Computing the Value of a Boolean Expression with Interval Inputs is NP-Hard

Leticia S. Chee

Expert Knowledge is Often Uncertain. In many areas, e.g., in medicine, geology, etc., we do not have the complete knowledge of the phenomena that we are interested in, so, we have to rely on expert knowledge. In every field, there are usually only a few high quality experts, so, it is desirable to pit the experts' knowledge inside the computer and design a system that would let everyone use this knowledge. Such systems are called *expert systems*.

Expert knowledge is often formulated in not very precise terms (e.g., "the temperature is high", or "the success of this operation is highly probable"). The degree of uncertainty is very important for decision making: e.g., if the success of the operation is "highly probable", a surgeon would most likely perform it, while if this success is simply "probable", he will have to take other factors into consideration. So, when we design an expert system, we need to describe this uncertainty for a computer.

Bags as a Natural Generalization of Sets Suited for Describing Uncertainty. A precise knowledge can be naturally described in terms of *sets*: Indeed, suppose, e.g., that under certain conditions, an object that we are currently analyzing, can be, in principle, in n different states $s_1, ..., s_n$; for example, the patient's temperature, measured with an accuracy of 1 degree, can take values from, say, $s_1 = 90$, through $s_2 = 91$, $s_3 = 92$, ..., to $s_n = 105$. Let S denote the set of all such states.

If all experts' statements are precise, then each state $s \in S$ is either consistent with these statements, or it is not. So, we can describe this knowledge by describing the set of all states that are consistent with it. For example, if one expert says that 98 is a possible temperature under given conditions, and another says that 99 is possible, and we have no other experts, then we conclude that the set of possible states is $\{98, 99\}$. If a third expert appears who believes that 98 is a possible state, then we can formally add his opinion to our set, thus transforming it into $\{98, 99, 98\}$, but this addition will not change the set itself, because in a set an element either belongs to it or not, so $\{98, 98\} = \{98\}$. Therefore, in our description of the experts' knowledge, we can simply eliminate duplicate elements.

For uncertain knowledge, the situation is different: if two different experts state (with some degree of certainty) that 98 is a possible temperature, then this joint opinion is therefore more reliable than if it was an opinion of only one expert. Hence, in this case, it makes sense, when combining together the opinions of different experts, not to eliminate duplicate elements, because these duplicate elements contain additional information about the experts' degree of belief in different statements. In the above example, when two experts claim 98 to be possible, and one claims 99 to be possible, then the resulting expression $\{98, 99, 98\}$ means that both 98 and 99 are probably possible, but we are more certain about 98 (because two experts said that). Such "sets with repetitions" have been considered in mathematics under the name of *bags* or *multisets* [?].

Bags are actively used not only in representing uncertainty, but also in numerous other areas of computer science (see, e.g., [?, ?, ?]).

How To Represent Bags in a Computer. As we have seen, bags of statements (as opposed to *sets* of statements that represent precise knowledge) are a natural way to describe uncertainty in a knowledge base. From the purely mathematical viewpoint, we can represent a bag as a list of its elements, with all necessary repetitions. However, from the computer viewpoint, it is a waste of space to store several identical statements in the knowledge base. It is quite sufficient to store with each statement, the number of times that this statement occurs in the knowledge base. For example, we can represent a bag $\{98, 99, 98\}$ as $\{98^{|2}, 99^{|1}\}$. For every bag B, the function $\mu_B(s)$ that maps every element $s \in S$ into the number of times that s occurs in B is called a *characteristic function* of the bag. The value $\mu_B(s)$ characterizes the resulting degree of confidence.

This is not a perfect characterization, however. If we simply know that, say, 5 experts believe in a

Department of Computer Science and Department of Electrical and Computer Engineering, University of Texas at El Paso, El Paso, TX 79968. This work was done under the supervision of Professors D. Cooke and V. Kreinovich. The author was partially supported by NSF grant No. CDA 9522903.

certain statement A, then the corresponding degree of confidence will depend on how many experts were asked: if it is 5 out of 5, we can be practically certain that A is true. If it is 5 out of 100, then A is hardly probable. Therefore, to get a better representation of uncertainty, it is natural, instead of the actual *number* $N(A)$ of the experts who believe in A, to consider the *ratio* $d(A) = N(A)/N$, where N is the total number of experts. Thus, we get a degree of certainty that takes values $0, 1/N, 2/N, ..., 1$, from the interval $[0, 1]$.

Logical Operations with Degrees of Belief. Queries often correspond to *logical combinations* of statements from the knowledge base. For example, if a knowledge base contains two statements: "a temperature of 99^o means fever" (we will denote this statement by A) and "if a patient has a fever, he should take an aspirin" (we will denote it by B). In both statements, experts are not 100% certain, so, we have some degrees of certainty $d(A)$ and $d(B)$ stored with these statements.

If a user asks a query "a patient has a temperature 99^o; shall he take an aspirin?", this query can be represented as asking whether the conjunction $A\&B$ is true. Hence, we must somehow compute the degree of certainty $d(A\&B)$ in $A\&B$ based on the degrees of certainty in A and B. We will denote the algorithm that computes $d(A\&B)$ from the inputs $a = d(A)$ and $b = d(B)$ by $f_\&(a, b)$. In these terms, as an answer to the query $A\&B$?, we return $f_\&(d(A), d(B))$.

For statements that can only be true or false, the set of all x for which $A\&B$ is true is simply the *intersection* of the set of all x for which A is true, and the set of all x for which B is true; therefore, in our case, we can say that we want to generalize the intersection operation to bags. Similarly, when a query is equivalent to $A \vee B$, we want to generalize *union* to bags, or, in other terms, describe an operation $f_\vee(a, b)$. When a query is $\neg A$, we must describe a *complement* to a bag, or, in other terms, a function $f_\neg(a)$. There exist several different generalizations of intersection, union, and complement to bags (see, e.g., [?]); they correspond to different operations $f_\&$, $f_\vee$, and $f_\neg$. For example, one can use $f_\&(a, b) = \min(a, b)$, or $f_\&(a, b) = a \cdot b$, etc.

For a general Boolean expression $F(A, ..., B)$, we can define the corresponding operation $f_F(a, ..., b)$ on degrees of certainty if we replace in F, $\&$ by $f_\&$, $\vee$ by $f_\vee$, and $\neg$ by $f_\neg$. For example, for the expression F of the form $A\&(B \vee \neg C)$, we get $f_F(a, b, c) = f_\&(a, f_\vee(b, f_\neg(c)))$. This means that if we know the degrees of certainty $d(A)$, $d(B)$, and $d(C)$ that correspond to the statements A, B, and C, then we take $f_F(d(A), d(B), d(C)) = f_\&(d(A), f_\vee(d(B), f_\neg(d(C))))$ as the degree of certainty of $F(A, B, C)$.

Intervals of Degrees of Belief. To explain why intervals are needed, let us give an example. Suppose 6 out of 10 experts say that a statement A is true. Then, according to our formula, we get $d(A) = 0.6$. To get a better idea of expert's certainty, we may ask more experts, and then, reevaluate $d(A)$. If we ask, say, 100 experts, then it is possible that exactly 60 of them will claim A true, in which case the value $d(A)$ stays exactly the same. However, it can also happen that 61 or 58 experts out of 100 will claim A to be true, in which case, we will have the revaluated value to be 0.61 or 0.58. So, we cannot claim that the initial value of 0.6 was precise, it is rather an *approximation* to the more precise value 0.61 or 0.58. If the initial group of experts was representative, we can assume that the initial approximate value was a correct approximation in the sense that the more precise value is closer to 0.6 than to 0.5 or 0.7. The borderline of values that are closer to 0.6 than to 0.5 is the midpoint 0.55. So, from the fact that 6 experts out of 10 claimed A to be true, we can conclude that the (unknown) more precise value of $d(A)$ must be somewhere on the interval $[0.55, 0.65]$. So, it is natural, to describe the experts' certainty in a statement A, to use not a single number, but an *interval* $\mathbf{d}(A) = [d^-(A), d^+(A)]$.

Now that we have interval values of degree of belief (instead of numbers), the following problem of *computing the value of a Boolean expression with interval inputs* arises:

Given: *A Boolean expression* $F(A, ..., B)$, *intervals* $\mathbf{a}, ..., \mathbf{b}$, *and operations* $f_\&$, $f_\vee$, *and* $f_\neg$.

To compute: *The bounds of the range* $f_F(\mathbf{a}, ..., \mathbf{b})$.

Theorem. *Computing the value of a Boolean expression with interval inputs is NP-hard.*

Idea of the proof. We show that this problem can be reduced to propositional satisfiability problem that is known to be NP-hard: namely, for a formula F, the upper bound of the range $f_F([0, 1], ..., [0, 1])$ is 1 iff F is satisfiable.

References

[1] D. E. Cooke, "A high level computer language based upon ordered multisets", In: *Proceedings of the IEEE Fifth International Conference on Software Engineering and Knowledge Engineering*, San Francisco, June 1993, pp. 117–124.

[2] J. Banâtre and D. L. Métayer, "Programming by multiset transformation", *Communications of the ACM*, 1993, Vol. 36, No. 1, pp. 98–111.

[3] D. Gries and F. B. Schneider, *A logical approach to discrete math*, Springer-Verlag, N.Y., 1993.