho(\lambda_1 \cdot v_1, \lambda_2 \cdot v_2) = a(\lambda_1, \lambda_2) \cdot \rho(v_1, v_2)$ for some function $a(\lambda_1, \lambda_2)$.

From the measurement viewpoint, a shift means changing the starting point for measuring a quantity, and a scaling means changing a unit in which we measure this quantity. These assumptions work well if v_i are different quantities which can be independently shifted or scaled. In some practical situations, however, values v_1 and v_2 represent the same quantity. We can only shift both values by the same quantity a or scale both by the scale quantity λ. It is therefore desirable to describe probability distributions which are invariant relative to such shifts and scalings.

Formulation of the problem in precise terms. We want to find all symmetric functions $\rho(v_1, v_2) = \rho(v_2, v_1)$ for which $\rho(v_1 + a, v_2 + a) = \rho(v_1, v_2)$ for all a, and for some function $a(\lambda)$, $\rho(\lambda \cdot v_1, \lambda \cdot v_2) = a(\lambda) \cdot \rho(v_1, v_2)$ for all λ.

Towards a solution. Shift-invariance with $a = -v_1$ implies that $\rho(v_1, v_2) = \rho(0, v_2 - v_1)$, i.e., that $\rho(v_1, v_2) = \rho_0(v_2 - v_1)$ for an appropriate function $\rho_0(v)$. Since we want a symmetric distribution $\rho(v_1, v_2)$, we must have $\rho_0(-v) = \rho_0(v)$, i.e., $\rho_0(v) = \rho_0(|v|)$.

In terms of this function $\rho_0(v)$, scale-invariance means that for all λ, we have $\rho_0(\lambda \cdot v) = a(\lambda) \cdot \rho_0(v)$. It is known (see, e.g., [1, 5]) that all measurable solutions of this functional equation have the form $\rho_0(v) = A \cdot v^{-\alpha}$. Since we allow generalized functions, we can also have terms proportional to the δ-function, hence $\rho_0(v) = \varepsilon \cdot \delta(v) + A \cdot v^{-\alpha}$, and

$$\rho(v_1, v_2) = \varepsilon \cdot \delta(v_1 - v_2) + A \cdot |v_1 - v_2|^{-\alpha}.$$

Comment. When both intervals $[\underline{v}_i, \overline{v}_i]$ are non-degenerate, for the uniform distribution, the probability that $v_1 = v_2$ is 0. In contrast, for $\varepsilon > 0$, this probability is positive. This makes sense since degenerate situations (like $v_1 = v_2$) do occur in practice.

Algorithm for computing $p(v_1 > v_2)$. For the case of two alternatives with values $v_1 \in [\underline{v}_1, \overline{v}_1]$ and $v_2 \in [\underline{v}_2, \overline{v}_2]$, we can use Monte-Carlo simulations to find $p(v_1 > v_2)$, $p(v_1 < v_2)$, and $p(v_1 = v_2)$.

Open question. How can we generalize these formulas to the general case of $n \geq$ alternatives?

Acknowledgments. This work was partially supported by NSF Grants CCF-0202042 and EAR-0225670, 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. The authors are thankful to anonymous referees for valuable suggestions.

References

1. Aczel, J.: Lectures on Functional Equations and Their Applications, Dover, New York, 2006
2. Ferson, S., Ginzburg, L., Kreinovich, V., Longpré, L., Aviles, M.: Exact bounds on finite populations of interval data. Reliable Computing **11**, No. 3 (2005) 207–233
3. Huynh, V. N., Nakamori, Y., Lawry, J. A probability-based approach to comparison of fuzzy numbers and applications to target-oriented decision making. IEEE Transactions on Fuzzy Systems, to appear
4. Jaynes, E. T., Bretthorst, G. L.: Probability Theory: The Logic of Science. Cambridge University Press, Cambridge, UK, 2003
5. Nguyen, H. T., Kreinovich, V.: Applications of continuous mathematics in computer science, Kluwer, Dordrecht, 1997
6. Nguyen, H. T., Kreinovich, V., Longpré, L.: Dirty pages of logarithm tables, lifetime of the universe, and (subjective) probabilities on finite and infinite intervals. Reliable Computing **10**, No. 2 (2004) 83–106
7. Robert, C. P., Casella, G., Monte Carlo Statistical Methods, Springer-Verlag, New York, 2004
8. Sevastjanov, P., Venberg, A.: Modelling and simulation of power units work under interval uncertainty. Energy **3** (1998) 66–70 (in Russian)
9. Sevastjanov, P. V., Róg P.: Two-objective method for crisp and fuzzy interval comparison in optimization. Computers & Operations Research **33** (2006) 115–131
10. Sevastianov, P., Róg, P., Venberg, A.: The constructive numerical method of interval comparison. Proceedings of the International Conference PPAM01, Naleczow, 2001, 756–761
11. Wagman, D., Schneider, M., Schnaider, E.: On the use of interval mathematics in fuzzy expert systems. International Journal of Intelligent Systems **9** (1994) 241–259
12. Yager, R. R., Detyniecki, M., Bouchon-Meunier, B.: A context-dependent method for ordering fuzzy numbers using probabilities. Information Sciences **138** (2001) 237–255