2013 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 22–25, 2013, SOUTHAMPTON, UK

ADAPTED GEOMETRIC SEMANTIC GENETIC PROGRAMMING FOR DIABETES AND BREAST CANCER CLASSIFICATION

Zhechen Zhu[*]*, Asoke K. Nandi*[†]

Electronic and Computer Engineering
Brunel University
Uxbridge, Middlesex, UB8 3PH, UK
{zhechen.zhu; asoke.nandi}@brunel.ac.uk

Muhammad Waqar Aslam[‡]

Electrical Engineering and Electronics
The University of Liverpool
Liverpool, L69 3GJ, UK
waqaraslam271@gmail.com

ABSTRACT

In this paper, we explore new Adapted Geometric Semantic (AGS) operators in the case where Genetic programming (GP) is used as a feature generator for signal classification. Also to control the computational complexity, a devolution scheme is introduced to reduce the solution complexity without any significant impact on their fitness. Fisher's criterion is employed as fitness function in GP. The proposed method is tested using diabetes and breast cancer datasets. According to the experimental results, GP with AGS operators and devolution mechanism provides better classification performance while requiring less training time as compared to standard GP.

Index Terms— Genetic programming, genetic operator, breast cancer diagnosis, diabetes detection

1. INTRODUCTION

Signal classification is an important application of Genetic Programming (GP). Since the introduction of GP [1], researchers have tried to exploit its potential in many classification problems [2]. In this paper, we focus on using GP as a feature selector and generator. As a feature generator, GP is given a set of original features and undergoes a supervised learning process to select and combine these original features. The combination is expected to produce a new feature which will enhance classification performance. The actual classification can then be completed with a suitable classifier. It has been successfully implemented in some existing literature which report better classification performance as compared to other classification solutions [3–7]. Though GP has yielded superior performance in these cases, its demand for high computational power and long training time has always been an issue. Therefore, a more efficient GP is needed.

A standard GP comprises several key parts: population initialization, fitness evaluation, parent selection, and reproduction. There are some studies which contribute to the improvement of the first three [8–10]. However, few consider alternatives to the standard reproduction operators (standard mutation and crossover). It is mostly due to the difficulty in understanding how genotype (solutions) and phenotype (fitness) are related. Standard genetic operators are based on syntax of solutions and operate in a random way. It is easy to doubt their efficiency considering the trial and error style. Meanwhile, some researchers have started to adopt a new perspective in establishing a connection between genetic operators and fitness landscapes [11–13]. The outcomes are promising new operators which use solution semantics instead of syntax.

In one of the recent works [13], Moraglio et al. give detailed analysis on semantic geometric operators and semantic fitness landscapes. Definition of Geometric Semantic (GS) operators are given for different problems. However, these GS operators are designed with fitness functions based on Euclidean and Manhattan distances in mind. Feature generation with Fisher's criterion evaluation is a rather different matter. Therefore we propose new Adapted Geometric Semantic (AGS) operators. The design of the AGS operators is largely inspired by the GS operators as well as the problem at hand. As the development of these operators is only on an experimental level, theoretical proof has not yet been established. Most of the conclusions drawn in this paper will be an abstract interpretation of the experimental results.

During the design of the AGS operators, it was found that they are prone to growth which creates continuously increasing demand for computational resource. To enable the operators to function in a sustainable style, a devolution mechanism is proposed to mimic the devolution of organs in the natural world where trivial organs are removed to reduce the individual's complexity as well as reducing the overall resource consumption. Some simplification method developed

[*]Zhechen Zhu would like to thank the School of Engineering and Design, Brunel University, for the financial support.

[†]Asoke K. Nandi would like to thank TEKES for their award of the Finland Distinguished Professorship.

[‡]Muhammad Waqar Aslam would like to acknowledge the financial support of the University of Azad Jammu and Kashmir, Pakistan.

978-1-4799-1180-6/13/$31.00

in the past can fulfill the same job [14]. However, we choose to adopt a new approach which is better tailored for the AGS operator and operates in a more intuitive way.

2. METHODOLOGY

2.1. Fitness Evaluation

Before the AGS operators and devolution mechanism could be discussed, it is helpful to establish the fitness evaluation method first. In this paper, Fisher's criterion is employed as a measure of feature quality. The combination of GP and Fisher's criterion for feature generation was first introduced in [15] by Guo et al. The actual expression of fitness function is given in Equation (1)

$$f_{AB} = \frac{|\frac{1}{m}\sum_{i=1}^{m} S_i^A - \frac{1}{n}\sum_{j=1}^{n} S_j^B|^2}{\frac{\sum_{i=1}^{m}(S_i^A - \frac{1}{m}\sum_{i=1}^{m} S_i^A)^2}{m-1} + \frac{\sum_{j=1}^{n}(S_j^B - \frac{1}{n}\sum_{j=1}^{n} S_j^B)^2}{n-1}} \tag{1}$$

where S_i^A and S_j^B are data samples from class A and B respectively. The variables m and n denote the total number of samples available to the training task from class A and B. Fisher's criterion is useful in feature generations to encourages between-class scatter while limiting within-class scatter. There are other ways of evaluating quality of generated features. Conducting mini classification tasks and using the results as fitness values could be a more accurate evaluation of feature quality. However, due to the need for multiple runs and large number of generations for analyzing the new operators, Fisher's criterion is preferred because of its relatively simple form. It is a balanced approach to evaluate feature quality without overburdening the GP program.

2.2. Adapted Geometric Semantic Operators

In [13], Moraglio et al. gave the definition of Geometric Semantic (GS) operators for fitness functions based on Euclidean and Manhattan distances. The expression of GS crossover is given as

$$T3 = (T1 \cdot PR) + ((1 - PR) \cdot T2) \tag{2}$$

where $T3$ is the child tree produced with parent trees $T1$ and $T2$. It is a linear combination with a random factor PR in the domain of [0,1].

In the case of feature generation, as the fitness function is based on neither of these two mentioned distance metrics, a new definition of GS operators is needed to exploit their potential in the new scenario. Treating each reproduction as a combination of two existing features, we relax the linear combination rule in [13] by replacing the addition with an operator randomly chosen from a pre-defined function pool. Then we end up with the following AGS crossover definition expressed in Equation (3)

$$T3 = g_c(PR \cdot T1, (1 - PR) \cdot T2) \tag{3}$$

where g_c is a mathematical operator drawn randomly from a pool with two-input operators. In the paper, the pool is limited to four basic types of operators: plus, minus, multiplication and division (protected). When two parent trees are combined, their output terminals are first multiplied by PR or $(1 - PR)$ then attached to the inputs of g_c. The output of g_c then becomes the new output terminal for the combined child tree. The graphical illustration of AGS crossover is shown in Figure 1.

In a similar way, we present an AGS mutation which is given in Equation (4)

$$T3 = g_m(T1, TR) \tag{4}$$

where TR is a randomly generated branch and g_m is a random mathematical operator. Figure 2 shows how this operation is done.

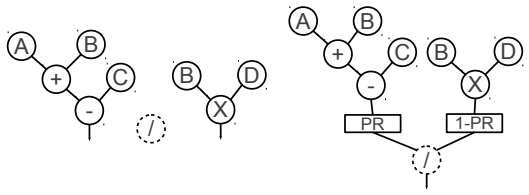

Fig. 1: AGS Crossover: The two trees on the left side are parent trees. The operator in dashed circle is the random operator. The right side tree gives the child of the AGS crossover. A, B, C, D are the input terminals (original features).

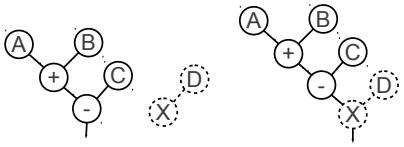

Fig. 2: AGS Mutation: The most left side tree is the parent tree. The branch in dashed lines is a random branch. The right side tree gives the child tree of the AGS mutation. A, B, C, D are the input terminals (original features).

2.3. Devolution for AGS Operators

Based on the definition of AGS operators, it is not difficult to see that the child tree always has increased depth (number of layers between top input terminal and output terminal) as compared to its parents. As the growth is inevitable, the computational complexity of AGS-GP after certain number of generations will be difficult to handle. Without a simplification mechanism, the implementation of such genetic operator is unpractical. For this reason, a devolution module is added to GP to control the growth.

Devolution or degeneration is a concept borrowed from biological evolution theory. It is believed that some time the

evolution of a species can change into a more "primitive" form. In many cases it means certain organs disappearing or being transformed into a less complex form. There is much evidence that such a process is beneficial for the overall fitness of a species and helps their chance of survival.

In this study, we implement the devolution on bloat trees where some trivial parts can be removed to off load the demand for computation power. When searching for the less significant parts, the branches at the furthest distance from the output terminals are considered as tail branches and have little impact on the overall fitness of an individual. There are different ways of defining a notion of distance between tail branches and output terminals. In this research, we use depth as the sole parameter for distance measurement. It is the authors' understanding that this notion of distance becomes more robust with increased depth in a tree. Therefore, only the trees reaching the maximum depth limitation are qualified for devolution. In Figure 3, it is demonstrated that with sufficient tree size, as indicated by $realmaxdepth$ (maximum depth allowed) in the GP program, the impact of devolution can be significantly reduced. In addition, it also shows that when compared to the standard resource control method, which reject children larger than the limited tree size, our devolution mechanism enables the AGS operator and achieves much better fitness improvement.

Having located the tail branches to remove, the actual

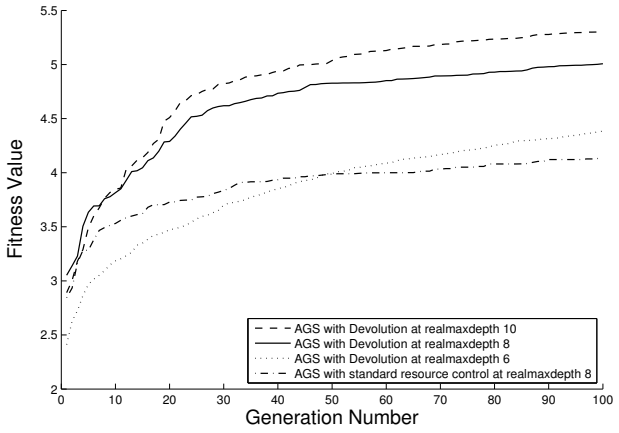

Fig. 3: Fitness records applying devolution at different $realmaxdepth$ and with standard resource control averaged over 30 runs in each case using 70% WBCD data for training.

devolution is executed by replacing the tail branches with one of its input terminals. The selection of replacing terminals is random. The process is illustrated in Figure 4.

3. EXPERIMENTS AND GP PARAMETERS

To evaluate the actual performance of the proposed AGS operators, Wisconsin Breast Cancer Dataset (WBCD) [16] and

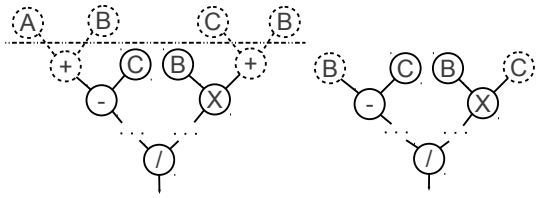

Fig. 4: Devolution: The left side tree is the individual before devolution. Nodes in dashed outlines are tail branch. The right side tree is the devolved form.

Pima Indians Diabetes Dataset (PIDD) [16] are used. Training and testing datasets are generated in three different ratios (train/test): 50%/50%, 70%/30% and 80%/20%. Using 50%/50% for WBCD as an example, in a total number of 569 samples which contain 212 malignant cases and 357 benign cases, 106 samples from malignant cases and 179 samples from benign cases are selected randomly as training samples. The rest are used as testing data.

The actual development of GP with AGS operators and devolution model is based on Silva's GPlab toolbox for MATLAB [17]. GP parameters have been tuned for establishing a fair ground for standard GP and AGS GP as well as for achieving a better overall performance. Some key parameters are listed in Table 1. The explanation of each term can be found in [17].

Table 1: Parameters Used In Experiments

Parameters	Values
generations	1000
population	25
operatorprobstype	fixed
initialfixedprobs	50% 50%
reproduction	10%
functions	plus, minus, times, divide
PR factor	0.5
inicmaxlevel	3
realmaxlevel/realmaxdepth	8
sampling	lexictour
elitism	halfelitism

There are three operator combinations presented and analyzed in this paper: standard mutation and standard crossover, AGS mutation and AGS crossover, as well as standard mutation and AGS crossover. Other combinations will be investigated in future. For each operator combination, 30 GP runs are executed where population status including tree information, population fitness and population complexity are recorded at each generation. When testing the classification accuracy, linear discriminant analysis (LDA) is used to classify the testing samples using the same training data as in GP runs.

4. RESULTS AND ANALYSIS

In Figure 5, fitness improvement over 1000 generations using different operator combinations is shown. The fitness values are averaged "best so far"fitness values (best fitness value in all previous generations in current run) over 30 runs. When using AGS mutation and AGS crossover, the fitness value sees a sharp improvement in the first 30 generations. However, the improvement of fitness slows down significantly after 30 generations. An early convergence is reached before 100 generations. At the same time, while having a more modest fitness improvement at first 30 generations, GP with standard mutation and crossover maintains a healthy increment rate. The rate becomes slower gradually but shows no obvious sign of convergence even at 1000th generations. The different results between purely AGS GP and purely standard GP indicate that the AGS operators are quick to find local optimum with initial building block, though, the lack of ability to maintain the population diversity leads to a premature conclusion in the solution searching process.

In order to utilize the strength of AGS operators and

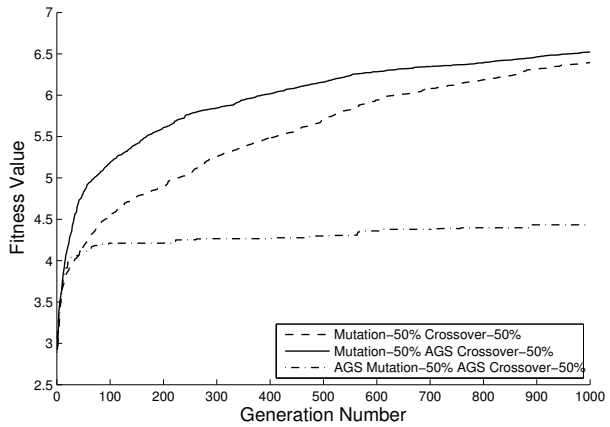

Fig. 5: Fitness records from each operator combination averaged over 30 runs in each case using 70% WBCD data for training.

complement their weaknesses, AGS mutation is replaced by standard mutation to achieve better searching distance and to maintain the diversity in later generations. The new combination will be named as AGS-GP in the rest of this paper. According to the results in Figure 5, the AGS-GP delivered a quick burst of fitness improvement in the first 100 generations. The rate of improvement is much faster than either of the two previous operator combinations at any point. Between 100 and 300 generations, it is clear that the fitness achieved by the AGS-GP is significantly higher than the standard mutation and crossover combination (standard GP). After 300 generations, the evolution of fitness slows down, yet it remains on a similar level as standard GP. At 1000th generation, AGS-GP still maintains a superior fitness over the standard GP while both carry on improving at a much slower speed.

As the fitness improvement is only part of the story, the actual classification performance need to be investigated to verify if the performance of GP generated features coincide with their behavior in the Fisher's criterion evaluation. In this experiment, we extract the best so far solution, which is the best solution found from all previous generations in each run, and use LDA classifier along with the testing data to obtain the classification performance at each generation. The classification accuracy is averaged over 30 runs and plotted in Figure 6. It can be seen that the performance plot echoes what has been seen in the fitness plot. The only difference is that, after 500 generations, even though fitness values are still improving, the actual classification accuracy from all operator combinations sees no obvious improvement.

Experiments with the same setup are repeated for different

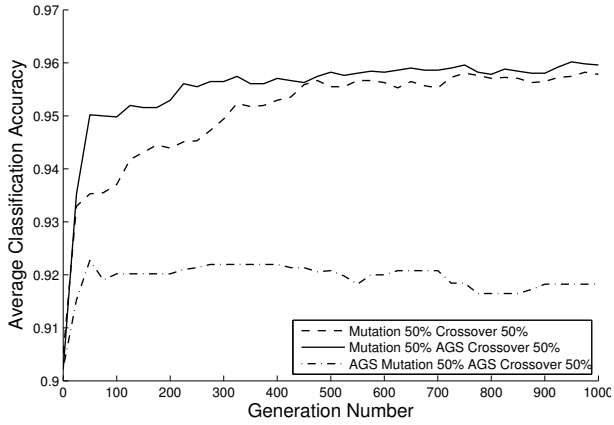

Fig. 6: Classification results from each operator combination averaged over 30 runs in each case using 30% WBCD data for testing.

Table 2: Breast Cancer Classification Test Results

Classifier	Data Ratio	Average	Best	Std
AGS-GP	50% / 50%	97.0%	98.6%	0.4%
Standard GP	50% / 50%	96.7%	97.9%	1.2%
AGS-GP	70% / 30%	95.9%	98.8%	1.4%
Standard GP	70% / 30%	95.7%	98.2%	1.9%
AGS-GP	80% / 20%	96.5%	99.2%	0.8%
Standard GP	80% / 20%	95.8%	98.3%	1.8%

data subsets. The final classification results are collected and presented in Table 2 and Table 3. It is shown that, in all cases, AGS-GP outperforms the standard GP. It is worth mentioning that if less training time is given, AGS-GP would have a bigger advantage as demonstrated in Figure 6.

Table 3: Diabetes Classification Test Results

Classifier	Data Ratio	Average	Best	Std
AGS-GP	50% / 50%	75.2%	78.7%	1.3%
Standard GP	50% / 50%	74.6%	76.8%	1.4%
AGS-GP	70% / 30%	75.7%	79.1%	1.4%
Standard GP	70% / 30%	75.1%	77.4%	1.0%
AGS-GP	80% / 20%	75.5%	79.1%	2.2%
Standard GP	80% / 20%	73.9%	76.4%	2.7%

5. CONCLUSION

New AGS genetic operators are proposed for the specific task of using GP as feature generator for diabetes and breast cancer classification. With the added devolution mechanism, the growth of GP using AGS operators are successfully controlled. The interpretation of experimental results is that AGS operators are good for searching optimum in a limited space. However, they lack the ability to explore more globally. This weakness can be overcome with the addition of traditional mutation. The end results show that the the propose AGS-GP classifier is superior to the stand GP classifier in all aspects and provides a potentially improved solution for many classification problems. Extended research will be conducted to validate the propose method with more datasets and comparison with more existing classifiers.

6. REFERENCES

[1] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, Massachusetts: The MIT Press, 1992.

[2] P. G. Espejo, S. Ventura, and F. Herrera, "A Survey on the Application of Genetic Programming to Classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 2, pp. 121–144, 2010.

[3] L. Zhang, L. B. Jack, and A. K. Nandi, "Extending genetic programming for multi-class classification by combining k-nearest neighbor," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2005, pp. 349–352.

[4] H. Guo and A. K. Nandi, "Breast cancer diagnosis using genetic programming generated feature," in *Machine Learning for Signal Processing, IEEE Workshop on (MLSP)*, 2006, pp. 215–220.

[5] H. Guo, Q. Zhang, and A. K. Nandi, "Feature Generation Using Genetic Programming Based on Fisher Criterion," in *15th European Signal Processing Conference (EUSIPCO)*, 2007, pp. 1867–1871.

[6] Z. Zhu, M. W. Aslam, and A. K. Nandi, "Augmented Genetic Programming for automatic digital modulation classification," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2010, pp. 391–396.

[7] M. W. Aslam, Z. Zhu, and A. K. Nandi, "Automatic Modulation Classification Using Combination of Genetic Programming and KNN," *IEEE Transactions on Wireless Communications*, vol. 11, no. 8, pp. 2742–2750, 2012.

[8] S. Luke, "Two fast tree-creation algorithms for genetic programming," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 274–283, 2000.

[9] P. Day and A. K. Nandi, "Binary string fitness characterization and comparative partner selection in genetic programming," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 724–735, 2008.

[10] U. Bhowan, M. Johnston, and M. Zhang, "Developing new fitness functions in genetic programming for classification with unbalanced data." *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics*, vol. 42, no. 2, pp. 406–421, 2012.

[11] F. D. Francone, M. Conrads, W. Banzhaf, and P. Nordin, "Homologous Crossover in Genetic Programming," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 1994, pp. 1021–1026.

[12] T. Jones, "Evolutionary algorithms, fitness landscapes and search," Ph.D. dissertation, 1995.

[13] A. Moraglio, K. Krawiec, and C. G. Johnson, "Geometric Semantic Genetic Programming," pp. 21–31, 2012.

[14] M. Zhang, P. Wong, and D. Qian, "Online program simplification in genetic programming," in *6th International Conference on Simulated Evolution and Learning*, 2006, pp. 592–600.

[15] H. Guo, L. B. Jack, and A. K. Nandi, "Feature generation using genetic programming with application to fault classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 1, pp. 89–99, 2005.

[16] A. Frank and A. Asuncion, "Pima Indians Diabetes Data Set," *http://archive.ics.uci.edu/ml*, 2010.

[17] S. Silva, "GPLAB A Genetic Programming Toolbox for MATLAB." [Online]. Available: http://gplab.sourceforge.net/