

State Space Models

Eric Lee, 11793249

2022-09-05

INTRODUCTION

A state-space model is a linear representation of a dynamic system in either continuous or discrete form. This dynamic system is driven by unobservable variables represented in figure 1 by the X vector. We can only observe X indirectly through a series of variables that are associated with X , which is represented by the Y vector. The aim of state space modeling is to make inferences about the unobservable variables (our state process) from the observable variables. This is done by the state space model, which is a function that describes the relationship between the observable and latent variables (Lesniewski, 2019). Given new observations we want to update our estimate of where we are in the state process.

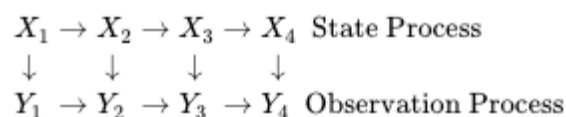


Figure 1

MATHEMATICS OF STATE SPACE MODELS

Below is a general form of a one-dimensional state space model:

$$\begin{aligned} y_t &= Z^T \alpha_t + \epsilon_t && \text{Observation equation} \\ \alpha_{t+1} &= T_t \alpha_t + R_t \eta_t && \text{State equation} \\ \epsilon_t &\sim N(0, \sigma_t^2) \\ \eta_t &\sim N(0, Q_t) \end{aligned}$$

Where y_t is the observed value of a series at time t , α_t is a m -dimensional state vector, Z_t is a m -dimensional output vector, T_t is a $m \times m$ transition matrix, R_t is a $m \times q$ control matrix, ϵ_t is the observation error, and η_t is the error for the state equation (Kotze, n.d.). Essentially, the observation equation connects the observed data y_t with the unobserved latent state α_t . The above equations are flexible enough to incorporate many types of time series models such as, ARIMA and AR. To fit such models, we would just have to place it in the same mathematical structure as above.

AUTOREGRESSIVE ORDER ONE PROCESS IN STATE SPACE FORM

A state space model is a form of an autoregressive time series model. It differs in the traditional AR(p) model in that it typically only includes the first autoregressive component. The distinguishing feature of state space models are that it allows us to integrate uncertainty in the observation process, which is done by modeling the state process and the observation process separately. The state process is how the true system changes over time, whereas the observation process models how observations are made on the underlying state process. Thus, we can incorporate factors such as observation error and indirect observation. Below is an example of how an autoregressive process of order one can be modeled using our general state space form.

$$\begin{aligned} &\textbf{State Model} \\ x_{t+1} &= \phi_1 x_t + w_t & (1) \\ W_t &\sim N(0, \sigma_w^2) \end{aligned}$$

$$\begin{aligned} &\textbf{Observation Model} \\ y_t &= Ax_t + v_t & (2) \\ \epsilon_t &\sim N(0, \sigma_v^2) \end{aligned}$$

Figure 2

In the simplest form, the state process may be modeled by an AR(1) process, which is shown by equation one. We may state the observation model as a function of the actual state at time t , which some noise, which is shown by equation two. An example we can use to illustrate the equations would be to apply it regarding the economy. The economy is a ongoing process that cannot directly be observed. We can only glimpse into the state process of the economy through indirect measures such as gross domestic product (GDP), inflation and growth in money supply. In figure 2, the state model would represent the economy. The observation model can be any of the mentioned measures of the economy. The important points are that these observations are all driven by the same state process, x_t . Although, they will have different errors terms, as some measures are less noisy than others. Meaning they are more like the state process. For example, GDP growth will be a less error prone measure of the economy than ice cream sales. Even though both are fundamentally driven by the same underlying force.

KALMAN FILTERS

The whole premise of state space modelling is to infer about the unobservable state, x_t , using the observations y_s . This is achieved using Kalman filters (Shumway & Stoffer, n.d.). Kalman filters have three main types of inference. These being:

1. Predictions: when $t > s$, we are predicting the future values of our observed or state variable.
2. Filtering: when $t = s$, we are trying to remove measurement error associated with our observed variable to get a clearer estimate of the state variable.
3. Smoothing: when $t < s$, we are estimating the past hidden states from the measurements up to a certain period.

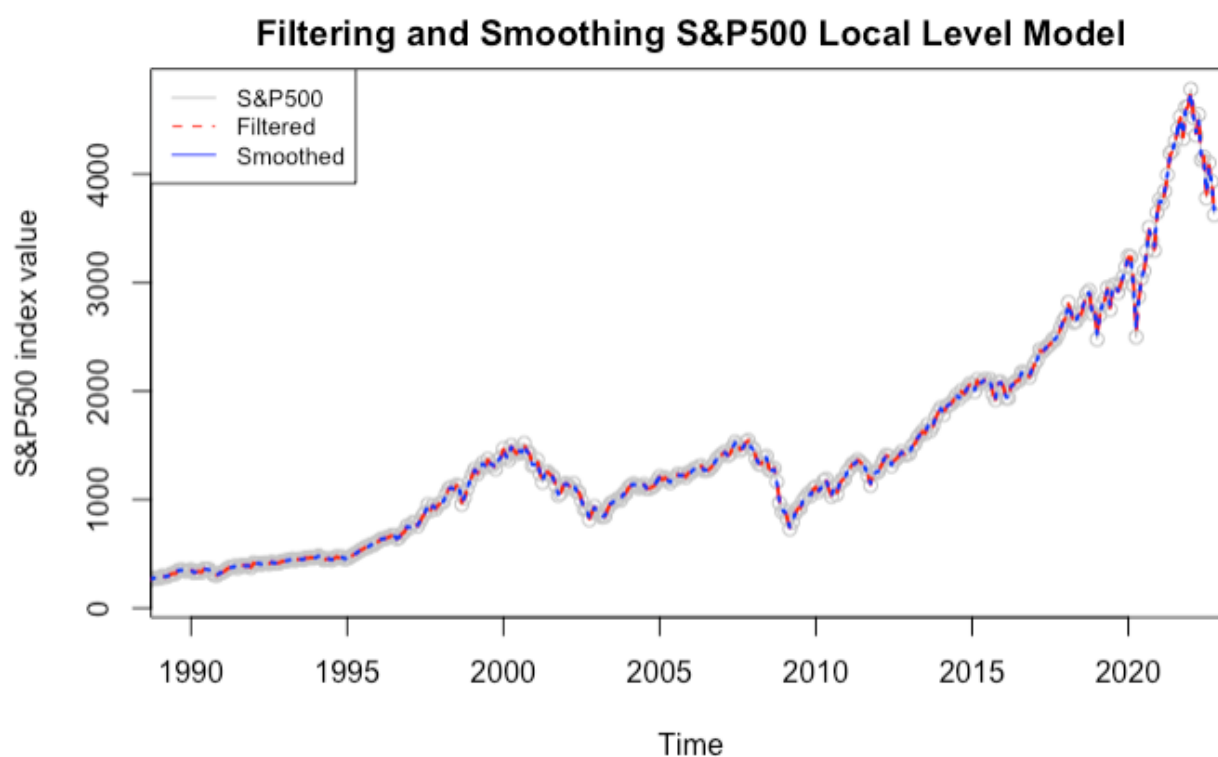
The mathematics of Kalman Filters is quite rigorous and outside the scope of this text. Instead, we will take a more qualitative approach and explain the intuition, so that we can apply it later when analyzing our data. The Kalman Filter can be thought of as an iterative algorithm that dynamically adapts to new information by constantly adapting parameters. First, we initialized the mean and covariance of the series through measurements of the underlying series. After, we estimate the values of x_t and y_t in figure 2. Once we have an estimate, the covariance is compared to the real time series value and a parameter called K, the Kalman Gain, is adjusted and our prior estimate is updated (Trading, 2009).

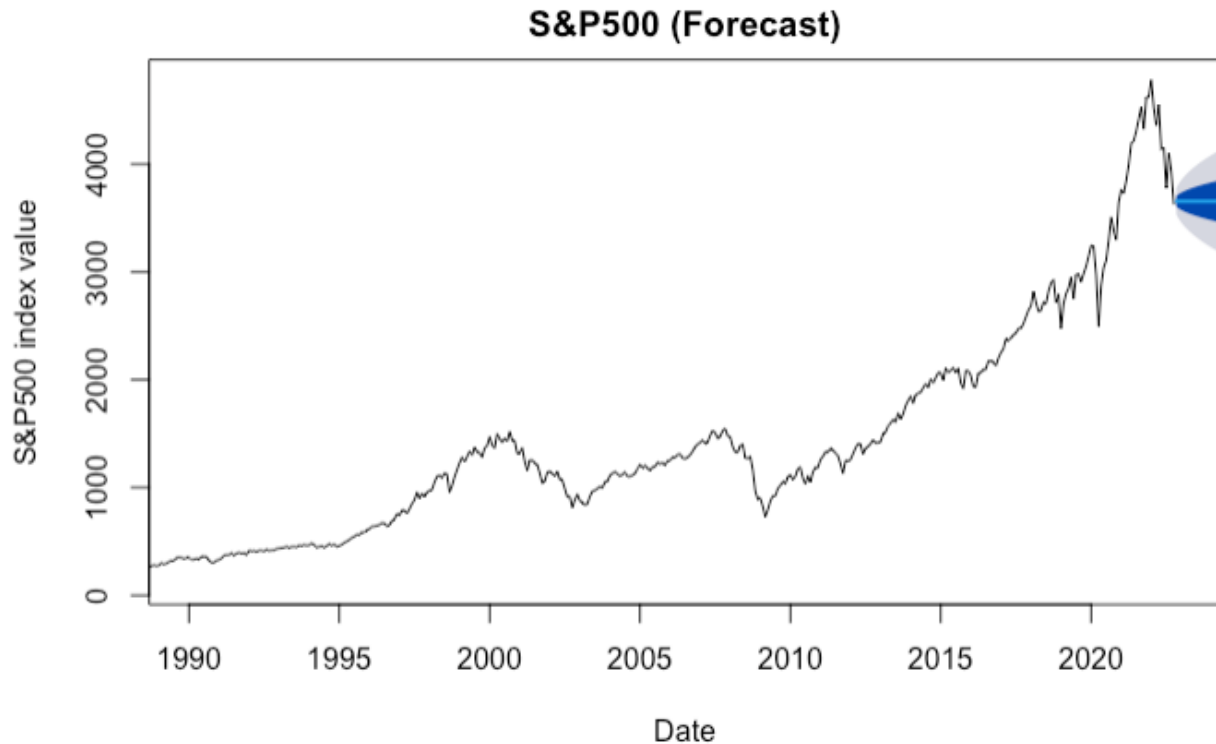
The intuition behind Kalman filters is that it estimates the trajectory of a moving object, which applies to time series objects. Kalman filters infer about the true state, x_t , from the noisy estimate, which is our observed variable, y_t . The word filter is used because it filters out the noise in our observation variable to reveal the underlying structure of the systems that drives it. Essentially, Kalman filters smooth the underlying series in a different way than standard moving average techniques. Typically, moving averages weight past data points and errors to achieve a smooth line. Whereas Kalman filters can acknowledge some data points are just noise. The classic example given regarding Kalman filters are GPS systems. GPS use satellites to triangulate your location and map your desired direction. The signals coming from these satellites are very noisy. For example, the direction from the University to the BP on the intersection of Clyde and Fendalton road will very jagged. It's only straight and even because we filter the noise through Kalman filters.

DATA

The data we are going to use to fit a space state model is the Standard & Poor's 500 index, which is a index that tracks the 500 largest companies in the United States. It is thought of as a leading indicator of the economy as financial market participants try to anticipate economic moves.

Therefore, the S&P500 is a good gauge for our latent variable, which is the economy. We will only fit a local level model, meaning we observe a time series y_t which is a sum of another time series, X_t with white noise. This is the same set up as shown in figure 2. The model will be fit using the StuctTs function from base R. This method was chosen for convenience, as we do not need to separately state the vector of parameters associated with the model.





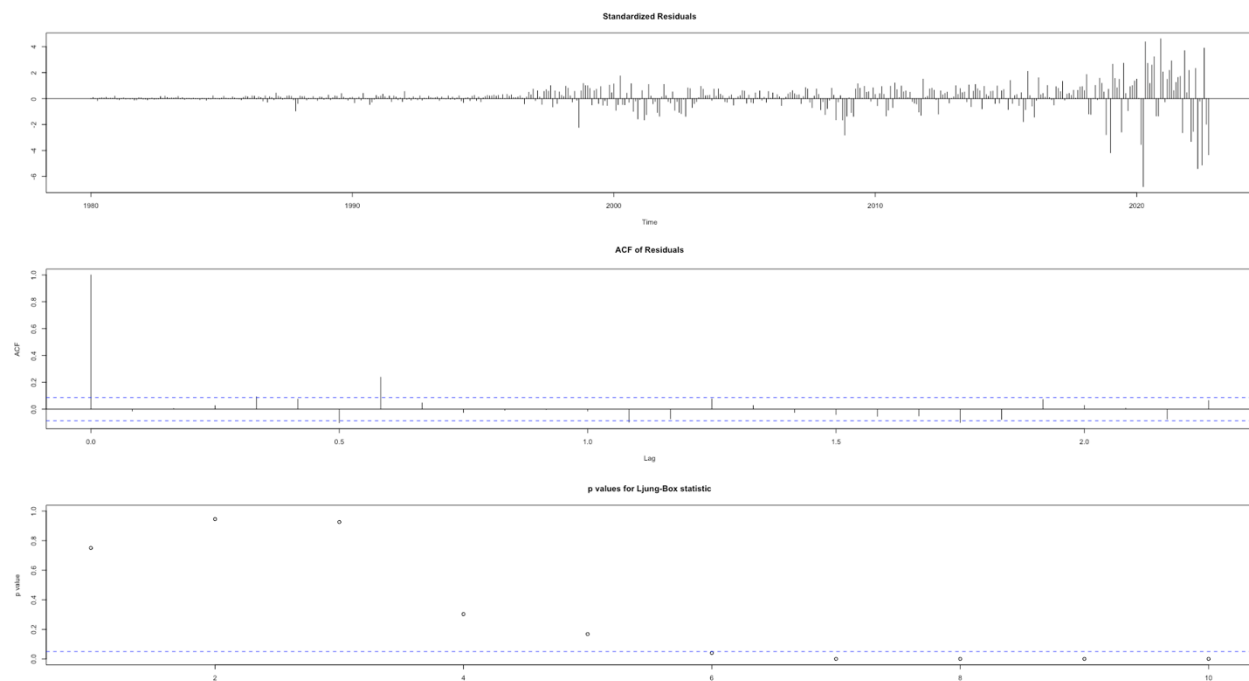
From the above series we can see that, the filtered and smooth value has fit the underlying S&P500 model very tightly. It seems the Kalman filter has failed to infer the underlying state process that drives the S&P500, which can be explained with financial theory. One of the most important/controversial financial theory is the Efficient Market Hypothesis. Originally, thought of by Eugene Fama, the theory states all available information about a company is already reflected in the stock price. Therefore, market participants cannot consistently outperform the index with available information (Fama, 1970). Let's say there is a piece of information about Apple that they have exceeded their projected revenue for the fourth quarter. If the stock price does not reflect the information, then people can trade on it to make a profit. As people do this, the stock price will head higher until the information is baked in and is no longer of use. Thereby, past information about a stock has no effect on future price. The financial markets are forward looking and try anticipating these moves before they happen. Thus, our structural model

is implying that there is very little information about stock prices from their prior estimates, which results in the smoothing becoming the previous estimate. This fits the random walk description of stock price behavior well. When looking at the forecast, it is rather small and quite uninformative. The forecast is a horizontal line from the current price with some variance around it, which again suggest there is not much signal in past stock prices.

MODEL OUTPUT

```
## Call:
## StructTS(x = spx_data_ts, type = "level")
##
## Variances:
##   level   epsilon
## 4449.4    523.2
```

RESIDUAL PLOTS



APPENDIX

Fama, E. F. (1970). *Efficient Capital Markets: A Review of Theory and Empirical Work*. JSTOR.

<https://www.jstor.org/stable/2325486?origin=crossref>

Kotze, K. (n.d.). *State-Space Modelling*. Github. <https://kevinkotze.github.io/ts-4-state-space/#fnref4>

Lesniewski, A. (2019). *Time Series Analysis*. Baruch College. https://mfe.baruch.cuny.edu/wp-content/uploads/2014/12/TS_Lecture5_2019.pdf

Petris, G., & Petrone, S. (2011). *State Space Models in R*. Journal of Statistical Software. <https://www.jstatsoft.org/article/view/v041i04>

Shumway, R. H., & Stoffer, D. S. (n.d.). *Time Series Analysis and Its Applications With R Examples* (3rd ed.). Springer.

Trading, I. (2009). *The Kalman Filter for Financial Time Series*. R-bloggers. <https://www.r-bloggers.com/2010/05/the-kalman-filter-for-financial-time-series/>

Yu, O. (2019). *Structual Time-Series Models*. Github. <https://oliviayu.github.io/post/2019-03-21-bsts/>

CODE

```
# loading and transforming data
spx_data <- read.csv("S&P500.csv")
spx_data$Date <- as.Date(spx_data$Date, format = "%m/%d/%Y")
spx_data$Open <- as.numeric(gsub(",", "", spx_data$Open))
spx_data <- spx_data[spx_data$Date >= "1980-01-01", ]

# changing data to ts object
spx_data_ts <- ts(rev(spx_data$Open), start = c(1980), frequency = 12)

# fitting state space model
spx_state_space <- StructTS(spx_data_ts, type = "level")
spx_state_space

library(forecast)
```



```

plot(spx_data_ts,
     xlim = c(1990, 2022),
     type = "o",
     col = "grey",
     ylab = "S&P500 index value",
     main = "Filtering and Smoothing S&P500 Local Level Model")
lines(fitted(spx_state_space), lty = "dashed", lwd = 2, col = "red")
lines(tsSmooth(spx_state_space), lty = "dotted", lwd = 2, col = "blue")
legend("topleft", legend=c("S&P500", "Filtered", "Smoothed"),
      col=c("grey", "red", "blue"), lty=1:2, cex=0.8)

plot(forecast(spx_state_space, level = c(50, 90)), xlim = c(1990, 2023),
     xlab = "Date",
     ylab = "S&P500 index value",
     main = "S&P500 (Forecast)")

tsdiag(spx_state_space)

```