Visual Knowledge Mining and Utilization in the Inductive Expert System

Nittaya Kerdprasop and Kittisak Kerdprasop

***Abstract*—**Advances in computer graphics and the human-machine visual systems have made visualization become an important tool in current data exploration and analysis tasks. Visual data mining is the combination of visualization and data mining algorithms in such a way that users can explore their data and extract the models in an interactive way. Existing visual data mining tools allow users to interactively control the three main steps of data mining: input data, explore data distribution, and extract patterns or models from data. In this paper, we propose a framework to extend these visually controlled steps to the level of model deployment. We demonstrate in this paper that both model induction and model deployment can be done through the visual method using the KNIME and Win-Prolog tools for knowledge acquisition and knowledge deployment, respectively. Model deployment presented in this paper is the utilization of induced data model as an inductive knowledge source for the inductive expert system, which is the next generation of knowledge base systems that integrate automatic learning ability in their knowledge acquisition part.

***Keywords*—** Visual data mining, Inductive learning, Inductive expert system, Visual logic program.

I. Introduction

VISUAL data mining is an automatic and intelligent data analysis technique that utilizes visualization as a means to communicate between user and the computer to explore data and to extract hidden patterns from stored databases [33], [18]. The main benefit of visualization is that it allows easy understanding for novice users and it is also natural to human perception [15], [28], [36]. Recent trend in intelligent manufacturing and other engineering fields [8], [13], [17], [32] is to apply data mining techniques to automatically identify patterns and causal relationships that are too obscure and unobvious to be detected by human's eyes.

Applying data mining technique to high dimensional and large amount data is however not a straightforward task because the induced patterns are normally low accurate if the input data are not well prepared or not in an appropriate form. Numerous available learning algorithms and many data preparation techniques supported by most data mining systems are also a hindrance to users who are unfamiliar with the knowledge discovery process.

We thus illustrate in this paper a natural way to do data mining through visualization. We also propose a semi-automatic technique to transfer the data mining output to be a knowledge base content in the inductive expert system.

The main characteristic of current expert systems is the separation of a knowledge base that may be changed from one application to another from the inference engine that still remains the same across applications. The delay in the development of many expert systems is due to the difficulty in acquiring and eliciting knowledge from the human domain experts [11], [20].

The concept of inductive expert system is thus been devised to overcome such bottleneck by incorporating automatic knowledge acquisition module in the system. According to this new concept, knowledge can now be induced or learned in an automatic way from archived databases that are normally available in most organizations. In this paper, we propose an architecture of the inductive expert system that includes the knowledge mining engine part to automatically forming expert rules from the stored data. The learned knowledge can be visually transformed to be the knowledge base and automatically encoded as inference rules of the inductive expert system. We show in this paper that the processes of knowledge mining and knowledge acquisition in the inductive expert system can be done through the support of available visual tools, namely the KNIME system [7] and the Win-Prolog [23].

II. Preliminaries and Related Work

A. The Visual Data Mining Tools

Data mining is the search for useful patterns that normally are difficult to be recognized by human's eyes due to the large amount of data items stored in the databases. Most algorithms used in the data mining software construct a mathematical model from the data instances for the purpose of describing common patterns or predicting some unknown attribute values

Manuscript received May 20, 2014: Revised version received July 4, 2014. This work was supported by grants from the National Research Council of Thailand (NRCT) and Suranaree University of Technology through the funding of Data Engineering Research Unit.

N. Kerdprasop is an associate professor and the director of Data Engineering Research Unit, School of Computer Engineering, Suranaree University of Technology, 111 University Avenue, Muang District, Nakhon Ratchasima 30000, Thailand (phone: +66-44-224-432; fax: +66-44-224-602; e-mail: nittaya@sut.ac.th).

K. Kerdprasop is with the School of Computer Engineering and Data Engineering Research Unit, Suranaree University of Technology, Nakhon Ratchasima, Thailand (e-mail: KittisakThailand@gmail.com).

in the future cases. Model construction is a core step of data mining process. To train the learning algorithm to construct an accurate predictive model needs some parameter tuning, which is absolutely not an easy task for casual users. Therefore, modern data mining software provides some graphical interfaces to help users in controlling the process [4], [10], [21].

Visualization can be used in several stages of data mining: data exploration, analysis, and knowledge representation. The existing visual exploring and analytic tools [2], [3], [30] are fast and effective enough for the interactive induction of hidden knowledge. In this work, we adopt the KNIME visual data mining system [7] for knowledge mining.

B. Expert System and Data Mining

Since the release of DENDRAL in the 1960s from the Stanford Heuristic Programming Project [22] as the first practical knowledge-driven program, expert systems have enormously proliferated and been applied to all areas of computer-based problem solving [34]. The inventors of DENDRAL system have introduced the novel and important concept of knowledge base separation in that the content of knowledge could be added and refined independently from the program module. This module is called the inference engine responsible for interpreting and using the knowledge. The loosely coupling of a knowledge base and an inference engine is an influential concept to all successor rule-based expert systems such as MYCIN [35], INTERNIST-1 [26], and many others [14], [16].

Since the 1980s expert systems, also called knowledge-based systems, have shifted from the medical and scientific application domains to various areas. In manufacturing, mechanical analysis, and other engineering applications, rule-based expert systems are commonly applied to solve optimization problems, diagnose equipment failures, plan manufacturing scheduling, and other stages of the manufacturing process [6].

The increasing popularity of rule-based expert systems is due to the simplicity of the *if-then* rules that are easy to comprehend by humans. Many expert system tools such as Clips and Jess are available as a rule engine to facilitate rule generation for a knowledge base. These tools help facilitating the part of knowledge representation, but knowledge acquisition and elicitation are still the labor-intensive tasks for most knowledge engineers.

Modern expert system development process has thus moved toward the automating methodology by applying intelligent knowledge extraction techniques [12], [29]. Such intelligent techniques can be acquired through the machine learning and data mining technologies. There have been increasing numbers of research work attempting to apply learning techniques to automatically extract end elicit knowledge [1], [19], [27], [37]. These attempts have pushed the current expert system technology to the next generation of an inductive expert system in the sense that besides the knowledge base and the inference engine, the system now includes the learning component.

The research work presented in this paper takes the same direction as most researchers in an attempt to automate knowledge extraction and elicitation with machine learning and data mining techniques. Our work, however, is different from others in that not only proposing an architecture of the learnable inductive expert system, but we also cover the knowledge mining from existing databases, knowledge transfer as a set of rules to be stored in the knowledge base, and knowledge reasoning through a logic-based inference engine. The process of knowledge mining and knowledge utilization have been demonstrated through the adoption of existing visual tools.

III. Bridging Mining Model with the Knowledge Acquisition of Inductive Expert System

Knowledge mining [24], [25] is the discovery of hidden knowledge stored possibly in various forms and places in large data repositories. The whole process of knowledge mining works around data, meta-data, and previously discovered patterns. It can be conceptually shown as in Figure 1.

The initial step of knowledge mining focuses on setting the mining goal which can be achieved through understanding the task objectives and organization requirements. Problem defining is important because it will guide activities in subsequent steps to collect only relevant data, to do mining with appropriate algorithm, and to keep only pertinent and actionable knowledge.

The second step covers all activities necessary for preparing high quality data suitable for mining algorithm. This data preparation step includes collecting data from multiple sources, transforming the data format, selecting data representatives with minimum but sufficient attributes. Data preparation is typically time consuming and likely to be performed iteratively. Meta-data and background knowledge are kinds of supportive information that can be applied in this step.

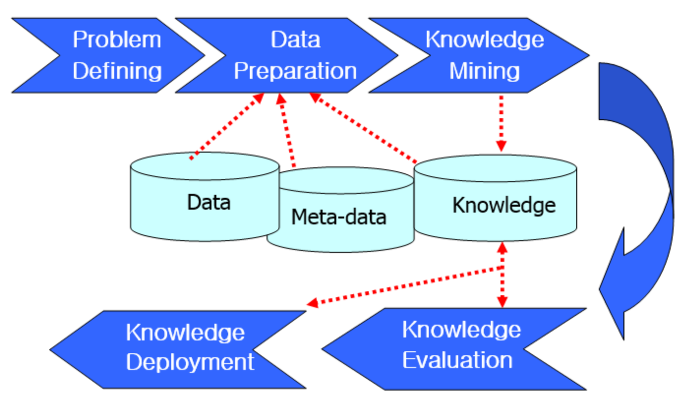


Fig. 1 The knowledge mining process

The third step is knowledge mining which is the search and extraction of interesting patterns (local generalized structures) or models (global generalized structures) from data. Such patterns and models are called knowledge. This step is the backbone of the knowledge mining process.

The fourth step is for evaluating accuracy, significance, and interestingness of the discovered knowledge based on some threshold values. The accurate, significant, and interesting knowledge is finally fed to the deployment step to be actionable information for the organization, or it can even be put back into the repositories to be background knowledge for other knowledge mining tasks.

We design (in Figure 2) an architecture of the inductive expert system to include the knowledge engine facility. Knowledge induction phase is the back-end of the system responsible for acquiring and discovering new and useful knowledge. Usefulness is to be validated at the final step by human experts. Discovered knowledge is stored in the knowledge base to be applied to solve new cases in knowledge inferring phase which is the front-end of the proposed system.

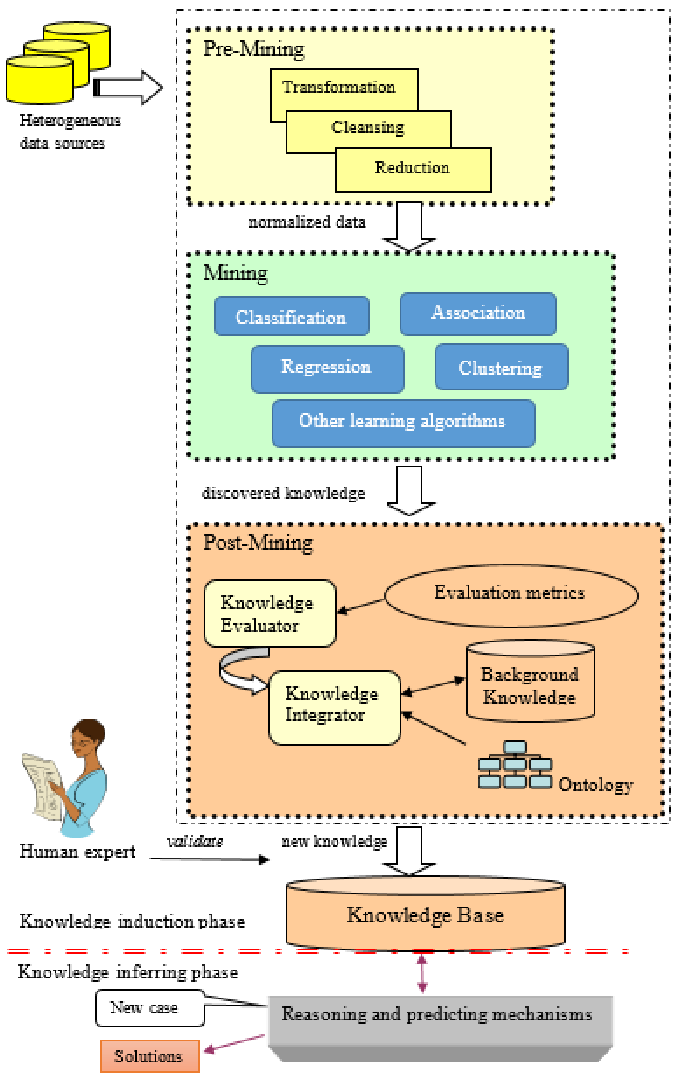


Fig. 2 Architecture of an inductive expert system

The knowledge induction phase is comprised of three main modules: pre-mining, mining, post-mining. The term *mining* means automatic learning of patterns or models from specific data. *Pattern* is an expression describing a subset of the data. For example, $f(x) = 3x^2 + 3$ is a pattern induced from a given 2-dimensional dataset {(0,3), (1,6), (2,15), (3,30), (4,51)}, whereas the term *model* refers to a function representing the source that generates the data. For this example, the model is $f(x) = ax^2 + b$. In his paper we refer to both patterns and models as new knowledge discovered from data sources.

The pre-mining module performs data preparation tasks such as

- locate and access relevant data sets,
- transform the data format,
- clean the data if there exists noise and missing values,
- reduce the data to a reasonable and sufficient size with only relevant attributes.

The mining module performs mining tasks including classification, regression, clustering, association, and other learning task. The post-mining module is composed of two main components: knowledge evaluator and knowledge integrator. These components perform major functionalities aiming at a feasible deployment of the discovered knowledge.

Knowledge evaluator involves evaluation, based on corresponding measurement metrics, of the mining results. Knowledge integrator examines the induced patterns to remove redundant knowledge. Ontology has also been applied at this step to provide essential semantics regarding the domain problems.

IV. Demonstration and Results

A. Data Set

The purpose of this experimentation is to illustrate the proposed automatic knowledge base creation method with real data. We use a car evaluation data set [9], [38] obtain from the UCI Machine Learning Repository [5]. In this data set, each car is to be evaluated its acceptability level as either very good (vgood), good, acceptable (acc), or unacceptable (unacc).

The car acceptability has been evaluated from the six attributes: the buying price (buying), price of maintenance (maint), number of doors (doors), capacity in terms of persons to carry (persons), the size of luggage boot (lug_boot), and the estimated safety of the car (safety). This data set has 1728 data instances. Examples of data instances are shown in Table1.

This data set has been used in this paper as a training set for constructing a conceptual model of car acceptability decision based on the price and other technical characteristics. Class distribution of each acceptability levels is as follows: unacc = 70.02%, acc = 22.22%, good = 3.99%, and v-good = 3.76%.

Table 1. Some instances of a car evaluation data set

buying	maint	doors	persons	lug_ boot	safety	class
vhigh	vhigh	2	2	small	low	unacc
high	high	4	more	small	low	unacc
vhigh	med	2	4	big	high	acc
high	high	4	4	big	med	acc
med	low	4	more	small	high	good
med	low	4	4	big	high	vgood

B. Visual Data Mining and Its Result

The car evaluation data set has been mined with the visual data mining tool named KNIME [7]. The visual data mining process is illustrated in Figure 3, and the mining result as decision tree is shown in Figure 4. The learning algorithm used in this demonstration is the decision tree induction algorithm [31] because of its efficiency. Moreover, the structure of the induced tree is appropriate for generating reasoning and explanation part in the expert system shell.

The steps graphically shown in Figure 3 are the process to generate a decision tree model. The first step is to read the input data; this can be done through the icon ‘File Reader’. We then partition the input data into the training set and the test set (through the ‘Partitioning’ icon). The model induction part is accomplished through the use of ‘Decision Tree Learner’ icon. The output of this process is the learned knowledge to be used by the expert system shell. The ‘Decision Tree Predictor’ and ‘Scorer’ icons are used only for evaluating the accuracy of the tree model. Evaluation result of the model accuracy is shown in Figure 5.

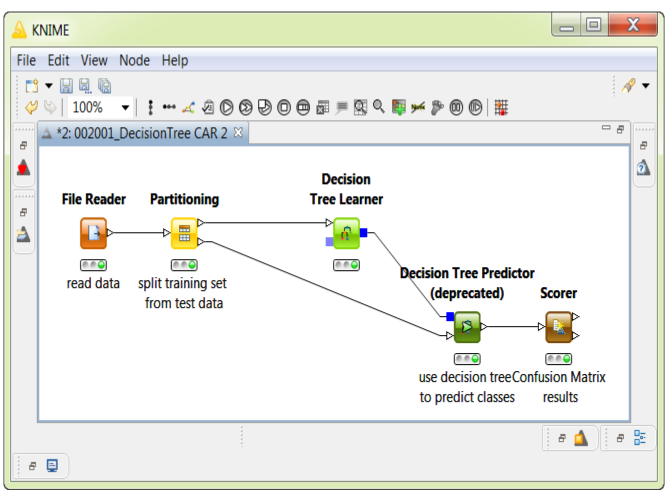


Fig. 3 Data mining process through the connection of visual icons in KNIME system

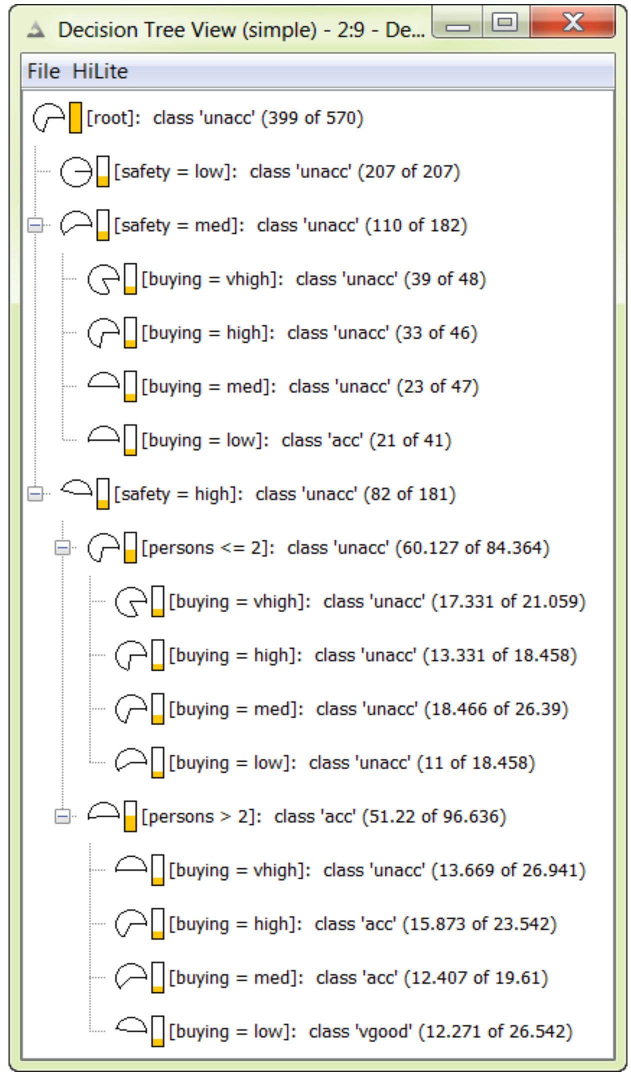
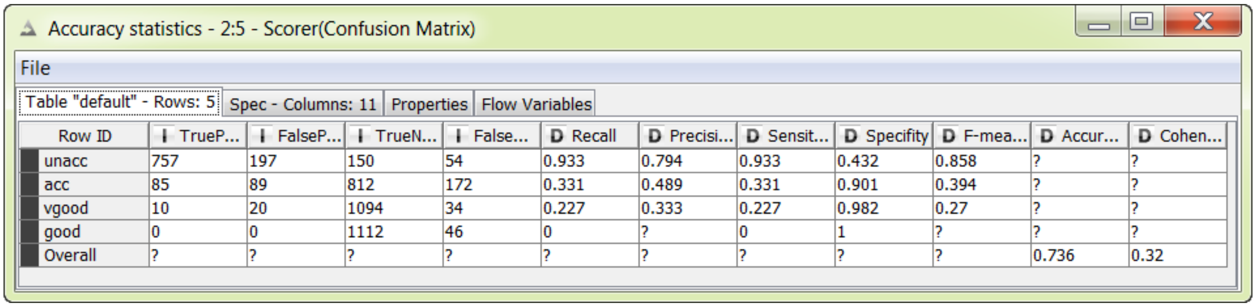


Fig. 4 A decision tree model to classify acceptability of a car as unacc/acc/vgood

C. Visual Knowledge Acquisition and Consultation

The induced knowledge as a decision tree is subsequently to be transformed into a format of decision rules by the Win-Prolog visual tool [23]. The rule generation can be done automatically. These rules are to be used by the inference engine for giving recommendation to users. These rules are also capable of giving explanation when requested by the users.

The knowledge acquisition starts when the decision tree has been built and output by the KNIME tool. The output from KMINE (Figure 4) has to be manually transformed into a format understandable by the Win-Prolog. The transformation is however easy via the support of visualization tools. The transformed rules are graphically shown in Figure 6. The meaning of the two formats is the same; only graphical representation is slightly different.


Accuracy statistics - 2:5 - Scorer(Confusion Matrix)

File

Table "default" - Rows: 5 | Spec - Columns: 11 | Properties | Flow Variables

Row ID	TrueP...	FalseP...	TrueN...	False...	Recall	Precisi...	Sensit...	Specifity	F-mea...	Accur...	Cohen...
unacc	757	197	150	54	0.933	0.794	0.933	0.432	0.858	?	?
acc	85	89	812	172	0.331	0.489	0.331	0.901	0.394	?	?
vgood	10	20	1094	34	0.227	0.333	0.227	0.982	0.27	?	?
good	0	0	1112	46	0	?	0	1	?	?	?
Overall	?	?	?	?	?	?	?	?	?	0.736	0.32

Fig. 5 Accuracy evaluation results of the induced decision tree model

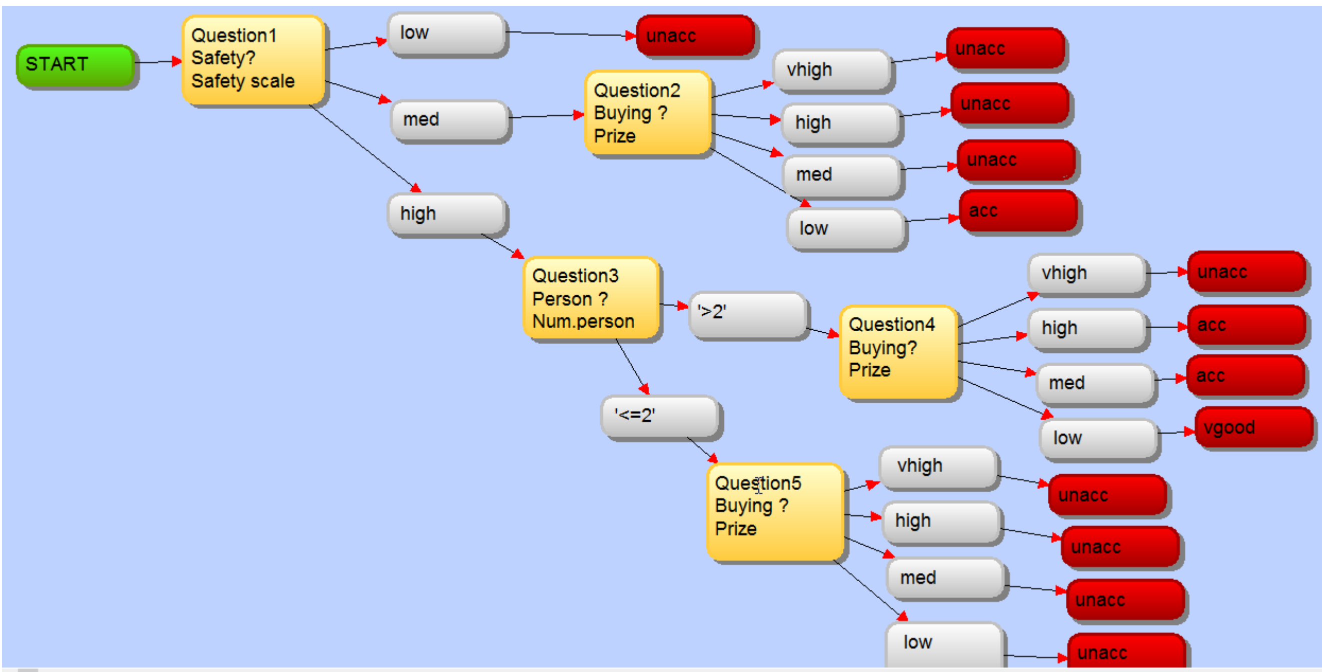


Fig. 6 A decision tree model in a Win-Prolog format

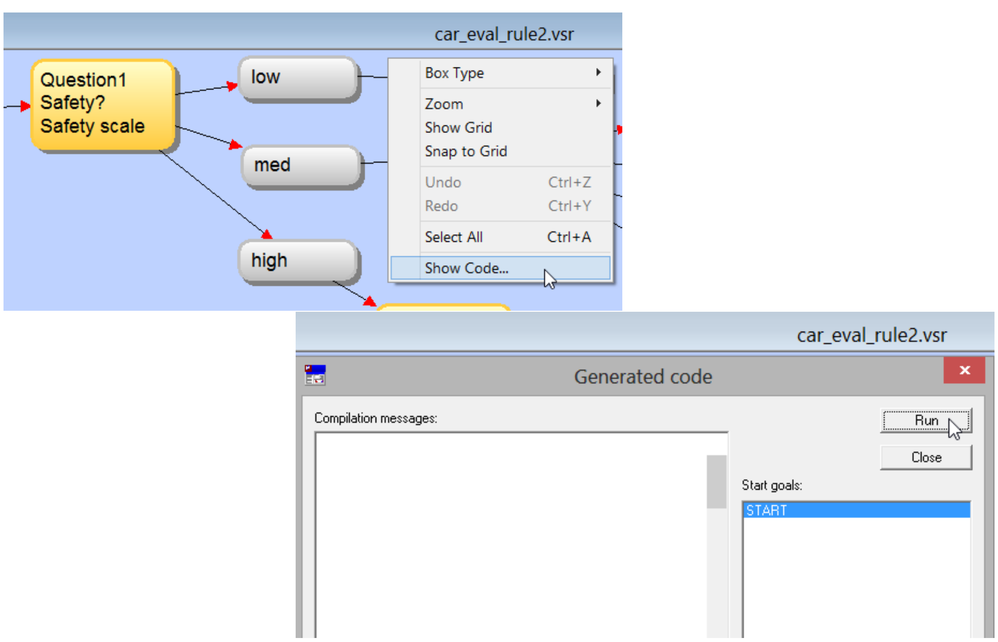


Fig. 7 Steps to generate Prolog code from a given decision tree model

To generate expert system shell for a specific decision tree model, we run the Win-Prolog and use the command:

```
load_files(system(visirule)).
```

Then, click ‘File’ to open the saved ‘VisiRule Files’. In this demonstration, our model (Figure 6) has been saved in a file named ‘car_eval_rule2.vsr’. When the model appears, user can right-click on the screen to choose ‘Show Code…’ and click ‘Run’ button on the ‘Generated code’ window. These steps are graphically shown in Figure 7.

The Prolog code automatically generated from the visual form of a decision tree model is as follows:

```
do ensure_loaded( system(vrlib) ) .

relation 'START'( Conclusion ) if
   q_Question1( Conclusion ) .

relation q_Question1( Conclusion ) if
   the answer to 'Question1' is _ and
   check( 'Question1', =, high ) and
   q_Question3( Conclusion ) .

relation q_Question1( Conclusion ) if
   the answer to 'Question1' is _ and
   check( 'Question1', =, low ) and
   Conclusion = unacc  .

relation q_Question1( Conclusion ) if
   the answer to 'Question1' is _ and
   check( 'Question1', =, med ) and
   q_Question2( Conclusion ) .

relation q_Question3( Conclusion ) if
   the answer to 'Question3' is _ and
   check( 'Question3', =, '<=2' ) and
   q_Question5( Conclusion ) .

relation q_Question3( Conclusion ) if
   the answer to 'Question3' is _ and
   check( 'Question3', =, '>2' ) and
   q_Question4( Conclusion ) .

relation q_Question5( Conclusion ) if
   the answer to 'Question5' is _ and
   check( 'Question5', =, vhigh ) and
   Conclusion = unacc  .

relation q_Question5( Conclusion ) if
   the answer to 'Question5' is _ and
   check( 'Question5', =, high ) and
   Conclusion = unacc  .

relation q_Question5( Conclusion ) if
   the answer to 'Question5' is _ and
   check( 'Question5', =, low ) and
   Conclusion = unacc  .

relation q_Question5( Conclusion ) if
   the answer to 'Question5' is _ and
   check( 'Question5', =, med ) and
   Conclusion = unacc  .

relation q_Question4( Conclusion ) if
   the answer to 'Question4' is _ and
   check( 'Question4', =, high ) and
   Conclusion = acc  .

relation q_Question4( Conclusion ) if
   the answer to 'Question4' is _ and
   check( 'Question4', =, vhigh ) and
   Conclusion = unacc  .

relation q_Question4( Conclusion ) if
   the answer to 'Question4' is _ and
   check( 'Question4', =, med ) and
   Conclusion = acc  .

relation q_Question4( Conclusion ) if
   the answer to 'Question4' is _ and
   check( 'Question4', =, low ) and
   Conclusion = vgood  .

relation q_Question2( Conclusion ) if
   the answer to 'Question2' is _ and
   check( 'Question2', =, vhigh ) and
   Conclusion = unacc  .

relation q_Question2( Conclusion ) if
   the answer to 'Question2' is _ and
   check( 'Question2', =, high ) and
   Conclusion = unacc  .

relation q_Question2( Conclusion ) if
   the answer to 'Question2' is _ and
   check( 'Question2', =, med ) and
   Conclusion = unacc  .

relation q_Question2( Conclusion ) if
   the answer to 'Question2' is _ and
   check( 'Question2', =, low ) and
   Conclusion = acc  .

group group1
   low, high, med .

question 'Question1'
   'Safety?' ;
   choose one of group1
   because 'Safety scale' .

group group2
   vhigh, high, med, low .

question 'Question4'
   'Buying?' ;
   choose one of group2
   because 'Prize' .

group group3
   '<=2', '>2' .

question 'Question3'
   'Person ?' ;
   choose one of group3
   because 'Num.person' .

group group4
   vhigh, high, low, med .

question 'Question5'
   'Buying ?' ;
   choose one of group4
   because 'Prize' .

question 'Question2'
   'Buying ?' ;
```

```
choose one of group2
because 'Prize' .
```

The generated Prolog program acts as an inference engine of the expert system. We test the recommendation given by the system for two cases: unacceptable car and a very good car. The unacceptable case (Figure 8) has been recommended after asking for only one question. That is, for the low safety car, it is unacceptable.

For the second case, we provide the system the following information:

Safety = high
Number of persons that a car can carry > 2
Buying price = low

The recommendation given by the system is that the acceptability level of this car is very good (shown in Figure 9). As the inference engine is encoded in Prolog language that has the inherent ability of backtracking, the system can also search for other solutions if they exist (shown in Figure 10).

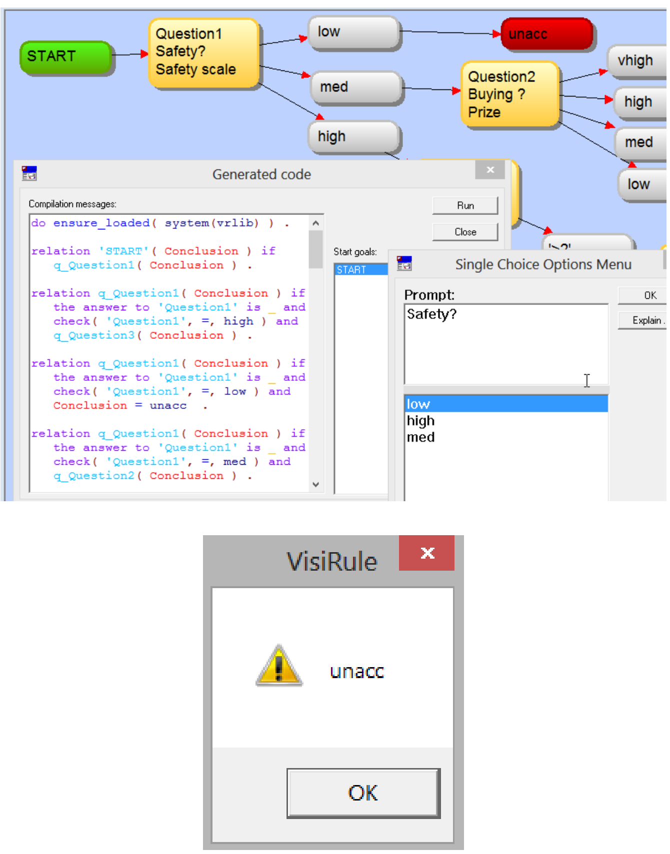


Fig. 8 The case of unacceptable car decision

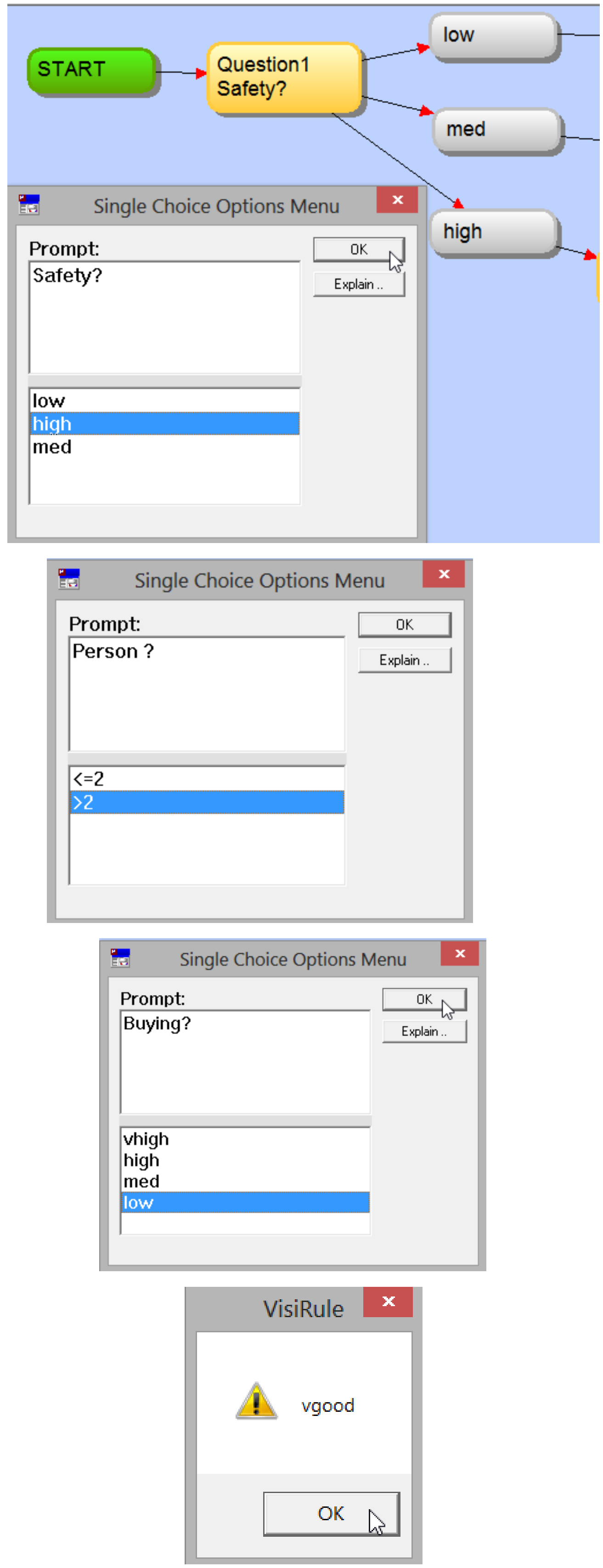


Fig. 9 The case of a very good car decision

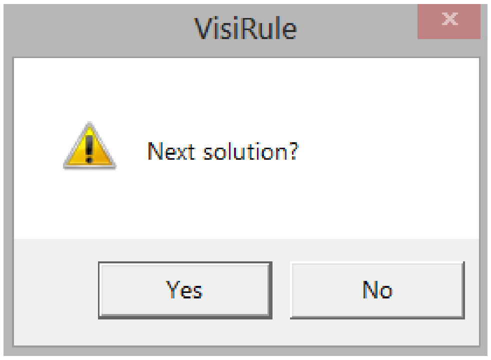


Fig. 10 Screenshot asking user for alternative solutions

From the experiments, we can observe that the generated expert system shell contains both the rule based knowledge and the inference ability. These rules are specific to the knowledge model, which is a decision tree in this demonstration. If the knowledge model has been changed, the generated rules would be changed accordingly.

V. Conclusion

Artificial intelligence, specifically expert systems, has played an important role in solving complex engineering, manufacturing, medicine, and many other problems for more than four decades. Knowledge base and inference procedures have been employed to solve the problems that require significant human expertise and domain-specific knowledge. The required knowledge has to be elicited by knowledge engineers. It is a labor-intensive task, and thus a bottle neck in building intelligent systems.

We propose to apply data mining technique as a major step in a knowledge engine component of the inductive expert system to assist the knowledge elicitation task. The proposed technique is a novel method for automating knowledge acquisition that help supporting manufacturing and other intelligent systems. We demonstrate knowledge mining through the visual tool called KNIME, which has many visualization features to support users who are not an expert in data mining.

Knowledge as a learned tree structure is then transformed by another visual tool called Win-Prolog to generate a Prolog program as a rule set that can be integrated into the knowledge base. The demonstration given in this paper has proved applicability and appropriateness for inferring and reasoning from the knowledge base that can be automatically induced from the stored database.

References

[1] F. Alonso, L. Martinez, A. Perez, and J.P. Valente, “Cooperation between expert knowledge and data mining discovered knowledge: Lesson learned,” *Expert Systems with Applications*, vol.39, 2012, pp.7524-7535.

[2] A. Ammoura, O. Zaiane, and Y. Ji, “Immersed visual data mining: walking the walk,” in *Proceedings of the 18th British National Conference on Databases*, Chilton, U.K., 2001, pp. 202-218.

[3] R. Anderson, *Visual Data Mining: The VisMiner Approach*. Wiley, 2012.

[4] M. Ankerst, C. Elsen, M. Ester, and H. Kriegel, “Visual classification: an interactive approach to decision tree construction,” in *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining*, 1999, pp. 392-396.

[5] K. Bache, and M. Lichman, *UCI Machine Learning Repository*, http://archive.ics.uci.edu/ml, University of California, Irvine, 2013.

[6] A. Badiru, *Expert Systems: Applications in Engineering and Manufacturing*, Prentice-Hall, 1992.

[7] M. Berthold, N. Cebron, F. Dill, T. Gabriel, T. Kotter, T. Meini, P. Ohl, C. Sieb, K. Thiel and B. Wiswedel, “KNIME: The Konstanz Information Miner,” in *Proceedings of the 31st Annual Conference of the Gesellschaft für Klassifikation e.V.*, Albert-Ludwigs-Universität Freiburg, 2007, pp.319-326.

[8] F. Bezerra, and J. Wainer, “Algorithms for anomaly detection of traces in logs of process aware information systems,” *Information Systems*, vol. 38, 2013, pp. 33-44.

[9] M. Bohanec, and V. Rajkovic, “Knowledge acquisition and explanation for multi-attribute decision making,” in *Proceedings of the 8th International Workshop on Expert Systems and Their Applications*, France, 1988, pp. 59-78.

[10] M. de Oliveira, and H. Levkowitz, “From visual data exploration to visual data mining: a survey,” *IEEE Transactions on Visualization and Computer Graphics*, 9, 3, 2003, pp.378-394.

[11] E.A. Feigenbaum, “Expert systems: principles and practice,” in *The Encyclopedia of Computer Science and Engineering*, Stanford: Knowledge Systems Laboratory, 1992.

[12] E. Flior, T. Anaya, C. Moody, M. Beheshti, H. Jianchao, and K. Kowalski, “A knowledge-based system implementation of intrusion detection rules,” in *Proceedings of the 7th International Conference on Information Technology: New Generations*, 2010, pp. 738-742.

[13] J. Fodor, R. Klempous and C. Suárez Araujo: *Recent Advances in Intelligent Engineering Systems*, Springer, 2012.

[14] J.C. Giarratans and G.D. Riley, *Expert systems: Principles and Programming*, fourth edition, Canada: Thomason Learning, 2005.

[15] Y. Gingchi, S. Chao, and H. Yang, "Identifying the technology trend of visual language researches," *WSEAS Transactions on Computers*, vol. 9, issue 11, 2010, pp. 1369-1379.

[16] R. Girratano, *Expert Systems, Principles and Programming*, PWS, 1998.

[17] L.-C. Huang, S.-S. Tseng, and Y.-S. Chu, “Building a wafer fab lot scheduling knowledge-based system,” *WSEAS Transactions on Information Science and Applications*, vol. 3, no. 10, 2006, pp. 1994-2001.

[18] D. Keim, “Information visualization and visual data mining,” *IEEE Transactions on Visualization and Computer Graphics*, 8, 1, 2002, pp.1-8.

[19] C. Leon, F. Biscarri, I. Monedero, J.I. Guerrero, J. Biscarri, and R. Millan, “Integrated expert system applied to the analysis of non-technical losses in power utilities,” *Expert Systems with Applications*, vol. 38, 2011, pp. 10274-10285.

[20] S.-H. Liao, “Expert system methodologies and applications – a decade review from 1995 to 2004,” *Expert Systems with Applications*, vol. 28, 2005, pp. 93-103.

[21] J. Lin, E. Keogh, S. Lonardi, J. Lankford, and D. Nystrom, “VizTree: a tool for visually mining and monitoring massive time series databases,” in *Proceedings of the 30th VLDB Conference*, Toronto, Canada, 2004, pp. 1269-1272.

[22] R. K. Lindsay, B. G. Buchanan, E. A. Feigenbaum, and J. Lederberg, “DENDRAL: A case study of the first expert system for scientific hypothesis formation,” *Artificial Intelligence*, vol.61. no.2, 1993, pp.209-261.

[23] Logic Programming Associates, Win-Prolog, http://www.lpa.co.uk

[24] P. Markellou, M. Rigou, and S. Sirmakessis, “Knowledge mining: A quantitative synthesis of research results and findings,” in S. Sirmakessis (Ed.), *Knowledge Mining*. The Netherlands: Springer-Verlag, 2005.

[25] R. Michalski, “Knowledge mining: A proposed new direction,” *Invited Talk at the Sanken Symposium on Data Mining and Semantic Web*, Osaka University, Japan, 10-11 March 2003. Retrieved from http://www.mli.gmu.edu/papers/2003-2004/03-5.pdf

[26] R.A. Miller, H.E. Pople, and J.D. Myers, “INTERNIST-1, an experimental computer-based diagnostic consultant for general internal medicine,” *New England Journal of Medicine*, vol. 307, no. 8, 1982, pp. 468-476.

[27] A. H. Mohammad and N. A. M. Al Saiyd, “A framework for expert knowledge acquisition,” *International Journal of Computer Science and Network Security*, vol.10, no.11, 2010, pp.145-151.

[28] K. Navarro, E. Lawrence, and D. Martin, "Visual categorization of brain computer interface technologies," *WSEAS Transactions on Information Science and Applications*, vol. 2, issue 8, 2005, pp. 1184-1188.

[29] R. A. Perez, L. O. Hall, S. Romaniuk, and J. T. Lilkendey, “Inductive learning for expert systems in manufacturing,” in *Proceedings of the 25th Hawaii International Conference on System Sciences*, 1992, pp.14-25.

[30] K. Puolamaki, P. Papapetrou, and J. Lijffijt, “Visually controllable data mining methods,” in *Proceedings of the IEEE International Conference on Data Mining Workshops*, Sydney, Australia, 2010, pp. 409-417.

[31] J. Quinlan, “Induction of decision trees,” *Machine Learning*, 1, 1, 1986, pp.81-106.

[32] N. Ruschin-Rimini, O. Maimon and R. Romano, “Visual analysis of quality-related manufacturing data using fractal geometry,” *Journal of Intelligent manufacturing*, 23, 3, 2012, pp.481-495.

[33] S. Simoff, M. Bohlen, and A. Mazeika (Eds.), *Visual Data Mining: Theory, Techniques and Tools for Visual Analytics*, Springer, 2008.

[34] W.B. Schwartz, “Medicine and the computer: The promise and problems of changes,” *New England Journal of Medicine*, vol. 283, 1970, pp. 1257-1264.

[35] E.H. Shortliffe, *Computer-Based Medical Consultations: MYCIN*, Elsevier, 1976.

[36] A. Tsolakidis, C. Sgouropoulou, E. Papageorgiou, O. Terraz, and G. Miaoulis, “Using visual representation for decision support in institutional research evaluation,” in D. Plemenos, and G. Miaoulis (Eds.), *Intelligent Computer Graphics 2012*, Springer, 2012, pp. 41-57.

[37] T. Witkowski, P. Antezak and A. Antezak, “Machine leaning-based classification in manufacturing system,” in *Proceedings of the 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, Czech Republic, 2011, pp.580-585.

[38] B. Zupan, M. Bohanec, I. Bratko, and J. Demsar, “Machine learning by function decomposition,” in *Proceedings of the 14th International Conference on Machine Learning*, Nashville, U.S.A.,1997, pp. 421-429.

Nittaya Kerdprasop is an associate professor and the director of Data Engineering Research Unit, School of Computer Engineering, Suranaree University of Technology, Thailand. She received her B.S. in radiation techniques 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, U.S.A., in 1999. She is a member of IAENG, ACM, and IEEE Computer Society. Her research of interest includes Knowledge Discovery in Databases, Data Mining, Artificial Intelligence, Logic and Constraint Programming, Deductive and Active Databases.

Kittisak Kerdprasop is an associate professor at the School of Computer Engineering and one of the principal researchers of Data Engineering Research Unit, 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, Machine Learning, Artificial Intelligence, Logic and Functional Programming, Probabilistic Databases and Knowledge Bases.