Complexity Measures for Multi-Objective Symbolic Regression

Michael Kommenda[1,2], Andreas Beham[1,2],
Michael Affenzeller[1,2], and Gabriel Kronberger[1]

[1] Heuristic and Evolutionary Algorithms Laboratory
School of Informatics, Communications and Media
University of Applied Sciences Upper Austria
Softwarepark 11, 4232 Hagenberg, Austria

[2] Institute for Formal Models and Verification
Johannes Kepler University Linz
Altenbergerstr. 69, 4040 Linz, Austria

{`michael.kommenda, andreas.beham, michael.affenzeller,`
`gabriel.kronberger`}`@fh-hagenberg.at`

Abstract. Multi-objective symbolic regression has the advantage that while the accuracy of the learned models is maximized, the complexity is automatically adapted and need not be specified a-priori. The result of the optimization is not a single solution anymore, but a whole Pareto-front describing the trade-off between accuracy and complexity.
In this contribution we study which complexity measures are most appropriately used in symbolic regression when performing multi-objective optimization with NSGA-II. Furthermore, we present a novel complexity measure that includes semantic information based on the function symbols occurring in the models and test its effects on several benchmark datasets. Results comparing multiple complexity measures are presented in terms of the achieved accuracy and model length to illustrate how the search direction of the algorithm is affected.

Keywords: Symbolic Regression, Complexity Measures, Multi-objective Optimization, NSGA-II, Genetic Programming

1 Introduction

Symbolic regression is a data-based machine learning method, where the relation between several independent and one dependent variable is modeled. Contrary to other modeling methods the structure of the learned model is not specified a-priori, but determined during the algorithm execution. Symbolic regression problems are commonly solved by genetic programming (GP) [6], because the variable-length encoding used in GP is particularly suited for the evolution of the

model structure. An expression tree encoding is frequently used in GP for symbolic regression, where every leaf node represent either a variable or a numeric constant and every internal node a mathematical function. Thus, every expression tree represents a mathematical formula that can be interpreted, validated and easily incorporated in other programs [1].

The phenomenon of bloat in GP [7] (an increase in the average size of the individuals without an corresponding increase in fitness), overly complex and large individuals, or the excessive use of variables reduce the interpretability of symbolic regression methods. As bloat and introns are not specific to symbolic regression, but rather occur when using arbitrary-sized representations in evolutionary computation [8], several methods to limit the growth of GP individuals and to counteract bloat have been suggested previously. One approach is to specify static size and depth limits for the symbolic expression trees used in GP [10] that must not be exceeded. However, these two limits are highly problem-dependent, cannot be known a-priori and must be adapted for each problem so that the trees can grow large enough to model the data accurately while unnecessary complexity is avoided. Other methods of controlling the tree size range from dynamic size limits [11] or parsimony pressure methods [9] to controlling the distribution of tree sizes [3].

The previously mentioned methods have been developed to limit the growth of symbolic expression trees in GP and do not include any semantic information about the mathematical formulas represented by the expression trees when performing symbolic regression. Although complexity is to some extent correlated to size, it is not necessarily the case that the complexity of a formula is reflected in its size. For example the formula $f(x) = e^{\sin\sqrt{x}}$ consists of three operations, one variable, and one constant, while $f(x) = 7x^2 + 3x + 5$ consists of five operations, two variable, and three constants and is intuitively less complex.

A different approach for managing complexity and model size in symbolic regression is to use multi-objective optimization, where while maximizing the prediction accuracy the complexity is minimized [12,8]. Hence, no size limits or other complexity related parameters must be configured and the optimization algorithm is expected to automatically evolve solutions of appropriate length and complexity. In this contribution we compare the effects on algorithm performance of several complexity and size related quality criteria for multi-objective symbolic regression on benchmark problems.

2 Multi-objective Symbolic Regression

The nondominant sorting genetic algorithm II (NSGA-II) [2] is one of the most prominent algorithms for multi-objective optimization. It uses a novel selection mechanism based on the nondomination rank and crowding distance for selection to build a uniformly spread Pareto-optimal front. In the case of multi-objective symbolic regression the objectives to be optimized are the prediction accuracy and the complexity of the learned models. The prediction accuracy can be expressed by any error or correlation measure such as the coefficient of determina-

tion R^2, the mean squared error, or the mean absolute percentage error between the estimated and observed values.

The complexity of a symbolic regression model can be calculated as the *tree length* or the *expressional complexity* (visitation length) [12,5]. More sophisticated measures that also include semantics of the evolved models range from the number of included variables, or the *order of nonlinearity* [14] to the *functional complexity* [13]. While the *order of nonlinearity* and the *functional complexity* express the complexity of a symbolic regression model rather accurately, they are computationally expensive to calculate. On the other hand, measures such as the *tree length* or the *expressional complexity* are efficiently calculated, but do not include any information except the shape of the symbolic expression tree encoding the model.

We propose a new complexity measure that is easy to calculate and includes semantics about the regression model, so that the search direction of the multi-objective algorithm is altered towards simple and parsimonious models. The measure is calculated by recursive iteration over the symbolic expression tree and accumulates the individual complexity values for each subtree while taking into account different complexity values for the encountered function symbols. The mathematical definition for the calculation of this new complexity measure is given in Equation 1, where n denotes a tree node and c a direct child node of n. The complexity of the whole symbolic expression tree encoding the regression model can then be calculated by recursive application of Equation 1 starting at the root node.

$$\text{Complexity}(n) = \begin{cases} 1 & \text{if } n \equiv \text{constant} \\ 2 & \text{if } n \equiv \text{variable} \\ \sum \text{Complexity}(c) & \text{if } n \in (+, -) \\ \prod \text{Complexity}(c) + 1 & \text{if } n \in (*, /) \\ \text{Complexity}(c)^2 & \text{if } n \equiv \text{square} \\ \text{Complexity}(c)^3 & \text{if } n \equiv \text{squareroot} \\ 2^{\text{Complexity}(c)} & \text{if } n \in (\sin, \cos, \tan, \exp, \log) \end{cases} \tag{1}$$

In the following we explore the complexity differences of two exemplary models, $f_1(x) = e^{\sin \sqrt{x}}$ and $f_2(x) = 7x^2 + 3x + 5$. The corresponding symbolic expression trees representing these formulas are illustrated in Figure 1. The tree length for $f_1(x)$ is 4 and 9 for $f_2(x)$, which indicates that representation of $f_1(x)$ is more compact. However, Figure 1 also shows the calculation steps for the new complexity measure according to Equation 1, which is iteratively applied starting at the leaf nodes. The complexity measure results in 65536 for $f_1(x)$ and 17 for $f_2(x)$, which reflects our intuition that $f_2(x)$ is less complex and easier to interpret than $f_1(x)$, whereas according to the tree length the contrary is true.
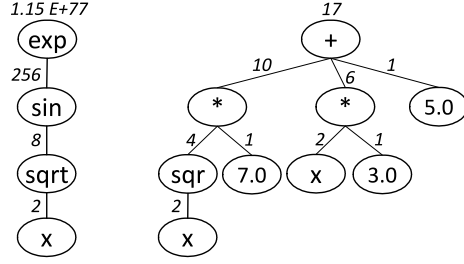
Fig. 1. Symbolic expression tree representation of $f_1(x) = e^{\sin\sqrt{x}}$ and $f_2(x) = 7x^2 + 3x + 5$, where the calculation steps for the complexity measure are indicated next to the arcs.

3 Experiments

We used an NSGA-II algorithm to test the effects of different complexity measures for multi-objective symbolic regression. The first objective for the algorithm is the Pearson's R^2 correlation describing the model accuracy. Varying complexity measures such as the number of used variables (*Variables*), the model length (*Tree Length*), the expressional complexity (*Visitation Length*) and the new complexity measure (*Complexity*) described in Equation 1 have been used as second optimization objective. Despite maximizing the Pearson's R^2, the results regarding accuracy are presented in terms of the normalized mean squared error (NMSE), so that for both measures smaller values are better.

NSGA-II was configured to evolve models with a maximum tree length of 100 that are allowed to include arithmetic $(+, -, *, /)$, trigonometric $(\sin, \cos, \tan)$, power$(^2, \sqrt{})$ and exponential $(\exp, \log)$ symbols and stops when the termination criterion of $200{,}000$ model evaluations has been reached. The benchmark problems used for testing have been selected from [15,4] and the generating formulas are listed in Table 1.

Table 1. Definition of benchmark problems.

Name	Function
Keijzer-5	$f(x_1, x_2, x_3) = 30x_1x_3/[(x_1 - 10)x_2^2]$
Vladislavleva-1	$f(x_1, x_2) = e^{-(x_1-1)^2)}/[1.2 + (x_2 - 2.5)^2]$
Vladislavleva-2	$f(x_1) = e^{-x_1}x_1^3\cos(x)\sin(x)(\cos(x)\sin^2(x) - 1)$
Vladislavleva-7	$f(x_1, x_2) = (x_1 - 3)(x_2 - 3) + 2\sin((x_1 - 4)(x_2 - 4))$
Pagie-1	$f(x_1, x_2) = 1/[1 + x_1^{-4}] + 1/[1 + x_2^{-4}]$
Poly-10	$f(x_1 - x_{10}) = x_1x_2 + x_3x_4 + x_5x_6 + x_1x_7x_9 + x_3x_6x_{10}$
Friedman-1	$f(x_1 - x_{10}) = 0.1e^{4x_1} + 4/[1 + e^{-20x_2} + 10] + 3x_3 + 2x_2 + x_5 + N$
Friedman-2	$f(x_1 - x_{10}) = 10\sin(\pi x_1x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + N$
Tower	Real world data

When switching from single to multi-objective algorithms the result of an algorithm execution is not a single solution any more, but rather a Pareto-front showing the trade-off between accuracy and complexity of the models. An example of a Pareto-front generated by the NSGA-II is shown in Figure 2, where besides the training qualities the models' accuracies on the test set are shown to evaluate their generalization capabilities. However, whole Pareto-fronts are hard to compare, especially in the case of symbolic regression where a high training quality doesn't necessary indicate a good model due to overfitting reasons (for example the largest models in Figure 2 whose test NMSEs exceeds 1.0 and are not displayed at all). Hence, we used only single models of a Pareto-front for algorithm comparison and used the model with the highest training accuracy. A better way for model selection would be, if an additional validation partition is defined, to use the model with the highest accuracy on this partition.
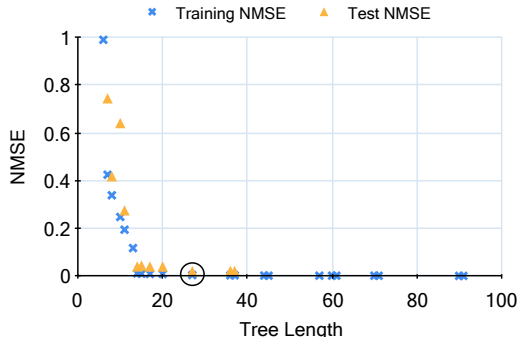

Fig. 2. Exemplary Pareto-front evolved by NSGA-II showing the trade-off between accuracy in terms of the NMSE and the model length. Every model contributes to two data points, one for training and one for test evaluation (if no test evaluation is shown, the NMSE exceeds 1.0). The *best* model, which is rather small but still accurate, is encircled.

4   Results

We have performed 50 repetitions of each NSGA-II configuration to account for the stochastic nature of the algorithm. We extracted the most accurate models, at the same time the most complex ones, from each generated Pareto-front and compared them against each other. In Table 2 the average and standard deviation of those models are displayed. Although for most of the problems the results are quite similar, there are some differences. The *Complexity* configuration obtains by far the best results with the smallest variation on the Poly-10 problem. The *Variables* configuration works best on the Friedman-2 problem and both configuration perfrom well on the Pagie-1 problem.

Table 2. Average and standard deviation ($\mu \pm \sigma$) of the training qualities (NMSE) of the best individual for 50 repetitions of NSGA-II with varying complexity measures.

Problem	Variables	Tree Length	Visitation Length	Complexity
Keijzer-5	0.000 ± 0.000	0.003 ± 0.003	0.002 ± 0.003	0.000 ± 0.000
Vladislavleva-1	0.002 ± 0.002	0.005 ± 0.011	0.004 ± 0.004	0.002 ± 0.002
Vladislavleva-2	0.022 ± 0.020	0.018 ± 0.016	0.014 ± 0.012	0.016 ± 0.013
Vladislavleva-7	0.111 ± 0.020	0.147 ± 0.077	0.116 ± 0.027	0.115 ± 0.027
Pagie-1	0.001 ± 0.003	0.011 ± 0.017	0.012 ± 0.018	0.007 ± 0.003
Poly-10	0.294 ± 0.126	0.356 ± 0.165	0.341 ± 0.168	0.202 ± 0.079
Friedman-1	0.140 ± 0.004	0.157 ± 0.019	0.163 ± 0.023	0.175 ± 0.021
Friedman-2	0.081 ± 0.031	0.134 ± 0.050	0.146 ± 0.038	0.143 ± 0.049
Tower	0.148 ± 0.025	0.148 ± 0.020	0.146 ± 0.016	0.140 ± 0.018

The generalization capabilities of the best models have been evaluated on a separate test partition and those results are shown in Table 3. Every configuration produces, at least on some problems, overfit models, which is indicate by an high average normalized squared error and high standard deviations. The *Tree Length* and *Visitation Length* perform equally well with the exception of the Vladislavleva-1 and Pagie-1 problem. It still holds that the *Complexity* and *Variables* algorithm runs produced the best models on the Poly-10 and Friedman-2 problem respectively. An interesting observation is that excluding Vladislavleva-1 and Vladislavleva-2 the *Complexity* measure as second objective performs either better or equally well as the other configurations. A reason for this might be that those two problems consist of complicated formulas that have to be discovered and the algorithm is not able to build that complicated yet accurate formulas. The other complexity measure that do not include semantic information about the models, have no such limitations as long as the models are compact enough.

Table 3. Average and standard deviation ($\mu \pm \sigma$) of the quality of the best training individual evaluated on the test partition for 50 repetitions of the multi-objective symbolic regression algorithm with varying complexity measures.

Problem	Variables	Tree Length	Visitation Length	Complexity
Keijzer-5	0.000 ± 0.000	0.004 ± 0.003	0.002 ± 0.003	0.000 ± 0.000
Vladislavleva-1	1.568 ± 1.568	0.047 ± 0.062	1.460 ± 8.291	5.509 ± 12.00
Vladislavleva-2	0.112 ± 0.587	0.023 ± 0.022	0.019 ± 0.014	0.823 ± 3.193
Vladislavleva-7	0.529 ± 2.753	0.168 ± 0.099	0.138 ± 0.037	0.138 ± 0.125
Pagie-1	0.015 ± 2.174	0.445 ± 1.737	0.061 ± 0.087	0.028 ± 0.046
Poly-10	0.457 ± 0.211	0.558 ± 1.188	0.510 ± 0.626	0.301 ± 0.129
Friedman-1	0.150 ± 0.000	0.156 ± 0.019	0.169 ± 0.024	0.175 ± 0.021
Friedman-2	0.083 ± 0.032	0.143 ± 0.054	0.149 ± 0.039	0.140 ± 0.049
Tower	0.138 ± 0.026	0.144 ± 0.020	0.144 ± 0.019	0.138 ± 0.018

Next to the accuracy of the models their interpretability is of importance, because model interpretability is one of the major reasons to use symbolic regression. The lengths of the models generated by NSGA-II with varying complexity measures have been compared and the results are stated in Table 4. It is clear that the *Variables* configuration, which just counts the variable occurrences, generates by far the largest models as no selection pressure towards more parsimonious ones is applied. The *Tree Length* and *Visitation Length* produce the smallest models and no significant differences could be found between these two variations. Although no explicit parsimony pressure is applied to models created by NSGA-II with the new complexity measure, these are smaller than the ones produced by *Variables*, but still larger than those using explicitly the tree and visitation length for optimization.

Table 4. Average and standard deviation ($\mu \pm \sigma$) of the length of the best training individuals for 50 repetitions of NSGA-II with varying complexity measures.

Problem	Variables	Tree Length	Visitation Length	Complexity
Keijzer-5	86.3 ± 19.34	21.2 ± 13.30	22.7 ± 15.75	47.0 ± 18.59
Vladislavleva-1	91.8 ± 15.27	44.6 ± 22.72	41.7 ± 24.71	85.7 ± 22.22
Vladislavleva-2	88.7 ± 16.81	42.6 ± 23.47	37.7 ± 21.21	79.8 ± 21.77
Vladislavleva-7	94.4 ± 10.93	44.0 ± 29.41	48.4 ± 29.98	81.3 ± 24.08
Pagie-1	91.7 ± 16.50	51.7 ± 28.88	40.6 ± 23.34	72.4 ± 28.01
Poly-10	96.8 ± 7.320	53.0 ± 30.79	54.8 ± 30.06	72.7 ± 24.70
Friedman-1	90.5 ± 12.96	61.0 ± 26.81	55.1 ± 29.74	69.2 ± 26.53
Friedman-2	91.4 ± 12.00	47.1 ± 28.60	40.8 ± 25.30	68.3 ± 25.54
Tower	94.4 ± 8.860	59.8 ± 29.30	58.9 ± 27.02	78.2 ± 21.07

5 Conclusion

In this publication we have investigated the effects of different complexity measures on multi-objective symbolic regression. Furthermore, we have presented a novel complexity measure based on the mathematical symbols occurring in the formulas. The differences with respect to the accuracies of the models on the tested benchmark problems were in most cases not significant. An exception is the new complexity measure that performs best on problems with simpler data generating formulas, but on the other hand fails to evolve well-fitting models on the most complex problems.

When comparing the length of the evolved models to give an indication of their interpretability, the algorithm configurations explicitly using the tree length as an optimization objective generated the most parsimonious models. As previously argued and used as motivation for the development of the new complexity measure, the length is only to some extend correlated to simplicity and interpretability and more research to illustrate the differences of the complexity measures has to be performed.

References

1. Affenzeller, M., Winkler, S., Kronberger, G., Kommenda, M., Burlacu, B., Wagner, S.: Gaining deeper insights in symbolic regression. In: Riolo, R., Moore, J.H., Kotanchek, M. (eds.) Genetic Programming Theory and Practice XI. Genetic and Evolutionary Computation, Springer (2014)
2. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. Evolutionary Computation, IEEE Transactions on 6(2), 182–197 (2002)
3. Dignum, S., Poli, R.: Operator equalisation and bloat free gp. In: Genetic Programming, pp. 110–121. Springer (2008)
4. Friedman, J.H.: Multivariate adaptive regression splines. The annals of statistics pp. 1–67 (1991)
5. Keijzer, M., Foster, J.: Crossover bias in genetic programming. In: Genetic Programming, pp. 33–44. Springer (2007)
6. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge, MA, USA (1992)
7. Luke, S.: Issues in scaling genetic programming: breeding strategies, tree generation, and code bloat. Ph.D. thesis, Dept. of Computer Science.University of Maryland, College Park (2000)
8. Luke, S., Panait, L., et al.: Lexicographic parsimony pressure. In: GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference. vol. 2, pp. 829–836 (2002)
9. Poli, R.: Covariant tarpeian method for bloat control in genetic programming. Genetic Programming Theory and Practice VIII 8, 71–90 (2010)
10. Poli, R., Langdon, W.B., McPhee, N.F.: A field guide to genetic programming. Published via `http://lulu.com` (2008)
11. Silva, S., Costa, E.: Dynamic limits for bloat control in genetic programming and a review of past and current bloat theories. Genetic Programming and Evolvable Machines 10(2), 141–179 (2009)
12. Smits, G.F., Kotanchek, M.: Pareto-front exploitation in symbolic regression. In: Genetic programming theory and practice II, pp. 283–299. Springer (2005)
13. 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. pp. 877–884. ACM (2010)
14. Vladislavleva, E.J., Smits, G.F., Den Hertog, D.: Order of nonlinearity as a complexity measure for models generated by symbolic regression via pareto genetic programming. Evolutionary Computation, IEEE Transactions on 13(2), 333–349 (2009)
15. White, D.R., McDermott, J., Castelli, M., Manzoni, L., Goldman, B.W., Kronberger, G., Jaskowski, W., O'Reilly, U.M., Luke, S.: Better GP benchmarks: community survey results and proposals. Genetic Programming and Evolvable Machines 14(1), 3–29 (Mar 2013)