Novel GP Operator for GP Classification

Venkatadri M, Hanumat G Sastry, Dr. Lokanatha C Reddy

Abstract: Genetic Programming (GP) is one of the most promising Evolutionary Programming techniques for Data Classification. Despite GP is more expedient for Classification, still it suffers from certain drawbacks due to the lack of efficient GP Operators. Hence, this paper presents the various identified issues of the existing GP Operators and proposes the desirable characteristics for the new GP Operator for efficient Data Classification.

Key words: Evolutionary Programming, Genetic Programming, Data Mining, Classification, GP operators, New GP Operators.

1 INTRODUCTION

Since the emergence of massive data collection technology, the implementation of modern Data Mining (DM) techniques are always playing vital role in the analysis of large volumes of data (Zeta Bytes) with complex data formats. Among the various DM techniques, Classification is the most studied issue, due to its significance in decision making process [1]. Genetic Programming is an evolutionary programming strategy to address Classification task more effectively [2] [4] [5] [6] [14]. Among the various steps of GP Classification, GP Operators play a vital role in carrying out the Classification task. At present there are certain issues with the existing GP Operators and there is a great need of new GP Operators for efficient Classification [16]. Hence this paper proposes the desirable characteristics for new GP Operator by presenting the identified drawbacks of existing GP Operators. Section 2 provides the GP Classification process, Section 3 presents standard GP operators with their drawbacks, and section 4 presents a proposal for desirable characteristics for the proposed GP operator and Conclusion in section 5.

2 GP CLASSIFICATION PROCESS

The typical GP Classification Algorithm consists of subtasks like individual representation, applying the GP operators and classification accuracy computation [6]. The following diagram depicts the GP Classification process.

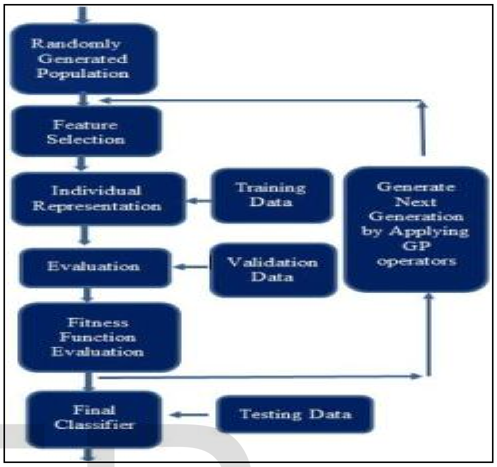
Randomly Generated Population
Feature Selection
Individual Representation
Training Data
Generate Next Generation by Applying GP operators
Evaluation
Validation Data
Fitness Function Evaluation
Final Classifier
Testing Data

Figure 1: Typical GP Classification Process

Among various subtasks of GP Classification algorithm, GP Operators play a vital role.

3 STANDARD GP OPERATORS AND THEIR ISSUES

The GP Operators are applied to the individuals of the population to give birth to the new population (next generation) [3] [12]. The GP Operators perform operations that simulate the natural genetic operations and manipulate the structures of individuals during evolution process. The following are the standard GP Operators [3] [10] [11].

1. Reproduction
2. Crossover
3. Mutation

The selection process of GP operators to form the new offspring from the current population is based on Probability theory. The Statistical Probability is calculated for each of the selection operators to generate the offspring in tree evolution process. The overall probability to create new offspring in the next generation of tree is equals to 1.

$P_m + P_c + P_r = 1$

Where:-

P_m : The probability of Mutation
P_c : The probability of Crossover
P_r : The probability of Reproduction

3.1. Reproduction

Reproduction operator selects individuals from the

population, preferably the fittest ones, and copies them into the next generation [3] [11]. The process of copying the fittest individuals to the next generation is called elitism.
The reproduction operator primarily involves making an exact copy of the individual and placing into the population [9]. The goal of reproduction is to increase the prevalence of individuals which have proven themselves fit to solve the problem. This gives good individuals a greater chance of being preserved, reducing the risk of losing fit individuals.

3.2. Issues in Reproduction Operator

Reproduction operator is important, but too much can really hurt a GP run [13]. Reproduction operator eliminates the diversity and prevents the further change in the evolution process due to its nature in selection of the fittest individuals for next generation.

3.3. Crossover

The Crossover operator is generally most prevalent operator used in GP [3] [11]. The objective of Crossover is to exploit the existing genetic material in a population. Crossover operator combines the genetic material from two parent program trees to generate two offspring trees. In each parent tree, the crossover point is randomly selected; the two offspring trees are created by swapping the sub-trees below the crossover point on each of the parent trees. This process can be seen in the following fig

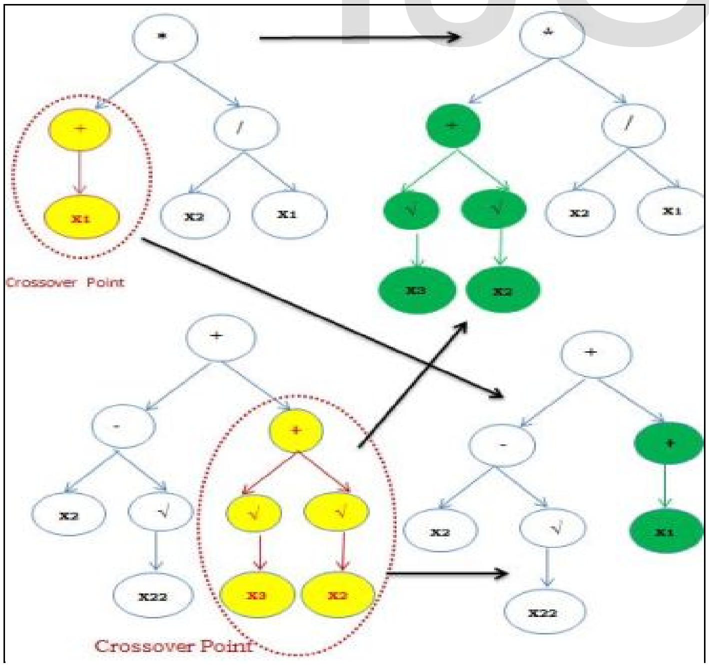

Figure 2: Evolution Process with Crossover Operator

3.4. Issues in Crossover Operator

The following are the major issues with the Crossover operator.

- Un-manageable fitness Evaluation
- Code Growth

After few generations with Crossover operation, the population would be with big individuals, carrying out fitness calculation on these big individuals is unmanageable and unfeasible. Crossover operation produces a large change in behavior of individuals; generally this nature tends to the Code growth problem [8] [15]. The large amount of code growth results an increase in average tree size without a corresponding increase in fitness [7].

3.5. Mutation

The Mutation operator performs random changes in the existing program Tree [3] . The Mutation operator selects an individual based on fitness proportional selection randomly from among the set of functions and terminals making up the original individual as the point of mutation [10] [11]. The mutation point, along with the sub-tree stemming from the mutation point, is then removed from the tree, and replaced with a new, randomly generated sub-tree. The new sub tree being inserted into the tree to form a new individual in the population described in the following fig.

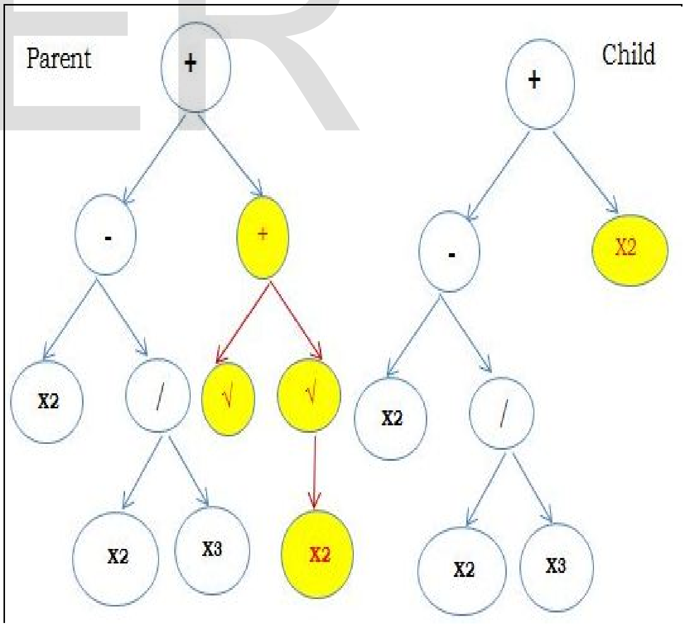

Figure 3: Evolution Process with Mutation Operator

3.6. Issues in Mutation Operator

The Mutation operator affects the individual by replacing a selected node randomly in the population [11]. Due to random selection behavior of the Mutation operator in evolution process, the occurrence of poor individual programs would increase. Fitness evaluation on poor individuals adds more complexity.

The following table presents the abstract of standard GP Operators and their issues.

Table 1: Standard GP Operators and their issues

GP Operator	Selection Behavior of the Individuals for next generation	Issues
Reproduction	Selects the fittest individual only	Prevents the evolution changes.
Crossover	Random Selection	Un-manageable fitness evaluation and Code growth
Mutation	Fitness proportional selection randomly from Function and Terminal sets.	Poor individuals and more time consumption for fitness evaluation.

4 Proposed GP Operator

To overcome above mentioned drawbacks of existing GP Operators, it is desirable to design new GP operator for efficient GP classification process. The following are the desirable characteristics for the proposed new GP operator for efficient GP Classification.

4.1. Desirable characteristics for New GP Operator

- The new operator should be constructive, so that the successor individuals would be better than their parents in evolution process.
- The new operator must preserve the syntactic correctness in each generation and must maintain the legal syntactic structure for new individuals.
- The new operator must be able to force population to converge to a better space of solution even maintaining the diversity for exploitation.
- The new operator must not lead to unmanageable individuals in order avoid the bloat problem.
- The new operator must yield the highest overall fitness before bloat puts an effective stop to evolution.
    - o A new operator must increase the likelihood of high fitness offspring without significantly increasing the computational cost.
    - o Be applicable to all kinds of problem-solutions and should not involve extra computation time.
- It should be easy to implement.

5 Conclusion

In this paper we have proposed a new GP Operator in order to improvise the GP Classification technique for efficient Classification. Presently, we are implementing the proposed GP Operator. In future, we will present the complete description, implementation details with test results of the proposed GP Operator.

6 ACKNOWLEDGMENTS

The authors thank the authorities of the University of Petroleum and Energy Studies, India and Dravidian University, India for the support extended to them for research in thrust areas of Computer Science.

References

[1] Xin dong Wu, Vipin Kumar et al, "Top 10 algorithms in data mining", Spinger-Verlag London Limited, 2007.

[2] Aleksandra Takac, "Application of Cellular Genetic Programming in Data Mining", Proceedings of Conference Knowledge, Brno, Czech Republic, 2003.

[3] John R, Koza , "Genetic Programming: On the programming of Computers by Means of Natural Selection", MIT Press , Cambridge, MA , USA, 1992.

[4] *Freitas, A, "A survey of evolutionary algorithms for data mining and knowledge discovery", Advances in Evolutionary Computation, 2002.*

[5] Nabil, H.,Kadhi, EL, "Data Mining Classification: The Potential of Genetic Programming", The Sixth International Multi-Conference on Computing in the Global Information Technology (ICCGI 2011), 2011.

[6] Venkatadri, M., Sastry, Hanumat G. & Reddy, Lokanatha C. "Genetic Programming for Data Mining Tasks", International Journal of Advanced Research in Computer Science (IJARCS), Issue Vol 3 No 2. 2012

[7] Peter A Whigam, " Implicitly controlling bloat in GP", IEE Transactions on Evolutionary Computation, Vol-14 Issue-2, 2010.

[8] T. Soule, R.B. Heekenodorn, "An Analysis of the causes of code growth in GP, Klwer Academic Publishers, 2002.

[9] C.J. Veenman, "Positional Genetic Programming", Thesis, Vrije University, Netherlands.

[10] Steven Matt Gustason, "An analysis of diversity in Genetic Programming", Ph.D Thesis, University of Nottinghan.

[11] Adan Robinson, "Genetic Programming: theory, implementation and evolution of unconstrained solutions", Hampshire College, 2000.

[12] Wolfgang Banzaf, Frank D. Francone, Robert E. Keller, and Peter Nordin, "Genetic Programming : An introduction" , Morgan Kaufmann, San Francisco , CA, 1998.

[13] M. Brameier and W.Banzhaf, "Explicit Control of Diversity and Effective Variation Distance in Linear Genetic Programming", Genetic Programming Proceedings of 5th European Conference (EuroGP 2002), Berlin, Springer –Verlag, LNCS , 2002.

[14] Jabeen, Hajira et al , "Review of Classification Using Genetic Programming", International Journal of Engineering Science and Technology. Vol.2 (2), 2010, pp 94-103.

[15] Smith, P.W.H. and Harries, K, " Code growth, explicitly defined introns, and alternative Selection schemes", Evolutionary Computation, Vol 6(4), 1998.

[16] J. M. Daida, H. Li, R. Tang, and A. M. Hilss, " What makes a problem GP-hard? Validating a hypothesis of structural causes", In

GECCO 2003, number 2724 in LNCS, Springer, 2003.