Association for Information Systems
AIS Electronic Library (AISeL)

PACIS 2011 Proceedings

Pacific Asia Conference on Information Systems (PACIS)

9 July 2011

Understading Black Boxes: Knowledge Induction From Models

Jinhwa Kim
Sogang University, jinhwakim@sogang.ac.kr

Jae Kwon Bae
Dongyang University, jkbae@dyu.ac.kr

Dursun Delen
Oklahoma State University, delen@okstate.edu

ISBN: [978-1-86435-644-1]; Full paper

UNDERSTADING BLACK BOXES
: KNOWLEDGE INDUCTION FROM MODELS

Jinhwa Kim, School of Business, Sogang University, S. Seoul, Korea, jinhwakim@sogang.ac.kr

Jaekwon Bae, Department of Railroad Management Information, Young Ju, S. Korea, Dongyang University, jkbae@dyu.ac.kr

Dursun Delen, School of Business, Oklahoma State University, OK, USA, delen@okstate.edu

Abstract

Due to regurations and laws prohibiting uses of private data on customers and their transactions in customer data base, most customer data sets are not easily accessable even in the same organizations. A solutio for this reguatory problems can be providing statistical summary of the data or models induced from the dat, instead of providing raw data sets. The models, however, have limited information on the original raw data set. This study explores possible solutions for these problems. The study uses prediction models from data on credit information of customers provided by a local bank in Seoul, S. Korea. This study suggests approaches in figuring what is inside of the non-rules based models such as regression models or neural network models. The study proposes several rule accumulation algorithms such as (RAA) and a GA-based rule refinement algorithm (GA-RRA) as possible solutions for the problems. The experiments show the performance of the random dataset, RAA, elimination of redundant rules (ERR), and GA-RRA.

Keywords: Personal credit rating; logistic regression model; rule accumulation algorithm (RAA); GA-based rule refinement algorithm (GA-RRA).

1 INTRODUCTION

In these days, it is very difficult to obtain customer credit data from financial institutions because of the laws and regulations on security of data and privacy protection of customer information. The situation is expected to be worse in the future for researchers who work with raw data on customers and their transactions. However, if personal credit rating models, such as logistic regression or neural networks constructed from customer credit data, are provided instead of the data itself, there will be less risk of exposing critical financial information of customers, and also less chance of violation of regulations on privacy.

This study follows two steps. Firstly, the personal credit rating models such as logistic regression from the raw data are provided instead of raw customer credit data in the financial institutions. It is important to identify if a knowledge base, in the form of rule set (master knowledge base) that uses secondary data without the raw data, can produce prediction performance that are close to those of the logistic regression. Secondly, after 100 random datasets are generated from this logistic regression model provided by financial institutions, rules are then extracted from these generated data using a rule induction algorithm such as a C5.0 algorithm, and these rules are sequentially accumulated into the master knowledge base. The study shows how to handle the redundant rules with high consistency and reliability, along with the unique characteristics and hidden patterns in them. Based on these, this study proposes a rule accumulation algorithm (RAA) and a GA-based rule refinement algorithm (GA-RRA) as an efficient approaches in figuring rules inside the non rule-based models.

The RAA is an algorithm that accumulates rules into the knowledge base using an iterative inductive learning mechanism. Based on the domain knowledge in the knowledge base that contains these cumulative rules, the personal credit ratings of customers are predicted. The rules of the RAA are considered to be historical rules that are cumulated over time, the nonvolatile rules that are used for query purposes (without being updated), and subject-oriented rules for decision making.

However, the RAA has some drawbacks. During the processes of rule accumulation, multiple knowledge sources in the knowledge base can lead to knowledge conflicts and knowledge overlapping. Further, factors such as changes in time and space, new techniques, new regulations, new approaches, and new evidence may make collection of the good or useful knowledge difficult. The most important fact in knowledge management is that incorrect knowledge will ultimately lead to wrong decisions. The existing rule application and knowledge application treatments place more emphasis on finding, setting up, saving, spreading, and sharing the direct use of knowledge without considering correctness checks, conflicts, overlaps, variable data sizes, and annotation inconsistencies. This often results in incomplete implementation of knowledge applications, resulting zero value addition, It may also have negative effects. Furthermore, there are not many methods that can verify correctness of knowledge, assess the degree to which related knowledge or knowledge with similar meaning can be merged or integrated, and estimate the extent to which conflicting or overlapping knowledge can be deleted or updated. Therefore, to solve these problems, a new knowledge base is generated by removing the redundant rules from the knowledge base of RAA. The GA-RRA is tested to refine the multiple rules extracted through the RAA from datasets using genetic algorithm. The GA-RRA combines the advantage of both the iterative inductive learning mechanism and genetic algorithm.

2 BACKGROUND AND LIETATURE REVIEW

This study converts non-rule based models into rule based forms. The literature review on rule refinement and that of knowledge base shows that rules from data can be refined for better accuracy. It shows different approaches and techniques in rule refinement and knowledge base refinement. The study shows that mechanism of rule refinement can be applied to generating rules from non rule-based models.

2.1 Rule Refinement and Knowledge Base Refinement Methods

Theory of rule refinement (a.k.a. theory revision or knowledge base refinement) is the task of modifying an initial imperfect knowledge base to make it consistent with empirical data. The goal is to improve the performance of learned models by exploiting prior knowledge, and to acquire knowledge which is more comprehensible. Another motivation in this study is to automate the process of knowledge refinement in the development of expert systems and other knowledge based systems. Most researchers have developed systems for refining knowledge in various forms including: (1) Propositional logical rules systems such as EITHER (Ourston & Mooney 1994), NEITHER (Baffes & Mooney 1993), (2) First order logical rules (logic programs) systems such as FORTE (Richards & Mooney 1995), (3) Certainty-factor rules systems such as RAPTURE (Mahoney & Mooney, 1993), (4) Bayesian networks systems such as BANNER (Ramachandran & Mooney 1998), These systems have demonstrated an ability to revise real knowledge bases and improve learning in several real-world domains (Ourston & Mooney 1994; Baffes & Mooney 1996).
Rule refinement in knowledge based systems is defined as the process of improving the quality of an existing set of rules to obtain an accurate and effective knowledge base. Efforts in rule refinement can be found in SEEK (Politakis & Weiss,1984), SEEK2 (Ginsberg et al. 1988), FOIL (Quinlan 1990), GOLEM (Muggleton & Feng 1990), KBANN (Shavlik & Towell 1989), and KREFS (Park et al. 2001).
While these researches are on the imrpovement of the quality of rules induced from a data set, this study focuses on the production of quality rules from non-rule based models.

2.2 The Iterative Refinement Algorithm (IRA)

Delen et al. (2005) suggested a new scalable classification algorithm (IRA: Iterative Refinement Algorithm) that builds domain knowledge from very large datasets using an iterative inductive learning mechanism. Unlike existing algorithms that build the complete domain knowledge from a dataset all at once, IRA builds the initial domain knowledge from a subset of the available data and then iteratively improves, sharpens and polishes it using the chucks from the remaining data. The IRA method has two main phases:
(1) Build the domain knowledge base with an iterative, weight-based knowledge specialization process.
(2) Revise the knowledge base and the rule weights obtained from phase 1 above (referred to as the original weights) through iterative refinement.
The IRA fits most closely into the data mining method category of merging knowledge bases from very large datasets. Unlike its counterparts in this category, however, it establishes and maintains one “master” domain knowledge base that is initially built with information gleaned from one random sample taken from the large dataset.
IRA can be applied to induction of rules from non-rule based model as it uses incremental approach in improving the quality of the knowledge bases in the form of rules.

3 SUGGESTED ALGORITHMS

3.1 The Rule Accumulation Algorithm (RAA)

The RAA is an algorithm in demonstration with the prediction of personal credit ratings, wherein rules are extracted with rule-based algorithms from customer database, and are accumulated in the knowledge base using an iterative inductive learning mechanism. Based on the domain knowledge in the knowledge base that contains these cumulative rules, the personal credit ratings are predicted. The RAA can effectively integrate multiple rule sets into one centralized knowledge base.

A Scenario on Rule Induction From a Model

We make a request for customer credit data to establish the personal credit rating models for the Credit Analysis & Assessment Division of *Bank A* in Korea, but we receive a reply that the data can not be provided because customer credit data are confidential. *Bank A* adds, however, that instead of the raw data, they can provide the personal credit rating models, including the logistic regression or neural networks, which are constructed using the data. Therefore, the logistic regression for credit rating and the data regarding the variable attributes used in the regression model are obtained from the Credit Analysis & Assessment Division of *Bank A*. The variable attributes refer to descriptive statistics, including variable names, variable definitions, and the data value ranges of variables (minimum and maximum values).

It is useful to identify if optimal decision tree or optimal rule set, that use secondary data without knowing the raw data, can lead to prediction performance that are close to those of the logistic regression for the validation subsets of raw datasets.

Below shows an example of steps in rule refinement processes for above case:

Step1: Generation of random dataset and rule extraction using C5.0 algorithm

The logistic regression, which is constructed using the training set of customer credit data, isprovided by *Bank A*. The numerical expression of the provided logistic regression is shown in eq. (1).

$$Y = -3.426 + (-3.828 * \textit{average balance for past six months}) + (5.362 * \textit{days in arrears}) + (2.603 * \textit{cash dispenser amount}) + (3.595 * \textit{other cash dispenser total amounts}) \quad \text{eq. (1)}$$

First, 100 random datasets (datasets 1 to 100) that had 500 records are made by assigning random numbers to four independent variables. The assigned random numbers had to be within the data range of each variable (between the minimum and maximum values). Then the value of a dependent variable is calculated by substituting eq. (1) of the logistic regression with random independent variables. That is, the value of a dependent variable is obtained by substituting the regression equation from *Bank A* with random independent variables. Thus, the dependent variable values of 100 random datasets are created, from which decision trees and induced rules are derived using the C5.0 algorithm.

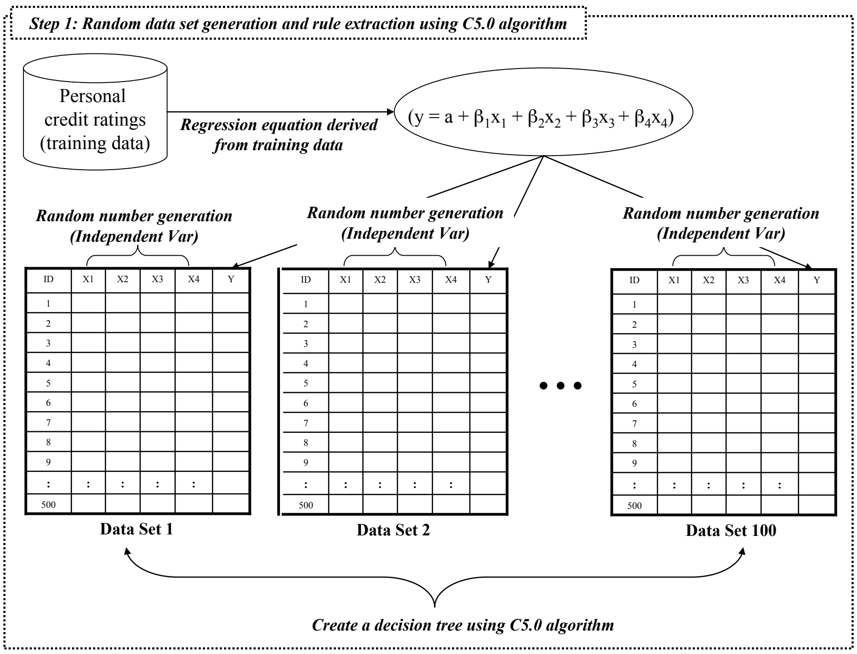


Figure 1. Random dataset generation and rule extraction

Step 2: Development of the knowledge base using RAA

The prediction performances of the 100 decision trees created from 100 datasets are measured using the validation sets of the customer credit data. The rules that are extracted from dataset 1 are first stored in the knowledge base (iteration 1); the rules that are extracted from dataset 2 are also sequentially stored in the knowledge base (iteration 2); and the performances of these accumulated rules are measured using the validation sets of the customer credit data.

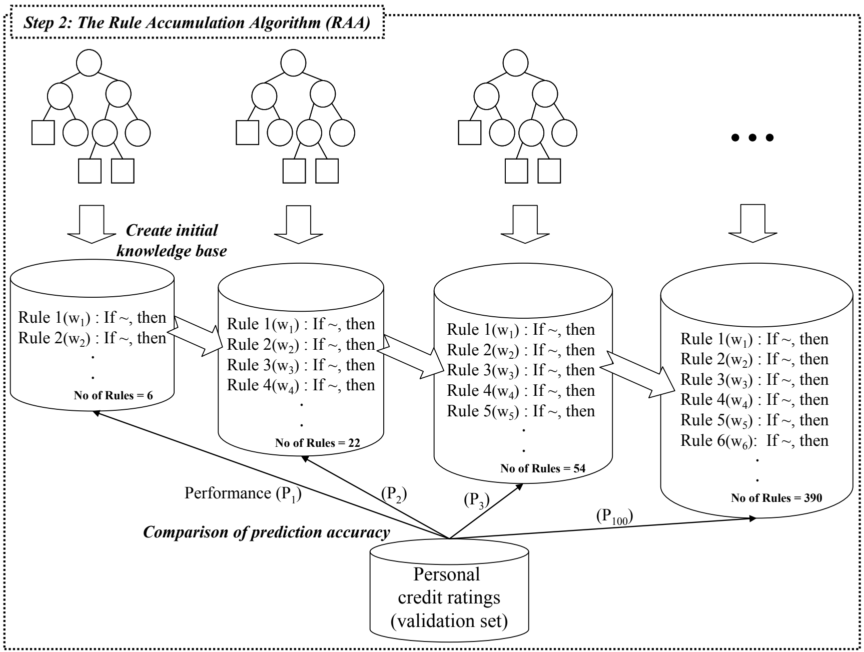


Figure 2. The iterative refinement of rules in RAA

Rules are now weighted, with weights being correlated with the number of times they show up. This process is continued with a third trial (iteration), and a fourth, etc., with the knowledge induced at each trial being merged into the "master" domain knowledge base, with appropriately revised weights. Analogically, the RAA is like stroking with a pencil across a piece of paper laid over a coin to bring out the image engraved on the coin. As repetitive rubbings are conducted, the picture becomes clearer and clearer until at last the entire image shows in total clarity (see Figure 3 for a graphical illustration).

Figure 3. Analogy of discovering the complete picture of a specific domain in the form of a set of rules (engraving the perfect picture via coin scrubbing)

3.2 The GA-based Rule Refinement Algorithm (GA-RRA)

The new technique, called "GA-RRA (GA-based Rule Refinement Algorithm)", aims to combine the advantage of both the rule accumulation (iterative inductive learning mechanism) and GA. Figure 4 shows a schematic diagram of GA-RRA. Details of GA-RRA are described below.

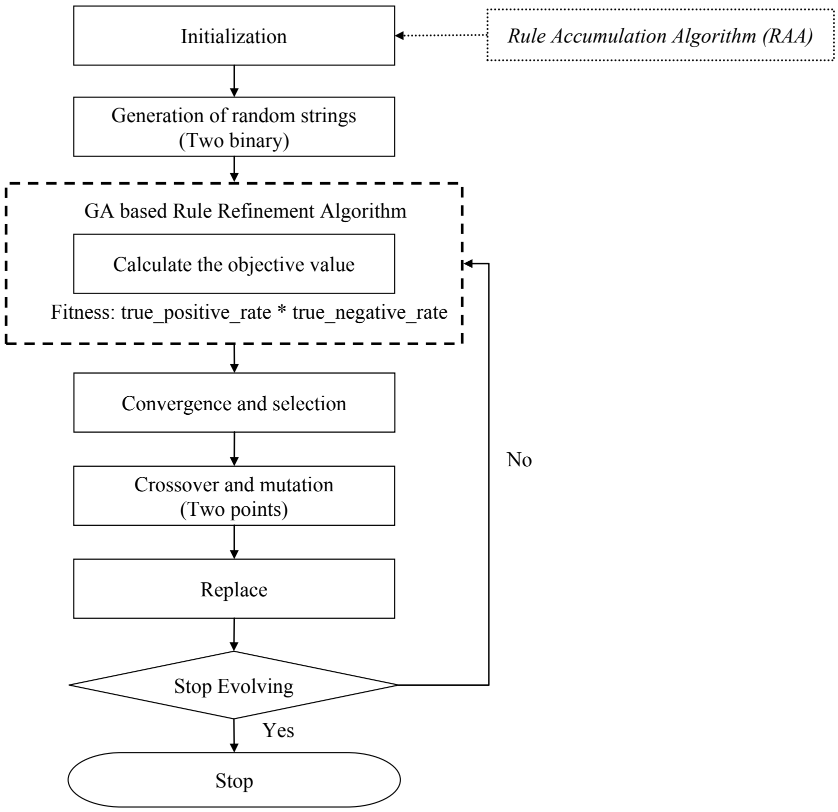


Figure 4. Algorithm of GA-RRA

Step 1: Initialization
All knowledge sources are represented by rules, since almost all knowledge derived by knowledge acquisition tools, or induced by machine learning methods (rule-based algorithms) may easily be translated into or represented by rules.
Step 2: Generation of random strings
We use a pure binary string to do the genetic coding. A classification rule can be coded since one chromosome consists of several segments. Each segment corresponds to either an attribute in the condition part (the IF part) of the rule or to a class in the conclusion part (the THEN part) of the rule.
Step 3: Calculation of the objective value
The fitness function of GA-RRA is used to evaluate how good a rule fits RAA (master knowledge base). Rules need to be evaluated during the training process in order to establish points of reference for GA-RRA. The fitness function considers the datasets as: correctly classified, left to be classified, and the wrongly classified ones. In GA-RRA, the fitness function (see eq. 2), which was suggested by Carvalho & Freitas (2004) is used. The fitness function evaluates the predictive accuracy of a rule based on both *true positive rate* (see eq. 3) and *true negative rate* (see eq. 4) that considerably mitigates some pitfalls associated with the problems of overfitting and lack of balance,

Fitness = true_positive_rate * true_negative_rate eq. (2)

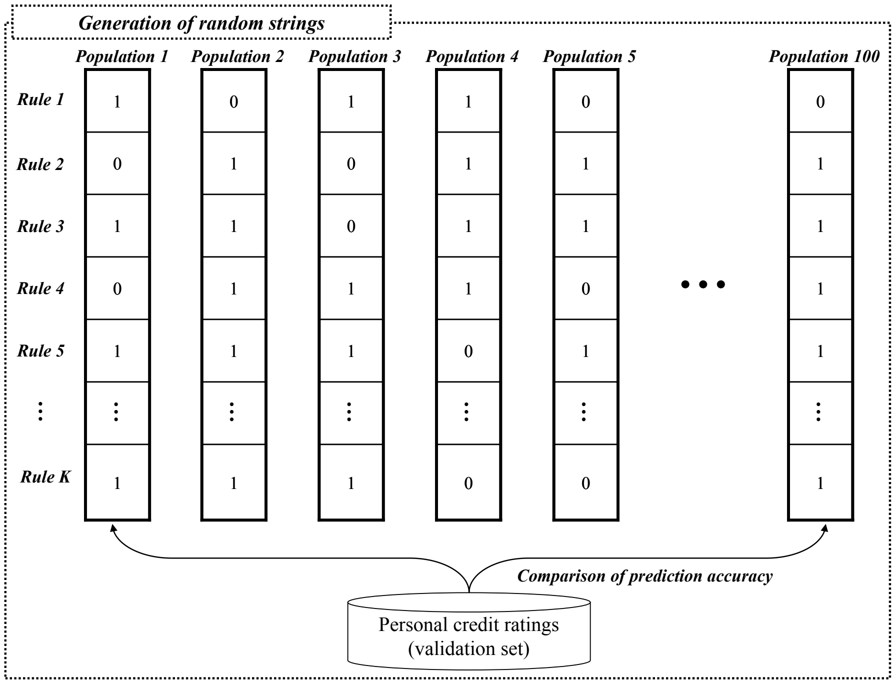
where true_posivive_rate = (no. of TP) / (no. of TP) + (no. of FN) eq. (3)

and true_negative_rate = (no. of TN) / (no. of TN) + (no. of FP) eq. (4)

with

- TP means True Positive which refers to the datasets covered by the rule correctly classified;
- FP means False Positive which refers to the datasets covered by the rule wrongly classified;
- TN means True Negatives which refers to the datasets not covered by the rule but differing from the training target class;

- FN means False Negatives which refers to the datasets not covered by the rule but matching the training target class.



Note: 1(applicable), 0(not applicable)

Figure 5. Generation of random strings

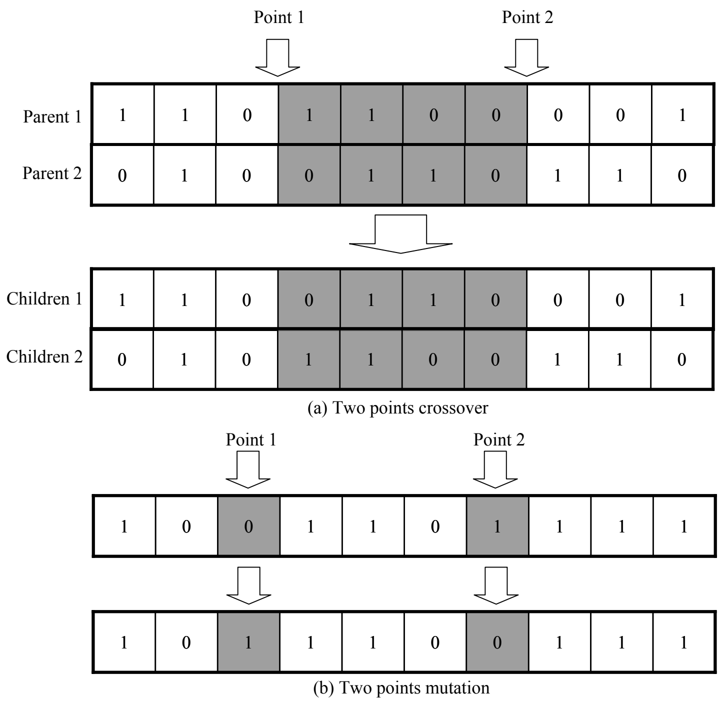


Figure 6. The crossover and mutation operation for GA

Step 4: Convergence and selection

The population is evolved and improved in each generation until a stopping condition is met. In GA, there are quite a few stopping conditions. In this research, the stopping criterion is fulfilled when the number of generations is equal to a pre-defined number of generations or one of the solutions in the population of the GA achieves a full fitness score of 1.

Step 5: Crossover and mutation

The effectiveness of GA depends on complementary crossover and mutation operators. The crossover operator determines the rate of convergence, while the mutation operator makes the GA' search jump out of the local optimum, thus avoiding the premature convergence to a local optimum. Two points crossover method and two points mutation method are applied and it is shown in Figure 6.

Step 6: Replacement

If the fitness function of the current chromosome is better than the best one generated from the previous generation, replace it with the current one.

Step 7: Termination

If the stopping criterion is satisfied, terminate the GA-RRA process and output the best result. The GA-RRA aims to combine the advantage of both iterative inductive learning mechanism and GA.

4 RESEARH DESIGN

For the tests, we randomly choose 8,234 customers of a local bank in Seoul, Republic of Korea and collect their personal credit information for the fiscal year of 2004. The data consist of 4,117 bad customers and 4,117 good customers. The variables are adjusted to follow standard normal distribution, as this helps reduce measurement errors. Out of 28 variables in total, 12 variables are selected using two-sample t-tests as a preliminary screening and then 4 variables are finally selected using stepwise logistic regression. Table 1 summarizes the variables and the definitions of the variables used in this study.

Variables	Definitions
X1	Average balance for past six months
X2	Days in arrears
X3	Cash dispenser amount
X4	Other cash dispenser total amounts

Table 1. Definition of selected variables

Each dataset is splitted into two subsets: a training set and a validation (holdout) set. The training subset is used to train the prediction models. The validation subset is used to test the model's prediction performance for data that have not been used to develop the classification models. Both the training subset (70% of the larger dataset with 5,764 customers) and the validation subset (30% of the larger dataset with 2,470 customers) are randomly selected.

5 EXPERIMENTAL RESULTS

5.1 The Result of RAA

The prediction accuracy is calculated at each iteration when rules from each iteration are sequentially accumulated into master knowledge base. Prediction accuracy shows the best performance (66.03%) in iteration 17 and is saturated without any further increase in iteration 18 and over. As shown in

Figure 7, the *x*-axis and the *y*-axis show the iteration numbers and the prediction accuracy of RAA (%), respectively. From Figure 7, we can observe that iteration 17 can find saturation point of RAA.

5.2 The Result of GA-RRA

The RAA has a problem of redundant rules. Redundant rules are considered to be relatively important rules compared to other rules. Different weights can be assigned to rules following the degrees of redundancy. Secondly, there can be inconsistencies among rules. These inconsistent rules can be deleted in a simple way.

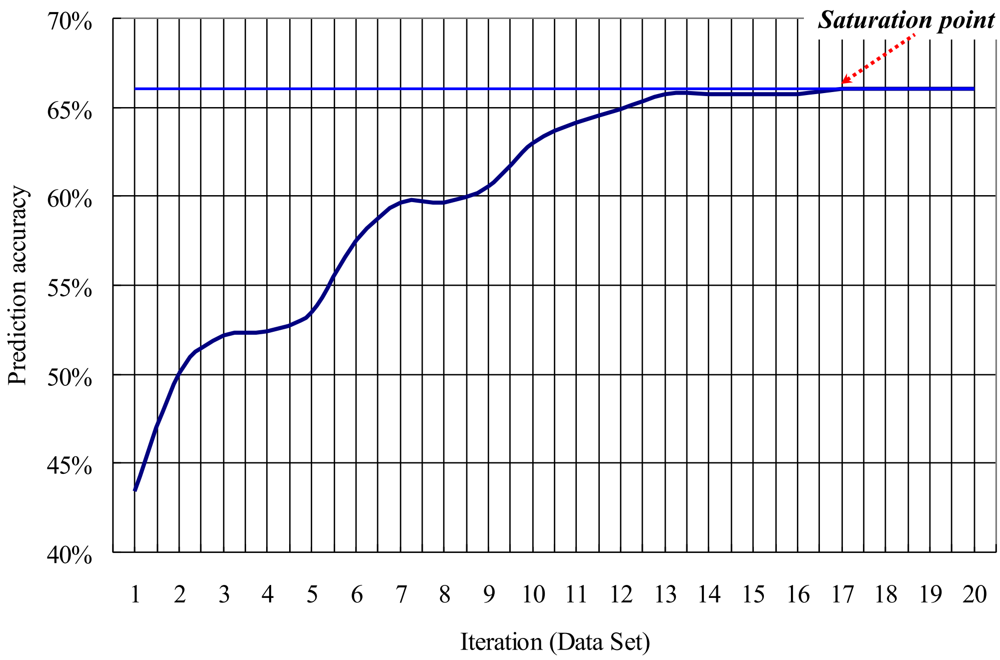


Figure 7. Prediction accuracy growth over number of iterations

There are 24 redundant rules among the 90 rules that are accumulated in the knowledge base of RAA, and the overall redundancy rate is 26.67% (24/90 = 0.2667) as show in Table 2. When the new knowledge base is generated from removing the 24 redundant rules, the prediction accuracy with validation data sets is 66.48%. This is higher than the prediction accuracy of the validation sets (66.03%) in the knowledge base of the RAA that is generated with 90 rules. This means that the performance of prediction accuracy can be better if the redundant rules are removed from the knowledge base of the RAA. It is observed from the above-mentioned results that the rule refinement algorithm for refining the rules in the knowledge base is essential.

	RAA[a)]	ERR[b)]
Number of Rules	90(Iteration 17)	66
Redundancy Rate (%)	26.67	0
Prediction Accuracy (%)	66.03	66.48

Table 2. The result of ERR (validation subsets)
Note: a) RAA: Rule Accumulation Algorithm
b) ERR: Elimination of Redundant Rules.

The convergence happens from the seventeenth generation for both the best and average testing accuracy in figure 8. The accuracy and stability from the other rounds could not be superior to those of the first round, so the outcome in the first round could represent the cross-validation results.
Figure 8 shows that the best and average testing accuracy of the cross validations are almost converged from the seventeenth generation.
The results from the tests show that the performance of GA-RRA is superior to that of the other algorithms such as random dataset, ERR, and RAA.
As for the number of rules that are used for each algorithm, RAA used 90 rules, ERR 66 rules, and GA-RRA 58 rules.

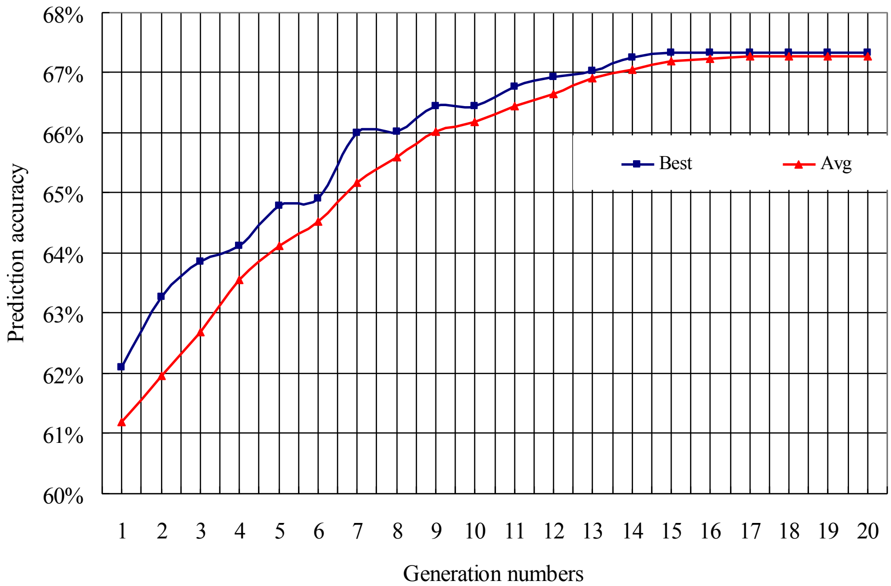


Figure 8. Prediction accuracy growth over number of generations for GA-RRA

The GA-RRA showed the highest prediction performance even if it uses the smallest number of rules among the aforementioned algorithms. This indicates that 58 rules of GA-RRA are optimal decision tree or optimal rule set, which result in prediction performances that are closest to those of the logistic regression with raw data.

	Random Data Set	RAA[a)]	ERR[b)]	GA-RRA[c)]
Number of Rules	5 – 7	90(Iteration 17)	66	58(Generation 17)
Prediction Accuracy (%)	43.84	66.03	66.48	67.33

Table 3. Predictive accuracies of models in validation sets
Note: a) RAA: Rule Accumulation Algorithm
b) ERR: Elimination of Redundant Rules
c) GA-RRA: GA-based Rule Refinement Algorithm

This enhancement in predictability of personal credit ratings can significantly contribute to the correct credit admission evaluation of loan customers, and hence financial institutions can make use of GA-RRA for the better decision makings which can lead to higher profits and firm values eventually.
The theoretical and practical contributions of this study are as follows:

- Rules are extracted from a non rule-based model such as a the regression model, which is a representative model of the black box using the RAA and GA-RRA proposed in this study. A knowledge base can be constructed by combining the rules that were extracted from the regression model, the neural-network rule extraction algorithms (NeuroRule), the inductive learning algorithms (CHAID, CART, QUEST, and C5.0), etc. This combines knowledge base will be very helpful in establishing more elaborate financial forecasting models.

- Although the corresponding raw data are required to evaluate the prediction performances of the standardized financial forecasting models (e.g., Altman Z-Score), the prediction performance can be measured without the raw data, using the optimal rule set that are constructed using GA-RRA.
- Diverse financial data and professional knowledge can be obtained by forming virtual data that are close to the raw data of the rival companies, through the financial forecasting models for them, using RAA and GA-RRA.

6 CONCLUSIONS

Due to the trend in 21 centry, when private information is strictly protected by law and regulations, accessing raw data on customers and POS(point of sales) data is getting difficult. The major contributions of this paper can be providing approaches in configuring what is inside non rule-based models such as regression models and neural network models. If a non rule-based model is re-represented in the form of rules, there can be diverse possible applications. If non-rule based models can be converted into the form of knowledge, normaly in the form of rules, this knowledge can be applied to many applications again.

This study explores approaches in figuring out non-rule based models such as logistic regression or neural networks. The study compares the performance of these approaches: the random dataset, RAA, ERR, and GA-RRA. The tests show that the performance of GA-RRA is superior to that of the other algorithms such as random dataset, ERR, and RAA.

Despite the many findings from this study, it has some limitations. Future studies may explore diverse real world applications of these approaches.

References

Atiya, A. (2001). "Bankruptcy prediction for credit risk using neural networks: A survey and new results," IEEE Transactions on Neural Networks, 12(4), 929-935.

Baffes, P. and Mooney, R. J. (1993). "Extending theory refinement to M-of-N rules," Informatica, 17(4), 387-397.

Baffes, P. and Mooney, R. J. (1996). "Refinement-based student modeling and automated bug library construction," Journal of Artificial Intelligence in Education, 7(1), 75-116.

Carvalho, D. R. and Freitas, A. A. (2004). "A hybrid decision tree/genetic algorithm method for data mining," Information Sciences, 163(1/3), 13-35.

Chen, M. C. and Huang, S. H. (2003). "Credit scoring and rejected instances reassigning through evolutionary computation techniques," Expert Systems with Applications, 24(4), 433-441.

Delen, D., Kletke, M. G. and Kim, J. H. (2005). "A scalable classification algorithm for very large datasets," Journal of Information and Knowledge Management, 4(2), 83-94.

Ginsberg, A., Weiss, S. M. and Politakis, P. (1988). "Automatic knowledge based refinement for classification systems," Artificial Intelligence, 35(2), 197-226.

Goldberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning, 1st ed., Boston, MA:Addison-Wesley.

Goldberg, D. E. and Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms, Foundations of genetic algorithms, Morgan Kaufmann, 69-93.

Mahoney, J. J. and Mooney, R. J. (1993). "Combining connectionist and symbolic learning to refine certainty-factor rule-bases," Connection Science, 5, 339-364.

Muggleton, S. and Feng, C. (1990). "Efficient induction of logic programs," Proceedings of the First Conference on Algorithmic Learning Theory, Ohmsma, Tokyo, Japan, 368-381.

Ourston, D. and Mooney, R. J. (1994). "Theory refinement combining analytical and empirical methods," Artificial Intelligence, 66(2), 273-309.

Park, S., Piramuthu, S. and Shaw, M. (2001). "Dynamic rule refinement in knowledge based data mining systems," Decision Support Systems, 31(2), 205-222.

Politakis, P. and Weiss, S. M. (1984). "Using empirical analysis to refine expert system knowledge bases," Artificial Intelligence, 22(1), 23-48.

Richards, B. L. and Mooney R. J. (1995). "Automated refinement of first-order horn-clause domain theories," Machine Learning, 19(2), 95-131.

Ramachandran, S. and Mooney, R. J. (1998). "Theory refinement of bayesian networks with hidden variables," Proceedings of the Fifteenth International Conference on Machine Learning (ICML-98), Madison, WI, 454-462.

Quinlan, J .R. (1990). "Decision trees and decision-making," IEEE Transaction on Systems, Man and Cybernetics, 20(2), 339-346.

Shavlik, J. W. and Towell, G. G. (1989). "Combining explanation-based and neural learning: An algorithm and empirical results," Connection Science, 1, 233-255.

Tsai, C. F. and Wu, J. W. (2008). "Using neural network ensembles for bankruptcy prediction and credit scoring," Expert Systems with Applications, 34(4), 2639-2649.

Weiss, S. M., and Kulikowski, C. A. (1991). Computer systems that learn: Classification and prediction methods from statistics, neural nets, machine learning, and expert systems, 1st ed., CA: Morgan Kaufmann.

West, D. (2000). "Neural network credit scoring models," Computers and Operations Research, 27(11/12), 1131-1152.

Whitley, D. (1989). "The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best," Proceedings of the third international conference on genetic algorithms, San Mateo, CA: Morgan Kaufman, 116-121.

Zhang, G., Hu, M. Y., Patuwo, B. E. and Indro, D. C. (1999). "Artificial neural networks in bankruptcy prediction: General framework and cross-validation analysis," European Journal of Operational Research, 116(1), 16-32.