A novel Genetic Programming technique benchmarked on a real life data set

Antonio Curado,[1] Joris Bertens,[2] Manuel Demetriades,[3] Morten Dahl[4]

[1]M20180032@novaims.unl.pt
[2]M20180423@novaims.unl.pt
[3]M20180425@novaims.unl.pt
[4]M20180047@novaims.unl.pt

The following work introduced new ways of calculating fitness, selecting individuals and a probabilistic approach of selecting function and terminal. The research found, that introducing probabilities to function and terminal selection yield in a better fitness. All this has been benchmarked on a Portuguese socio-economic municipalities dataset as the goal of predicting unemployment rates for each municipality.

Introduction

Genetic programming is a field in machine learning that aims to utilise nature inspired concepts of Biology to build computer programs. It draws inspiration for concepts such as the fitness of an individual as an allusion to how close to a given target it is. Selection of the best individuals, crossover between individuals and mutations that change the ”genetic structure” of an individual are other components imitated from the natural evolution process.
The main focus of this work is on how this fitness is evaluated and how the best individuals are selected. It furthermore places increased significance on how each individual experiences each mutation. To assess the quality of results, a real dataset consisting of Portuguese municipality data was collected with the aim of predicting their unemployment. Occupational opportunities are limited and an issue in Portugal. Hence, it should be a priority for local governments to create more job opportunities and economic growth in their municipalities.

Proposed techniques

Selection Methods

Selection Methods are a crucial operator in all evolutionary algorithms and hence contribute to the success or failure of applying certain variations of the algorithm. The three main types of it are tournament selection, fitness-based selection (roulette-wheel) and rank-based selection. As in the baseline only the tournament selection was implemented, the other two were tried out in their standard appearance in order to have a more complete picture. In general the rank- and fitness-based selection have the disadvantage of having to evaluate the entire population. This aspect could be neglected in our case as it was computationally relatively cheap.

Special fitness

As a way to penalize long programs it was introduced the concept of a parsimony coefficient that calculates a new fitness taking in consideration the size of the program

$$\mathbf{penalty} = parsimony\ coefficient * size(program) * sign \tag{1}$$

The current gplearn framework approach (2) consists of subtracting this new penalty from the current fitness

$$fitness = raw\ fitness - \mathbf{penalty} \tag{2}$$

Although, this calculation is not taking into consideration the order of magnitude of fitness. Meaning that when fitness is close to 0 it will be highly penalized and when fitness is calculated in higher orders of magnitude, this penalization will hardly be taken into consideration. We

therefore propose (3), a similar, but improved way to calculate this new fitness that takes into consideration the the order of magnitude that is being dealt with

$$fitness = raw\ fitness - (\mathbf{penalty} * raw\ fitness) \tag{3}$$

Operator probabilities

On standard GP, when performing mutation and generating new individuals, the likelihood of picking an operator (function or terminal value) always remains the same regardless of the impact on the population or the individual.
As shown by (*Niehaus Jens, 2001*) adaptive mutation and crossover likelihoods can highly affect end results in genetic programming. The proposed technique aims to leverage possible variations of these otherwise constant operator probabilities to increase search space exploration, increase fitness and decrease bloat. Three different ways to adapt the likelihood were developed, based on:

Its usage - **Naive**
The increase in fitness - **Phenotype**
The increase of population diversity - **Genotype**

Naive operator likelihood update

The first and most simple approach was to decrease the likelihood of using an operator when the operator is used

mul	*div*	*sum*
0.33	0.33	0.33
	div is used	
0.36	0.30	0.36

This approach increases the chance of picking an operator the less it is used. The expectation is that it increases early search space exploration and leads to a faster convergence of results

Phenotypic operator likelihood update

The phenotypic update uses the change in fitness from parent to child as a way to update probabilities. An operator that increases fitness will see their likelihood increased or decreased otherwise.

mul	*div*	*sum*
0.33	0.33	0.33
	div is used and decreased fitness	
0.36	0.30	0.36

This approach increases the chance of picking operators that increase fitness. We expect this to increase the probability of selecting meaningful operators, lead to overall better results as well as providing an improved solution to dealing with a larger operator space.

Genotypic operator likelihood update

The genotypic update uses the diversity of operators within a population(the count of all operators) to update the likelihoods for the next one. If a population mostly has one particular operator, its probability will be heavily decreased for the next generation.

mul	*div*	*sum*
0.33	0.33	0.33
	Diversity within the population is calculated	
1	4	1
	Likelihood updated	
0.16	0.66	0.16

This approach is similar to the naive one but takes into consideration the whole population to calculate its diversity. The same assumptions are held.

All above mentioned examples only shown function as for simplicity of demonstration. As previously mentioned all the techniques bot use functions and terminals. In the below sections a comprehensive analysis of results was conducted in order to evaluate the practical truthfulness of the above mentioned techniques.

Experimental Methodology

All experiments were conducted using the unemployment dataset which contained 12 continuous variables and 308 observations

Variable Group	Description
Composition of Unemployment	- unemployment based on age groups, education per sex and job seek per sex
Average Wages	- average earnings/wages in each municipality per education, per position and per industry sector
Municipal Finances & Economy	- current revenue, current expenditure and capital expenditure per capita - enterprise density and labor productivity
Population Composition	- share of foreigners living in a municipality, - age composition, young-dependency ratio, ageing index, potential sustainability index - population density
Political Affiliation	- result of the 2011 parliament election

Table 1: Unemployment dataset variables

The data retrieved and utilized to compose this dataset was taken from pordata, the database of Contempory Portugal. All data was retrieved on a municipality level and joined on the unique municipality. The collected data is exclusively from 2011. As this was the year the census was conducted, data is particularly versatile and consists of numerous features. The data can be retrieved here. An exploratory data analysis has been conducted for which results have been summarized in a separated exploratory data analysis workbook.

The target to be predicted by the model was the actual unemployment for each municipality, also from 2011. Error is computed as the Root Mean Squared Error (RMSE) between the outputs of the model and the targets of the dataset. The genetic programming algorithm default parameters have been inpired by (*Gonalves Ivo, 2017*) with changes to the particular dataset such as decreasing the validation dataset size. As for other tested algorithms no parameter optimization has been performed as this was out of scope of the proposed project. All results presented are run on an average of 5 seeds.

Parameter	Value
Data partition	Training 80% - Unseen 20%
Runs	20
Population size	1000
Initialization	Ramped Half-and-Half, maximum depth 6
Function set	+, -, *, and / (protected)
Terminal set	Input variables, plus constants

Table 2: GP parameters table

Analysis of results

GP vs. State of art methodologies

Several Machine Learning Algorithms were implemented to reveal the performance of a standard GP and standard GSGP against these algorithms. There we decided to benchmark on algorithm types such as a Regression & Tree based algorithms such as a Decision tree, Tree-boosting, tree-bagging and a random forest. A regression was chosen because it makes sense to use it on this regression problem.

	Baseline GP	GSGP	Decision Tree	Adaptive Tree Boosting	Gradient Tree Boosting	Random Forest	Tree Bagging	Linear Regression
Mean	2.29	2.37	2.17	1.78	1.63	1.77	1.79	1.69
Standard Deviation	0.36	0.37	0.20	0.19	0.13	0.14	0.14	0.16
Best	1.83	1.93	1.91	1.51	1.46	1.57	1.61	1.47

Table 3: Analysis of results of different machine learning algorithms

What can be seen in the table above is that the standard GP and GSGP are heavily outperformed by their competitors. Not only is the standard deviation bigger in comparison to other algorithms, but also the mean of the mean squared error over five seeds does not get close to the other algorithms. The assumption made that a regression would be a good fit seems to hold.

Geometric Semantic GP

In order to find out how the GS-GP performs against the baseline GP, several operators of the GS-GP were tested. The parameter settings over this grid search were based on the parameter settings used in the research of (*Gonalves Ivo, 2017*) and (*Castelli Mauro, 2015*). The operators that have been tuned are the probability over, the neighborhood size as well as the stopping criteria EDV & TIE. A few highlights of this benchmark are discussed and the remaining results can be found in the referenced code.

The first result show that a GSGP mutation of 5% overall scores the best, having a relatively low mean as well as a good best fitness. This might indicate that for this specific problem a mutation probability of 5% could yield better results.

	Baseline GP	GSGP Mutation 1%	GSGP Mutation 2%	GSGP Mutation 5%	GSGP Mutation 10%
Mean	2.29	2.34	2.14	2.20	2.50
Standard Deviation	0.36	0.32	0.27	0.16	1.04
Best	1.83	2.04	1.83	1.93	1.87

Table 4: Analysis of results of different mutation probabilities

Special Fitness & Different Selections

Using results from the previously mentioned GP baseline, we tested the newly introduced techniques for fitness calculation and individuals selection as shown in Table 6

	Baseline GP	Rank	Roulette	Special fitness
Mean	2.29	7.33	2.39	2.55
Standard Deviation	0.36	3.98	0.47	0.68
Best	1.83	3.26	2.00	1.80

Table 5: Analysis of results for special fitness and selection methods

Tournament selection, the one used in the baseline, outperforms rank and roulette. It is the default method in GP and it is so for a reason as results show.
As for special fitness, even though the highest gains are expected to be of fitness with unusual orders of magnitude, in one of the seeds it resulted as the best solution.

Probability operators

The above described probability operators were tested against the GP baseline in order to draw conclusions for further research. During our analysis it became clear that the enhancements made did not have the desired output of enhancing the fitness significantly. The pheno and the naive approach did not yield significant improvements and the geno approach only outperformed the baseline by 0.1 in terms of mean squared error. Nevertheless, some interesting findings were made which give several starting points for further research.

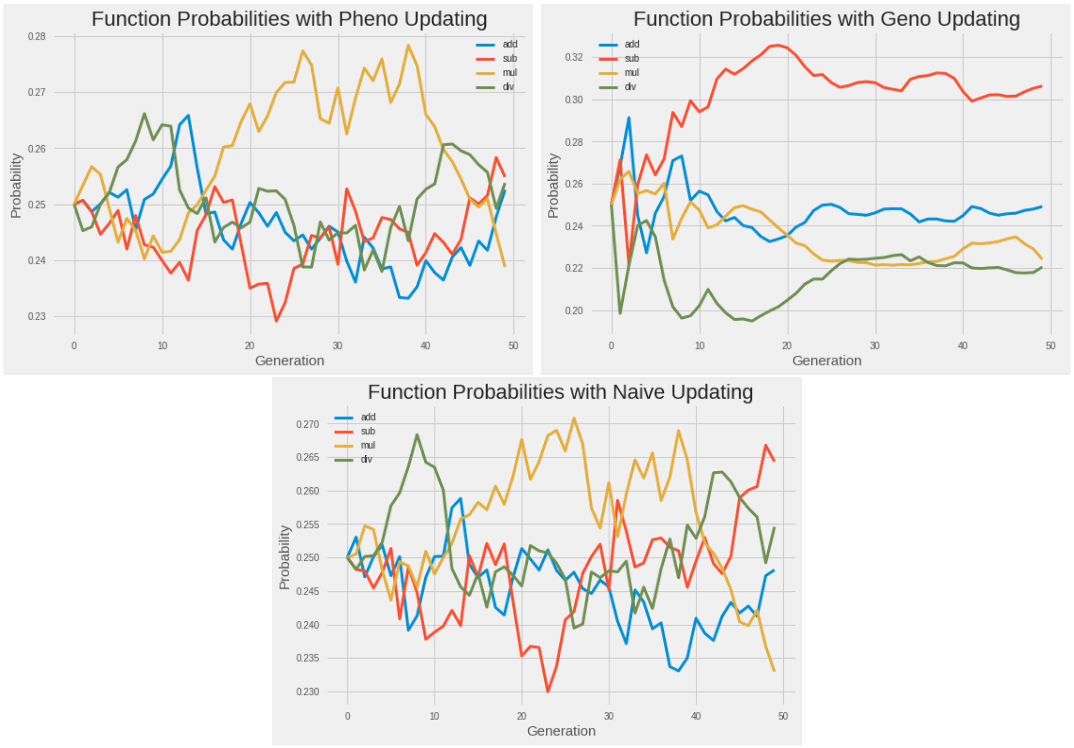

Figure 1: Evolution of function probabilities over generations

The a-priori notion was, that changing the probabilities will lead to a more diverse use of operators within the population, which will ultimately lead to an increase in fitness by more extensively exploiting the search space. The results of the geno updating method seem to give hints of supporting this notion for the usage of functions. As this method promotes the usage of the divide and multiply operator further, compared to the baseline, the final population actually got more diverse compared to the baseline. However, the usage of terminals does not show this phenomenon as strongly as the usage of functions. It got apparent, that even the geno operator goes with a general trend as the usage of all terminals are more or less the same. Another a-priori expectation, that of this method improving feature selection, seems to not hold.

The expected diversity in the population of trees did occur. The usage of all terminals is actually constant and only reveal minor deviations with a small outbreak of the geno approach, which yielded bigger deviations. However, when examining the subtract operator, the change in picking functions gets apparent. Through the naive and pheno approach the subtract probability exceeded all other probabilities by far so this operator was more likely to be chosen. In the final population the occurrence of the subtract operator was higher for these two approaches than for the other approaches.

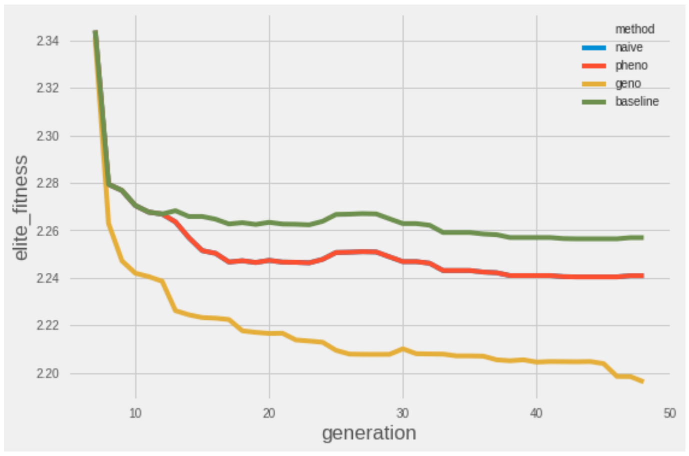

Figure 2: Evolution of fitness from generation 7 to 50

Figure 3: Evolution of function probabilities over generations

Another finding is, that the naive and the pheno approach yield the same results and show even the same results when comparing the averages. This can be explained by the low magnitude of fitness increases as both techniques have the same effect on probability, when there's no fitness increase.

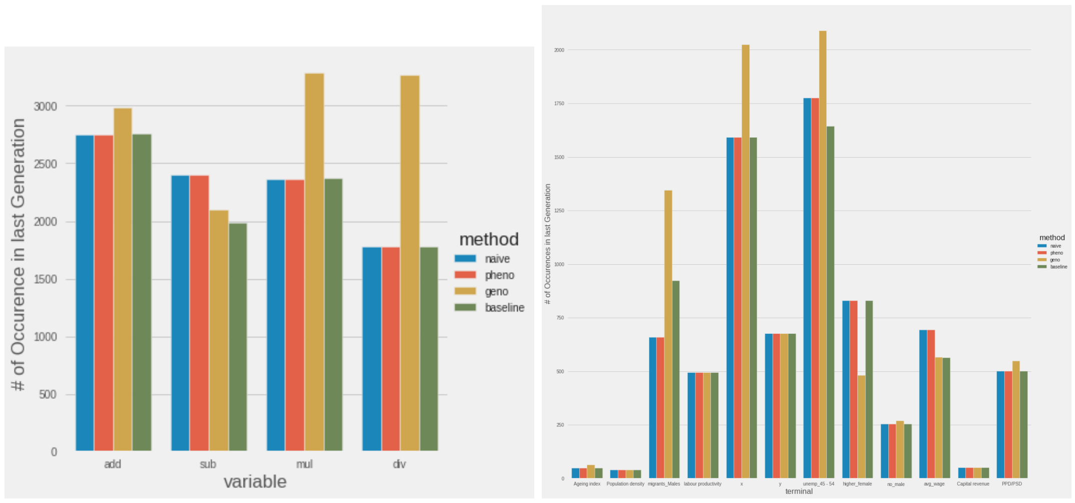

Figure 4: Final generation operator counts

Figure 5: Evolution of function probabilities over generations

All of these findings nevertheless give an intuition, that the usage of probabilities on functions and terminals might bring some value as another technique in the field of genetic programming given the tweaking of the update parameter and the introduction of other update methods are a novel field of further research.

	Baseline GP	Geno	Pheno	Naive
Mean	2.29	2.19	2.24	2.24
Standard Deviation	0.22	0.24	0.22	0.22
Best	2.62	2.62	2.62	2.62

Table 6: Analysis of fitness for probability operators

Conclusions

Genetic programming has been a field of research since 1988 (*Koza, 1990*) having research papers published every year. From this work we can confirm the reasoning behind tournament selection being the generally accepted default selection method and that when evaluating fitness, any penalty applied must take into fitness magnitude into consideration otherwise risking to lose the capability to generalize. Even though these are important findings that confirm previously conducted research, the core finding lies with the results of the probabilistic operators. Being a novel approach in the field of genetic programming, a lot of research could be conducted. Our initial and sometimes naive approaches lead to a marginal increase in fitness compared to the baseline. By tweaking update rates and introducing other update methods, these results could be enhanced and make a significant contribution to the field of genetic programming.

Further work

As shown this approach has shown some interesting results and as so, possible next steps are to:

- Test the special fitness with different orders of magnitude. In more extreme scenarios, special fitness is expected to show a statistically significant difference.
- Run further experiments with
  - A higher number of operators to verify that the proposed approach is well suited for higher dimensional search spaces
  - Different data sets, preferably with a larger amount of variables
  - Different update strategies for the probabilities of operators
- Use different initialization approaches, such as DEAP to set initial weights of the operators instead of attributing the same initial likelihoods

References and Notes

Castelli Mauro, 2015. Castelli Mauro, Vanneschi Leonardo, P. (2015). Controlling individuals growth in semantic genetic programming through elitist replacement.

Gonalves Ivo, 2017. Gonalves Ivo, Silva Sara, F. C. M. C. M. (2017). *Unsure When to Stop? Ask Your Semantic Neighbors*.

Koza, 1990. Koza, J. R. (1990). Non-linear genetic algorithms for solving problems. United States Patent 4935877. filed may 20, 1988, issued june 19, 1990, 4,935,877. Australian patent 611,350 issued september 21, 1991. Canadian patent 1,311,561 issued december 15, 1992.

Niehaus Jens, 2001. Niehaus Jens, B. W. (2001). Adaption of operator probabilities in genetic programming. In Miller, Tomassini, L. R. T. L., editor, *Genetic Programming 4th European Conference, EuroGP 2001*, pages 325–336. Springer.