Segment-Based Genetic Programming

Nailah Al-Madi and Simone A. Ludwig
Department of Computer Science
North Dakota State University
Fargo, ND 58102, USA
{nailah.almadi,simone.ludwig}@ndsu.edu

ABSTRACT

Genetic Programming (GP) is one of the successful evolutionary computation techniques applied to solve classification problems, by searching for the best classification model applying the fitness evaluation. The fitness evaluation process greatly impacts the overall execution time of GP and is therefore the focus of this research study. This paper proposes a segment-based GP (SegGP) technique that reduces the execution time of GP by partitioning the dataset into segments, and using the segments in the fitness evaluation process. Experiments were done using four datasets and the results show that SegGP can obtain higher or similar accuracy results in shorter execution time compared to standard GP.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization, Global optimization; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Optimization, Algorithms, Performance, Experimentation

Keywords

Evolutionary computation, genetic programming, fitness evaluation, data classification

1. INTRODUCTION

Evolutionary computation is a problem solving technique that mimics the mechanism of natural operations and survival of the fittest. Genetic Programming (GP) [1] is one of the evolutionary computation approaches that proved its efficiency in optimization and problem solving such as classification. GP solves the problems by generating programs that are constructed from mathematical and logical functions and terminals (variables and constants). The solution quality provided by each program is evaluated using a fitness function. The main drawback that prevents the use of GP in real time problem solving is its long execution time, which mainly depends on the number of generations, the population size, and the fitness evaluation. The fitness evaluation is the most computational intense process in GP [2].

GECCO'13 Companion, July 6–10, 2013, Amsterdam, The Netherlands.
ACM 978-1-4503-1964-5/13/07.

When GP is applied to solve a classification task, each program presents a classifier model that distinguishes classes in order to apply this model to new data records. The fitness function for a classification task is the classifier accuracy (number of correctly classified records). Therefore, the execution time is dependent on the dataset size.

This paper proposes a segment-based GP (SegGP) that accelerates the fitness evaluation process without affecting the classification accuracy. SegGP is based on the idea of decreasing the training dataset size, but at the same time covering the whole training dataset by partitioning the training dataset into segments.

2. PROPOSED APPROACH

The fitness evaluation step when solving a classification problem using GP, refers to the program execution and the calculation of its classification accuracy. Therefore, the number of fitness evaluations (FE) of an independent GP run is related to population size (P), dataset size (D), and number of generations (G). A single program fitness calculation is equal to (D) since the program is executed for every record in the dataset. Thus, the total number of fitness evaluations for a GP run is $FE = D \times P \times G$.

The proposed SegGP approach accelerates the fitness evaluation time during the GP run by reducing the dataset size (D). It is based on creating segments of the training dataset, which is done using the resample supervised instance filtering of the Weka Data mining software [3]. Therefore, whenever GP performs a fitness evaluation for a program it randomly chooses one of these segments and executes the program to obtain the fitness. Each program may choose different segments. The training dataset is partitioned into ten segments. Size of the segments (S) is a percentage of the training dataset, where $S < D$. Using this technique, the number of fitness evaluations is reduced to $FE = S \times P \times G$.

The whole training dataset is covered by the population. In addition, if a program survives for multiple generations, it is likely to cover the whole dataset by using different segments in each generation, and thus, the program is trained on the whole dataset. There is a high probability that this occurs with segments of large percentage of the whole dataset. It is important to note that the entire testing dataset is used without any partitioning.

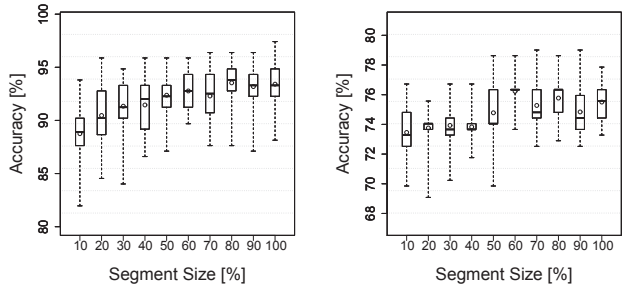
3. EXPERIMENTS AND RESULTS

For the evaluation of SegGP, experiments were performed using the JGAP [4] on four datasets [5]. The description of these datasets are shown in Table 1 (reduced number of

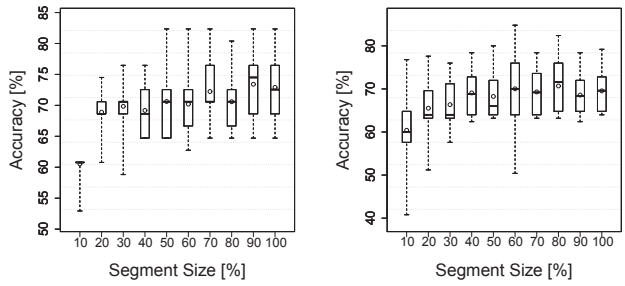
Table 1: Datasets

	Dataset	Classes	Features	Records
D1	Breast	2	31 (11)	568
D2	Diabetes	2	8 (4)	768
D3	Lymph	4	19 (10)	148
D4	Dermatology	6	33 (19)	366

features are shown in the brackets). The GP settings were as follows: Population size = 500, Number of generations = 1000, Crossover probability = 0.5, Mutation probability = 0.1. Moreover, 23 functions and 14 variables/constants are used. The experiments are applied using 66% of the dataset for training, and the remaining 34% for testing. In order to compare our proposed SegGP with standard GP 50 independent runs were performed.



(a) D1 Accuracy (b) D2 Accuracy



(c) D3 Accuracy (d) D4 Accuracy

Figure 1: Accuracy results

To find the best segment size to use, we ran SegGP with different segments sizes, starting with a segment size of only 10% of the training dataset ("Seg10"), up to using the whole dataset (100%, "Seg100") in steps of 10% increments (Seg20, Seg30, Seg40 ... Seg90, Seg100). The accuracy results for the four datasets are shown using the box plots in Figure 1. It can be concluded that Seg90 has higher accuracy values than Seg100 for D3. Seg80 has three accuracy values higher than Seg100 for D1, D2, and D4. Moreover, Seg60 has two values higher than Seg100 for D2 and D4. However, Seg70 has 3 accuracy values close to Seg100 for D2, D3 and D4. The execution time results averaged over 50 runs are shown in Figure 2, where all the results confirm that a smaller segment size runs faster than a larger segment size.

4. CONCLUSION

A segment-based genetic programming (SegGP) approach is proposed, that accelerates the genetic programming fitness

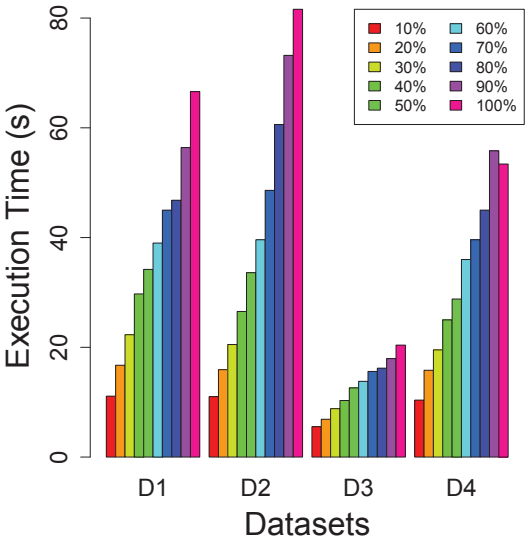


Figure 2: Execution time

evaluation process. SegGP is based on the idea of reducing the training dataset size but at the same time covering the whole training dataset. SegGP aims to obtain the same classification accuracy with a shorter execution time than standard GP. A comparison of standard GP and SegGP was conducted using ten segments with different percentage sizes of the whole dataset, and applied on four datasets. In summary, SegGP obtains the same or in some cases higher accuracy than standard GP, using Seg80 or Seg60, and a speedup up to 24.14% or 39.54%, respectively is achieved.

Future work will involve testing SegGP on larger datasets and also involve different optimization tasks such as regression and clustering.

References

[1] A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. SpringerVerlag, 2003.

[2] R. Poli, W. B. Langdon, and N. F. McPhee. *A field guide to genetic programming*. Published via `http://lulu.com` and freely available at `http://www.gp-field-guide.org.uk`, 2008.

[3] I. Witten, E. Frank, and M. Hall. *Data Mining: Practical Machine Learning Tools And Techniques, 3rd Edition*. Morgan Kaufmann, 2011.

[4] K. Meffert et al., JGAP - java genetic algorithms and genetic programming package [online]. available: http://jgap.sf.net, January 2012.

[5] A. Frank and A. Asuncion. UCI machine learning repository, available: http://archive.ics.uci.edu/ml, 2010.