Less is More: A Call to Focus on Simpler Models in Genetic Programming for Interpretable Machine Learning

Marco VIRGOLIN [a,1], Eric MEDVET [b], Tanja ALDERLIESTEN [c] and
Peter A.N. BOSMAN [a]

[a] *Centrum Wiskunde & Informatica, Amsterdam, The Netherlands*
[b] *University of Trieste, Trieste, Italy*
[c] *Leiden University Medical Center, Leiden, The Netherlands*

Abstract. Interpretability can be critical for the safe and responsible use of machine learning models in high-stakes applications. So far, evolutionary computation (EC), in particular in the form of genetic programming (GP), represents a key enabler for the discovery of interpretable machine learning (IML) models. In this short paper, we argue that research in GP for IML needs to focus on searching in the space of low-complexity models, by investigating new kinds of search strategies and recombination methods. Moreover, based on our experience of bringing research into clinical practice, we believe that research should strive to design better ways of modelling and pursuing interpretability, for the obtained solutions to ultimately be most useful.

Keywords. evolutionary computation, genetic programming, interpretable machine learning, symbolic regression, human-machine interaction

1. Background

The field of explainable artificial intelligence (XAI) makes important contributions toward meeting the challenge of fairness and accountability in AI. The goal of XAI is to design algorithms that can help explain the decisions taken by unintelligible models (e.g., deep neural networks) or that can generate inherently interpretable machine learning (IML) models [1,2]. Evolutionary computation (EC) is routinely used as a powerful engine for both cases, see, e.g., [3,4,5,6,7,8]. Here, we focus on how EC is often used to generate IML models. A prime example of this is embodied by genetic programming (GP) [9] (or similar algorithms, such as grammatical evolution [10]) for symbolic regression (SR), i.e., techniques inspired by natural evolution to discover a governing equation from data through such means as selection and recombination [11]. GP and similar techniques have also been applied to other types of problems than SR (see, e.g., [5,6,12,13,14]); using other types of *symbolic models*, i.e., models composed of high-level operations such as arithmetic operations, Boolean conditions, software instructions (see, e.g., [4,15,16,17]).

---

[1]Corresponding Author: Marco Virgolin, Centrum Wiskunde & Informatica, Science Park 123, 1098 XG Amsterdam, The Netherlands; E-mail: marco.virgolin@cwi.nl.
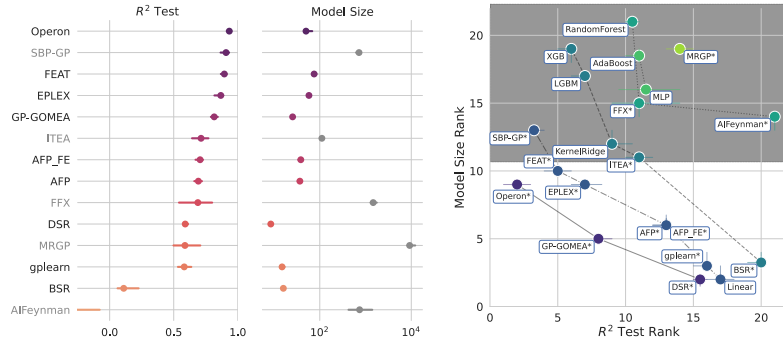
Figure 1. Algorithms in gray (left) or in the gray area (right) produce models with > 100 components on average on SRBench v.2.0. **Left:** Pairgrid of model accuracy (measured in terms of R^2 on the test set) and model size (measured in terms of total number of components) for the symbolic algorithms of SRBench. **Right:** Trade-off fronts in terms of model accuracy and size ranks for all the algorithms of SRBench (here, symbolic algorithms are starred). Plots made with [21].

2. Open problems and possible directions in GP for IML

A large drive in GP is to improve model accuracy much akin to a key performance indicator in ML. Unfortunately, sometimes this is done at the cost of ignoring the applicability of the approach. In fact, this has pushed research towards designing GP algorithms that, striving for maximizing accuracy, end up delivering models with hundreds, thousands, or even more components [18]. If we consider the current results reported in SRBench (v.2.0), a large benchmark on SR [19], a number of the competing algorithms produce models with more than 100 components; see Fig. 1. Based on our experience, even if composed of high-level operations, such models are already too large to be interpretable [20]. In our view, GP research aimed at IML should strive to improve the discovery of accurate models, subject to these models having limited complexity. Even so, the use of GP is likely to remain justified, as the number and type of operations that can be considered still leads to large search spaces that lack linearity or convexity. Searching for low-complexity models therefore calls for better search strategies. In fact, typical recombination methods (e.g., subtree crossover and mutation in GP [9]) but also new ones (e.g., semantic recombination [22,23]) replace or stack entire (and often large) blocks of components, leading to model growth [24], and are ineffective when the models at play need to remain small [20,25]. We are attempting to answer this need by studying how pattern identification and propagation studied in classic genetic algorithms [26] can be incorporated in GP: the result of our efforts so far, GP-GOMEA, achieves some of the best trade-offs between model accuracy and simplicity on SRBench (see Fig. 1).

Besides this, we believe that better objectives that act as proxies for interpretability are needed. Literature research typically adopts simplistic metrics as proxies for interpretability, such as model size or hand-crafted scores (see, e.g., [13,27]). In our experience in clinical applications [28,29,30], obtaining a good objective function of what the user truly needs requires several sessions of interaction. A promising direction here is to use ML itself to learn what specific users find to be more or less interpretable in an automatic fashion [31,32]. Moreover, researchers should strive to include user studies, to assess whether the proposed objective (interpretability, trust, etc.) obtains the desired effect in practice [32].

References

[1] Adadi A, Berrada M. Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). IEEE Access. 2018;6:52138-60.

[2] Arrieta AB, Díaz-Rodríguez N, Del Ser J, Bennetot A, Tabik S, Barbado A, et al. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion. 2020;58:82-115.

[3] Johnson HE, Gilbert RJ, Winson MK, Goodacre R, Smith AR, Rowland JJ, et al. Explanatory analysis of the metabolome using genetic programming of simple, interpretable rules. Genetic Programming and Evolvable Machines. 2000;1(3):243-58.

[4] Cano A, Zafra A, Ventura S. An interpretable classification rule mining algorithm. Information Sciences. 2013;240:1-20.

[5] Virgolin M, Alderliesten T, Bosman PAN. On explaining machine learning models by evolving crucial and compact features. Swarm and Evolutionary Computation. 2020;53:100640.

[6] Lensen A, Xue B, Zhang M. Genetic programming for evolving a front of interpretable models for data visualization. IEEE Transactions on Cybernetics. 2020;51(11):5468-82.

[7] Dandl S, Molnar C, Binder M, Bischl B. Multi-objective counterfactual explanations. In: International Conference on Parallel Problem Solving from Nature. Springer; 2020. p. 448-69.

[8] Virgolin M, Fracaros S. On the robustness of counterfactual explanations to adverse perturbations. arXiv preprint arXiv:220109051. 2022.

[9] Koza JR. Genetic programming as a means for programming computers by natural selection. Statistics and Computing. 1994;4(2):87-112.

[10] O'Neill M, Ryan C. Grammatical evolution. IEEE Transactions on Evolutionary Computation. 2001;5(4):349-58.

[11] Schmidt M, Lipson H. Distilling free-form natural laws from experimental data. science. 2009;324(5923):81-5.

[12] Espejo PG, Ventura S, Herrera F. A survey on the application of genetic programming to classification. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews). 2009;40(2):121-44.

[13] Hein D, Udluft S, Runkler TA. Interpretable policies for reinforcement learning by genetic programming. Engineering Applications of Artificial Intelligence. 2018;76:158-69.

[14] De Lorenzo A, Bartoli A, Castelli M, Medvet E, Xue B. Genetic programming in the twenty-first century: A bibliometric and content-based analysis from both sides of the fence. Genetic Programming and Evolvable Machines. 2020;21(1):181-204.

[15] Mansoori EG, Zolghadri MJ, Katebi SD. SGERD: A steady-state genetic algorithm for extracting fuzzy classification rules from data. IEEE Transactions on Fuzzy Systems. 2008;16(4):1061-71.

[16] Aitkenhead MJ. A co-evolving decision tree classification method. Expert Systems with Applications. 2008;34(1):18-25.

[17] Zhang H, Zhou A, Qian H, Zhang H. PS-Tree: A piecewise symbolic regression tree. Swarm and Evolutionary Computation. 2022:101061.

[18] Martins JFB, Oliveira LOVB, Miranda LF, Casadei F, Pappa GL. Solving the exponential growth of symbolic regression trees in geometric semantic genetic programming. In: Proceedings of the Genetic and Evolutionary Computation Conference; 2018. p. 1151-8.

[19] La Cava W, Orzechowski P, Burlacu B, de França FO, Virgolin M, Jin Y, et al. Contemporary symbolic regression methods and their relative performance. Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks. 2021.

[20] Virgolin M, Alderliesten T, Witteveen C, Bosman PAN. Improving model-based genetic programming for symbolic regression of small expressions. Evolutionary computation. 2021;29(2):211-37.

[21] La Cava W, other contributors. SRBench (v2.0); 2022. Code repository, accessed April 1 2022. Available from: https://github.com/cavalab/srbench.

[22] Moraglio A, Krawiec K, Johnson CG. Geometric semantic genetic programming. In: International Conference on Parallel Problem Solving from Nature. Springer; 2012. p. 21-31.

[23] Pawlak TP, Wieloch B, Krawiec K. Semantic backpropagation for designing search operators in genetic programming. IEEE Transactions on Evolutionary Computation. 2014;19(3):326-40.

[24] Vanneschi L, Castelli M, Silva S. Measuring bloat, overfitting and functional complexity in genetic programming. In: Proceedings of the 12th annual conference on Genetic and evolutionary computation; 2010. p. 877-84.

[25]   Liu D, Virgolin M, Alderliesten T, Bosman PAN. Evolvability degeneration in multi-objective genetic programming for symbolic regression. arXiv preprint arXiv:220206983 (to appear in GECCO '22). 2022.

[26]   Thierens D, Bosman PAN. Optimal mixing evolutionary algorithms. In: Proceedings of the Genetic and Evolutionary Computation Conference; 2011. p. 617-24.

[27]   Vladislavleva EJ, Smits GF, Den Hertog D. Order of nonlinearity as a complexity measure for models generated by symbolic regression via pareto genetic programming. IEEE Transactions on Evolutionary Computation. 2008;13(2):333-49.

[28]   Maree SC, Luong NH, Kooreman ES, van Wieringen N, Bel A, Hinnen KA, et al. Evaluation of bi-objective treatment planning for high-dose-rate prostate brachytherapy—A retrospective observer study. Brachytherapy. 2019;18(3):396-403.

[29]   Virgolin M, Wang Z, Balgobind BV, van Dijk IW, Wiersma J, Kroon PS, et al. Surrogate-free machine learning-based organ dose reconstruction for pediatric abdominal radiotherapy. Physics in Medicine & Biology. 2020;65(24):245021.

[30]   van der Meer MC, Bosman PAN, Niatsetski Y, Alderliesten T, Pieters BR, Bel A. Robust optimization for HDR prostate brachytherapy applied to organ reconstruction uncertainty. Physics in Medicine & Biology. 2021;66(5):055001.

[31]   Virgolin M, De Lorenzo A, Medvet E, Randone F. Learning a formula of interpretability to learn interpretable formulas. In: International Conference on Parallel Problem Solving from Nature. Springer; 2020. p. 79-93.

[32]   Virgolin M, De Lorenzo A, Randone F, Medvet E, Wahde M. Model learning with personalized interpretability estimation (ML-PIE). In: Proceedings of the Genetic and Evolutionary Computation Conference Companion; 2021. p. 1355-64.