PERGAMON

Expert Systems with Applications 25 (2003) 51–62

Expert Systems with Applications

www.elsevier.com/locate/eswa

The hybrid of association rule algorithms and genetic algorithms for tree induction: an example of predicting the student course performance

P.L. Hsu[a], R. Lai[a], C.C. Chiu[b,*], C.I. Hsu[b]

[a]Department of Computer Science and Engineering, Yuan Ze University, Taiwan, ROC
[b]Department of Information Management, Yuan Ze University, Taiwan, ROC

Abstract

Revealing valuable knowledge hidden in corporate data becomes more critical for enterprise decision making. When more data is collected and accumulated, extensive data analysis would not be easier without effective and efficient data mining methods. This paper proposes a hybrid of the association rule algorithm and genetic algorithms (GAs) approach to discover a classification tree. The association rule algorithm is adopted to obtain useful clues based on which the GA is able to proceed its searching tasks in a more efficient way. In addition an association rule algorithm is employed to acquire the insights for those input variables most associated with the outcome variable before executing the evolutionary process. These derived insights are converted into GA's seeding chromosomes. The proposed approach is experimented and compared with a regular genetic algorithm in predicting a student's course performance.

Keywords: Genetic algorithms; Association rule; Classification trees; Student course performance

1. Introduction

Revealing valuable knowledge hidden in corporate data becomes more critical for enterprise decision making. When more data is collected and accumulated, extensive data analysis would not be easier without effective and efficient data mining methods.

Tree induction is one of the most common methods of knowledge discovery. It is a method for discovering a tree-like pattern that can be used for classification or estimation. Some of the often-mentioned tree induction methods such as C4.5 (Quinlan 1992), CART (Breiman, Friedman, Olshen, & Stone 1984), and Quest (Loh & Shih, 1997) are not evolutionary-based approaches. Basically an ideal tree induction technique has to carefully tackle those aspects, such as model comprehensibility and interestingness, attributes selection, learning efficiency and effectiveness, and etc. Genetic algorithms (GAs), one of the often used evolutionary computation technique, has been increasingly aware for its superior flexibility and expressiveness of problem representation as well as its fast searching capability for knowledge discovery. In the past GAs were mostly employed to enhance the learning process of data mining algorithms such as neural nets or fuzzy expert systems, but rather to discover models or patterns. That is, GAs act as a method for performing a guided search for good models in the solution space. While GAs are an interesting approach for discovering hidden valuable knowledge, they have to handle computation efficiency with large volume data.

Generally, tree induction methods are used to automatically produce rule sets for predicting the expected outcomes as accurately as possible. However, the emphasis on revealing novel or interesting knowledge has become a recent research issue in data mining. These attempts may impose additional trees discovery constraints, and thereby produce additional computation overhead. For regular GAs operations, constraint validation is proceeded after a candidate chromosome is produced. That is, several iterations may be required to determine a valid chromosome (i.e. patterns). One way to improve the computation load problem is to obtain associated information such as attributes or the attributes values before initial chromosomes are

* Corresponding author. Fax: +886-3-4352077.
E-mail addresses: imchiu@im.yzu.edu.tw, imchiu@saturn.yzu.edu.tw (C.C. Chiu), hsupl@cyit.edu.tw (P.L. Hsu), krlai@cs.yzu.edu.tw (R. Lai), imchsu@mail.cysu.edu.tw (C.I. Hsu).

0957-4174/03/$ - see front matter © 2003 Elsevier Science Ltd. All rights reserved.
doi:10.1016/S0957-4174(03)00005-8

generated; thereby accelerating the evolution efficiency and effectiveness. Potentially, this can be done by applying association rule algorithms to find out the clues related to the classification results.

This research proposes a novel approach that integrates an association rule algorithm with a GA to discover a classification tree. An association rule algorithm, which is also knows as market basket analysis, is used for attributes selection; therefore those related input variables can be determined before proceeding the GA's evolution.

Apriori algorithm is a popular association rule technique for discovering the attributes relationship that is converted to formulate initial GA population. This proposed method attempts to enhance the GA's searching performance by gaining more important clues leading to the final patterns. A prototype system based on AGA (Association-based GA) approach is developed to predict the learning performance of college students. The application data that consists of student learning profiles and related course information was derived from one university in Taiwan.

The remainder of this paper is organized as follows. In Section 2, previous research works and related techniques are reviewed. The detailed AGA procedures are then introduced in Section 3. Section 4 presents the experiments and results with financial data sets followed by discussion and conclusions.

2. The literature review

2.1. Genetic algorithm for rule induction

Current rule induction systems typically fall into two categories: 'divide and conquer' (Quinlan 1993) and 'separate and conquer' (Clark & Niblett, 1989). The former recursively partitions the instance space until regions of roughly uniform class membership are obtained. The latter induces one rule at a time, separates out the covered instances. Rule induction methods may also be categorized into either tree based or non-tree based methods (Abdullah, 1999). Some of the often-mentioned decision tree induction methods include C4.5 (Quinlan 1993), CART (Breiman et al., 1984) and GOTA (Hartmann, Varshney, Mehrotra, & Gerberich, 1982) algorithms. Both decision trees and rules can be described as disjunctive normal form (DNF) models. Decision trees are generated from data in a top-down, general to specific direction (Chidanand & Sholom, 1997). Each path to a terminal node is represented as a rule consisting of a conjunction of tests on the path's internal nodes. These ordered rules are mutually exclusive (Clark & Boswell, 1991). Quinlan (1993) introduced techniques to transform an induced decision tree into a set of production rules.

Michalski, Mozetic, Hong and Lavrac (1986) proposed AQ15 algorithms to generate a disjunctive set of classification rules. The CN2 rule induction algorithms also use a modified AQ algorithm that involves a top-down beam search procedure (Clark & Niblett, 1989). It adopts entropy as its search heuristic and is only able to generate an ordered list of rules. The Basic Exclusion Algorithm (BEXA) is another type of rule induction method proposed by Theron and Cloete (1996). It follows a general-to-specific search procedure in which disjunctive conjunctions are allowed. Every conjunction is evaluated using the Laplace error estimate. More recently Witten and Frank (1999) described covering algorithms for discovering rule sets in a conjunctive form.

GAs have been successfully applied to data mining for rule discovery in literatures. There are some techniques using one-rule-per-individual encoding proposed by Greene and Smith (1993) and Noda, Freitas, and Lopes (1999). For the one-rule-per-individual encoding approach, a chromosome usually can be identical to a linear string of rule conditions, where each condition is often an attribute-value pair, to represent a rule or a rule set. Although the individual encoding is simpler and syntactically shorter, the problem is that the fitness of a single rule is not necessarily the best indicator of the quality of the discovered rule set. Then, the several-rules-per-individual approach (De Jong, Spears, & Gordon, 1993; Janikow, 1993) has the advantage by considering its rule set as a whole, by taking into account rule interactions. However, this approach makes the chromosome encoding more complicated and syntactically longer, which usually requires more complex genetic operators.

Hu (1998) proposed a Genetic Programming (GP) approach in which a program can be represented by a tree with rule condition and/or attribute values in the leaf nodes and functions in the internal nodes. The challenge is that a tree can grow in size with a shape in a very dynamical way. Thus, an efficient tree-pruning algorithm would be required to prune unsatisfied parts of within a tree to avoid infeasible solutions. Bojarczuk, Lopes, and Freitas (2001) proposed a constrained-syntax GP approach to build a decision model, particularly with emphasis on the discovery of comprehensible knowledge. The constrained-syntax mechanism was applied in verifying the relationship between operators and data types of operands during the tree building process.

In order to discover high-level prediction rules, Freitas (1999) applied a first-order relationship such as 'Salary > Age' by checking an Attribute Compatibility Table (ACT) during the discovery process with GA-Nuggests. ACT was claimed particularly effective for its knowledge representation capability. By extending the use of ACT, our proposed approach allows other complicated attributes relationships such as linear or non-linear quantitative relationship among multi-attributes. This mechanism attempts to aid reducing the search spaces during the GA's evolution process.

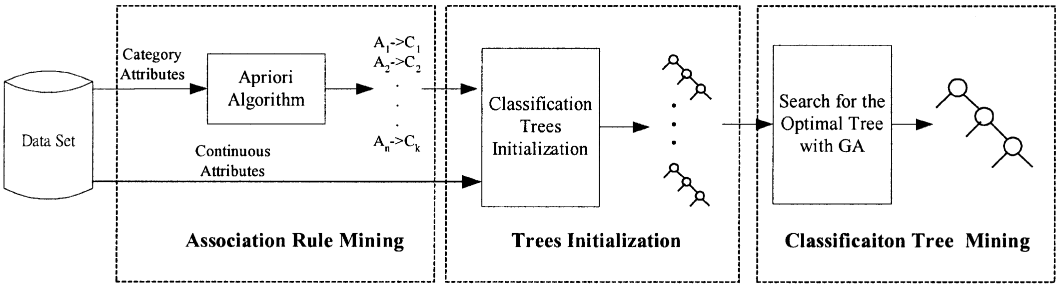

Fig. 1. The conceptual framework of AGA.

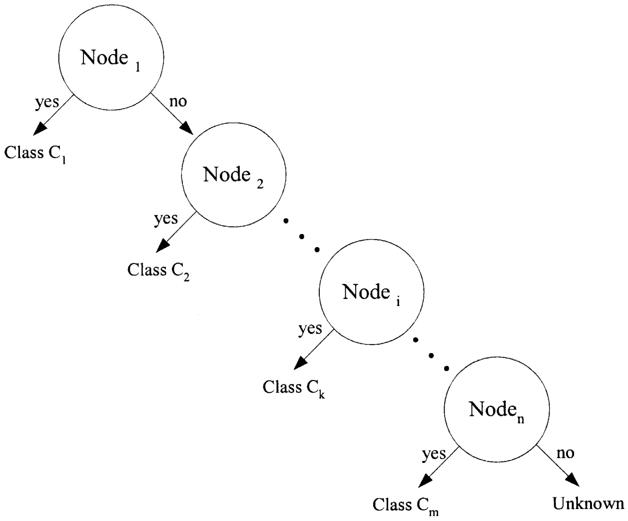

Fig. 2. The illustration for the classification tree.

2.2. Classification trees

Among data mining techniques, a decision tree is one of the most commonly used methods for knowledge discovery. A decision tree is used to discover rules and relationships by systematically breaking down and subdividing the information contained in data (Chou, 1991). A decision tree features its easy understanding and a simple top-down tree structure where decisions are made at each node. The nodes at the bottom of the resulting tree provide the final outcome, either of a discrete or continuous value. When the outcome is of a discrete value, a classification tree is developed (Hunt, 1993), while a regression tree is developed when the outcome is numerical and continuous (Bala & De Jong, 1996).

Classification is a critical type of prediction problems. Classification aims to examine the features of a newly presented object and assign it to one of a predefined set of

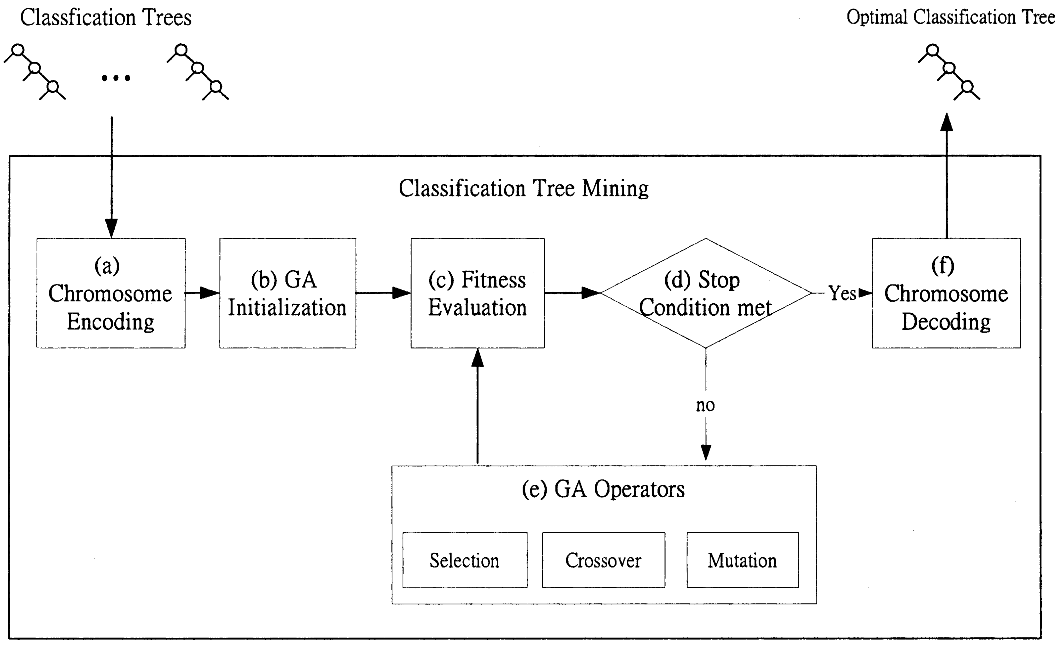

Fig. 3. The conceptual diagram of the proposed classification tree mining.

Table 1
The GA parameter settings

Item	Value
Population size	100
Generations	100/200/300/400/500
Crossover rate	0.6
Mutation rate	0.01
Selection method	Roulette wheel
Training time (House-Votes)	1.5 min[a]
Training time (student learning performance)	0.8 min[a]

[a] The hardware platform is Pentium III 1.0 GHz with 256 MB RAM.

classes (Michael & Gordon, 1997). Classification trees are used to predict membership of cases or objects in the classes of a categorical dependent variable from their measurements on one or more predictor variables. The classification trees induction process often selects the attributes by the algorithm at a given node according to such as Quinlan's information gain (ID3), gain ratio (C4.5) criterion, Quest's statistics-based approach to determine proper attributes and split points for a tree (Loh and Shih, 1997; Quinlan, 1986, 1992).

2.3. Association rules algorithms for attributes selection

Many algorithms can be used to discover association rules from data in order to identify patterns of behavior. One of the most used and famous is the apriori algorithm that can be detailed in (Agrawal, Imielinski, & Swami, 1993; Agrawal & Srikant, 1994). For instance, an association rule algorithm is able to produce a rule as follows:

When people buy Bankers Trust they also buy Dow Chemical 20% of the time

An association rule algorithm, given the minimum support and confidence levels, is able to quickly produce rules from a set of data through the discovery of the so-called itemset. A rule has two measures, called confidence and support. Support (or prevalence) measures how often items occur together, as a percentage of the total records. Confidence (or predictability) measures how much a particular item is dependent on another.

Due to association rule algorithms' advantage in deriving association among data items efficiently, recent data mining research in classification trees construction has attempted to adopt this mechanism for knowledge preprocessing. For example, apriori algorithm was applied to produce those association rules that can be converted into initial population of GP (Niimi & Tazaki, 2000, 2001). Improved learning efficiency for the proposed method was demonstrated when compared with a GP without using association rule algorithms. However, the handling of multivariate classification problems and better learning accuracy were not specified in these researches.

Table 2
The five-fold training/testing results for SGA and AGA (House-Votes Data)

Gen.	100		200		300		400		500	
	Train (%)	Test (%)	Train (%)	Test (%)	Train (%)	Test (%)	Train (%)	Test (%)	Train (%)	Test (%)
SGA	95.98	94.25	97.36	94.71	97.82	94.71	97.87	94.71	97.99	94.71
AGA	97.93	95.63	98.28	95.17	98.51	95.40	98.56	95.40	98.62	95.40

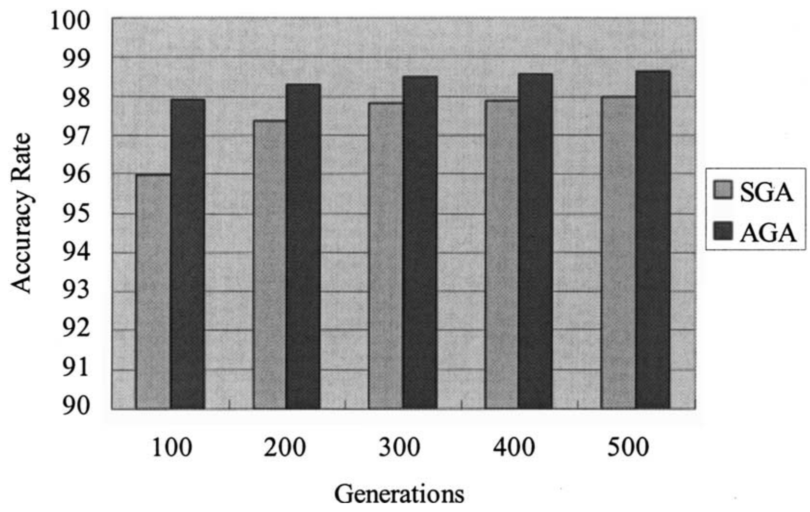

Fig. 4. Training results with various generations.

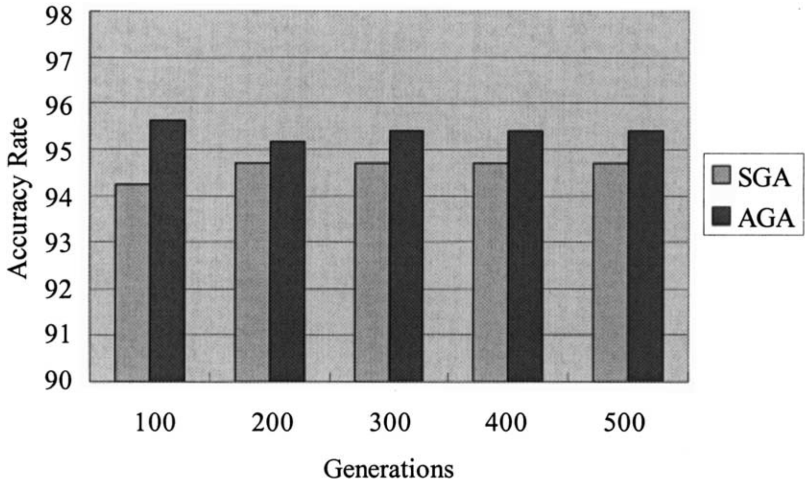

Fig. 5. Testing results with various generations.

3. The hybrid of association rule algorithms and genetic algorithms (AGA)

The proposed AGA approach consists of three modules. According to Fig. 1, these modules are:

- Association rule mining;
- Tree initialization; and
- Classification tree mining.

The association rule mining module generates association rules by apriori algorithm. In this research, the items

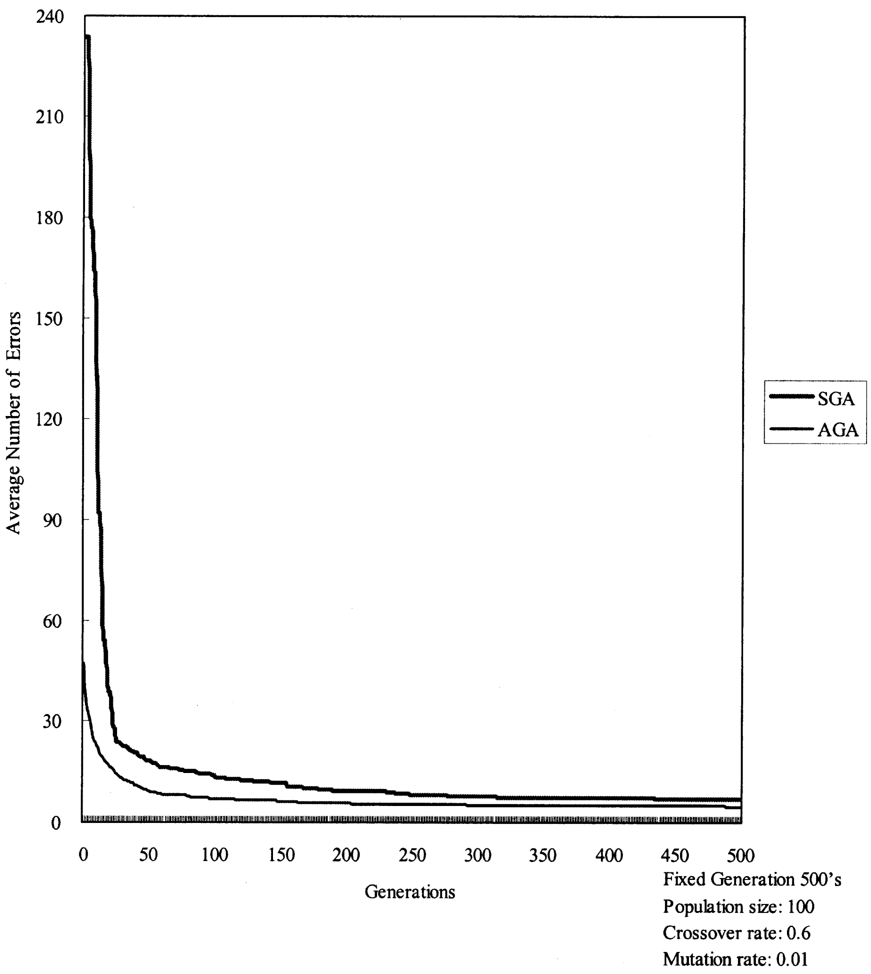

Fig. 6. The learning progress over generations (based on five-fold average).

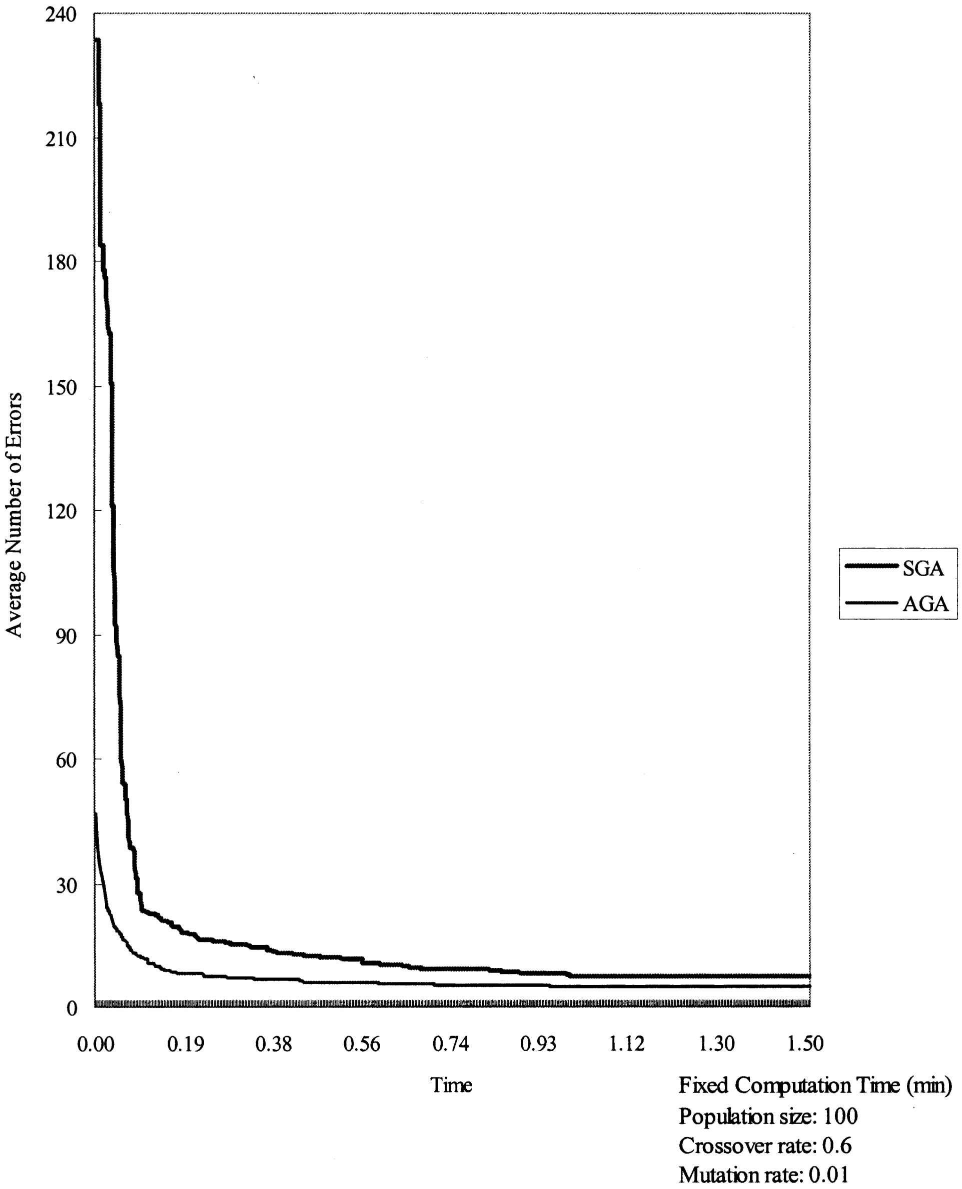

Fig. 7. The learning progress over time (based on five-fold average).

derived from category attributes can be used to construct the association rules. The association rule here is an implication of the form $X \rightarrow Y$ where X is the conjunction of conditions, and Y is the type of classification. The rule $X \rightarrow Y$ has to satisfy user-specified minimum support and minimum confidence levels.

The tree initialization module constructs the potential candidates of classification trees for better predictive accuracy. Fig. 2 shows the classification tree with n nodes. Each node consists of two types of attributes: categorical and continuous. Categorical attributes provide the partial combination of conjunction of conditions and the others are contributed by continuous attributes in the form of inequality. The formal form of a tree node is presented by

A and B $\rightarrow$ C;

where A denotes the antecedent part of the association rule; B is the conjunction of inequality functions in which the continuous attributes, relational operators and splitting values are determined by the GA; and C is the classification result directly obtained from the association rule.

For example, the x_1, x_2, x_3 are categorical attributes and x_4, x_5 are continuous attributes. Assume that one association rule $x_1 = 5$ and $x_3 = 3 \rightarrow C_k$ is selected as a tree node specified as follows.

Node$_i$: IF $x_1 = 5$ and $x_3 = 3$ and $x_4 \leq 10$ and $x_5 > 1$. Then Class C_k;The execution steps of the classification tree initialization module are stated as follows.

Step (a)
For each node, the antecedent condition is obtained from one association rule that is randomly selected out of

Table 3
The notation for each tree node

Node 1	IF X_4 = Yes and X_6 = Others and X_9 = Others and X_{11} = Others and X_{12} = Yes THEN Republican
Node 2	IF X_1 = Others and X_2 = No and X_3 = No and X_4 = No and X_8 = No and X_{11} = Others and X_{14} = No THEN Republican
Node 3	IF X_4 = No THEN Democrat
Node 4	IF X_{11} = No and X_{12} = Yes and X_{14} = Yes and X_{16} = Yes THEN Republican
Node 5	IF X_4 = Yes and X_{10} = Yes and X_{13} = No and X_{14} = Yes THEN Republican
Node 6	IF X_1 = Others and X_5 = Others and X_9 = Yes and X_{13} = Others THEN Democrat
Node 7	IF X_3 = Yes and X_7 = No THEN Democrat
Node 8	IF X_3 = No and X_4 = Others and X_8 = Others and X_9 = No and X_{10} = Yes and X_{11} = No and X_{12} = Others and X_{15} = Others and X_{16} = Others THEN Republican
Node 9	IF a11 = No THEN Republican
Node 10	IF X_4 = Yes and X_5 = Yes and X_6 = Yes and X_7 = No and X_9 = No and X_{10} = Others and X_{11} = No and X_{12} = Yes and X_{13} = No and X_{15} = Others THEN Republican
Node 11	IF X_1 = Others and X_2 = Yes and X_3 = No and X_4 = Others and X_7 = Yes and X_{14} = Yes and X_{15} = Yes and X_{16} = Yes THEN Democrat
Node 12	IF X_7 = Others and X_{11} = Others and X_{13} = Others THEN Republican
Node 13	IF X_4 = Yes and X_9 = No THEN Republican
Node 14	IF X_3 = Others and X_{12} = No and X_{14} = No THEN Democrat
Node 15	IF X_9 = Yes THEN Democrat Else Unknown

the entire association rules set generated in advance. That is, the A part of one tree node formal form is determined.
Step (b)
By applying the GA, the selected relational operator ($<=$ or $>$) and splitting point are determined for each continuous attribute. Therefore the B part of one tree node formal form is determined.
Step (c)
The classification result on a tree node part comes directly from the consequence of the derived association rule. Subsequently each tree node is generated by repeatedly applying Step 1 and 2 for n times; where n is a value automatically determined by the GA.

The classification tree search module applies the GA to search for a superior classification tree based on the potential candidates generated in the tree initialization module. Fig. 3 shows the proposed AGA approach for classification trees mining. The details of each step are illustrated as follows.

Step (a)
Chromosome Encoding—The tree nodes can be easily presented in the form of $\text{Node}_1, \text{Node}_2, \ldots, \text{Node}_n$ where n is the total number of the tree node, and is automatically determined by the GA.
Step (b)
The GA Initialization—to generate a potential chromosome in the beginning. The initial population is obtained from the tree initialization module.
Step (c)
Fitness Evaluation—calculate the fitness value for each chromosome in the current population. The fitness function is defined as the total number of misclassification.
Step (d)
Stop Condition Met—If the specified stopping condition is satisfied, then the entire process is terminated, and the optimal classification tree is confirmed; otherwise, the GA operations are continued.
Step (e)
GA Operators—Each GA operation contains chromosomes selection, crossover, and mutation in order to produce offspring generation based on different GA parameter settings.
Step (f)
Chromosome Decoding—The best chromosome is thus transformed to the optimal classification tree.

4. The experiments and results

Before mining the student learning performance data set, the house-votes data sets from the UCI repository (Blake & Merz, 1998) was used to validate our proposed approach. This data set was derived from the 1984 United States Congressional Voting Records database. For the comparing purpose, a simple GA (SGA) was applied to both the data sets.

Generally association rules extracted by apriori algorithm would be varied depending on the defined support and confidence values. Different association rules extracted may result in different impacts on AGA learning performances. Therefore this research experimented with different sets of minimum support and confidence values to both the credit screening and financial performance prediction problems. The evaluation of those classification trees generated by SGA and AGA approaches was based on a five-fold cross validation. That is, each training stage used 4/5 of the entire data records; with the rest 1/5 data records used for testing

Table 4
The variables used in the model

	Descriptions	Data type
X_1:	Department code (five categories)	Category
X_2:	Gender (male/female)	Category
X_3:	Mid-term rating (three levels)	Category
X_4:	Course credits (1–3 credits)	Category
X_5:	Course type (required/optional)	Category
X_6:	Course difficulty	Continuous
X_7:	Instructor's track record of flunk ratio	Continuous

Table 5
The five-fold training/testing results for SGA and AGA (student learning performance data)

Gen.	100		200		300		400		500	
	Train (%)	Test (%)	Train (%)	Test (%)	Train (%)	Test (%)	Train (%)	Test (%)	Train (%)	Test (%)
SGA	76.34	69.76	76.59	69.76	77.20	69.76	77.32	69.51	77.56	70.00
AGA	76.89	73.17	77.68	73.17	78.29	73.17	78.78	73.41	79.15	73.66

stage. The GA parameter settings for both the applications are summarized in Table 1.

4.1. The application of house-votes data set

The collected 435 House-Votes data records consist of 16 categorical attributes expressed by ‘Y’, ‘N’, and ‘?’ for the input part. The output part is a categorical attribute with two classes (Democrat, Republican). Data of the 16 categorical attributes were fed into apriori algorithm to produce the association rules. Among the entire data records, 267 are democrats, 168 are republicans. Instead of precluding these records from training, this research denotes these ‘?’ values by ‘Others’. After several trials with different sets of minimum support values and confidence values, the best AGA learning performance is obtained. Both the training and testing results are summarized in Table 2, along with their corresponding representation shown in Figs. 4 and 5. These results are based on the minimum support value (= 20) and confidence value (= 100). The derived

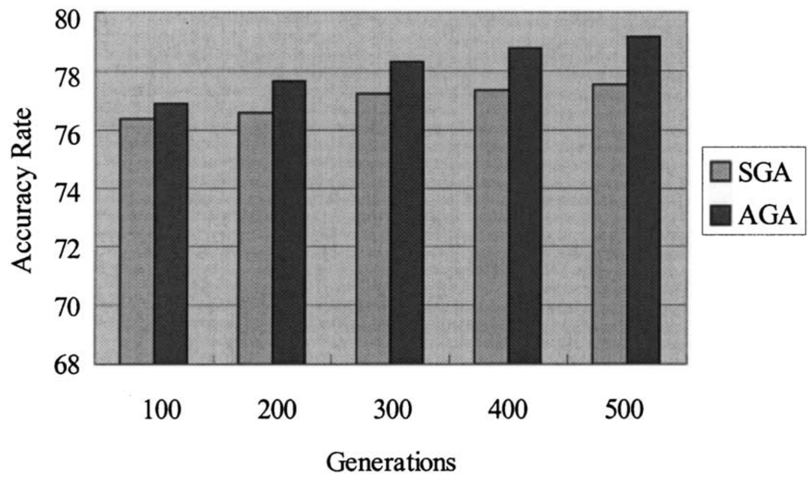

Fig. 8. Training results with various generations.

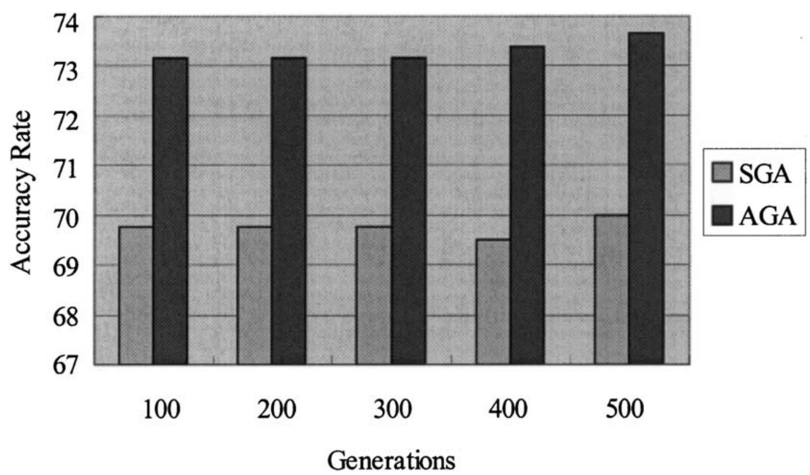

Fig. 9. Testing results with various generations.

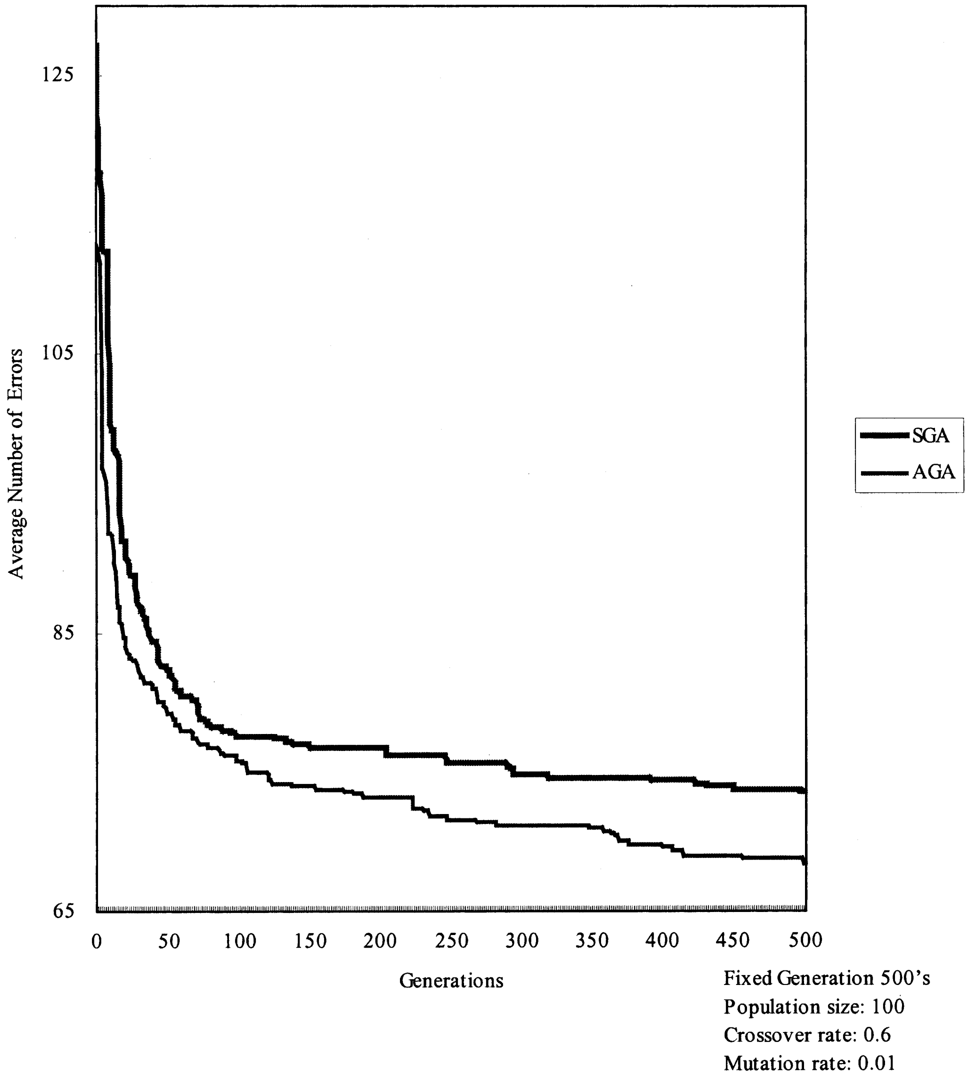

Fig. 10. The learning progress over generations (based on five-fold average).

association rule sets consists of 1743 rules for 'Democrat' output category and 2215 association rules for 'Republican' output category. In order to obtain more details about the learning progress for the two approaches, learning tract behavior were recorded in sessions. Figs. 6 and 7 depict the entire learning progresses monitored over generations and time. Table 3 presents the detail tree nodes notation for one of the relatively better classification trees derived. Based on this produced classification tree, the accuracy rates for the training and testing stages are able to reach 98.28 and 100.00%, respectively.

4.2. The application of student learning performance data set

4.2.1. The data description

In order to better monitor a student's learning performance, one university in Taiwan designed a pre-warning system that requires each course instructor to input a student's up-to-present learning performance grade one week after mid-term exam. The student's mid-term rating is categorized into three levels—'A', 'B', and 'C' among which 'A' implies EXCELLENT; 'B' for O.K.; and 'C' for POOR. Generally the more 'C's' a students receives, the higher failure probability a student will have for the course taken. However, purely replying on this mid-term grading information is not sufficient to determine whether the student will survive for a course in the end of the semester. Therefore, other supporting information such as the course difficulty, the grading tract records of the instructor, and the student profile information were collected. Each data record contains five categorical attributes and two continuous attributes for the input part. The output part is one categorical attribute that is the class of 'pass' or 'fail'. The notation

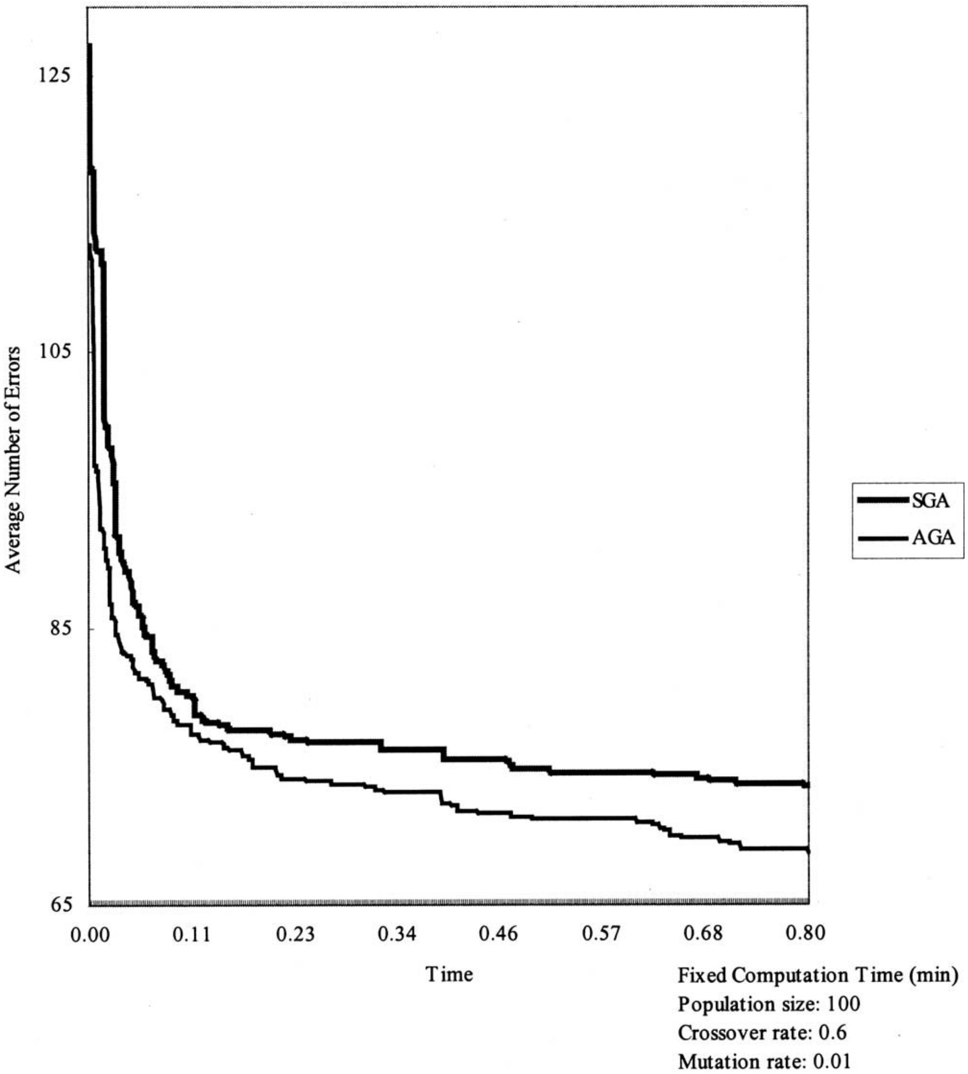

Fig. 11. The learning progress over time (based on five-fold average).

for the variables used in this prediction model is specified in Table 4. 410 student records containing both the passed (total of 146) and failed (total of 264) records from 48 freshman and sophomore dropped out students in Engineering School were collected for analysis.

After several trials with different sets of minimum support values and confidence values, the best AGA learning performance is obtained. Both the training and testing performance are summarized in Table 5, along with their corresponding representation shown in Figs. 8 and 9. These results are based on the minimum support value ($=5$) and confidence value ($=100$). The derived association rule sets consists of 100 rules for Pass output category and 58 association rules for Fail output category. In order to obtain more details about the learning progress for the two approaches, learning tract behavior were recorded in sessions. Figs. 10 and 11 show the entire learning progresses monitored over generations and time. Table 6 presents the detail tree nodes notation for one of the superior classification trees derived. Based on this produced classification tree, the accuracy rates for the training and testing stages are able to reach 79.27 and 80.49%, respectively.

Table 6
The notation for each tree node

Node 1	IF X_3 = 'A' and $X_6 > 1$ THEN Pass
Node 2	IF X_3 = 'C' THEN Fail
Node 3	IF X_1 = CS and Sex = Male and $X_4 = 2$ and $X_6 > 1$ THEN Pass
Node 4	IF X_1 = CE and $X_4 = 3$ and X_5 = Required THEN Pass
Node 5	IF X_1 = CE and $X_4 = 3$ and X_5 = Required THEN Pass
Node 6	IF X_1 = IE and Sex = Male and $X_4 = 2$ and $X_7 \leq 0.421918$ THEN Fail
Node 7	IF $X_7 \leq 0.0684225$ THEN Pass
Node 8	IF X_1 = EE and X_3 = B and $X_4 = 1$ and X_5 = Optional THEN Fail
Node 9	IF $X_7 \leq 0.330424 \times X_6$ THEN Fail

5. Discussion

According to the results indicated above AGA achieves superior learning performance than SGA in terms of computation efficiency and accuracy. By applying association rule process, the partial knowledge is extracted and transformed as seeding chromosomes. According to Fig. 6, the initial average number of errors for both SGA and AGA are 230 and 50, respectively, with House-Votes data. In Fig. 10, the initial average number of errors for both SGA and AGA are 127 and 113, respectively, with student learning performance data. This improvement of initial learning performance can be resulted from the derived association rules that are then transformed into GA's seeding chromosomes.

According to Fig. 6, for the training stage SGA takes 500 generations to reach the similar performance that takes AGA only 40 generations to reach. The outcomes can be attributed to the adoption of apriori algorithm by which the GA search space is substantially reduced. Also it can be seen that AGA consistently outperforms SGA over generations. Further, for the computation time, AGA takes.19 min to reach the learning performance that takes SGA at least 1.5 min to reach.

As shown in Fig. 10, for the training stage SGA takes 500 generations to reach the similar performance that takes AGA 100 generations to reach. Also it can be seen that AGA consistently outperforms SGA over generations. Further for the computation time, AGA takes 17 min to reach the learning performance that takes SGA 0.80 min to reach.

6. Conclusions and future development

We have introduced the AGA approach that hybridizes apriori algorithm and the genetic algorithm for classification tree induction. Incorporating the associated knowledge related to the classification results is crucial for improving evolutionary-based mining tasks. By employing the association rule algorithm to acquire partial knowledge from data, our proposed approach is able to more effectively and efficiently induce a classification tree by converting the derived association rules into the GA's seeding chromosomes.

Comparing with SGA, AGA achieves higher predictive accuracy and less computation time required for the classification tree induction by experimenting a UCI benchmark data set as well as the student learning performance data set.

Predicting a student's course performance from the data derived from the student/course profiles as well as the mid-term rating information is a novel way to aid both the students and the university to grasp further information about the approximate student course performance before too late to recover. According to the experiment results, AGA has been proved to be a feasible way to provide a decently acceptable solution for predicting a student's course performance with near 80% classification accuracy.

In addition, the classification trees discovered by AGA not only obtain higher predictive accuracy and computation efficiency, but also may produce more user transparent or significant knowledge.

The proposed AGA is generic and problem independent. Besides to integrating with association rule algorithms for knowledge preprocessing, AGA is flexible to incorporate the user information or domain knowledge via the expressive power of first order logic into a tree induction process. The proposed approach is not only applicable for binary classification problems, but also applicable for multi-category classification problems.

Currently AGA approach is able to reveal tree splitting nodes that may allow complex rule sets-like discriminating formats such as $\text{Attribute}_i \leq w \times \text{Attribute}_j$ relationship which can be extended to express more complicated multivariate inequations with either a linear or nonlinear form in the future.

Acknowledgements

This research was supported by the National Science Council of Taiwan, ROC, under the contract number NSC91-2745-P-155-03.

References

Abdullah, M. K. (1999). CAN: chain of nodes approach to direct rule induction. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *29*(6), 758–770.

Agrawal, R., & Srikant, R. (1994). *Fast Algorithms for Mining Association Rules. Proceedings 20th International Conference on Very Large Databases, Santiago, Chile*, pp. 478–499.

Agrawal, R., Imielinski, T., & Swami, A. (1993). *Mining association between sets of items in massive databases. Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 207–216.

Bala, J. (1996). Using learning to facilitate the evolution of features for recognizing visual concepts. *Evolutionary Computation*, *4*(3), 297–312. Fall.

Blake, C. L., & Merz, C. J (1998). UCI Repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html, Irvine, University of California, Department of Information and Computer Science

Bojarczuk, C. E., Lopes, H. S., & Freitas, A. A. (2001). *Data mining with constrained-syntax genetic programming: applications in medical data sets. Proc. Intelligent Data Analysis in Medicine and Pharmacology (IDAMAP-2001), a Workshop at Medinfo-2001, London, UK.*

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Los Angeles, CA: Wadsworth.

Chidanand, A., & Sholom, W. (1997). Data mining with decision trees and decision rules. *Future Generation Computer Systems*, *13*, 197–210.

Chou, P. A. (1991). Optimal partitioning for classification and regression trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13*, 340–354.

Clark, P., & Boswell, R. (1991). Rule induction with CN2: some recent improvements. In Y. Kodratoff (Ed.), *Machine learning-European working session on learning EWSL-91* (pp. 151–163). Berlin: Springer.

Clark, P., & Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning Journal*, *3*, 261–283.

De Jong, K. A., Spears, W. M., & Gordon, D. F. (1993). Using genetic algorithms for concept learning. *Machine Learning*, *13*, 161–188.

Freitas, A. A. (1999). *A genetic algorithm for generalized rule induction. Advanced in Soft Computing—Engineering Design and Manufacturing*, Berlin: Springer, pp. 340–353.

Greene, D. P., & Smith, S. F. (1993). Competition-based induction of decision models from examples. *Machine Learning*, *13*, 229–257.

Hartmann, C. R. P., Varshney, P. K., Mehrotra, K. G., & Gerberich, C. L. (1982). Application of information theory to the construction of efficient decision trees. *IEEE Transactions Information Theory*, *IT-28*, 565–577.

Hu, Y. J. (1998). *A Genetic Programming Approach to Constructive Induction, Genetic Programming. Proceedings of the Third Annual Conference, San Mateo, California*, pp.146–151.

Hunt, K. J. (1993). Classification by induction: application to modeling and control of non-linear dynamical systems. *Intelligent Systems Engineering*, *24*, 231–245.

Janikow, C. Z. (1993). A knowledge-intensive genetic algorithm for supervised learning. *Machine Learning*, *13*, 189–228.

Loh, W. Y., & Shih, Y. S. (1997). Split selection methods for classification trees. *Statistica Sinica*, *7*, 815–840.

Michalski, R. S., Mozetic, I., Hong, J., & Lavrac, N. (1986). *The Multi-purpose incremental learning system AQ15 and its testing application to three medical domains. Proceedings of the Fifth National Conference on Artificial Intelligence*, Philadelphia, PA: Morgan Kaufmann, pp.1041–1045.

Michael, J. A., & Gordon, L. (1997). *Data mining techniques: for marketing, sales, and customer support*. New York: Wiley Computer Pub.

Niimi, A., & Tazaki, E. (2000). *Genetic programming with association rule algorithm for decision tree construction. Fourth International Conference on Knowledge-Based Intelligence Engineering Systems and Allied Technologies, Brighton, UK*, pp. 746–749.

Niimi, A., & Tazaki, E. (2001). Combined method of genetic programming and association rule algorithm. *Applied Artificial Intelligence*, *15*, 825–842.

Noda, E., Freitas, A. A., & Lopes, H. S. (1999). *Discovering interesting prediction rules with a genetic Algorithm. Proceedings of Congress on Evolutionary Computation, Washington, DC*, pp. 1322–1329.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*, 81–106.

Quinlan, J. R. (1992). *C4.5: Programs for machine learning. Morgan Kaufman Series, in Machine Learning*, Dordrecht: Kluwer Academic Publishers.

Theron, H., & Cloete, I. (1996). BEXA: a covering algorithm for learning prepositional concept descriptions. *Machine Learning*, *24*, 5–40.

Witten, I. H., & Frank, E. (1999). *Data mining*. San Francisco, CA: Morgan Kaufmann Publishers, pp. 97–104.