Computer Science Journal of Moldova, vol.2, no.1(4), 1994

Fuzzy expert system for diagnosis

O.Păduraru H.N.Teodorescu M.Costin A.Ciobanu
S.Grădinaru

Abstract

This paper presents the basic features of a fuzzy expert system shell designed particularly for diagnosis applications. In the first chapter these features are briefly pointed out while in the next chapters the four main modules (expert interface, inference engine, user interface and explanatory module) are detailed. This shell is particularly suited to the medical diagnosis reasonings due to its ability of modeling the action of the influential and aggravation/alleviation factors.

1 Introduction

The shell we intend to present in this paper can be used to build up a wide range of expert systems but it is better suited to diagnosis problems.

Some of its basic features are outlined below:

a) DESS (diagnosis expert system shell) is able to deal with fuzzy entities and to perform fuzzy reasoning.

b) DESS works under uncertainty conditions. We mention that the rules, as well as the facts entered from outside or inferred during the inference process, may be uncertain. The framework offered by the possibility theory was used as a basis to perform such reasonings.

c) The emphasis is put on the main idea expressed within a sentence which may be used as a premise or a consequent. This sentence must be able to be turned in the expert's mind according to the following pattern:

"The feature F of the entity E takes the value V".

d) The system deals with possibility distributions which may be either numerical or discrete, convex or concave, fuzzy or not. Figures 1 to 4 depict some shapes of distributions allowed by the system.

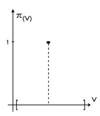

Fig. 1 Crisp value

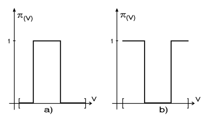

Fig. 2 Ordinary-subset distributions

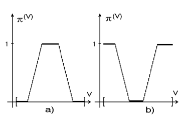

Fig. 3 Fuzzy distributions

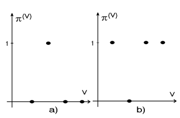

Fig. 4 Non-numerical distributions

The value V assigned to a premise or to the consequent of a rule may be of one of the above forms while a value V' attached to a fact may contain an uncertainty level, unlike the V–distributions which may be fuzzy but always certain.

e) Most if–then rules used by the shell are of "if and only if" type. A pair of two numbers (s, n) is assigned to such a rule representing the sufficiency degree (the certainty degree attached to the direct implication) and the necessity degree (the certainty degree attached to the reverse implication).

f) The knowledge base consists of a concept base and a rule base. A concept extends the notion of variable in the way it may be either numerical or not and represents the pair (F, E) together with the domain of allowable values for F. A numerical concept contains the lower and upper limits of the universe of discourse while a non–numerical concept comprises the alternative set which F relies on.

g) The whole conception of this shell and mostly the knowledge–base design is supported by the connexion between a predefined concept and the idea expressed within a premise or a consequent.

h) The expert uses the natural language when editing a concept or a rule.

i) The structure of a fact comprises the structural description of a concept plus the space allocated to store the possibility distribution. During the inference process a fact is considered to be "empty" if it has not received yet a distribution, otherwise it is considered to be "full".

j) The knowledge base is a very flexible one. Both the rule manager and the concept manager allows almost any kind of changes to be operated into the knowledge base structure. These changes may be performed either within a rule or concept or within the knowledge base regarded as a whole. Thus the knowledge base may be extended forward (by adding rules beyond the end–level) or downward (by adding rules placed before the start level). Actually, a new rule can be added in any place within the knowledge base global structure.

k) This shell covers a wide range of reasonings. To reach this goal, two lines (types) of reasonings are used. One part of the rules is designed to generate distributions to the facts corresponding to the concepts referred to in the consequent and another part is designed to modify the already generated distributions.

l) The system is able to model two notions frequently met in the medical field: the influential factors and the aggravating/alleviation factors.

m) The inference engine control strategy uses a variant of backward chaining technique.

n) The user interface enables the user to input knowledge from outside and to display the final results (most being diagnoses) together with their possibility and nece-ssity (certainty) degrees.

o) On the disk, rules are not stored in an assambled form. There are separate files which store the structural descriptions of the premises, and of the consequents and files which contain the text descriptions of them.

The shell deals with several dictionaries. One of them comprises those texts which are used either as premises or as consequents when a rule is to be edited. Actually, editing a rule means to make all the necessary connexions between these pieces. Therefore, when the user edits a new rule, it is not always necessary to create new texts for that rule because, in many cases, it is sufficient to select one or more texts from the appropriate dictionary.

2 The expert interface

As we have mentioned before, the knowledge base consists of a concept base and a rule base.

The expert interface is that module which allows the expert to add, display, or modify any of the two bases. It works as a manager for which the acquisition of new knowledge is one of its most important tasks.

2.1 The structure of the knowledge base

2.1.1 The concept base

From a logical point of view, a concept represents a pair (F, E) together with the domain of allowable values for F, where E is referring to a particular entity and F to a particular feature of that entity.

From the editor's point of view, a concept includes:

a) A definition which must be as comprehensive as possible, frequently looking like a comment on F and E.

b) The lower and upper limits which F may reach in case of a numerical concept or the alternative set which F can rely on in case of a non–numerical concept.

c) A series of attributes which receive particular values when a new rule is edited or during the inference process.

In section 1, we have explained what a fact represents from the point of view of this shell.

The facts are of two types: output facts and input facts.

An output fact is a structure where the result of a class A or D is stored. An input fact is a structure which stores the result achieved after aggregating several output facts. Obviously, the output facts store the output distributions provided by those rules having a consequent which points to the same target concept.

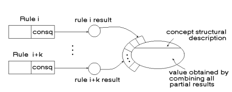

Fig. 5 Combining several partial results

2.1.2 The rule base

The rule base comprises rules structured into four classes.

2.1.2.1. Class A. Class A contains rules as below:

if p_1 and/or p_2 and/or ... and/or p_n then (s, n) q

Such a rule is represented in Fig.6.

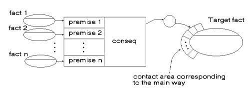

Fig. 6 Connecting a class A rule

The premises $prem_1, \ldots, prem_n$ are expressed by the sentences $p_1, p_2, \ldots, p_n$, which in their turn are referring to the concepts corresponding to $fact_1, \ldots, fact_n$, respectively.

This rule is connected to a concept called "target concept" which contains an area where several class A or D rules could be connected.

We say that all these rules (all having consequents which aim to that target concept) are connected through the "main way".

Assuming that m rules are connected to this target then the result of each rule evaluation is placed in an output fact and the final result is obtained by combining all these partial results. The final result represents the value received by the target fact.

2.1.2.2. Class B. Class B contains rules as below (see Fig.7):

if p_1 and/or p_2 and/or ... and/or p_n
then (s) the certainty that
q
is modified by $k\%$

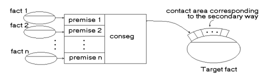

Fig.7 Connecting a class B rule

This rule is connected to a target concept which contains an area where several class B or C rules could be connected.

We say that all these rules (all having cosequents which aim to the same target concept) are connected through the secondary way.

The goal of a class B rule is not to generate distributions for the facts but to act upon some of them — which had already received a value — in order to increase or decrease the certainty of the target fact (that means to decrease or increase the uncertainty level of the target fact possibility distribution).

2.1.2.3. Class C. Class C contains rules as below:

$$\begin{array}{llllll} \text{if} & p_1 & \text{and/or} & p2 & \text{and/or} \ \ldots \ \text{and/or} & p_n \\ \text{then} & (s,n) & & F \leftarrow k1 * F + k2 & & \end{array}$$

Such a rule is represented in Fig.8.

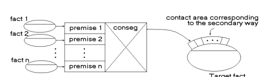

Fig. 8 Connecting a class C rule

This rule is connected to the target fact in the same way a class B rule is.

Unlike class A or B rules, the consequent of a class C rule does not contain a possibility distribution. The goal of this rule is to shift the significant part of the possibility distribution.

2.1.2.4. Class D. Clas D contains rules as below:

$$\begin{array}{l} \text{if} \quad p_1 \quad \text{and/or} \quad p2 \quad \text{and/or} \ \ldots \ \text{and/or} \quad p_n \\ \text{then} \quad (s,n) \qquad F2 \leftarrow k1 * F1 + k2 \end{array}$$

Such a rule is represented in Fig.9.

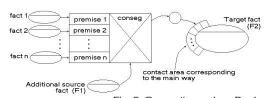

Fig. 9 Connecting a class D rule

The consequent of such a rule does not contain a possibility distribution. Such a rule is connected to the target fact like a class A rule, therefore to the area corresponding to the main way.

The goal of this rule is similar to that of a class C rule, with the difference that now the possibility distribution of the target is obtained by shifting the significant part of a distribution corresponding to an additional source concept.

2.2 Some conclusions on the rulebase structure

Class A and D rules have a generating effect providing possibility distributions to the target fact starting either from the possibility distribution of the consequent (in case of a class A rule) or from the distribution of an additional source fact (in case of a class D rule). Such rules are connected to the target concept through the main way and correspond to the first line of reasoning.

Class B and C rules act upon already generated distributions and are connected to the target through the secondary way, corresponding to the second line of reasoning.

In case of a medical diagnosis fuzzy expert system build up by means of this shell, class B rules are able to model the action of the aggravation/alleviation factors.

This shell can also be used to build up prediction–oriented expert systems in which class C or D rules play a central role.

2.3 The knowledge–base manager

According to the way the knowledge base was structured, its management program is structured into a rule base manager and a concept base manager.

The concept manager allows the expert to edit new concepts, to display or to modify them.

Concerning the modifications over a concept, we mention that the expert may operate changes both in the definition text of the concept and in the alternative set. The expert must take care not to modify the meaning of the concept because in this case all the connections between those rules which refer to this concept and the concept itself are damaged.

Once edited, rules may be subsequently modified if the expert makes such a decision.

The text and/or the possibility distribution assigned to a premise may be changed under two constraints:

- the premise must not lose its previous meaning;
- the correspondence between text and structure must be maintained.

The expert may also delete a premise or add a new one.

The consequent may be changed in the same way a premise is, but now, unlike the changes operated on a premise, it is allowed to replace the consequent by a completely new one. This operation is called redirectation and it is obvious that from now on the consequent will refer to another concept.

Within a class C rule, the target concept (F) may be replaced by another.

Within a class D rule, either the target ($F2$) or the additional source concept ($F1$) may be replaced. The coefficients $k, k1, k2, s, n$ may also be modified.

A rule may be temporary "deleted" in the way it is temporary disconnected from its previous target.

3 The Inference Engine

3.1 Rule Evaluation

3.1.1 Premise part evaluation

Let $\mu_i(u_i)$ denote the possibility distribution of the i premise of a rule, and $\mu_i'(u_i)$ the possibility distribution of that input fact which is to be compared with this premise. Here $u_i \in U_i$, where U_i is the universe of discourse of the concept referred to by the premise. The degree of matching is expressed by two numbers defined as below:

$$Pos_i = \max_{u_i} \min\ (\mu_i(u_i), \mu_i'(u_i)) \tag{1}$$

$$Nec_i = \min_{u_i} \max\ (\mu_i(u_i), 1 - \mu_i'(u_i)) \tag{2}$$

Pos_i shows to what extent the premise is possible and Nec_i shows to what extent the premise is certain when comparing it with a given input fact.

If these values do not satisfy

$$\max\ (Pos_i, 1 - Nec_i) = 1 \tag{3}$$

then they will be normalized.

The entire premise part will be therefore characterized by two numbers $GPos$ and $GNec$, computed according to the rule type. $GPos$ and $GNec$ always satisfy:

$$\max\ (GPos, 1 - GNec) = 1 \tag{4}$$

3.1.2 Class A

a) If $GPos = 1$ and $GNec \in (0, 1]$ then the rule is considered to be "roughly" satisfied and the evaluation result is obtained by assigning a particular uncertainty level to the consequent distribution according to $GNec$ and s (Fig.10, 11, 12 and 13).

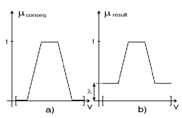

Fig. 10

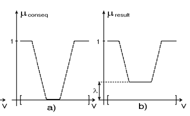

Fig. 11

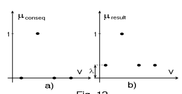

Fig. 12

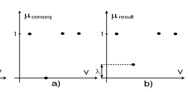

Fig. 13

Fig. 10-13. Some distributions obtained after processing a class a rule A (GPos = 1, GNec > 0)

b) If $GPos < 1$ (then necessary $GNec = 0$) — the rule is considered to be "roughly" not satisfied. In this case:

- if $n > 0$, the rule corresponding to the reverse implication is evaluated, that is

$$\text{if } \neg p_1 \text{ or } \neg p_2 \text{ or } \ldots \text{ or } \neg p_n \text{ then } (n, s)\ \neg q$$

  (here the connective in the initial rule is considered to be AND). The resulted distribution is obtained by inverting the initial μ_{conseq}, then assigning an uncertainty level to it according to $GPos$ and n (Fig.14 and 15);

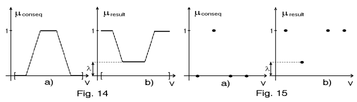

Fig. 14-15. Some distributions obtained after processing a class A rule (GPos < 1, GNec = 0, n > 0)

- if $n = 0$ the result expresses a full uncertainty (Fig.16 and 17) meaning that all the values in the universe of discourse are equally possible;

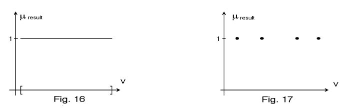

Fig. 16-17 Some distributions obtained after processing a class A rule (GPos < 1 and n = 0) or (GPos = 1 and GNec = 0)

c) If $GPos = 1$ and $GNec = 0$ the result expresses a full uncertainty (Fig.16 and 17).

3.1.3 Class B

a) If $GPos = 1$ and μ_{conseq} and μ^1_{targ} are compatible, that means:

$$\max\ \min\ (\mu_{conseq},\ \mu^1_{targ}) = 1,$$

then the uncertainty level of the target is modified as follows:

- if $k > 0$, it is decreased;
- if $k < 0$, it is increased;
- if $k = 0$, it is not modified.

b) If $GPos = 1$ and μ_{conseq} and μ^1_{targ} are not compatible, then the uncertainty level is shifted into the opposite direction (if $k > 0$ is increased, if $k < 0$ is decreased).

c) If $GPos < 1$ the rule has no effect, μ^1_{targ} remaining unmodified.

Figure 18 shows the target distributions before and after the rule evaluation. We assumed $GPos = 1$.

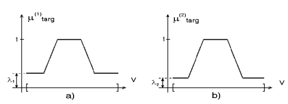

Fig. 18 The target distribution before and after a class B rule evaluation (GPos = 1)

3.1.4 Class C

a) If $GPos = 1$ then the shift is performed according to $k1$ and $k2$, and a new uncertainty level is assigned to the target.

b) If $GPos < 1$ then the shift is not performed, but a new uncertainty level is still assigned to the target.

Note: Normally, in this case, the rule corresponding to the reverse implication should have been evaluated. That means:

$$\begin{array}{ll} \text{if} & \neg p_1 \text{ or } \neg p_2 \text{ or } \ldots \text{ or } \neg p_n \\ \text{then} & (n,s) \quad \neg(F \leftarrow k1 * F + k2) \end{array}$$

if the initial connective was AND. This would imply the new target distribution to be obtained by shifting the significant part of the old distribution and then by inverting and assigning a new uncertainty level to it. Nevertheless, we have preferred the interpretation expressed by b) considering the usual meaning of the rule. Figure 19 shows the target before and after rule evaluation, assuming $GPos = 1$.

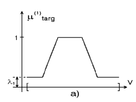

Fig. 19 The target distribution before and after a class C rule evaluation (GPos = 1)

3.1.5 Class D

A class D rule is evaluated similarly to a class C rule with the difference that now μ_{targ} is obtained by shifting (if necessary) the significant part of an additional source distribution.

3.2 The Control Strategy

The inference engine uses a backward chaining variant to perform all the required operations for inferring the final results (in our case, diagnoses).

To accomplish this task, all the terminal facts (or only a pre–selected part of them) are designated as goals, which will be further evaluated one by one. After selecting a goal, the inference engine builds a tree having that goal as a top. This tree consists of rules loaded from the disk and facts generated by the corresponding concepts which are also loaded from the disk. Actually, the backward loading and chaining on one hand and the rules evaluation on the other hand, are somehow interlaced.

At the beginning, a first path is built starting from the selected goal by a depth–backward loading of rules and concepts, until a rule which lies on the base–level is encountered, that is a rule which gets its input data from outside. After entering the required data this rule is evaluated, then the process continues by attempting to evaluate the immediate next rule. If all its input data are prepared, that means all its input facts are "full", this rule is evaluated; if not, the inference engine builds another path started from the first "empty" input fact.

In this way, the backward loading and the forward evaluation are interlaced until the final target (the goal) receives a value.

4 The User Interface

The user interface has two major goals:

- to enable the user to enter all the input data required by those rules which lie on the base level;
- to display the final results, in our case, the diagnoses found to be true (for which the possibility degree $Pos = 1$ and the certainty degree $Nec \in (0, 1]$.

5 The Explanatory Module

The explanatory module allows the user to keep track how the inference engine has reached the final conclusion.

6 Discussion and conclusions

By using this shell, one can build up expert systems mainly for diagnosis–oriented applications.

Its expert interface allows the expert to build up a very flexible knowledge base consisting of rules and concepts. Almost any kind of modifications can be operated into this base.

The inference engine is able to process four classes of if–then rules, most of them presented in an "if and only if" format. Both the premise part and the consequent may be fuzzy, but certain. In addition, a sufficiency and a necessity degree are assigned to the direct and reverse implications, representing the certainty coefficients attached to the direct and the reverse rule, respectively. The facts may be fuzzy and frequently uncertain.

This system, even if it was initially designed to fit diagnosis problems, would be successfully used to build up prediction–oriented expert systems.

References

[1] O.Păduraru, M.Costin, A.Ciobanu, S.Grădinaru, C.Teodorescu. On the Treatment of Uncertainty and Imprecision in an Expert System for Impairment Diagnosis and Prediction, Fuzzy Systems and Artificial Intelligence, An Advanced TextBook, Iaşi, 1991.

[2] H.N.Teodorescu, O.Păduraru, M.Costin, S.Grădinaru, A.Ciobanu, G.Stavri, I.Cornea, V.Constinescu, C.Teodorescu. Fuzzy Expert System for Hearing Loss Risk Assessing, Proceedings of the Fourth Annual Meeting of Biomedical Fuzzy Systems Association, Iizuka, Japan, May 7-8, 1992.

[3] D.Dubois, H.Prade. Theorie des Possibilites. Applications a la representation des connaissances en informatique, Mason, Paris, 1988.

[4] D.Dubois, H.Prade. The Treatment of Uncertainty in Knowledge-Based Systems Using Fuzzy Sets and Possibility Theory, International Journal of Intelligent System, vol.3, no.2, Summer 1988.

[5] J.Lebailly, R.Martin-Clouaire, H.Prade. Use of Fuzzy Logic in a Rule-Based System in Petrolium Geology, Langages et Systemes Informatiques, Fevrier, 1987.

[6] J.C.Buisson, H.Farreny, H.Prade. Dealing with Imprecision and Uncertainty in the Expert System Diabeto III, Langages et Systemes Informatiques, Fevrier, 1987.

[7] H.Farreny, H.Prade, E.Wyss. Approximate Reasoning in a Rule-Based Expert System Using Possibility Theory: A Case Study, Langages et Systemes Informatiques, Fevrier, 1987.

[8] D.Dubois, H.Prade. Handling Uncertainty in Expert Systems: Pitfalls, Difficulties, Remedies, Langages et Systemes Informatiques, Novembre, 1988.

[9] M.Portmann, C.Portmann. Precis d'audiometrie clinique, Masson & CIE, Paris, 1972.

Otilia Păduraru, Horia Nicolai Teodorescu, Mihaela Costin,
Adrian Ciobanu, Sanda Grădinaru Received 3 September, 1993
Institute for Information Science of Roumanian Academy
Branch Iaşi
Copou bd., 22A, Iaşi, 6600, România
tel. (1040 98 146–534)