NASA-CR-192947

N93-25196

Unclas

G3/63 0159185

(NASA-CR-192947) HOW TO HELP INTELLIGENT SYSTEMS WITH DIFFERENT UNCERTAINTY REPRESENTATIONS COOPERATE WITH EACH OTHER (NASA) 6 p

Cybernetics and Systems: An International Journal, 22:217-222, 1991

NAG 9-432
IN-63-CR
159135
P.6

HOW TO HELP INTELLIGENT SYSTEMS WITH DIFFERENT UNCERTAINTY REPRESENTATIONS COOPERATE WITH EACH OTHER

VLADIK KREINOVICH AND SUNDEEP KUMAR

Computer Science Department, The University of Texas at EL Paso, El Paso, TX USA

In order to solve a complicated problem one must use the knowledge from different domains. Therefore, if we want to automatize the solution of these problems, we have to help the knowledge-based systems that correspond to these domains cooperate, that is, communicate facts and conclusions to each other in the process of decision making. One of the main obstacles to such cooperation is the fact that different intelligent systems use different methods of knowledge acquisition and different methods and formalisms for uncertainty representation. So we need an interface f, "translating" the values x, y, which represent uncertainty of the experts' knowledge in one system, into the values $f(x)$, $f(y)$ appropriate for another one.

In the present report we formulate the problem of designing such an interface as a mathematical problem, and solve it. We show that the interface must be fractionally linear: $f(x) = (ax + b)/(cx + d)$.

WHY IT IS NECESSARY TO COOPERATE

In order to solve complicated problems one must often use the knowledge from different domains. Therefore, if we want to automatize the solution of these problems, we must help the computer-based systems that correspond to these domains cooperate, that is, communicate facts, conclusions, and solutions to each other in the process of decision making.

DIFFERENT UNCERTAINTY REPRESENTATIONS—ONE OF THE MAIN OBSTACLES TO COOPERATION

The cooperating systems must be able to use each other's facts and conclusions. The knowledge itself is normally represented in knowledge bases in a more or less standard form, close to the language of mathematical logic. However, working intelligent systems use several different methods (see, e.g., a survey by Smets et al., 1988) of representing uncertainty of the corresponding statements: fuzzy logic, certainty values formalism, probabilistic reasoning, Dempster-Shafer formalism, etc. All these methods are based on representing the experts' uncertainty by real numbers—usually from the interval [0, 1]. The value 1 assigned to some statement S means that an expert is absolutely sure about this statement, and the value of 0 that the expert is absolutely sure that S is false (some systems, such as MYCIN [Shortliffe, 1976], use different intervals, like $[-1, 1]$ instead of [0, 1]. However, the operations with these values are essentially different: for example, if to A and B we assign the values a and b, then to $A \vee B$ we assign $\max(a, b)$ in fuzzy logic, $a + b - ab$ in probabilistic reasoning, and a complicated expression in MYCIN. This choice of operations is vitally important (and depends on the domain): in MYCIN, for example, it turned out that diagnostic efficiency essentially decreases if we change the formulas for operating with uncertainty (Shortliffe, 1976). Therefore we cannot simply place the statements from one knowledge base into another and derive conclusions: we need some *interface* allowing us to transform the uncertainty values from one knowledge base into another.

Moreover, even if the knowledge bases use the same formalism for dealing with the certainty values of their statements, they can be based on different knowledge acquisition algorithms, so the same level of the expert's uncertainty, corresponding to the same word like "for sure," can be represented by essentially different numbers in these bases. If we, for example, transfer to a knowledge base in which 0.7 means "for sure" a statement from another knowledge base in which the same value 0.7 means "maybe," we make this first system mistakenly understand this external statement as highly realiable, whereas in reality it is only hypothetic. Clearly we need an interface to transform certainty values.

THE CHOICE OF THE UNCERTAINTY-TRANSFORMING INTERFACE AS A MATHEMATICAL PROBLEM: SEMIFORMAL MOTIVATION

The desired interface should be a program (or, in mathematical terms, a function) transforming the certainty value of a statement in one system into a value appropriate for some other system.

Because there are many different types of uncertainty representations, it is reasonable to construct a universal interface that can be adjusted to arbitrary pair by an appropriate choice of its parameters. In order to represent this fact mathematically let us denote the number of parameters by n. For every choice of these parameters we have a function $f(x)$ transforming real values into real values, so the whole universal interface corresponds to an n-dimensional family of real-valued functions of real argument (n-dimensional means that there are n parameters, by fixing which we determine the function uniquely).

In order to construct such a family F let us enumerate the properties that we want to be fulfilled.

(1) If $x \to f(x)$ is an efficient transformation of uncertainty values from a system A to some other system B, and $y \to g(y)$ is an efficient transformation from B to some C, then if we want to translate from A directly to C, the result of translating x will be $g(f(x))$. Therefore it is necessary to demand that this composition function $x \to g(f(x))$ belong to the same family of functions (i.e., can be obtained from the universal interface program by fixing some parameters). In mathematical terms this means that the desired family F must be closed under composition.

(2) Suppose $x \to f(x)$ is an efficient transformation from A to B. Then if we want to transform the uncertainty values from B to A, to every value y in B we must put into correspondence a value x in A such that $f(x) = y$.

This correspondence $y \to x$ is called in mathematics an inverse function to f. So the next demand is that for every function f from F the inverse function must also belong to F.

These two demands mean that the family F must be closed under composition and inverse function operation, or, in mathematical terms, that F must be a *group* with respect to composition. Such groups are called *transformation groups.*

(3) For some pairs of acquisition procedures we can calculate the best interface transformation. In these cases it is necessary to demand that the desired family F contain these transformations. For example, a natural measure of the experts' uncertainty in some statement S is the percentage $p(S) = n(s)/N$ of those experts who think that S is true. If all the experts believe in S, this value is 1 (= 100%), if half, it is 0.5 (50%) etc.

Knowledge engineers want the system to include the knowledge of the whole scientific community, so they ask as many experts as possible. But asking too many experts leads to the following negative phenomenon: in the presence of the most respected professors—Nobel Prize winners and the like—some less self-confident experts are not sufficiently brave to express their own opinions, so they will either say nothing or follow the opinion of the majority.

How does their presence influence the percentage values? Let us denote by N the initial number of experts, by $n(S)$ the number of those who believe in S, and by M the number of shy experts added. Initially, $p = n(s)/N$. After we add to our experts sample M experts who do not answer anything when asked about S the number of experts who believe in S is still $n(S)$, so the ratio expressing the uncertainty values is now $p' = n(S)/(N + M) = cp$, where $c = N/(M + N)$. When we added the experts who give the same answers as the majority of N renowned experts, we get $n(S) + M$ experts saying that S is true, so the uncertainty value is now $p'' = (n(S) + M)/(N + M) = (Np + M)/(N + M)$. If we need an interface transforming the values obtained without the new experts into the values obtained with them, then the corresponding transformation will be a linear function $p \to ap + b$, where a and b are constants (independent of p). If we add M "silent" experts and M' "conformists" (who vote as the majority), then we get a two-parametric (in mathematical terms two-dimensional) family of linear functions $p \to ap + b$.

So the desired family F must contain a two-parametric family of linear functions. We arrive at the following mathematical problem.

THE CHOICE OF INTERFACES: FORMULATION AND SOLUTION OF THE MATHEMATICAL PROBLEM

Formulation. Find a finite-dimensional group of transformation $R \to R$ (i.e., real-values functions of real argument) that includes a two-parametric family of linear functions.

Comment on the Solution. The problem of classifying all possible transformation groups of an n-dimensional space R^n, $n = 1,2,3,\ldots$, that include a sufficiently big family of linear transformations, was first formulated by Norbert Wiener (see, e.g., Wiener, 1962). His hypothesis was confirmed in Guillemin and Sternberg (1964) and Singer and Sternberg (1965). It turned out that for $n = 1$ the only possible groups are the group L of all linear transformations and the group FL of all fractionally linear transformations $x \to (ax + b)/(cx + d)$ (the simplified proof for the one-dimensional case is given in Kreinovich, 1987; see also Kreinovich, 1990). In general, the relationship between the uncertainty values assigned to one and the same word in different acquisition procedures (and thus in different knowledge-based systems) cannot be expressed by a linear function, so the desired family of the interface transformations is *FL*.

Solution. The desired universal interface must implement fractionally-linear functions $x \to (ax + b)/(cx + d)$ to transform uncertainty values of one knowledge base into the values appropriate for some other system, where the parameters a, b. c, d must be adjusted to these two systems.

Comment. The concrete values of a–d can be obtained, for example, as follows: take four arbitrary statements S_1, S_2, S_3, S_4, estimate their uncertainty values in the first intelligent system (by means of the procedure used to create that system), and obtain p_1–p_4; then do the same with the second system, obtaining q_1–q_4, and then estimate a–d by solving the system of four equations

$$q_i = \frac{ap_i + b}{cp_i + d} \qquad i = 1,2,3,4$$

with four unknowns a–d. This system can be reduced to a linear one, if we multiply both sides by the denominator:

$$(cp_i + d)\, q_i = ap_i + b.$$

APPLICATIONS

One of the most widespread methods of solving complicated problems is that of computer simulation. This is practically the only way to get predictions about global ecology, complicated transportation systems, military conflicts, and the line. If all our knowledge is already expressed in probabilistic terms, we can apply Monte Carlo methods. But normally the essential part of our knowledge is expressed in fuzzy terms, so that the corresponding knowledge bases use nonprobabilistic formalisms for knowledge representation. How to use this additional knowledge? If we simply interpret the uncertainty values as probabilities and apply Monte Carlo methods, the results are often far from reality (and always not justified, hence not convincing).

Our results show that in this case we must apply an appropriate fractionally linear transformation. Application of this idea to the computer simulation of an automated manufacturing unit has allowed us to make essentially better predictions (Kozlenko and Kreinovich, 1986).

20-5

ACKNOWLEDGMENTS

One of the authors (V. K.) is grateful to Y. M. Eliashberg and P. Suppes (Stanford): I. B. Frenkel, A. Mostow, and G. Zuckerman (Yale): S. Sternberg (Harvard); and Y. Gurevich (Ann Arbor, Michigan) for valuable discussions and consultations. Part of this work was supported by the Soviet Science Foundation grant #0107 4346 10.

REFERENCES

Guillemin, V. M. and S. Sternberg. An Algebraic Model of Transitive Differential Geometry. *Bull. Amer. Math. Soc.*, *70*(1), 16–47, 1964.

Kozlenko, V. Y. and V. Y. Kreinovich. Designing an Adaptive Robototechnical Module for Highly Reliable Flexible Automatic Industrial Systems. Leningrad, Technical Report on Grant # 0107 4346 10, 1986.

Kreinovich, V. Y. A Mathematical Supplement to the Paper: Krotkov, I. N.: Kreinovich, V. Y.; Mazin V. D. A General Formula for the Measurement Transformations, Allowing the Numerical Methods of Analyzing the Measuring and Computational Systems. Measurement Techniques, 10, 8–10, 1987.

———. Arbitrary Nonlinearity is Sufficient to Represent All Functions by Neural Networks: A Theorem. University of Texas at El Paso, Computer Science Department, Technical Report UTEP-CS-90-11, June 1990.

Shortliffe, E. H. *Computer-based Medical Consultation: MYCIN.* Elsevier, New York: 1976.

Singer, I. M. and S. Sternberg. The Infinite Group of Lie and Cartan, Part 1. *Journal d'Analyse Mathematique, 15,* 1–113, 1965.

Smets, P., et al. *Nonstandard Logics for Automated Reasoning.* Academic Press, London, 1988.

Wiener, N. *Cybernetics, or Control and Communication in the Animal and the Machine.* MIT Press, Cambridge, Massachusetts, 1962.

Requests for reprints should be sent to Vladik Kreinovich, Computer Science Department, The University of Texas at El Paso, El Paso, TX 79968-0518 USA.

20-6