Estimating Information Amount under Interval Uncertainty: Algorithmic Solvability and Computational Complexity

Gang Xiang[1], Olga Kosheleva[1],
Vladik Kreinovich[1], and George J. Klir[2]

[1]University of Texas, El Paso, TX 79968, USA
{gxiang,olgak,vladik}@utep.edu
[2]Center for Intelligent Systems
Thomas J. Watson School of Engineering and Applied Science
State University of New York, Binghamton, NY 13902-6000
gklir@binghamton.edu

Abstract

In most real-life situations, we have *uncertainty*: we do not know the exact state of the world, there are several (n) different states which are consistent with our knowledge. In such situations, it is desirable to gauge how much information we need to gain to determine the actual state of the world. A natural measure of this amount of information is the average number of "yes"-"no" questions that we need to ask to find the exact state. When we know the probabilities $p_1, \ldots, p_n$ of different states, then, as Shannon has shown, this number of questions can be determined as $S = -\sum\limits_{i=1}^{n} p_i \cdot \log_2(p_i)$.

In many real-life situations, we only have partial information about the probabilities; for example, we may only know intervals $\mathbf{p}_i = [\underline{p}_i, \overline{p}_i]$ of possible values of p_i. For different values $p_i \in \mathbf{p}_i$, we get different values S. So, to gauge the corresponding uncertainty, we must find the range $\mathbf{S} = [\underline{S}, \overline{S}]$ of possible values of S. In this paper, we show that the problem of computing $\mathbf{S}$ is, in general, NP-hard, and we provide algorithms that efficiently compute $\mathbf{S}$ in many practically important situations.

1 Formulation of the Problem

In most practical situations, our knowledge is incomplete: there are several (n) different states which are consistent with our knowledge. How can we gauge this uncertainty? A natural measure of uncertainty is the average number of binary ("yes"-"no") questions that we need to ask to find the exact state. This idea is behind Shannon's *information theory*: according to this theory, when we know the probabilities $p_1, \ldots, p_n$ of different states (for which $\sum p_i = 1$), then this average number of questions is equal to $S = -\sum\limits_{i=1}^{n} p_i \cdot \log_2(p_i)$. In information theory, this average number of question is called the *amount of information*.

In practice, we rarely know the exact values of the probabilities p_i; these probabilities come from experiments and are, therefore, only known with uncertainty. Usually, from the experiments, we can find *confidence intervals* $\mathbf{p}_i = [\underline{p}_i, \overline{p}_i]$, i.e., intervals which contain the (unknown) values p_i. Since $p_i \ge 0$ and $\sum p_i = 1$, we must have $\underline{p}_i \ge 0$ and $\sum \underline{p}_i \le 1 \le \sum \overline{p}_i$. How can we estimate the amount of information under such interval uncertainty?

For different values $p_i \in \mathbf{p}_i$, we get, in general, different values of the amount of information S. Since S is a continuous function, the set of possible values of S is an interval. So, to gauge the corresponding uncertainty, we must find the range $\mathbf{S} = [\underline{S}, \overline{S}]$ of possible values of S.

Thus, we arrive at the following computational problem: given n intervals $\mathbf{p}_i = [\underline{p}_i, \overline{p}_i]$, find the range

$$\mathbf{S} = [\underline{S}, \overline{S}] = \left\{ -\sum_{i=1}^{n} p_i \cdot \log_2(p_i) \,\middle|\, p_i \in \mathbf{p}_i \,\&\, \sum_{i=1}^{n} p_i = 1 \right\}.$$

Shannon's entropy is not the only way to describe uncertainty. Researchers have observed that many practically useful properties of the Shannon's entropy function S do not use its specific form, they only use the fact that the expression $f(p) = -p \cdot \log_2(p)$ is equal to 0 for $p = 0$ and for $p = 1$ and that this expression is differentiable and strictly concave – i.e., that its second derivative $f''(p)$ is negative for all p.

As a result of this observation, they proposed to use *generalized entropy* measures $S = \sum\limits_{i=1}^{n} f(p_i)$ for some differentiable strictly concave function $f(p)$ for which $f(0) = f(1) = 0$. Such generalized entropy measures are indeed useful in many practical applications; see, e.g., [7]. In addition to Shannon's entropy function $f(p) = -p \cdot \log_2(p)$, several other functions are used in practice such as $f(p) = p \cdot (1 - p^{\beta})$ for some $\beta > 0$ – a function that tends to Shannon's entropy function when $\beta \to 0$.

For such generalized information measures, we have a similar problem: given n intervals $\mathbf{p}_i = [\underline{p}_i, \overline{p}_i]$, find the range

$$\mathbf{S} = [\underline{S}, \overline{S}] = \left\{ \sum_{i=1}^{n} f(p_i) \,\middle|\, p_i \in \mathbf{p}_i \,\&\, \sum_{i=1}^{n} p_i = 1 \right\}.$$

In this paper, we show that the problem of computing $\mathbf{S}$ is, in general, NP-hard, and we provide algorithms that efficiently compute $\mathbf{S}$ in many practically important situations.

Comment. For the case of Shannon's entropy, these results were first announced in our conference paper [13].

2 Effective Algorithm for Computing $\overline{S}$

In this section, we will describe an algorithm that computes the upper endpoint $\overline{S}$ of the information interval $\mathbf{S}$ in time $O(n \cdot \log(n))$.

Analysis of the problem. Let $(p_1, \ldots, p_n)$ be the values of probabilities at which the entropy S attains its maximum. The fact that S attains its maximum means that if we change the values p_i, then the corresponding change ΔS in S is non-positive: $\Delta S \le 0$. We will use this condition for different changes in p_i.

For each value of p_i, we have three possibilities:

- this value can be strictly inside the corresponding interval $[\underline{p}_i, \overline{p}_i]$;
- this value can be at the left end of this interval, i.e., $p_i = \underline{p}_i$; and
- this value can be at the right end of this interval, i.e., $p_i = \overline{p}_i$.

Let us consider these possibilities one by one.

Let us first consider the values p_j which are strictly inside the corresponding intervals. If for some j and k, the corresponding probabilities are strictly inside the corresponding intervals, i.e., if we have $p_j \in (\underline{p}_j, \overline{p}_j)$ and $p_k \in (\underline{p}_k, \overline{p}_k)$, then for a sufficiently small real number Δ, we can replace p_j with $p_j + \Delta$ and p_k with $p_k - \Delta$ and still get a sequence of probabilities for which $p_i \in [\underline{p}_i, \overline{p}_i]$ for all i and $\sum p_i = 1$. For small Δ, the corresponding change ΔS in entropy is equal to

$$\left(\frac{\partial S}{\partial p_j} - \frac{\partial S}{\partial p_k}\right) \cdot \Delta + o(\Delta) = (f'(p_j) - f'(p_k)) \cdot \Delta + o(\Delta).$$

Since Δ can be positive or negative, the only way to have $\Delta S \le 0$ for all small Δ is to make sure that the coefficient at Δ is equal to 0, i.e., that $f'(p_j) - f'(p_k) = 0$. Since $f(p)$ is a strictly concave function, i.e., $f''(p) < 0$, the derivative $f''(p) = (f'(p))'$ of the function $f'(p)$ is always negative – which means that this derivative is a strictly decreasing function. Thus, $f'(p_j) = f'(p_k)$ implies that $p_j = p_k$ – i.e., that all the values p_j which are inside the corresponding intervals coincide. Let us denote this common value of p_j by p.

Let us now consider the situation when p_j is at the left end of the corresponding interval, i.e., when $p_j = \underline{p}_j$. If for some other k, the corresponding value p_k is at the right end or strictly inside the corresponding interval, then $p_k > \underline{p}_k$. In this case, we can only make a similar change $p_j \to p_j + \Delta$ and $p_k \to p_k - \Delta$ when $\Delta > 0$. Then, the requirement that $\Delta S \le 0$ means that the coefficient at Δ should be non-positive, i.e., that $f'(p_j) - f'(p_k) \le 0$. Since the derivative $f'(p)$ is a strictly decreasing function, we conclude that $p_k \le p_j$. In particular, for the case when p_k is inside the corresponding interval – and is, thus, equal to p – we conclude that $p \le p_j$.

Similarly, if p_j is at the right end of the corresponding interval, i.e., if $p_j = \overline{p}_j$, then, for every k for which $p_k > \underline{p}_k$, we conclude that $p_k \ge p_j$. In particular, we can conclude that $p_j \le p$.

Let us now consider the case when there are some values p_i strictly inside the corresponding interval, so there is a value p. Let us show that is we know where p is located in comparison with all the endpoints $[\underline{p}_i, \overline{p}_i]$, then we can uniquely determine all the values p_i.

Indeed, if the entire interval $[\underline{p}_i, \overline{p}_i]$ is located to the left of p, i.e., if $\overline{p}_i < p$, then:

- the minimum cannot be attained strictly inside the interval – because it would have been attained at the point $p_i = p$, and we are considering the case when the entire interval $[\underline{p}_i, \overline{p}_i]$ is located to the left of p;

- similarly, the minimum cannot be attained for $p_i = \overline{p}_i$, because then, as we have proven, we should have $p \le p_i$, and the entire interval $[\underline{p}_i, \overline{p}_i]$ is located to the left of p.

Thus, in this case, the only remaining possibility is $p_i = \overline{p}_i$.

Similarly, if the entire interval $[\underline{p}_i, \overline{p}_i]$ is located to the right of p, i.e., if $p < \underline{p}_i$, then $p_i = \underline{p}_i$.

If $\underline{p}_i < p < \overline{p}_i$, then, similarly, we cannot have $p_i = \underline{p}_i$ and $p_i = \overline{p}_i$, so we must have p_i inside and hence, $p_i = p$.

To exploit this conclusion, let us formalize how we can describe the location of p in relation to $2n$ endpoints. If we sort these endpoints $\underline{p}_i$ and $\overline{p}_i$ into a sequence $p_{(1)} \le p_{(2)} \le \ldots \le p_{(2n)}$, then we divide the entire real line into $2n+1$ "zones" $[p_{(k)}, p_{(k+1)}]$, where we denoted $p_{(0)} \stackrel{\text{def}}{=} 0$ and $p_{(2n+1)} \stackrel{\text{def}}{=} 1$.

Let us pick a zone $[p_{(k)}, p_{(k+1)}]$, and show how we can find the possibly optimal values p_i (and the corresponding value of the entropy) under the assumption that the (unknown) value p belongs to the this zone.

If $\overline{p}_i < p$, then we must have $\overline{p}_i \le p_{(k)}$ – otherwise, if $\overline{p}_i > p_{(k)}$, then, since $p_{(k)}$ describe all the endpoints, we would have $\overline{p}_i \ge p_{(k+1)}$ and hence $\overline{p}_i > p$. Thus, in the optimal arrangement of probabilities, we have $p_i = \overline{p}_i$.

Similarly, if $\underline{p}_i > p$, then we have $p_i = \underline{p}_i$. For all other i, we have $p_i = p$. This value p can be computed based on the fact that $\sum p_i = 1$.

For each of $2n+1$ zones, we need to analyze n values p_i; thus, for each of the zones, we need $O(n)$ computation steps. Overall, we get a quadratic algorithm for computing $\overline{S}$.

Before we describe this algorithm, we should mention that the above description only works when we actually have an index i for which p_i is strictly inside the corresponding interval. If no such index exists, then we can still conclude that every value $p_j = \overline{p}_j$ is smaller than or equal than every value $p_k = \underline{p}_k$. Thus, there exists a value p that is greater than or equal than all j for which $p_j = \overline{p}_j$ and less than or equal than all k for which $p_k = \underline{p}_k$. By using this p, we arrive at the same conclusion about the values p_i.

Thus, in general, we arrive at the following algorithm (first described in [10]).

Quadratic-time algorithm for computing $\overline{S}$.

- First, we sort $2n$ endpoints of n intervals $\mathbf{p}_i$ into an increasing sequence $p_{(0)} = 0 < p_{(1)} < p_{(2)} < \ldots < p_{(m)} < p_{(m+1)} = 1$. (If all the endpoints are different, then $m = 2n$, but since some endpoints may coincide, we may have $m < 2n$; in general, $m \le 2n$.)
- Second, for every k from 0 to $m-1$, we compute the following three values:
$$M_k = \sum_{i:\overline{p}_i \le p_{(k)}} f(\overline{p}_i) + \sum_{j:\underline{p}_j \ge p_{(k+1)}} f(\underline{p}_j);$$
$$P_k = \sum_{i:\overline{p}_i \le p_{(k)}} \overline{p}_i + \sum_{j:\underline{p}_j \ge p_{(k+1)}} \underline{p}_j;$$
$$n_k = \#\{i : \overline{p}_i \le p_{(k)} \vee \underline{p}_i \ge p_{(k+1)}\}.$$
- If $n_k = n$, we take $S_k = M_k$.
- If $n_k < n$, then we compute $p = \dfrac{1 - P_k}{n - n_k}$.
  - If $p \in [p_{(k)}, p_{(k+1)}]$, then we compute
  $$S_k = M_k + (n - n_k) \cdot f(p).$$
  - Otherwise, we ignore this k.
- Finally, we find the largest of these values S_k as the desired bound $\overline{S}$.

How to reduce the computation time to $O(n \cdot \log(n))$. Let us show that the computation time for this algorithm can be reduced to $O(n \cdot \log_2(n))$. Indeed, sorting requires $O(n \cdot \log_2(n))$ steps; see, e.g., [6]. Once we have a sorted list, we can find, for each of the $2n$ endpoints $\underline{p}_i$ and $\overline{p}_i$, where they are in this sorting. We can thus, for each of the values $p_{(j)}$, mark which endpoints coincide with this value.

The initial computation of the values M_0, P_0, and n_0 requires $O(n)$ steps. Once we go from M_k to M_{k+1} (or from P_k to P_{k+1}), we only need to update the values corresponding to the endpoints of this zone. Overall, for all the updates, we thus need as much time as there are updated values p_i overall.

Each endpoint in this arrangement changes only once, so overall, we need a linear number of steps $(2n)$ to update all the values M_k, all the values P_k, and all the values n_k. Thus, overall, we need time $O(n \cdot \log_2(n)) + O(n) + O(n) = O(n \cdot \log_2(n))$.

Let us describe this new algorithm in precise terms.

New algorithm for computing $\overline{S}$.

- First, we sort $2n$ endpoints $\underline{p}_i$ and $\overline{p}_i$ into a sequence

$$0 = p_{(0)} < p_{(1)} < p_{(2)} < \ldots < p_{(m)} < p_{(m+1)} = 1.$$

  In the process of this sorting, for each k from 1 to m, we form the sets $A_k^- = \{i : \underline{p}_i = p_{(k)}\}$ and $A_k^+ = \{i : \overline{p}_i = p_{(k)}\}$.

- Then, for each k from 0 to m, we compute the values M_k, P_k, and n_k as follows.

  – We start with $M_0 = \sum\limits_{i=1}^{n} f(\underline{p}_i)$, $P_0 = \sum\limits_{i=1}^{n} \underline{p}_i$, and $n_0 = n$.

  – Once we know M_k, P_k, and n_k, we compute the next values of these quantities as follows:

$$M_{k+1} = M_k - \sum_{j \in A_{k+1}^-} f(\underline{p}_j) + \sum_{j \in A_{k+1}^+} f(\overline{p}_j);$$

$$P_{k+1} = P_k - \sum_{j \in A_{k+1}^-} \underline{p}_j + \sum_{j \in A_{k+1}^+} \overline{p}_j;$$

$$n_{k+1} = n_k - \#(A_{k+1}^-) + \#(A_{k+1}^+).$$

- If $n_k = n$, we take $S_k = M_k$.

- If $n_k < n$, then we compute $p = \dfrac{1 - P_k}{n - n_k}$.

  – If $p \in [p_{(k)}, p_{(k+1)}]$, then we compute

$$S_k = M_k + (n - n_k) \cdot f(p).$$

  – Otherwise, we ignore this k.

- Finally, we find the largest of these values S_k as the desired bound $\overline{S}$.

3 Computing $\underline{S}$ is, in General, NP-Hard

Several algorithms for computing $\underline{S}$ are known; see, e.g., [1, 2, 3, 4, 5]. In the worst case, these algorithms require time that grows exponentially with n. In this section, we will show that computing $\underline{S}$ in general case is, indeed, NP-hard – even when we consider Shannon's entropy.

Proof of NP-hardness. By definition, a problem is called NP-hard if every problem from the class NP can be reduced to it; see, e.g., [12]. To prove that a problem $\mathcal{P}$ is NP-hard, it is sufficient to reduce one of the known NP-hard problems $\mathcal{P}_0$ to $\mathcal{P}$. The reason for this is as follows: since $\mathcal{P}_0$ is known to be NP-hard, it means that every problem from the class NP can be reduced to $\mathcal{P}_0$, and since $\mathcal{P}_0$ can be reduced to $\mathcal{P}$, thus, we can deduce that every problem from the class NP can be reduced to $\mathcal{P}$.

1°. For our proof, we will select the following *subset* problem as the known NP-hard problem $\mathcal{P}_0$: given n positive integers $s_1, \ldots, s_n$, check whether there exist signs $\eta_i \in \{-1, +1\}$ for which the signed sum $\sum\limits_{i=1}^{n} \eta_i \cdot s_i$ equals to 0.

We will eventually prove that this problem can be reduced to the problem of computing $\underline{S}$; this computational problem will be denoted by $\mathcal{P}$. However, directly proving that $\mathcal{P}_0$ can be reduced to $\mathcal{P}$ seems to be difficult. Therefore, we introduce the following auxiliary problem, denoted as $\mathcal{P}_1$: given a real number $a > 0$ and n intervals $\mathbf{q}_1 = [\underline{q}_1, \overline{q}_1], \mathbf{q}_2 = [\underline{q}_2, \overline{q}_2], \ldots, \mathbf{q}_n = [\underline{q}_n, \overline{q}_n]$, where $\sum\limits_{i=1}^{n} \underline{q}_i \le a \le \sum\limits_{i=1}^{n} \overline{q}_i$ and $0 \le \underline{q}_i$ for all i, find the lower endpoint $\underline{L}$ of the range

$$\mathbf{L} = [\underline{L}, \overline{L}] = \left\{ -\sum_{i=1}^{n} q_i \cdot \log_2(q_i) \,\middle|\, q_i \in \mathbf{q}_i \;\&\; \sum_{i=1}^{n} q_i = a \right\}$$

Comment. Similarly to our problem $\mathcal{P}$, the new problem $\mathcal{P}_1$ is also about minimizing entropy S: the only difference is that instead of the restriction $\sum\limits_{i=1}^{n} p_i = 1$, we have a new restriction $\sum\limits_{i=1}^{n} q_i = a$.

2°. To reduce $\mathcal{P}_0$ to $\mathcal{P}_1$ means that for every instance $(s_1, \ldots, s_n)$ of the problem $\mathcal{P}_0$, we can find a corresponding instance of the problem $\mathcal{P}_1$ from whose solution, we can easily check whether the desired signs η_i in $\mathcal{P}_0$ exist.

In order to select an appropriate instance, let us first analyze the function $-q \cdot \log_2(q)$. This function is equal to 0 for $q = 0$ and for $q = 1$. It attains its maximum when

$$\frac{\partial}{\partial q}(-q \cdot \log_2(q)) = -\log_2(e) \cdot (1 + \ln(q)) = 0,$$

i.e., when $q = \frac{1}{e}$. The corresponding maximum is equal to $-\frac{1}{e} \cdot \log_2\left(\frac{1}{e}\right) = \frac{\log_2(e)}{e}$. We have already mentioned that the function $-q \cdot \log_2(q)$ is concave; therefore, for every real number r between 0 and the maximum – i.e., for which $0 < r < \frac{\log_2(e)}{e}$, there exist exactly two different values q for which $-q \cdot \log_2(q) = r$. Let us denote the smaller of these two values by $q^-(r)$, and the larger one by $q^+(r)$. We can check that that $0 < q^-(r) < q^+(r) < 1$ and $0 < q^+(r) - q^-(r) < 1$. As r grows from 0 to its largest value, the difference $q^+(r) - q^-(r)$ decreases from 1 to 0.

Now, for each instance $(s_1, \ldots, s_n)$ of the problem $\mathcal{P}_0$, we select the corresponding instance of the problem $\mathcal{P}_1$, i.e., the intervals $[\underline{q}_i, \overline{q}_i]$ and the real number a, as follows:

- First, we select a positive real number z for which $z \cdot \max(s_i) < 1$.
- Next, for each i from 1 to n, we find r_i for which $q^+(r_i) - q^-(r_i) = z \cdot s_i$, and take $\underline{q}_i = q^-(r_i)$ and $\overline{q}_i = q^+(r_i)$.
- Finally, we select $a = \sum\limits_{i=1}^{n} \frac{\underline{q}_i + \overline{q}_i}{2}$.

It is easy to check that for thus selected values, $\underline{q}_i \geq 0$ and $\sum\limits_{i=1}^{n} \underline{q}_i \leq a \leq \sum\limits_{i=1}^{n} \overline{q}_i$.

Let $L_0 \stackrel{\text{def}}{=} -\sum\limits_{i=1}^{n} \underline{q}_i \cdot \log_2(\underline{q}_i)$. We will show that $\underline{L} = L_0$ if and only if there exist signs η_i for which $\sum\limits_{i=1}^{n} \eta_i \cdot s_i = 0$.

3°. Let us first prove that $\underline{L} \geq L_o$.

Indeed, due to our choice of $\underline{q}_i$ and $\overline{q}_i$, the function $-q \cdot \log_2(q)$ attains the same value at the two endpoints of the interval $[\underline{q}_i, \overline{q}_i]$ and is larger everywhere inside this interval. Thus, for every i and for every $q_i \in [\underline{q}_i, \overline{q}_i]$, we have $-q_i \cdot \log_2(q_i) \geq -\underline{q}_i \cdot \log_2(\underline{q}_i)$. By adding these inequalities, we conclude that

$$L = -\sum_{i=1}^{n} q_i \cdot \log_2(q_i) \geq -\sum_{i=1}^{n} -\underline{q}_i \cdot \log_2(\underline{q}_i) = L_0.$$

Since all the values of L are larger than or equal to L_0, the smallest possible value $\underline{L}$ of the function L also satisfies the inequality $\underline{L} = L_0$.

4°. Let us first prove that if the desired signs η_i exist, then $\underline{L} = L_0$.

Indeed, in this case, we can select $q_i = \underline{q}_i$ when $\eta_i = -1$ and $q_i = \overline{q}_i$ when $\eta_i = 1$. Both cases can be described by a single formula

$$q_i = \frac{\underline{q}_i + \overline{q}_i}{2} + \frac{\eta_i \cdot (\overline{q}_i - \underline{q}_i)}{2} = \frac{\underline{q}_i + \overline{q}_i}{2} + \frac{\eta_i \cdot z \cdot s_i}{2}.$$

Since $-\underline{q}_i \cdot \log_2(\underline{q}_i) = -\overline{q}_i \cdot \log_2(\overline{q}_i)$, for this choice of q_i, we have

$$L = -\sum_{i=1}^n q_i \cdot \log_2(q_i) = -\sum_{i=1}^n \underline{q}_i \cdot \log_2(\underline{q}_i) = L_0.$$

In this case,

$$\sum_{i=1}^n q_i = \sum_{i=1}^n \left(\frac{\underline{q}_i + \overline{q}_i}{2} + \frac{\eta_i \cdot z \cdot s_i}{2}\right) =$$

$$\sum_{i=1}^n \frac{\underline{q}_i + \overline{q}_i}{2} + \frac{z}{2} \cdot \sum_{i=1}^n \eta_i \cdot s_i = \sum_{i=1}^n \frac{\underline{q}_i + \overline{q}_i}{2} = a.$$

Since for this choice of q_i, we have $L = L_0$, we can thus onclude that the smallest possible value $\underline{L}$ of L cannot exceed L_0: $\underline{L} \leq L_0$.

We have already proven that $\underline{L} \geq L_0$, so we can conclude that $\underline{L} = L_0$.

5°. Now let us prove that if $\underline{L} = L_0$, then the desired signs η_i exists.

Let $q_1, \ldots, q_n$ be the values that minimize L, i.e., for which $L = \underline{L}$. From the equality $\underline{L} = L_0$, we will conclude that for every i, we have either $q_i = \underline{q}_i$ or $q_i = \overline{q}_i$. This can be proven by reduction to a contradiction: if for some j, we have $q_j \neq \underline{q}_j$ and $q_j \neq \overline{q}_j$, then we will get $-q_j \cdot \log_2(q_j) > -\underline{q}_j \cdot \log_2(\underline{q}_j)$. For every other i, we have $-q_i \cdot \log_2(q_i) \geq -\underline{q}_i \cdot \log_2(\underline{q}_i) = -\overline{q}_i \cdot \log_2(\overline{q}_i)$. By adding all these inequalities, we can conclude that

$$\underline{L} = L = -\sum_{i=1}^n q_i \cdot \log_2(q_i) > -\sum_{i=1}^n \underline{q}_i \cdot \log_2(\underline{q}_i) = L_0,$$

which contradicts to our assumption that $\underline{L} = L_0$. This contradiction shows that indeed, for every i, we have either $q_i = \underline{q}_i$ or $q_i = \overline{q}_i$.

Let us set $\eta_i = -1$ when $q_i = \underline{q}_i$ and $\eta_i = 1$ when $q_i = \overline{q}_i$. Then,

$$q_i = \frac{\underline{q}_i + \overline{q}_i}{2} + \frac{\eta_i \cdot z \cdot s_i}{2}.$$

From the condition $\sum q_i = a$, we now conclude that

$$a = \sum_{i=1}^n q_i = \sum_{i=1}^n \frac{\underline{q}_i + \overline{q}_i}{2} + \frac{\eta_i \cdot z \cdot s_i}{2} = a + z \cdot \sum_{i=1}^n \eta_i \cdot s_i,$$

hence $\sum\limits_{i=1}^n \eta_i \cdot s_i = 0$.

Therefore, we have proven that the subset problem $\mathcal{P}_0$ can be reduced to the auxiliary problem $\mathcal{P}_1$. Thus, the auxiliary problem $\mathcal{P}_1$ is also NP-hard.

6°. To complete the proof, we need to show that the auxiliary problem $\mathcal{P}_1$ can be reduced to our $\mathcal{P}$. In other words, for every instance of the auxiliary problem $\mathcal{P}_1$, we can find the corresponding instance of the original problem $\mathcal{P}$, from whose solution we can easily find the solution to the instance of $\mathcal{P}_1$.

Indeed, let us consider an instance of the auxiliary $\mathcal{P}_1$, i.e., the intervals $[\underline{q}_i, \overline{q}_i]$ and the real number a for which $\underline{q}_i \geq 0$ and $\sum\limits_{i=1}^{n} \underline{q}_i \leq a \leq \sum\limits_{i=1}^{n} \overline{q}_i$. As the corresponding instance of the original problem, we will take $\underline{p}_i = \dfrac{\underline{q}_i}{a}$ and $\overline{p}_i = \dfrac{\overline{q}_i}{a}$.

Possible values $p_i \in [\underline{p}_i, \overline{p}_i]$ and $q_i \in [\underline{q}_i, \overline{q}_i]$ can be obtained from each other by, correspondingly, multiplying or dividing by a. For each set $q_i = p_i \cdot a$, we have

$$L = -\sum_{i=1}^{n} q_i \cdot \log_2(q_i) = -\sum_{i=1}^{n} a \cdot p_i \cdot \log_2(a \cdot p_i) = -a \cdot \sum_{i=1}^{n} p_i \cdot \log_2(a \cdot p_i) =$$

$$-a \cdot \sum_{i=1}^{n} p_i \cdot \log_2(p_i) - a \cdot \log_2(a) \cdot \sum_{i=1}^{n} p_i =$$

$$-a \cdot \sum_{i=1}^{n} p_i \cdot \log_2(p_i) - a \cdot \log_2(a) = a \cdot S - a \cdot \log_2(a).$$

Thus, L is an increasing function of S, hence the minimum $\underline{L}$ is equal to

$$\underline{L} = a \cdot \underline{S} - a \cdot \log(a).$$

Therefore, if we get the solution $\underline{S}$ to the above instance of our original problem $\mathcal{P}$, we will thus be able to easily compute the solution $\underline{L}$ to the corresponding instance of the auxiliary problem $\mathcal{P}_1$.

Therefore, the auxiliary problem $\mathcal{P}_1$ – whose NP-hardness we have already proven – can be reduced to the original problem $\mathcal{P}$. So, we have prove that the original problem $\mathcal{P}$ of computing $\underline{S}$ is indeed NP-hard.

4 Effective Algorithm for Computing $\underline{S}$ When Intervals Are Not Contained in Each Other

Motivation. Usually, when we know p_i with some uncertainty, we know the approximate values $\widetilde{p}_i$ and the accuracy Δ of this approximation. In this case, we know that the actual (unknown) value of p_i belongs to the interval $[\widetilde{p}_i - \Delta, \widetilde{p}_i + \Delta]$. Since these intervals all have the same width 2Δ, none of them can be a proper subset of the other. It turns out that if we restrict ourselves to intervals that satisfy this condition, then it is possible to compute $\underline{S}$ efficiently.

Result. We say that intervals $[\underline{p}_i, \overline{p}_i]$ satisfy the *subset property* if $[\underline{x}_i, \overline{x}_i] \not\subset (\underline{x}_j, \overline{x}_j)$ for all i and j (for which the intervals $\mathbf{x}_i$ and $\mathbf{x}_j$ are non-degenerate). In this section, we describe an $O(n \cdot \log(n))$ algorithm that computes $\underline{S}$ for all cases when the subset property holds.

Algorithm.

- First, we sort n intervals $\mathbf{p}_i$ in lexicographic order:

$$\mathbf{p}_1 \le_{\text{lex}} \mathbf{p}_2 \le_{\text{lex}} \ldots \le_{\text{lex}} \mathbf{p}_n$$

  where $[\underline{a}, \overline{a}] \le_{\text{lex}} [\underline{b}, \overline{b}]$ if and only if either $\underline{a} < \overline{b}$, or $\underline{a} = \underline{b}$ and $\overline{a} \le \overline{b}$.

- Second, for each i from 1 to n, we compute

$$M_i = \sum_{j:j<i} f(\underline{p}_j) + \sum_{m:m>i} f(\overline{p}_m);$$

$$P_i = \sum_{j:j<i} \underline{p}_j + \sum_{m:m>i} \overline{p}_m.$$

  First, we compute $M_1 = \sum\limits_{j=2}^{n} f(\overline{p}_j)$ and $P_1 = \sum\limits_{j=2}^{n} \overline{p}_j$; then, we sequentially compute other values as

$$M_i = M_{i-1} + f(\underline{p}_{i-1}) - f(\overline{p}_i);$$

$$P_i = P_{i-1} + \underline{p}_{i-1} - \overline{p}_i.$$

- For every i, we compute $p_i = \dfrac{1 - P_i}{n - 1}$. If $p_i \in [\underline{p}_i, \overline{p}_i]$, we compute

$$S_i = M_i + f(p_i).$$

- Finally, we return the smallest of these values S_i as $\underline{S}$.

Justification of the algorithm. It is easy to show that when we sort the intervals in lexicographic order, then both their lower endpoints $\underline{p}_i$ and upper endpoints $\overline{p}_i$ are also sorted: $\underline{p}_i \le \underline{p}_{i+1}$ and $\overline{p}_i \le \overline{p}_{i+1}$. (Indeed, otherwise, we would get a violation of the subset property.) Let us thus assume that the intervals are thus sorted.

Let us now show that it is sufficient to consider monotonic optimal tuples $p_1, \ldots, p_n$, for which $p_i \le p_{i+1}$ for all i. Indeed, if $p_i > p_{i+1}$, then, since $p_i \le \overline{p}_i \le \overline{p}_{i+1}$ and $p_i > p_{i+1} \ge \underline{p}_{i+1}$, we have $p_i \in [\underline{p}_{i+1}, \overline{p}_{i+1}]$ and similarly $p_{i+1} \in [\underline{p}_i, \overline{p}_i]$. Thus, we can swap the values p_i and p_{i+1} without changing the value of S. We can repeat this swap as many times as necessary until we get a monotonic tuple that has the exact same value $S = \underline{S}$.

Let us now show that in the optimal tuple, at most one p_i can be inside the corresponding interval. Indeed, if we have two values p_j and p_k strictly inside their intervals, then, similarly to the case of $\overline{S}$, we can conclude that $p_j = p_k$. Now, for $p_j - \Delta = p - \Delta$ and $p_k + \Delta = p + \Delta$, the function S should have a minimum at $\Delta = 0$ and thus, its second derivative relative to Δ should be non-negative. However, an explicit computation shows that this derivative is negative. Thus, our assumption is false, and at most one p_j can be inside the corresponding interval.

Similar to the case of $\overline{S}$, we can now conclude that:

- if $p_j = \underline{p}_j$ and $p_m > \underline{p}_m$, then $p_j \le p_m$; and
- if $p_m = \overline{p}_m$ and $p_j < \overline{p}_j$, then $p_m \ge p_j$.

Thus, each value $p_j = \underline{p}_j$ precede all the values $p_m = \overline{p}_m$, and the only value p_i which is strictly inside the corresponding interval lies in between these values. Thus, in a monotonic optimal tuple $p_1, \ldots, p_n$, the first elements are equal to $\underline{p}_j$, then we may have one element which is strictly inside its interval, and then we have values $p_m = \overline{p}_m$.

The above algorithm tests all such (possibly optimal) sequences and finds the one for which the entropy is the largest.

Acknowledgments.

This work was supported in part by NASA under cooperative agreement NCC5-209, by NSF grant EAR-0225670, NIH grant 3T34GM008048-20S1, and Army Research Lab grant DATM-05-02-C-0046.

The authors are thankful to the participants of the 24nd International Conference of the North American Fuzzy Information Processing Society NAFIPS'2005 (Ann Arbor, Michigan, June 22–25, 2005) for valuable discussions.

References

[1] J. Abellan and S. Moral, "A non-specificity measure for convex sets of probability distributions", *Intern. J. of Uncertainty, Fuzziness and Knowledge-Based Systems*, 2000, Vol. 8, No. 3, pp. 357–367.

[2] J. Abellan and S. Moral, "Maximum of entropy for credal sets", *Intern. J. of Uncertainty, Fuzziness, and Knowledge-Based Systems*, 2003, Vol. 11, No. 5, pp. 587–597.

[3] J. Abellan and S. Moral, “Range of entropy for credal sets”, In: M. López-Diaz et al. (eds.), *Soft Methodology and Random Information Systems*, Springer, Berlin and Heidelberg, 2004, pp. 157–164.

[4] J. Abellan and S. Moral, “Difference of entropies as a nonspecificity function on credal sets”, *Intern. J. of General Systems*, 2005, Vol. 34, No. 3, pp. 201–214.

[5] J. Abellan and S. Moral, “An algorithm that attains the maximum of entropy for order-2 capacities”, *Intern. J. of Uncertainty, Fuzziness, and Knowledge-Based Systems* (to appear).

[6] Th. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 2001.

[7] G. J. Klir, *Uncertainty and Information: Foundations of Generalized Information Theory*, J. Wiley, Hoboken, New Jersey, 2005.

[8] G. J. Klir, “Measuring Uncertainty Associated with Convex Sets of Probability Distributions: A New Approach”, *Proc. NAFIPS'2005*, Ann Arbor, Michigan, June 22–25, 2005, pp. 61–64.

[9] G. J. Klir and M. J. Wierman, *Uncertainty-Based Information: Elements of Generalized Information Theory*, Physica-Verlag/Springer-Verlag, Heidelberg and New York, 1999.

[10] V. Kreinovich, “Maximum entropy and interval computations”, *Reliable Computing*, 1996, Vol. 2, No. 1, pp. 63–79.

[11] V. Kreinovich, G. Xiang, and S. Ferson, “How the Concept of Information as Average Number of ‘Yes-No’ Questions (Bits) Can Be Extended to Intervals, P-Boxes, and more General Uncertainty”, *Proc. NAFIPS'2005*, Ann Arbor, Michigan, June 22–25, 2005, pp. 80–85.

[12] C. H. Papadimitriou, *Computational Complexity*, Addison-Wesley, Reading, Massachusetts, 1994.

[13] G. Xiang, O. Kosheleva, and G. J. Klir, “Estimating Information Amount under Interval Uncertainty: Algorithmic Solvability and Computational Complexity”, *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMU'06*, Paris, France, July 2–7, 2006 (to appear).