International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 2, Issue 12, December 2012)

To Accomplish Amelioration Of Classifier Using Gene-Mutation Tactics In Genetic Programming

Ankit Bakshi[1] Pallavi Pandit[2] Santosh Easo[3]

[1]*Department of Information Technology, Medicaps, Indore*
[2]*Department of Computer Science & Engineering, Medicaps, Indore*

***Abstract*— A phenomenon for designing classifier for three or more classes (Multiclass) problem using genetic programming (GP) is multiclass classifier. In this scenario we purported three methods named Double Tournament Method, Gene-Mutation Method and a Plain Crossover method. In Double Tournament Method, we pick out two idiosyncratic for the crossover operation on the basis of size and fitness. In Gene-Mutation tactic we are propagating two child from single parent and selecting one of them on the basis of fitness and also bring into play elitism on the child so that the mutation operation does not degrade the fitness of the distinct, whereas in Plain Crossover we select the two child for the succeeding generation on the basis of size, depth and fitness along with elitism on each step from the six child which is generated during crossover. To exhibit our approach we have designed a Multiclass Classifier using GP by taking some standard datasets. The results attained show that by applying Plain crossover together with Gene-Mutation refined the performance of the classifier.**

***Keywords*— Gene-Mutation, Elitism, Double Tournament, Plain Crossover.**

I. INTRODUCTION

Genetic programming (GP) is a progressive ciphering approach that significantly resolve problems, So as the user being unknown of the form or structure of the solution beforehand. GP has already spawned numerous interesting applications such as [1]–[2]. GP has been used by many authors [3] to design classifiers or to generate rules for two class problems. Genetic programming has seldom been used for Multi-class classification purposes. Previously, it was achieved by separating the output manually for different classes or expanding an n-class problem to n two-class problems. Multiclass or multi-label classification is the special case within statistical classification of assigning one of several class labels to an input object. An approach takes an conjoined view of all classes when the GP evolves. A Multitree representation of chromosomes is used. GP is a method of solving problems using computers through a correlation of natural selection. A way in which computers self resolve problems and even the computer being unknown of the form or structure of the solution ahead of time.

GP initiate by procreating some initial population. Genetic programming is a cluster of methods for the automatic generation of computer programs that resolve carefully particular problems, but highly abstracted principles of natural selection. It is the composite species of computer programs, where only the associated more successful idiosyncratic only get the opportunity so as to pass on genetic material that is programs and program chunks to the next generation. Genetic programming starts from a high-level statement of the needs of a problem and tryout to produce a computer program that resolves the difficulty. Tournament selection is the most usual method for the selection of the individuals in GP**.** But in our paper we have come up with a significant type of tournament known as Double tournament. Mainly in double tournament method we randomly select 9 individuals from the population and then from those selected individual we select 5 on the basis of size and from those 5 individual we select two on the basis of fitness.

In this we also come up with a plain crossover technique and a Gene-mutation activity that reduces the devastating nature of common genetic operations. Mutation influence an individual in the population. It can replace a whole node in the selected individual, or it can replace just the node's information. Mutation is a genetic operator that alters one or more gene values in a chromosome from its initial state. This can result in entirely new gene values being added to the gene pool. With these new gene values, the genetic algorithm may be able to arrive at better solution than was previously possible. Mutation is an important part of the genetic search as help to prevent the population from decaying at any local optima. Mutation occurs during evolution according to a user-definable mutation probability. Gene-Mutation operates on a single individual from the population. Mutation operations mainly terminate by decreasing the fitness of an individual because the new material is untried. They likely to be used less than crossover; however, they are still important to a successful GP run.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 2, Issue 12, December 2012)

In Gene-Mutation technique we generate the two individual from the selected parent and from the two generated individual we reject one individual on the basis of fitness and the remaining individual is now compared with the parent if the fitness of the parent is better than the selected individual than with a probability of 0.5 we transfer the parent to the next generation. Plain crossover enables the algorithm to extract the best genes from different individuals and recombine them into potentially superior children. An individual which has high fitness rate and low size and depth will takeover.

II. Proposed Work

In this paper we have sketched a Multiclass Classifier using Plain Crossover with elitism and Gene-Mutation tactics.

A. Initialization

Firstly in genetic programming we have to generate initial population so as to perform operations on that. Each of the trees for each individual is initialized randomly using the function set F which consists of arithmetic functions and the terminal set T containing feature variables and constants. The function set F and terminal set T used here are as follows:

F = {+, -, *, %} and
T = {feature variables, R}

Where R contains randomly generated constants in [0.0 to 10.0].We have initialized trees using the ramped half-and-half method.

B. Fitness Evaluation

Fitness function is the most challenging and most valuable idea of genetic programming The fitness function concludes how fine a program is able to resolve the problem. GP is guided by the fitness function to search for the most effective computer program to solve a given problem. A simple measure of fitness has been adopted for the pattern classification problem. Fitness is mainly calculated by dividing the number of samples classified correctly to the number of samples used for training during evolution.

C. Selection

In selection process we select the parent for crossover on the basis of double tournament selection method, whereas in mutation the parent is selected randomly and for reproduction we use roulette wheel selection method. In gene-mutation we select one best child on the basis of fitness and reject another one.

In simple crossover the remaining child is wiped out step by step and the best child is selected.

D. Algorithm for Double Tournament

1. Initially we pick out 9 individual from the population in an irregular manner or we can say randomly.
2. Secondly we pick out 5 individuals out of 9 individuals on the basis of size.
3. After this on the basis of fitness we discard 3 individuals from 5 and the individual with smaller size are selected.
4. Now finally we get 2 individual with highest fitness.

E. Gene-Mutation

G-mutation is simply defined as "random changes in genetic material".Mutation plays an important role in evolution, genetic programming is a distinct technique in which we randomly select an individual from the population and from this randomly selected individual we get the two subordinate children. From this two generated children one is discarded on the basis of fitness the children with the lower fitness is rejected. now we valuate the selected individual with the parent and examine in contrast its fitness with the parent, if the fitness of parent is greater than the child than with a probability of 0.5, parent is transferred to the next generation else child is transferred to the next generation. By applying this Gene-mutation technique we are sure that the generated individual does not reduce the fitness and the individuals are diversified as well.

F. Algorithm for Gene-Mutation Method

1. Initially we give birth to the two children from the randomly generated individual.
2. Secondly the individual with greater fitness is discarded and the lesser one is discarded.
3. Lastly we apply the most important step, that is elitism. In elitism we analyze the selected individual with parent and if the fitness of parent is finer than the child, than with a probability of 0.5 we transfer the parent to the next generation else we transfer the child to the next generation.

G. Plain Crossover

The plain crossover is performed by the two individual which are chosen by the double tournament, so as to generate six children. As we move forward, from these six generated children, two are rejected on the basis of depth and size, though children with the smaller depth and size are selected. Further from these 4 selected children discard two on the basis of fitness, as similarly children with the greater fitness are accepted.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 2, Issue 12, December 2012)

After this, apply the elitism on the generated children and analyze the fitness of the children with the parent. If the fitness of the parent is greater than the children than with the probability of 0.5 parent is transferred to the next generation else children is shifted to the further generation.

H. Algorithm for Plain crossover

1. Indiscriminately select individual from the population for double tournament selection.
2. Fetch the best two individuals of the double tournament for plain crossover operation.
3. Select the subtree or a node from a parent randomly and place it at three different positions in another parent and generate the three children and the same process is repeated with another parent also, so the total number of generated children are six.
4. As we will move further we will select the best four child among the six child on the basis of depth and size, the child with the smaller depth and size are picked up.
5. And finally the two children are chosen on the basis of fitness, the individuals with the higher fitness are selected.
6. Finally we apply the elitism on the selected individual if the fitness of the parent is better than the child than we transfer the parent with a probability of 0.5 to the next generation otherwise we transfer the child to the next generation.

I. GP Algorithm with Plain Crossover and Gene-Mutation

1. GP begins with a randomly generated population of solutions of size N.
2. A fitness value is assigned to each solution of the population.
3. A genetic operator is selected probabilistically.
4. If it the reproduction operator, then an individual is selected (we use fitness proportion based selection) from the current population and it is copied into the new population. Reproduction replicates the principle of natural selection and survival of the fittest.
5. If it is the crossover operator, then we apply the Plain Crossover.
6. If the selected operator is mutation, we apply a Gene-mutation.
7. Continue 3. Until the new population gets solutions.
8. This completes one generation.
9. Step 3 to 7are repeated till a desired solution is achieved. Otherwise, terminate the GP operation after a predefined number of generations.

III. EXPERIMENTAL RESULTS

We have designed a MultiClass Classifier to demonstrate our results. We have used Java 6.0 as a front end tool and Oracle 10g as a back end tool to develop our project. We have used 5 real data sets for training and validating our methodology. These are IRIS, WINE, ABALONE. Table I gives a brief description about all the data sets used.

A. Data Sets

IRIS Data: This is the well-known Anderson's Iris dataset. It contains a set of 150 measurements in four dimensions taken on Iris flowers of three different species or classes. The classes are Iris Setosa, Iris Versicolour and Iris Virginica. The four features are sepal length, sepal width, petal length, and petal width. The data set contains 50 instances of each of the three classes.

WINE: This dataset contains 178 instances and 13 attributes. These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.

ABALONE: This dataset predicting the age of abalone from physical measurements. The age of abalone is determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope -- a boring and time-consuming task. Other measurements, which are easier to obtain, are used to predict the age. Further information, such as weather patterns and location (hence food availability) may be required to solve the problem. It has 4177 instances and 8 features.

TABLE I
DATASETS

Name of Data Set	**No of classes**	**No of Features**
IRIS	3	4
WINE	3	13
ABALONE	4177	8

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 2, Issue 12, December 2012)

PARAMETERS

Table II describes the common parameters used for all the data sets.

Parameters	Values
Probability of Crossover Operation	80%
Probability of Reproduction Operation	10%
Probability of Mutation Operation	10%
Population Size	100-400
Minimum Tree Depth	2
Maximum Tree Depth	6
Number of Generations	20-60

IV. Results

To demonstrate our approach we have designed a MultiClass Classifier using genetic programming. The training and testing of the classifier generated is done using real data sets. We have compared the outcome of our results with the conventional crossover and mutation method shown in Table III and found that our method outperforms the conventional crossover and mutation method and improves the accuracy of the classifier with a fair amount.

Table III
Comparison of Conventional Crossover and Mutation Method with Plain Crossover and Gene-Mutation Method

Name Of Datasets	Conventional Crossover and Mutation Method		Plain crossover and Gene-Mutation Method	
	Training Accuracy	Generalization Accuracy	Training Accuracy	Generalization Accuracy
IRIS	84.64%	82.36%	93%	91%
WINE	85.32%	84.45%	90.58%	89%
ABALONE	77.24%	71.62%	75.82%	76.68%

V. Conclusions

In this paper, we have purported a Double Tournament technique to select the parent for the crossover operation which helps in improving the overall fitness and applying the limit of size and depth on the individual which are selected for crossover operation. We are also using a unique gene-Mutation technique in which we are generating two children and applying the fitness limit on the two generated children and also applying the elitism on the selected individual due to which the mutation operation also can't reduce the fitness of the classifier and it also provides the diversity among the individuals. We also proposed a Plain crossover technique in which we generate the 6 child from two parent and simple eliminate the 4 child on the basis of size, depth and fitness and apply the elitism on the two children and compare with parent. To demonstrate our approach we have designed a MultiClass Classifier and presented the results on different datasets. To describe the usefulness of our approach, we have compared our method with the conventional crossover and mutation method and obtained adequate results in terms of exactness.

REFERENCES

[1] W. Banzhaf, P. Nordin, R. E.Keller, and F. D. Francone, Genetic Programming: An Introduction. San Mateo, CA: Morgan Kaufmann, 1998.

[2] R. Poli, "Genetic Programming for image analysis," in Proc. 1st Int. Conf. Genetic Programming, Stanford, CA, July 1996,pp. 363–368.

[3] D.Agnelli, A. Bollini, and L. Lombardi, "Image classification: an evolutionary approach," Pattern Recognit. Lett., vol.23, pp. 303–309,2002.

[4] Mengjie Zhang and Will Smart "Genetic Programming with Gradient Descent Search for Multiclass Object Classification".

[5] D J Nagendra Kumar, Suresh Chandra Satapathy, J V R Murthy "a scalable genetic programming multi-class ensemble classifier" World Congress on Nature Biologically Inspired Computing NaBIC 2009 (2009).