Data-driven partial derivative equations discovery with evolutionary approach

Mikhail Maslyaev, Alexander Hvatov, and Anna Kalyuzhnaya

ITMO University, Kronsersky pr. 49, 197101, St. Petersburg, Russia
alex_hvatov@corp.ifmo.ru

Abstract. The data-driven models are able to study the model structure in cases when a priori information is not sufficient to build other types of models. The possible way to obtain physical interpretation is the data-driven differential equation discovery techniques. The existing methods of PDE (partial derivative equations) discovery are bound with the sparse regression. However, sparse regression is restricting the resulting model form, since the terms for PDE are defined before regression. The evolutionary approach, described in the article, has a symbolic regression as the background instead and thus has fewer restrictions on the PDE form. The evolutionary method of PDE discovery (EPDE) is tested on several canonical PDEs. The question of robustness is examined on a noised data example.

Keywords: data-driven model · PDE discovery · evolutionary algorithms · symbolic regression.

1 Introduction

Data-driven algorithms are usually considered as the source of models when the connection between the data samples is not known a priori. There are various data-driven models. As an example, deep neural networks models, regression, combined evolutionary-based models [5] and other models and their combinations. However, most of the existing models are unsuitable for interpretation. Therefore, for cases, when the researcher is interested in the process of the model's decision making, other methods should be applied.

Data-driven algorithms are a solution for cases of systems, that we lack knowledge about. Nevertheless, in most cases raw observational data are available. The data-driven algorithms bring the ability to build the model for dynamical systems from time-series of data, received from in-field or laboratory observations. The development of the data-driven methodology of partial differential equations (PDE) derivation, combined with recent advances in technologies of measurements and probing, brings new opportunities for studying of metocean dynamic systems.

Sparse regression is considered to be the main tool for selection of the leading terms of the differential equations [7]. The applied regularization is based on the addition of the L1 norm of the calculated weights to the least-square expression.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_61

One of the most popular methods used in PDE discovery is the least absolute shrinkage and selection operator (LASSO). The main feature of LASSO is the ability to mutate the loss function. Zero weights are chosen for terms, that poorly fit the input data, and, therefore, identify the structure of the PDE.

Previously, the problem of the discovery of the differential equation structure has been developed in a number of papers. From derivation of systems of equations, defining physical laws, by means of symbolic regression [3, 9] to study dynamic systems, that are represented by a system of partial differential equations [2, 8]. Also, in the last years, the deep learning methods are becoming popular [1, 6].

The methods of PDE derivation, used in previous papers, usually utilize regression over the set of the pre-determined terms, that are usually comprised of different polynomial combinations of derivatives and functions. This limitation provides only the discovery of equations, that have a corresponding structure. The method, presented in this paper is referred below as EPDE. It is based on a combination of sparse regression and an evolutionary algorithm. The proposed way of calculation of terms weights values includes the application of linear regression over the non-normalized data for selected terms.

The paper is organized as follows, Section 2 describes the problem of data-driven PDE discovery in details. Also, in Section 2 dataset for experiments is described. Section 3 describes the data-driven PDE discovery algorithm based on evolutionary optimization. Section 4 is dedicated to the analysis of algorithm precision, stability, and robustness. Section 5 concludes the paper.

2 Problem statement and data acquisition

The developed EPDE algorithm is aimed at the derivation of the dynamic systems governing equation by time series, containing information about the studied variable (temperature, velocity, etc.). At first, the approach must be applied for test cases, including artificially created data, acquired from numerically solved equations to check the algorithm convergence. For further tests, a noise of selected magnitude can be added to data to investigate the reaction of the algorithm to it.

In this work, the algorithm was tested on the wave equation, Burgers and Korteweg-de Vries equations. They were solved numerically with the application of a finite-difference scheme to approximate time and spatial derivatives. For instance, the Crank-Nicolson method was utilized to solve the Burgers equation.

From the acquired field of equation solution, its time and spatial derivatives are calculated in order to be utilized further in regression. These derivatives are calculated by the finite-difference method, or from polynomial interpolation function depending on the presence of noise in the data.

After derivatives are obtained, it is possible to create vectors of spatial data for a specific time point. In the same time normalization of each of these time frames should be held. It can be done with the highest variable value for that

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_61

time point, or by time frame's L2-norm. Finally, data vectors are created by compositions of all time frames for the studied period.

Finally, the feature vectors $F(j)$ are formed by structures, such as the product shown in Eq. 1:

$$F(j) = \begin{bmatrix} (u'(t_1, x_0) * u_t(t_1, x_0))^N \\ \vdots \\ (u'(t_m, x_n) * u_t(t_m, x_n))^N \end{bmatrix} = U_x * U_t \tag{1}$$

On a balance, the data preparation step consists of representing data and it's spatial and time derivatives in vectors. After these steps, features are collected in forms, similar to Eq. 1, in order to perform the optimization procedure.

3 Algorithm description

The proposed algorithm includes two parts: the evolutionary algorithm that generates a small group of terms that are called individuals and sparse regression that allows choosing significant terms in the set of individuals.

To find values of the weights α, that is representing the systems PDE, it is possible to define the loss function (Eq. 2) in the following way, using the defined set of features and target vectors, created in the previous section:

$$\min_{\alpha} \left(\sum_{k=0}^{p} \|F_k \alpha - F_{target,k}\|_2^2 + \lambda \|\alpha\|_1 \right) \tag{2}$$

Where p is the number of features selected for the regression algorithm and λ represents a regularization parameter. This application of the regularized regression is not able to discover the true values of the weights due to the fact, that it uses normalized vectors of target and features. However, it is able to select leading ones with their sign. Due to the addition of L1-norm, the loss functions must be minimized, using optimization algorithms, that are able to work with non-differentiable functions, such as the subgradient method.

After the structure is found, the coefficients are defined with non-normalized data, i.e. features are constructed from their initial form and regression is used to find the final values coefficients. Usually, in regression all possible combinations [8] of the feature vectors Eq. 1 are chosen for minimization problem Eq. 2. Thus, the optimization problem complexity grows exponentially as the maximal order of the derivative increases. With the evolutionary algorithm, described below, one can use multiple reduced optimization problems instead of full regression on a complete terms library.

The second element of the EPDE method is the evolutionary algorithm. By its iterations, the evolutionary algorithm should be able to select and preserve the most appropriate elements of the resulting equation. Therefore, the sparse regression is done on every iteration of the evolutionary algorithm for every candidate in the population with a random selection of target among the set of terms.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_61

To initiate the method, it is required to create a population of individuals, represented by chromosomes, where each gene represents a combination of functions and their derivatives. An evolutionary algorithm is able to vary the chromosomes in two ways: crossover, that represents the exchange of corresponding genes between two individuals, and mutation, which involves random alteration of chromosomes genes. In the examined case, the mutation is held by the conversion of one term to the other randomly generated one.

Due to the specification of the task, every individual represents a specific case of the equation, having its own features matrix and the target vector. Vectors F(i), that compose the columns of the feature matrix S (Eq. 3), are created as a product of a randomly selected number of feature factors Eq. 1:

$$S = \begin{bmatrix} | & | & | & \\ F(1) & F(2) & F(3) & ... \\ | & | & | & \end{bmatrix} \tag{3}$$

It should be emphasized, that the number of feature vectors in Eq. 3 is the parameter of the evolutionary algorithm. The second remark is that, in contrast to the existing algorithms [7, 8, 4], the target feature is chosen randomly, whereas in the sparse-regression only cases time-derivative is used.

While mutation is usually applied to all individuals of the population, crossover occurs only between the most eligible of them. To select candidates for crossover, the fitness function should be introduced. For the task of partial differential equation derivation, it can be introduced by a norm of the difference between the target term and the expression with other ones i.e. regression error, calculated for all of the training data:

$$f_{fitness} = \frac{1}{\|F \cdot \alpha - F_{target}\|_2} \tag{4}$$

A manner of the populations participation in crossover should be defined before the initiation of an algorithm. The crossover procedure is schematically shown in Fig. 1.

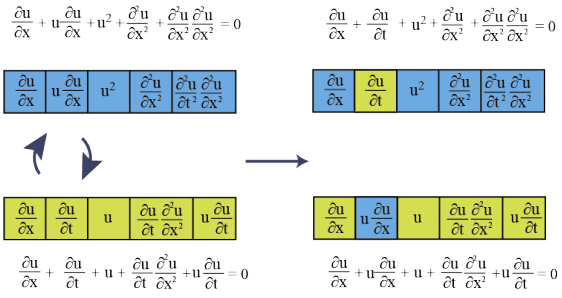

Fig. 1. An example of implemented crossover between two chromosomes, where each of them represent PDE

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_61

After the sparse regression application, one more regression step is required. It is initialized over the set of terms, selected by non-zero weights in the previous step. In this step, non-normalized fields of variables are used as a feature and target vectors. This approach is uncommon in general machine learning due to its limitations on variables of different scale, where the algorithm is not able to properly generalize data and discover a contribution of each feature. However, in this particular case, the structure of the an equation, represented by weights of features, is already known, and these variables must be evaluated according to their scale.

The described algorithm allows one to reduce regression space. Additionally, it allows to theoretically find ordinary differential equation instead of the PDE since target feature is not restricted by the highest time-derivative. This is required for potential one-dimensional static problems ODE discovery.

4 Validation

To analyze the algorithm performance, it is necessary to make sure, that it has the following qualities: stability, approximation, and convergence. These qualities are dependent on each other, and to prove them, it is enough to check, if any two of them are fulfilled. Due to the reasons of convenience, in the research, stability, and convergence of the algorithm are studied. Convergence of the PDE deriving algorithm manifests in the improvement of the quality of the algorithm with the reduction of a step of the grid, from that is adopts data. Stability can be proved by addition of the noise to the input PDE solution and test, how this corruption affects the structure of the resulting equation.

The algorithm has proved to be capable of discovering partial differential equations structure and calculating the values of weights for the selected terms for all of the studied equations.

The selected part of the solution matrix has influence over the results of regressions and, therefore, defines the equations structure. The results were tested on the parts of the matrix from 1.0 to 0.1 of its size. On the lesser sizes of the selected matrix part, especially for cases, when the selected part contains an only small part of the solution non-zero values, the algorithm can have difficulties, deriving wrong structures. The results of the matrix division are presented in Table 1.

Table 1. Discovered structure of the equations for different input matrix section.

Data part	Burger's correct	Burger's wrong	KdV correct	KdV wrong
0.9	$\frac{\partial u}{\partial t}$, $\frac{\partial^2 u}{\partial x^2}$, $u\frac{\partial^2 u}{\partial x}$	-	$\frac{\partial u}{\partial t}$, $\frac{\partial^3 u}{\partial x^3}$, $u\frac{\partial^2 u}{\partial x}$	-
0.7	$\frac{\partial u}{\partial t}$, $\frac{\partial^2 u}{\partial x^2}$, $u\frac{\partial^2 u}{\partial x}$	-	$\frac{\partial u}{\partial t}$, $\frac{\partial^3 u}{\partial x^3}$, $u\frac{\partial^2 u}{\partial x}$	-
0.5	$\frac{\partial^2 u}{\partial x^2}$, $\frac{\partial u}{\partial t}$	$\frac{\partial^2 u}{\partial x^2}$ $\frac{\partial u}{\partial t}$	$\frac{\partial u}{\partial t}$, $\frac{\partial^3 u}{\partial x^3}$, $u\frac{\partial^2 u}{\partial x}$	-
0.4	$\frac{\partial u}{\partial t}$, $\frac{\partial^2 u}{\partial x^2}$, $u\frac{\partial^2 u}{\partial x}$	-	$\frac{\partial u}{\partial t}$, $\frac{\partial^3 u}{\partial x^3}$, $u\frac{\partial^2 u}{\partial x}$	-
0.1	$\frac{\partial^2 u}{\partial x^2}$, $\frac{\partial u}{\partial t}$	$\frac{\partial^2 u}{\partial x^2}$ $\frac{\partial u}{\partial t}$	-	$\frac{\partial^2 u}{\partial x^2}$, $\frac{\partial^2 u}{\partial t^2}$

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_61

We note that for Table 1 different number of points was taken in order to check the performance of the algorithm. For Burger's equation, the 256x256 grid was taken whereas for the Korteweg-de Vries equation - 1024x1024 points.

The previously mention effect remained in this scenario: the algorithm only had issues in discovering the structure of the governing equation. For cases, when it succeeded, the true values of the weights were calculated correctly, even on minor parts of the equations solution matrix.

To check the evolutionary algorithm stability, the noise is added to the entire solution's field. It is added from a normally distributed random variable with zero mean value and dispersion taken as the fraction of maximal value. As the invariant noise measure, Eq.5 is used.

$$Q_{noise} = \frac{\|w_0 - \widetilde{w}\|_2}{\|w_0\|_2} * 100 \tag{5}$$

With w_0 in Eq.5 the initial (clean) solution field is designated, $\widetilde{w}$ is the field with noise added, $\|\cdot\|_2$ is the matrix's Frobenius norm.

For comparison, we take the latest supplementary code for the article [7] from GitHub repository. Same Burger's equation solution field and same noise procedure implementation were taken. It should be noted, that we compare "basic" versions of the algorithms. For the sparse regression more sophisticated derivative procedure and meta-parameter optimization for the regression algorithm could be implemented, which, definitely, increases the quality of both algorithms.

Polynomial derivatives procedure was utilized, also for the sparse regression improved ridge regression with $\alpha = 10^{-6}$ was taken.

For the Bruger's equation, after certain noise level limit $Q_{noise} \approx 0.11$ the classical algorithm loses the ability to discover the term $\frac{\partial^2 u}{\partial x^2}$ without an additional regression tuning. However, it is still able to catch the leading term. The EPDE, however, is able to find the right structure up to the $Q_{noise} \approx 0.11$.

As seen the evolutionary approach allows one to extend the noise level which is allowed for all terms of the initial equation discovery. The term coefficients discovery precision is increased, which leads to more stable equation discovery and allows one to discover the equations in a more robust way.

5 Conclusions and discussion

In the paper evolutionary approach for PDE discovery is described. In contrast to the existing algorithms based on the regression on a complete terms library it has the following advantages:

- Regression is done on a reduced space, i.e. only a small amount of features is taken for the regression;
- More flexible features choice allows to obtain wider space of possible differential operators;
- No restriction on the target function is allowing to obtain more sophisticated forms of differential operators including ODEs;

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_61

The possible disadvantages could be:

- Possible extended computation time due to the stochastic process of the initial population initialization, population crossover and mutation;
- Additional procedures are required in order to maintain the robustness of the algorithm, i.e. in order to obtain the same model for the data of the same origin;

The proposed method can be considered as a base point for the data-driven PDE discovery with an evolutionary approach. In the article, the main stages of the methods are shown. Every stage could be improved, for example, a more sophisticated grid function differentiation method could be taken to increase precision and stability. Also, more advanced evolution methods could be used in order to increase computation efficiency and stability.

References

1. Berg, J., Nyström, K.: Data-driven discovery of pdes in complex datasets. Journal of Computational Physics **384**, 239–252 (2019)
2. Bongard, J., Lipson, H.: Automated reverse engineering of nonlinear dynamical systems. Proceedings of the National Academy of Sciences **104**(24), 9943–9948 (2007)
3. Gray, G.J., Murray-Smith, D.J., Li, Y., Sharman, K.C., Weinbrenner, T.: Nonlinear model structure identification using genetic programming. Control Engineering Practice **6**(11), 1341–1352 (1998)
4. Kondrashov, D., Chekroun, M.D., Ghil, M.: Data-driven non-markovian closure models. Physica D: Nonlinear Phenomena **297**, 33–55 (2015)
5. Kovalchuk, S.V., Metsker, O.G., Funkner, A.A., Kisliakovskii, I.O., Nikitin, N.O., Kalyuzhnaya, A.V., Vaganov, D.A., Bochenina, K.O.: A conceptual approach to complex model management with generalized modelling patterns and evolutionary identification. Complexity **2018** (2018)
6. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations. arXiv preprint arXiv:1711.10566 (2017)
7. Rudy, S.H., Brunton, S.L., Proctor, J.L., Kutz, J.N.: Data-driven discovery of partial differential equations. Science Advances **3**(4), e1602614 (2017)
8. Schaeffer, H.: Learning partial differential equations via data discovery and sparse optimization. Proc. R. Soc. A **473**(2197), 20160446 (2017)
9. Winkler, S., Affenzeller, M., Wagner, S.: New methods for the identification of nonlinear model structures based upon genetic programming techniques. SYSTEMS SCIENCE-WROCLAW- **31**(1), 5 (2005)

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_61