Uncovering differential equations from data with hidden variables

Matthieu Jonckheere[1], Vincent Lefieux[2], Ezequiel Smucler[3,4], Agustín Somacal[4], and Dominique Picard [5]

[1]Instituto de Calculo-CONICET, Universidad de Buenos Aires
[2]RTE
[3]Universidad Torcuato Di Tella
[4]Aristas S.R.L.
[5]Laboratoire de Probabilités et Modèles aléatoires Université Paris VII

May 2019

Abstract

Finding a set of differential equations to model dynamical system is a difficult task present in many branches of science and engineering. Recently, Rudy et al. 2017, Brunton, Proctor, and Kutz 2016 and Quade et al. 2018 developed algorithms based on sparse regression techniques to automate the discovery of equations compatible with the observed measurements using only observed data. Unfortunately, their methods do not work in the presence of latent variables, that is, when some of the variables of the dynamical system cannot be observed. In this paper we propose a method hat can recover dynamical systems with latent variables from time series measurements. The main idea of our method is to regress, using a Lasso type estimator, a temporal derivative of the observable variables on a dictionary of functions that includes lower order temporal derivatives of the observable variables. Extensive numerical studies show that our method can recover useful representations of the dynamical system that generated the data even when some variables are not observed. Moreover, being based on solving a convex optimization problem, our method is much faster than competing approaches based on solving combinatorial problems. Finally, we apply our methodology to a real data-set of temperature time series.

1 Introduction

Many branches of science are based on the study of dynamical systems. Examples include meteorology, biology and physics. The usual way to model deterministic dynamical systems is by using (partial) differential equations. Typically, differential equations models for a given dynamical system are derived using a-priori insight into the problem at hand; then the model is validated using empirical observations. In an era in which massive data-sets pertaining to different fields of science are widely available, an interesting problem is whether it is possible for a *useful* differential equations model to be learnt directly from data, without any major modelling effort required by the researcher.

Our goal in this paper is to develop a general methodology for building such differential equations models in contexts in which not all relevant variables are observed, that is, in cases in which the main variable of interest depends on other variables of which no measurements are available. As a concrete example, consider the following problem. RTE, the electricity transmission system operator of France, uses high level simulations of hourly temperature series to study the impact different climate scenarios have on electricity consumption, and hence on the French electrical power grid. The underlying simulations are based on the Navier-Stokes equations and include as variables the wind velocity, density, pressure, etc. Moreover, the resulting dynamic system is known to be chaotic, see Kida, Yamada, and Ohkitani 1989.

In the case in which all relevant variables are observed, the problem of recovering the differential equations that generated the observed data has been studied recently. The papers Brunton, Proctor, and Kutz 2016, Rudy et al. 2017 and Quade et al. 2018, which motivated our research, developed an approach that consists on linking the dynamical system discovery problem to a statistical regression problem, making it scalable and easy to apply in different contexts. The main idea of their approach is to consider a set of differential operators (possibly in both time and space if appropriate), discretize them, for example by using finite differences, and then regress the outcome of interest on the discretized differential operators. By solving the regression problem using an ad-hoc thresholded least-squares algorithm, they are able to build sparse, interpretable models, that use mostly low order derivatives. They explored the applicability of their method on simulated data, but only in situations in which all the variables of the simulated models are observed. We provide further details of their approach in Section 3. We highlight that in place of the thresholded least-squares algorithm, any other regularised linear regression estimator could be used.

To accommodate the possibility of latent variables, we note that, for a large class of dynamical systems, it is possible to reconstruct a trajectory (equivalent to the original one) given only one of the model variables, using its delayed lags. See for instance Takens 1981. Based on this result, we propose to augment the methodology developed in Brunton, Proctor, and Kutz 2016, Rudy et al. 2017 and Quade et al. 2018 by including higher order time derivatives, in order to tackle situations in which not all relevant variables are observed. We estimate the coefficients of the dynamical system using the Lasso estimator: an ℓ_1-regularised least squares regression estimator (Tibshirani 2011). We choose to use the Lasso due to its simplicity, the abundance of theoretical guarantees on its performance (Hastie, Tibshirani, and Wainwright 2015) and the availability of efficient algorithms to solve the convex optimisation problem that defines the estimator. After estimating the regression coefficients, we build a forecasting method by integrating the retrieved differential equation using the Runge-Kutta-4 method. We call our methodology pdefind-latent.

Most related to our approach are the proposals of Bongard and Lipson 2007a and Mangiarotti et al. 2012, whose method, GPoMo, addresses the recovery problem via a combinatorial search between a predefined set of polynomial functions of the observable variables. The method proceeds by iteratively choosing a family of combination of terms that are able to better reproduce the phase diagram of the system. Finally, it returns the set of models that were best. The authors also discuss the ability of their algorithm to find equations able to capture the dynamics in the case in which only some variables are observed. However, we will show that this approach tends to be slow and does not scale well to large problems. Other approaches for learning dynamical systems from data are available in the literature, such as those based on symbolic regression (Bongard and Lipson 2007b; Schmidt and Lipson 2009), also have the drawback of being too computationally expensive.

We conduct extensive numerical experiments comparing the performance of the algorithm proposed in this paper and the GPoMo methodology. Our results can be summarised as follows:

1. When all variables are observed, both methods have similar performance, with pdefind-latent outperforming GPoMo.

2. When only one variable of the system is observed and no noise is added both methods have similar performance, being able to recover dynamical systems with phase diagrams similar to those of the ground truth. Again in this case the performance of pdefind-latent is slightly superior.

3. The performance of both methods deteriorates when only noisy measurements are available.

4. Our method is orders of magnitude faster than GPoMo. This can be explained by the fact that GPoMo does not solve a convex optimization problem, but rather conducts a combinatorial search in the space of polynomial models.

The rest of this paper is organised as follows. In Section 3 we describe our methodology in detail and provide an overview of the GPoMo method. Section 4 presents the results of our experiments. In particular, in Section 4.1, we compare the performance of these methods in recovering differential equations using empirical data in the case in which all relevant variables are observed. In Section 4.2 we compare them in the harder case in which at least one relevant variable driving the dynamical system is latent. We apply both methods to the RTE data in Section 4.3. Finally in Section 5 we discuss future work and possible extensions.

2 Differential equations recovery as a linear regression problem

We consider dynamical systems represented by functions $f_1(x, y, z, t) \dots f_H(x, y, z, t)$ satisfying a set of differential equations of the form

$$\mathbb{D}\mathbf{f} = \mathbf{U}(\mathbf{f}, \mathbb{E}\mathbf{f}), \tag{1}$$

where $\mathbf{f} = (f_1, \dots, f_H)$, $\mathbb{D}, \mathbb{E}$ are differential operators in, possibly, both spatial (x, y, z) and temporal variables (t) and $\mathbf{U} : \mathbb{R}^H \to \mathbb{R}^H$ is an unknown map. Suppose we have a multidimensional set of time series corresponding to observations of the dynamical system measured at regular intervals (and possibly also over a regular spatial grid). More precisely suppose we have measurements on an $M \times M \times M \times T$ grid, that is, we observe

$$f_h(x_i, y_j, z_k, t_l) \quad i, j, k \in \{1, \dots, M\}, l \in \{1, \dots, T\}, h \in \{1, \dots, H\}. \tag{2}$$

An example of a dynamical system we will study in this paper is the classical Lorenz system (Lorenz 1963). The Lorenz system is a simplified model for atmospheric convection. The system is given by

$$\begin{aligned} \frac{df_1(t)}{dt} &= \alpha \{f_2(t) - f_1(t)\} \\ \frac{df_2(t)}{dt} &= f_1(t) \{\rho - f_3(t)\} - f_2(t) \\ \frac{df_3(t)}{dt} &= f_1(t) f_2(t) - \beta f_3(t), \end{aligned} \tag{3}$$

for constants α, β, ρ. This system can be written in the form (2) by taking $\mathbf{f} = (f_1, f_2, f_3)$, $\mathbb{D} = (d/dt, d/dt, d/dt)$ and $\mathbf{U} = (U_1, U_2, U_3)$ where $U_1(v_1, v_2, v_3) = \alpha(v_1 - v_2)$, $U_2(v_1, v_2, v_3) = v_1(\rho - v_3) - v_2$ and $U_3(v_1, v_2, v_3) = v_1 v_2 - \beta v_3$. For certain values of the parameters α, β, ρ, the system is known to have chaotic solutions.

Our objective is to find some system of differential equations that can explain the behaviour of the measurements. As discussed in the introduction, we follow the approach outlined by Brunton, Proctor, and Kutz 2016, Rudy et al. 2017 and Quade et al. 2018. Their approach works by choosing a large dictionary of functions and regressing discretizations of

$$\frac{\partial f_1}{\partial t}, \dots, \frac{\partial f_H}{\partial t}$$

on the dictionary. The dictionary in question can be formed, for example, by collecting polynomial powers of f_h, $h = 1, \dots, H$, spatial derivatives of $f_1 \dots, f_H$ and trigonometric functions of t. A concrete simple example of such a dictionary in the case in which $H = 1$ is the following:

$$\mathcal{A} = \left\{ x, y, z, x^2, y^2, z^2, \frac{\partial f(x, y, z, t)}{\partial x}, \frac{\partial f(x, y, z, t)}{\partial y}, \frac{\partial f(x, y, z, z)}{\partial z}, \sin(t) \right\}.$$

Of course in practice all derivatives are replaced by the corresponding finite differences taken from the measurements represented in f. Further details on this point will be provided shortly. Having chosen a dictionary, we let $\mathcal{A} = (A_1, \dots, A_p)$ be the vector collecting all members of the dictionary.

Using the observations of the dynamical system, a regression model can be fitted to find the combination of the elements of the dictionary of functions that adequately explains the behaviour of $\frac{\partial f_1}{\partial t}, \dots, \frac{\partial f_H}{\partial t}$. That is, we look for a vector of regression coefficients $\mathbf{c} = (c_1, \dots, c_p)$ such that for all $h = 1, \dots, H$

$$\frac{\partial f_h(x, y, z, t)}{\partial t} \approx \sum_{i=1}^{p} c_{i,h}.A_i(x, y, z, t). \tag{4}$$

In Brunton, Proctor, and Kutz 2016, Rudy et al. 2017 and Quade et al. 2018 the authors propose to use an ad-hoc linear regression estimator based on iteratively thresholding the least-squares estimator. Through extensive numerical experiments, they show that this methodology is able to learn systems of partial differential equations that adequately model the dynamical system that generated the data. Unfortunately, if some variables are latent, that is, if one is unable to measure at least one of $f_1, \dots, f_H$, the approach described above breaks down. Next, we describe a way of extending this methodology to deal with the case in which some variables are latent.

2.1 Our proposal

It is known that, for a large class of dynamical systems, it is possible to reconstruct a trajectory (equivalent to the original one) given only some of the model variables, using its delayed lags. See Takens 1981 for a mathematical formulation. In the same direction, the differential embedding method of Packard et al. 1980 deals with the problem of single time series reconstruction by adding higher order derivatives. Based on this result, we propose to augment the methodology developed in Brunton, Proctor, and Kutz 2016, Rudy et al. 2017 and Quade et al. 2018 by including higher order derivatives.

More precisely, our approach works by expanding the dictionary of functions by including higher order derivatives of the observable variables. On top of including linear and non-linear functions of x, y, z, t, polynomial powers and spatial derivatives of the observable f_hs, we add higher order time derivatives of the observable f_hs. For example, if only f_1 is latent, our dictionary might include

$$\frac{\partial^2 f_2}{\partial t^2}, \dots, \frac{\partial^2 f_H}{\partial t^2}.$$

A regression model can now be built to find combination of the elements of the dictionary of functions that adequately explains the behaviour of some time derivative that is of higher order than those included in the dictionary. That is, we look for vector of regression coefficients $\mathbf{c}_h^* = (c_{h,1}, \dots, c_{h,p})$ such that, for an appropriately chosen n, and for all h such that f_h is observable,

$$\frac{\partial^n f_h(x, y, z, t)}{\partial t^n} \approx \sum_{i=1}^{p} c_{i,h}^* A_i(x, y, z, t). \tag{5}$$

In our analysis the response variable is always taken as the time derivative of order n, where n is equal to the order of the derivative of highest order in the dictionary, plus one.

The regression model has to be learned using the available data. This regression problem could be solved in principle using least-squares. However, the ordinary least-squares regression estimator is ill-defined in cases in which the number of predictor variables p is larger than the number of observations. Since the analyst is usually uncertain about the number of elements in the dictionary needed to adequately model the system of interest, the method used to solve the regression problem at hand should allow for large number of predictor variables (possibly larger than the number of observations) and automatically estimate sparse models, that is, generate accurate models that only use a relatively small fraction of predictor variables. The Lasso regression technique (**??**) is perfectly suited for this task. The Lasso is a ℓ_1-regularized least-squares regression estimator, defined as follows. For $h = 1, \ldots, H$ such that f_h is observable we let

$$\mathbf{c}_h^* = \arg\min_{\mathbf{c}\in\mathbb{R}^p} \sum_{i,j,k,l} \left(\frac{\partial^n f_h(x_i, y_j, z_k, t_l)}{\partial t^n} - \sum_{g=1}^{p} c_{h,g} A_g(x_i, y_j, z_k, t_l) \right)^2 + \lambda \|\mathbf{c}_h\|_1, \tag{6}$$

where $\lambda > 0$ is a tuning constant, measuring the amount of regularization. It can be shown (Hastie, Tibshirani, and Wainwright 2015) that the ℓ_1 penalty encourages sparse solutions and that the larger $\lambda > 0$ the sparser the solution vector $\mathbf{c}_h^*$ will be. In practice, λ is usually chosen by cross-validation. Note that in (**??**), each regressor, that is, each element of the dictionary $\mathbf{A}$, may vary at different scales. In general, the Lasso will penalize more severely those variables that vary on a smaller scale, which is undesirable. In order to avoid this, a z-score normalization is performed prior to the Lasso fit. Incidentally, this results in better numerical stability. At the end, the normalization is undone, so that the regression coefficients are in the original scale. Note that any other sparse regression technique could have been used to estimate the coefficients. We prefer the Lasso due to its simplicity and the wide availability of efficient algorithms to compute it. See for example Friedman et al. 2007.

The main assumption behind this methodology is that the dynamical system that generated the data at hand can, in reality, be at least approximated using a sparse model. This hypothesis is known to hold for several dynamical systems of interest in different fields of science. See Quade et al. 2018. If the hypothesis holds, we can expect the Lasso estimates to select only a few elements of the dictionary, namely, those that do a good job at explaining variations in the response variable (Hastie, Tibshirani, and Wainwright 2015).

2.1.1 Implementation details

To form the dictionary of functions our method requires, we discretize all differential operators involved in building the dictionary using finite differences. Since the discretised derivative

$$\frac{f(x, y, z, t + dt) - f(x, y, z, t)}{dt}$$

makes it unclear to what time (t) we should assign the derivative we preferred central finite differences calculated using the neighbouring function values

$$\frac{f(x, y, z, t + dt) - f(x, y, z, t - dt)}{2dt}.$$

The analogous statement holds for spatial derivatives. Higher order derivatives are dealt with similarly. In our analysis, the discretization parameters (dt, dx, dy, dz) were obtained from the minimum space and time intervals given by the data.

In our implementation, the penalty constant λ is chosen by 20-fold cross-validation, seeking to minimize the prediction mean square error.

The comparison with the *GPoMo* method was performed using the implementation of the method available through th GPoMo function provided by the gPoM `R` package, using the following set-up,

```
out <- GPoMo(data=time_series, tin=time, dMax=2, nS=c(1, 1, 1), IstepMin =10,
    IstepMax=15000, nPmin=5, nPmax=20, method='rk4').
```

The maximum polynomial degree $dMax$ was set to 2 in the case of full information (that is, no latent variables) and 3 when only one variable was observed.

For our method, we create the same dictionaries as the ones created with *GPoMo*, in the case of full information it will be polynomials of order 2 of the variables,

$$\{1, x, y, z, xy, xz, yz, x^2, y^2, z^2\},$$

and in the case of latent variables, derivatives until order two and polynomials of order three

$$\{1, x, \frac{\partial x}{\partial t}, \frac{\partial^2 x}{\partial t^2}, x\frac{\partial x}{\partial t}, \frac{\partial x}{\partial t}\frac{\partial^2 x}{\partial t^2}, x^2, \frac{\partial x}{\partial t}^2, \frac{\partial^2 x}{\partial t^2}^2, \dots\}.$$

3 Numerical experiments

To evaluate the performance of the proposed method and compare it with GPoMo algorithm we used 18 dynamical systems provided by the GPoM `R` package [1]. For all these dynamical systems, we use the exact same configuration (coefficients, time duration, etc.) used in the GPoMo package. All of them have three variables $\{x, y, z\}$. Among the systems studied are the Lorenz system

$$\begin{aligned}\frac{dx}{dt} &= 10(y - x)\\ \frac{dy}{dt} &= x(28 - z) - y\\ \frac{dz}{dt} &= xy - \frac{8}{3}z,\end{aligned} \tag{7}$$

and the Rössler (1976) system (**??**), a chaotic system extensively used by Mangiarotti et al. 2012 to show how their method succeds in finding addecuate models

$$\begin{aligned}\frac{dx}{dt} &= -z - y\\ \frac{dy}{dt} &= 0.52y + x\\ \frac{dz}{dt} &= 2 - 4z + xz, .\end{aligned} \tag{8}$$

We evaluated the reconstruction power of our method and of GPoMo in two situations:

- when all relevant variables are observed.
- when only one variable of the system is observed.

Evaluating the equations found by the methods is difficult, due to the chaotic nature of the systems considered and the lack of a ground truth equation to compare with in the case in which only one variable is observed. This difficulty makes any metric that compares the prediction accuracy of a method against the true series (like the mean square error) unreliable. However, visual inspection of the estimated phase diagrams can reveal which models are reasonable or not, by noticing that the trajectories of two systems that behave similarly will produce series that evolve in the same regions of the phase diagram. This insight led us to propose the Wasserstein distance, also called earth-mover distance, (Villani 2008), to compare the phase diagram obtained by each method to the true phase diagram. Given a trajectory over the phase space we can collect all its points and think of them as drawn from a certain distribution characteristic of the system that generated those data points. Let ρ_{true} be that distribution and ρ_{method} the one obtained through proper integration of the equation found by some method. The earth-mover distance $emd(\rho_{true}, \rho_{method})$ between ρ_{true} and ρ_{method} is computed as the infimum over all couplings $\pi(X, Y)$ of ρ_{true} and ρ_{method} of $E_\pi^{1/2}\left[(X - Y)^2\right]$. We evaluate the system recovered by a given method by computing

$$emd(\rho_{true}, \rho_{method})$$

[1] Systems: Nosé-Hoover 1986, Genesio-Tesi 1992, Sprott-F, H, K, O, P, G, M, Q and S, Lorenz 1963 and 1984, Burke and Shaw 1981, Rosseler 1976, Chlouverakis-Sprott 2004, Li 2007, Cord 2012

where smaller values are preferable.

To study the impact noisy measurements have on pdefind-latent and on GPoMo, we introduced various levels of random noise to the dynamical systems. More precisely, for each time point and each variable in the system, we add to the true data a zero mean normal random variable with standard deviation equal to $k\sigma$, where σ is the sample standard deviation of the variable in question and

$$k \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5\},$$

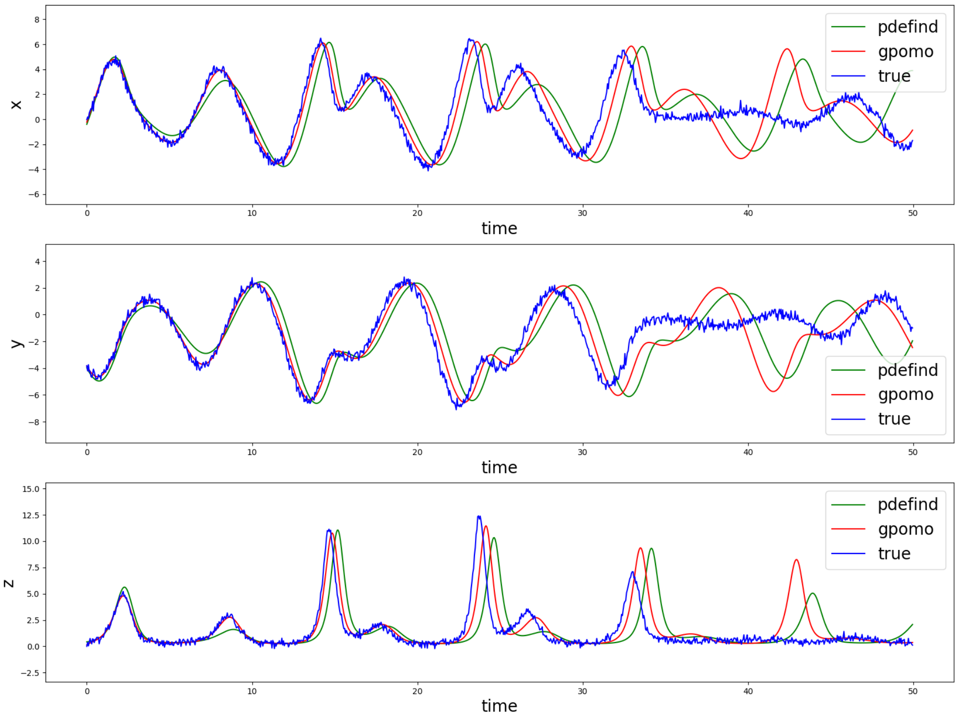
so that larger values of k imply more noise is being added to the system. Of particular interest is to understand how the performance of the methods deteriorates as the amount of noise added increases.

4 Results

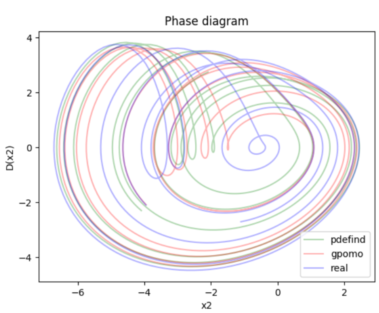
4.1 Full information

When all the information is available and no noise is added, both methods are capable of recovering the dynamical system. This can be seen in Figure 1 where the predictions using each of the methods follow the true dynamics without much variation for 20 time units. Moreover, when looking at the phase diagram, it is clear that both can reproduce the particular dynamics of each variable. In Figure 2, we can see that both methods fail at following the true series trend due to the added noise ($k = 0.3$). GPoMo, in particular, is not able neither to reproduce the phase diagram, while the pdefind-latent method succeeds in obtaining shapes similar to those of the original system.

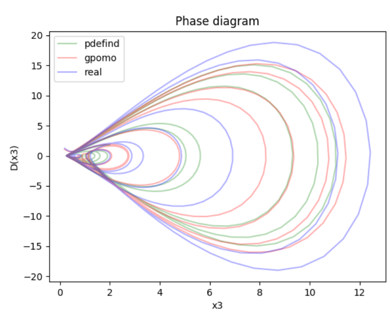
After applying GPoMo and pdefind-latent to all the dynamical systems provided by GPoMo package (under different noise conditions) we calculated the emd distance over the resulting phase diagrams and plotted the corresponding distributions to compare the performance of both methods. In the violin plots of Figure 3 we can see that both methods perform similarly with and without noise, with pdefind-latent's metric distribution nearer to zero meaning it's phase portraits are more similar to the true ones than the ones obtained with GPoMo. When noise is added the performance of both methods progressively deteriorates as k increases.



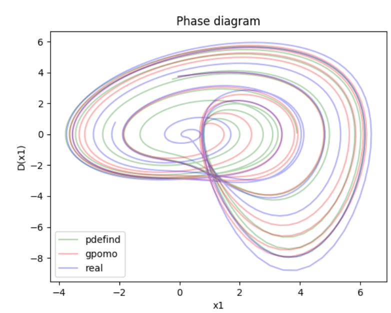
(a)



(b)

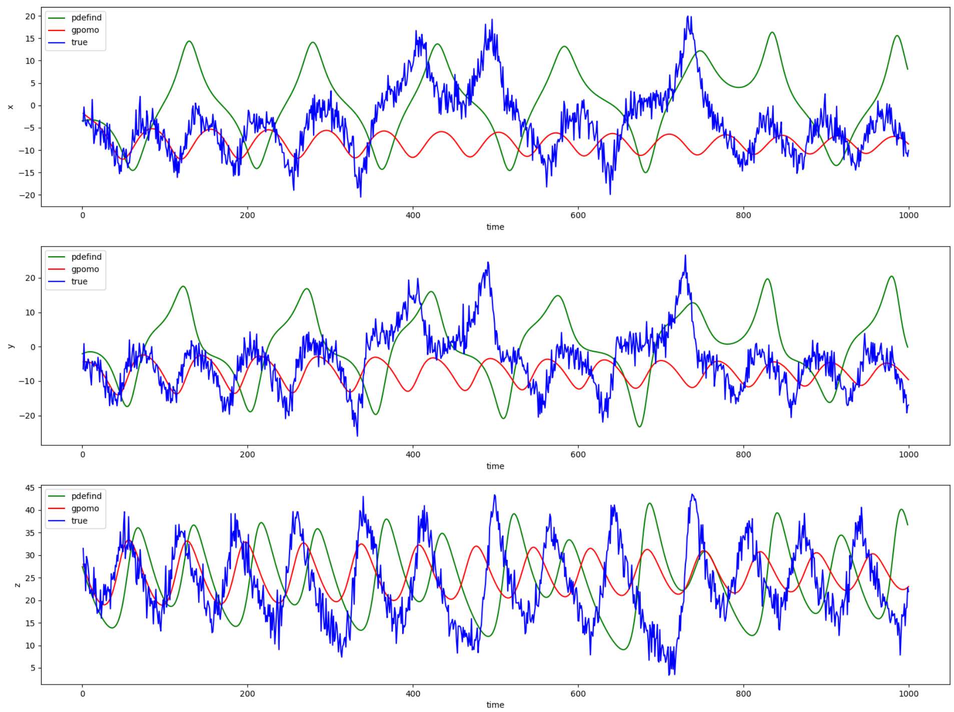


(c)

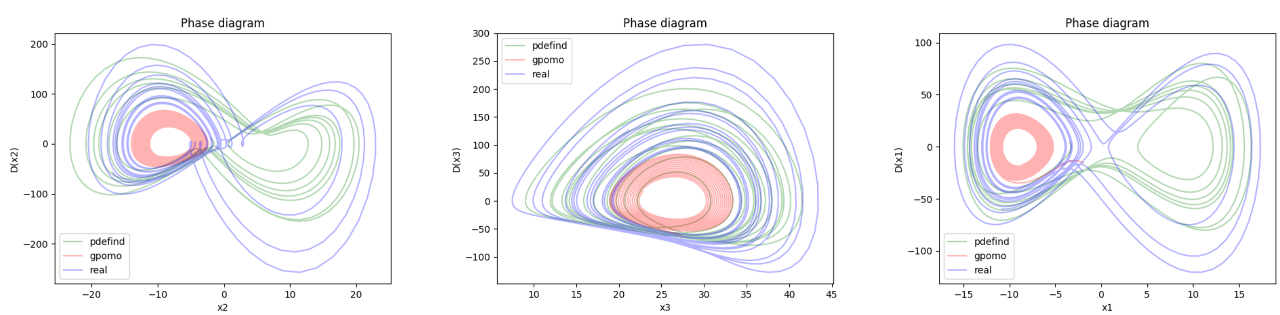


(d)

Figure 1: *For the Rosseler system (a) the true time series (blue) with $k = 0.1$ level of noise, and integrated time series using the models fitted with pdfind-latent (green), GPoMo (red). In (b, c, d) the true phase diagram together with the ones obtained with both methods.*



(a)



(b) *(c)* *(d)*

Figure 2: *For the Lorenz 1963 system (a) the true time series (blue) with $k = 0.3$ level of noise, and integrated time series using the models fitted with pdfind-latent (green), GPoMo (red). In (b, c, d) the true phase diagram together with the ones obtained with both methods.*

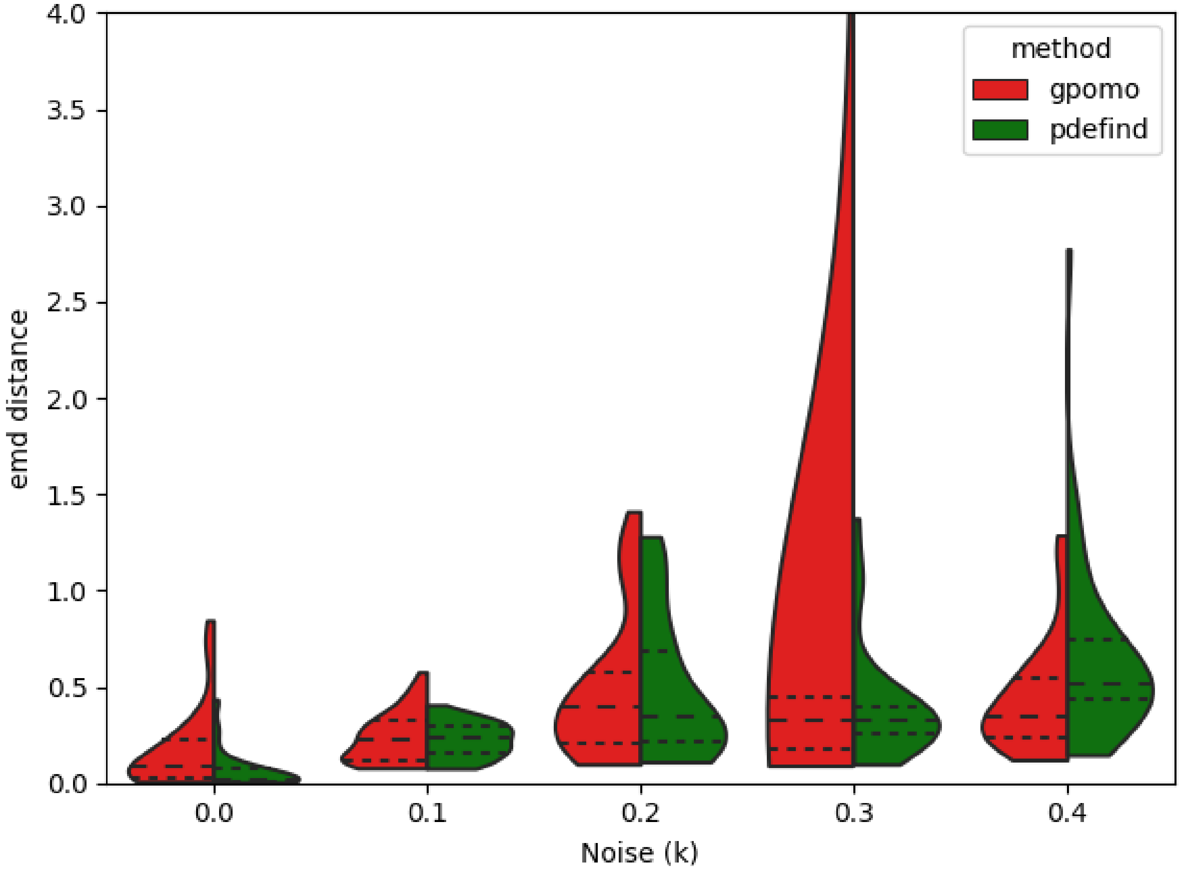


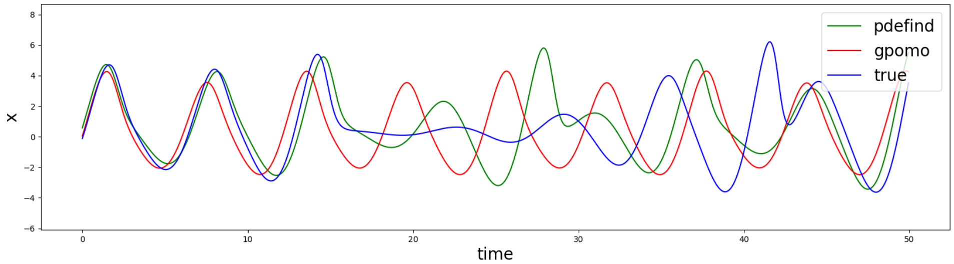
Figure 3: *Violin plot showing the behaviour of the methods through the emd metric over the phase diagrams. We applied GPoMo and pdefind-latent to all the dynamical systems provided by GPoMo package and calculated the emd distance over the phase diagrams giving the resulting distributions. The black horizontal lines in each distribution show the quartiles of the distributions. We can see that pdefind-latent has better performance when there is no noise as the values are nearer to zero and when noise is present both methods diminish their power.*

4.2 Latent variables

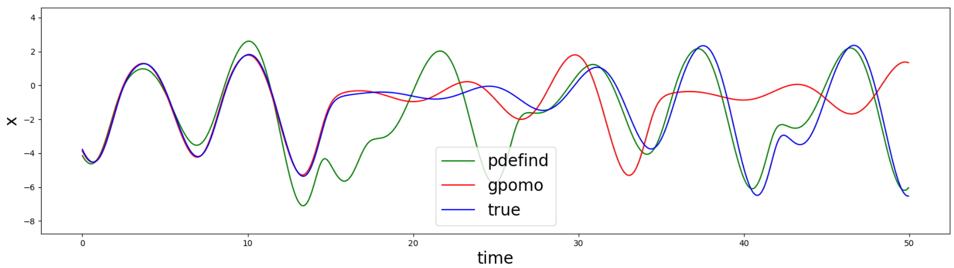
In this section we explore the feasibility of using higher order time derivatives to tackle the lack of information by using the same dynamical systems but simulating the situation in which only one of the three variables is observed. For both methods we fixed the maximum polynomial order in 3 and choose derivatives up to the $3th$ order.

When no noise is added, both methods are capable in most of the cases to recover an equivalent dynamical system (Figure 5). This can be seen in Figure 4 where predictions and phase diagrams reproduce the same behaviour as the real series (specially for variable y). In the case of variable z of the Rosseler system, we see that GPoMo method diverges quickly while our method maintains the geometry of the phase dynamics for a longer time, although it also eventually diverges.

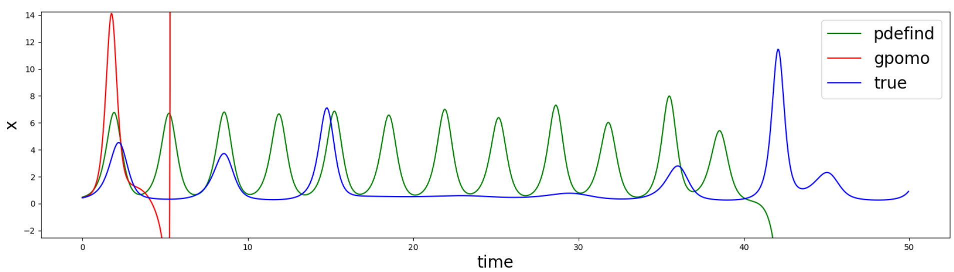
A full comparison of the methods can be seen in Figure 6, where for each noise level we plot the distribution of *emd* distances, over the phase diagrams. The distributions where generated by applying to each of the three variables of all the available dynamical systems both methods. The pdefind-latent method produces models that are better at capturing the geometry of the true phase diagram trajectories, as the *emd* metric obtained is mostly lower than GPoMo. When noise is added, both methods lose their power.



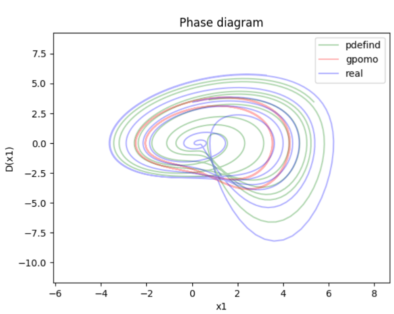
(a)

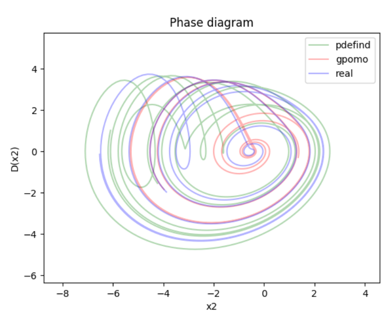


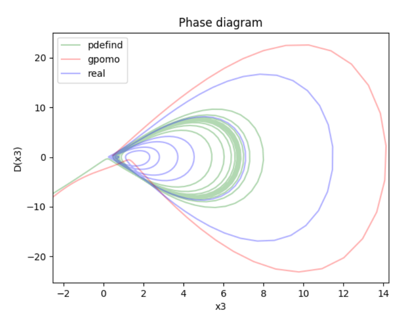
(b)



(c)

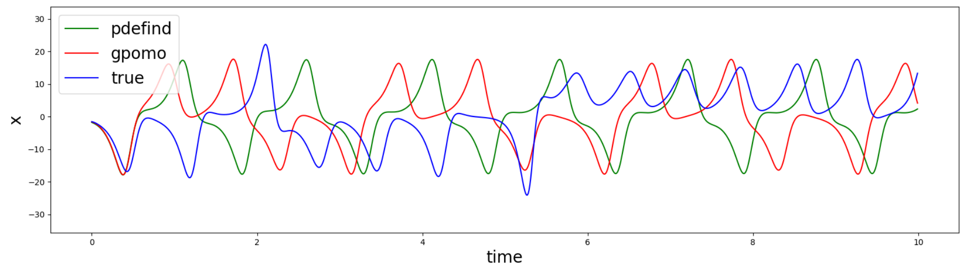




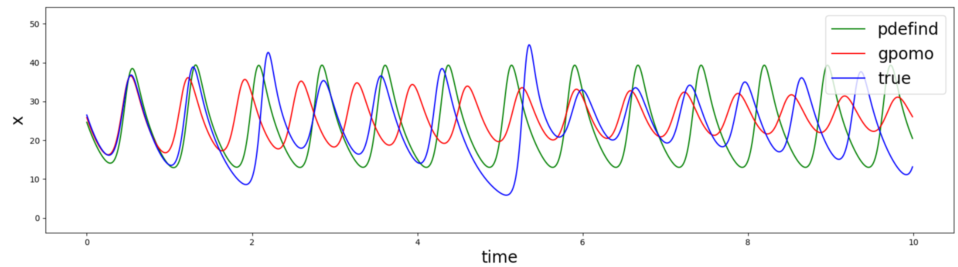


(d) ***(e)*** ***(f)***

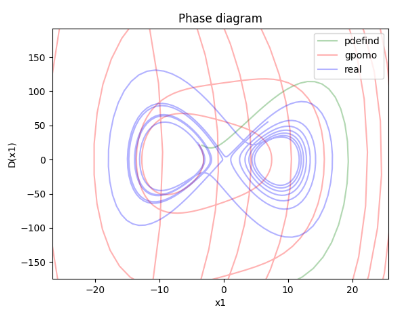
Figure 4: *For the Rosseler system the time series for variables (a) y and (b) z with no noise. In (c) variable x, (d) variable y and (e) variable z phase diagrams.*



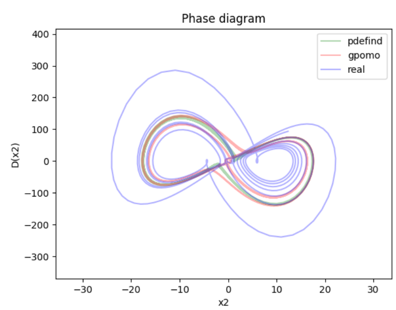
(a)



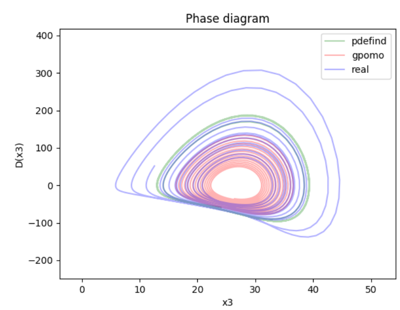
(b)



(c)



(d)



(e)

Figure 5: *For the Lorenz 1963 system the time series for variables (a) y and (b) z with no noise. In (c) variable x, (d) variable y and (e) variable z phase diagrams. The best reconstruction is obtained with y*

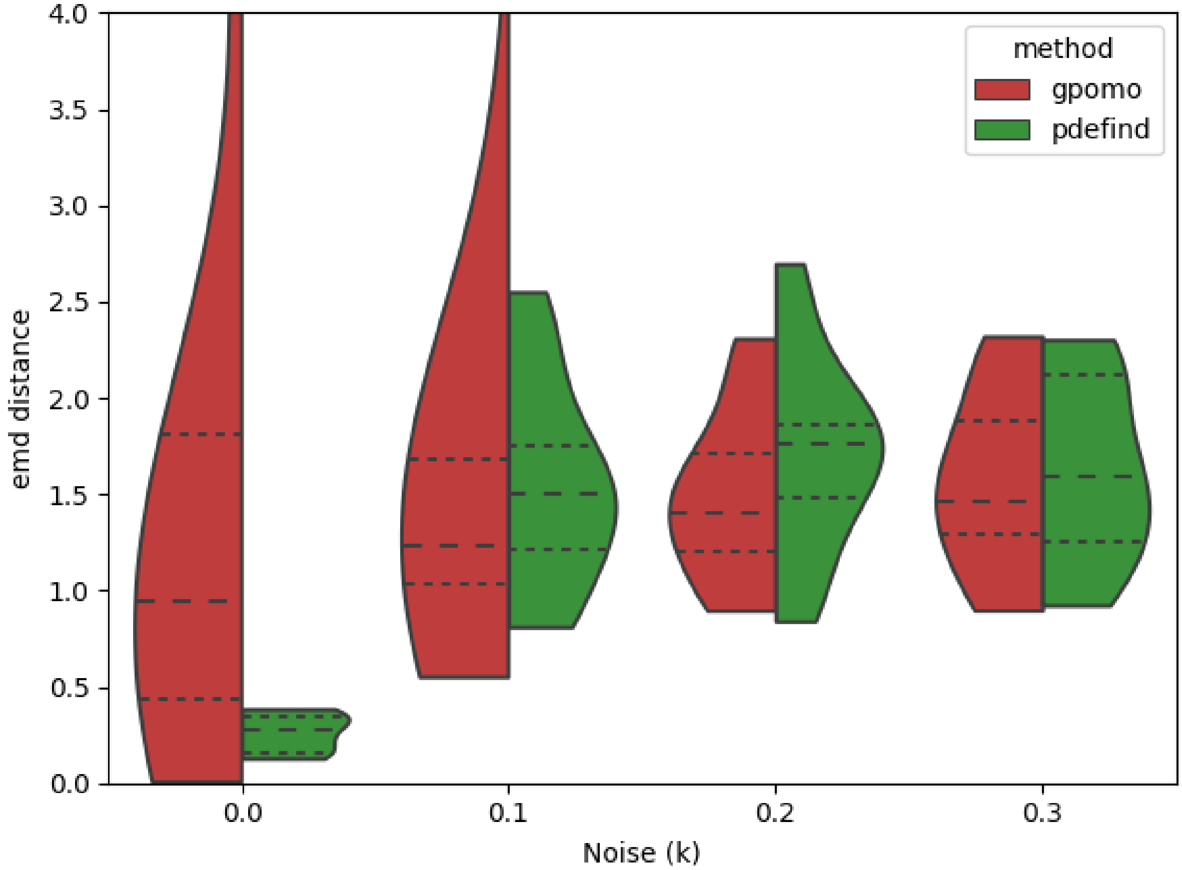


Figure 6: *Violin plot showing the behaviour of the methods through the emd metric over the phase diagrams. The black horizontal lines in each distribution show the quartiles of the distributions. We can see that pdefind-latent has better performance when thare is no noise as the values are nearer to zero and when noise is present both methods diminish their power.*

We close this section by highlighting the following important fact. pdefind-latent is orders of magnitude faster than GPoMo. For the data-sets analysed in this paper, with or without latent variables, it takes less than 10 seconds to compute pdefind-latent, whereas it takes minutes or even hours to compute GPoMo using the GPoMo `R` package (see Figure 7). This can be explained by the fact that pdefind-latent solves a convex optimization problem, but GPoMo performs a complicated combinatorial search.

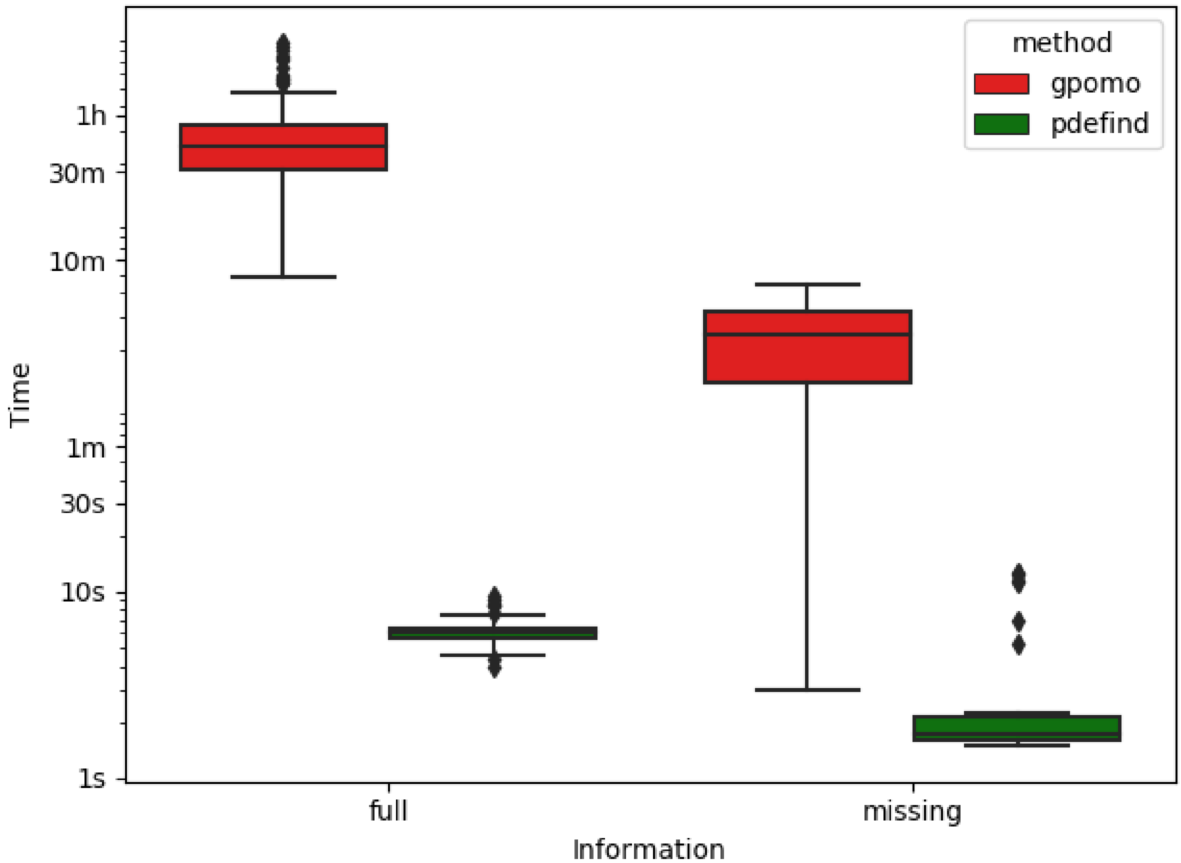


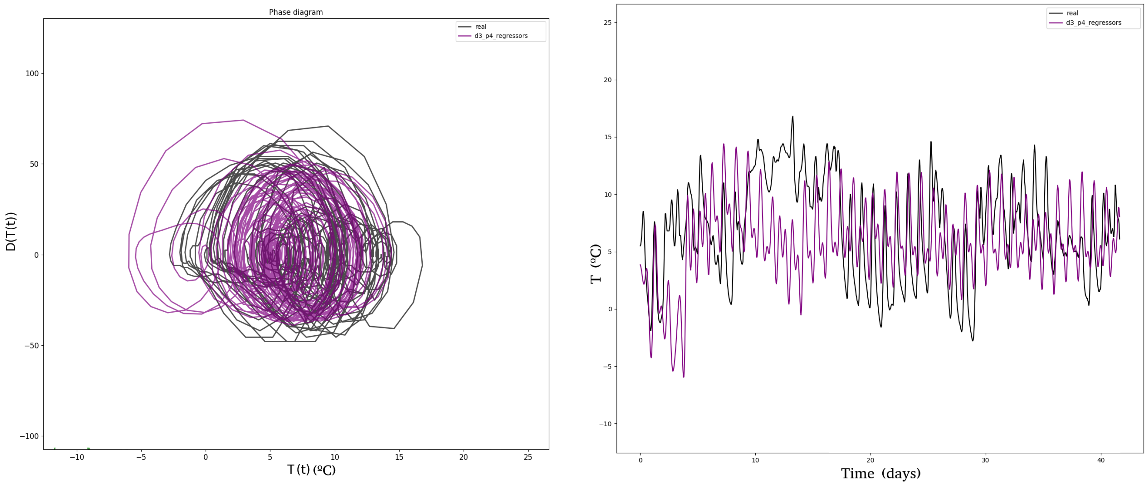
Figure 7: *Comparison of the time taken by both methods for scenarios grouped according to whether all information is available (full) or only one variable is available (missing). The pdefind-latent method is orders of magnitude faster then GPoMo usually taking less than* 10 *seconds compared to more than a minute for GPoMo taken even hours in some cases.*

4.3 RTE data

Finally, we address the problem of recovering a differential equation using the temperature series provided by RTE. We try different settings: one time series of one geographical point, two geographical points far from each other or two neighbouring points . At the same time we created dictionaries of functions to feed the algorithm using different combinations of derivatives, polynomials and regressors. The two regressors added are trigonometric functions with period of 1 day and 1 year fitted using the temperature series.

Derivative order	Polynomial order	Regressors
2	4	day/year or None
3	4	day/year or None
8	1	day/year or None

As the task was indeed difficult, the majority of the models where not compatible with the real data when looking at the predictions in the phase diagram. Many of them eventually diverged except for the case of using regressors with two separate geographical points and derivative order three and polynomials up to $4th$ order (Figure 8).



(a) *(b)*

Figure 8: *RTE data fitted using 2 temperature series from distant locations and 3rd order derivatives, 4th polynomial order and regressors for day and year. (a) Phase diagram from one of the geografical points, (b) real and predicted series. The behaviour of the predictions using the fitted model despite having a different trajectory it generates data similar to the original series.*

5 Discusion

When dealing with information loss in a known and controlled systems we demonstrated that the methodology proposed by Brunton, Proctor, and Kutz 2016, Rudy et al. 2017 and Quade et al. 2018 is a powerful tool to have in hand when looking to learn systems of differential equation to model data measurements. When all information is available, pdefind-latent gives results are competitive with those obtained using the GPoMo algorithm when there is no noise added. When only noisy measurements are available, the performance of both algorithms deteriorates. When not all variables of the system are observed, both methods are able to recover useful dynamical systems when no noise is added to the observations. In this cases, pdefind-latent has a superior performance than GPoMo, as measured by the quality of the estimates phase diagrams. When using the temperature data from RTE wwe found an equation whose predictions yielded a behaviour compatible with the real series when looking at both the phase diagram and the future predictions.

Importantly, pdefind-latent is orders of magnitude faster than GPoMo (see Figure 7): for the data-sets analysed in this paper it takes less than 10 seconds to compute, whereas it takes minutes or even hours to compute GPoMo using the GPoMo `R` package.

References

[BL07a] Josh Bongard and Hod Lipson. "Automated reverse engineering of nonlinear dynamical systems". In: *Proceedings of the National Academy of Sciences* 104.24 (2007), pp. 9943–9948. ISSN: 0027-8424. DOI: 10.1073/pnas.0609476104. eprint: https://www.pnas.org/content/104/24/9943.full.pdf. URL: https://www.pnas.org/content/104/24/9943.

[BL07b] Josh Bongard and Hod Lipson. "Automated reverse engineering of nonlinear dynamical systems". In: *Proceedings of the National Academy of Sciences* 104.24 (2007), pp. 9943–9948.

[BPK16] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. "Discovering governing equations from data by sparse identification of nonlinear dynamical systems". In: *Proceedings of the National Academy of Sciences* 113.15 (2016), pp. 3932–3937. ISSN: 0027-8424. DOI: 10.1073/pnas.1517384113. eprint: https://www.pnas.org/content/113/15/3932.full.pdf. URL: https://www.pnas.org/content/113/15/3932.

[Fri+07] Jerome Friedman et al. "Pathwise coordinate optimization". In: *Ann. Appl. Stat.* 1.2 (Dec. 2007), pp. 302–332. DOI: 10.1214/07-AOAS131. URL: https://doi.org/10.1214/07-AOAS131.

[HTW15] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical learning with sparsity: the lasso and generalizations*. Chapman and Hall/CRC, 2015.

[KYO89] Shigeo Kida, Michio Yamada, and Kohji Ohkitani. "Route to Chaos in a Navier-Stokes Flow". In: *Recent Topics in Nonlinear PDE IV*. Ed. by Masayasu Mimura and Takaaki Nishida. Vol. 160. North-Holland Mathematics Studies. North-Holland, 1989, pp. 31–47. DOI: https://doi.org/10.1016/S0304-0208(08)70505-X. URL: http://www.sciencedirect.com/science/article/pii/S030402080870505X.

[Lor63] Edward N Lorenz. "Deterministic nonperiodic flow". In: *Journal of the atmospheric sciences* 20.2 (1963), pp. 130–141.

[Man+12] S. Mangiarotti et al. "Polynomial search and global modeling: Two algorithms for modeling chaos". In: 86.4, 046205 (Oct. 2012), p. 046205. DOI: 10.1103/PhysRevE.86.046205.

[Pac+80] N. H. Packard et al. "Geometry from a time series". In: 45.9 (Sept. 1980), pp. 712–716. DOI: 10.1103/PhysRevLett.45.712.

[Qua+18] Markus Quade et al. "Sparse identification of nonlinear dynamics for rapid model recovery". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 28.6 (2018), p. 063116.

[Rud+17] Samuel H. Rudy et al. "Data-driven discovery of partial differential equations". In: *Science Advances* 3.4 (2017). DOI: 10.1126/sciadv.1602614. eprint: https://advances.sciencemag.org/content/3/4/e1602614.full.pdf. URL: https://advances.sciencemag.org/content/3/4/e1602614.

[SL09] Michael Schmidt and Hod Lipson. "Distilling free-form natural laws from experimental data". In: *science* 324.5923 (2009), pp. 81–85.

[Tak81] F. Takens. "Detecting strange attractors in turbulence". In: *Lecture Notes in Mathematics, Berlin Springer Verlag* 898 (1981), p. 366. DOI: 10.1007/BFb0091924.

[Tib11] Robert Tibshirani. "Regression shrinkage and selection via the lasso: a retrospective". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73.3 (2011), pp. 273–282.

[Vil08] Cédric Villani. *Optimal transport: old and new*. Vol. 338. Springer Science & Business Media, 2008.