Volume 3, Issue 10, October 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Diagnostic Fuzzy Logic Expert System for Urology Diseases

Raad Alwan
Department of Computer Science,
Philadelphia University, Jordan

Abstract — A prototype Expert System (ES) is built to deal and diagnose disorders in Urology domain, where the knowledge had been acquired from a specialist and medical text books, and tested in real world on a number of cases under the supervision of specialists' medical team. The architecture of the Urology Expert System (UES) contains five subsystems: the production rule knowledge representation has been used in the knowledge base subsystem in order to model the cases that the system deals with. A backward chaining strategy is the control strategy that has been used in the Inference Engine subsystem. While the used search strategy is the backtracking with promising node (BPN). Reasoning under uncertainty using fuzzy logic has been used in this part.

Keywords — Expert System, Knowledge Base, Urology, Branch and bound, backtracking with promising node.

I. Introduction

Expert system technology emerged as a successful branch of Artificial Intelligence (AI), which is a branch of computer science that is concerned with the design and implementation of programs that are able to emulate human cognitive skills such as: problem solving, visual perception, and language understanding (Shu-Hsien, 2005). It is successfully applies to a wide range of domains such as mineral exploration, organic chemistry, and medical diagnosis (Sucevic and Ilic, 1991; Price, 1994; Zetian et al., 2005; Olabiyisi et al., 2011; Schatz and Schneider, 2011; Badaracco and MartÃnez, 2013; Dall'Agno and Norberto de Souza, 2013). The expert system differs from a conventional applications program in many ways; one of these is that expert system simulates human reasoning about a problem domain, while the application program simulates the domain itself. From another point of view, ES performs reasoning over representations of human knowledge, and it solves problems in heuristic approach rather that algorithmic solutions. Expert Systems in general are able to explain and justify solutions or recommendations in order to convince the user that its reasoning is correct.

The process of building an expert system is often called knowledge engineering. It is typically involves a special forms of interaction between the Expert System builder (Knowledge Engineer) and one or more human expert in some problem area (Ruiz-Mezcua et al., 2011). The knowledge engineer extracts from the human expert their procedures, strategies and rules of thumb for problem solving, and build this knowledge into the expert systems (Karabatak and Ince, 2009; Hariharan et al., 2013). It is not typically concerned with the physical details of how knowledge is encoded, but rather with what the overall conceptual scheme might look like (Sucevic and Ilic, 1991; Price, 1994; Su et al., 2001).

This paper focuses on the field of diagnostic Expert System that uses fuzzy logic to deal with uncertainty about the premises utilized to predict Urology diseases. Several diagnostic Expert Syatem spot the light on dealing with uncertain knowledge either by using fuzzy logic (Han et al., 2001; Castanho et al., 2013) or neuro-fuzzy logic (Benecchi, 2006; Papageorgiou et al., 2008) during the process of building the knowledge base . The overall aim of this work is to gain some insight into the potential use of the expert systems technology. The ultimate goal is to develop a prototype expert system (UES) which allows the user to take the full benefit of the expert system in the field of Urology disorders.

II. The Architecture of UES

The architecture of this system can be classified as a mixed-paradigm programming environment which provides a wide range of representational devices and control mechanisms. As illustrated in **Error! Reference source not found.**, the system consists of five main subsystems, these are:

1. User Interface.
2. Knowledge base.
3. Inference Engine.
4. Explanation Subsystem.
5. Problem Data Base

A. User Interface

The user can interact with UES through an easy and friendly interface. It uses a simple form of natural language (questions and answers). Where the user enters a text and the system will pick up the keywords and pass it to special tables where they can be used later on.

B. Knowledge Base

The knowledge base is a collection of facts (assertions), rules, heuristics, and procedural knowledge. Facts are short term information that can be changed rapidly (during the course of consultation). While rules are long term information about how to generate new facts or hypothesis from what is presently known.

All rules in the system are fall into two categories; either reversible or non-reversible. The main characteristic of the reversible rule is that it remains applicable for any level of belief, whereas the non reversible rule is good only when its level of belief is positive. If the premise has a negative value, the rule should not be used in whatever reasoning is being attempted. UES can deal with a type of rules represented as a production rule (IF-THEN rule). It provides a facility to build four types of rules (Simple, OR, AND, and Formula). Therefore, a module has been built which consists of four procedures in order to build these types. Suppose we need to build the following compound rule:

$$if\ A\ OR\ NOT\ B\ Then\ C$$

This rule is converted by the module and saved in the knowledge base file in the following form:

$$Inf(\vee, \mathcal{R}, C, \oplus, A, \ominus, B, severe)$$

Where, each rule must have eight arguments. These arguments mean the following:

1. The first argument specifies the rule type, such that ($\vee$) means OR type, ($\wedge$) for AND type, ($\mathbb{S}$) for simple type, and ($\mathbb{F}$) for formula type.
2. The second argument determines whether the rule is reversible or not ($\mathcal{R}, \sim \mathcal{R}$). The difference between them is that reversible rule returns its sense of both the premise and conclusion as negated. The non reversible rule can not be manipulated in this way. It does not seem reasonable to apply negation to the premise and conclusion.
3. The third argument represents the name of the inferring part of the rule.
4. The fourth argument refers to the sign of the first condition in the premise (i.e. if the condition is processed by NOT denote it by "$\ominus$" otherwise by "$\oplus$").
5. The fifth represents the name of the first condition in the premise.
6. The sixth refers to the sign of the second argument in the premise (as in 4 above).
7. The seventh represents the name of the second condition in the premise.
8. The eighth argument refers to the level of belief for this rule alone.

The 6th and 7th arguments must have a dummy value when a simple or formula rule types are used.

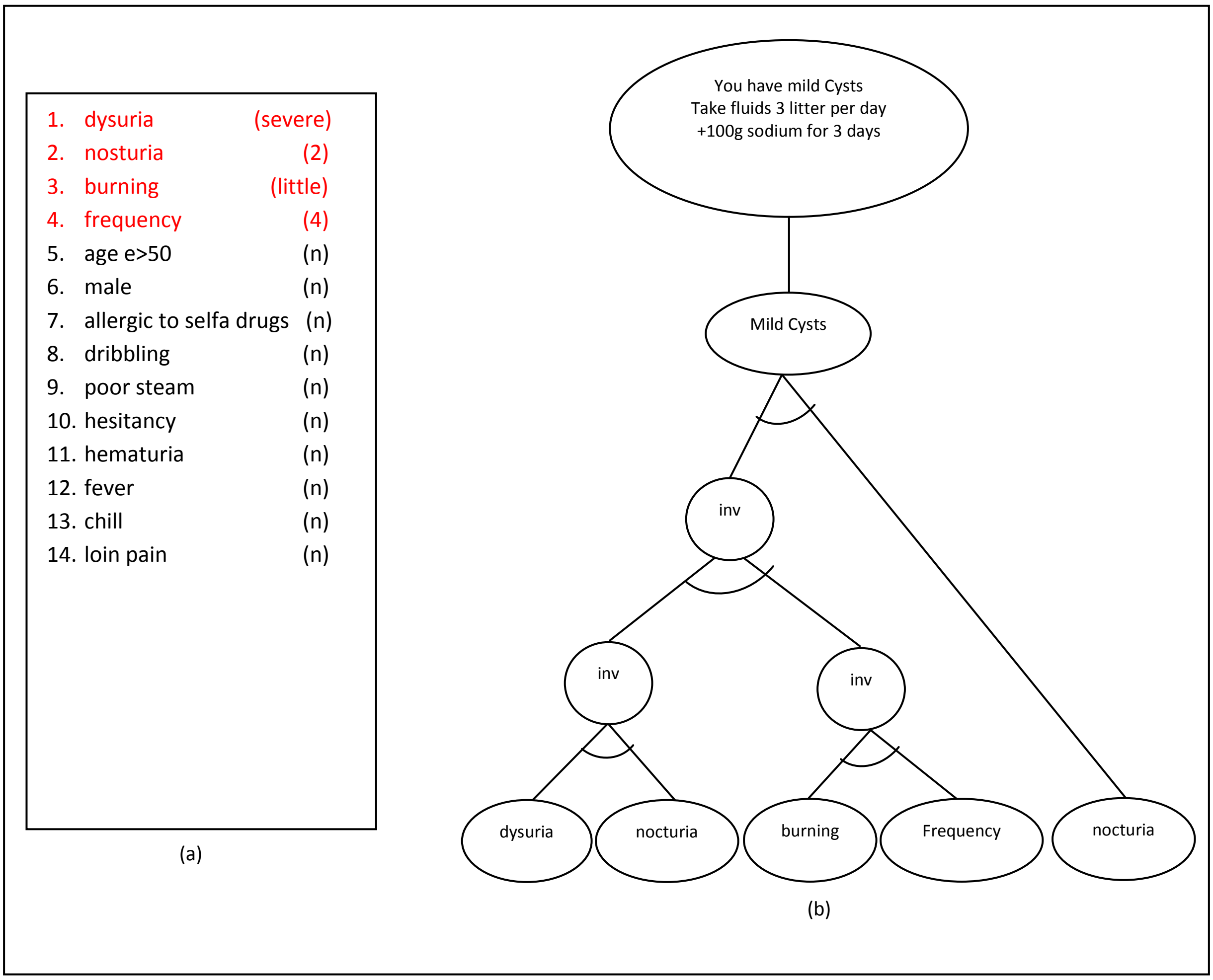

Figure 1: (a) Questions and answers, (b) State space tree using Backtracking with promising node, generated from user's answer in (a).

C. The Inference Engine

Designing an Expert System needs a close look on the details of the meta-knowledge which is how the knowledge is accessed and applied during the search for a solution (Su et al., 2001; Chen et al., 2012). The inference engine of the system used a backward chaining as a control strategy. This type of strategy is similar to the way of thinking used by the medical doctors to reach to their diagnosis. The specialists in the medical domains begin their diagnosis of the disease by creating a few sub-goals in their mind. Each sub-goal refers to a specific disease. Most diseases share with each others a few symptoms. Hence, the specialist picks the most common disease (sub-goal) first, and tries to prove it. So, he starts collecting information about it. If he found that one of the symptoms is not occurring, he cancels this sub-goal and picks up another one in an attempt to prove it and so on, until he reaches an accurate diagnosis. Therefore, we need to sort the sub-goal nodes (diseases) according to their common occurrence in life, from left to right in order to select the most common one and work on it (see **Error! Reference source not found.**) Backtracking with promising node (BPN) is used as a search strategy for manipulating the rules in the system. This type of search is prove to be a cost effective method and can improve the search by reducing the number of rules to be searched.

BPN strategy converts the graph of diagnosis process into a pruned state space tree. The final goal node (advise) is the root of the tree and the module picks the first adjacent vertex which is the most left node (have sever BPH) and moves through this branch until reaches the first terminal node (dysuria), then the system will ask the following question:

Do you have dysuria?

Then the user can answer either by Yes or No. If Yes then the system promotes to the user with the following question:

How bad?

Then the user can answer using a specific word such as (mild, severe, moderate, etc.), or by a statement including one of these words such as (yes, I have a severe dysuria). The system can extract the keyword from the user's answer like (severe). The keyword will be converted to a real number in the range of (0.0-1.0), using the procedure called (m) procedure, where these keywords and their values are saved in an association array where each record has two fields (key, value). For instance:

Keyword	Value
Severe	0.98

The first field refers to the information extracted from the user's response, and the second represent the equivalent certainty value of the answer. This table is searched by the (m) procedure in sequential manner because the keywords are not ordered and they are few in number.

In order to establish the correctness of the sub hypothesis (have severe BPH), the system requires another piece of information to be collected. Thus, the system generates a question (about a nocturia):

How many times you pass water through sleeping?

The user can respond to this question using an integer number, and then the system generates a level of confidence (ranking between 0-1) for this answer. Therefore, the system uses another procedure called (mm) to search in another table. The entries in this table consist of three values: node name, user's answer, and the level of confidence generated from the user's answer. If the user answers the above question by (1), then the system generates the following record:

Node name	User answer	Level of confidence
Nocturia	1	0.01

Since these two terminal nodes (dysuria, nocturia) are connected by an OR operator, the following procedure is invoked by the inference engine subsystem in order to compute the certainty factor for this type of rule:

Proc (Node, CM):- infer(Node, C).
infer(Nodel, CC):- inf(∨,U,Node 1,SignL,Node 2,SignR,Node3,C1),
asserta(o,U,Node 1,SignL,Node2,SignR,Node3,CI)),
asserta(o,U,NodeL,SignL,Node2,SignR,Node3,CI)),
allinfer(Node2, C2), allinfer(Node3, C3),
find-mult(SigL, MultL, SigR, MultR),
*C2F=MultL*C2, C3F=MultR*C3,*
max(C2F, C3F, CE),
*qualify(UmCC, Q), CC=CE*Ct*Q*
assertz(inf_sum(inf(∨,U,Node 1,Sign1,Node 2,SignR,Node3,C1)),
retract(dpinf(∨, U, Node 1, SignL, Node 2, SignR, Node3, C1)).

retract(tdinf(∨, U, Node 1, SignL, Node 2, SignR, Node3, C1)).

The reasoning accuracy increases when the system recognizes a reversible rules from non-reversible rules. This is done by using a (qualify) function. This function takes the value (C), which is an outcome from the level of confidence given by this rule, and a parameter ("$\mathcal{R}$" or "∼ $\mathcal{R}$") which represents reversible or non-reversible type respectively. The (qualify) function described below gives the multiplier (Q) the value (zero or one).

qualfy(U, C, Q):- U="$\mathcal{R}$", Q=1, l.
qualfy(U, C, Q):- U="∼ $\mathcal{R}$", C>0, Q=1, l.
qualfy(U, C, Q):- U="∼ $\mathcal{R}$", C<0, Q=0, l.

Then we compute the level of belief for this rule using the above procedure, as follows:

dysuria→severe→0.98
nocturia→1→0.10

If we give this rule the level of confidence (moderate), then it is converted by procedure (m) into real number (0.66). The multiplier (Q) becomes (1), because the rule is reversible. Thus the system can compute (CC) as follows:

Max(0.98, 0.10, 0.98),
*CC=0.98*0.66*1.*
=0.584

This real number (0.584) defines the level of confidence of this rule which is equivalent to (moderate) in the fuzzy values.

After the system has computed the level of confidence for the rule, it uses the backtracking strategy to pick another terminal node (e.g. burning), by asking the user another question, the search continues in this manner until all the nodes in the tree are visited and the level of confidence at each node are calculated.

After all the questions have been answered, the system starts to calculate the final conclusion for this session (i.e. the value of the root), which represents the diagnosis and the remedy required to treat it.

The final result, which is a real-number, is converted to its equivalent keyword. This conversion is done by the procedure called (ml), which searches an associative array of records, where each record has two entries. The first one represents a range of real number and the other represents its equivalent keyword.

Range	Keyword
0.89-0.99	severe

The (ml) procedure is also invoked when the system responds to the (WHY and HOW) questions.

The final advice given by the system could be:

The diagnosis: You have mild BPH & Ns
*The remedy: Take mithprime tab 2*2 for 3 days. To be seen after 3 days.*

The system works with reasoning under uncertainty using fuzzy logic (i.e. the system could give a conclusion with partial information), and can interact with the user by an easy language form of natural language used in the real life.

D. Explanation Subsystem

The system can explain its action to the user when he asked by (WHY or HOW) question. The WHY question can be asked at any node in the pruned state space tree and the system can explain why it needs the information at this node. For example, if the user responds by WHY question to the following system's question:

System: Do you have dysuria?
User: why?
System:
I am currently trying to use a rule of the OR type. to conclude (ivs) which comes from:
premise1: dysuria
This premise was not negated.
premise2: nocturia
This premise was not negated.
With level of confidence: (moderate).

The above dialogue is done by a why-description procedure. When the system gives its final conclusion, the user can ask how it has reached this conclusion. For example, if the user asked the system how it has reached the following diagnosis?

System: the diagnosis: You have mild BPH&Ns
*The remedy: take nithprim tab 2*2 for 3 days, to be seen after 3 days.*
User: how I have mild BPH&Ns?
System: (You have mild BPH & Ns) concluded from:
premise 1: mild BPH
premise 2: NOT (allergy from selfa drugs)

The user can continue asking (HOW) questions to any of the above premises. If the user asks the system about a terminal node, the system can give a direct answer as follows:

User: How allergy from selfa drug?
System: You told me that.
You are not allergic to selfa drug.

The HOW message dialogue is done using how-explain procedure.

E. Problem Database

All the fuzzy keywords that are related to a specific domain must be distinguished and stored with its values in special symbol table in order to ease the search and locate such values during the consultation. This is what we call a specific database. It is also used when the system answers the WHY and HOW questions.

III. Conclusion & future work

Several concluding remarks have been drawn from the process of designing UES. The main ones are:

1. From the knowledge base representation, we found that the rule-based system is found to be more suitable for diagnosis domain, because it is compatible with human thinking.
2. Using the fuzzy logic under uncertainty improved the reasoning strategy of the system, where the system can work with partial information.
3. Although the diagnosis system uses forward chaining as a control strategy, using backward chaining is found to be more cost effective and increase the performance of the system, since the domain goals are relatively few in number.
4. Using the backtracking with promising node as a search strategy is more appropriate to UES, since the details are pursed as deeply as possible until the search fails. While in Branch-and-Bound, all the possible premises at one level are scanned before moving to the next detail level which requires a greater demand on memory usage, because it carries so many parallel paths.

Since different experts in a specific domain could have different decisions about a specific medical case, it is better to extract the knowledge base from several experts rather than depends on few ones. To improve the performance of the system, a fuzzy ontology techniques could be applied with the cooperation of the domain specialists to enrich the used knowledge based.

References

1) Badaracco, M. & MartÃnez, L. (2013) A fuzzy linguistic algorithm for adaptive test in Intelligent Tutoring System based on competences. *Expert Systems with Applications,* 40**,** 3073-3086.
2) Benecchi, L. 2006. Neuro-fuzzy system for prostate cancer diagnosis. Urology, 357-361.
3) Castanho, M. J. P., Hernandes, F., De RÃ©, A. M., Rautenberg, S. & Billis, A. (2013) Fuzzy expert system for predicting pathological stage of prostate cancer. *Expert Systems with Applications,* 40**,** 466-470.
4) Chen, Y., Hsu, C.-Y., Liu, L. & Yang, S. (2012) Constructing a nutrition diagnosis expert system. *Expert Systems with Applications,* 39**,** 2132-2156.
5) Dall'agno, K. C. M. & Norberto De Souza, O. (2013) An expert protein loop refinement protocol by molecular dynamics simulations with restraints. *Expert Systems with Applications,* 40**,** 2568-2574.
6) Han, M., Snow, P., Brandt, J. M. & Partin, A. 2001. Evaluation of artificial neural networks for the prediction of pathologic stage in prostate carcinoma. Cancer Supplement, 1661-1666.
7) Hariharan, M., Polat, K., Sindhu, R. & Yaacob, S. (2013) A hybrid expert system approach for telemonitoring of vocal fold pathology. *Applied Soft Computing,* 13**,** 4148-4161.
8) Karabatak, M. & Ince, M. C. (2009) An expert system for detection of breast cancer based on association rules and neural network. *Expert Systems with Applications,* 36**,** 3465-3469.
9) Olabiyisi, S. O., Omidiora, E. O., Olaniyan, M. O. & Derikoma, O. (2011) A Decision Support System Model for Diagnosing Tropical Diseases Using Fuzzy Logic. *African Journal of Computing & ICT,* 4.
10) Papageorgiou, E., Spyridonos, P., Glotsos, D., Stylios, C., Ravazoulad, P. & Nikiforidis, G. (2008) Brain tumor characterization using the softcomputing
11) technique of fuzzy cognitive maps. *Applied Soft Computing,* 8**,** 820-828.

12) Price, C. (1994), *Computer-Based Diagnostic Systems*, University of Wales Aberystwyth,
13) Ruiz-Mezcua, B., Garcia-Crespo, A., Lopez-Cuadrado, J. L. & Gonzalez-Carrasco, I. (2011) An expert system development tool for non AI experts. *Expert Systems with Applications,* 38**,** 597-609.
14) Schatz, C. V. D. & Schneider, F. K. (2011) Intelligent and Expert Systems in Medicine - A Review. *XVIII Congreso Argentino de Bioingeniería SABI 2011 - VII Jornadas de Ingeniería Clínica,* 28.
15) Shu-Hsien, L. (2005) Expert system methodologies and applications - a decade review from 1995 to 2004. *Expert Systems with Applications,* 28**,** 93-103.
16) Su, K.-W., Liu, T.-H. & Hwang, S.-L. (2001) A developed model of expert system interface (DMESI). *Expert Systems with Applications,* 20**,** 337-346.
17) Sucevic, D. & Ilic, I. 1991. Uncertain knowledge processing in urology diagnostic problems based expert system. Electrotechnical Conference, 1991. Proceedings., 6th Mediterranean, 741-743 vol.1.
18) Zetian, F., Feng, X., Yun, Z. & Xiaoshuan, Z. (2005) Pig-vet: a web-based expert system for pig disease diagnosis. *Expert Systems with Applications,* 29**,** 93-103.

