Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 15 (2011) 3658 – 3662

Procedia Engineering

www.elsevier.com/locate/procedia

Advanced in Control Engineering and Information Science

A new genetic programming algorithm for Building decision tree

Li Yi[a] and Kang Wanli[b,] a*

[a] [b] *ShiJiaZhuang Vocational Technology Institute, ShiJiaZhuang, 050000, China*
Hebei Vocational College For Correctional Police , HanDan，056000, China

Abstract

Genetic programming (GP) is a flexible and powerful evolutionary technique with some special features that are suitable for building a classifier of tree representation. However, unsuitable step size of editing operator will destroy the continuity of the evolution. In this paper, we propose a multiage genetic programming (MGP) algorithm to build a classifier on a given training set. Individuals are grouped into different groups according to their ages (tree size). The competitions between individuals are limited in the same groups. That prevents the structure editing operators from destroying the continuity of the evolution. The experimental results showed that the MGP algorithm is superior to the traditional genetic programming algorithm (GP) in building decision tree.

Keywords--Genetic programming; decision tree; grouping; representation;

1. Introduction

GP is a soft computing search technique, which was used to evolve a tree-structured program toward minimizing the fitness value of it. The distinctive features of GP make it very convenient for classification and the benefit of it is the flexibility, which allows the algorithm to be adapted to the needs of each particular problem. In data mining, many GP algorithms were proposed to build a decision tree for classification. However, when the optimal solution is a very full or narrow tree or the structure of it is very sparse, it is very hard for algorithms to get a satisfying result efficiently [1,2]. And in general, the

* Corresponding author. Tel.: 086-15209255838;
E-mail address: liyi_sjz@126.com.

doi:10.1016/j.proeng.2011.08.685

convergence rate coming from the pressure of the selection operator is largely greater than that of crossover and mutation. So, it always leads to a local maximum solution, not a global one.

1.1. GP

Genetic Programming is a systematic method to make computers solve problem automatically and was invented by Koza [3]. In GP, the populations of computer programs were generated automatically and the general operators used in it contain reproduction, crossover, mutation, gene duplication, and gene deletion. It can be seen as an extension of GA [4]. The search space of GP was largely decided by the choice of components and the fitness function. At the same time, the difficulty and the success rate were also decided by them too. The control parameters include population size, crossover rate etc. The termination criterion was defined as a rule for stopping. The typical termination rule is to stop when a satisfying solution was found or a given number of generations have arrived.

1.2. Decision Tree

Decision trees are one of the most frequently used representations for classifiers. Usually, the procedure of building a decision tree is a top-down sequential optimization procedure which begins from the root node. Nodes of the tree were split on the measure of the gain in likelihood on the given training dataset. The splitting process is repeated until the likelihood gain decreases to a given threshold. We can use the minimum occupation count to ensure that sufficient training data were associated with a terminal node [5].

2. Multiage genetic programming

In the proposed MGP algorithm, we maintain a series-parallel hybrid evolutionary process, in which the decision trees were seen as individuals of different ages (the size of the tree). Every generation was grouped into different groups on age. As the increasing of age, the fitness of the individuals becomes more and more large. The fitness values were used for selecting good individuals within the same group. The competitions between different individuals are permitted only in the same groups. So, the selection pressure was limited in a special area (group) and the operations of crossover and mutation will not destroy the continuity of the evolution.

Definition 1 (Group): Individuals in a population were grouped into different group, $group_g$, by the parameter Δage. The age span of individuals in the same group is defined as below:

$$Span(group_i)=[\Delta age\times(g-1),\ \Delta age\times g] \quad (1)$$

Definition 2 (Age): The age of a tree, age_i, is the number of the terminals appeared in the tree i.

Definition 3 (Fitness): Let $N_{correct(i)}$ be the number of correctly classified samples with individual i, N_{total} be total number of the training samples. The fitness of an individual i was defined as follow:

$$Fitness(i)= N_{correct(i)}/(LOG(age_{(i)})\times N_{total}) \quad (2)$$

The proposed multiage genetic programming algorithm for classification is presented below and is illustrated in Fig.1:

Step1. Randomly initialize a population P_t, $P_t=p_{1,t}+p_{2,t}+\ldots+p_{M,t}$. (t=0). Each individual $p_{i,t}$ (i=1,2,…,M) in it is a decision tree.

Step2. Select individuals in population P_t to form different groups, $group_{g,t}$. Each individual $p_{i,t}$ in a group has the same age. The age is the size of the tree.

Step3. Calculate the fitness values of individuals, $p_{i,t}$ in different groups.

Step4. Do a traditional selection, crossover and mutation operations within each group.

Step5. Integrate the individuals $p_{i,t}$ from different group $group_{g,t}$ into one population P_{t+1}, $P_{t+1}=p_{1,t+1}+p_{2,t+1}+\ldots+p_{M,t+1}$.

Step6. If termination condition is met, stop. The termination condition is that the classification accuracy is greater than a fixed threshold or the maximum generation arrived. Otherwise, t=t+1; Go step 2.

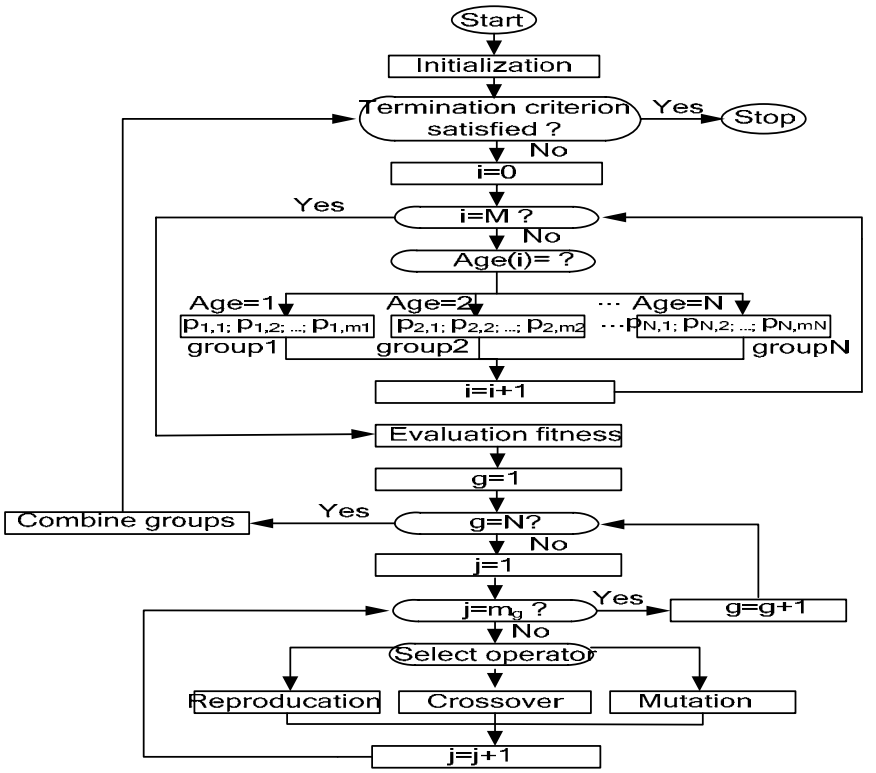

Figure 1. The Proposed MGP Algorithm

3. Experiment

In this section, we look on the combination of classification accuracy and age of a tree as the fitness function. The experiments were conducted on a 3.0 GHz Pentium PC with 512MB of memory running with Microsoft Windows XP. The datasets used were obtained from the UCI Mache Learning Repository [6].

3.1 Different parameters

For MGP, ⊿age is the age span between two continuous groups. We examined the effects of different values of ⊿age with dataset Pima Indians diabetes. From Fig.2 we can see that, for the same dataset, the proposed MGP algorithm can not find the optimal individual within fixed generations (250 generations) if the parameter ⊿age was set to a high value (⊿age =4). This is because that when ⊿age was set to a higher value, individuals with different ages will be permitted to compete in a same group. The convergence rate coming from the pressure of the selection operator is greater than that of crossover and mutation. So, the algorithm with a higher ⊿age will not find the optimal solution and this leads to a local maximum solution.

There are two phases exists in the evolutionary process. In the former phase, the competition between individuals promotes the increasing of the fitness of individuals which leads to an increasing of the tree

size. In the later phase, due to the use of a complex fitness function which combines accuracy and tree size, the tree size keeps decreasing. Since the competition between individuals was limited in a smaller area, the MGP algorithm with a small value of ⊿age converges faster than that with a large one.

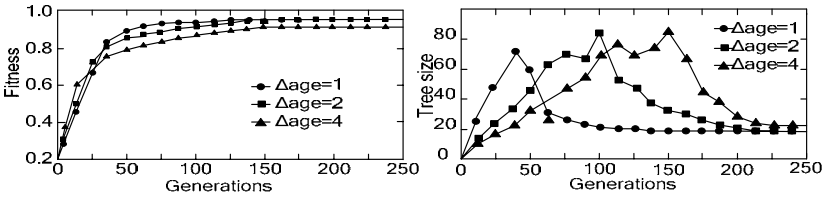

Figure 2. The Different Parameters ⊿age

3.2 Grouping and No-grouping

In Fig.3-A and B, the effects of selection within a group or within a population were examined. It is very obvious that the algorithm can find the optimal solution within the given generations based on the selection manner of group. However, for the manner of population, the algorithm could not get the optimum. The fitness of a larger tree is usually greater than a smaller one. But this does not indicate that the small one is a truly bad one. As the evolution proceeds, when the small tree grows up, perhaps the fitness of it will greater than the larger one. But with the evolutionary manner of doing selection within the whole population, the tree with a small size will lost the chance to grow up.

Additionally, we investigated the two operation manners of crossover and mutation: grouping and no-grouping. From the comparison we can see that when the crossover and mutation operators were limited in a group, the algorithm can still find the optimal solution. But the time it cost was longer than that cost by unlimited operators (performing within the whole population). But the algorithm with limited crossover and mutation operators could find a smaller decision tree than that with unlimited operators in restricted generations. We can see it from Fig.3-C and D. The reason of that lies in the protection of the groups which prevent the two operators destroying the continuity of the evolution.

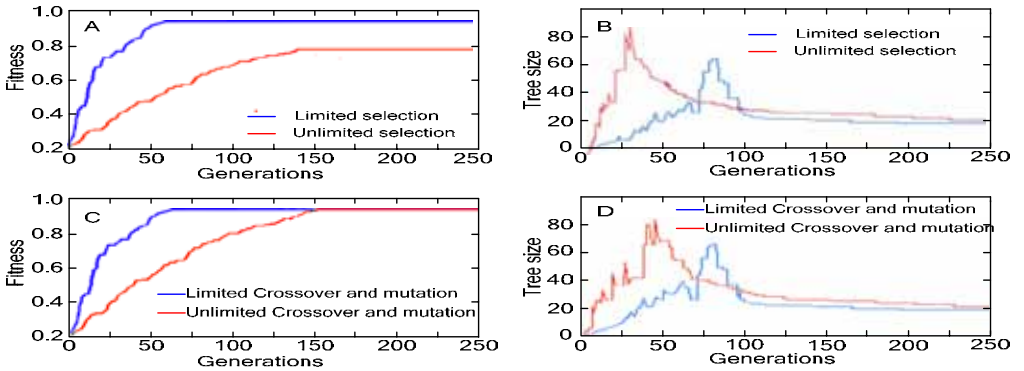

Figure 3. Grouping and No-grouping

3.3 Different datasets

In this section, we compared the three algorithms, GP, DTiGP [7], and MGP, with different datasets. The results were shown in Table.1, which illustrates the statistic results of computation time, the accuracy

of classification, the number of attributes, instances and classes of the dataset. For the three algorithms, the maximum generations, population size, maximum mutation rate were set to 100, 100, 0.1. We performed ten independent ten-fold cross-validation tests for each dataset. The algorithms were allowed to run 50 generation. The average accuracy and runtime over tests were presented in Table 1.From Table.1, we can see that MGP achieves better accuracy in most benchmarks. Moreover, as the increasing of instance number, the time cost by MGP became smaller than GP and DTiGP. The reason lies in the cost of grouping. For algorithm DTiGP, the accuracy of it was greater than GP too. But it cost more time to find the optimal solutions than GP.

TABLE I. COMPARISION IN DIFFERENT DATASETS

Dataset	Attributes	Instances	Classes	C4.5 Accuracy	GP		DTiGP		MGP	
					Accuracy	Time	Accuracy	Time	Accuracy	Time
Lung cancer	56	32	3	45.3±22.4%	55.8±12.6%	8	58.8±11.8%	9	72.4±3.4%	8
Zoo	17	101	7	88.6±3.8%	92.6±4.7%	26	93.3±3.8%	29	93.9±4.9%	27
Iris	4	150	3	94.1±4.7%	94.9±4.8%	55	95.1±3.8%	63	96.4±2.6%	56
Wine	14	178	3	90.9±7.1%	92.6±5.6%	84	93.4±4.2%	91	95.8±1.8%	85
Glass	9	214	7	65.5±5.8%	65.0±7.6%	206	68.0±4.7%	246	71.6±6.8%	194
Heart Disease	13	270	2	74.5±8.2%	78.8±6.8%	103	78.2±4.8%	116	79.9±5.2%	94
Ionosphere	34	351	2	92.0±7.9%	91.2±3.5%	155	92.8±2.8%	181	95.4±3.4%	138
Balance Scale	4	625	3	77.8±6.2%	98.6±0.8%	288	98.8±0.6%	326	100.0±0.0%	204
Breast cancer	9	683	2	95.1±1.3%	95.9±1.6%	326	97.2±1.0%	384	97.8±1.2%	263
Pima Indians	8	768	2	73.9±5.7%	70.2±4.2%	299	70.0±3.8%	346	68.9±3.4%	243
Car	6	1748	4	87.0±3.5%	91.0±1.5%	624	93.0±0.6%	719	94.6.0±1.7%	552
Waveform	21	5000	3	75.2±1.6%	77.2±2.0%	2702	82.5±3.6%	3911	92.5±2.0%	1782

4. Conclusion

Inspired by the natural evolutionary process, we proposed a multiage genetic programming algorithm, MGP, to deal with the problems of searching decision tree with maximal classification accuracy. In MGP, each population was divided into several groups. Each individual has a flexible life span during which individuals can grow and reproduce. The individuals compete with others only when they were in the same groups or at the same ages. By doing so, the competitions between individuals were limited in a special area (group) and the operations of crossover and mutation will not destroy the continuity of the evolution. The evaluation results have shown that, comparing with traditional GP, the proposed MGP approach achieved better performance in searching for classification trees.

5. References

[1]Daida, JM., Polito, JA., et al., What makes a problem GP-Hard? Analysis of a tunably difficult problem in genetic programming, Genetic Programming and Evolvable Machines,2001, vol. 2, 165-191.

[2] Daida, JM., Li, H., Tang, R., and Hilss, AM. What makes a problem GP-hard? Validating a hypothesis of structural causes. in Proc. Genetic Algorithms Evol.Comput.Conf., Lecture Notes Computer Science 2724, 2003, pp.1665–1677.

[3] Koza, J. R., 'Genetic Programming: On the programming of computers by means of natural selection'. MIT Press, Cambridge, 1992.

[4] Koza, J. R. 'Introduction to genetic progamming' Cambridge: MIT Press, 1994.

[5] Simard, M., Saatchi, S., and Grandi, GD., The Use of Decision Tree and Multiscale Texture for Classification of JERS-1 SAR Data over Tropical Forest, IEEE Transactions on Geoscience and Remote Sensing, 2000, Vol. 38, No.5, pp.2310-2321.

[6] Newman, D. J., Hettich, S., Blake, C., and Merz, C., UCI Repository of Machine Learning Databases. Berleley, CA: Dept. Information Comput. Sci., University of California, 1998.

[7] Kőnig, R., Johanooson, U., Lőfström, T., and Niklasson, L., Improving GP Classifcation Performance by Injection of Decision Trees, Evolutionary Computation (CEC), 2010, IEEE Congress on, Page: 1-8