MODELLING STOCHASTIC TIME DELAY FOR REGRESSION ANALYSIS

AARON PICKERING AND JUAN CAMILO ORDUZ

Abstract. Systems with stochastic time delay between the input and output present a number of unique challenges. Time domain noise leads to irregular alignments, obfuscates relationships and attenuates inferred coefficients. To handle these challenges, we introduce a maximum likelihood regression model that regards stochastic time delay as an 'error' in the time domain. For a certain subset of problems, by modelling both prediction *and* time errors it is possible to outperform traditional models. Through a simulated experiment of a univariate problem, we demonstrate results that significantly improve upon Ordinary Least Squares (OLS) regression.

1. Introduction

Consider the typical univariate regression problem where the intention is to find the relationship between x and y. When extended to time series, a number of time specific complexities arise. For the specific case where the input x affects the output y, with a random time delay in between, the estimated coefficients (or weights, predictions etc) are significantly attenuated [1]. This attenuation occurs both in traditional statistical models and machine learning models, and for certain problems can be a serious limitation. The following document proposes techniques for handling this class of time series regression.

As an example, let us take the management of blood sugar level in diabetes patients. In people with diabetes, the pancreas can't effectively regulate blood sugar levels. Therefore, these levels must be controlled by insulin injections and a special diet. The challenge for many people is that the relationship between the input (insulin) and output (blood sugar) is extremely complex [2]. The effect of the insulin injection might be observed after 15, 20 or t minutes depending on a number of factors, many of which are unknown. Due to the stochastic nature of the time delays, the actual effect can't be easily determined. It is difficult to differentiate the effect of the insulin injection from other factors and accurately determine how much one should take.

Inference for this type of problem is especially challenging. Typical regression models require a fixed alignment between cause and effect. Using

Date: April 10, 2021.

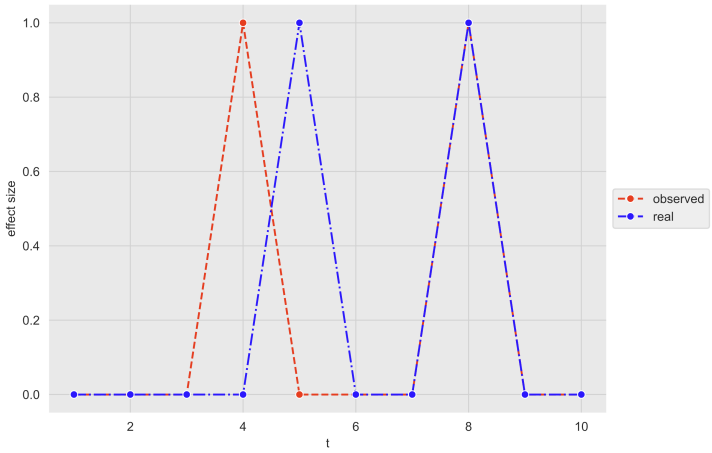

FIGURE 1. Add figure caption

standard methods, we'd need to assume that the effect occurs after some fixed time t which can be inferred from the data. However, if there is any uncertainty in the parameter t (t changes or is noisy) the resulting estimates will be significantly attenuated.

Consider the simple example in Figure 1 where the effect of the input is 1. The observed input is given by the red line, the blue line is when the effect actually occurs. The first effect happens one time point after the input. The second effect happens at the same time as the input. A fixed time delay isn't valid in this case because the time shifts differ.

If one were to model the effect using a fixed time delay and OLS, the estimate would only be half the true value because only one of the outputs is aligned. Obviously, this isn't ideal, we want the parameter estimates to be as close to the real values as possible, regardless of any noise in the lag structures. One can mitigate the problem via time aggregations, however in complicated cases with multiple factors, this is just not feasible.

Therefore, we propose a regression model which can handle stochastic time delay structures. We treat the stochastic time delay components as an 'error' in the time axis. Next, we find the maximum likelihood estimate, for a given set of parameters, considering the time error (t-axis) and the regression error (y-axis) simultaneously.

2. Related Work

There is extensive existing literature on time series problems with time delay dependencies. In the statistical disciplines there a number of treatments which focus on fixed time delay dependencies. For example, Granger Causality [3] is used to determine whether one time series is useful for forecasting another, across some fixed delay. Distributed Lag [4] and Dynamic

Regression models (e.g [5]) are able to handle linear and non-linear cause and effect relations that occur across multiple time lags. Nonetheless, these models assume a fixed time lag dependency.

In the machine learning literature, there are a number of models which can handle complex dependencies across time. Sequence models such as the RNN, LSTM and GRU [6] are able to generate predictions which incorporate time delays between input and the output variables. More recently, attention based models such as the Transformer have become the state-of-the-art for a variety of tasks, including time series forecasting [7]. Yet, all of these models have a tacit assumption of a fixed time lag dependency. There is also Dynamic Time Warping (DTW) [8] which tries to find the optimal alignment between time series sequences by minimizing the distance between the respective inputs. DTW is able to model varying time lag dependencies but is not purposed for regression, instead being mainly used for pattern matching and sequence alignment.

The field of system identification also has considerable literature on dynamic non-linear systems. For example, the NARX [6] model can be used to identify non-linear systems with fixed time delays between input and output. At the same time, there are also treatments on systems with disturbances in the input [9], known as EIV systems. To our knowledge, these works do not deal with uncertainty in the time domain.

Finally, recent papers such as Dynamic Time Lag Regression [10] and Variable-Lag Granger Causality [11] deal with non-stationary time lag dependencies. In other words, the lag structure is assumed to evolve over time.

In this paper, we deal with the specific case of *stochastic time delays* i.e. time delays which vary randomly. There appears to be little existing research on this topic.

3. Methodology

We consider the problem as analogous to the typical error-in-variables (EIV) regression. Ordinary regression analyses (and machine learning models) define the loss function with respect to errors in the y axis only. For EIV, errors are considered in both the y-axis and the x-axis [12]. This is useful when there are measurement errors in the independent variable e.g. because the physical measurements have some degree of random error. Similarly, for this problem we assume that we have errors in the y-axis and the t-axis. That is, there are random prediction errors and random errors in the time domain.

3.1. Model Specification.

Let $x(t)$ be a discrete time series. We want to determine the functional relationship $f : \mathbb{R} \longrightarrow \mathbb{R}$ to some other (target) variable y on observed data $\{y_i\}_{i=1}^{n}$, that is $y = f(x(t))$. The series $x(t)$ must be stationary with a known value for the point at which $f(x(t)) = 0$. That is, we know at which point the input series has no effect on the output

(when it's off). The size of the support of $f(x(t))$ should be small relative to number of points in the domain (we we will explore this condition further in 'Limitations'). We utilise the terminology *impulse* for an individual element of the support of f.

Firstly, let us take the input series $x(t)$ and decompose it into its constituent non-zero impulse components. So, if $x(t)$ is a vector given by

$$x(t) = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

then we decompose the vector into a matrix

$$X(t) = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

where each impulse is treated separately.

Given that each non-zero impulse (row) of $X(t)$ is affected by a random time delay (denoted τ), we then model:

$$y(t) = f(1^T X(t + \mathcal{T})) + \varepsilon \tag{1}$$

where $X(t + \mathcal{T})$ is the matrix of time shifted impulses, $X(t)$ is the observed input, $\mathcal{T}$ is the set of single impulse time delays (τ's), 1 is the vector $(1, 1, \cdots, 1) \in \mathbb{R}^{1\times k}$, where k is the number of impulses and ε is a noise term.

The model definition represents the application of time shifts to the matrix rows and a subsequent reduction by summation over the columns.

We also assume that $\mathcal{T}$ is not a constant, but rather a random draw from some discrete distribution (e.g. discrete gaussian, poisson etc). For the discrete gaussian kernel $\mathcal{T} \sim T(\mu, \sigma)$ or for the poisson distribution $\mathcal{T} \sim Pois(\lambda)$. Similarly one can model the errors as $\varepsilon \sim N(\mu, \sigma_\varepsilon)$.

3.2. **Inference.** In order to find the best estimate of f, we would like to find the function f which maximises the joint log-likelihood of the time-domain shifts and the prediction residuals. Specifically, we maximise:

$$\mathcal{L} = \underbrace{\sum_{i=1}^{n} \log(p(y_i | X_{\tau_i} = x_i; \theta_f, \tau_i))}_{\mathcal{L}_1} + \underbrace{\sum_{i=1}^{n} \log(p(\tau_i | \theta_\mathcal{T}, \theta_f))}_{\mathcal{L}_2} \tag{2}$$

where θ_f and $\theta_\mathcal{T}$ represent the parameters of the model f and time shifts $\mathcal{T}$ respectively. In other words, the $\mathcal{L}_2$ term represents likelihood of time shifts and the $\mathcal{L}_1$ term represents the likelihood of the prediction residuals. We maximise these terms simultaneously. For simplicity, we assume that the time shift distribution and error distributions are independent.

4. Algorithm

Before optimisation, the values of each individual time shift τ are not known. In addition, the prediction errors ε can only be calculated if each value of τ is available (because for each time shift there is a different prediction and hence prediction error). Therefore, our algorithm finds the optimum set of time shifts in an inner optimisation ($\mathcal{L}_2$), while iteratively searching for the optimum parameters of f in an outer optimisation loop ($\mathcal{L}_1$). Firstly, we define a parametric form $f(x;\theta_f)$ and some initial parameters to be estimated for the function f. For simplicity, let's take the univariate linear model with gaussian errors, where the $f(x(t))$ is parameterised by β and σ_ε (error standard deviation). In addition, we choose a poisson distribution for $\mathcal{T}$, where $\theta_\mathcal{T}$ is a parameterised by its mean $\lambda_\mathcal{T}$. The choice of a poisson distribution for $\mathcal{T}$ ensures discrete, positive time shifts only.

First, let us initialise some starting values for each of these parameters.

Now, we want to find the best possible time shift τ for each input impulse in $X(t)$. It stands to reason that the best possible time shift would be one that is not too distant from the observed impulse (i.e. has a high likelihood given some distribution) and also produces the best possible prediction. In this example, we can calculate the prediction y by simply multiplying the shifted value by its parameter β. From there, the likelihood estimate is derived from the time shift and the prediction error.

Finally, we iterate over a number of values of τ optimising until we maximise the likelihood for the specific impulse.

However, we must also consider that the impulses in $X(t)$ are not independent from each other. After shifting, it's possible that two or more effects occur simultaneously.

This is particularly problematic if there are multiple impulses within a short period of time or impulses have a distributed effect over multiple time points.

As an example consider the series

$$x(t) = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

with $\mathcal{T} = (1, 0)$ and $\beta = 1$. For this case,

$$X(t+\mathcal{T}) = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

and the effect is therefore

$$y = \begin{pmatrix} 0 & 0 & 0 & 2 & 0 & 0 \end{pmatrix}$$

To accurately calculate the likelihood, we must optimise the time shifts simultaneously. Therefore, we treat the problem of finding the best set of

time shifts as a discrete optimisation problem. For the optimisation step we utilise two assumptions.

(1) Firstly, smaller shifts are more likely than larger shifts (proportionate to the dispersion of the $\mathcal{T}$ distribution). The algorithm should explore the space of smaller shifts more often than larger shifts.
(2) Secondly, impulses close to each other are more likely to be dependent than impulses further away.

Accordingly, the optimisation procedure is:

(1) Initialise the set of parameters β, σ_ε and λ_τ for the function f.
(2) Find the $\mathcal{T}$ which maximises the likelihood for the given set of parameters:
    (i) Initialize the discrete optimisation algorithm with all parameters set to the mean of the $\mathcal{T}$ distribution (λ_τ).
    (ii) Next, randomly select a small number of impulses, with the value treated as a hyperparameter. For each impulse, a random time shift is drawn from the $\mathcal{T}$ distribution creating a proposal vector. The likelihood (both time shift and prediction error) for the proposal is calculated.
    (iii) If the proposal likelihood is higher than the current maximum likelihood, our estimate is updated.
    (iv) Return to (i) and repeat. The best estimate of the set of $\mathcal{T}$ improves each iteration. The number of iterations (n) is also treated as a hyperparameter.
(3) Optimise the model parameters β, σ_ε and λ_τ.

The procedure consists of an inner optimisation $\mathcal{L}_2$ to find the set of $\mathcal{T}$ and an outer optimisation $\mathcal{L}_1$ to find the set of parameters β, σ_ε and λ_τ. By iteratively alternating between the time shift optimisation and the parameter optimisation we are able to maximise equation (2). For the outer parameter optimisation, typical methods such as gradient descent, genetic algorithms and simulated annealing can be used. In our implementation, we have used the differential evolution algorithm [13] (scipiy.optimize [14]) because of its ability to handle noisy objective functions [13]. In order to improve convergence, we also standardize all input variables to the range (0,1).

We also note that the accuracy of the final parameter estimate is relative to the ratio of the y-axis error and the effect size $f(x(t))$. As the ratio of noise σ_ϵ to effect $f(x(t))$ increases, the time shift distribution $\mathcal{T}$ shrinks. In the event that λ_τ becomes 0, the model becomes a 'fixed lag' model, and the parameter estimate β tends to the standard linear regression coefficient. Therefore, the method provides no guarantee on recovering the exact time shifts, only that the coefficient estimates are equal to or better than their OLS counterparts.

5. Limitations

5.1. **Scaling.** As the length of $x(t)$ increases, more impulses are introduced and the size of the decomposed matrix $X(t)$ also increases. At some point, handling $X(t)$ becomes impractical. To account for this, we assume that distant impulses do not affect each other. Concretely, for the i_{th} and j_{th} impulse (row) in $X(t)$, separated by time $t \longrightarrow \infty$, the effect vectors $f(X_i(t+\tau_i))$ and $f(X_j(t+\tau_j))$ are linearly independent.

Therefore, the matrix $X(t)$ is decomposed into a number of segments, where each segment is composed of a smaller number of dependent impulses. To select the right split locations, we search the time series for locations with the lowest impulse density. Given that we wish to split $X(t)$ into independent matrices $U_n(t)$ and $V_m(t)$, if we choose the split locations such that $\forall_i \, \epsilon \, n$ and $\forall_j \, \epsilon \, m$, $f(U_i(t+\tau_i))$ is linearly independent from $f(V_j(t+\tau_j))$, then the segments will also be independent and the procedure unique (fix notation). In practice, it may not be possible to guarantee complete independence of every segment, in which case results may vary. After segmentation, each matrix can be optimised independently in parallel, making the inner optimisation procedure tractable for longer length sequences.

5.2. **Constraints.** Finally, a note on problem constraints. The likelihood estimates are dependent on the inner optimisation procedure. There is no gaurantee that the global maximum will be found, particularly for sequences with a high density of impulses.

For example, consider the example sequence:

$$x(t) = \begin{pmatrix} 0.5 & 0.6 & 0.4 & 1 & 2 & 1 & 0.3 & 0.2 & 0.5 & 0.8 \end{pmatrix}$$

where each impulse can shift between -2 and +2 positions. Ignoring shifts beyond the edge of the vector, the solutions space of $\mathcal{T}$ is 5^t combinations. In such cases, the estimated likelihood is likely to be close to but not exactly the same as the real value. As the density (in time) of impulses increases, the accuracy of the estimates decrease. Therefore, this model is most appropriate for sparse time series inputs.

6. Experiment

For the model, we use the nomenclature Time Varying Stochastic (TVS) Regression and the full code to reproduce the example can be found via TVS Regression on GitHub.

The following section demonstrates a simulated univariate example of TVS Regression. Figure 2 shows the simulated time series.

The input signal has 20 non-zero impulses which have been drawn from the standard normal distribution. The system is 'off' when $x(t) = 0$. In other words, when $x(t) = 0$, $f(x(t)) = 0$.

FIGURE 2. Simulated Example Data

The true shift distribution $\mathcal{T}$ is given by $\mathcal{T} \sim Poisson(\lambda = 2)$. There is one value of τ for each of the 20 impulses.

The green line represents the shifted series, corresponding to the time at which the effect occurs. The blue line is the output y which includes a small amount of gaussian noise.

The values for the parameters were arbitrarily selected. The output y is therefore defined by the following equation:

$$y = 2(1^T X(t + \mathcal{T})) + 6.5 \tag{3}$$

A histogram of the actual distribution of $\mathcal{T}$ is shown in Figure 3.

After fitting the model, we obtain the following results. Figure 4 shows the model fit (denoted TVS) and a comparison with standard OLS. Figures 5 and Figure 6 show the error distribution of the TVS fit and the convergence of the TVS model respectively.

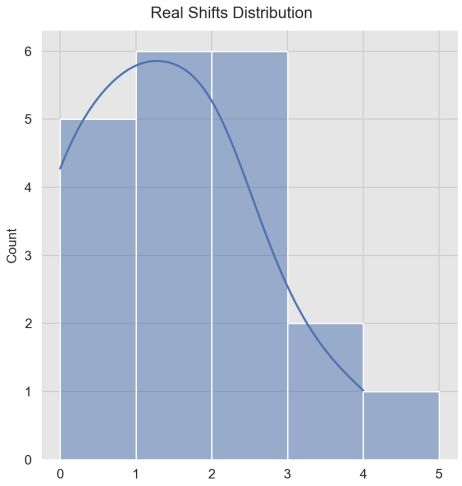

FIGURE 3. Real Shift Distribution $\mathcal{T}$.

Parameter	TVS Regression	OLS Regression	True Value
β	2.09	0.50	2.00
$Intercept$	6.53	6.62	6.50
λ_τ	1.54	N/A	1.40
σ_ϵ	0.20	1.03	0.20

TABLE 1. ...

The true set of $\mathcal{T}$ is:

(4)
$$\mathcal{T} = \begin{pmatrix} 2 & 1 & 2 & 0 & 2 & 3 & 1 & 0 & 2 & 4 & 1 & 2 & 3 & 2 & 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Compared to the estimated $\mathcal{T}$:

(5)
$$\mathcal{T}_{est} = \begin{pmatrix} 2 & 1 & 2 & 0 & 2 & 3 & 1 & 1 & 2 & 4 & 1 & 2 & 3 & 2 & 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

As shown in Table 1, the estimated values for β and σ_ϵ are significantly improved by taking into account the stochastic time delay noise. The estimated β using OLS regression is 0.5, compared to the true value 2. Even a small amount of noise in the time axis causes attenuation, limiting the usefulness of OLS for these problems. In comparison, the TVS Regression algorithm estimates β to be 2.09, much closer to the true value.

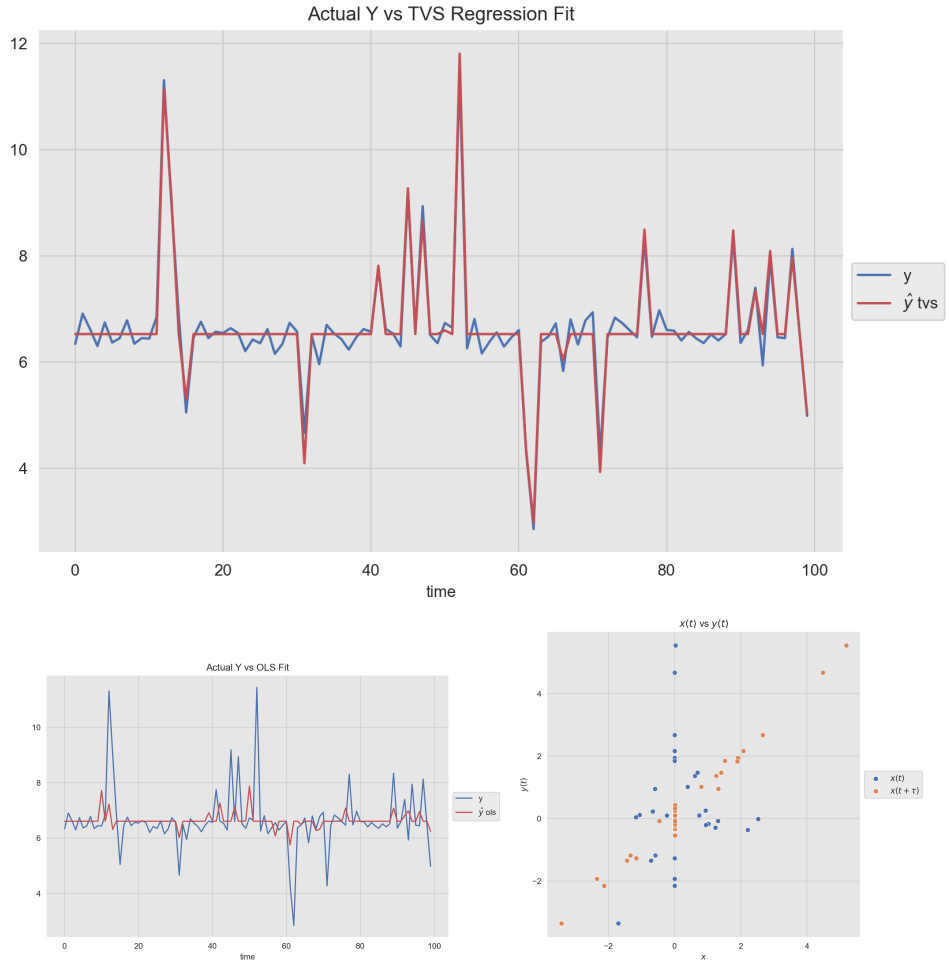

FIGURE 4. TVS vs OLS model fit.

While the example is based on simulated data, we believe that the experiment demonstrates clear potential for improvement in the modelling of real world systems with stochastic time delay noise.

7. Future Work

In the univariate case, the utility of the described method is limited. However, we propose three extensions as future work. The first is to extend the method to multiple regression. We believe that the extension can be built on the same fundamental ideas presented in this document. Next, the model could be extended to include distributed lag structures. A distributed lag structure is where past values of the impulse influence future values of the output [4]. Finally, in relating $x(t)$ to $f(x(t))$, the function f could be extended to the modelling of non-linear forms.

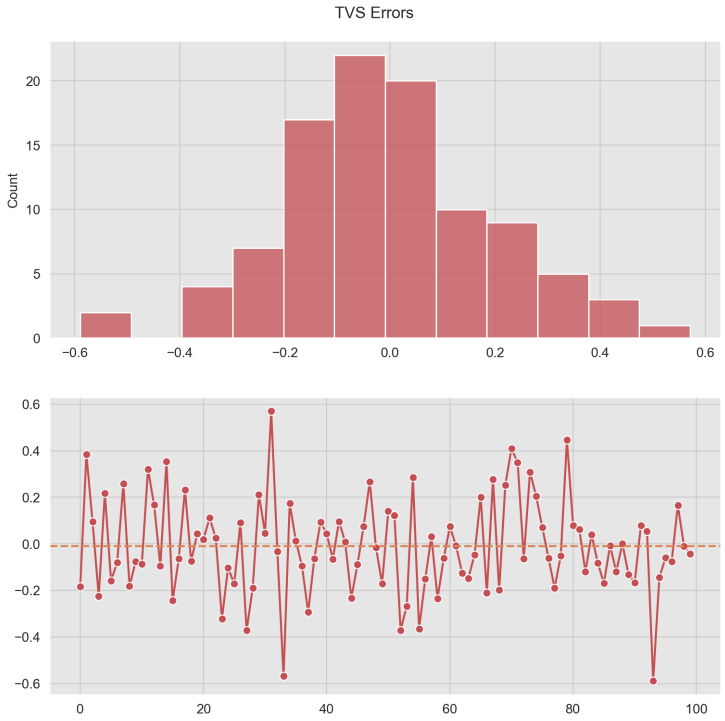

FIGURE 5. Fit Residuals

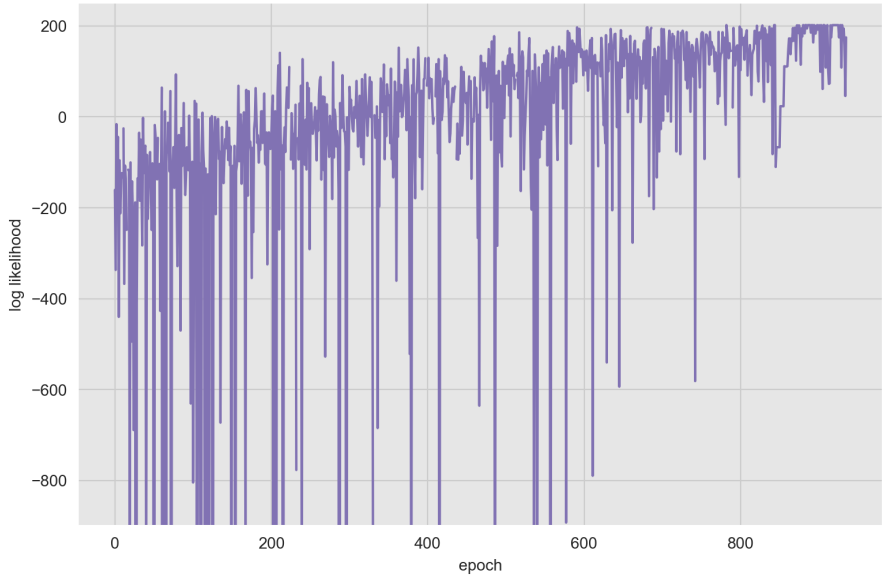

FIGURE 6. Convergence

Through these extensions we can begin to tackle a larger number of practical problems.

In addition, by incorporating these elements we could begin to explore alternative forms of time series prediction. Essentially, existing prediction mechanisms predict the average across time shifts. However, in some situations the predictions could be considered nonsensical.

For example, let us consider a regression model to predict how much coffee someone drinks between 9am and 9:30am, at work. They set off every day in their car at exactly 8:30am and it takes about 30 minutes ($\pm\tau$) to get to work. The time shift τ could be caused by something as simple as variations in traffic. When they get to work, they always have a coffee. A typical regression might predict 0.5 coffees between 8:30 am and 9am and 0.5 coffees between 9am and 9:30am. Alternative prediction forms such as: 'When they get to work, they will have 1 coffee' might be more useful.

8. Conclusion

We have proposed a form of regression analysis suited to the modelling of stochastic time delay problems. In addition, we have shown the feasibility of the approach and its performance on simulated data. Our approach allows for consistently improved estimation and prediction when the input is affected by noise in the time domain. To our knowledge, the method is novel for this class of problem.

References

[1] C. Spearman, "The proof and measurement of association between two things," *American Journal of Psychology*, no. 15, p. 72–101, 1904.

[2] A. Z. Woldaregay, E. Årsand, S. Walderhaug, D. Albers, L. Mamykina, T. Botsis, and G. Hartvigsen, "Data-driven modeling and prediction of blood glucose dynamics: Machine learning applications in type 1 diabetes," *Artificial Intelligence in Medicine*, vol. 98, pp. 109–134, 2019.

[3] C. Granger, "Investigating causal relations by econometric models and cross-spectral methods," *Econometrica*, no. 37, pp. 424–438, 1969.

[4] S. Almon, "The distributed lag between capital appropriations and net expenditures,," *Econometrica*, no. 33, pp. 178–196, 1965.

[5] L. Ou, M. D. Hunter, and S.-M. Chow, "What's for dynr: A package for linear and nonlinear dynamic modeling in r," *The R Journal*, vol. 11, pp. 1–20, 2019.

[6] F. M. Bianchi, E. Maiorino, M. Kampffmeyer, A. Rizzi, and R. Jenssen, *Recurrent Neural Networks for Short-Term Load Forecasting: An Overview and Comparative Analysis*. 01 2017.

[7] N. Wu, B. Green, X. Ben, and S. O'Banion, "Deep transformer models for time series forecasting: The influenza prevalence case," 2020.

[8] M. Müller, *Dynamic Time Warping. Information Retrieval for Music and Motion*. Springer, 2007.

[9] T. Söderström, *Errors-in-Variables Methods in System Identification*. Springer, 2018.

[10] M. Chandorkar, C. Furtlehner, B. Poduval, E. Camporeale, and M. Sebag, "Dynamictime lag regression: Predicting what and when," *ICLR 2020 - 8th International Conference onLearning Representations*, April 2020.

[11] C. Amornbunchornvej, E. Zheleva, and T. Y. Berger-Wolf, "Variable-lag granger causality for time series analysis," *2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Oct 2019.

[12] W. Deming, "The application of least squares," *Philos. Mag Ser. 7*, vol. 11, pp. 146–158, 1931.

[13] R. Storn and Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, 1997.

[14] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

Email address: aaron1rcl@gmail.com

Email address: juanitorduz@gmail.com
URL: juanitorduz.github.io