Notes on Filtering Piecewise Linear Data

Philip Bretz

November 19, 2020

1 Filtering

1.1 Introduction

While analyzing time dependent data, we frequently encounter the situation where the data has an underlying trend obscured by noise. The goal then, is to extract that trend, or smooth out the noise. This is filtering.

There are a variety of tools for doing this. However, a particularly powerful method is the Kalman filter. In the Kalman filter, we treat the underlying trend as an unknown state. We then infer information about that state by applying Bayesian methods to the observed data.

The Kalman filter requires that the system in question follows a particular model. The less the data follows that model, the worse the filter is at extracting the trend. In some cases, this necessitates the use of alternative approaches, such as applying transformations to the data or using the more versatile particle filter.

1.2 General Filtering

A standard state-space model is a sequence of states θ_t and observations Y_t. Typically, the states are unobservable, but of interest. It has the following dependence structure:

1. θ_t is dependent only on θ_{t-1},
2. Y_t is dependent on θ_t, and
3. $Y_t|\theta_t$ is independent of $Y_{t-1}|\theta_{t-1}$.

Filtering is the process of finding the distribution of the state θ_t given the observations $Y_{1:t}$. The standard framework is recursive, where $\pi(\theta_{t-1}|Y_{1:t-1})$ is known and used in a three step process to adjust to the new piece of data Y_t and get the distribution $\pi(\theta_t|Y_{1:t})$. The following is from p. 51 - 52 of *Dynamic Linear Models* [1]:

1. The first step is finding the one-step ahead state probability density. We need to know the distribution of θ_{t-1}, $\pi(\theta_{t-1}|Y_{1:t-1})$ as well as the transitional conditional $\pi(\theta_t|\theta_{t-1})$. It is computed by

$$\pi(\theta_t|Y_{1:t-1}) = \int \pi(\theta_t|\theta_{t-1})\pi(\theta_{t-1}|Y_{1:t-1})d\theta_{t-1}. \tag{1}$$

2. The next step is finding the one-step ahead observation probability. We need the conditional $\theta_t|Y_{1:t-1}$ from the previous part as well as the conditional $\pi(Y_t|\theta_t)$. It is computed by

$$\pi(Y_t|Y_{1:t-1}) = \int \pi(Y_t|\theta_t)\pi(\theta_t|Y_{1:t-1})d\theta_t. \tag{2}$$

3. The last step is computing the desired filtering density. This is found simply by applying Bayes' rule using the previous parts:

$$\pi(\theta_t|Y_{1:t}) = \frac{\pi(Y_t|\theta_t)\pi(\theta_t|Y_{1:t-1})}{\pi(Y_t|Y_{1:t-1})}. \tag{3}$$

At the beginning, when no observations are available, θ_0 is given a chosen prior distribution. Also, note that equation 2 will yield a numerical value, since Y_t is a known data point. Alternatively, we can see that it is simply the denominator in equation 3, which is a normalizing constant.

1.3 Kalman Filter

Dynamic Linear Model The specific case when all the conditionals are linear and Gaussian is known as a Dynamic Linear Model (DLM). In a DLM, all posteriors are also Gaussian and can thus be described purely by two parameters, mean and variance. When applying this recursive filtering process in that case, one only needs to keep track of those two parameters. Moreover, those two parameters behave in a set manner that depends only on the model parameters and the new data, Y_t. This is the Kalman filter.

The standard state-space form of a DLM is

$$\begin{aligned} Y_t &= F_t\theta_t + \nu_t \\ \theta_t &= G_t\theta_{t-1} + \omega_t \end{aligned} \tag{4}$$

where $\nu_t \sim N(0, V_t)$ and $\omega_t \sim N(0, W_t)$. In this there are four parameters, F_t, G_t, V_t, W_t that are chosen beforehand according to knowledge about the underlying system and/or statistical estimation from subsets of the data. They are allowed to be time-varying, but again, must be predetermined.

In our specific case, Y_t and θ_t are scalars, but in general, a DLM allows for these to be vector valued.

Kalman Filter The Kalman filter updates sequentially, requiring knowledge of the distribution of the state at the previous time given the data up to that time. I.e., before updating at time t, we need $\theta_{t-1}|Y_{1:t-1} \sim N(m_{t-1}, C_{t-1})$. We keep track of how the parameters m_t and C_t (the mean and variance of the state distribution) evolve over time. Our goal in the update process is to get the values m_t and C_t that are the correct parameters for the distribution $\theta_t|Y_{1:t} \sim N(m_t, C_t)$.

For the above DLM, the Kalman filter updates in this manner[1].

First, the one-step ahead state distribution is Gaussian with parameters:

$$a_t = G_t m_{t-1}$$
$$R_t = G_t C_{t-1} G_t' + W_t.$$

Next, the one-step ahead predictive distribution is Gaussian with parameters:

$$f_t = F_t a_t$$
$$Q_t = F_t R_t F_t' + V_t.$$

Last, the filtered distribution is Gaussian with parameters:

$$m_t = a_t + R_t F_t' Q_t^{-1}(Y_t - f_t)$$
$$C_t = R_t - R_t F_t' Q_t^{-1} F_t R_t.$$

Typically, we keep track of the quantity $Y_t - f_t$, which is exactly the forecast error.

2 Our Data

2.1 Context

The data I am examining is drawn from a stick-slip simulation where constant shearing force is applied to a system of granular media. The simulations are designed to mimic earthquake behavior, where pressure in the system builds up due to shearing force, eventually resulting in a slip, followed by another period of pressure build up, and so on.

Using persistent homology, my colleagues calculated a persistence diagram for each frame of the simulation. The data that I am looking at is, for each time, the Wasserstein distance between the current persistence diagram and the persistence diagram of the previous frame. Essentially, this is a measure of how the system changes as a whole at each time.

With this time series that gives a broad overview of the system, my colleagues are interested in predicting when an upcoming slip will occur. However, the data appears to have a

great deal of noise around an underlying curve. Consequently, this is an extremely natural application of the Kalman filter, where we can remove the noise in order to analyze the underlying curve itself.

2.2 Parameters

I did not apply the Kalman filter directly to the W_2 (Wasserstein) distances. The W_2 distances appear to have an exponentially decreasing behavior during a stick, punctuated by wild jumps during a slip; hardly linear. Instead, I took the logarithm of the distances, yielding data that is (roughly) piecewise linear.

Before applying the Kalman filter, we need to set the four parameters F_t, G_t, V_t, W_t. I chose $F_t = 1, G_t = 1, V_t = 0.5, W_t = 0.01$. What do each of these mean? $F_t = 1, V_t = 0.5$ means that the observations are based directly on the state with observation noise that is Gaussian with variance 0.5. $G_t = 1, W_t = 0.01$ means that the state evolves by only small (normal, variance 0.01) jumps from the previous state. Heuristically, these choices mean that the filter treats changes in the observations as arising from observation noise unless there is strong evidence that the true state has changed.

2.3 Results

Applying the filter to the logarithm of the W_2 distances yields excellent smoothing. Additionally, when I examined the forecast error in a few examples, I saw distinct behavior during a stick, which changed leading up to a slip, and then during the slip itself. Below is the filtered log-transformed data and the forecast error, with the slip occurring between the black lines.

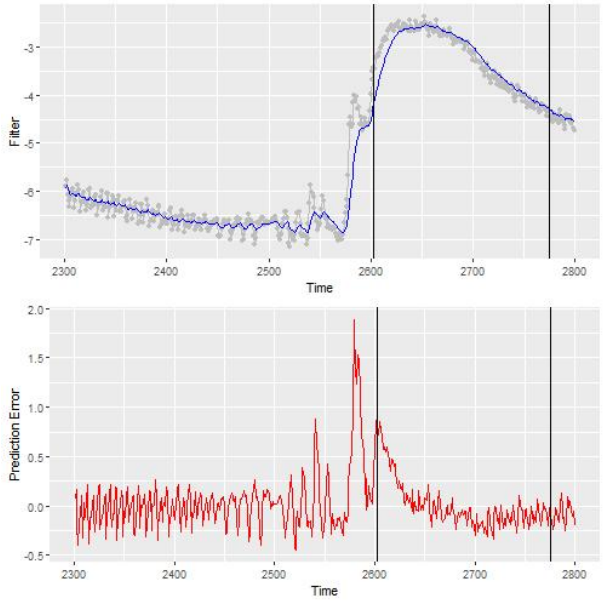

Figure 1: Kalman Filter

2.4 Issues

Unfortunately, we can note that the filter lags behind the data, especially during the slip. This is entirely expected. Why?

The Kalman filter, with the parameters that I used, assumes that the data is basically a random walk with observational noise. This is, of course, clearly not the case. During the stick, the data has a distinctly decreasing trend, while at the beginning of the slip, the data has a sharply increasing trend. Neither of these are a random walk, so the Kalman filter has trouble smoothing the curve.

A possible solution is to incorporate a drift term that is allowed to be time-varying. If we allow the drift term to learn, in a Bayesian manner, when the data is decreasing/increasing and how much, this will give a more responsive filter. A potential method is incorporating Bayesian changepoint detection [2].

References

[1] Giovanni Petris, Sonia Petrone, and Patrizia Campagnoli. *Dynamic Linear Models with R*. Springer, New York, NY, 2009.

[2] Ryan Prescott Adams and David J.C. MacKay. Bayesian online changepoint detection. *Arxiv*, 2007.