MAXIMUM ENTROPY (MAXENT) METHOD IN EXPERT SYSTEMS AND INTELLIGENT CONTROL: NEW POSSIBILITIES AND LIMITATIONS

V. KREINOVICH
Department of Computer Science
University of Texas at El Paso
El Paso, TX 79968, USA[§]

AND

H. T. NGUYEN AND E. A. WALKER
Department of Mathematical Sciences
New Mexico State University
Las Cruces, NM 88003, USA[¶] [‖]

Abstract. To describe uncertainty of experts' statements $E_1, ..., E_n$ that form a knowledge base, it is natural to use (subjective) probabilities p_i. Correspondingly, it is natural to use probabilities to describe uncertainty of the system's answer to a given query Q. Since it is impossible to inquire about the expert's probabilities for all possible ($\geq 2^n$) propositional combinations of E_i, a knowledge base is usually *incomplete* in the sense that there are many probability distributions consistent with this knowledge. If we want to return a single probability, we must select *one* of these distributions. MaxEnt is a natural selection rule, but for expert systems, computing the MaxEnt distribution often takes an unrealistically long time.

In this paper, we propose computationally feasible approximate MaxEnt techniques for expert systems and intelligent control.

Key words: MaxEnt, expert systems, uncertainty, commonsense reasoning, intelligent control

[§]Email: vladik@cs.utep.edu. This work was partially supported by NSF Grants No. CDA-9015006 and EEC-9322370, and by NASA Grant No. NAG 9-757.

[¶]Email: {hunguyen,elbert}@nmsu.edu. This work was partially carried out while Hung T. Nguyen was on sabbatical leave at the University of Southern California, Los Angeles, Spring 1995.

[‖]The authors are thankful to Peter C. Cheeseman for his interest and encouragement, and to all the participants of the workshop, especially to Professor Myron Tribus, for helpful discussions.

1. Expert Systems

1.1. INTRODUCTION: WHAT ARE EXPERT SYSTEMS, WHY DO WE NEED MAXENT FOR EXPERT SYSTEMS, AND WHY TRADITIONAL MAXENT METHODS ARE NOT ALWAYS SUFFICIENT

The Basic Idea of an Expert System. In many fields (geology, medicine, etc.), there are experts who can make outstanding decisions. Unfortunately, the number of these super-experts is usually small, and we cannot use them for every problem we encounter: e.g., we cannot show all the patients with heart disease to the greatest specialist.

A natural idea is, therefore, to design a computer-based system that would use the knowledge of these experts and make conclusions and decisions based on that knowledge. Such systems need not be completely autonomous. If we can design a system that will prompt one or several different answers, this system will also be of great value to the corresponding community. Such helper systems are called *expert systems* .

To make expert systems successful, we must be able to incorporate the experts' knowledge inside the computer: First of all, we need to describe statements $E_1, ..., E_n$ that comprise the experts' knowledge. Then, if a user asks a query Q, the system must be able to return a "yes" or "no" answer.

To Describe Experts' Knowledge Adequately, We Must Also Describe Uncertainty. Our knowledge is not simply the set of statements that we make. In some of these statements, we are more certain; in some, we are less certain. A natural measure of an expert's certainty in a statement E_i is a (subjective) probability p_i. Therefore, in a knowledge base, we must have not only the experts' statements $E_1, ..., E_n$, but also the probabilities p_i of these statements. This uncertainty changes the desired answer to a query Q: since the experts are not absolutely certain about the statements E_i that comprise the knowledge base, we cannot be absolute certain about the answer to Q deduced from these statements. In other words, due to uncertainty in E_i, we are uncertain about Q. As a result, we want the expert system to return the degree of uncertainty (i.e., the subjective probability) of a queried statement Q.

A Problem: Our Information about Experts' Probabilities is Not Complete. A typical knowledge base, that consists of experts statements $E_1, ..., E_n$ with their probabilities $p_1, ..., p_n$, does not contain the complete information about the expert's probabilities. For example, if the only thing that we know about the two statements E_1 and E_2 is that both have probability 0.5, then:

- it could happen that E_1 and E_2 are equivalent, in which case $E_1 \& E_2$ has probability 0.5, and
- it could also happen that E_1 and E_2 are incompatible, in which case $E_1 \& E_2$ has probability 0.

To get a complete description of subjective probabilities, it is thus not sufficient to know n probabilities p_i of the statements E_i themselves; we also need to know at least 2^n probabilities of the possible propositional combinations of E_i (e.g., probabilities of 2^n statements of the type $E_1\&\neg E_2\&...\&E_n$ that correspond to different combinations of basic statements E_i and their negations). Realistic knowledge bases contain hundreds of statements; it is absolutely impossible to present $2^{100} \approx 10^{30}$ questions to the experts. Hence, no matter how many questions we ask, the probabilistic information contained in the knowledge base will not be complete.

One Possible Solution: Return Intervals of Probabilities As Answers to the Queries. On possible solution to this problem is as follows: There are many probability distributions that are consistent with our knowledge; different distributions may lead to different probabilities $p(Q)$ of the query. Hence, let us provide the user with the *set* of all possible values of $p(Q)$. (Usually, this set is an *interval.*)

In the above pedagogical example, the interval of possible values of $p(E_1\&E_2)$ is $[0, 0.5]$.

There are several successful expert systems that are based on this idea (see, e.g., Kohout et al. [5] and references therein).

A Problem with Interval Approach. In general, the resulting interval of probability is too wide. In many non-trivial cases, it is so close to the interval $[0, 1]$ (that corresponds to our not knowing anything about $p(Q)$) that this interval becomes absolutely useless to the user who asked this query: e.g., if a doctor is told that the probability of a successful surgery is from the interval $[0.1, 0.9]$, this will not help her decide whether to perform this surgery or not.

MaxEnt Approach. A MaxEnt approach to this problem was proposed by P. Cheeseman (see, e.g., [1]): he proposed to choose, from all probability distributions that are consistent with the given knowledge, the one p with the largest entropy, and for every query Q, return $p(Q)$ (for this chosen p) as a probability that Q is true.

Formally, we can define a *possible world* W as a consistent statement of the type $E_1\&\neg E_2\&...\&E_n$, and choose the *MaxEnt* probability distribution $\{p(W)\}$ so that $S = -\sum p(W) \log p(W) \to \max$ under the following three conditions:

- that the probabilities $p(W)$ are non-negative: $p(W) \geq 0$;
- that the total probability is 1: $\sum p(W) = 1$; and
- that the probability of each statement E_i is equal to the given value p_i, i.e., that $\sum\{p(W) \mid W \vdash E_i\} = p_i$.

Then, for every query Q, we define $p(Q)$ as $p(Q) = \sum\{p(W) \mid W \vdash Q\}$.

Main Problem: MaxEnt Approach Is Often Computationally Intractable. In many cases, MaxEnt approach has lead to successful expert systems (see, e.g., Pearl [11]), but in general, it can be proven to be computationally intractable (see, e.g., Dantsin et al. [2]).

There exist reasonably good techniques that approximate the value $p(Q)$ for the MaxEnt distribution p. One of these techniques is a Monte-Carlo method (see, e.g., [2,7]) that estimates $p(Q)$ by repeatedly asking a query Q to a randomly perturbed knowledge base. To get the probability $p(Q)$ with the desired accuracy $\approx 10\%$, we must repeat this simulation at least 100 times. For a realistic knowledge base, an answer to a query already takes reasonably long (often, around several minutes), so, if we must increase it by a factor of hundred, it becomes unrealistically long.

1.2. METHODS USED IN THE EXISTING EXPERT SYSTEMS

Description of the Existing Methods. MaxEnt approach is not always computationally feasible for expert systems; on the other hand, expert systems do exist, and they are often very successful. How do they do that?

The existing expert systems use a *heuristic* approach. To explain this approach, let us describe how it is applied to the problem with which we started: we know the probabilities p_1 and p_2 of the two statements E_1 and E_2, and we want to know the probability $p(E_1\&E_2)$. MaxEnt approach is *global* in the sense that to determine this probability, we must know (and take into consideration) the information about all other statements E_i. This necessity of processing all the statements from a knowledge base of size n is what makes the computation time grow exponentially with n. Hence, a natural way to get an easily computable approximation to $p(E_1\&E_2)$ is to use only the values $p(E_1)$ $(= p_1)$ and $p(E_2)$. In other words, to estimate the probability of $E_1\&E_2$, a function $f_\& : [0,1]\times[0,1] \to [0,1]$ is chosen, and $p(E_1\&E_2)$ is estimated as $f_\&(p(E_1), p(E_2))$ (examples of such $\&-$*operations* are $f_\& = \min$, $f_\&(a,b) = a \cdot b$, etc.). Similarly, we estimate $p(E_1 \vee E_2)$ as $f_\vee(p(E_1), p(E_2))$ for some $\vee-$*operation* $f_\vee$.

If a query Q is more complicated than a disjunction or a conjunction of basic statements, then we express Q in terms of &, $\vee$, and $\neg$, and use this representation to estimate $p(Q)$ step-by-step: e.g., we estimate $p((E_1\&(E_2 \vee \neg E_1))$ as $f_\&(p(E_1), f_\vee(p(E_2), 1 - p(E_1))$.

This approach is efficiently used in the existing expert systems, but it has the following two problems:

First Problem with the Existing Approach. The first problem is that for complicated queries, the result of the step-by-step application depends on the representation of the query. For example, a query $E_1 \to E_2$ can be represented either as $\neg E_1 \vee E_2$, or as $\neg E_1 \vee (E_2\&E_1)$. In classical logic, these two representations are equivalent. However, step-by-step procedure will lead to different results: e.g., for $p_1 = p_2 = 0.5$, $f_\&(a,b) = a\cdot b$, and $f_\vee(a,b) = a+b-a\cdot b$, we get $p(\neg E_1 \vee E_2) = 0.5+0.5-0.25 = 0.75$, $p(E_2\&E_1) = 0.25$, and $p(\neg E_1 \vee (E_2\&E_1)) = 0.5+0.25-0.125 = 0.625 \neq 0.75$. This problem was recently emphasized by Elkan [3].

Second Problem. In the existing expert systems, , the $\&-$operation $f_\&$ and the $\vee-$operations $f_\vee$ are chosen rather arbitrarily.

1.3. THE PROPOSED APPROACH

The Main Idea. We propose to combine the advantages of MaxEnt (whose formulas are justified and unambiguous) and of the existing approach (whose formulas are computable): Namely, we propose to use the functions $f_\&$ and $f_\vee$, but choose them by using MaxEnt.

Definitions and the Main Result. In precise terms, as $f_\&(a,b)$, we choose $p(A\&B)$ for a distribution p that has the largest entropy among all distributions for which $p(A) = a$ and $p(B) = b$. Similarly, we can define $f_\vee(a,b)$ as $p(A \vee B)$ for the

MaxEnt distribution p. We will call the resulting operations $f_{\&}$ and $f_{\vee}$ *MaxEnt operations.*

To find $f_{\&}(a,b)$ and $f_{\vee}(a,b)$, we thus have to solve a conditional optimization problem with four unknowns $p(W)$ (for $W = A\&B$, $A\&\neg B$, $\neg A\&B$, and $\neg A\&\neg B$). This problem has the following explicit solution:

Theorem 1. *For & and $\vee$, the MaxEnt operations are $f_{\&}(a,b) = a \cdot b$ and $f_{\vee}(a,b) = a + b - a \cdot b$.*

Comment. For a modified entropy function, a similar problem is solved in Klir et al. [4]; it leads to a different set of &− and ∨−operations. Both pairs of operations are in good accordance with the general group-theoretic approach [6,8,9] for describing operations $f_{\&}$ and $f_{\vee}$ which can be optimal w.r.t. different criteria.

The Proposed Approach Is a Solution to the Above Two Problems of Traditional Expert System Methodology. First, we have a *justification* for the chosen $f_{\&}$ and $f_{\vee}$ in terms of MaxEnt.

Second, we can repeat the same procedure for any propositional formula $F(A,...,B)$ (and not only for $A\&B$ and $A \vee B$), and come up with a function $f_F(a,...,b)$; therefore, we do not need to represent F in terms of &, ∨, and ¬ any more: we get f_F straight from F. Let us describe this general formula:

Theorem 2. *Let $F \equiv \vee(A_1^{\varepsilon_1}\&...\&A_n^{\varepsilon_n})$ be a conjunctive normal form (CNF) for F, where $\varepsilon_i = \pm$, A^+ means A, and A^- means $\neg A$. Then, $f_F(a_1,...,a_n) = \sum a_1^{\varepsilon_1} \cdot ... \cdot a_n^{\varepsilon_n}$, where $a^+ = a$ and $a^- = 1 - a$.*

In particular, we get the following two results:

Theorem 3. *For $A \to B$, the corresponding MaxEnt operation is $f_{\to}(a,b) = 1 - a + a \cdot b$.*

Proof. Indeed, CNF for $a \to b$ is $(a\&b)\vee(\neg a\&b)\vee(\neg a\&\neg b)$; therefore, $f_{\to}(a,b) = a \cdot b + (1-a) \cdot b + (1-a) \cdot (1-b) = ab + b - ab + 1 - a - b + ab = 1 - a + ab$. Q.E.D.

This result coincides with the result of a step-by-step application of MaxEnt operations $f_{\&}$ and $f_{\vee}$ to the formula $B \vee \neg A$ (which is a representation of $A \to B$ in terms of &, ∨, and ¬).

Theorem 4. *For $A \equiv B$, the corresponding MaxEnt operation is $f_{\equiv}(a,b) = 1 - a - b + 2a \cdot b$.*

Unlike $f_{\to}$, the resulting expression *does not* cannot be obtained by a step-by-step application of $f_{\&}$, $f_{\vee}$, and $f\neg$ to any propositional formula:

Theorem 5. *The expression $f_{\equiv}(a,b) = 1 - a - b + 2a \cdot b$ cannot be obtained by a step-by-step application of MaxEnt operations $f_{\&}(a,b) = a \cdot b$, $f_{\vee}(a,b) = a + b - a \cdot b$, and $f_{\neg}(a) = 1 - a$ to a propositional formula $F(A,B)$.*

Idea of the proof. Let us prove this theorem by reduction to a contradiction. Let us assume that there exists a propositional formula F for which the step-by-step procedure leads to $f_{\equiv}$.

One can easily see that $f_{\vee}(p,q) = f_{\neg}(f_{\&}(f_{\neg}(p), f_{\neg}(q)))$ for all p and q, so, we can replace in F every occurrence $P \vee Q$ of $\vee$ by $\neg(\neg P\&\neg Q)$ and still get the same step-by-step result. After doing this, we get a new formula G that only contains & and ¬ and still leads to $f_{\equiv}$.

If G is of the form $\neg\neg H$, then, since $f_\neg(p) = 1-p$, we have $f_G = 1-(1-f_H) = f_H$, so, we can simply take H as the desired formula.

As a result of this reduction, we get a formula H that represents $f_\equiv$ and that is either of the type $P\&Q$ for some subformulas P and Q, or of the type $\neg(P\&Q)$. Since both operations $f_\&$ and $f_\neg$ are polynomial, the expressions f_P and f_Q that we get after applying these operations step-by-step are also polynomials. Hence:

- In the first case, the function $f_\equiv(a,b)$ is a product of two polynomials $f_P(a,b)$ and $f_Q(a,b)$; however, one can easily show that the polynomial $f_\equiv(a,b)$ is not factorizable.
- In the second case, the function $1 - f_\equiv(a,b)$ is a product of two polynomials f_P and f_Q; however, this polynomial $1 - f_\equiv$ is also not factorizable.

In both cases, we have a contradiction, so, our initial assumption was wrong. Q.E.D.

Comment 1: Relationship to Common Sense Reasoning. Theorem 4 is in good accordance with *common sense reasoning* (that expert systems try to formalize): Indeed, mathematically, "A implies B" means the same as "B or not A", but the resulting formulas of the type "if $2+2=5$, then I am the King of France" are counterintuitive. From the common sense viewpoint, implication is a separate operation that is different from its mathematical representations in terms of &, $\vee$, and $\neg$. So, we can conclude that *the described MaxEnt approach for choosing "logical" operations with probabilities is closer to common sense reasoning than the traditional expert system approach.*

Comment 2: There Still is a Basis for Propositional Functions. In classical logic, three connectives &, $\vee$, and $\neg$ form a *basis* in the sense that every propositional formula can be expressed in terms of them. The fact that in MaxEnt, we cannot express $\equiv$ in terms of &, $\vee$, and $\neg$, simply means that, unlike classical logic, there connectives do not form a basis for "MaxEnt" probabilistic logic. This does not mean that this new "logic" does not have a finite basis: due to Theorem 2, we can express every function $f_F(a,...,b)$ in terms of &, $\neg$, and a special partially defined operation $\dot\vee$ ("disjoint union") for which $f_{\dot\vee}(a,b) = a+b$.

Comment 3: MaxEnt Can Also Handle Additional Interval Uncertainty. In practice, for statements E_i from the knowledge base, we often know only approximate values of the probabilities p_i, e.g., *intervals* $[p_i^-, p_i^+]$ of their possible values. In this case, we can first select the probabilities p_i from MaxEnt.

This additional uncertainty is often described by a Dempster-Shafer formalism. For this case, MaxEnt approach is also helpful (see, e.g., Nguyen Walker [10]).

2. Intelligent Control

The Main Problem. If we use an expert system to describe a real-valued variable A, then, for each possible value u of this variable, we can compute the probability $p_A(u)$ that the value A is consistent with the experts' knowledge. Based on this information, a specialist makes a decision (e.g., a doctor chooses whether to operate or not).

If we want an expert system to be used in *control* (e.g., in controlling a spaceship), then, for each possible control value u, we can also get the probability $p(u)$ that u is reasonable, but for control, we often do not have time to ask a specialist. The system must make a decision automatically, all by itself. Such knowledge-based automatic control systems are called *intelligent control* systems. For intelligent control systems, we arrive at the following problem: How can we translate the probabilities $p(u)$ into a single control value $\bar{u}$?

MaxEnt Helps. These probabilities describe the (unknown) set U of control values that are consistent with our knowledge. Therefore, we have a *probability measure* (*random set*) $P(U)$ on the class of all such sets U. Probabilities $p(u)$ are then interpreted as $P(u \in U)$. These probabilities do not determine $P(U)$ uniquely, so, we have to use MaxEnt. MaxEnt leads to the measure in which $u \in U$ and $u' \in U$ are independent events for $u \neq u'$.

Even if the set U of possible values is fixed, we still do not know the probabilities of different values of $u \in U$. To get a reasonable probability distribution $P(u|U)$ on U, we can use MaxEnt again. (E.g., for finite U, we get each $u \in U$ with equal probability.) As a result, we can find the probability $p_{\rm act}(u)$ of u being actually the best control as $\sum_U P(U)P(u|U)$ (in the finite case, and $E_U(p(u|U))$ in the infinite case). Now, for each value u, we know the probability of this u being the best, so, we can choose $\bar{u}$ as the mathematical expectation of u w.r.t. this distribution. These definitions are easily formalized in a finite case (when u only takes finitely many possible values $u_1, ..., u_m$), and can be extended to $u \in R$ by a limit $m \to \infty$.

How to compute this $\bar{u}$? Due to lack of space, we skip the technical details, and only present the following result:

Result: *For a continuous function* $p(u)$, $\bar{u} = \int u p(u)\,du / \int p(u)\,du$.

Idea of the proof: For the finite approximation, when we have N points u_i per unit length, for each choice of U, the average of u is simply $\sum u/\#u$. Since p is continuous, it can be approximated (with arbitrary accuracy) by piecewise-constant functions. On each interval of constancy $I = [\tilde{u} - \Delta u/2, \tilde{u} + \Delta u/2]$, of length Δu, the probability of $u \in I$ to be chosen is the same for all u, and the events of choosing or not choosing two different elements are statistically independent. Totally, we have $\approx N\Delta u$ points u_i on I; therefore, due to large numbers theorem, $\approx p \cdot \Delta u \cdot N$ points will be chosen, and the ratio of the number of chosen points to $\Delta u \cdot N$ tends to p. Since all points $\in I$ are approximately equal to $\tilde{u}$, the total contribution of points from I to $\sum u$ is $\approx \tilde{u} \cdot p(\tilde{u}) \cdot \Delta u \cdot N$, and the total contribution to $\#u$ is $\approx p(\tilde{u}) \cdot \Delta u \cdot N$. Thus, $\sum u \approx N[\sum \tilde{u} \cdot p(\tilde{u}) \cdot \Delta u]$, and $\#u \approx N[\sum p(\tilde{u}) \cdot \Delta u]$. The resulting approximating expressions do not depend on U at all. In the limit, when $N \to \infty$ and $\Delta u \to 0$, both sums (divided by N) turn into the desired integrals. Q.E.D.

The above formula for $\bar{u}$ is actively used in intelligent control (see, e.g., [9]). Thus, *MaxEnt justifies one of the basic formulas used in intelligent control.*

Choice of $\&-$ *and* $\vee-$*Operations Revisited.* Since for intelligent control, all we are interested in is u, it makes sense to choose $f_\&$ and $f_\vee$ from the condition that the entropy of the resulting distribution for u is the largest possible. This control-oriented approximating use of MaxEnt leads to different results than the one used for expert systems: namely, the corresponding MaxEnt operations are $f_{\&}(a,b)=\min(a,b)$ and $f_{\vee}(a,b)=\min(a+b,1)$ (for proofs, see [12,13]).

These operations also have a direct control sense: e.g., in many control problems (e.g., in tracking the Space Shuttle), we want the control that leads to the largest *stability* in the sense that if we deviate a little bit from the desired trajectory, we want to guarantee the fastest possible return. It turns out that the above-described MaxEnt operations are exactly the ones that lead to the maximally stable control [9,14]. Thus, *MaxEnt leads to the maximum stability of the controlled system.*

MinEnt? Stability is not the only desired property of control. The most stable system operates very fast. In many control problems, e.g., in docking the Space Shuttle to a space station, such a speed may be dangerous: we can damage the station if we hit it fast. In such cases, we want the *smoothest* possible control. It turns out [9,14] that the smoothest control is attained when we choose the operations $f_{\&}(a,b)=ab$ and $f_{\vee}(a,b)=\min(a,b)$. These operations turn out to correspond to the ... *minimal* entropy [12,13].

References

1. P. Cheeseman, "In Defense of Probability", in *Proceedings of the 8-th International Joint Conference on AI*, pp. 1002–1009, Los Angeles, CA, 1985.
2. E. Y. Dantsin and V. Kreinovich, "Probabilistic inference in prediction systems," *Soviet Mathematics Doklady*, 1990, **40**(1), pp. 8–12, 1990.
3. C. Elkan, "The paradoxical success of fuzzy logic", in *Proceedings of AAAI-93, American Association for Artificial Intelligence 1993 Conference*, pp. 698–703.
4. G. J. Klir and Bo Yuan, *Fuzzy Sets and Fuzzy Logic*, Prentice Hall, NJ, 1995.
5. L. J. Kohout and W. Bandler, "Fuzzy Interval Inference Utilizing the Checklist Paradigm and BK-Relational Products", pp. 289–334, in R. B. Kearfott and V. Kreinovich, eds., *Applications of Interval Computations*, Kluwer, Boston, MA, 1995 (to appear).
6. V. Kreinovich, "Group-theoretic approach to intractable problems," *Lecture Notes in Computer Science* (Springer-Verlag, Berlin), **417**, pp. 112–121, 1990.
7. V. Kreinovich et al., "Monte-Carlo methods make Dempster-Shafer formalism feasible," in *Advances in the Dempster-Shafer Theory of Evidence*, R. R. Yager et al., eds., pp. 175–191, Wiley, N.Y., 1994.
8. V. Kreinovich and S. Kumar. "Optimal choice of &- and ∨-operations for expert values," in *Proceedings of the 3rd University of New Brunswick Artificial Intelligence Workshop*, pp. 169–178, Fredericton, N.B., Canada, 1990.
9. V. Kreinovich et al., "What non-linearity to choose? Mathematical foundations of fuzzy control", in *Proceedings of the 1992 International Conference on Fuzzy Systems and Intelligent Control*, Louisville, KY, 1992, pp. 349–412.
10. H. T. Nguyen and E. A. Walker, "On decision making using belief functions", in *Advances in the Dempster-Shafer Theory of Evidence*, R. R. Yager, J. Kacprzyk, and M. Pedrizzi, eds., pp. 311–330, Wiley, N.Y., 1994.
11. J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, San Mateo, CA, 1988.
12. A. Ramer and V. Kreinovich, "Maximum entropy approach to fuzzy control", *Information Sciences*, **81**(3–4), pp. 235–260, 1994.
13. A. Ramer and V. Kreinovich, "Information complexity and fuzzy control", in *Fuzzy Control Systems*, A. Kandel and G. Langholtz, eds., pp. 75–97, CRC Press, Boca Raton, FL, 1994,
14. M. H. Smith and V. Kreinovich. "Optimal strategy of switching reasoning methods in fuzzy control", in *Theoretical aspects of fuzzy control*, H. T. Nguyen, M. Sugeno, R. Tong, and R. Yager, eds., pp. 117–146, J. Wiley, N.Y., 1995.