computing@tanet.edu.te.ua
www.tanet.edu.te.ua/computing

ISSN 1727-6209
International Scientific
Journal of Computing

A HYBRID ALGORITHM FOR DECISION TREE GENERATION

Yuri Kornienko [1], Arkady Borisov [2]

Institute of Information Technology, Riga Technical University,
1 Kalku Str., Riga LV-1658, Latvia
1) j.kornienko@pf.lv
2) aborisov@egle.sc.rtu.lv

Abstract: *The paper discusses the experiments performed with Machine Learning algorithms (ID3, C4.5, Bagged-C4.5, Boosted-C4.5 and Naive Bayes) and an algorithm made on the basis of a combination of genetic algorithms (GA) and ID3. The latter algorithm is implemented as an extension of the MLC++ Library of Stanford University. The behaviour of the algorithm is tested using 24 databases including those with a large number of attributes. It is shown that owing to "hill-climbing" problem solving, the characteristics of the classifier made with the help of the new algorithm became significantly better. The behaviour of the algorithm is examined when constructing pruned classifiers. The ways to improve standard Machine Learning algorithms are suggested.*

Keywords: *genetic algorithm, ID3, tree generation, "hill-climbing", ensembles of classifiers*

1. INTRODUCTION

Many decision making problems represent a category of classification tasks. The developers of these systems are usually concentrated on such parameters as computation cost and learning method accuracy. Normally, a single classifier is constructed as a result of standard use of Machine Learning (ML) algorithms, however, methods of classifier ensembles construction became very popular lately. These ensembles demonstrate better recognition quality for the objects that were not represented in the test set. Nowadays, several methods exist that might be employed to construct classifier ensembles [1], including stacking, windowing, bagging and boosting.

This study proposes an analysis of the algorithm for the combined use of genetic algorithms (GA) and heuristic search strategies [6] that enables using the advantages of GA along with such possibilities of heuristic search as decision tree construction and finding a set of rules that are able to explain the hidden regularities in the domain under consideration. This method is another technique for obtaining not a single classifier but the whole set. Moreover, the implementation of the new algorithm makes it possible to apply these three methods of new instances classification:

- Voting, when each instance out of the test set is passed through the stage of classifier ensemble classification. The class wins which gets more votes.
- The best classifier of the ensemble is used for classification.
- Based on the classifier ensemble, a decision tree is constructed which is common for all classifiers, that is a combined classifier is made.

The task of this study is to examine the behaviour of the algorithm developed on the basis of combination of decision trees and GA:

1. To compare the algorithm based on the combination of decision trees and GA with other algorithms– ID3 [2], C4.5 [3], Bagged-C4.5, Boosted-C4.5 [1], and Naive Bayes.
2. To examine the behaviour of the new algorithm in the course of working with the data collected using real-world knowledge areas. To prove that as a result of "hill-climbing" problem solving, the characteristics of such classifier were improved.
3. To study the behaviour of the algorithm when constructing pruned classifiers. The pruned classifiers have both advantages and disadvantages. On the one hand, in these classifiers a small number of elements (rules or decision trees) are employed , which gives good possibilities of understanding reasons and hidden regularities in the given domain. On the other hand, the threshold of classification might suffer. Thus, in the course of experiments it is planned to

study the accuracy of classification depending on pruning extent of the classifier taught.

2. DESCRIPTION OF THE ALGORITHM

As a result of the GA-ID3 hybrid working, a decision tree is constructed that can be used to generate rules. The nodes of the tree are its attributes but its branches are the corresponding values of those attributes. The role of GA in the hybrid is to supply ID3 a number of possible attributes on the basis of which splitting of a subset of instances will then be made. At the beginning of the algorithm's execution it is necessary to create an initial population each string of which would represent a set of attributes chosen at random. The size of the population cannot be larger than the number of attributes as each attribute may be employed only once on each branch. To evaluate the utility of each string, splitting measures developed for the ID3 algorithm can be used. For each string, an average measure of attribute efficiency will be determined. Thus it becomes possible to employ all the GA operators.

Table 1 shows an abstracted description of the algorithm execution. As a whole, the execution of the combined GA and ID3 algorithm is an iterative procedure (GA-ID3 procedure). Each iteration results in a decision tree. After *n* iterations, a series of trees will be obtained the best of which could be used to generate rules. The measure of error of the best tree has to meet the User-defined-criteria.

Table 1. Algorithm of the GA-ID3 combined method operation

Let E be a set of instances to be classified.
Let A be a set of attributes for the description of each instance.
Let P be an initial population, each string of which is a set of attributes chosen at random. Let us consider an attribute in the string to be a chromosome.
Let TE(E) be a stopping criterion of instance splitting procedure.
Let IDM(AI,E) be an evaluation function for attributes where Ai$\in$A.

PROCEDURE GA-ID3 (E)
// Based on the population, a decision tree is generated by this procedure :
    Tree = ID3 (P, E);
IF Error-Complexity (Tree) > User_defined_criteria THEN
// It means that the tree is not optimal and the algorithm has to generate a new population //which may be better than the present one…
GA(P);
Go to the beginning of the algorithm.
Return from the algorithm, as the Tree is optimal.

PROCEDURE ID3 (P, E)
IF E satisfies the termination criterion TE (E), (say, all instances describe the same class)
THEN return the leaf of the tree, which describes the most general class from set E
ELSE find attribute Abest $\in$pi (where pi$\in$A, pi – a string of the current population P, i – is the number of the string) having the largest value of function IDM(Abest, E), such that it is never met when climbing from the given node to the root of the tree and is contained in the given string.
For each value of attribute Abest Vj , generate a subtree using ID3(Ej) where Ej are such instances from E for which Abest =Vj.
Return a tree node marked as testing one for attribute Abest, with the attached subtree for each value Abest.

PROCEDURE GA (P)
For each string the number of non-similar chromosomes is calculated. Respectively, the maximal value of fitness is equal to the number of chromosomes in the string but the minimal value is equal to 1. The strings with larger fitness value have a greater probability of having successors.
Apply one of GA methods to select pairs for mating (say, Monte-Carlo method).
Perform random choice of pairs for mating and of the size of exchange material.
As a result of the previous operation, we arrive at the offspring appearance and generation of a new population.
    Mutation.
Return.

PROCEDURE Error-Complexity (Tree)
    Calculate the error measure for the Tree.
Return error measure.

The implementation of the algorithm is based on the MLC library. The MLC library is an object-oriented library of the classes maximally adjusted for writing, testing and comparison of new ML algorithms. The library contains certain general self-learning algorithms which can be employed by any user, procedures for operations of data reading and result output, and testing the accuracy of algorithms forecast. The library is supplied with an open code, which enables adding of new algorithms. Due to the aforementioned features of the library, it was possible to compare and analyse the behaviour of the new algorithms and standard ML algorithms.

3. DESCRIPTION OF EXPERIMENTS

To perform the experiments, the ID3_GA class contains special parameters that were employed to ensure a finer tuning of the algorithm:

- Population size – size of attribute population. This parameter allows one to regulate the depth of the decision tree.
- Size of population string – is responsible for the number of attributes that are available for node formation procedure at each stage of decision tree construction.
- Number of GA iterations. This parameter enables regulation of the number of classifiers created in an ensemble. The experiments were performed using 20, 50 and 100 classifiers.
- Mutation probability . The probability of mutation determines a chance of changing an attribute in the string to an other attribute randomly selected. Testing was made with 10%, 5% and 0% probabilities.

The algorithms were evaluated using the representative collection of databases from UCI Machine Learning Repository. 24 databases significantly differ in sizes, numbers of classes and attributes. All those databases contain real-world data and are used for classification and testing of new and existing algorithms in Machine Learning. So, they are a kind of standard for new ML algorithms.

4. PLAN OF EXPERIMENTS

To solve the tasks stated, it is planned to perform a series of these experiments with all databases:

1. To ascertain the dependence of classification accuracy of the ID3 algorithm with the combined GA taking into consideration the following settings:
   - population size,
   - population string size,
   - mutation probability.
2. To check the classification quality of the algorithm depending on the number of iterations when constructing an ensemble of classifiers.
3. To analyse decision trees constructed with the help of standard ID3 algorithm and ID3-GA algorithm..
4. To compare the results of testing certain databases with other ML algorithms.

5. ID3 BEHAVIOUR EXAMINATION DEPENDING ON GA TUNING

To examine the dependence of classification accuracy of the ID3 algorithm with the combined GA algorithm on the main settings (population size, population string size, mutation probability , etc.), a number of experiments were performed. For each database from the repository, a table of results was formed on the basis of a group of tests (see Table 1). Each cell of the table represents one run of the algorithm execution procedure and the minimal error extent which was obtained during that run. In the columns *20 iterations* and *50 iterations* the results of run with 20 and 50 iterations , respectively, are given. The size of population in the experiment is varied from 5 through 20 with step 5 and enables algorithm behaviour's examination when the depth of the decision tree is restricted. The probability of mutation is varied from 0 through 10% with step 5.

When analysing the experimental results, it can be stated for sure that the use of mutation in the procedure of genetic algorithms provides good results. The error significantly differs at the 10% and 0% mutation. It can be explained by the fact that mutation makes the population renew sufficiently and -due to that- to test different attributes. However, in certain cases, especially when the size of population (the number of tree levels) is large, the 0% mutation shows smaller error, than the 5% and the 10% mutation.

The most interesting results were obtained when studying the influence of the population size (decision tree depth). It was assumed a priori that the deeper the tree is, the more correctly the decision tree looks and the more instances can be described. Approximately the same results were obtained for most of databases, however certain results differed greatly from the assumption made. Moreover, it is worth testing the dependence of the error extent on the decision tree depth only in the case when the number of attributes is about 20, and enough values of attributes are available. Thus, as a whole , all the databases under examination were divided into four groups:

1. Databases where there is no significant border between the results obtained for the populations of the size 5, 10, 15 and 20 strings (Cleve, Glass,

Hepatitis, Horse-colic, Ionosphere, Labor-neg, Sonar). The indicator of the mean value of the error did not change sufficiently depending on the population size.

2. Databases, where this tendency can be seen clearly: the less the number of decision tree levels is, the smaller error extent is obtained (German-org, Hypotiroid). Table 2 represents testing results for the German-org database.

Table 2. Testing results for the German-org database (20 and 50 iterations)

Population size	Probability of mutation, %	String size in a population (number of attributes)							
		20 iterations				50 iterations			
		18	25	32	Mean value	18	25	32	Mean value
5	10	0.2524	0.2766	0.2509	0.2600	0.2524	0.2748	0.2509	0.1293
	5	0.2810	0.2554	0.2584	0.2649	0.2809	0.2554	0.2584	0.1293
	0	0.2810	0.2673	0.2808	0.2764	0.2810	0.2673	0.2808	0.1401
10	10	0.2988	0.3017	0.2912	0.2972	0.2988	0.3017	0.2837	0.1538
	5	0.2897	0.2972	0.2868	0.2912	0.2897	0.2943	0.2868	0.1483
	0	0.2959	0.2988	0.2989	0.2979	0.2959	0.2988	0.2989	0.1503
15	10	0.2973	0.2914	0.3092	0.2993	0.2973	0.2914	0.2958	0.1490
	5	0.3033	0.3110	0.2942	0.3028	0.3033	0.3110	0.2717	0.1558
	0	0.2868	0.2943	0.2974	0.2928	0.2868	0.2943	0.2974	0.1639
20	10	0.3061	0.3005	0.2927	0.2998	0.3061	0.2839	0.2927	0.1531
	5	0.3049	0.3019	0.3033	0.3034	0.3049	0.3019	0.3033	0.1544
	0	0.3078	0.3078	0.3047	0.3068	0.3077	0.3078	0.3047	0.1483
Mean value:		0.1454	0.1541	0.1485	0.1493	0.1436	0.1528	0.1476	0.1480

3. Databases which showed a decrease of error extent when the size of population and, respectively, of possible decision tree grew, which gives evidence of that a more branched tree has to produce a more qualitative result. (Anneal, Auto, Chess, Soybean-large, Vehicle).
4. Databases Breast-w, Breast-cancer, Crx, Diabetes, Heart, Iris, Pima, Solar, Vote, and Zoo cannot be taken into consideration in that experiment because they are described with a number of attributes smaller than 15.

6. DECISION TREE ANALYSIS

To analyse the decision trees constructed with the help of standard ID3 algorithm and those constructed using ID3-GA combination, let us use the Zoo database that is one of the smallest databases of the repository. Zoo is a theoretical database containing 17 Boolean attributes and 100 instances. Attribute *Type* is a class. Due to its size, the database is best suited to our experiment. Let us divide the database into two sets: a training set and a test set (67 and 34 instances, respectively).

Then a classifier is constructed using a combination of ID3 and GA. For that, the following parameters of learning procedure are set: number of iterations is 50, size of population is 20, string size is 4 attributes but mutation probability is 10%. In the course of learning the most optimal generation was obtained at the 15^{th} iteration and the average error was 4.3% (see above for the classifier quality analysis depending on the iteration number). When testing that optimal population on a test set of instances, a result was obtained shown in Table 2.

Table 3 demonstrates the results of simple implementation of ID3 using the same learning and test set.

Table 3. Zoo – the results of ID3-GA testing

Classifying (% done): 10% 20% 30% 40% 50% 60% 70% 80% 90% 100% done.
Number of training instances: 67
Number of test instances: 34. Unseen: 17, seen 17.
Number correct: 32. Number incorrect: 2
Generalization error: 11.76%. Memorization error: 0.00%
Error: 5.88% +- 4.10% [1.63% - 19.09%]
Average Normalized Mean Squared Error: 5.88%
Average Normalized Mean Absolute Error: 5.88%
Classifying (% done): 10% 20% 30% 40% 50% 60% 70% 80% 90% 100% done.
Number of training instances: 67

Number of test instances: 34. Unseen: 17, seen 17.
Number correct: 29. Number incorrect: 5
Generalization error: 23.53%. Memorization error: 5.88%
Error: 14.71% +- 6.17% [6.45% - 30.13%]
Average Normalized Mean Squared Error: 14.71%
Average Normalized Mean Absolute Error: 14.71%

When comparing the results of the two algorithms execution, it can be noted that ID3 in combination with GA has shown very good results in testing on the test set, Average Normalized Mean Squared Error was 5.88% against 14.71% shown in case of ID3 for the same set.

On the other hand , upon the analysis of decision trees it was concluded that a tree constructed as a result of simple implementation of ID3 was the smallest. Both decision trees for the Zoo database consist of 7 levels, however, the number of nodes in the ID3-GA combination is 23 (14 of them being terminal nodes) but in the simple ID3 implementation it is equal to 19 (12 nodes are terminal nodes). As our investigation shows, this is a normal situation, because when a tree is constructed by a simple ID3, it selects a local maximum, which often leads to simpler decision trees. Moreover, a population constructed with the help of ID3, enables weaker attributes to participate in decision tree construction, which affects testing results of both trees. Similar tendency can be seen when testing other databases (see Table 4). From this table it follows that the results obtained when testing the combination of the ID3 algorithm and GA on a test set turn out to be better except for certain databases.

Table 1. Decision tree analysis for ID3-GA and ID3 algorithms

Database	ID3 in combination with GA		ID3
	No. of the best population	Number of nodes	Number of nodes
Anneal	41	65	40
Auto	4	56	45
Br. cancer	38	276	171
Breast	49	37	27
Chess	48	177	85
Cleve	17	86	51
Crx	12	159	113
Diabetes	14	199	136
German-org	44	293	229
Glass	5	71	53
Heart	42	67	47
Hepatitis	12	27	23
Horse-colic	38	98	63
Hypothyroid	41	49	47
Ionosphere	21	23	29
Iris	4	9	9
labor-neg	3	10	10
Pima	44	211	149
Solar	45	74	80
Sonar	48	31	25
Soybean	4	141	111
Vehicle	46	185	137
Vote	26	64	43
Zoo	15	23	19

As a last experiment, let us compare the results of the new algorithm and standard ML techniques (see Table 5). In contrast to the previous experiments, this table shows a 10-fold cross validation result for 50 iterations [5]. The following algorithms were employed for comparison: ID3, C4.5, Bagged-C4.5, Boosted-C4.5, Naive Bayes classifier and a neural network. The data for Bagged-C4.5 and Boosted-C4.5 were taken from [1] since the authors failed to find implementations of these algorithms. That paper presents the data on the experiments with ten runs of 10-fold cross-validation. According to Quinlan [1], such a run is optimal for those algorithms. Testing results for neural networks are only available for the databases containing real attributes

.

Table 2. Comparison of ID3-GA with other ML methods

Database	Testing error (%)						
	ID3-GA	ID3	C4.5	Bagged-C4.5	Boosted-C4.5	Naive Bayes	Perceptron
Anneal	0.67	2.17	7.33	6.25	4.73	8.20	
Auto	15.50	15.34	37.68	19.66	15.22	41.80	
Br. cancer	28.84	30.85	25.26			35.08	
Breast-w.	3.21	5.14	4.29	4.23	4.09	4.93	5.15
Chess	2.11	0.09	0.47	8.33	4.59	13.00	
Cleve	21.26	28.23	22.77			17.84	
Crx	16.12	20.20	17.00			22.04	
Diabetes	26.36	31.83	30.86	23.63	28.18	23.64	30.47
German-org	28.97	34.53	25.15			27.17	

Glass	31.00	34.43	37.50	27.01	23.55	49.90	
Heart	20.00	27.78	16.67	21.52	21.39	18.33	
Hepatitis	15.73	16.58	19.23	18.52	17.68	13.67	
Horse-colic	18.33	23.33	14.71			20.00	
Hypothyroid	1.33	1.51	0.76			1.90	
Ionosphere	6.40	9.00	11.97			17.04	17.95
Iris	2.00	6.00	8.00	5.13	6.53	3.00	
labor-neg	7.50	30.00	17.65	14.39	13.86	17.50	
Pima	28.52	30.67	23.44			25.59	23.44
Solar	30.54	34.41	26.85			37.21	
Sonar	19.01	25.42	25.71	23.80	19.62	32.68	20.00
Soybean	7.01	10.99	10.53	7.58	7.16	16.70	
Vehicle	25.19	27.85	32.27	25.54	22.72	53.20	
Vote	5.00	7.33	2.96	4.37	5.29	11.00	
Zoo	4.29	7.47	14.71			10.44	
Mean value:	15.68	19.21	18.07			21.74	

7. CONCLUSION

This study presents a comparison of a new algorithm developed as a combination of genetic algorithms and a method of decision tree construction, ID3, with four standard inductive learning algorithms including the C4.5 algorithm and its bagged and boosted versions. The analysis was performed using 24 databases of the Irwin repository containing information from real-world domains. Each of the sets is analysed or described in statistical medical literature or in the inductive learning related works.

The main result of the study can be stated as follows. It was proved that "hill-climbing" problem solving yielded sufficiently better characteristics of the classifiers. As can be seen from Table 5, the mean error for the 24 databases selected is 15.68% for the ID3 algorithm combination with genetic algorithms and 19.21% for the simple ID3 implementation. These results are especially important because they confirm that the approach under consideration can serve as a basis for other greedy-search algorithms improvement, including C4.5 and CART [5].

An analysis of the decision trees constructed with the help of the new algorithm and by ID3, is performed. As a result of the analysis, these conclusions can be made:

- A decision tree constructed with the help of ID3 on the basis of genetic algorithms is more branched. It only partly resembles a decision tree created by a standard ID3. On the one hand, such a tree represents all the peculiarities of the training set more accurately. On the other hand, an analysis of that tree turns to be more difficult for the experts.
- Classifier construction on the basis of genetic algorithms requires more training cost compared to standard ML algorithms. For example, for ID3 in combination with GA it constitutes 20-50 efforts of ID3.

8. FUTURE WORKS

Positive results of the experiments performed with decision tree construction on the basis of procedures of genetic algorithms are hopeful and encourage making further steps in their investigation.

The authors intend to conduct a number of experiments in that direction, namely:

1. To examine the methodology of decision tree construction on the basis of genetic algorithms for other ML methods, in particular, for C4.5 and CART.
2. To develop a new evaluation system for the population string's fitness as the old one does not meet the requirements laid down.
3. To implement and study the two other mentioned techniques for operation with classifier ensembles and for decision tree construction on the basis of genetic algorithms:
   - Voting procedure, when each new instance passes the procedure of classifier ensemble classification. The class wins which has the largest number of votes.
   - Construction of a classifier ensemble based decision tree that is common for all the classifiers.

9. REFERENCES

[1] J. R. Quinlan. *Bagging, Boosting, and C4.5*. University of Sydney, Sydney, 1998

[2] T. M. Mitchell. *Machine Learning*. The McGraw-Hill Comp., New York, 1997

[3] J. R. Quinlan. *C4.5: Programs for Machine Learning*. San Mateo, Morgan Kaufman, 1993

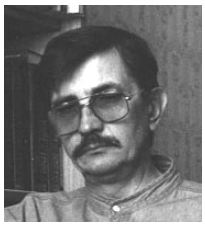
[4] L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone. *Classification and Regression*

Trees, Wadsworth Int. Group, Belmont, CA, USA, 1984

[5] S. M. Weiss. I. Kapouleas. An Empirical Comparison of Pattern Recognition, Neural Nets, and Machine Learning Classification Methods. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pp. 57-74, 1989

[6] Y. Kornienko and A. Borisov. Production Rules Induction Algorithm Based On The Finish Learning Principle. *Fourth International Conference on Application of Fuzzy Systems and Soft Computing*, Siegen, Germany, June 27-29, pp. 287-292, *2000*

Yuri Kornienko *is a Ph.D. Student in the Faculty of Computer Science and Information Technology at RigaTechnical University (Latvia). He received his M.Sc. degree in Decision Support Systems from Riga Technical University in 1997. His research interests include inductive learning, classification trees, regression trees and medical diagnostics. He has 8 publications in the area.*



Arkady Borisov *is Professor of Computer Science in the Faculty of Computer Science and Information Technology at Riga Technical University (Latvia). He received his Ph.D. degree in Technical Cybernetics from Riga Polytechnic Institute in 1970 and Dr.hab.sci.comp. degree in Technical Cybernetics from Taganrog State Radio-Engineering University (Russia) in 1986. His research interests include inductive learning, fuzzy set theory and applications, artificial neural networks, genetic algorithms, and adaptive agents. He has over 180 publications in the area. He is Past President of the Baltic Operations Research Society, former President of the Baltic Operations Research Society (1996-1998), Member of the Latvian Automatic National Organisation, Member of the Editorial Board of Automatic Control and Computer Science, Member of the Editorial Board of Computer Science of Moldova, Member of the Editorial Board of Scientific Proceedings of Riga Technical University.*