Mansoura Engineering Journal, (MEJ), Vol. 25, No. 3, September 2000. E. 1

DECISION TREE LEARNING APPROACH FOR KNOWLEDGE ACQUISITION FROM A MEDICAL DATABASE

مدخل تعلم شجرة القرارات لاكتساب المعرفة من قاعدة بيانات طبية

A. E. EL-Alfy
Dep. of Educational Technology
Specific Education Faculty
Mansoura University
Mansoura Egypt

Prof. M. Bazeed
Dep. Of Urology
Faculty of medicine
Mansoura University
Mansoura Egypt

خلاصة :

تعـد مرحـلة اكتساب الخبرة عنق الزجاجة أثناء بناء النظم القائمة على المعرفة لذلك يمكن لوسائل ميكنة اكتساب المعرفة أن تساعد بدرجة كبيرة فى التغلب على مشاكل تلك المرحلة. تحتفظ معظم المستشفيات بسجلات تفصيلية للمرضى تتضمن وصفا دقيقا للأعراض والتشخيص والدواء المقترح والتغييرات التى يتم ملاحظتها مـع مـرور الوقـت. تتضمن تلك السجلات من الناحية الكمية معرفة قيمة عن التشخيص الطبى والعلاج فإذا توفر نظام مميكن لديه القدرة على التعامل مع هذه السـجلات واستنتاج جوانب المعرفة المفيدة منها عندئذ يحتاج الأطباء فقط التأكد من المخرج الناتج بدلا من قضاء ساعات طويلة فى لقاءات مهندس المعرفة.

تقـدم هـذه الورقـة منهاجية جديدة لإكتساب المعرفة من قواعد البيانات ، تتفق المـنهاجية المقترحة مع خوارزم RITIO فى استنباط القواعد دون رسم شجرة القـرارات وفى حذف السمات الأقل تأثيرا عند البحث ، تبنى المنهاجية المقترحة شـجرة القرارات عندما يطلب ذلك ، تمتاز المنهاجية عن خوارزم ID3 ولواحقه كمـا تمتاز عن RITIO فى أنها توجد الحلول المثالية من خلال النهج العكسى ، اختبرت الطريقة على مثال قياسى وطبقت على قاعدة بيانات واقعية ،

Accepted July 15, 2000

Abstract

Automated knowledge acquisition (AKA) tools hold promise of preventing the process of acquiring knowledge from becoming the bottleneck in the development of knowledge base systems (KBSs). Most hospitals keep detailed records of patients containing descriptions of symptoms, diagnosis, prescribed medicine, and observed changes over time. Cumulatively, these records contain a wealth knowledge about medical diagnosis and treatment. The knowledge acquisition process would be significantly simplified if an automated system could look at these records, and extract valuable pieces of knowledge. Doctors would only need to verify the output of these knowledge acquisition tools rather than sit through hours of interview with knowledge engineers.

This paper presents a new methodology for knowledge acquisition from databases. The proposed methodology coincide with RITIO algorithm [3] in the rule induction without drawing the decision tree and in the eliminating less effective attributes. It builds the decision tree when needed. The new methodology differs from the ID3' likes algorithms [15] and RITIO since it finds the global optimal solutions via back tracking. It is tested on standard example and applied on a real world database.

KEY WORDS

Machine learning, knowledge base systems, knowledge acquestion, data mining

1 Introduction

Expert systems and knowledge-based systems are among the fruitful spin-off from artificial intelligence [1]. An important part of building expert system is acquiring knowledge needed to achieve the desired levels of performance. In most cases, this is a painstaking task requiring a knowledge engineer familiar with the structure and representation of a system to interview a domain expert.

The automation of knowledge acquisition is useful because valuable knowledge is often resident in historical records, books, and manuals within organizations.

Tools for AKA would have the potential of not only reliving the knowledge engineer of the interview burden but also of allowing a rapid assessment of the strength of the current knowledge base and selection of interview questions designed to compensate the weakness in knowledge acquisition.

The AKA tools can be grouped into two categories [2]: Autonomous tools, and tools supporting a knowledge engineer during knowledge acquisition task.

Autonomous tools attempt to learn relevant knowledge from domain sources (such as historical records) independently, with little or no supervision from a knowledge engineer.

Real data is not clean. Data cleaning, including the removal of contradictory and redundant data items and the elimination of irrelevant attributes, has been an important topic in data-mining research and development [3]. Inductive machine learning has become an important approach to automated knowledge acquisition from databases [4]. The decision trees and decision tables (rules) are two important structures in inductive machine learning. The discrete function is their common analytical representation [5].

Existing data – mining algorithms such as C4.5 and HCV [3] start with all attributes in a database and choose useful attributes for concept description. The Rule Induction Two In One (RITIO) algorithm eliminates attributes in database for decreasing irrelevancy from the very beginning of its induction [3]. The elimination it self is an advantage since the most irrelevant attributes are eliminated from the beginning. Decision tree structures as induced by ID3 – like algorithms are known to cause fragmentation of the database whenever a high- arity attribute is tested at a node [15]. This diminishes the understandability of the induced concept hypotheses. Furthermore, ID3-like decision tree structures have a tendency to repeat subtrees when expressing disjunctive concepts. This is referred to as the replication problem. As a consequence of these two problems decision trees tend to grow very large in most realistic problem domains. The RITIO algorithm performs a rule induction without generating the decision trees. RITIO, like ID3-like algorithms makes use of the entropy measure, albeit in a different way, as a means of constraining the hypothesis search space but, unlike ID3-like algorithms, the hypotheses language is the rule structure. ID3- like algorithms, including ID3 and C4.5, need a decompiler (such as C4.5 rules) to transform decision trees into rules, whereas RITIO carries out rule induction without decision trees construction.

However, RITIO and ID3-like algorithms perform no backtracking in their search. Therefore, they are susceptible to the usual risks of hillclimbing search without backtracking: converting to locally optimal solutions that are not globally optimal. The local optimal solution may be less desirable than trees that would have encountered along a different branch of the search.

This paper presents a new efficient algorithm, which carries out rule induction without decision tree construction and eliminates attributes in the given database from the beginning of its induction too (like RITIO). However, it constructs the trees from the rules as byproduct if it desired. The new algorithm is superior than ID3- like algorithms and RITIO in the important point that it finds the global optimal solution through backtracking. A rule filter is used in order to prevent rule redundancy in the rule induction stage. The algorithm is implemented by the object oriented expert system language CLIPS [16] and is applied on a medical database.

2. Problem Formulation

Given a learning task which have the following characteristics:-

- A set of training examples (real world database), S
- Each example (record) has a set of candidate attributes (fields), C_{attr}
- A Target attribute, T_{attr}
- The target attribute has a set of values

It is required to:

a) Determine the less effective attributes for elimination
b) Extract knowledge (rules) from the hypothesis space.
c) Prevent rule redundancy via rule filter

In order to achieve the task, the paper is organized as follows; section 3 explains the knowledge discovery processes. Section 4 presents a survey on the data mining techniques and concentrates on the statistical information measures. In section 5 the proposed algorithm is presented. Section 6 presents the application and results. Finally, the conclusion and references are presented.

3 Knowledge discovery process

Knowledge discovery is the process of finding useful knowledge from large amount of data. Knowledge discovery from databases (KDD) incorporates a sequence of steps such as processing of data, data mining, model selection, post-processing and evaluation for finding useful structure from data [6]. Machine learning (ML) is used as the data-mining step of KDD. Machine learning theory attempts to answer questions such as "How does learning knowledge acquisition performance vary with the number of training examples presented?" and "which learning algorithms are most appropriate for various types of learning tasks?"

Machine learning techniques have been explored for disease screening [7], differential diagnosis [8,9] and other outcome measures [10,11]. Knowledge discovery process is composed of a number of stages [12]. The sequence of knowledge discovery can be represented as shown in fig.1.

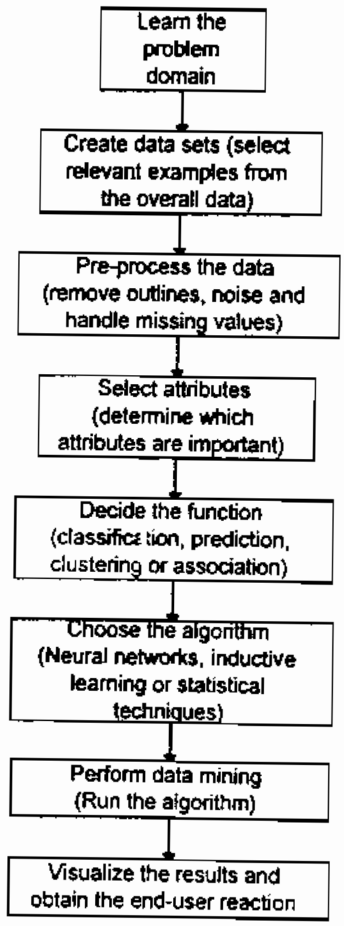

Fig. 1 The stages of the Knowledge discovery process

4 Data mining techniques

The techniques that have been frequently employed in data mining are inductive learning and neural networks.

Inductive learning - which is extremely common in human cognition - is the process of hypothesizing general rules from specific examples or instances. Inductive learning algorithms create rules from a training set containing a number of examples. Each example consists of several attribute values and a class type. Thus, the problem of inductive learning is to generate rules which, given the attribute values of an example can determine the types. The rules generated have the format:

IF attribute x has the value y
THEN class type is z (1)

Or

IF attribute x_1 has the value y_1 AND attribute x_2 has the value y_2
THEN class type is z (2)

The first rule is said to have one *condition* and the second rule, two conditions. Normally, the more conditions that a rule has the fewer examples it covers. Therefore, rules with less conditions usually have a greater generalization ability.

4.1 Artificial neural networks

Artificial neural networks (ANNs) provide a general, practical method for learning real–valued, discrete–valued, and vector–valued functions from examples.

Back propagation algorithm uses gradient descent to tune network parameters to best fit a training set of input- output pairs.

ANN learning is robust to errors in the training data and has been successfully applied to problems such as interpreting visual scenes, speech recognition and learning robot control strategies.

4.2 Statistical methods

Statistical methods for estimating hypothesis accuracy are fundamental to machine learning. The methods focus on three questions. The first, given the observed accuracy of a hypothesis over a limited sample of data, how well does this estimate its accuracy over additional examples? Second, given that one hypothesis outperforms another some sample of data, how probable is it that this hypothesis is more accurate in general? Third, when data is limited what is the best way to use this data to both learn a hypothesis and estimate its accuracy?

4.3 Bayesian reasoning

Bayesian reasoning provides a probabilistic approach to inference. It is based on the assumption that the quantities of interest are governed by probability distribution and that optimal decision can be made by reasoning about these probabilities together with observed data. It is important to machine learning because it provides a quantitative approach to weighing the evidence supporting alternative hypotheses. It also provides the basis for learning algorithms that directly manipulate probabilities, as well as a framework for analyzing the operation of other algorithms that explicitly manipulate probabilities.

4.3.1 Brute – Force Bayes Concept Learning

Assume some finite hypothesis space H defined over the instance space X, in which the task is to learn some target concept c:X?{0,1}.
The learner is given some sequence of training examples $<<x_1,d_1>..<x_m,d_m>>$ where x_i is some instance from X and where d_i is the target value of x_i (i.e., $d_i = c(x_i)$). Assume the sequence of instances $<x_1,\ldots x_m,>$ is held fixed so that the training data D can be written as the sequence of target values $D = < d_1, \ldots d_m,>$.
A straightforward concept learning algorithm to output the maximum a posteriori hypothesis, based on Bayes theorem as follows [13]:

1. For each hypothesis h in H, calculate the posterior probability
$$P(h|D) = P(D|h)P(h)/P(D) \qquad (3)$$
2. Output the hypothesis h_{MAP} with the highest posterior probability
$$h_{MAP} = \operatorname{argmax}_{h\in H} P(h|D) \qquad (4)$$

4.3.2 Bayes optimal classification :

So far the question "what is the most probable hypothesis given the training data?" is considered. The most significant question is "what is the most probable classification of the new instance given the training data?" The most probable classification is obtained by combining the prediction of all hypotheses, weighted by their posterior probabilities. If the possible classification of the new example can take on any value v_j from some set V, then the probability $P(v_j|D)$ that the correct classification for the new instance is v_j is given by:

$$P(v_j |D) = \sum_{hi\in H} P(v_j |h_i)P(h_i D) \qquad (5)$$

The optimal classification of the instance is the value v_j for which $P(v_j |D)$ is maximum. Bayes optimal classification :

$$\operatorname{argmax}_{v_j\in V} \sum_{hi\in H} P(v_j |h_i)P(h_i |D) \qquad (6)$$

4.4 Inductive Learning Using Genetic Algorithm

The genetic algorithms (GAs) can be applied to problems that need search for a solution. Possible solutions are represented as genes and a fitness function is used to determine which solutions survive to the next iteration. Genes with a low fitness are replaced by new genes. The genes in a GA are represented in binary form, that is, as string of ones and zeros. A population of genes is stored at any time. Reproduction is performed by an operation called crossover, where two genes are combined to create a new gene. Diversity is maintained in the population by the mutation operation that generates a new gene by selecting a gene and inverting one of its elements. The proportion of times the mutation operation is applied is determined by a user – specified parameter called the mutation rate. The inductive learning can be seen as a searching problem. There are a large number of possible rules that could be created and it is necessary to search for the set of rules giving the best performance. Thus, it is possible to utilize GAs for inductive learning.

4.5 Decision tree learning

Decision tree learning (DTL) is one of the most widely used and practical methods for inductive inference. It is used for approximating discrete-valued functions that is robust to noisy data and capable of learning disjunctive expressions. The DTL methods search a completely expressive hypothesis space and thus avoid the difficulties of restricted hypothesis spaces.

4.5.1 Decision Tree Representation

Decision trees (DTs) classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. Each node in the tree specifies a test of some attribute of the instance, and each branch descending from that node corresponds to one of the possible values for this attribute. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute in the given example. This process is repeated for the subtree rooted at the new node.

Several inductive learning algorithms and families of algorithms have been developed. These include ID3, AQ and RULES. The ID3 algorithm, developed by Quinlan [14] produces a decision tree. Summary of the ID3 algorithm is given in Appendix 1.

4.5.2 Information Gain Measures

At each node of the tree, an attribute is selected and examples are split according to the value that they have for that attribute. The attribute to employ for the split is the one with the higest information gain for the examples [12]. The entropy measure characterises the (im)purity of an arbitrary collection of examples. Given a collection S, containing positive and negative examples of some target concept, the entropy of S relative to this boolean classification is given by [14]:

$$\text{Entropy (S)} = - p_{(+)}\log_2 p_{(+)} - p_{(-)}\log_2 p_{(-)} \qquad (7)$$

Where $p_{(+)}$ is the proportion of postive example in S

$p_{(-)}$ is the proportion of negative example in S

If the target attribute can take on c different values, then the entropy of S relative to this c-wise classification is defined as:

$$\text{Entropy (S)} = \sum_{i=1}^{c} - p_i \log_2 p_i \qquad (8)$$

Where p_i is the proportion of S belonging to class i

Given the entropy as a measure of the impurity in a collection of training examples, the information gain is simply the expected reduction in entropy caused by partitioning the examples according to this attribute. The information gain Gain(S,A) of an attribute A, relative to a collection of examples S is defined as :

$$\text{Gain(S,A)} = \mathit{Entropy\ (S)} - \sum_{v \in Values\ (A)} \frac{|S_v|}{|S|} \mathit{Entropy}\ (S_v) \qquad (9)$$

Where:

values (A) is the set of all possible values for attribute A

s_v is the subset for which attribute A has values:

v (s_v = {s S|A(s) = v)

5 The proposed Rule Extraction Algorithm

- Determine the candidate attributes' priority list for each target attribute value inside the target concept. The is done at the very beginning calculation of the information gain.
- Determine the information bias value to climinate the less required candidate attribute form competition.
- Form the new candidate attributes list
- For each candidate attribute in the new candidate list do

  For each value in the target attribute **do**

  -Assign Y or N for the target concept

-**Use** the breadth first spread to make instances of the candidate attribute path with their associated values which meet the target attribute value "Yes".
-**Form** the **IF THEN** parts of the extracted rules.
-**Calculate** the number of instances performed.
End do
End do

- **For all** extracted rules **do**
  Chick the redundancy of the rules
  Print the rules
  Perform the decision trees instruction If they are required.

End do

5.1 A Test Example

The operation of the ID3 [13] and RITIO [3] algorithms is based on the learning task, which is represented by the training examples of table 1 [17]. The target attribute *PlayTennis*, which can have values *yes* or *no* for different Saturday mornings, is to be predicted based on other attributes of the morning in question. The information gain measures used in the three algorithms are shown in table 2.

Table 1 The Golf data set

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Table 3 illustrates the ID3, RITIO and the proposed algorithm. In this table, the proposed algorithm is applied on three cases: case 1 no backtracking is used. In case 2 a backtracking is used but no rule filteration. The case 3 uses backtracking and rule filtering. Case 1 gives the same results as ID3 while cases 2 and 3 exceeds with 7-8 rules.

Table 2 The measures of information gain

Attributes	Entropy - RITIO	Information Gain ID3	Information Gain New Algorithm
Outlook	0.694	0.246	0.246
Temperature	0.911	0.029	0.029
Humidity	0.788	0.151	0.151
Wind	0.892	0.048	0.048

Table 3 The measures of information gain

Algorithms	No	Extracted Rules
ID3	1	[If] ***Outlook*** = Sunny **[and]** ***Humidity*** = Normal **[then]** ***PlayTennis*** = Yes
	2	[If] ***Outlook*** = Overcast [then] ***PlayTennis*** = Yes
	3	[If] ***Outlook*** = Rain [and] ***Wind*** = Weak **[then]** ***PlayTennis*** = Yes
RITIO	1	[If] ***Outlook*** = Overcast [and] ***Humidity*** = High [then] ***PlayTennis*** = Yes
	2	[If] ***Humidity*** = normal [then] ***PlayTennis*** = Yes
	3	[If] ***Humidity*** = normal **[and]** ***Wind*** = Weak **[then]** ***PlayTennis*** = Yes
	4	[If] ***Outlook*** = Rain **[and]** ***Humidity*** = High **[and]** ***Wind*** = Weak **[then]** ***PlayTennis*** = Yes
Proposed Case 1	1	[If] ***Outlook*** = Sunny [and] ***Humidity*** = Normal [then] ***PlayTennis*** = Yes
	2	[If] ***Outlook*** = Overcast **[then]** ***PlayTennis*** = Yes
	3	[If] ***Outlook*** = Rain [and] ***Wind*** = Weak [then] ***PlayTennis*** = Yes
Proposed Case 2 (Backtracking)	1	[If] ***Outlook*** = Sunny [and] ***Humidity*** = Normal [then] ***PlayTennis*** = Yes
	2	[If] ***Outlook*** = Rain [and] ***Wind*** = Weak **[then]** ***PlayTennis*** = Yes
	3	[If] ***Outlook*** = Overcast [then] ***PlayTennis*** = Yes
	4	[If] ***Wind*** = weak [and] Outlook = ***sunny*** **[and]** ***Humidity*** = Normal [then] ***PlayTennis*** = Yes
	5	[If] ***Wind*** = weak [and] Outlook = ***Rain*** [then] ***PlayTennis*** = Yes

Algorithms	No	Extracted Rules
	6	**[If]** ***Wind*** = weak **[and]** Outlook = Overcast **[then]** ***PlayTennis*** = Yes
	7	**[If]** ***Humidity*** = high **[and]** Outlook = Rain **[and]** *Wind* = ***Weak*** **[then]** ***PlayTennis*** = Yes
	8	**[If]** ***Humidity*** = high **[and]** Outlook = Overcast **[then]** ***PlayTennis*** = Yes/
	9	**[If]** ***Temp*** = Cold **[and]** Wind =Weak **[then]** ***PlayTennis*** = Yes/
	10	**[If]** ***Temp*** = Mild **[and]** Humidity = ***Normal*** **[then]** ***PlayTennis*** = Yes/
	11	**[If]** ***Temp*** = High **[and]** Outlook = Overcast **[then]** ***PlayTennis*** = Yes/
Proposed Case 3 (Backtracking & Rule filtering)	1	**[If]** ***Outlook*** = Sunny **[and]** ***Humidity*** = Normal **[then]** ***PlayTennis*** = Yes
	2	**[If]** ***Outlook*** = Overcast **[then]** ***PlayTennis*** = Yes
	3	**[If]** ***Wind*** = weak **[and]** Outlook = ***sunny*** **[and]** ***Humidity*** = Normal **[then]** ***PlayTennis*** = Yes
	4	**[If]** ***Wind*** = weak **[and]** Outlook = ***Rain*** **[then]** ***PlayTennis*** = Yes
	5	**[If]** ***Wind*** = weak **[and]** Outlook = Overcast **[then]** ***PlayTennis*** = Yes
	6	**[If]** ***Humidity*** = high **[and]** Outlook = Rain **[and]** *Wind* = ***Weak*** **[then]** ***PlayTennis*** = Yes
	7	**[If]** ***Humidity*** = high **[and]** Outlook = Overcast **[then]** ***PlayTennis*** = Yes/
	8	**[If]** ***Temp*** = Cold **[and]** Wind =Weak **[then]** ***PlayTennis*** = Yes/
	9	**[If]** ***Temp*** = Mild **[and]** Humidity = ***Normal*** **[then]** ***PlayTennis*** = Yes/
	10	**[If]** ***Temp*** = High **[and]** Outlook = Overcast **[then]** ***PlayTennis*** = Yes/

5.2 Program description

The program begins with the main menu window as shown in fig. 2. If the user chooses the **DataBase** button, database simulation is performed as shown in fig.3. If the user chooses the **Data Selection** button, the **Data_selection** form will displayed as shown in fig. 4. This form shows the attributes selection. The target attribute(s) or diseases are displayed automatically in the **Diseases** selection control box. The

number of attributes, patient cases and disease are calculated and displayed too. The user has the ability to choose all attributes, or separate attributes.
The **Solve** button in the main menu (fig. 2) runs the CLIPS program, which extracts the rules and generates the decision tree.
In order to display the rules and the trees the user has to click on the **Show Rules** button. The **Result** screen (Fig. 5) displays the decision trees and the extracted rules.

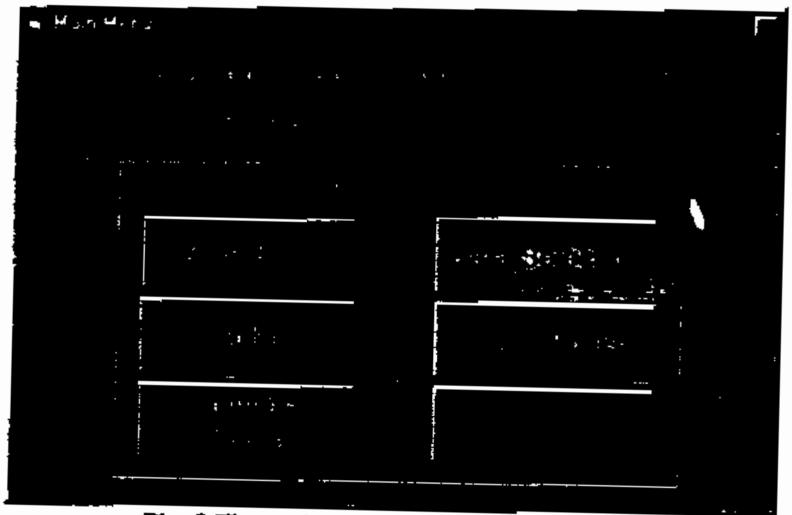

Fig. 2 The decision tree learning main menu

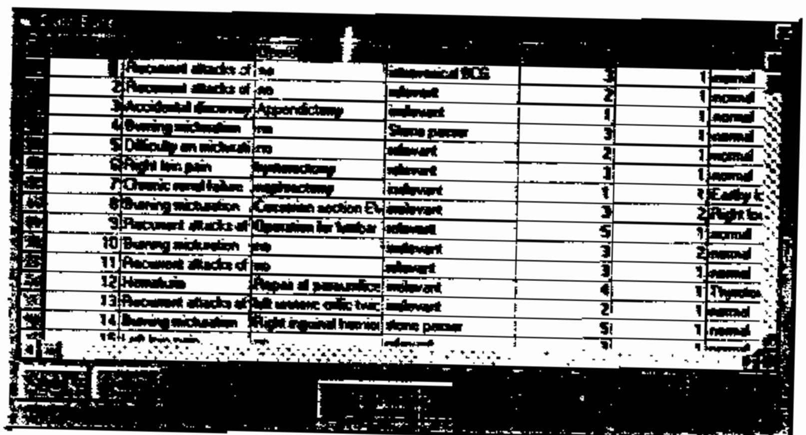

Fig. 3 The simulation of the Urology center database

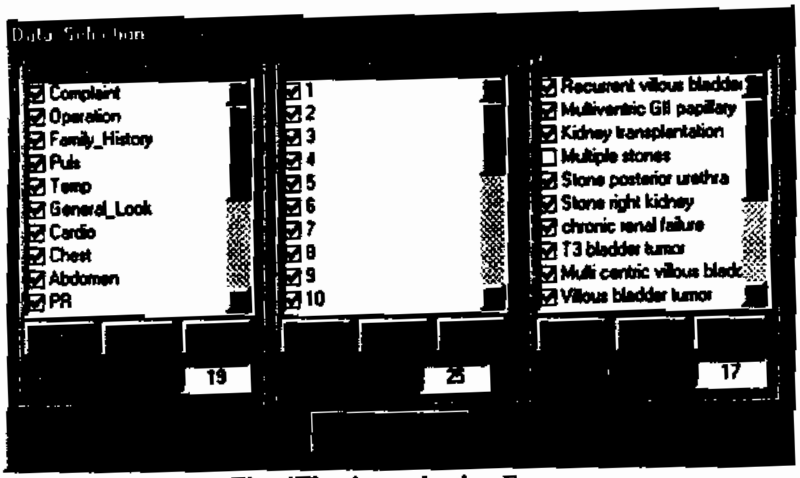

Fig. 4The data selection Form

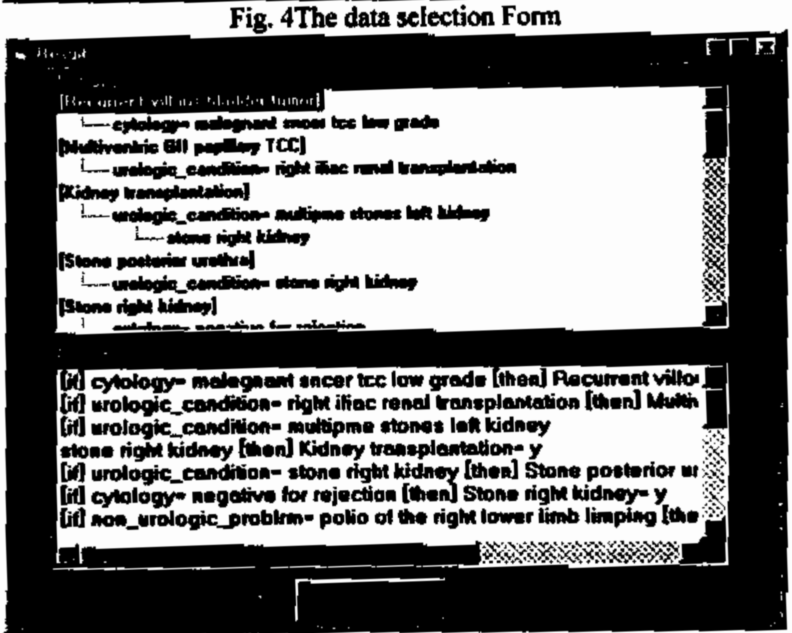

Fig. 5 The decision tree and rule extraction form

6 Applications and Results

The original subject data used from the Urology and Nephrology center' relational database at Mansoura University and transferred into formats acceptable to the proposed machine learning program. The medical abbreviation used are:

KUB : Kidney Ueter Bladder IVU : Intravenous Urography

Samples of knowledge extracted from the proposed algorithm when it is applied on the Urology and Nephrology center' relational database are shown in tables 4, 5, 6, and 7.

Table 4 Case 1 rule extraction with total attributes

Rule No.	Extracted rules with detailed data
1	[if] kub = bad preperation [and] cytology = cyt_no [then] bladder tumor = y
2	[if] kub= bad preperation [and] cytology = malignant cncer TCC low grade [then] bladder tumor = y
3	[if] kub= Faint shadow in left kidney [then] bladder tumor= y
4	[if] kub= Pelvic phlebotomy on the left side [then] bladder tumor= y
5	[if] biopsy= early acute vascular rejection [then] renal allotransplantation = y
6	[if] ultrasonic= not done [and] ivu = perfect [then] stones= y
7	[if] ultrasonic= not done [and] ivu = not done [then] stones= y
8	[if] ultrasonic= stone renal pelvis [then] stones= y
9	[if] ultrasonic= grade II hyperchogenic parenchyma [then] stones= y
10	[if] cytology= negative for rejection [then] chronic renal failure = y
11	[if] ultrasonic = Multiple stones middle & lower calyx [then] canser prostate = y
12	[if] ivu= Mild hydronephrosis, mildy dialated renal pelvis [then] pleviureteral junction obstruction = y

Table 5 Case 2 rule extraction when eliminating Ultrasonic, Endoscopy, Biopsy and Cytology attributes

Rule No.	Extracted rules with detailed data
1	[if] kub= bad preparation [and] ivu= not done [then] bladder tumor= y
2	[if] kub= Faint shadow in left kidney [then] bladder tumor= y
3	[if] kub= Pelvic phlebotomy on the left side [then] bladder tumor = y
4	[if] operation = Appendictomy [then] renal allotransplantation = y
5	[if] ivu = not done [and] kub= stone posterior urethra [then] stones= y
6	[if] ivu = stone renal pelvis, patent ureter down bladder [then] stones= y
7	[if] ivu = evidence of multiple stones [then] stones= y
8	[if] abdomen = scar of graft nephrectomy [then] chronic renal failure= y
9	[if] ivu = Small stones middle & lower calyces [then] cancer prostate= y
10	[if] ivu = Mild hydronephrosis, mildy dilated renal pelvis [then] pleviureteral junction obstruction= y

Table 6 Case 3 rule extraction when eliminating Biopsy and Cytology attributes

Rule No.	Extracted rules with detailed data
1	[if] kub= bad preparation [and] endoscopy= Multiconcentric villous bladder tumor [then] bladder tumor= y
2	[if] kub= bad preparation [and] endoscopy= Domal fungating tumor [then] bladder tumor= y
3	[if] kub= Faint shadow in left kidney [then] bladder tumor= y
4	[if] kub= Pelvic phlebotomy on the left side [then] bladder tumor= y
5	[if] operation= Appendictomy [then] renal allotransplantation= y
6	[if] ultrasonic= not done [and] ivu= perfect [then] stones= y
7	[if] ultrasonic= not done [and] ivu= not done [then] stones= y
8	[if] ultrasonic= stone renal pelvis [then] stones= y
9	[if] ultrasonic= grade II hyperechogenic parenchyma [then] stones= y
10	[if] abdomen= scar of graft nephrectomy [then] chronic renal failure= y
11	[if] ultrasonic= Multiple stones middle & lower calyces [then] cancer prostate= y
12	[if] ivu= Mild hydronephrosis, mild dilated renal pelvis [then] pleviureteral junction obstruction = y

Table 7 Case 4 rule extraction when eliminating Ultrasonic and Endoscopy attributes

Rule No.	Extracted rules with detailed data
1	[if] kub= bad preparation [and] cytology= cyt_no [then] bladder tumor= y
2	[if] kub= bad preparation [and] cytology= malignant sncer, low grade TCC [then] bladder tumor= y
3	[if] kub= Faint shadow in left kidney [then] bladder tumor= y
4	[if] kub= Pelvic phlebotomy on the left side [then] bladder tumor= y
5	[if] biopsy= early acute vascular rejection [then] renal allotransplantation= y
6	[if] ivu= not done [and] biopsy= biop_no [then] stones= y
7	[if] ivu= stone renal pelvis, patent ureter down bladder [then] stones= y
8	[if] ivu= evidence of multiple stones [then] stones= y
9	[if] cytology= negative for rejection [then] chronic renal failure= y
10	[if] ivu= Small stones middle & lower calyces [then] canser prostate = y
11	[if] ivu= Mild hydronephrosis, mild dialated renal pelvis [then] pleviureteral junction obstruction= y

Figure 6 shows the bias values (used for attribute elimination) and the number of instances generated for rule extraction. Figure 7 shows the relation between the number of rules generated and the attribute elimination bias.

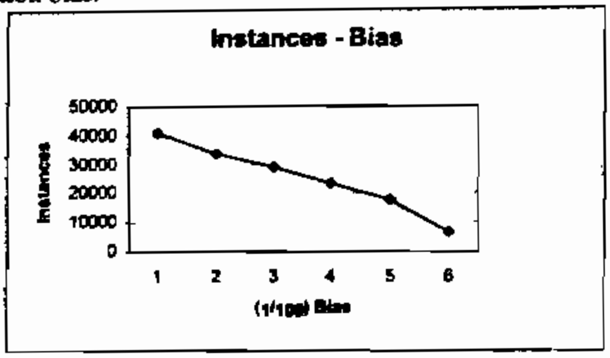

Fig. 6 The instances and the attribute elimination bias

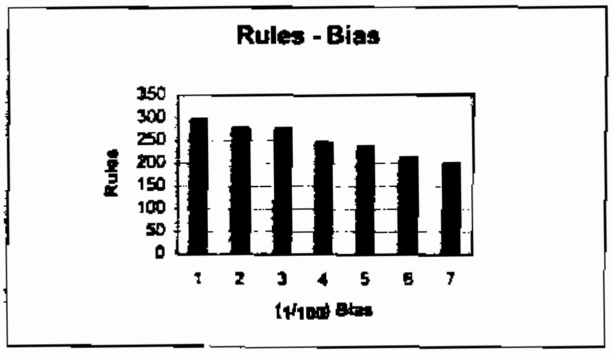

Fig. 7 The number of rules and the attribute elimination bias

7 Conclusions

As the real world databases are large and noisy, the problem of focusing on relevant information has become increasingly important in data mining. The paper uses the induction algorithm, which uses the information theoretic function to induce rules by attribute elimination from a medical database (in Urology and Nephrology center at Mansoura University). It is similar to the decision tree induced in ID3 and C4.5, which uses the most relevant attribute first to branch on. The direct contract between them is that the proposed algorithm uses backtracking to extract rules first then a rule filter is used to eliminate the rule redundancy. This approach does not need de compilation to extract rules from decision trees. The decision tree is constructed if the user needs displaying it. The system is applied on a real database.

Acknowledgment
The authors are grateful to the staff of the Urology and Nephrology Center at Mansoura University for their support and encouragement.

8 References

[1] Wei-Tek Tsai, Rama Vishnuvajjala and Du Zhang "Verification and Validation of Knowledge- based Systems" IEEE Transactions on Knowledge and Data Engineering Vol. 11, No. 1 January/February 1999 pp 202-211.

[2] Soumitra Dutta "Knowledge processing and applied artificial intelligence" Rutter worth – Heinemam Ltd. 1993.

[3] Xindong Wu and David Urpani " Induction by Attribute Elimination" IEEE Trans. On Knowledge and data Engineering Vol. 11 No. 5 Sept /October 1999 PP 805-812.

[4] Xiao-Jia M. Zhou and Tharam S. Dillon " Theoretical and Practical Considerations of Uncertainty and Complexity in Automated Knowledge Acquisition" IEEE Transactions on Knowledge and Data Engineering Vol. 7, No. 5, October 1995 pp 699- 712

[5] B. M. E. Moret. "Decision trees and diagrams" Computing Surveys, vol. 14, no. 4 pp593-623, 1982.

[6] Mani, S. Shankle, R, Dick, M, and Pazzani, M. (1999) "Two stage Machine learning Model for Guideline Development" Artificial Intelligence in Medicine 16, (51-80)

[7] W. R. Shankle, S. Mani, M.J. Pazzani, and Symth. "Detecting Very Early Stages Of Dementia From Normal Aging Within Machine Learning Methods". Artificial Intelligence in Medicine AIME97, volume 1211 pp73-85, Spring 1997.

[8] S. Mani, W. R. Shankle, M.J. Pazzani, P. Symth and M.B. Dick. "Differential Diagnosis of Dementia: A Knowledge discovery and data mining (KDD) Approach" JAMIA supplement p 875, 1997.

[9] C. Ohmann, Q. Yang, V Moustakis, K. Lange and P. J. van Elk. "Machine Learning Techniques Applied to Diagnosis of Acute Abdominal Pain". Artificial Intelligence in Medicine AIME95, volume 934 pp276-281 Spring 1995

[10] G., F. Cooper, C. F. Aliferis, R. Ambrosino, J. Arnois, B. G. Buchanan, R. Crauana, M. J. Fine, Fine C Glymour, G. Gordon, B. H.

Hanusa, J, E. Jansosky, C. Meek, T. Mitchell, T. Richardson and P. Sprites. "An evaluation of machine learning methods for predicting pneumonia mortality." Artificial Intelligence in Medicine 9: 107-138 1997.

[11] K. C. Abston, T. A. Pryor, P. J. Haug and J. L. Andreson. "Inducing practice guidelines from a hospital database" JAMIA supplement 1997, 168-172.

[12] R. J. Alock and Y. Manolopoulos "Using Genetic Algorithms For Inductive Learning" Computational Intelligence And Applications 1999, World Scientific And Engineering Society Press.

[13] Tom M. Mitchell "Machine learning" McGraw-Hill Companies 1997.

[14] Quinlan J. R. "Learning Efficient Classification Procedures and their Applications to Chess End Games in Machine learning , an Artificial Intelligence Approach. Eds Michalski R. S. Carbonell J. G. and Mitchell T. M. Morgan Kaufmann, San Mateo, California, 1983, pp 463-482

[15] G. Pallo and D. Haussler " Boolean Feature Discovery in Empirical Learning" Machine learning pp 71-99 vol. 5 1990.

[16] Joseph C. Giarratano " CLIPS Version 6.0" Software Technology Branch, NASA/ Lyndon B. Johnson Space Center.

[17] J.R. Quinlan "C4.5 programs for machine learning". Morgan Kaufmann, 1993.

Appendix 1

Summary of the ID3 algorithm specilized to learning boolean – valued functions.

ID3 (Examples, Target_attribute, Attributes)

> Examples are the training examples. Target_attribute is the attribute whose value is to be predicted by the tree. Attributes is a list of other attributes that may be tested by the learned decision tree. Returns a decision tree that correctly classifies the given Examples.

- Create a root node for the tree
- If all Examples are positive, Return the single- node tree Root, with label = +
- If all Examples are negative, Return the single- node tree Root, with label = -
- If attribute is empty, Return the single- node tree Root, with label = most common value of Target_attribute in Examples
- Otherwise Begin

- A ? the attribute from Attribute that best (with the highest information gain) classifies Examples
- The decision attribute for Root ? A
- For each possible value v_i of A,
  - Add a new branch below Root , corresponding to the test $A = v_i$
  - Let the $Examples_{vi}$ be the subset of Examples that have value v_i for A
  - If $Examples_{vi}$ is empty
  - Then below this new branch add a leaf node with label = most common value of Target_attribute in Examples
  - Else below this new branch add the subtree ID3 ($Examples_{vi}$, Target_attribute, Attributes – {A}))
- End
- Return Root