Corporate Governance
& Audit Archive Review

E-ISSN: 3006-144X
P-ISSN: 3006-1431

Finding The Governing Differential Equations In Free Form Through Data Analysis

Prof - Fardeen Ali Khan
University of Lahore

ABSTRACT | *This paper proposes a methodology for the identification of governing differential equations based on empirical data. This method does not require any a priori specification of the equation's constituent pieces to be carried out in order to be successful. Because of this, there is no requirement for a pre-determined specification of the components that are to be included. The strategy that has been suggested has as its primary objective the simplification of the incorporation of a dataset or a set of datasets that are relevant to a particular solution or a group of particular solutions of a differential equation. The problem-solving approach that has been suggested is developed with the purpose of being flexible enough to adapt to either scenario. The end result is a differential equation that has been written down in a way that is simple enough for folks to comprehend. The equation has been changed and tweaked in order to provide a more accurate representation of the particular solutions that have been presented. Improving one's understanding of differentiable data models is the primary objective of this research project. In the following step, the outputs that are produced by these models are then utilized as inputs inside the framework of genetic programming. This approach makes use of graphical representations to explain calculations by employing a wide variety of functions, parameters, and sometimes differential operators that are applied to functions. Our method, which makes use of recursive applications of automatic differentiation, has the capacity to investigate any arbitrary combination of operators without needing any input from the user. This is made possible by the fact that it employs automatic differentiation. This method makes it easier to simplify the design and evaluation of differential operators. It also makes the process more efficient. In addition, we describe a methodology for participating in active learning with the purpose of identifying and addressing flaws within the suggested governing equations. Our ultimate goal is to improve the system. The implementation of this measure was done so in order to improve the accuracy of the results.*

KEYWORDS | *governing differential equations, free form, data analysis, mathematical modeling, parameter estimation, system identification*

How to Cite

Prof - Fardeen Ali Khan. (2024). Finding the governing differential equations in free form through data analysis. Corporate Governance &Amp; Audit Archive Review, 2(01), 98–104. Retrieved from http://cgareview.com/index.php/Journal/article/view/33

Corporate Governance & Audit Archive Review

E-ISSN: 3006-144X
P-ISSN: 3006-1431

INTRODUCTION

Scientists in the modern day have access to a growing bank of laboratory data, which makes the investigation of new topics and the acquisition of new information much easier. However, when attempting to apply standard deductive approaches, which are normally the basis for constructing guiding principles, the enormous volume of data may present challenges. We provide a method for automating the study and manipulation of data in order to generate hypotheses, evaluate the validity of those hypotheses, and ultimately find new fundamental principles that control the natural world. This work was motivated by the idea of machine learning as a collaborator in scientific activities. Our approach is analogous to the growing corpus of published research on the application of differential programming and machine learning to the problem of solving differential equations (1, 2, 3, 4, 5) and to the task of determining the accuracy of previously established physical laws.

Our approach to finding free-form equations is differentiated from that of earlier research in that we take functions as input and apply a number of fundamental (algebraic and differential) operators on those functions. The expressions used to generate, edit, and evaluate real-time data are built with the help of these operators. Because of this, there is no longer a requirement to provide an a priori definition of any elements that might exist in the equation that is still to be specified. The finished product is a methodology that, despite offering a great deal of flexibility in the pursuit of differential equations, still creates a well-structured output (expressed as a graph) that is suitable for further inquiry and analysis. Despite the fact that it offers a significant deal of flexibility in the pursuit of differential equations, the final product is a methodology.

METHODOLOGY

The following is the official definition of the space that is symbolized by the symbol UL and which stands for the set of function pairs: Let there be a function known as u that is defined on a spatiotemporal domain of interest, which will be referenced as x R du from here on out. Let us assume that there is a function that is described using the real numbers. The provided criterion, which is represented by the notation L[u] = f (u, f) UL, is an example of a particular operator that works with the variables u and f. It has been determined that the functions that are being looked at meet the requirements of this criterion. The dimension denoted by the notation dst is present in the spatiotemporal domain x R. By utilizing the observation operator O: U D, one is able to gather discrete samples of functions, some of which may have stochastic properties. The samples of the ith function are represented by the data set Di, and the value dij indicates the jth sample of the ith function (for example, the digital readouts of sensors). For the sake of this investigation, we have compiled a set of N experiments, which we will refer to as "Di" where i equals 1, and each experiment consists of potentially distinct pairings (u, f). The differential equation L that regulates the physical systems that are of interest to us in this class is our goal, and our mission is to determine it. In addition, we make an effort to modify any parameters contained within the framework L that are unique to the data that was seen. In the current inquiry, we offer an approach, which can be seen displayed in Figure 1. This is done so that we can address this difficulty. In the next sections, we will examine this approach in greater depth and go through further specifics. To begin the study, we used explicit and differentiable models, which are indicated as Mij N,du i,j=1 and Mij M, to fit individual scalar output data sets dij. These models were used to find a good match for the data. The Gaussian processes3 and the fully-connected feedforward neural networks (15) models were investigated to see which one would be more appropriate. It is vital that the importance of ensuring that the selected model architectures, including the nonlinearities and kernels that are used, align with the desired differentiability properties that align with our assumptions regarding the qualities of unsupervised learning be acknowledged. This is because it is imperative that the desired differentiability properties align with our assumptions.

Following this, we will proceed to add into our dataset dij representations of differentiable functions, which will be denoted by the notation Mij. After that, differential equations are built by utilizing a genetic programming approach as the method of construction. In this investigation, we will look at a differential equation that stands alone and is referred to as an individual. This equation can be portrayed as a tree graph with the notation G(V, E). The graph that is depicted in this visual representation features parent vertices that represent n-ary operators that have been picked by the user from a library of operators. On the other hand, the leaves contained inside the network stand in for specific applications of fitted models. Our scope of coverage encompasses not only algebraic operators like addition and subtraction, but also differential operators like parentheses and partial derivatives, in addition to algebraic operators like addition and subtraction. The graph incorporates all of the leaves, or realizations, that correspond to a particular "experiment" that is designated by the variable i. These realizations are contained within the graph. The method of computing differential operators is made easier by the employment of PyTorch for automatic differentiation (19). The operator nodes used in this study utilise functions rather than arrays of integers as their argument type, which is a significant departure from the operator nodes used in previous studies. This extremely important distinction makes it possible to formulate G as a function that can be evaluated at any particular value of x, resulting in a residual that is denoted by the symbol r(x). In Figure 2, we illustrate a collection of visually represented expressions that our system is able to interpret in a relatively short amount of time.

In the end, the parameters that need to be calibrated are included in the function G as constant entities that are represented by leaf nodes. The method of calibrating makes use of black-box variational inference while operating inside a Bayesian framework. Our limited comprehension of the functions connected to the variables i's in G is the impetus for our decision to go with a non-factored multivariate Gaussian variational posterior q() and a flat prior for p(). A Gaussian distribution is used to mimic the probability distribution of a variable, and the estimated variance of the distribution is represented by the symbol r(x).

This particular research endeavor utilizes genetic programming as a method for investigating the hypothesis space that is associated with graphs (21). Deap (22) uses the evolutionary technique to generate, change, and combine the graphs. This process takes place over time. Participants compete in the fitness function L by making use of the evidence lower bound (ELBO) that is calculated at the inputs that are related to the Di's. An iterative process is what allows the evolutionary algorithm, often known as EA, to generate differential equations all by itself and then evaluate the results of those equations. As a consequence of this, the evolutionary algorithm (EA) generates a collection of prospective differential equations (G1, G2,...), which are then grouped according to the degree to which they are suitable and are designated using the notation L(G1, D), L(G2, D), and so on. The degree to which these equations, when applied in the form of fitted models (M), shed light on the gathered information is reflected in the rating that has been provided. Previous research (23, 24) has shown that the advocating of minimalism can be accomplished by using multi-objective optimization approaches that contain a supplementary fitness function that places an emphasis on parsimony. In the course of this investigation, we did not make use of this method; nevertheless, the choice not to do so was made due to the fact that it was not necessary to identify the precise equations that were operating in the background of our particular scenarios. If a person is capable of gathering more data, our method can be used in conjunction with an active learning loop to gradually acquire more samples. The goal is to improve the Di's by employing the acquisition function a(x) = r 2 (x) within the confines of the G1 framework. Because of this, it is necessary to conduct a more extensive examination of the areas in which the interpretation of the gathered data that is the least acceptable is most widespread. When a tolerance is chosen in such a way that a = arg maxxx a(x), one can say that the iterative process has reached the point where it can be said to have achieved convergence, and the resulting equation can adequately explain the observed data.

CONCLUSION

In this study, we have presented and demonstrated the effectiveness of a new approach in identifying governing differential equations with diverse structures from raw data pertaining to different ordinary differential equations. The aforementioned methodology was devised with the objective of identifying governing differential equations possessing arbitrary structure. Due to the used methodology, researchers are now afforded the opportunity to utilize an artificial intelligence-driven "research assistant," thereby enhancing their capacity to comprehend and engage with advanced concepts in the field of physics. Furthermore, the results produced by our approach, including of differential equations that are easily comprehensible to individuals, are compatible with the existing methodologies employed by scientists and engineers. These techniques encompass numerical simulation, theoretical analysis, and the utilization of physics-informed machine learning approaches.

References

- Ioannis G Tsoulos and Isaac E Lagaris. Solving differential equations with genetic programming. Genetic Programming and Evolvable Machines, 7(1):33–54, 2006.
- Dario Izzo, Francesco Biscani, and Alessio Mereta. Differentiable genetic programming. In European Conference on Genetic Programming, pages 35–51. Springer, 2017
- Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. PDE-net: Learning PDEs from data. arXiv preprint arXiv:1710.09668, 2017.
- Waldir Jesus de Araujo Lobão, Marco Aurélio Cavalcanti Pacheco, Douglas Mota Dias, and Ana Carolina Alves Abreu. Solving stochastic differential equations through genetic programming and automatic differentiation. Engineering Applications of Artificial Intelligence, 68:110–120, 2018.
- Josh Bongard and Hod Lipson. Automated reverse engineering of nonlinear dynamical systems. Proceedings of the National Academy of Sciences, 104(24):9943–9948, 2007.
- Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. science, 324(5923):81–85, 2009.
- Daniel L Ly and Hod Lipson. Learning symbolic representations of hybrid dynamical systems. Journal of Machine Learning Research, 13(Dec):3585–3618, 2012.
- Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proceedings of the National Academy of Sciences, 113(15):3932–3937, 2016.
- Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Machine learning of linear differential equations using gaussian processes. Journal of Computational Physics, 348:683– 693, 2017.
- Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. arXiv preprint arXiv:1711.10561, 2017.
- Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations. arXiv preprint arXiv:1711.10566, 2017.
- Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. Science Advances, 3(4):e1602614, 2017.
- Hayden Schaeffer. Learning partial differential equations via data discovery and sparse optimization. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 473(2197):20160446, 2017.

- Maziar Raissi and George Em Karniadakis. Hidden physics models: Machine learning of nonlinear partial differential equations. Journal of Computational Physics, 357:125–141, 2018.

- Maziar Raissi. Deep hidden physics models: Deep learning of nonlinear partial differential equations. The Journal of Machine Learning Research, 19(1):932–955, 2018.

- Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Multistep neural networks for data-driven discovery of nonlinear dynamical systems. arXiv preprint arXiv:1801.01236, 2018.

- Samuel Rudy, Alessandro Alla, Steven L Brunton, and J Nathan Kutz. Data-driven identification of parametric partial differential equations. SIAM Journal on Applied Dynamical Systems, 18(2):643–660, 2019.

- Seungjoon Lee, Mahdi Kooshkbaghi, Konstantinos Spiliotis, Constantinos I. Siettos, and Ioannis G. Kevrekidis. Coarse-scale PDEs from fine-scale observations via machine learning. arXiv preprint arXiv:1909.05707, 2019.

- Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

- Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In Artificial Intelligence and Statistics, pages 814–822, 2014.
- Wolfgang Banzhaf, Peter Nordin, Robert E Keller, and Frank D Francone. Genetic programming: an introduction, volume 1. Morgan Kaufmann San Francisco, 1998.
- Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. DEAP: Evolutionary algorithms made easy. Journal of Machine Learning Research, 13:2171–2175, jul 2012.
- Jeff Clune, Jean-Baptiste Mouret, and Hod Lipson. The evolutionary origins of modularity. Proceedings of the Royal Society b: Biological sciences, 280(1755):20122863, 2013.
- Adam Gaier and David Ha. Weight agnostic neural networks. arXiv preprint arXiv:1906.04358, 2019.
- Carl Edward Rasmussen and Christopher KI Williams. Gaussian processes for machine learning. MIT press Cambridge, MA, 2006.
- Anders Logg, Kent-Andre Mardal, and Garth Wells. Automated solution of differential equations by the finite element method: The FEniCS book, volume 84. Springer Science & Business Media, 2012.
- Steven Atkinson and Nicholas Zabaras. Structured bayesian gaussian process latent variable model: Applications to data-driven dimensionality reduction and high-dimensional inversion. Journal of Computational Physics, 383:166–195, 2019.