Reuse of Program Trees in Genetic Programming with a New Fitness Function in High-dimensional Unbalanced Classification

Wenbin Pei[1], Bing Xue[1], Lin Shang[2] and Mengjie Zhang[1]
1. School of Engineering and Computer Science, Victoria University of Wellington, New Zealand
2. State Key Laboratory for Novel Software Technology, Nanjing University, China
{Wenbin.Pei,bing.xue,mengjie.zhang}@ecs.vuw.ac.nz,shanglin@nju.edu.cn

ABSTRACT

Genetic programming (GP) may also evolve biased classifiers when having the class imbalance issue. Class imbalance is a difficult but important issue, and high-dimensionality brings difficulty when addressing the class imbalance issue. This paper focuses on addressing the performance bias of GP in classification with high-dimensional unbalanced data, with the goal of increasing the accuracies of the majority class and the minority class, as well as saving the training time. In this paper, a new fitness function is developed to address the class imbalanced issue, and moreover, a strategy is proposed to reuse previous good GP trees when using multiple GP processes to build a multi-classifier system. Experimental results show the better performance of the proposed method.

CCS CONCEPTS

• **Computing methodologies** → **Genetic programming**;

KEYWORDS

Genetic Programming, Class Imbalance, High-dimensionality

ACM Reference Format:
Wenbin Pei[1], Bing Xue[1], Lin Shang[2] and Mengjie Zhang[1]. 2019. Reuse of Program Trees in Genetic Programming with a New Fitness Function in High-dimensional Unbalanced Classification. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3321958

1 INTRODUCTION

Class imbalance is an important but common issue in some domains, such as fraud detection, medical diagnosis, financial analysis of loan policy or bankruptcy. Like other classification algorithms, GP may also suffer from the performance bias caused by class imbalance issue. A main reason is that many GP methods often employ the overall classification accuracy or error rate as the fitness function. This fitness function considers all training instances being equally important, having the same misclassification cost. To address the class imbalance issue in GP, the fitness function is used to adjust cost [1] or incorporate cost information [4]. In [2, 3], multi-objective GP (MOGP) is developed to solve the class imbalance issue. However, MOGP is time-consuming to obtain the complete Pareto front, and for existing MOGP methods in this topic, the diversity measure is also time-consuming. Recently, there are the growing number of datasets, where the number of features is far more than the number of instances. High-dimensionality causes challenges to solve the class imbalance issue, and vice versa.

The overall goal of this paper is to improve the performance of GP-based methods in classification with high-dimensional unbalanced data, in terms of enhancing the accuracy, as well as saving training time. This paper proposes a new fitness function to solve the class imbalance issue and attempts to reuse some good GP trees when evolving a multi-classifier system.

2 THE PROPOSED METHOD

2.1 The New Fitness Function

Using the fitness function for cost adjustment is one of the most important methods in GP to address the performance bias caused by the class imbalanced issue [1]. The new fitness function is:

$$Corr\_Min = \sqrt{\frac{\sum_{c=1}^{K} N_c(\mu_c - \overline{\mu})^2}{\sum_{c=1}^{K}\sum_{i=1}^{N_c}(P_{ci} - \overline{\mu})^2}} + \frac{\sum_{i \in Min} I(P_i, t)}{|Min|} \quad (1)$$

where K is the number of classes, N_c means the number of instances in the class c, $\mu_c = \frac{\sum_{i=1}^{N_c} P_{ci}}{N_c}$, $\overline{\mu} = \frac{\sum_{c=1}^{K} N_c \mu_c}{\sum_{c=1}^{K} N_c}$, P_{ci} represents the output of a genetic program when evaluated on an instance in a class c, μ_c indicates the mean of program outputs for a class c, and $\overline{\mu}$ represents the mean of μ_c for both the minority and majority classes; $I(q,t) = \begin{cases} 1, & q > t \text{ and } q \geq 0 \\ 0, & otherwise \end{cases}$, q is the program output when evaluated an instance from the minority class, t is a maximum output of a program for instances in the majority class, and $|Min|$ indicates a number of the instances in the minority class.

2.2 Overall Design

All features are randomly divided into five feature sets ($F1$, $F2$, $F3$, $F4$ and $F5$) with a roughly same size. The first GP process (GP1) is employed to evolve classifiers. $F1$ is fed to GP1 as terminals, using Ramped half-and-half for initialization and using the new proposed fitness function for evaluation. After the first evolutionary process, some good GP trees are discovered, which can be reused for the next GP process in initialization to further enhance the effectiveness and efficiency. Those selected features ($F1^*$) by these good trees, combined with $F2$ are fed to GP2. The similar processes continue until the fifth GP process (GP5) finishes to evolve the fifth classifier.

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3321958

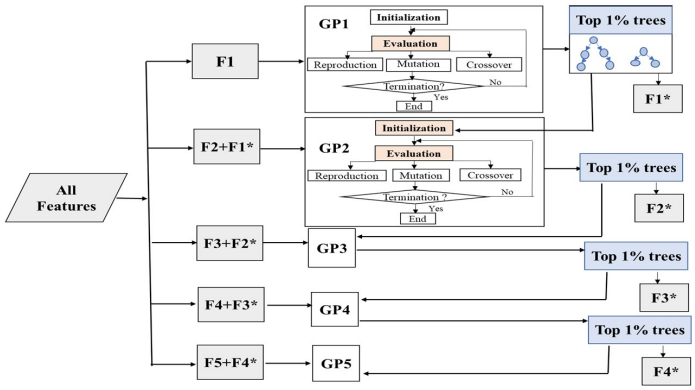

Figure 1: Overall Design

In the test process, the best individual from each GP process is chosen, using soft voting for making a final decision.

3 EXPERIMENT DESIGN

Table 1 describes the details of five high-dimensional unbalanced datasets having different dimensions and imbalance levels. In the experiment, the datasets are divided into the training set (70%) and test set (30%), based on the stratified sampling.

Table 1: Dataset Description

Dataset	#Features	#Instances	Majority (%)	Minority (%)	Imbalance level
Armstrong-2002-v1	1,081	72	66	34	2:1
Leukemia	7,129	72	65	35	2:1
DLBCL	5,469	77	75	25	3:1
Yeoh-2002-v1	2,526	248	82.66	17.34	5:1
Lung	12,600	156	88.99	11.01	8:1

Notes: For dataset Lung, label 1 as the majority class and label 2 as the minority class.
http://www.gems-system.org; https://schlieplab.org/Static/Supplements/CompCancer/datasets.htm

Table 2: Parameter Settings

Parameters	Standard GP	Each GP of new method
Population size	1024	256
Generations	50	40
Initial population	Ramped half-and-half	Ramped half-and-half
Maximum tree depth	10	10
Mutation rate	0.2	0.2
Crossover rate	0.8	0.8
Elitism	1	1
Selection method	Tournament (size=6)	Tournament (size=6)
Function set	+, −, ×, ÷, IF	+, −, ×, ÷, IF
Terminal set	Features of a dataset a random constant	Features in the F_q and F^*_{q-1} a random constant

Table 2 describes the parameter settings. The proposed method with the new fitness function (*GP_New*) is compared with GP with the fitness function *Acc* (overall classification accuracy, GP_{Acc}), *Ave* (balanced accuracy, GP_{Ave}) and *Corr_Min* (GP_{Corr_Min}). The population size of GP_{Acc}, GP_{Ave}, and GP_{Corr_Min} is 1024 for 50 generations. For each GP process in *GP_New*, the population size is 256 for 40 generations so that the total number of evaluations (i.e. 256*40*5) being similar to that of GP_{Corr_Min} (i.e. 1024*50).

4 RESULTS

Each methods have been conducted 30 independent runs with 30 different random seeds and the averaged results are shown in Table 3. Wilcoxon statistical significance test is conducted, with the

Table 3: Results on the Test Sets

		Balanced Accuracy (%)			Training Time (s)
Dataset	Method	Best	Mean±Std	ST	Mean
Armstrong	GP_{Acc}	100	87.63± 8.26	+	116.17
	GP_{Ave}	100	88.41 ± 6.09	+	114.86
	GP_{Corr_Min}	100	91.62 ± 6.29	+	143.6
	GP_New	100	**98.6 ± 3.56**		42.13
Leukemia	GP_{Acc}	96.43	78.07 ± 10.65	+	976.92
	GP_{Ave}	93.75	80.36 ± 7.3	+	975.7
	GP_{Corr_Min}	100	85.6 ± 7.05	=	793.88
	GP_New	90.18	**88.01 ± 4.47**		157.34
DLBCL	GP_{Acc}	97.22	74.44 ± 13.48	+	733.43
	GP_{Ave}	91.67	68.89 ± 10.7	+	740.03
	GP_{Corr_Min}	100	74.07 ± 14.15	+	670.87
	GP_New	100	**86.2 ± 7.84**		153.92
Yeoh	GP_{Acc}	86.04	67.98 ± 10.15	+	779.75
	GP_{Ave}	99.19	82.14 ± 11.42	+	773.26
	GP_{Corr_Min}	100	**98.45 ± 3.02**	=	703.08
	GP_New	100	97.44 ± 2.77		152.78
Lung	GP_{Acc}	100	71.44 ± 16.1	+	3044.48
	GP_{Ave}	97.62	74.95 ± 13.12	+	3048.62
	GP_{Corr_Min}	100	84.71 ± 9.79	+	2639.07
	GP_New	100	**95.52 ± 6.2**		413.20

Std means standard deviation; ST means significance test results.

significance level of 0.05, where "+", "=" and "−" are used to show that the new method is significantly better, similar, or significantly worse than other method.

According to Table 3, *GP_New* can achieve the significantly better performance than GP_{Acc} and GP_{Ave} on these high-dimensional unbalanced datasets. Moreover, *GP_New* can achieve the better performance than GP_{Corr_Min} on four datasets. In addition to increase averaged accuracy, the standard deviation of the balanced accuracy is decreased on all datasets, which may show that the stability of the *GP_New* is better than GP_{Acc}, GP_{Ave}, and GP_{Corr_Min}. The best balanced accuracy achieved by the new method is often at least better than other methods. More importantly, the new method often achieves the narrowest gap between the best balanced accuracy and the averaged balanced accuracy. The average training time of *GP_New* is much faster than other methods on all datasets.

5 CONCLUSIONS AND FUTURE WORK

In this paper, a new fitness function is proposed to solve the class imbalance issue in classification with high-dimensional unbalanced data. Based on the new fitness function, this paper proposed a method to evolve a multi-classifier system by employing multiple GP processes for classification with high-dimensional unbalanced data. This paper suggests not only reusing good features previously-selected but also good trees in the initialization for the later GP process. Experimental results show that the proposed method can reduce the training time, and increase the accuracy in most cases.

REFERENCES

[1] Urvesh Bhowan, Mark Johnston, and Mengjie Zhang. 2012. Developing new fitness functions in genetic programming for classification with unbalanced data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 42, 2 (2012), 406–421.

[2] Urvesh Bhowan, Mark Johnston, Mengjie Zhang, and Xin Yao. 2013. Evolving diverse ensembles using genetic programming for classification with unbalanced data. *IEEE Transactions on Evolutionary Computation* 17, 3 (2013), 368–386.

[3] Urvesh Bhowan, Mark Johnston, Mengjie Zhang, and Xin Yao. 2014. Reusing Genetic Programming for Ensemble Selection in Classification of Unbalanced Data. *IEEE Transaction on Evolutionary Computation* 18, 6 (2014), 893–908.

[4] Jin Li, Xiaoli Li, and Xin Yao. 2005. Cost-sensitive classification with genetic programming. In *The 2005 IEEE Congress on Evolutionary Computation*, Vol. 3. IEEE, 2114–2121.