Bridging Data Mining Model to the Automated Knowledge Base of Biomedical Informatics

Kittisak Kerdprasop and Nittaya Kerdprasop

Data Engineering Research Unit, School of Computer Engineering,
Suranaree University of Technology, Nakhon Ratchasima 30000 Thailand
{kerdpras, nittaya}@sut.ac.th

Abstract

The process of data mining comprises of seven major steps: (1) data integration, (2) data transformation, (3) data cleaning, (4) data selection, (5) pattern extraction or knowledge mining, (6) pattern evaluation, and (7) knowledge presentation. Steps 1 to 4 are pre-data mining, whereas steps 6 and 7 may be viewed as post-data mining. Therefore, the seven major steps can be grouped into pre-data mining, mining, and post-data mining. We focus our study on the post-data mining processing. Most data mining systems finish their processing at the knowledge presentation step. Our work further the regular post-data mining processing to the step of automatic knowledge deployment. This paper illustrates the knowledge deployment step in which its input is the induced knowledge, in the formalism of classification rules. These rules are evaluated and filtered on the basis of coverage measurement, that is from all the training cases, how many cases are covered by the rule. High coverage rules are transformed into decision rules to be used by the inference engine of the expert system. This post-data mining processing leads to a new design of the next generation rule-based expert system in a medical domain. It is a new idea in that in addition to the set of predefined rules in the knowledge base, the system includes rules that are automatically induced from the database instances. We design the inductive expert system such that the inductive process has been done through the tree-based knowledge discovery technique. Probabilistic decision rules are then transformed from the induced decision tree. The induced, as well as predefined, rules together form a knowledge base for the inductive expert system. Another feature of our system is the inference engine that can be created automatically. The system is intended to support decision making in biomedical informatics. The accuracy of recommendation given by the expert system is evaluated and compared to other three classification systems: decision-tree induction, rule induction, and neural network. The experimental results confirm the high accuracy of our inductive expert system and the automatically created knowledge base.

Keywords: *Post data mining processing, Automatic knowledge acquisition, Knowledge mining, Inductive expert system, Medical decision support system*

1. Introduction

Computers have been applied to medicine and health care since the 1950s as significant tools in medical diagnosis and therapy [22]. The success of medical expert systems such as MYCIN [24] and INTERNIST-1 [14] has attracted considerable attention from cross-discipline researchers including medical experts, computer scientists, engineers, decision analysts, and mathematicians. Later development of

applications such s electronic health records, hospital information system, medical decision support systems, and many others have contributed to the emergence of medical informatics as a new academic discipline. The term *medical informatics* was originally coined in Europe to address the focus on application of informatics to support medical practice and clinical research [25]. With the success of the human genome project [11] and the rise of bioinformatics, many observers [12, 13, 26] have argued that the name medical informatics should be replaced with *biomedical informatics* to reflect the broad range of issues in biomedical research, clinical practice, and health-related applications.

Biomedical informatics is thus an interdisciplinary science that involves the incorporation of knowledge from diverse disciplines, including health science (e.g., medicine, dentistry, pharmacy, nursing), computer science, engineering, information science, cognitive science, biostatistics and mathematics. This emerging field encompasses scientific endeavors ranging from theoretical model construction to the building and evaluation of practical tools to solve complex problems in prevention and treatment of diseases, clinical/medical decision making, and delivery of effective health care. The focus of this paper is to propose a new methodology in developing a computer-assisted decision support system based on first-order logic to improve medical practice. The main contribution of our work is the systematic process of deriving knowledge as a data model from existing patient records. In addition to data modeling, the inference system of derived knowledge for decision support can be created automatically as well.

The presentation of our work is organized as follows. We review related work in medical decision support system in Section 2. Then, we discuss the system design and algorithms for knowledge inducing and inferring in Section 3. The system implementation and its performance evaluation results are in Section 4. We conclude our work and discuss future research direction in Section 5.

2. Related Work

The automated learning of models from patient data and biomedical records has become more and more essential since the extensive computerization in healthcare industry and the significant advancement in genomic and proteomic technologies during the last decade. Medical and clinical databases have been created and constantly growing at an exponential rate. The development of an automatic and intelligent data analysis tool is an obvious solution to the data-flooding problem in medical domains [9, 18, 23, 27]. In recent years we have witnessed increasing number of applications on knowledge mining from biomedicine, clinical and health data. Roddick and his colleagues [20] discussed the two categories of mining techniques applied over medical data: explanatory and exploratory. Explanatory mining refers to techniques that are used for the purpose of confirmation or making decisions. Exploratory mining is data investigation normally done at an early stage of data analysis in which an exact mining objective has not yet been set.

Explanatory mining in medical data has been extensively studied in the past decade employing various learning techniques. Bojarczuk and colleagues [1] applied genetic programming method to discover classification rules from medical data sets. Ghazavi and Liao [6] proposed the idea of fuzzy modeling on selected features medical data. Huang and his team [8] introduced a system to apply mining techniques to discover rules from health examination data. Then they employed a case-based reasoning to

support the chronic disease diagnosis and treatments. The recent work of Zhuang and others [29] also combined mining with case-based reasoning, but applied a different mining method. Biomedical discovery support systems are recently proposed by a number of researchers [2, 3]. Some work [21] extended medical databases to the level of data warehouses.

Exploratory, as oppose to explanatory, is rarely applied to medical domains. Among the rare cases, Nguyen, Ho and Kawasaki [16] introduced knowledge visualization in the study of hepatitis patients. Palaniappan and Ling [17] applied the functionality of OLAP tools to improve visualization.

It can be seen from the literature that most medical knowledge discovery systems have been designed up to the stage of knowledge mining without further discussion on the final stage knowledge inferring and deployment. Kumar and team-mates [10] include decision-making unit with no detail regarding implementation in their decision-support system. Horng and colleagues [7] propose an expert system to classify microarray gene expression emphasizing only the gene selection and classification stages. The work of Exarchos and his team [4] is closely related to ours, but their methodology on the automatic creation of expert system is based on the fuzzy set. Our work, on the contrary, is a rule-based inductive expert system in that the knowledge is induced rules and the inference engine is also automatically generated from those rules. Uncertainty of knowledge applicability is based on the probabilistic concept

3. Probabilistic Knowledge Induction System: Its Design and Methodology

Electronic medical data are valuable resources for the automatic learning of useful knowledge to support scientific decision-making. Medical knowledge mining is an emerging area of computational intelligence applied to automatically analyze electronic medical records and health databases. The non-hypothesis driven analysis approach of data mining technology can induce knowledge from clinical data repositories and health databases. Various data mining methods have been proposed to learn useful knowledge from medical data, but major techniques adopted by many researchers are rule induction and classification tree generation [4, 7, 10, 28].

Our design of a knowledge induction system (Figure 1) is also based on a decision-tree induction concept. Decision tree induction [19] is a popular method for mining knowledge from data and representing the result as a classifier tree. Popularity is due to the fact that mining result in a form of decision tree is interpretability, which is more concern among casual users than a sophisticated method but lack of understandability. A decision tree is a hierarchical structure with each node contains decision attribute and node branches corresponding to different attribute values of the decision node. The goal of building decision tree is to partition data with mixing classes down the tree until each leaf node contains data with pure class.

In our system framework, we increase interpretability of the knowledge mining results by transforming the decision tree structure into a small set of decision rules. After a complete decision tree has been created, we calculate the probability of case occurrence augmented with each leaf node. In the phase of decision rule generation, these probability values will be sorted. Rules within the top ranking part will be displayed to assist medical practitioner for making decision. In the designed framework, probabilistic knowledge induction system is composed of four main components: data integration, tree induction, probabilistic-rule generation, and the knowledge inferring

and answering engines. Data integration component is responsible for collecting data from different sources, cleaning and format transforming. These data are to be used by the tree induction component.

In order to build a decision tree, we need to choose the best attribute that contributes the most towards partitioning data to the purity groups. The metric to measure attribute's ability to partition data into pure class is *Info*, which is the number of bits required to encode a data mixture. To choose the best attribute we have to calculate information gain, which is the yield we obtained from choosing that attribute. The gain value of each candidate attribute is calculated. The attribute with maximum gain value is chosen to be the decision node. The process of data partitioning continues until the data subset along each tree branch has the same class label.

Given the induced tree, the probabilistic-rule generation component traverses each tree branch to calculate the likelihood of path occurrence. This likelihood is interpreted as the probability of event and associated to the rule generated from the path traversal. The generated probabilistic rules are then sorted. Rules at the top ranking (specified by the given minimum probability) are stored in the knowledge base as the probabilistic knowledge and could be used for recommendation or answering query to the medical practitioner. Algorithms for knowledge induction based on tree structure (Algorithm 1), probabilistic-rule generation from decision tree (Algorithm 2), and probabilistic knowledge inferring to answer the most probable class decision on new case (Algorithm 3) are given in Figures 2, 3, and 4, respectively.

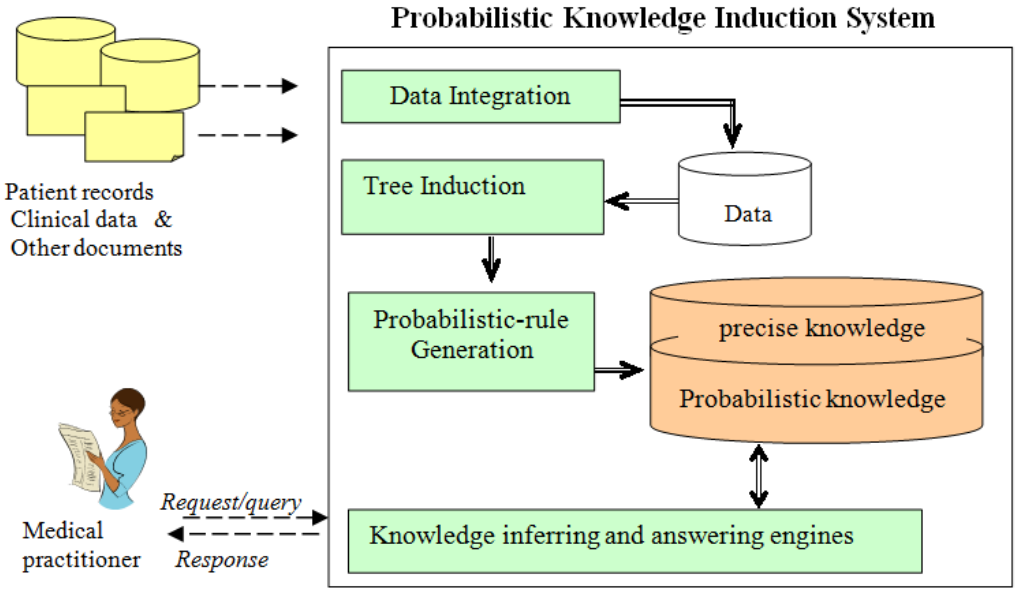

Figure 1. A Framework for Probabilistic Knowledge Induction Based on the Decision Tree

Algorithm 1 Knowledge induction

Input: a data set formatted as Prolog clauses
Output: a decision tree with node and edge structures

(1) Initialization

(1.1) Clear temporary knowledge base (KB)
(1.2) Set node counter = 0

(1.3) Scan data set to get information about data attributes and instances

(2) Building tree

(2.1) Increment node counter

(2.2) Repeat steps 2.2.1-2.2.4 until there is no more attributes left for creating decision nodes

(2.2.1) Compute Info value of each candidate attribute

(2.2.2) Choose the attribute that yields minimum Info to be decision node

(2.2.3) Assert edge and node information into KB

(2.2.4) Split data instances along node branches

(2.3) Repeat steps 2.1 and 2.2 until the lists of positive and negative instances are empty

(2.4) Output tree structure containing node and edge predicates

Figure 2. Knowledge Induction Algorithm to Generate a Decision-tree Structure

Algorithm 2 Probabilistic knowledge generation

Input: a decision tree with node and edge structures, and a probability threshold
Output: a set of probabilistic rules ranking from the highest probability

(1) Traverse tree from a root node to each leaf node

(1.1) Collect edge information and count number of data instances

(1.2) Compute probability as a proportion

(number of instances at leaf node) / (total data instances in a data set)

(1.3) Assert a rule containing a triplet (attribute-value pair, class, probability value) into KB

(2) Sort rules in the KB in descending order according to the rules' probability

(3) Remove rules that have probability less than the specified threshold

(4) Assert selected rules into the KB and return KB as an output

Figure 3. An Algorithm to Generate Probabilistic Decision Rules from the Tree Structure

Algorithm 3 Probabilistic knowledge inferring

Input: a KB containing probabilistic knowledge, and a new case with unknown class value

Output: a decision on most likely class of the new case

(1) Read all attribute-value pairs appeared in the given case

(2) Compare the pairs with relevant rules in the KB to get the decision class value

(3) Compute the decision confidence as

(number of matched attribute-value pair) × (probability of the decision rule)

(4) Output a final decision based on the voting scheme

Figure 4. A Utilization of Probabilistic Decision Rules to Predict Unknown Class

The induced probabilistic knowledge base is a major part of our medical inductive expert system. It is a rule-based expert system with two important automated components: automatic knowledge acquisition subsystem and the inference engine to support explanation and reasoning.

4. System Implementation and Experimental Results

The implementation of a probabilistic knowledge induction component and the rule-based inference engine is based on a logic programming paradigm. A rapid prototype of the proposed inductive expert system is provided in a declarative style using second-order Horn clauses [15]. Prolog code appeared in appendices follows the syntax of SWI Prolog (www.swi-prolog.org). The intuitive idea of our design and implementation is that for such a complicated knowledge-base system coding should be done declaratively at a high level to alleviate the burden of programmers. The advantages of declarative programming style are thus a decrease in program development time and the increase in expressiveness of knowledge representation and efficiency of knowledge utilization.

4.1. Data Format

In logic programming, program and data take the same format, i.e. all are in Prolog clausal form. Each data record is a fact in Prolog term. The fact is the declaration of a true statement. For the purpose of demonstration, we use the health examination data of 86 patients for discharge decision after their operations. Each patient record contains eight observed attributes. The general conditions such as blood pressure and temperature are observed to determine whether the patient is in good condition and should be sent home shortly (class=home), or the condition is quite moderate and should stay at the hospital ward for further follow up (class=ward). The post-operative data set, which is downloadable from the UCI repository [5], in Prolog clausal form is shown some part as the following:

```
attribute( internalTemp,          [mid, high, low] ).
attribute( surfaceTemp,           [mid, high, low] ).
attribute( oxygenSaturation,      [excellent, good, fair, poor] ).
attribute( bloodPressure,         [high, mid, low] ).
attribute( tempStability,         [stable, mod-stable, unstable] ).
attribute( coreTempStability,     [stable, mod-stable, unstable] ).
attribute( bpStability,           [stable, mod-stable, unstable] ).
attribute( comfort,               [5,7,10,15] ).
attribute( class,                 [home, ward] ).

instance(1, class=ward, [internalTemp=mid, surfaceTemp=low,
                oxygenSaturation=excellent, bloodPressure=mid,
                tempStability=stable, coreTempStability=stable, bpStability=stable,
                comfort=15] ).
instance(2, class=home, [internalTemp=mid, surfaceTemp=high,
                oxygenSaturation=excellent, bloodPressure=high,
                tempStability=stable, coreTempStability=stable, bpStability=stable,
                comfort=10] ).
instance(3, class=ward, [internalTemp=high, surfaceTemp=low,
                oxygenSaturation=excellent, bloodPressure=high,
                tempStability=stable, coreTempStability=stable,
                bpStability=mod_stable, comfort=10] ).
```

Another data set used in our experimentation is the diagnosis on breast-cancer recurrence. Training data contains 175 cases, whereas the test data contains 16 cases. Both training data and test data take the same format of Prolog clauses, which can be illustrated as follows:

```
% ------------------------------
%         Data: Breast-cancer.pl
% ------------------------------
%   Diagnosis recurrence of breast cancer
%   class = yes : recurrent events
%   class = no  : no recurrence
%
attribute( age,           [range20_29, range30_39, range40_49, range50_59,
                          range60_69] ).
attribute( menopause, [lt40, ge40, premeno] ).
attribute( tumorSize, [rang0_4, range5_9, range10_14, range15_19,
                          range20_24, range25_29, range30_34, range35_39,
                          range40_44, range45_49, range50_54] ).
attribute( invNodes,  [range0_2, range3_5, range6_8, range9_11, range15_17,
                          range24_26] ).
attribute( nodeCaps,  [missing, yes, no] ).
attribute( degMalig,  [1, 2, 3] ).
attribute( breast,        [left, right] ).
attribute( breastQuad, [left_up, left_low, right_up, right_low, central] ).
attribute( irradiat,      [yes, no] ).
attribute( class,         [no, yes] ).

instance(1, class=yes, [age=range40_49, menopause=premeno,
                              tumorSize=range15_19, invNodes=range0_2,
```

```
                        nodeCaps=yes, degMalig=3, breast=right,
                        breastQuad=left_up, irradiat=no]).
instance(2, class=no, [age=range50_59, menopause=ge40,
                        tumorSize=range15_19, invNodes=range0_2,
                        nodeCaps=no, degMalig=1, breast=right,
                        breastQuad=central, irradiat=no]).
instance(3, class=yes, [age=range50_59, menopause=ge40,
                        tumorSize=range35_39, invNodes=range0_2,
                        nodeCaps=no, degMalig=2, breast=left,
                        breastQuad=left_low, irradiat=no]).
```

4.2. Prolog Implementation: Knowledge Induction and Probabilistic Rule Generation

The three algorithms (explained in the previous section) are called by the main module, which is the top-level of our program implementation. The Prolog coding of main module is as follows:

```
main :- init(AllAttr,EdgeList),
        getnode(N),
        create_edge_onelevel(N,AllAttr,EdgeList),
        addKnowledge,
        write(chooseMinProb),
        read(Min),
        selectRule(Min,Res),
        maplist(writeln,Res).

getnode(X) :- current_node(X),
              X1 is X+1,
              retractall(current_node(_)),
              assert(current_node(X1)),
              X1 <4000.      % limit tree size at 4000 nodes

create_edge_onelevel(_,_,[]):-!.
create_edge_onelevel(_,[],_):-!.
create_edge_onelevel(N,AllAttr,EdgeList) :-
        create_nodes(N,AllAttr,EdgeList).

create_nodes(N,AllAttr,[H1-H2/PB-NB|T]) :-
        getnode(N1),
        assert(edge(N,H1=H2,N1)),
        assert(node(N1,PB-NB)),
        append(PB,NB,AllInst),
        ( (PB\==[], NB\==[])->
            (cand_node(AllAttr,AllInst,AllSplite),
             min_cand(AllSplite,[V,MinAttr,Splite]),
             delete(AllAttr,MinAttr,Attr2),
             create_edge_onelevel( N1,Attr2,Splite)) ; true ),
        create_nodes(N,AllAttr,T).

create_nodes(_,_,[]) :- !.
create_nodes(_,[],_) :- !.
```

The predicates init and getnode initialize the tree structure. The tree-based knowledge induction process starts when the main module invokes the predicate create_edge_onelevel to build a decision tree one level at a time. After the complete tree structures are created, the predicates addKnowledge and selectRule are invoked to compute probability along each tree branch to generate probabilistic rules, and then select only rules that could occur at the probability level higher than the specified threshold. User can control minimum probability level through the interactive interface of the system (Figure 5).

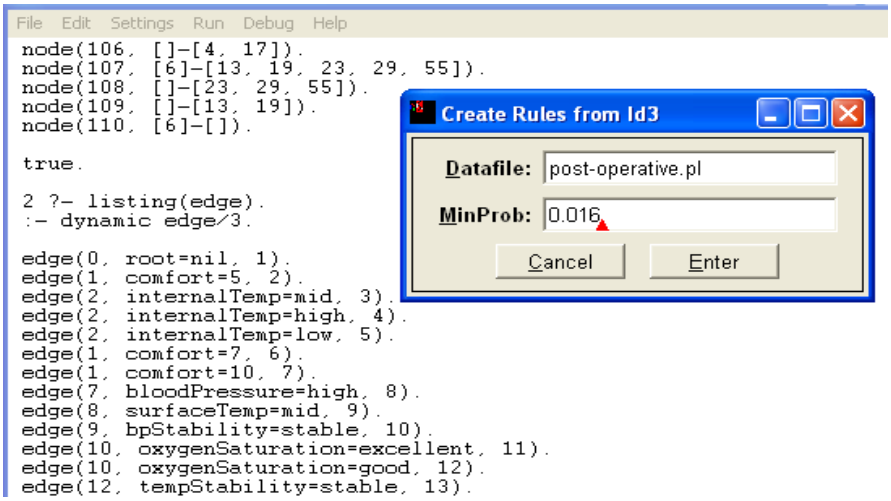

Figure 5. Interactive User Interface and the Generated Tree Structures

4.3. Prolog Implementation: Automatic Inference Engine and Knowledge Base Creation

The probabilistic rules are then written in file to serve as a knowledge base of the inductive expert system. A rule-based inference engine is also automatically created as follows (the screenshot is shown in Figure 6):

```
expertshell :-
        greeting,
        repeat,
        write('expert-shell> '),
        read(X),
        do(X),
        (X == quit;X == 99),
        writeln('Goodbye'), !.
greeting :-
        write('This is the Easy Expert System shell.'), nl,
        native_help.
do(help) :- native_help, !.
do(load) :- load_kb, !.
do(solve) :- solve, !.
do(why):-why,!.
do(quit). do(99).
do(X) :-    write(X),
        write(' is not a legal command.'), nl, fail.
native_help :-
```

```
            write('Type help. load. solve. why.  quit. or 99.'),nl,
            write('at the prompt.'), nl.

load_kb :-  write('Enter file name in single quotes (ex. ''1.knb''.): '),
            read(F),
            reconsult(F).

solve :-    retractall(known( _) ),
            retractall(answer(_,_)),
            top_goal(X,V),
            format('The answer is __~w__ with probability ~w',[X,V]),
            assert(answer(X,V)), nl.

solve :-    write('No answer found.'), nl.

menuask(Pred,Value,Menu) :-
            menuask(Pred,Menu),
            atomic_list_concat([Pred,'(',Value,')'],X),
            term_to_atom(T,X),known(T),!.

menuask(Pred,_) :-
            atomic_list_concat([Pred,'(','_',')'],X),
            term_to_atom(T,X),known(T),!.

menuask(Attribute,Menu):-
            nl,write('What is the value for '),
            write(Attribute),write('?'), nl,
            addchoice(Menu,MenuRes), writeln(MenuRes),
            write('Enter the  choice> '),
            read(C),
            member(C-V,MenuRes),
            (C=99 -> abort ; true),
            atomic_list_concat([Attribute,'(',V,')'],X),
            term_to_atom(T,X),
            asserta(known(T)).

why :- answer(A,V),
            format('~nThe answer is ...~w... with probability =
            ~w.~n',[A,V]),
            findall( X , known(X),Result),
            writeln('The known storage are'),
            writeln(Result).

addchoice(X,Res) :-
           length(X,Len),
           numlist(1,Len,NumL),
           map(NumL,X,Res).
```

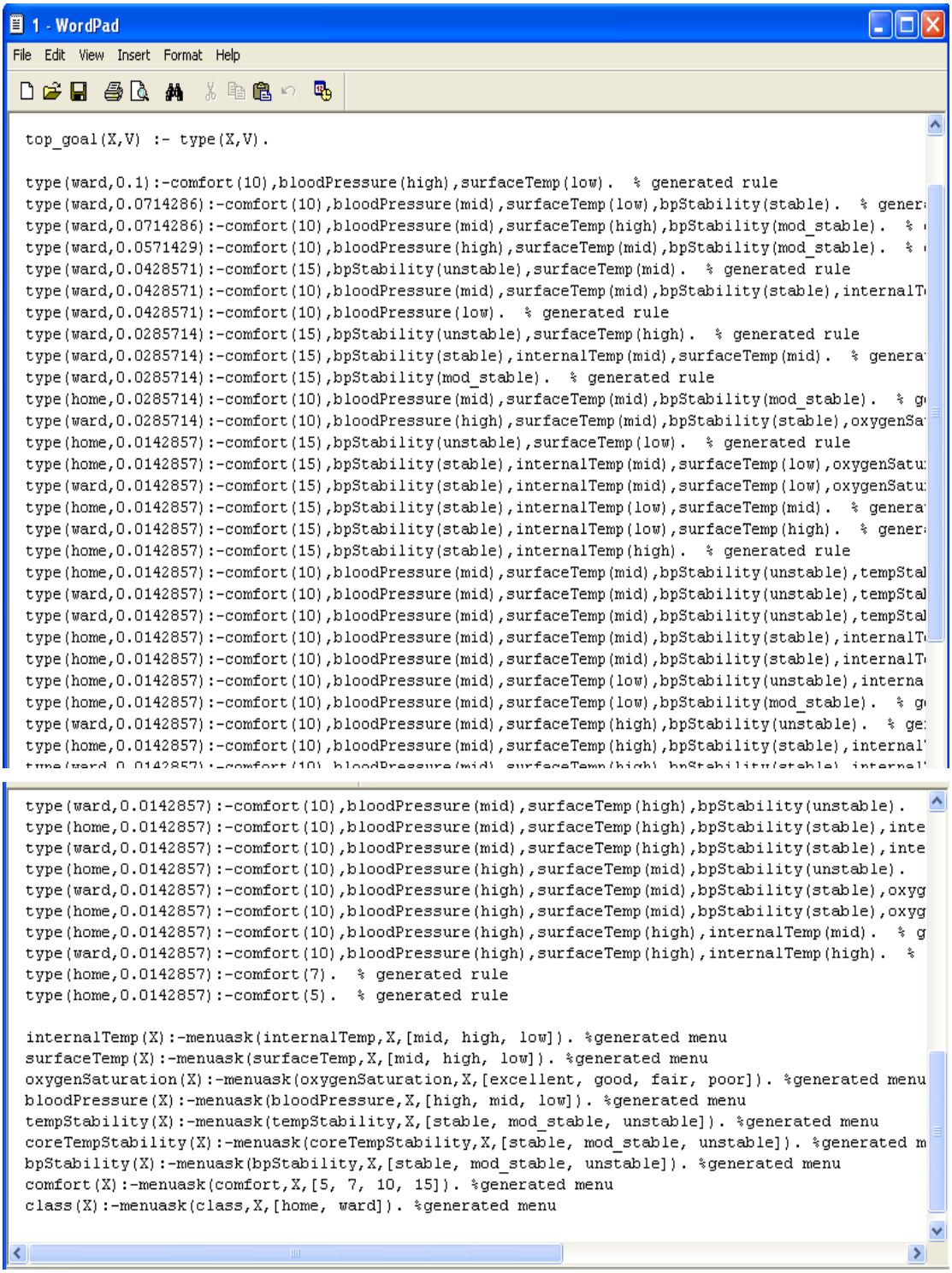

Figure 6. All Inductive Rules in the Knowledge Base that are Created with the Minimum Coverage Threshold 0.001

4.4. Performance Evaluation Result on Post-operative Data

We evaluate correctness of the induced probabilistic knowledge by dividing data set into two subsets: a training set containing 70 patient records, and a test set containing 16 patient records. A training set is used in the knowledge induction phase. The training results, which are knowledge base and inference rules, are then tested by the test set.

Figure 7 illustrates the test process of data instance number 71 by the expert system shell. The recommendation given by the inductive expert system is that the patient should be sent to the general ward with probability (or confidential level) 0.0348837. The actual diagnosis made by the doctor is also admission to the general ward. Therefore, the recommendation given by the inductive expert system for the specific case is correct.

```
%   * *** Test Data
%         =========
instance(71, class=ward, [internalTemp=mid, surfaceTemp=mid,
                          oxygenSaturation=excellent, bloodPressure=high,
                          tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
```

```
% 1.knb
% for expert shell. --- written by Postprocess
% top_goal where the inference starts.

top_goal(X,V) :- type(X,V).

type(ward,0.1):-comfort(10),bloodPressure(high),surfaceTemp(low).  % generated rule
type(ward,0.0714286):-comfort(10),bloodPressure(mid),surfaceTemp(low),bpStability(stable).  % generated rule
type(ward,0.0714286):-comfort(10),bloodPressure(mid),surfaceTemp(high),bpStability(mod_stable).  % generated rule
type(ward,0.0571429):-comfort(10),bloodPressure(high),surfaceTemp(mid),bpStability(mod_stable).  % generated rule
type(ward,0.0428571):-comfort(15),bpStability(unstable),surfaceTemp(mid).  % generated rule
type(ward,0.0428571):-comfort(10),bloodPressure(mid),surfaceTemp(mid),bpStability(stable),internalTemp(mid),tempStal
type(ward,0.0428571):-comfort(10),bloodPressure(low).  % generated rule
type(ward,0.0285714):-comfort(15),bpStability(unstable),surfaceTemp(high).  % generated rule
type(ward,0.0285714):-comfort(15),bpStability(stable),internalTemp(mid),surfaceTemp(mid).  % generated rule
type(ward,0.0285714):-comfort(15),bpStability(mod_stable).  % generated rule
type(home,0.0285714):-comfort(10),bloodPressure(mid),surfaceTemp(mid),bpStability(mod_stable).  % generated rule
type(ward,0.0285714):-comfort(10),bloodPressure(high),surfaceTemp(mid),bpStability(stable),oxygenSaturation(excelle
type(home,0.0142857):-comfort(15),bpStability(unstable),surfaceTemp(low).  % generated rule
type(home,0.0142857):-comfort(15),bpStability(stable),internalTemp(mid),surfaceTemp(low),oxygenSaturation(good).  %
```

```
File  Edit  Settings  Run  Debug  Help
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic). or ?- apropos(Word).

1 ?- expertshell.
This is the Easy Expert System shell.
Type  help.    load.    solve.    why.     quit. or  99.
at the prompt.
expert-shell> load.
Enter file name in single quotes (ex. '1.knb'.): '1.knb'.
% 1.knb compiled 0.01 sec, 5,728 bytes
expert-shell> solve.

What is the value for comfort?
[1-5, 2-7, 3-10, 4-15, 99-exitShell]
Enter the  choice> 3.

What is the value for bloodPressure?
[1-high, 2-mid, 3-low, 99-exitShell]
Enter the  choice> 1.

What is the value for surfaceTemp?
[1-mid, 2-high, 3-low, 99-exitShell]
Enter the  choice> 1.

What is the value for bpStability?
[1-stable, 2-mod_stable, 3-unstable, 99-exitShell]
Enter the  choice> 1.

What is the value for oxygenSaturation?
[1-excellent, 2-good, 3-fair, 4-poor, 99-exitShell]
Enter the  choice> 1.
The answer is __ward__ with probability 0.0348837
expert-shell> why.

The answer is ...ward... with probability = 0.0348837.
The known storage are
[oxygenSaturation(excellent), bpStability(stable), surface
Temp(mid), bloodPressure(high), comfort(10)]
expert-shell>
```

Figure 7. The Process of Correctness Testing of the Automatic Inductive Expert System

The performance of an inductive expert system is also tested against other machine learning methods: ID3, Prism, and Neural network. The experimental result (Figure 8) confirms that our inductive expert system can generate probabilistic decision rules as good as the data model obtained from the Neural network method, and better than the results produced by the ID3 and the Prism methods. The decisions recommended by our inductive expert system, as compared to the correct diagnosis of medical doctor as well as other decisions made by the data mining programs are summarized in Table 1.

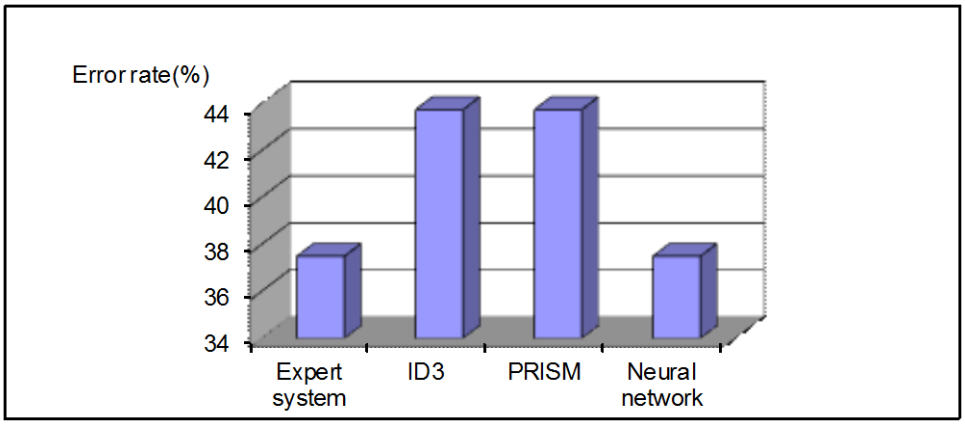

Figure 8. Comparison on Prediction Error Rate of the Inductive Expert System against the ID3, Prism, and Neural Network Methods

Table 1. Recommendations by Doctors, Inductive Expert System and Other Mining Programs

Test case	Doctor's diagnosis	Recommendation from inductive expert system	Decision made by decision tree (ID3)	Decision made by decision rules (PRISM)	Decision made by neural network
1	Ward	**Ward**	**Ward**	**Ward**	**Ward**
2	Ward	Ward	Ward	Ward	Ward
3	Ward	Ward	Unclassified	Home	Ward
4	Ward	**Home**	**Home**	**Home**	**Home**
5	Ward	**Ward**	**Ward**	**Unclassified**	**Ward**
6	Home	Ward	Ward	Home	Home
7	Ward	**Home**	**Home**	**Home**	**Home**
8	Ward	**Ward**	**Ward**	**Ward**	**Ward**
9	Home	No answer found	Ward	Ward	Ward
10	Ward	Home	Home	Home	Home
11	Ward	Ward	Ward	Unclassified	Ward
12	Ward	Ward	Ward	Ward	Ward
13	Home	**Ward**	**Ward**	**Ward**	**Ward**
14	Ward	**Ward**	**Ward**	**Ward**	**Ward**
15	Ward	Ward	Ward	Ward	Ward
16	Home	Ward	Ward	Ward	Ward

4.5. Performance Evaluation Result on Breast-cancer Recurrence Data

As the data set has been changed to the breast-cancer recurrence data, the induced knowledge base contents, as well as the inference rules of the inductive expert system have been changed accordingly. The new knowledge base contents are illustrated in Figure 9.

Figure 9. The Knowledge Base Contents that are Created from the Breast-Cancer Recurrence Data with the Minimum Coverage Threshold 0.001

The performance of an inductive expert system created from the breast-cancer recurrence data is also tested against other machine learning methods: ID3, Prism, and Neural network. The experimental results are shown in Figure 10. The decisions recommended by our inductive expert system, as compared to the correct diagnosis of medical doctor as well as other decisions made by the data mining programs are summarized in Table 2.

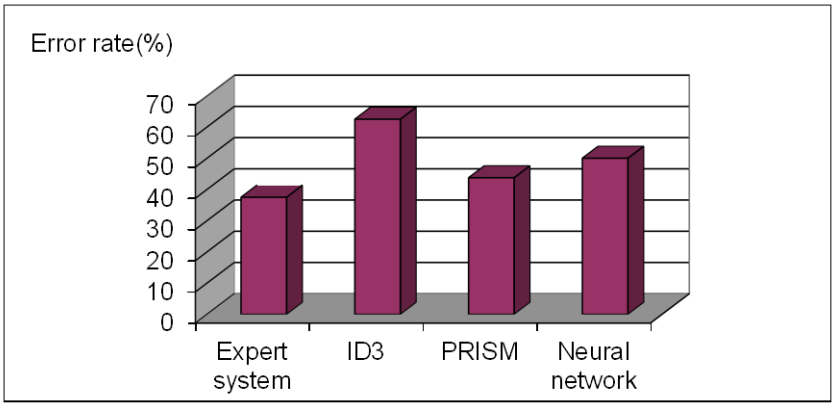

Figure 10. Prediction Error Rate of the Inductive Expert System Created from the Breast-cancer Recurrence Data, Compared Against the ID3, Prism, and Neural Network Methods

Table 2. Diagnoses by Doctors, Inductive Expert System and Other Data Mining Programs

Test case	Doctor's diagnosis	Diagnosis by inductive expert system	Diagnosis by decision tree (ID3)	Diagnosis by decision rules (PRISM)	Diagnosis by neural network
1	No	**No**	Yes	**No**	Yes
2	No	Yes	**Yes**	Yes	**Yes**
3	Yes	**Yes**	**No**	**No**	**No**
4	Yes	No answer found	**No**	**No**	**Yes**
5	No	Yes	Unclassified	**No**	No
6	No	No answer found	Yes	No	No
7	Yes	No	**No**	No	No
8	No	**No**	**Yes**	**Yes**	**No**
9	Yes	No	**No**	**Yes**	No
10	No	Yes	**No**	No	Yes
11	No	**No**	No	No	**No**
12	No	Yes	Yes	Yes	Yes
13	Yes	**Yes**	**No**	No	**Yes**
14	No	No answer found	**No**	**No**	No
15	No	**No**	Unclassified	No	Yes
16	Yes	Yes	**Yes**	Yes	**Yes**

From the experimental results on post-operative patients and breast-cancer recurrences data, it can be noticed that our inductive expert system as a medical consultant predicts diagnosis results at the lowest error rate as compared to the other data mining programs. Its general error rate, however, is higher than 30%. Nonetheless, its false negative error is less than 20% (as shown in Table 2).

Table 2. False Negative Error Evaluation of the Inductive Expert System and the other Data Mining Programs

False negative error	Inductive Expert system	ID3	PRISM	Neural network
Data: post-operative patients Error characteristics: Diagnosis that should be sending to 'ward' is wrongly put as sending 'home'	Error = 3/16 = 18.75%	Error = 3/16 = 18.75%	Error = 4/16 = 25%	Error = 3/16 = 18.75%
Data: breast-cancer recurrences Error characteristics: Diagnosis that should be cancer recurrence is wrongly put as no cancer recurrence	Error = 2/16 = 12.50%	Error = 5/16 = 31.25%	Error = 4/16 = 25%	Error = 3/16 = 18.75%

5. Conclusion

Biomedical informatics deals with biomedical information, its structure, acquisition and optimal use for problem solving and decision making. Medical knowledge discovery is a research area that employing machine learning techniques to acquire knowledge from health examination and clinical data. Knowledge extraction from huge amount of health databases is expected to ease the medical decision-making process. The ultimate goal of knowledge extraction is to generate the most accurate and useful knowledge and represent it in an understandable format. Such goal is, however, difficult to accomplish due to the learning complexity of knowledge induction methods and the nonconformity of the database contents. Most of the time knowledge discovery from medical databases results in reporting large number of irrelevant knowledge. We thus focus our study on this issue and devise a technique to extract a limited number of knowledge that is most likely relevant to the specific domain.

In medical domains, interpretability of results is an important feature of the data analysis tool. Medical practitioners need a system that can produce accurate results in an understandable form. Therefore, knowledge represented as rules has been widely used for knowledge discovery in medical applications. Nevertheless, in medical applications learning techniques tend to generate a lot of rules. Too many rules, some are redundant and uninteresting, cause problems to the medical practitioners because a truly relevant one can be easily overlooked. We thus propose a rule induction method based on the decision-tree structure that adopts the probability concept to select the

most probable applicable rules. The selected rules are then automatically transformed to be the knowledge and the inference engine in the medical expert system to support decision making. The experimentation on our inductive expert system confirms a good performance of the system on recommending patient discharge decision after an operation and breast cancer analysis. Direct application of medical probabilistic knowledge base is for medical related decision-making. Other indirect but obvious application of such knowledge is to pre-process other data sets by grouping it into focused subset containing only relevant data instances.

Acknowledgement

This work has been supported by grants from the National Research Council of Thailand (NRCT) and Suranaree University of Technology via the funding of Data Engineering Research Unit.

References

[1] Bojarczuk CC, Lopes HS, Freitas AA, and Michalkiewicz EL, "A constrained-based syntax genetic programming system for discovering classification rules: Application to medical data sets", Artificial Intelligence in Medicine, vol. 30, **(2004)**, pp. 27-48.

[2] Bratsas C, Koutkias V, Kaimakamis E, Bamidis PD, Pangalos GI, Maglaveras N, "KnowBaSICS-M: An ontology-based system for semantic management of medical problems and computerized algorithmic solutions", Computer Methods and Programs in Biomedicine, vol. 83, **(2007)**, pp. 39-51.

[3] Correia F, Kon R, Borboleta, "A mobile telehealth system for primary homecare", In Proc. ACM Symposium on Applied Computing, **(2008)**, pp.1343-1347.

[4] Exarchos TP, Tsipouras MG, Exarchos CP, Papaloukas C, Fotiadis DI, Michalis LK, "A methodology for the automatic creation of fuzzy expert systems for ischaemic and arrhythmic beat classification based on a set of rules obtained by a decision tree", Artificial Intelligence in Medicine, vol. 40, (2007), pp. 187-200.

[5] Frank A, Asuncion A, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml], Irvine, University of California, School of Information and Computer Science, **(2010)**.

[6] Ghazavi S, Liao T, "Medical data mining by fuzzy modeling with selected features", Artificial Intelligence in Medicine, vol. 43, no. 3, **(2008)**, pp. 195-206.

[7] Horng J, Wu L, Liu B, Kuo J, Kuo W, Zhang J, "An expert system to classify microarray gene expression data using gene selection by decision tree", Expert Systems with Applications, vol. 36, **(2009)**, pp. 9072-9081,

[8] Huang M, Chen M, Lee S, "Integrating data mining with case-based reasoning for chronic disease prognosis and diagnosis", Expert Systems with Applications, vol. 32, **(2007)**, pp. 856-867.

[9] Kononenko I, "Machine learning for medical diagnosis: History, state of the art and perspective", Artificial Intelligence in Medicine, vol. 1, **(2001)**, pp. 89-109.

[10] Kumar KA, Singh Y, Sanyal S, "Hybrid approach using case-based reasoning and rule-based reasoning for domain independent clinical decision support in ICU", Expert Systems with Applications, vol. 36, **(2009)**, pp. 65-71.

[11] Lander ES et al, "Initial sequencing and analysis of the human genome", Nature, vol. 409, **(2001)**, pp. 860-921.

[12] Maojo V, Kulikowski C, "Bioinformatics and medical informatics: Collaborations on the road to genomic medicine?", J American Medical Informatics Association, vol. 10, no. 6, **(2003)**, pp. 515-522.

[13] Maojo V, Iakovidis I, Martic-Sanchez F, Crespo J, Kulikowski C, "Medical informatics and bioinformatics: European efforts to facilitate synergy", J Biomedical Informatics, vol. 34, **(2001)**, pp. 423-427.

[14] Miller RA, Pople HE, Myers JD, "INTERNIST-1, An experimental computer-based diagnostic consultant for general internal medicine", New England J Medicine, vol. 307, no. 8, **(1982)**, pp. 468-476.

[15] Nadathur G, Miller D, "Higher-order Horn clauses", J ACM, vol. 37, **(1990)**, pp. 777-814.

[16] Nguyen D, Ho T, Kawasaki S, "Knowledge visualization in hepatitis study', In Proc Asia-Pacific Symposium on Information Visualization, **(2006)**, pp.59-62.

[17] Palaniappan S, Ling C, "Clinical decision support using OLAP with data mining", Int J Computer Science and Network Security, vol. 8, no. 9, **(2008)**, pp. 290-296.

[18] Pandey B, Mishra RB, "Knowledge and intelligent computing system in medicine", Computers in Biology and Medicine, vol. 39, **(2009)**, pp. 215-230.

[19] Quinlan JR, "Induction of decision trees", Machine Learning, vol. 1, **(1986)**, pp. 81-106.

[20] Roddick J, Fule P, Graco W, "Exploratory medical knowledge discovery: Experiences and issues", ACM SIGKDD Explorations Newsletter, vol. 5, no. 1, **(2003)**, pp. 94-99.

[21] Sahama T, Croll P, "A data warehouse architecture for clinical data warehousing", In Proc 12th Australasian Symposium on ACSW Frontiers, **(2007)**, pp. 227-232.

[22] Schwartz WB, "Medicine and the computer: The promise and problems of changes", New England J Medicine, vol. 283, **(1970)**, pp. 1257-1264.

[23] Shillabeer A, Roddick JF, "Establishing a lineage for medical knowledge discovery", In Proc 6th Australasian Conf Data Mining and Analytics, **(2007)**, pp.29-37.

[24] Shortliffe EH, "Computer-based medical consultations", MYCIN, Elsevier, **(1976)**.

[25] Shortliffe EH, Cimino JJ, "Biomedical informatics: Computer applications in health care and biomedicine", 3rd Edition, Springer, **(2006)**.

[26] Stead W, "The challenge of bridging between disciplines', J American Medical Informatics Association, vol. 8, no. 1, **(2001)**, pp. 105.

[27] Truemper K, "Design of logic-based intelligent systems", John Wiley & Sons, **(2004)**.

[28] Zhou Z, Jiang Y, "Medical diagnosis with C4.5 rule preceded by artificial neural network ensemble", IEEE Trans Information Technology in Biomedicine, vol. 1, **(2003)**, pp. 37-42.

[29] Zhuang Z, Churilov L, Burstein F, "Combining data mining and case-based reasoning for intelligent decision support for pathology ordering by general practitioners", European J Operational Research, vol. 195, no. 3, **(2009)**, pp. 662-675.

Authors

Kittisak Kerdprasop is an associate professor at the school of computer engineering, Suranaree University of Technology, Thailand. He received his bachelor degree in Mathematics from Srinakarinwirot University, Thailand, in 1986, master degree in computer science from the Prince of Songkla University, Thailand, in 1991 and doctoral degree in computer science from Nova Southeastern University, USA., in 1999. His current research includes Data mining, Artificial Intelligence, Functional and Logic Programming Languages, Computational Statistics.

Nittaya Kerdprasop is an associate professor at the school of computer engineering, Suranaree University of Technology, Thailand. She received her B.S. from Mahidol University, Thailand, in 1985, M.S. in computer science from the Prince of Songkla University, Thailand, in 1991 and Ph.D. in computer science from Nova Southeastern University, USA, in 1999. She is a member of ACM and IEEE Computer Society. Her research of interest includes Knowledge Discovery in Databases, Artificial Intelligence, Logic Programming, Deductive and Active Databases.