Residuals for Equation Discovery

Jannis Brugger [1 2 *], Viktor Pfanschilling[1 5], Mira Mezini[1 2 3] Stefan Kramer [4]
[1] Technical University of Darmstadt, 64289 Darmstadt, Germany
[2] Hessian Center for Artificial Intelligence (hessian.AI), 64293 Darmstadt, Germany
[3] National Research Center for Applied Cybersecurity ATHENE
[4] Johannes Gutenberg-Universität Mainz, 55128 Mainz, Germany
[5] German Research Center for Artificial Intelligence, 67663 Kaiserslautern, Germany
[*] Correspondence: jannis.brugger@tu-darmstadt.de

Equation discovery is the task, given a data set $D \in \mathbb{R}^{m \times n}$, to find for m examples the equation $f(\mathbf{x}, \mathbf{c}) = \mathbf{y}$ mapping the independent variables $x_i | 1 \leq i \leq n-1$ and the constants $\mathbf{c}$ to the dependent variable y, where the functional form of the equation is to be determined during the process. The search space is either implicitly given or given by a grammar. The idea of this work is to decompose the equation $f(\mathbf{x}) = g(\mathbf{x}) \circ h(\mathbf{x})$ into simpler parts, as done already, in different ways, in the BACON and AIFeynman systems. In recent years the usage of pretrained equation discovery systems have become popular. These systems use a neural architecture to embed the data set and train to predict the equation that generated the data set in a zero-shot way. *Residuals for equation discovery (RED)* combines the ideas of decomposition and the ability of zero-shot prediction. RED calculates and optimizes the residuals of the data set for a subequation X of the initial equation. In other words, it computes what that subequation should have yielded for each data point for the entire formula to predict the output correctly. These residuals $\mathbf{y}'$ formulate a new problem $f'(\mathbf{x}, \mathbf{c}) = \mathbf{y}'$, and the equation discovery system can predict a solution. If the new solution's error is lower than that of the old solution, the new solution can replace X in the original equation. By calculating residuals, the system can disentangle the original task into simpler tasks and interactively discover the original equation. Calculating the residuals is as fast as evaluating the initial equation when the equation is expressed in a syntax tree, where leaf nodes are constants or variables and the inner nodes are operators. For each operator, we define how it should behave when the syntax tree is evaluated or a residual is calculated. The equation $y = x_1 + 2$ can be represented as a Y-node (id:0) connected to a Plus node (id:1) with child nodes of type Variable with the value x_1 (id:2) and Constant with the value 2 (id:3). If the residual for the node with id:2 is calculated, the syntax tree evaluates the equation $R_2 = y - x_1$. The residual cannot be calculated for nodes that are the descendant of an operation that is not bijective (e.g., sin). We show, for the model NeSymReS [1] on the Feynman equations with up to two variables as reported in SRBench [2], that RED improves the median mean squared error from 0.89 (IQR 0.06-9.21) to 0.003 (IQR 0.001 - 0.08). While RED is independent of the functionality of the pre-trained equation discovery system, it depends on an initial solution, which has to enable the disentanglement.

References

[1] Luca Biggio et al. "Neural Symbolic Regression that scales". In: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021*. Ed. by Marina Meila et al. PMLR, 2021, pp. 936–945.

[2] William G. La Cava et al. "Contemporary Symbolic Regression Methods and their Relative Performance". In: *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*. 2021.