Maximum Entropy as a Feasible Way to Describe Joint Distribution in Expert Systems

Thongchai Dumrongpokaphan[†], **Vladik Kreinovich**[‡,1], **and**

Hung T. Nguyen[♭]

[†]Department of Mathematics, Faculty of Science
Chiang Mai University, Chiang Mai 50200, Thailand
e-mail : tcd43@hotmail.com

[‡]Department of Computer Science, University of Texas at El Paso
El Paso, TX 79968, USA
e-mail : vladik@utep.edu

[♭]Department of Mathematical Sciences, New Mexico State University
Las Cruces, NM 88003, USA, and
Faculty of Economics, Chiang Mai University
Chiang Mai 50200, Thailand
e-mail : hunguyen@nmsu.edu

Abstract : In expert systems, we elicit the probabilities of different statements from the experts. However, to adequately use the expert system, we also need to know the probabilities of different propositional combinations of the experts' statements – i.e., we need to know the corresponding joint distribution. The problem is that there are exponentially many such combinations, and it is not practically possible to elicit all their probabilities from the experts. So, we need to estimate this joint distribution based on the available information. For this

purpose, many practitioners use heuristic approaches – e.g., the t-norm approach of fuzzy logic. However, this is a particular case of a situation for which the maximum entropy approach has been invented, so why not use the maximum entropy approach? The problem is that in this case, the usual formulation of the maximum entropy approach requires maximizing a function with exponentially many unknowns – a task which is, in general, not practically feasible. In this paper, we show that in many reasonable example, the corresponding maximum entropy problem can be reduced to an equivalent problem with a much smaller (and feasible) number of unknowns – a problem which is, therefore, much easier to solve.

---

1 Formulation of the Problem: Maximum Entropy Is Needed, but the Corresponding Optimization Problem Does Not Look Feasible

Need for expert systems. In many application areas, ranging from finance to medicine to geosciences to law enforcement, a large portion of knowledge and skills comes from experts.

Some of these experts are more skilled, some are somewhat less experienced and less skilled. In the ideal world, everyone should have their finances managed by the best financial consultants and their illnesses treated by the world's best doctors. In the ideal world, all the crimes should be investigated by the world's best detectives. Unfortunately, there are few world's best financial consultants, doctors, etc., and they do not have enough time to help everyone.

A natural idea is thus to incorporate the knowledge of the best experts in a computer-based system, so that other experts – and maybe even non-expert users – can use the knowledge of the top experts. Such systems are known as *expert systems.*

Need to take uncertainty into account. Expert statements are rarely absolutely certain. For most of their statements, experts understand that the corresponding conclusion is valid only with a certain probability. For example, in finance, even a top expert cannot always guarantee that his or her investment advice will always lead to good gains, there is always a possibility that the investment will fail. In general, the difference between the top expert and a regular expert is not that the top expert guarantees the outcomes – no one can do that – but that for the rules formulated by the top expert, probability of success is higher. Similarly, even a top surgeon sometimes fails – but for the top surgeon, the probability of success is higher than what would have been achieved by a regular surgeon in a similar situation.

To make proper recommendations, it is therefore important not only to describe the expert's statements rules in the expert system, but also to describe the probabilities with which different statements and rules are true.

Where do we get these probabilities from. In some cases, we can get these probabilities by analyzing the real-life experience of following these rules: e.g., the experience of investing money or the experience of medical doctors in treating the corresponding disease.

However, such cases are rare – usually, if we have related data, then we can extract rules from this data, and the need for experts is not as critical. In most other situations, we have to reply on the experts themselves to provide us with the probabilities of different statements.

Need to consider a joint distribution. Experts rarely make decisions based on just one rule – such experts can be easily replaced by automatic systems. Usually, experts take into account many rules when making a decision. For example, a financial expert, on observing the behavior of a certain stock, concludes that this stock, if taken by itself, is about to be appreciated. However, before the expert gives the advice to invest in this particular stock, he/she also takes into account:

- the state of the economy as a whole – maybe investing in bonds is currently better – and

- the economic state of different countries – maybe an international investment is better for the client.

Similarly, when a skilled medical doctor proposes a cure, this doctor usually takes into account the patient's symptoms, the patient's history, the patient's allergies, etc.

In all these cases, the conclusion is based on considering several different expert statements and rules. To gauge our confidence in the corresponding recommendation, we therefore need to know the probability that all the used statements are correct. In other words, if we used statements $S_1, \ldots, S_k$, we need to find the probability that a composite statement $S_1 \& \ldots \& S_k$ is correct.

Often, the conclusion has several justifications. In this case, we are interested in the probability that at least one of these justifications is true, i.e., the probability of a complex statement of the type $(S_1 \& \ldots \& S_k) \vee (S_1' \& \ldots \& S_{k'}')$. We can have more complex situations.

In general, we may be interested in the probability of different Boolean combinations of the original statements $S_1, \ldots, S_n$.

It is not possible to extract all possible probabilities from an expert. We can (and do) extract the probabilities $p(S_i)$ of different expert's statements S_i from the experts themselves. However, it is not realistic to expert that experts can also provide the probabilities of all possible Boolean combinations. Indeed, a complete description of all such probabilities means knowing the probabilities of all 2^n atomic statements $S_1^{\varepsilon_1} \& \ldots \& S_n^{\varepsilon_n}$, where $\varepsilon_i \in \{-, +\}$, S^+ means S, and S^- means the negation $\neg S$. These probabilities should add up to 1, but other

than that, there are no restrictions. Thus, to fully describe the joint distribution of n statements, we need to describe $2^n - 1$ numbers. Even for a small knowledge base, with $n \approx 30$, this means that we need more than a billion numbers. There is no way that we ask a billion question to the expert and thus elicit all these probabilities.

Since we cannot directly determine the joint distribution, we must estimate it based on available information. Since we cannot elicit all the values describing the joint probability distribution from the experts, we therefore need to estimate this probability based on the available information.

How the joint distribution is usually estimated. Since we cannot directly elicit the probability of the "and"-combination $S_i \& \ldots \& S_j$ of different statements, a natural idea is therefore to estimate this probability based on the known probabilities $p(S_i)$, $\ldots$, $p(S_j)$ of the corresponding statements.

In other words, we need an algorithm that, given the probabilities a and b of two statements A and B, returns an estimate for the probability of $A \& B$. Let us denote this algorithm by $f_\&(a, b)$.

It is easy to formulate natural properties of this algorithm. For example, since $A \& B$ and $B \& A$ mean the same thing, it is reasonable to require that our estimates for the probability of these two composite statements be the same, i.e., that $f_\&(a, b) = f_\&(b, a)$ for all a and b. Similarly, since $A \& (B \& C)$ means the same as $(A \& B) \& C$, it is reasonable to require that our estimates for these two combinations are the same, i.e., that $f_\&(a, f_\&(b, c)) = f_\&(f_\&(a, b), c)$ for all a, b, and c. We can formulate several similar properties. This approach has been used by MYCIN, historically the first expert system (see, e.g., [1]), and it is one of the main ideas behind the fuzzy logic approach to uncertainty; see, e.g., [2, 3, 4]. In the fuzzy logic approach, the corresponding function $f_\&(a, b)$ is called an *"and"-operation* or, alternatively, a *t-norm*.

Fuzzy logic had many successful applications. However, there is a problem, the problem is that this approach is heuristic: in general, there are no good rules for selecting an appropriate t-norm (and many different t-norms are possible), and there is no guarantee that the results of applying this or that t-norm will indeed adequately reflect the expert knowledge.

Why not use maximum entropy? A natural question is: why not use the maximum entropy approach [5]? Our situation is a particular class of situations for which the maximum entropy approach has been invented:

- we have several possible probability distributions consistent with our knowledge (e.g., consistent with the marginal probabilities $P(S_i)$), and

- we want to select the most reasonable of these distributions.

The recommendation of the maximum entropy approach is to select a distribution with the largest possible value of entropy, i.e., in this case, with the largest

possible value of the quantity

$$S = - \sum_{\varepsilon_1, \ldots, \varepsilon_n} P(S_1^{\varepsilon_1} \& \ldots \& S_n^{\varepsilon_n}) \cdot \ln(P(S_1^{\varepsilon_1} \& \ldots \& S_n^{\varepsilon_n})) \qquad (1)$$

under the constraints that $P(S_1^{\varepsilon_1} \& \ldots \& S_n^{\varepsilon_n}) \geq 0$, that

$$\sum_{\varepsilon_1, \ldots, \varepsilon_n} P(S_1^{\varepsilon_1} \& \ldots \& S_n^{\varepsilon_n}) = 1, \qquad (2)$$

and that, for each i from 1 to n, the probability of each statement S_i is equal to the known value $p(S_i)$:

$$\sum_{\varepsilon_1, \ldots, \varepsilon_{i-1}, \varepsilon_{i+1}, \ldots, \varepsilon_n} P(S_1^{\varepsilon_1} \& \ldots \& S_{i-1}^{\varepsilon_{i-1}} \& S_i \& S_{i+1}^{\varepsilon_{i+1}} \& \ldots \& S_n^{\varepsilon_n}) = P(S_i), \qquad (3)$$

This idea was first proposed by P. Cheeseman [6] and has indeed been successfully used in many applications; see, e.g., [7].

Example. In the simplest case when we have no logical relation between different quantities S_i, the above constraint optimization problem is easy to solve. Indeed, it is known that if we have no information about the relation between two random variables, then the maximum entropy approach concludes that these variables are independent. In this case, this means that for each combination $S_1^{\varepsilon_1} \& \ldots \& S_n^{\varepsilon_n}$, the probability is equal to the product of the corresponding probabilities

$$P(S_1^{\varepsilon_1} \& \ldots \& S_n^{\varepsilon_n}) = \prod_{i=1}^{n} P(S_i^{\varepsilon_i}), \qquad (4)$$

where $P(S_i^-) = P(\neg S_i) = 1 - P(S_i)$.

This makes sense – and, by the way, this can be viewed as an example of the fuzzy approach, with the "and"-operation $f_\&(a, b) = a \cdot b$.

So why not use it in general? The maximum entropy approach works well if we have no logical relations between the statements, i.e., when all 2^n combinations $S_1^{\varepsilon_1} \& \ldots \& S_n^{\varepsilon_n}$ are logically possible. In practice, however, we often have some logical relations between the statements: e.g., we may have implications of the type $S_1 \& \ldots \& S_k \to S$ relating some of these statements.

In this case, we can no longer use the simple independence solution (4), we have to solve the corresponding optimization problem – and since this problem has 2^n unknowns, and we know that this number is in billions, this becomes a complicated computational problem.

Shall we give up on maximum entropy? No. So, shall we abandon hope and use heuristic methods instead? In this paper, we show that we should not do that, that even with logical relations between the statements, it is possible to reduce the number of unknowns to a reasonable one – and thus, effectively apply the maximum entropy approach.

2　In Many Situations, It Is Possible to Make the Maximum Entropy Approach Feasible

Analysis of the problem. Let us consider a situation when there is a logical relation between the statements $S_1, \ldots, S_n$. Because of this relation, not all logical combinations $S_1^{\varepsilon_1} \& \ldots \& S_n^{\varepsilon_n}$ are possible. Let us denote, by W, the set of all the vectors $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)$ for which the corresponding combination is logically possible. In this case, for $\varepsilon \notin W$, the probability of the corresponding logical combination is 0. So, we only need to consider the probabilities of combinations corresponding to $\varepsilon \in W$.

In this setting, the constraint optimization problem (1)-(3) takes the following form: maximize

$$S = -\sum_{\varepsilon \in W} P(S_1^{\varepsilon_1} \& \ldots \& S_n^{\varepsilon_n}) \cdot \ln(P(S_1^{\varepsilon_1} \& \ldots \& S_n^{\varepsilon_n})) \tag{5}$$

under the constraints that $P(S_1^{\varepsilon_1} \& \ldots \& S_n^{\varepsilon_n}) \geq 0$, that

$$\sum_{\varepsilon \in W} P(S_1^{\varepsilon_1} \& \ldots \& S_n^{\varepsilon_n}) = 1, \tag{6}$$

and that, for each i from 1 to n, the probability of each statement S_i is equal to the known value $p(S_i)$:

$$\sum_{\varepsilon \in W:\ \varepsilon_i=+} P(S_1^{\varepsilon_1} \& \ldots \& S_{i-1}^{\varepsilon_{i-1}} \& S_i \& S_{i+1}^{\varepsilon_{i+1}} \& \ldots \& S_n^{\varepsilon_n}) = P(S_i). \tag{7}$$

We can now use the Lagrange multiplier method to reduce this constraint optimization problem to the following easier-to-solve unconditional optimization problem:

$$\text{Maximize } J \stackrel{\text{def}}{=} -\sum_{\varepsilon \in W} P(S_1^{\varepsilon_1} \& \ldots \& S_n^{\varepsilon_n}) \cdot \ln(P(S_1^{\varepsilon_1} \& \ldots \& S_n^{\varepsilon_n})) +$$

$$\lambda \cdot \left(\sum_{\varepsilon \in W} P(S_1^{\varepsilon_1} \& \ldots \& S_n^{\varepsilon_n}) - 1 \right) + \tag{8}$$

$$\sum_{i=1}^{n} \lambda_i \cdot \left(\sum_{\varepsilon \in W:\ \varepsilon_i=+} P(S_1^{\varepsilon_1} \& \ldots \& S_{i-1}^{\varepsilon_{i-1}} \& S_i \& S_{i+1}^{\varepsilon_{i+1}} \& \ldots \& S_n^{\varepsilon_n}) - P(S_i) \right).$$

Let us denote $S_1^{\varepsilon_1} \& \ldots \& S_n^{\varepsilon_n}$ by S^ε. Differentiating the expression J (as described by the formula (8)) by $P(S^\varepsilon)$, we conclude that

$$-\ln(P(S^\varepsilon)) - 1 + \lambda + \sum_{i:\ \varepsilon_i=+} \lambda_i = 0. \tag{9}$$

Thus,

$$\ln(P(S^\varepsilon)) = -1 + \lambda + \sum_{i:\ \varepsilon_i=+} \lambda_i, \tag{10}$$

and, applying $\exp(x)$ to both sides of this equality, we get

$$P(S^\varepsilon) = C \cdot \prod_{i:\ \varepsilon_i=+} r_i, \tag{11}$$

where we denoted $C \stackrel{\text{def}}{=} \exp(\lambda - 1)$ and $r_i \stackrel{\text{def}}{=} \exp(\lambda_i)$.

We can simplify this formula is we represent C as

$$C = \frac{A}{\prod\limits_{i=1}^{n}(1 + r_i)}, \tag{12}$$

where

$$A \stackrel{\text{def}}{=} C \cdot \prod_{i=1}^{n}(1 + r_i).$$

Substituting the expression (12) into the formula (11), we conclude that

$$P(S^\varepsilon) = A \cdot \left(\prod_{i:\ \varepsilon_i=+} \frac{r_i}{1 + r_i} \right) \cdot \left(\prod_{i:\ \varepsilon_i=-} \frac{1}{1 + r_i} \right). \tag{13}$$

If we denote $q_i \stackrel{\text{def}}{=} \dfrac{r_i}{1 + r_i}$ and take into account that

$$\frac{1}{1 + r_i} = 1 - \frac{r_i}{1 + r_i} = 1 - q_i,$$

we thus conclude that

$$P(S^\varepsilon) = A \cdot \left(\prod_{i:\ \varepsilon_i=+} q_i \right) \cdot \left(\prod_{i:\ \varepsilon_i=-} (1 - q_i) \right). \tag{14}$$

Discussion: so we indeed have a feasible system. Interestingly, we get almost the same formula as in the independence case, with two differences:

- first, we have unknown values q_i instead of the original probabilities $p(S_i)$;

- second, all the probabilities $P(S^\varepsilon)$ are now multiplied by some constant A.

The constant A can be determined from the condition (6) which, in this case, takes the form

$$A \cdot \sum_{\varepsilon \in W} \left(\prod_{i:\ \varepsilon_i=+} q_i \right) \cdot \left(\prod_{i:\ \varepsilon_i=-} (1 - q_i) \right) = 1,$$

so

$$A = \frac{1}{\sum\limits_{\varepsilon \in W} \left(\prod\limits_{i:\ \varepsilon_i=+} q_i \right) \cdot \left(\prod\limits_{i:\ \varepsilon_i=-} (1 - q_i) \right)}. \tag{15}$$

Thus, instead of the original unfeasible set of $2^n - 1$ unknowns $P(S^\varepsilon)$, we now have a much smaller – and much more feasible – set of n unknowns $q_1, \ldots, q_n$ (with $n \ll 2^n - 1$). These n unknowns have to be determined from n equation (7) that now take the following form:

$$\sum_{\varepsilon \in W:\ \varepsilon_i = +} \left(\prod_{j:\ \varepsilon_i = +} q_i \right) \cdot \left(\prod_{j:\ \varepsilon_i = -} (1 - q_i) \right) =$$

$$p(S_i) \cdot \left(\sum_{\varepsilon \in W} \left(\prod_{i:\ \varepsilon_i = +} q_j \right) \cdot \left(\prod_{i:\ \varepsilon_i = -} (1 - q_j) \right) \right). \tag{16}$$

Examples. In practice, we can often reduce the number of unknowns even further. Let us consider two such examples. The first example if when we have a rule of the type $S_1 \& \ldots \& S_k \to S_{k+1}$, i.e., equivalently, a *clause*

$$S'_1 \vee \ldots \vee S'_k \vee S'_{k+1},$$

where we denoted $S'_i \stackrel{\text{def}}{=} \neg S_i$ for $i \leq k$ and $S'_{k+1} = S_{k+1}$. Another case when we have such a clause is when we know that something is wrong, we have at least one fault, but we do not know which of the faults it is.

In the maximum entropy approach, unaffected variables are independent, so in this example, we can ignore all the statements $S_{k+2}, \ldots$, and safely assume that $n = k + 1$. In this case, the only sequence ε which is excluded from the set W of possible sequences is the sequence $(-, \ldots, -)$. Thus, the formula (16) takes the form

$$q_i = p(S'_i) \cdot \alpha, \tag{17}$$

where we denoted

$$\alpha = 1 - \prod_{i=1}^n (1 - q_i). \tag{18}$$

Substituting the expression (17) for q_i into the formula (18), we conclude that

$$\alpha = 1 - \prod_{i=1}^n (1 - \alpha \cdot P(S'_i)). \tag{19}$$

Here, we have only one unknown α, and one equation (19) to find this unknown. For $n = 3$, the situation is even simpler: if we explicitly multiply the three differences, cancel out 1 and -1, and divide both sides by α, we get the following quadratic equation from which we can explicitly find α:

$$P(S'_1) \cdot P(S'_2) \cdot P(S'_3) \cdot \alpha^2 - (P(S'_1) \cdot P(S'_2) + P(S'_1) \cdot P(S'_3) + P(S'_2) \cdot P(S'_3)) \cdot \alpha +$$

$$(P(S'_1) + P(S'_2) + P(S'_3) - 1) = 0.$$

What if we know there are at least two faults, i.e., we know that at least two statements S_i must be true? In this case, to form the set W of possible combinations, we dismiss not only the vector $\varepsilon = (-, \ldots, -)$, but also vectors $(-, \ldots, -, +, -, \ldots, -)$ in which only one statement is true. Thus, the equation (16) takes the form

$$P(S_i) = A \cdot \left(q_i - q_i \cdot \prod_{j \neq i}(1 - q_j) \right).$$

The product $\prod\limits_{j \neq i}(1 - q_j)$ can be represented as

$$\prod_{j \neq i}(1 - q_j) = \frac{1}{1 - q_i} \cdot b,$$

where we denoted

$$b \overset{\text{def}}{=} \prod_{j=1}^{n}(1 - q_j).$$

Thus, we have

$$P(S_i) = A \cdot \left(q_i - b \cdot \frac{q_i}{1 - q_i} \right),$$

i.e., equivalently,

$$P(S_i) = A \cdot q_i - B \cdot \frac{q_i}{1 - q_i},$$

where $B \overset{\text{def}}{=} A \cdot b$. If we bring all the terms to a common denominator, we get the following quadratic equation for q_i for which we can find an explicit solution for q_i in terms of only two unknowns A and B:

$$A \cdot q_i^2 - q_i \cdot (P(S_i) + A - B) + P(S_i) = 0.$$

How to compute q_i in the general case: an iterative procedure. In both examples, a relatively small number of sequences ε is dismissed (at least small in comparison with the overall number of 2^n such sequences). Let us consider a general such case.

For any number q_i, let q_i^+ denote q_i, and let q_i^- denote $1 - q_i$. For every sequence of numbers $q = (q_1, \ldots, q_n)$ and for every sequence of signs $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)$, let q^ε denote the product

$$q^\varepsilon \overset{\text{def}}{=} \prod_{i=1}^{n} q_i^{\varepsilon_i}.$$

In these terms the formula (16) takes the form

$$q_i - \sum_{\varepsilon \notin W : \; \varepsilon_i = +} q^\varepsilon = P(S_i) \cdot \left(1 - \sum_{\varepsilon \notin W} q^\varepsilon \right). \tag{20}$$

Thus, we conclude that

$$q_i = \sum_{\varepsilon \notin W:\ \varepsilon_i=+} q^\varepsilon + P(S_i) \cdot \left(1 - \sum_{\varepsilon \notin W} q^\varepsilon\right). \tag{21}$$

In situations when few vectors ε are dismissed, the sums over such vectors are relatively small. Thus, in the first approximation, we can take $q_i^{(0)} = P(S_i)$, and in the following iterations, use the previous value $q_i^{(k)}$ in the right-hand side of the formula (21) to compute the next iteration $q^{(k+1)}$:

$$q_i^{(k+1)} = \sum_{\varepsilon \notin W:\ \varepsilon_i=+} (q^{(k)})^\varepsilon + P(S_i) \cdot \left(1 - \sum_{\varepsilon \notin W} (q^{(k)})^\varepsilon\right). \tag{22}$$

When this process converges, we get the desired formula (21).

References

[1] B. G. Buchanan, E. H. Shortliffe, Rule Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project, Addison-Wesley, Reading, Massachusetts, 1984.

[2] G. Klir, B. Yuan, Fuzzy Sets and Fuzzy Logic, Prentice Hall, Upper Saddle River, New Jersey, 1995.

[3] H. T. Nguyen, E. A. Walker, A First Course in Fuzzy Logic, Chapman and Hall/CRC, Boca Raton, Florida, 2006.

[4] L. A. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338–353.

[5] E. T. Jaynes, G. L. Bretthorst, Probability Theory: The Logic of Science, Cambridge University Press, Cambridge, UK, 2003.

[6] P. Cheeseman, A method of computing generalized Bayesian probability values for expert systems, Proceedings of the Eighth International Joint Conference on Artificial lntelligence IJCAI'83, Karlsruhe, Germany, August 8–12, (1983), 198–202.

[7] V. Kreinovich, H. T. Nguyen, E. A. Walker, Maximum entropy (MaxEnt) method in expert systems and intelligent control: new possibilities and limitations, In: K. M. Hanson and R. N. Silver, Eds., Maximum Entropy and Bayesian Methods, Kluwer Academic Publishers, Dordrecht (1996), 93–100.