[Supplementary Information]

Towards data-driven discovery of governing equations in geosciences

Wenxiang Song[1], Shijie Jiang[2,3], Gustau Camps-Valls[4], Mathew Williams[5], Lu Zhang[1,6], Markus Reichstein[2,3], Harry Vereecken[7], Leilei He[1], Xiaolong Hu[1], Liangsheng Shi[1*]

[1] State Key Laboratory of Water Resources Engineering and Management, Wuhan University, Wuhan, China.

[2] Department of Biogeochemical Integration, Max Planck Institute for Biogeochemistry, Jena, Germany.

[3] ELLIS Unit Jena, Jena, Germany.

[4] Image Processing Laboratory (IPL), Universitat de València, Valencia, Spain.

[5] School of GeoSciences, University of Edinburgh, Edinburgh, UK.

[6] CSIRO Land and Water, Black Mountain, Canberra, Australia.

[7] Institute of Bio- and Geosciences: Agrosphere (IBG-3), Forschungszentrum Jülich, Jülich, Germany.

* Corresponding author, E-mail: liangshs@whu.edu.cn

This Supplementary Information contains Supplementary Note 1.

Supplementary Note 1

Across the spectrum of data-driven discovery methods, distinct characteristics emerge in terms of applicability, level of prior information required, and actual performance on a variety of metrics. Here, we provide an introduction to different data-driven equation discovery approaches.

Sparse regression

Sparse regression is a regression method when feature selection is required[1]. The term "sparse" refers to only a small subset of features being essential, while the others are ignored and effectively set to zero. The central assumption of sparse regression for discovering governing equations from data is that most physical systems are governed by only a few terms, generally represented by derivative terms[2]. The sparsity-promoting methods, such as LASSO[3] and STRidge[4], are designed to solve equation discovery tasks without a brute-force search over all possible combinations. For example, if the goal is to discover the Burgers equations (e.g., $u_t = -uu_x + u_{xx}$), one can assume that the equation is $u_t = \Xi_1 u + \Xi_2 u_x + \Xi_3 uu_x + \Xi_4 u_{xx} + \Xi_5 uu_{xx} + \Xi_6 sin(u) u_{xx} + \cdots$. Sparse regression ensures that every correct nonzero Ξ_i can be detected. Due to its low computational cost and few hyperparameters, it has developed rapidly and is especially suitable for those high-dimensional PDE systems [5]. A recent notable improvement direction is to combine sparse regression with physics-informed neural networks[6,7]. In these methods, neural network parameters are determined by fitting to noisy data and satisfying governing equations. The exact form of these equations is left unspecified, with the PDE provided in an extended format that includes additional terms not present in the true governing equation and subsequently filtered out by sparse regression[8–14]. This approach can significantly enhance the performance of sparse regression, making it more robust to noise and sparse data.

Equation learner networks

Recently, a neural network architecture called the EQL network, tailored explicitly for equation discovery, has been proposed[15,16]. The EQL network replaces conventional activation functions (e.g., $\mathrm{Relu}, \tanh$) with basic mathematical operators (e.g., $\times, (\cdot)^2$) and establishes connections using fully connected layers. The training processes are the same as conventional neural networks, e.g., stochastic gradient descent methods. Once trained, the discovered equation can simply be read from the weights of the networks, analogous to the coefficients in linear regression. Regularization constraints are imposed during training to enforce the necessary sparsity for the physical equations. The EQL network has been extended to high-dimensional and dynamic systems, including PDEs and parametric PDEs[17–19]. The EQL network allows end-to-end training of the entire architecture through backpropagation, and it can interpolate seamlessly in time and make predictions at arbitrary time points[19], demonstrating remarkable flexibility. The drawback of the EQL network is that it does not consistently achieve optimal convergence. This issue arises because the regularization constraint is implemented through the loss function, which is a soft constraint and cannot strictly guide the network training.

Genetic programming

Genetic programming (GP) is an evolutionary algorithm that efficiently explores potential solutions to specific problem ions within a given search object. There are various applications of GP[20], and the first introduction for symbolic regression is generally attributed to Koza[21], who demonstrated that GP could be employed for symbolic equations by encoding mathematical expressions as computational trees. Each tree node represents an operation (e.g., $+,-,\times,\sin$) and an operand containing physical and constant variables (e.g., x,y,z,π). In the algorithm, a population of individuals represented by trees is randomly generated. This population then evolves according to established evolutionary rules that include mutation, crossover, and other advanced operations. Individuals with high fitness are selected from the current population to serve as optimal individuals and parents for the next generation. This iterative process continues until a predetermined level of accuracy is achieved. GP is flexible in that it requires minimal prior physical information to derive the governing equations from the data. However, the main drawbacks of GP are premature convergence (i.e., the algorithm stops evolving too early) and bloat (i.e., the equations become unnecessarily complex), leading to equations of poor accuracy and excessive complexity. Numerous improvements have been proposed to enhance the performance of GP and successfully discover complex ODEs and PDEs from data[22–26]. One approach to improvement is to incorporate structural assumptions into the equations by defining model encodings within GP, such as dimensional homogeneity that ensures terms in the generated equations are consistent in their units[27].

Heuristic symbolic regression

Heuristic symbolic regression (HSR) is another time-honored symbolic regression algorithm that facilitates data-driven discovery of mathematical expressions through iterative heuristic tests on a dataset. HSR uses underlying heuristic rules to simplify equation discovery, such as physical regulations. In the 1970s, Gerwin[28] developed the first HSR method for discovering complicated mathematical functions of a single variable from data. Subsequently, Langley[29] and Falkenhainer and Michalski[30] proposed analogous HSR approaches. In recent years, HSR has received renewed interest. AIFeynman[31] and its updated version[32] are innovative heuristic symbolic regression techniques that use a divide-and-conquer strategy and problem decomposition heuristics for equation extraction, which helps partition the data into simpler sub-problems and allows recursion. While current tests focus on algebraic equations, future research may extend to complex PDEs due to the scalability of HSR [31].

Mixed-integer nonlinear-programming approaches

Mixed-integer nonlinear programming (MINLP) is a branch of mathematical optimization that deals with the optimization of a mathematical model containing both continuous and discrete decision variables and nonlinear functions[33]. The equation discovery problem can be formulated as a MINLP problem and solving it using established optimization algorithms[34–38]. The advantage of MINLP is that deterministic optimization techniques provide globally optimal mathematical expressions, bypassing exhaustive solution space search, and are thus much faster than other symbolic regression methods[37]. However, it is crucial to note that the computational time for MINLP solvers grows exponentially with the size of the input data. This becomes particularly pronounced when dealing with noisy data in geoscience analysis, where substantial computational resources are required to explore the symbolic expression space and identify solutions with low error and simplified expressions[34].

Deep reinforcement learning

Deep reinforcement learning (DRL) is a deep learning algorithm that allows agents to learn and make decisions through trial and error, often used in tasks where an agent interacts with an environment to maximize a cumulative reward, such as autonomous vehicles and robotics control. Recently, DRL has been applied as an effective symbolic regression approach for discovering unknown governing equations[39,40]. This method employs a recurrent neural network (RNN) trained through reinforcement learning techniques. The RNN generates equations and utilizes a reward function to assess their performance in terms of equation accuracy by comparing predicted values with ground truth values. The original framework, known as Deep Symbolic Regression (DSR)[39], has been refined in subsequent studies to enhance model performance[41–43] and ensure physical consistency [44]. DRL is advantageous for its ability to narrow the search space and incorporate in-situ physical constraints (e.g., we can assume that the right child nodes of partial differential operators must be space variables when discovering PDEs), leading to superior performance in various tests of governing equation discovery[39,42,43].

Large-scale pre-trained Transformers

Transformers[45] is a deep learning model that employs self-attention mechanisms to dynamically weigh the importance of different parts of the input data, enabling exceptional performance in complex tasks such as machine translation and video understanding. Recently, significant progress has been made in developing large-scale pre-trained Transformers for symbolic regression. This approach generates substantial training data and then trains a transformer model using a supervised learning approach [46–50]. The essential principle of this approach is to take advantage of the ease with which symbolic expressions for scalar functions can be generated and evaluated with random inputs, resulting in an abundance of training data. The large-scale pre-trained Transformer method can recover symbolic equations much faster than other approaches, particularly for complex expressions because the inference process is performed by trained Transformers with fixed parameters. Moreover, their inherent capability of transfer learning makes them ideal for inferring new governing equations by utilizing knowledge from previous tasks with minimal additional training. It is worth noting that the lack of fine-tuning during testing can lead to reduced algorithm performance when faced with new out-of-distribution problems.

Supplementary References

1. Bertsimas, D., Pauphilet, J. & Van Parys, B. Sparse Regression: Scalable Algorithms and Empirical Performance. *Stat. Sci.* **35**, 555–578 (2020).

2. Brunton, S. L., Proctor, J. L., Kutz, J. N. & Bialek, W. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl. Acad. Sci. U. S. A.* **113**, 3932–3937 (2016).

3. Tibshirani, R. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B Methodol.* **58**, 267–288 (1996).

4. Rudy, S. H., Brunton, S. L., Proctor, J. L. & Kutz, J. N. Data-driven discovery of partial differential equations. *Sci. Adv.* **3**, 1–7 (2017).

5. Kutz, J. N. & Brunton, S. L. Parsimony as the ultimate regularizer for physics-informed machine learning. *Nonlinear Dyn.* (2022) doi:10.1007/s11071-021-07118-3.

6. Raissi, M., Perdikaris, P. & Karniadakis, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686–707 (2019).

7. Karniadakis, G. E. *et al.* Physics-informed machine learning. *Nat. Rev. Phys.* **3**, 422–440 (2021).

8. Chen, Z., Liu, Y. & Sun, H. Physics-informed learning of governing equations from scarce data. *Nat. Commun.* **12**, 1–13 (2021).

9. Both, G. J., Choudhury, S., Sens, P. & Kusters, R. DeepMoD: Deep learning for model discovery in noisy data. *J. Comput. Phys.* **428**, 109985 (2021).

10. Both, G.-J. & Kusters, R. Fully differentiable model discovery. *arXiv* 1–11 (2021).

11. Rao, C., Ren, P., Liu, Y. & Sun, H. Discovering Nonlinear PDEs from Scarce Data with Physics-encoded Learning. *ICLR* 1–19 (2022).

12. Sun, F., Liu, Y., Wang, Q. & Sun, H. PiSL: Physics-informed Spline Learning for data-driven identification of nonlinear dynamical systems. *Mech. Syst. Signal Process.* **191**, 110165 (2023).

13. Thanasutives, P., Morita, T., Numao, M. & Fukui, K. I. Noise-aware physics-informed machine learning for robust PDE discovery. *Mach. Learn. Sci. Technol.* **4**, (2023).

14. Kaheman, K., Brunton, S. L. & Nathan Kutz, J. Automatic differentiation to simultaneously identify nonlinear dynamics and extract noise probability distributions from data. *Mach. Learn. Sci. Technol.* **3**, (2022).

15. Martius, G. & Lampert, C. H. Extrapolation and learning equations. *arXiv* 1–13 (2016).

16. Sahoo, S. S., Lantpert, C. H. & Martius, G. Learning equations for extrapolation and control.

35th Int. Conf. Mach. Learn. ICML 2018 **10**, 7053–7061 (2018).

17. Kim, S. *et al.* Integration of Neural Network-Based Symbolic Regression in Deep Learning for Scientific Discovery. *IEEE Trans. Neural Netw. Learn. Syst.* **32**, 4166–4177 (2021).
18. Costa, A. *et al.* Fast Neural Models for Symbolic Regression at Scale. *arXiv* (2021).
19. Zhang, M., Kim, S., Lu, P. Y. & Soljačić, M. Deep Learning and Symbolic Regression for Discovering Parametric Equations. *ICML* (2022).
20. Gandomi, A. H., Alavi, A. H. & Ryan, C. *Handbook of Genetic Programming Applications*. (Springer, 2015).
21. Koza, J. R. Genetic programming as a means for programming computers by natural selection. *Stat. Comput.* **4**, 87–112 (1994).
22. Bongard, J. & Lipson, H. Automated reverse engineering of nonlinear dynamical systems. *Proc. Natl. Acad. Sci. U. S. A.* **104**, 9943–9948 (2007).
23. Schmidt, M. & Lipson, H. Distilling free-form natural laws from experimental data. *Science* **324**, 81–85 (2009).
24. Chen, Y., Luo, Y., Liu, Q., Xu, H. & Zhang, D. Symbolic genetic algorithm for discovering open-form partial differential equations (SGA-PDE). *Phys. Rev. Res.* **4**, (2022).
25. Xu, H., Chang, H. & Zhang, D. DLGA-PDE: Discovery of PDEs with incomplete candidate library via combination of deep learning and genetic algorithm. *J. Comput. Phys.* **418**, 109584 (2020).
26. Xu, H., Zeng, J. & Zhang, D. Discovery of partial differential equations from highly noisy and sparse data with physics-informed information criterion. *Research* 1–30 (2023) doi:10.34133/research.0147.
27. Ma, W., Zhang, J., Feng, K., Xing, H. & Wen, D. Dimensional homogeneity constrained gene expression programming for discovering governing equations from noisy and scarce data. *arXiv* (2022).
28. Gerwin, D. Information processing, data inferences, and scientific generalization. *Behav. Sci.* **19**, 314–325 (1974).
29. Langley, P. Data-driven discovery of physical laws. *Cogn. Sci.* **5**, 31–54 (1981).
30. Falkenhainer, B. C. & Michalski, R. S. Integrating quantitative and qualitative discovery: the ABACUS system. *Mach. Learn.* **1**, 367–401 (1986).
31. Udrescu, S. M. & Tegmark, M. AI Feynman: A physics-inspired method for symbolic regression. *Sci. Adv.* **6**, (2020).
32. Udrescu, S. M. *et al.* AI Feynman 2.0: Pareto-optimal symbolic regression exploiting graph modularity. *Adv. Neural Inf. Process. Syst.* **2020-Decem**, 1–12 (2020).
33. Lee, J. & Leyffer, S. *Mixed Integer Nonlinear Programming*. vol. 154 (Springer Science & Business Media, 2011).

34. Cornelio, C. *et al.* Combining data and theory for derivable scientific discovery with AI-Descartes. *Nat. Commun.* **14**, 1777 (2023).

35. Cozad, A. Data-and theory-driven techniques for surrogate-based optimization. (2014).

36. Austel, V. *et al.* Globally optimal symbolic regression. *arXiv* (2017).

37. Cozad, A. & Sahinidis, N. V. A global MINLP approach to symbolic regression. *Math. Program.* **170**, 97–119 (2018).

38. Engle, M. R. & Sahinidis, N. V. Deterministic symbolic regression with derivative information: General methodology and application to equations of state. *AIChE J.* **68**, (2022).

39. Petersen, B. K. *et al.* Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients. *ICLR* (2019).

40. Kim, J. T., Kim, S. & Petersen, B. K. An interactive visualization platform for deep symbolic regression. *IJCAI* **2021-Janua**, 5261–5263 (2020).

41. Mundhenk, T. N. *et al.* Symbolic Regression via Neural-Guided Genetic Programming Population Seeding. *NeurIPS* **30**, 24912–24923 (2021).

42. Sun, F., Liu, Y., Wang, J. & Sun, H. Symbolic Physics Learner: Discovering governing equations via Monte Carlo tree search. *arXiv* (2022).

43. Du, M., Chen, Y. & Zhang, D. DISCOVER: Deep identification of symbolic open-form PDEs via enhanced reinforcement-learning. *arXiv* 1–15 (2022).

44. Tenachi, W., Ibata, R. & Diakogiannis, F. I. Deep symbolic regression for physics guided by units constraints: toward the automated discovery of physical laws. *arXiv* (2023).

45. Vaswani, A. *et al.* Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30**, (2017).

46. Biggio, L., Bendinelli, T., Neitz, A., Lucchi, A. & Parascandolo, G. Neural Symbolic Regression that Scales. *PMLR* (2021).

47. Valipour, M., You, B., Panju, M. & Ghodsi, A. SymbolicGPT: A Generative Transformer Model for Symbolic Regression. *arXiv* (2021).

48. Kamienny, P.-A., D'Ascoli, S., Lample, G. & Charton, F. End-to-end symbolic regression with transformers. *NeurIPS* 1–13 (2022).

49. Vastl, M., Kulhánek, J., Kubalík, J., Derner, E. & Babuška, R. SymFormer: End-to-end symbolic regression using transformer-based architecture. *arXiv* (2022).

50. Shojaee, P., Meidani, K., Farimani, A. B. & Reddy, C. K. Transformer-based Planning for Symbolic Regression. *arXiv* 1–15 (2023).