SGP-DT: Towards Effective Symbolic Regression with a Semantic GP Approach Based on Dynamic Targets

Stefano Ruberto
Institute for Biomedical Informatics
Philadelphia, USA
stefano.ruberto@pennmedicine.upenn.edu

Valerio Terragni
Faculty of Informatics
Lugano, Switzerland
valerio.terragni@usi.ch

Jason H. Moore
Institute for Biomedical Informatics
Philadelphia, USA
jhmoore@upenn.edu

ABSTRACT

Semantic Genetic Programming (SGP) approaches demonstrated remarkable results in different domains. SGP-DT is one of the latest of such approaches. Notably, SGP-DT proposes a dynamic-target approach that combines multiple GP runs without relying on any form of crossover. On eight well-known datasets SGP-DT achieves small RMSE, on average 25% smaller than state-of-the-art approaches.

CCS CONCEPTS

• **Theory of computation → Bio-inspired optimization**; **Genetic programming**;

KEYWORDS

Genetic Programming, Semantic GP, Natural Selection, Symbolic Regression, Residuals, Linear Scaling, Crossover, Mutation

ACM Reference Format:
Stefano Ruberto, Valerio Terragni, and Jason H. Moore. 2020. SGP-DT: Towards Effective Symbolic Regression with a Semantic GP Approach Based on Dynamic Targets. In *Genetic and Evolutionary Computation Conference Companion (GECCO '20 Companion), July 8–12, 2020, Cancún, Mexico.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3377929.3397486

1 INTRODUCTION

Semantic Genetic Programming (SGP) uses richer feedback during the evolution that incorporates semantic awareness, which has the potential to improve the power of genetic programming [7, 8].

Symbolic Regression is characterized by a set of training cases, defined as m pairs of inputs $\hat{x}$ and desired output $\hat{y}$. Following most SGP approaches [8], we intend the semantics of an individual $\mathcal{I}$ as a vector $sem(\mathcal{I}) = \langle y_1, y_2, \cdots, y_m \rangle$ of responses to the m inputs of the training cases [4]. Let $sem(\hat{y}) = \langle \hat{y_1}, \hat{y_2}, \cdots, \hat{y_m} \rangle$ denote the semantic vector of the target (as defined in the training set), where $\hat{y_1}, \hat{y_2}, \cdots, \hat{y_m}$ are the desired outputs. SGP defines the *semantic space* [8] equipped with a distance between the semantic vectors of the individuals $sem(\mathcal{I})$ and the target $sem(\hat{y})$. The effectiveness of SGP depends on the availability of GP operators that can move in the semantic space towards the global optimum. Both classic and semantic-aware crossovers suffer from several drawbacks [4, 8].

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

GECCO '20 Companion, July 8–12, 2020, Cancún, Mexico

© 2020 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-7127-8/20/07.
https://doi.org/10.1145/3377929.3397486

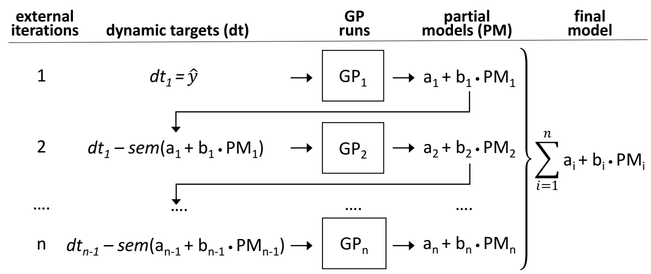

Figure 1: Overview of SGP-DT

Our recent paper proposes SGP-DT [6], a SGP technique that effectively navigates the semantic space without relying on any form of crossover. SGP-DT invokes multiple GP runs, each of which is guided by a (dynamic) target that focuses on a particular characteristic of the problem at hand. SGP-DT combines with linear scaling the models returned by each GP runs into a final model. In this paper, we summarize the technique and major accomplishments.

2 SGP-DT

Figure 1 gives an overview of our approach. SGP-DT runs a predefined number of GP algorithms $GP_1, GP_2, \cdots GP_n$. We call these runs *external* iterations. As opposed to the *internal* iterations (i.e., GP generations) that the GP algorithm performs at each run.

Each run has associated a *dynamic target* (dt) that changes at each external iteration. The dynamic target dictates the fitness of the individuals, defined as the variance of the difference between $sem(\mathcal{I})$ and the current target (i.e., $\sigma^2(sem(\mathcal{I}) - dt)$). Each GP run performs a fixed number of internal iterations and returns the fittest solution that we call *partial model* (PM). For each individual (and thus for each partial model) we compute the coefficients a and b of the *linear scaling* technique [2], which entails a bound on the worsening of the offspring at each internal and external iteration [6].

The dynamic target of the first external iteration is the desired output ($\hat{y}$) as specified by the training set. The dynamic target of the $i\text{-}th$ iteration is the difference between the previous target and the semantic of the partial model returned by the previous iteration (i.e, *residual error*): dt_{i-1} - $sem(a_{i-1} + b_{i-1} \cdot PM_{i-1})$. As such, SGP-DT leads to dynamic targets that change at each external iteration incorporating the semantic information. Each partial model focuses on a different characteristic of the problem that the fitness function recognized to be important (at that iteration). This makes the search more efficient because the evolution focuses on a single characteristic at a time leaving unaltered the already optimized ones (and thus preserving the already discovered functionalities).

Table 1: Datasets of regression problems.

name	# attributes	# instances	source	name	# attributes	# instances	source
airfoil	5	1,503	UCI	housing	14	506	UCI
concrete	8	1,030		tower	25	3,135	
enc	8	768		yacht	6	309	
enh	8	768		uball5d	5	6,024	[5]

SGP-DT obtains the final solution with a linear combination $\sum_{i=1}^{n} a_i + b_i \cdot PM_i$. Notably, SGP-DT does not rely on any form of crossover, neither semantic nor classic, and thus avoiding their intrinsic limitations. SGP-DT implicitly recombines different functionalities when it assembles the partial models into the final one.

3 EVALUATION RESULTS

We now summarize a series of experiments that we conducted to evaluate SGP-DT.

Datasets. We performed our experiments on eight well-known datasets of regression problems that have been used to evaluate most of the relate techniques. Table 1 shows their names, number of attributes, and number of instances.

Techniques under comparison. We compared SGP-DT with two techniques: LASSO [1] and ϵ-LEXICASE [3].

LASSO [1] is a regression analysis method that uses the least square regression to linearly combine solution components. More specifically, LASSO incorporates a regularization penalty into least-squares regression using an ℓ_1 norm of the model coefficients and uses a tuning parameter λ to specify the weight of this regularization [1].

ϵ-LEXICASE [3] is an evolutionary technique that adapts the *lexicase* selection operator for continuous domains. The idea behind ϵ-LEXICASE selection is to promote candidate solutions that perform well on unique subsets of samples in the training set, and thereby maintain and promote diverse building blocks of solutions [3].

For SGP-DT and ϵ-LEXICASE we set a population size of 1,000 and a budget of 1,000 generations. For SGP-DT, we divided the 1,000 generations in 20 external iterations ($N_{ext} = 20$), and thus the number of internal iterations of each GP run is 50. To cope with the stochastic nature of GP, we ran 50 trials for every technique on each dataset using 25% of the data for testing and 75% for training.

Root Mean Square Error (RMSE) comparison. Table 2 shows the median RMSE and the RMSE percentage decrease of SGP-DT with respect to LASSO and ϵ-LEXICASE[1]. A positive value means that SGP-DT has a lower (better) RMSE median.

SGP-DT achieves a smaller RMSE than LASSO for all the datasets, obtaining always statistical significance (p-value <0.05). The decrease of the RMSE medians ranges from 9.06% for *housing* to 88.67% for *yacht* (51.47% on average). SGP-DT has smaller RMSE medians than ϵ-LEXICASE for all datasets except *housing* (decrease -4.48%). This is the only comparison of SGP-DT and ϵ-LEXICASE without statistical significance. The decrease of the RMSE medians ranges from -4.48% for *housing* to 57.07% for *ench* (23.19% on average).

Computational effort. To compare the computational effort of SGP-DT and ϵ-LEXICASE we relied on the total number of evaluated nodes. Both SGP-DT and ϵ-LEXICASE operate on nodes, SGP-DT on tree-like data structures, while ϵ-LEXICASE on stack-based ones. Table 3 reports the median number of nodes (of the 50 trials) that the GP techniques evaluated to produce the final solution (all comparisons are statically significant). Comparing with ϵ-LEXICASE, SGP-DT reduces the amount of node evaluations by a factor between 4.01× and 9.26×, obtaining better RMSE values than ϵ-LEXICASE for seven out of eight datasets (see Table 2).

[1]calculated as $((M_T - M_D)/M_T) \cdot 100$, where M_D is the median RMSE of SGP-DT and M_T is the one of the competing technique

Table 2: Median RMSE of the 50 trials.

Data set	Median Root Mean Square Error (RMSE) SGP-DT	LASSO	ϵ-LEXICASE	Median RMSE % decrease LASSO	ϵ-LEXICASE
airfoil	2.4634	4.8484	3.6505	49.19 %	32.52 %
concrete	6.5123	10.5383	7.0707	38.20 %	7.90 %
enc	1.4838	3.2498	1.8647	54.34 %	20.43 %
enh	0.5560	2.9645	1.2952	81.25 %	57.07 %
housing	4.4700	4.9155	4.2785	9.06 %	-4.48 %
tower	0.2606	0.2953	0.2975	11.75 %	12.39 %
uball5d	0.0402	0.1939	0.0618	79.29 %	35.00 %
yacht	1.0221	9.0237	1.3577	88.67 %	24.72 %
			Average RMSE % decrease:	**51.47 %**	**23.19 %**

Table 3: Median number of evaluated nodes.

Data set	Median number of evaluated nodes SGP-DT	ϵ-LEXICASE	Reduction ratio ϵ-LEXICASE
airfoil	1.00E+10	9.28E+10	9.26×
concrete	1.14E+10	6.43E+10	5.64×
enc	1.18E+10	4.99E+10	4.25×
enh	1.18E+10	5.08E+10	4.30×
housing	7.70E+09	3.09E+10	4.02×
tower	7.21E+10	1.94E+11	2.69×
uball5d	9.83E+10	3.94E+11	4.01×
yacht	4.62E+09	2.00E+10	4.34×
		Average reduction ratio:	**4.81×**

4 CONCLUSION

SGP-DT presents a novel SGP approach that yields to low approximation error and computational cost when applied to the symbolic regression domain. On eight well-know datasets, SGP-DT outperforms both LASSO and ϵ-LEXICASE. This is an important result as ϵ-LEXICASE outperforms many GP-inspired algorithms [5].

REFERENCES

[1] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. 2004. Least Angle Regression. *The Annals of statistics* 32, 2 (2004), 407–499.

[2] Maarten Keijzer. 2003. Improving Symbolic Regression with Interval Arithmetic and Linear Scaling. In *European Conf. on Genetic Programming.* (EuroGP 03) 70–82.

[3] William La Cava, Lee Spector, and Kourosh Danai. 2016. Epsilon-lexicase Selection for Regression. In *Proceedings of the Genetic and Evolutionary Computation Conference.* (GECCO '16), 741–748.

[4] Nicholas Freitag McPhee, Brian Ohs, and Tyler Hutchison. 2008. Semantic Building Blocks in Genetic Programming. In *Genetic Programming*, 134–145.

[5] Patryk Orzechowski, William La Cava, and Jason H. Moore. 2018. Where Are We Now?: A Large Benchmark Study of Recent Symbolic Regression Methods. In *Proc. of the Genetic and Evolutionary Computation Conf. (GECCO '18).* 1183–1190.

[6] Stefano Ruberto, Valerio Terragni, and Jason H. Moore. 2020. SGP-DT: Semantic Genetic Programming Based on Dynamic Targets. In *Proceedings of the European Conference on Genetic Programming (EuroGP '20).* http://arxiv.org/abs/2001.11535.

[7] Stefano Ruberto, Leonardo Vanneschi, and Mauro Castelli. 2019. Genetic Programming with Semantic Equivalence Classes. In *Swarm and Evolutionary Computation 44 (2019), 453-469.*

[8] Leonardo Vanneschi, Mauro Castelli, and Sara Silva. 2014. A Survey of Semantic Methods in Genetic Programming. *Genetic Progr. and Evo. Machines* 15, 2, 195–214.