Learning the Dynamics of Sparsely Observed Interacting Systems - MVA 2024/2025

Thomas Gravier thomas.gravier@gadz.org
Thomas Loux thomas.loux@polytechnique.edu

December 19, 2024

1 Introduction

Our project follows the work from Learning the Dynamics of Sparsely Observed Interacting Systems (Bleistein, L., Fermanian, A., Jannot, A. S., & Guilloux, A.). This paper makes use of theory of signatures in order to perform time series prediction from multivariate temporal features. This theory allows to obtain a high dimenson linear equation to make this prediction. The authors then introduce their model SigLasso which performs a regression to solve this linear optimization problem. Additionally, this method can be applied for sparse and irregular sampling which is case if the sampling is expensive, difficult or with side-effects. The authors indeed use their model in a medical context to predict the growth rate of hospitalization in France during the coronavirus pandemic.

We set the problem in which one wants to predict the temporal evolution of a quantity $y_t \in \mathbf{R}^p$, $p \geq 1$ as a function of some features $x_t \in \mathbf{R}^d$ for t $\in [0,1]$. Actually this set up is vary common and corresponds to a differential equation of the form:

$$y_t = y_0 + \int_0^t G(y_s, x_s) ds \tag{1}$$

assuming that such a function G exists.

More specifically regarding the data, while a fixed grid is most of time considered in the literature of time series, the authors consider a setting in which the time between measurements may vary at sample level, the grid may be different betweens samples and the number of measurements may be different. The authors make a thorough mathematical work to show provide convergence condition and bounds on the error. They also show robustness to noise on features or targets.

In this project, we provide a comparison of their algorithm to ground truth values, that can be computed in the context of synthetic data (Thomas Loux). Moreover, we use them on irregular weather data (Thomas Gravier). Both were not present in the paper. We use the available code from the authors to use their SigLasso (in PyTorch) which uses torchcde, iisignature and scikit-learn. Torchcde allows to solve Controlled Differential Equations (CDE) which are a generalization of ODEs. iisignature is a library to compute signatures. In the following, we will use the same notation as the authors.

2 Method

2.1 Riemann-Stieljes Integral and paths

In the field of Controlled Differential Equations, we denote our temporal data as path. These paths are assumed to be continuous and with a bouded total variation: there exist $L \geq 0$ such that:

$$\|x\|_{TV} = \sup_{\pi} \sum_{i=1}^{n} \|x_{t_i} - x_{t_{i-1}}\| \leq L \tag{2}$$

for all partitions π of the interval $[0, 1]$. The total variation is a measure of the "wiggliness" of the path. Supposing that f and g are functions, with f continuous and g of bounded variation, the Riemann-Stieljes integral of f with respect to g is defined as:

$$\int_0^1 f(x)dg(x) = \lim_{n \to \infty} \sum_{i=1}^{n} f(x_i)[g(x_{i+1}) - g(x_i)] \tag{3}$$

2.2 Controlled Differential Equations

Under mild approximation, an ordinary differential equation can be rewritten like:

$$y_s = y_0 + \int_0^s F(y_s)dx_s \tag{4}$$

using the formalism Riemann-Stieljes integral, with $F : \mathbb{R}^p -> \mathbb{R}^{p \times d}$, $y_0 \in \mathbb{R}^p$ and $y_s \in \mathbb{R}^p$. This is a Controlled Differential Equation (CDE). Using the signature of the path x_s, one can rewrite the equation as:

$$y_s \approx S_N(x_{[0,s]})\theta_N(y_0) \tag{5}$$

With N the order of the signature considered. The important property is that θ does not depend on time. This means that given y_0, the signature of the path x_s is enough to compute the solution of the CDE (for a given function F), which are assumed to be know ahead of time. F does not need to be known anymore and the problem is reduced to a linear regression problem. The drawback is that θ depends on y_0 and the problem may be of high dimension in order to be a good approximation. The detailed demonstration can be found in the paper [1] and the studied provide conditions for this approximation to converge.

2.3 Signatures

Let x_s be a path in $\mathbb{R}^d$, continuous and of bounded variation, k the order of the signature and I = $a_1, ..., a_k$ a multi-index of length k, so that $a_i \in 1, ..., d$ for all i. The signature of the path is defined as:

$$S_k(x_{[0,s]})_I = \int_{0<t_1<...<t_k<s} dx_{t_1}^{a_1}...dx_{t_k}^{a_k} \tag{6}$$

The signature of order k in then defined as the concatenation of all the signatures of order k:

$$\mathbb{X}_k(x_{[0,s]}) = [S_k(x_{[0,s]})_I]_{I \in I_k} \tag{7}$$

where the multi-index is sorted in lexicographic order. The full signature is then defined as the concatenation of all the signatures of all orders:

$$\mathbb{X}(x_{[0,s]}) = [\mathbb{X}_k(x_{[0,s]})]_{k=1}^{N} \tag{8}$$

and the truncated signature is defined as:

$$\mathbb{X}_N(x_{[0,s]}) = [\mathbb{X}_k(x_{[0,s]})]_{k=1}^N \tag{9}$$

The size of the truncated signature is then: $s_d(N) = 1 + d + d^2 + ... + d^N = \frac{d^{(N+1)}-1}{d-1}$

We refer to this paper for more details on the signature theory and geometric interpretation of the signature [2]. In practise, the path is always considered to incorporate the time variable as its first component. The computation of the signatures at order 1 and 2 can be found in Appendix 5.1.

Computationally, the signature can be efficiently computed recursively on the order of the signature. With real data, one has only a discrete number of data points. A linear interpolation is used to compute the signature because of how easy one can compute the signature of a linear path and thus piecewise linear paths.

2.4 SigLasso model

With the previous model, learning the dynamics of the system boils down to the following regression problem:

$$min_\theta = \|Y - S_N(x_i)\theta\|^2 \tag{10}$$

The authors also introduce a L1 regularization term to control iid noise. Assuming the noise is subgaussian, or for simplicity that it is bounded then using the Hoefding inequality, an optimal regularization is:

$$\sigma(\theta) = \sum_{i=1}^{N} \frac{\sqrt{k}}{k!} \|\theta_i\|_1 \tag{11}$$

In order to make use of existing implementation of Lasso regression, the authors rescale accordingly the signature matrix. Hence the final optimization problem is:

$$min_\theta = \frac{1}{2M} \|Y - S_N(x_i)W\theta\|^2 + \lambda \|\theta\|_1 \tag{12}$$

Where M is the number of y samples.

3 Data

3.1 Synthetic data

We define the desired values of X and Y dimensions. We then generate random features X with cubic Hermite spline to have a smooth path. We then generate the target Y using a CDE with a known function F. We consider the following eqution with A a random matrix:

$$dy_t = Sigmoid(Ay_t) \times dx_t \tag{13}$$

all target paths start at the same value y_0. We then subsample the values in X and Y to simulate the case of irregular sampling. An example can be found in Appendix 5.2

3.2 Weather data

We aim to apply the models to real-world data and have chosen a weather dataset available at `https://www.kaggle.com/datasets/muthuj7/weather-dataset`. Since the data is unprocessed, we preprocess it by using histograms and subsampling to enhance its suitability for our model. Our objective is to explain the variable **Temperature** using the variables **Humidity**, **Time**, **Pressure**, and **Rain Intensity**. All target paths start at the same initial value, y_0. To simulate irregular sampling, we then subsample the values in both X and Y.

4 Results

4.1 Synthetic data

Knowing the exact expression of $\theta_N(y_0)$, we can use PyTorch to compute the successive Jacobian matrix from $y- > Sigmoid(Ay)$ at $y = y_0$. This experiment that was not present in our paper allows to obtain the best θ and assess the efficiency of the linearization. In practice, the reconstruction is very good, even this low order of signature.
Overall, we obtain a fairly low loss on the synthetic dataset, evaluated at the order of the signature from 1 to 5 3. The loss decreases as the order of the signature increases. The performance of this approximation decreases for higher dimensions of X and Y.

The main obstacle with this method (SigLasso or ground truth data) is the exponential complexity as one increases the order of the signature. In practice, the inference time is quite long as one may want to predict long time series (more than a hundred timesteps). In low dimension (here dim X is 3 and dim Y is 2), with 50 samples and 300 timesteps, the inference is too long on a consummer computer for a order bigger than 7.

Regarding SigLasso performance, it is best in low order, as the loss increases from order 5 in our experiments 5.7. When comparing the MSE on last point between SigLasso and groud truth θ for order 1 to 5, we find that SigLasso reaches the optimal loss for order 1 to 3 before increasing. The θ found is really closed to the ground truth 5.8.

Lastly, we find that the normalization of the data that is used, to set the total variation to 1, increases the loss 5.9. After checking the total variation of feature paths X, their values is above 1 (around 25), which is counterintuitive.

4.2 Weather Data

We use the model up to a signature order of 4, which allows us to capture higher-order interactions within the data. To evaluate the model's performance, we conducted four distinct experiments: one using all available features, one excluding pressure, and for each of these configurations, with and without normalization. This experimental setup enables us to assess the significance of pressure as a feature and the impact of normalization on prediction accuracy.

In each case, we compute up to the 4th signature and evaluate the model using the Mean Squared Error (MSE). The goal is to determine which configuration provides predictions that best align with the training data. The data is divided into various time steps to optimize training for prediction tasks, ensuring a robust evaluation process. Importantly, care is taken to ensure that the test set is never exposed to the model during training, preserving the integrity of the evaluation.

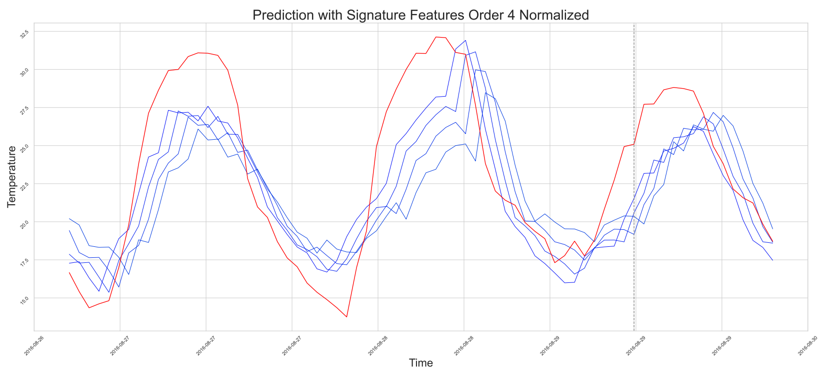


Figure 1: Prediction with all features included (normalized).

From the results, we observe that the prediction starting from the initial point y_0 performs significantly better when all features are included, indicating that pressure is a critical feature. Excluding pressure results in a noticeable decline in model performance, suggesting its importance in capturing the dynamics of the weather data.

Furthermore, unlike modeling with synthetic data, the irregularity of real-world weather data introduces additional challenges. Normalization becomes essential in this context, as it significantly improves the stability and accuracy of the predictions. Without normalization, the model struggles to generalize effectively, resulting in higher MSE values.

The results are visually presented in Figure 10, which demonstrates the superior performance of the configuration that includes all features with normalization. Additional experimental results, covering scenarios without pressure and non-normalized data, are discussed in the appendix. These findings highlight the importance of feature selection and preprocessing steps like normalization in improving the robustness of path signature-based models for real-world datasets.

4.3 Critique of the Article

The article is well-written, reproducible, and processes the data effectively. The mathematical concepts are clearly explained, making the subject accessible even to those new to the field. However, the application of prediction using controlled differential equations (CDEs) or prediction in general with signatures proves somewhat disappointing when working with real-world, irregular data.

Exploring other domains where path signatures are utilized, we found that they are significantly more effective in capturing interpretable features. Path signatures provide fixed-length vector outputs regardless of the number of input points or their sampling times. In several studies, path signatures have been employed to generate features for identifying, classifying, or clustering time series, often achieving improved accuracy compared to methods that do not utilize signatures.

References

[1] Adeline Fermanian, Pierre Marion, Jean-Philippe Vert, and Gérard Biau. Framing rnn as a kernel method: A neural ode approach. *Advances in Neural Information Processing Systems*, 34:3121–3134, 2021.

[2] Lajos Gergely Gyurkó, Terry Lyons, Mark Kontkowski, and Jonathan Field. Extracting information from the signature of a financial data stream. *arXiv preprint arXiv:1307.7244*, 2013.

5 Appendix

5.1 Signature at order 1 and 2

We derive the first two orders of the signature for a path $x_s = (t, f(t))$ for a differentiable function f and f(0) = 0. The signature of order 0 is trivially 1. The signature of order 1 is:

$$S^1(x_{[0,s]}) = \int_0^s dt = sS^2(x_{[0,s]}) = \int_0^s df(t) = \int_0^s f'(t)dt = f(s) \tag{14}$$

The signature of order 2 is:

$$S_{1,1}(x_{[0,s]}) = \int_{0<t_1<s} df(t_1)dt_1 = \int_{0<t_1<s} f'(t_1)dt_1 = \frac{f(s)^2}{2} \tag{15}$$

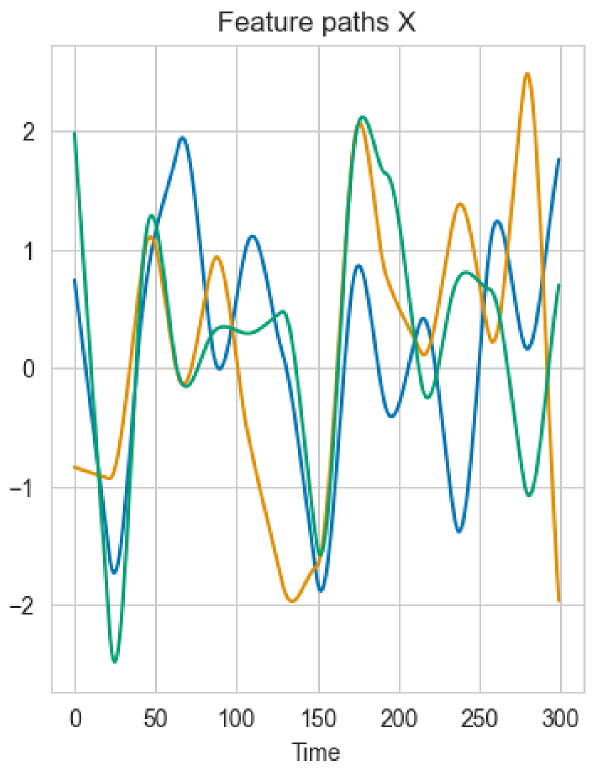
$$S_{0,1}(x_{[0,s]}) = \int_{0<t_1<s} dt_1 df(t_1) = \int_{0<t_1<s} dt_1 f'(t_1) = \frac{f(s)^2}{2} \tag{16}$$

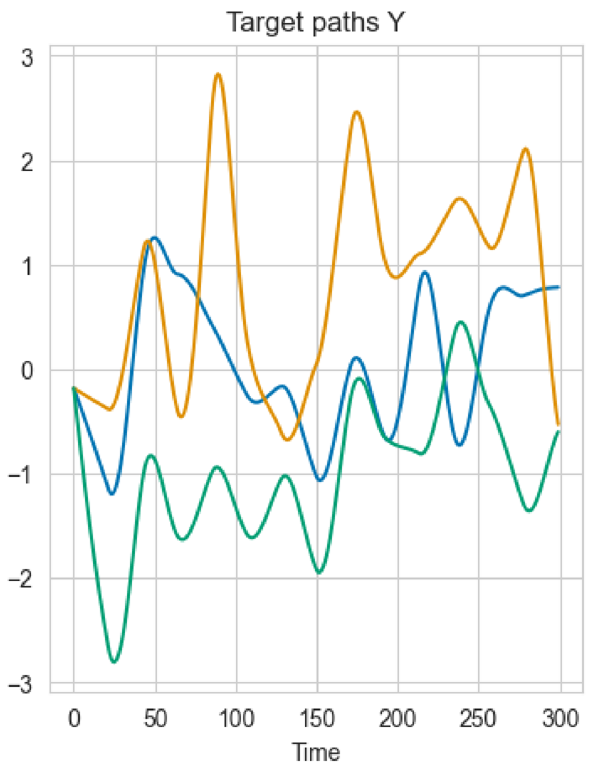
$$S_{1,0}(x_{[0,s]}) = \int_{0<t_1<s} dt_1 dt_1 = \frac{s^2}{2} \tag{17}$$

$$S_{0,0}(x_{[0,s]}) = \int_{0<t_1<s} dt_1 dt_1 = \frac{s^2}{2} \tag{18}$$

and it captures the evolution of the area under the curve of f.

5.2 Synthetic data example





5.3 Loss in low dimensions ($dim_X = 3$ and $dim_Y = 2$)

5.4 Reconstruction loss on ground truth θ

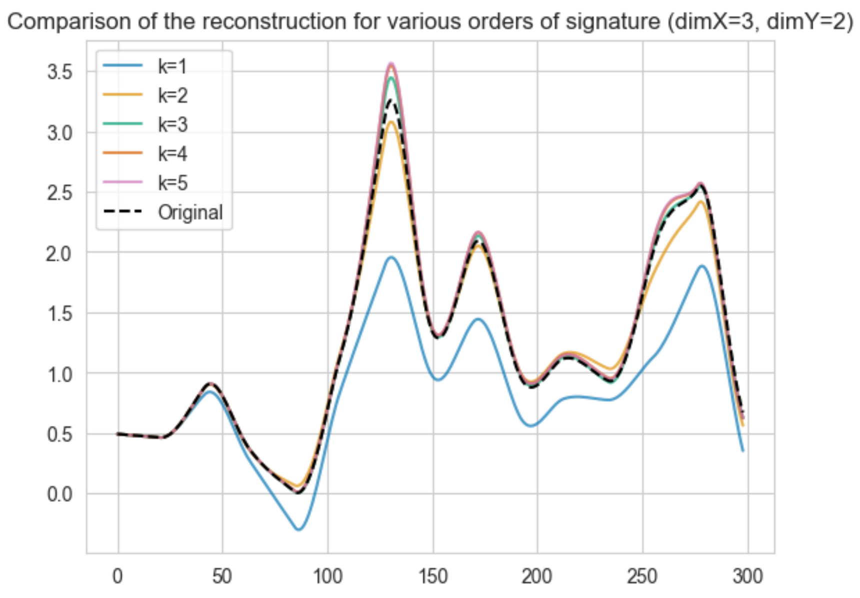


Figure 2: An example of reconstruction for orders from 1 to 5

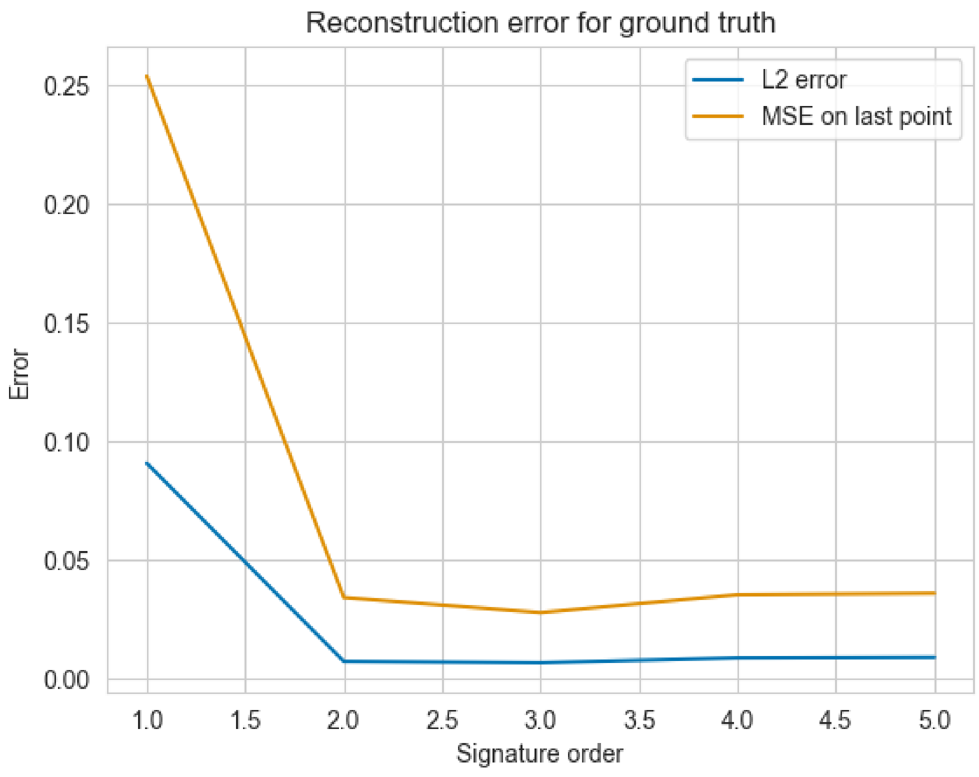


Figure 3: For $Dim_X = 3$ and $Dim_Y = 2$

5.5 Loss in high dimensions ($dim_X = 10$ and $dim_Y = 5$)

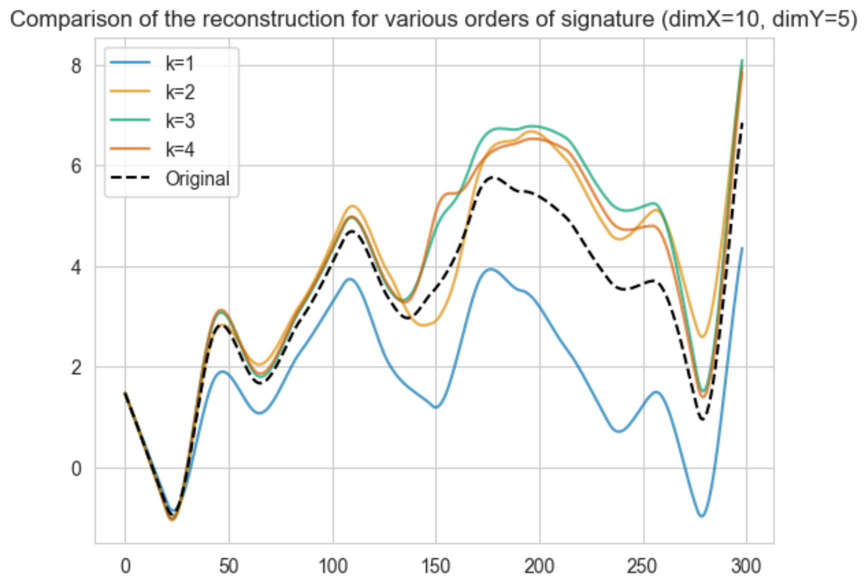


Figure 4: An example of reconstruction for orders from 1 to 4. We can observe a divergence as the time increases.

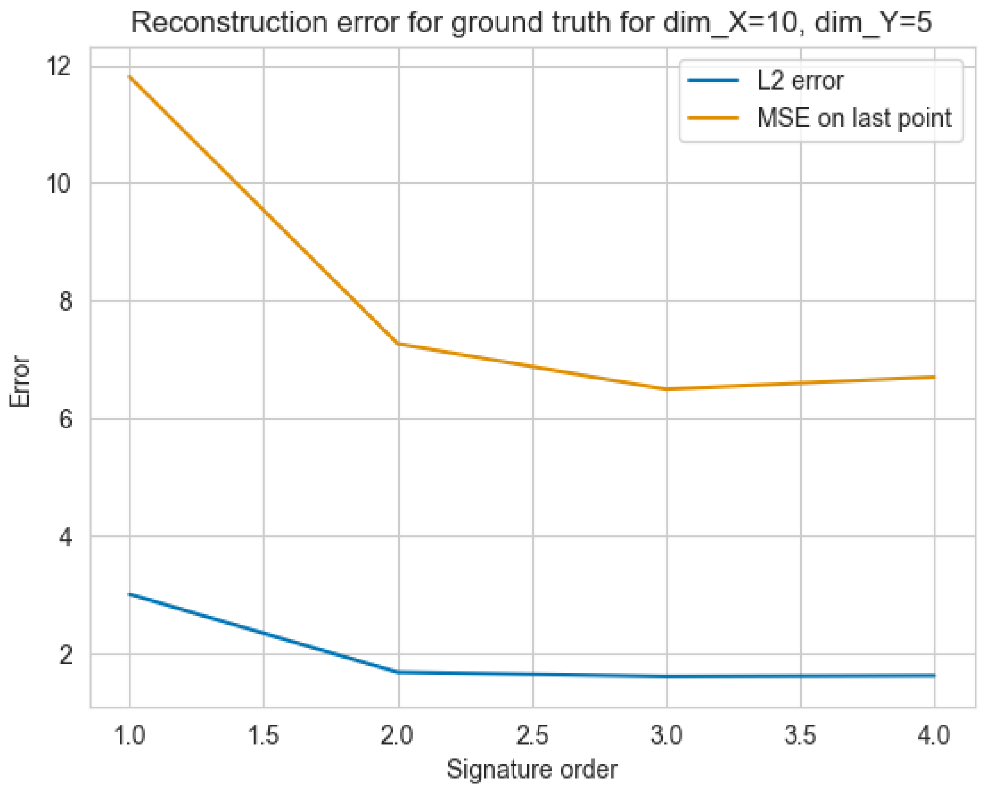


Figure 5: For $dim_X = 10$ and $dim_Y = 5$ the error is significantly higher.

5.6 Training time on synthetic dataset

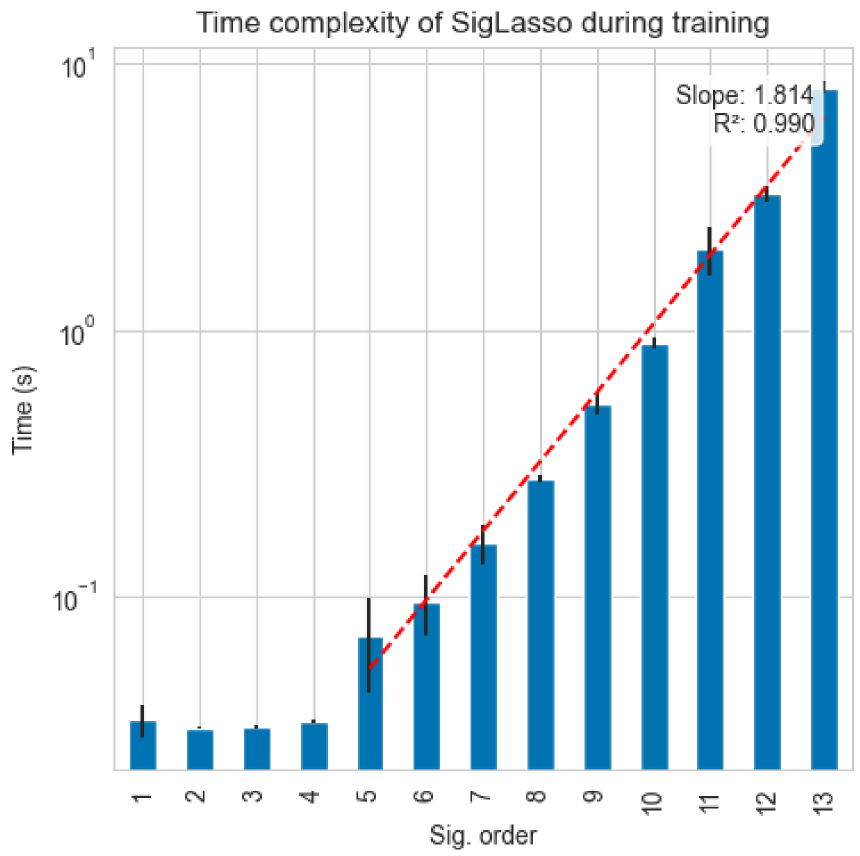


Figure 6: Training time for $dim_X = 3$ and $dim_Y = 2$. We indeed obtain an exponential time function.

5.7 SigLasso loss

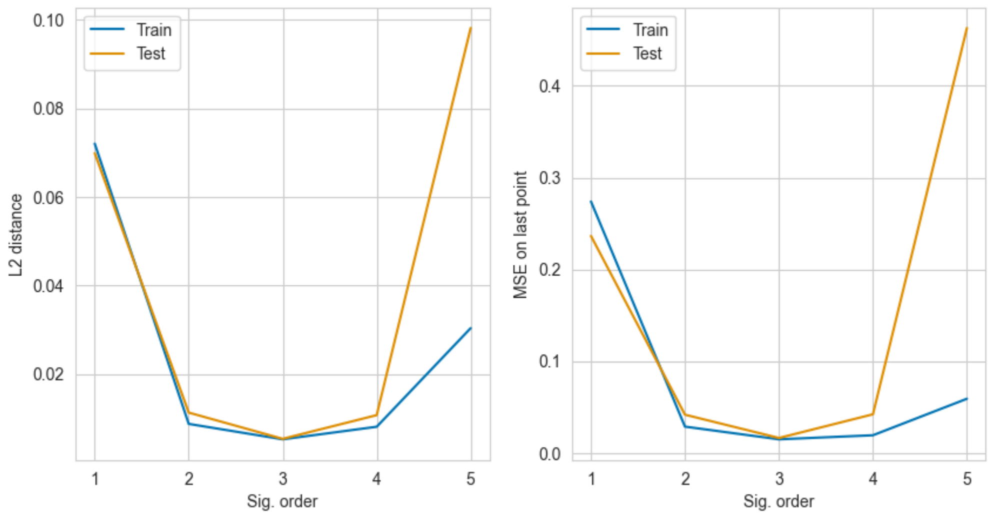


Figure 7: Reconstruction losses (L2 on train or test) for SigLasso on synthetic dataset

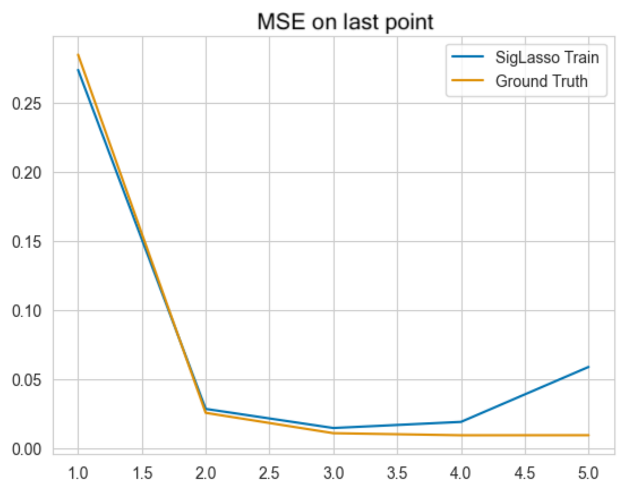


Figure 8: MSE on last point comparison between SigLasso and groud truth θ for order 1 to 5. In this experiment, SigLasso reaches the optimal loss.

5.8 θ comparison from SigLasso and ground truth

order	L2 distance
1	0.018
2	0.038
3	0.111
4	0.746
5	0.965
6	1.775

Table 1: Loss between θ obtained from SigLasso and ground truth in low dimension. SigLasso provides a value close to the optimal θ in this setting.

5.9 Normalization influence

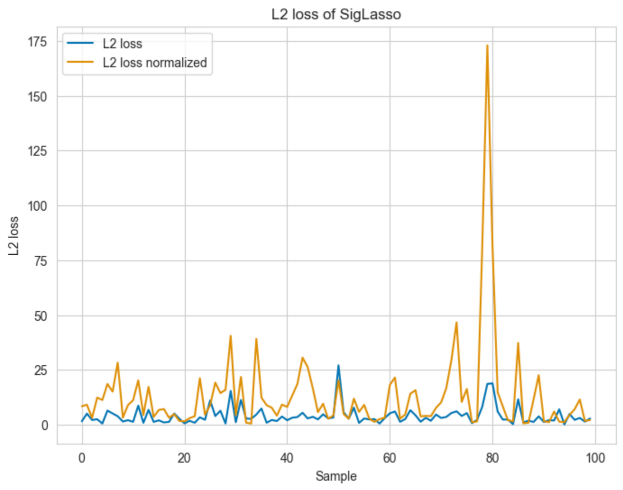


Figure 9: Surprisingly the L2 loss per sample is almost systematically higher in the normalized setting than without.

5.10 Prediction on weather data

In this section, we present predictions on weather data using path signatures under different configurations. Each plot corresponds to a specific experimental setup, with variations in included features and normalization.

5.11 Prediction on Weather Data

In this section, we present predictions on weather data using path signatures under different configurations. Each plot corresponds to a specific experimental setup, with variations in included features and normalization.

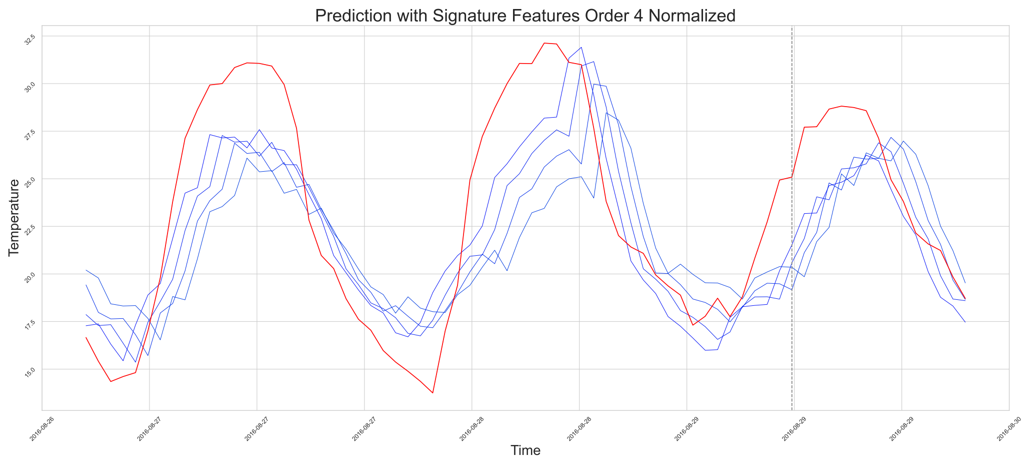


Figure 10: All features included (normalized). The model demonstrates improved performance when utilizing all available features, highlighting the importance of pressure in the dataset.

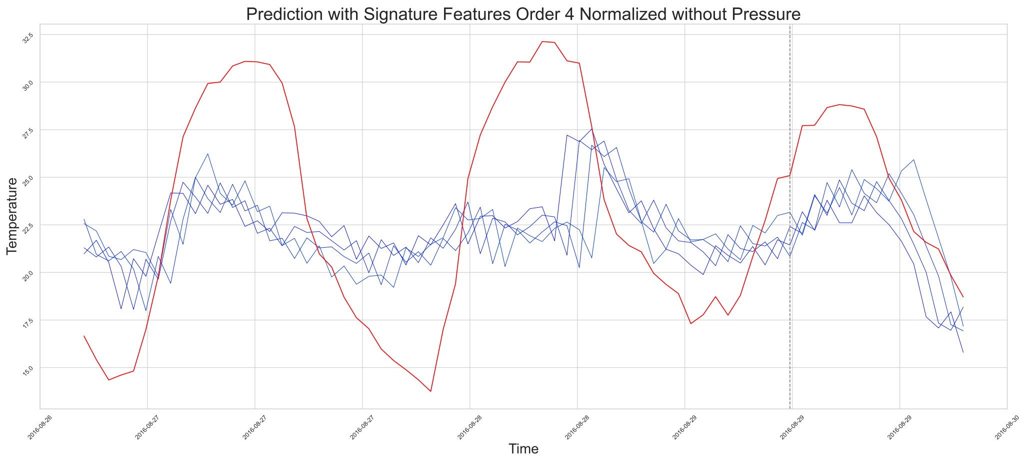


Figure 11: Without pressure (normalized). This configuration shows a reduction in prediction accuracy, underlining the significance of pressure as a feature.

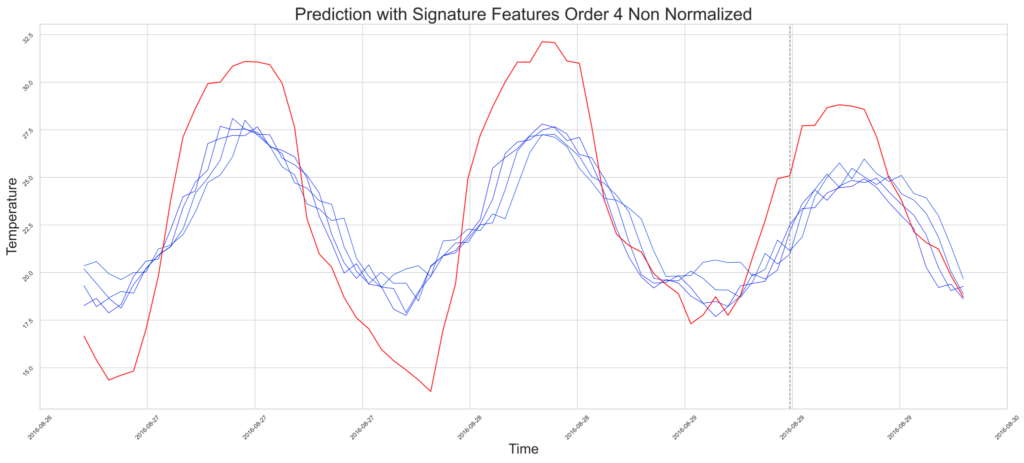


Figure 12: All features included (non-normalized). Predictions without normalization are less accurate, showing the importance of data preprocessing.

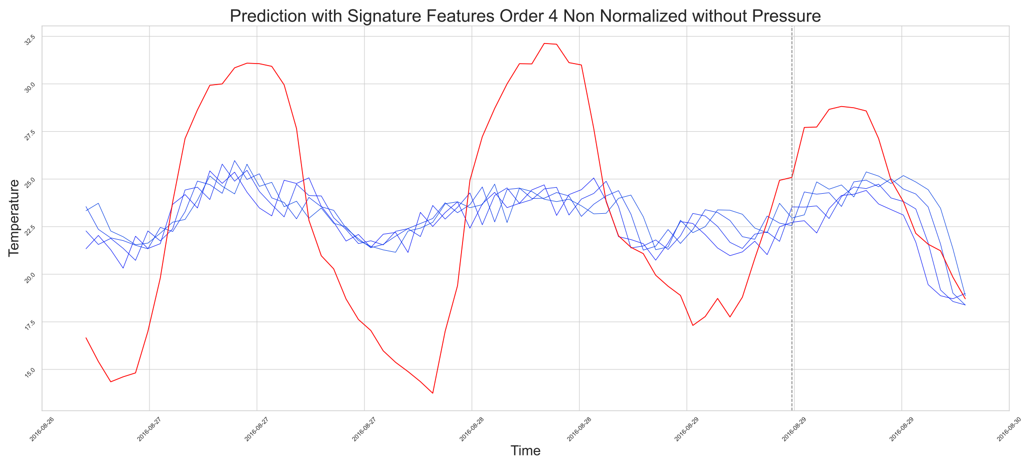


Figure 13: Without pressure (non-normalized). Removing pressure and skipping normalization further reduces prediction accuracy.