MODELLING STOCHASTIC TIME DELAY STRUCTURES FOR REGRESSION

AARON PICKERING

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. 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. 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 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 where the effect of the input is 1, as shown in the image below. 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.
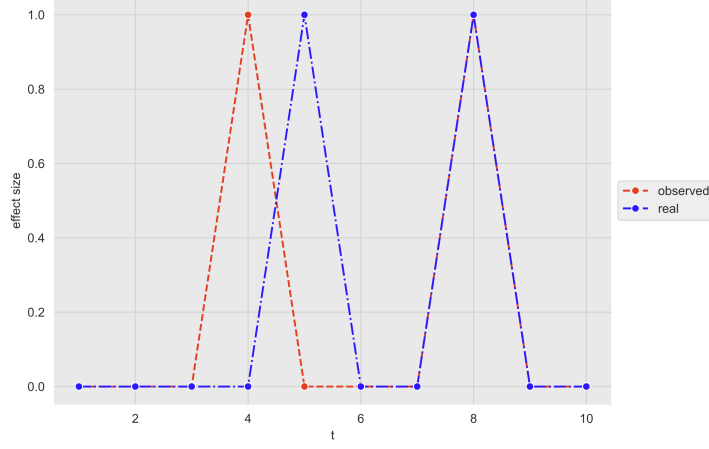

FIGURE 1. Add figure caption

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 multivariate cases, 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. 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 (REF). This is useful when there are measurement errors in the independent variable eg. because your 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.

Let $x : [a, b] \longrightarrow \mathbb{R}$ be a discrete time series with finite domain and we want to determine the injective, functional relationship f to some other variable y. The series $x(t)$ must be a stationary series 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).

For clarity, as per convention, we use the nomenclature 'support' for the elements $f(x(t)) \neq 0$. The size of the support of f(x(t)) should be small relative to the total vector length. We utilise the terminology 'impulse' for an individual element of the support set.

We also assume that the set of τ's, denoted $\mathcal{T}$, is not a constant, but rather a random draw from some discrete distribution (e.g. discrete gaussian, poisson etc). For the discrete gaussian kernel $\mathcal{T} \sim T(0, \sigma)$ or for the poisson distribution $\mathcal{T} \sim Pois(\lambda)$ and $\varepsilon \sim N(0, \sigma_\varepsilon)$.

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

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

then we decompose the vector into a matrix

$$(2) \qquad X(t) = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

where each impulse is treated separately.

Our algorithm then attempts to find the maximum likelihood estimate for the following equation:

$$(3) \qquad y(t) = f(X(t + \mathcal{T})) + \varepsilon$$

where $X(t + T)$ is the time shifted series and $X(t)$ is the observed input.

In order to find the best estimate of f, we would like to find the function f which maximises the joint likelihood of both Tau(?) and ε, the time-domain likelihood and the error likelihood respectively.

Specifically, we maximise the log-likelihood of the following relation:

$$(4) \qquad \max\left(\log\left(\mathcal{L}(\theta_\varepsilon, \varepsilon) \times \mathcal{L}(\theta_\tau, \tau)\right)\right)$$

where θ_ε and θ_τ represent the parameters of the prediction error and time error respectively. For simplicity, assume that the tau and error distributions are independent. To begin with, the values of each individual time shift (τ) are not known. In addition, prediction error distribution ε can only be determined if each τ is known (because for each time shift there is a different prediction and prediction error). Therefore, our algorithm finds the optimum set of time shifts in an inner optimisation, while iteratively searching for the optimum parameters of f in an outer optimisation step.

3. ALGORITHM

Firstly, we define a parametric form and some initial parameters to be estimated for the function f. For simplicity, let's take the univariate linear model where the $f(x(t))$ is parameterised by β, σ_ε(error standard deviation)

and σ_τ (time error standard deviation). First, we 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 far away from the observed impulse and also gives 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.

In principle, we can iterate over a number of values of τ (i.e. optimise) 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

$$(5) \qquad\qquad x(t) = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

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

$$(6) \qquad\qquad X(t + \mathcal{T}) = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

and the effect is therefore

$$(7) \qquad\qquad y = \begin{pmatrix} 0 & 0 & 0 & 2 & 0 & 0 \end{pmatrix}$$

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

We initialize the optimisation algorithm with all parameters set to the mean of the $\mathcal{T}$ distribution (e.g. zero for the zero mean gaussian). Next, we randomly select a small number of impulses, with the value treated as a hyperparameter. The shifts are then randomly changed to a value drawn for the T distribution (proposal vector), and the likelihood (both time shift and prediction error) for the proposal is calculated. If the proposal likelihood is higher than the current maximum likelihood, we update our initial estimate to the current best guess. This procedure is done in a loop with the best guess of the set of T improving over time.

Lastly, we optimise over the set of parameters β, σ_ε and σ_τ, iteratively alternating between the time shift optimisation and the parameter optimisation. For the parameter optimisation, typical methods such as gradient descent, genetic algorithms or annealing can be used. In our implementation we have used the differential evolution algorithm from scipiy.optimize (REF), because of its ability to handle noisy objective functions. In order to improve convergence, we also standardize all input variables to the range 0-1.

Through optimisation, we find the best fit for the values of β, σ_ε and σ_τ. The accuracy of the final parameter estimate is relative to the ratio of the y-axis error and the effect size $\beta X(t)$. As $\sigma_\varepsilon/(\beta X(t))$ tends to infinity, the estimated value of σ_τ tends to zero. In the event that $\sigma_\tau = 0$, the model becomes a 'fixed lag' model, and for the discrete gaussian case the parameter estimate β tends to the standard linear regression coefficient. On the other hand as $\sigma_\varepsilon/(\beta X(t))$ tends to zero, the coefficient estimate tends to the true value. Therefore, the method provides no guarantee on recovering the exact time shifts, only that the estimates are equal to or better than their OLS counterparts.

4. LIMITATIONS

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 ith and jth impulse (row) in $X(t)$, separated by time t, the effect vectors $f(Xi * (t + tau_i))$ and $f(Xj * (t + tau_j))$ 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 density of impulses. Assuming that we wish to split $X(t)$ into independent matrices $Un * (t)$ and $Vm * (t)$, if we choose the split locations such that $\forall_i \epsilon n and \forall_j \epsilon_m$, $f(Ui * (t + tau_i))$ is linearly independent from $f(Vj * (t + tau_j))$, then segments will also be independent and the procedure unique. In practice, it may not be possible to guarantee complete independence of every segment, in which case results will vary. After segmentation, each matrix can be optimised independently, in parallel, making the inner optimisation procedure tractable for longer length sequences.

Finally, a note on problem constraints. The likelihood estimates are dependent on the inner $(\mathcal{T})$ optimisation procedure. There is no guarantee that the global maximum will be found during this process, particularly for sequences with a high density of impulses.

For example, consider the example sequence

$$(8) \qquad x(t) = \begin{pmatrix} 0.5 & 0.6 & 0.4 & 1 & 2 & 1 & 0.3 & 0.2 & 0.5 & 0.8 \end{pmatrix}$$
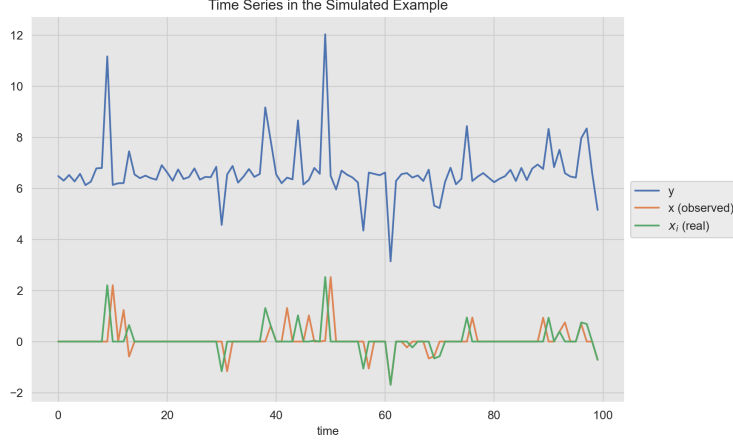
FIGURE 2. Simulated Example Data

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.

5. EXPERIMENTS

The following section demonstrates a simulated univariate example. Full code to reproduce the example can be found via TVS Regression on Github.

Figure 2. shows the simulated time series.

The input signal has 17 non-zero impulses which have been drawn from the standard normal distribution. The system is 'off' when $x(t) = 0$. In other words, when $x(t) = 0$, $f(x(t)) = 0$.

The true shift distribution $\mathcal{T}$ is given by $\mathcal{T} \sim N(0, 2)$. There is one value of τ for each of the 17 impulses.

The green line represents the shifted series, corresponding to the time at which the effect occurs. The blue line is the output 'y'.

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

$$(9) \qquad\qquad\qquad y = 2X(t + \mathcal{T}) + 6.5$$

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

After fitting the model, we obtain the following results. Figures 4 and 5 show the model fit (denoted TVS) and a comparison with standard OLS. Figures 6 and 7 fit convergence and scatterplot respectively.
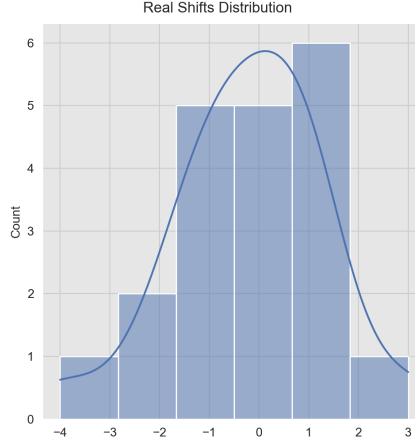
FIGURE 3. Real Shift Distribution $\mathcal{T}$

FIGURE 4. TVS Model Fit $\mathcal{T}$

The true set of $\mathcal{T}$ is:
Compared to the estimated $\mathcal{T}$:

6. FUTURE WORK

7. CONCLUSION

We have proposed a form of regression analysis for handling stochastic time delay problems. We have shown the feasibility of the approach and its performance on simulated data. The approach allows for better 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.

Email address: aaron1rcl@gmail.com