Stock Option Pricing Using Bayes Filters

Author: Ziyang Bob Ding

April 20, 2020

Abstract

In this project, I utilize EKF (extended Kalman Filter), which is an extension of Kalman Filter, with a combination of the GARCH(1,1) model that generates the dynamics for implied volatilities, to model SPX500 options (European) and its latent volatility dynamics. For variable estimation in GARCH(1,1) model and latent volatility process, the MCMC algorithm is used together with Extended Kalman Smoothing, Linear Regression, and Inverse Gamma update. The finalized model is able to conduct volatility prediction up to days in which the stock price is observed.

1 Introduction

Option is a financial contract that gives the holder the right to buy (*call option*) or sell (*put option*) an asset for a certain price (*strike price*) on (*European style*) or before (*American style*) a certain date (*maturity date*). Option trading allows the investors to bet on future events and to reduce the financial risks. However, what the contract is worth is anything but trivial. In the early 1970 s, Fischer Black, Myron Sc holes, and Robert Merton made a major breakthrough in the pricing of stock options. This involved the development of what has been known as the Black-Scholes (BS) formula, a formula that prices options based on stock price, stock return volatility, strike price, maturity time, and risk free interest rate. Among all these required parameters, the stock return volatility is the only not observable, and therefore needs be inferred.

There are basically two ways of determining the variance v of stock returns, one of which is to estimate it from time series of the stock prices, and the other to calibrate the model prices to market option prices. The v, and hence the model option prices, obtained in these two ways may not coincide, and this is often the case. It is not surprising that calibrated model prices are closer to observed market prices, but they say little about the connection to the actual dynamics of the underlying asset. It is, after all, the value of the underlying asset at expiration that determines the option pay-off.

Therefore, this project explores a calibration between stock return information and market option price information by combining GARCH model and Extend Kalman Filter to demonstrate the feasibility of such volatility modeling approach.

2 GARCH(1,1)

2.1 Variable Explanation

- p_t Stock price on day t.
- u_t Stock return on day t. $u_t = \frac{p_t - p_{t-1}}{p_{t-1}}$
- θ_t Stock volatility on day t

2.2 Sub-model Specification

$$\begin{aligned}
\theta_t &= \gamma V_L + \alpha u_t^2 + \beta\theta_{t-1} + w_t \\
w_t &\sim \mathcal{N}(0, W) \\
\gamma, \alpha, \beta &> 0 \\
\gamma + \alpha + \beta &= 1
\end{aligned}$$

Where V_L is the long-run average variance rate V_L. To better help our computation, we may relax our boundary constraint condition by treating γV_L as ω, therefore the model becomes:

$$\theta_t = \omega + \alpha u_t^2 + \beta\theta_{t-1} + w_t \tag{1}$$

$$w_t \sim \mathcal{N}(0, W) \tag{2}$$

$$\omega, \alpha, \beta > 0 \tag{3}$$

$$\alpha + \beta < 1 \tag{4}$$

3 Black-Schole's Formula

In this project, I will be dealing with European Option solely, considering it is easier for computation. Black-Schole's Formula is used to measure an European vanilla option's price based on the related assets (can be a portfolio of assets) return and volatility. The formula assumes the following:

1. **The risk-free rate and volatility of the underlying are known and constant.**
2. **The returns on the underlying are normally distributed.**

BS formula takes the following form:

$$\begin{aligned}
y_{ti} &= p_t\Phi(d_+) - K_{ti}e^{-r_tT_{ti}}\Phi(d_-) \\
d_+ &= \frac{\ln\left(\frac{p_t}{K_{ti}}\right) + \left(r_t + \frac{\theta_t}{2}\right)T_{ti}}{\sqrt{\theta_t T_{ti}}} \\
d_- &= \frac{\ln\left(\frac{p_t}{K_{ti}}\right) + \left(r_t - \frac{\theta_t}{2}\right)T_{ti}}{\sqrt{\theta_t T_{ti}}}
\end{aligned}$$

Where p_t is asset price; r_t is risk-free interest rate; T_{ti} is exercise time; K_{ti} is strike price; θ_t is volatility; Φ is CDF of normal.

4 Extended Kalman Filter

4.1 Sub-model Specification

$$\begin{aligned}
\theta_t &= g_t(\theta_{t-1}, u_t, w_t) \\
y_{it} &= f_{ti}(\theta_t, p_t, K_{ti}, T_{ti}) + \nu_t \\
\nu_t &\sim N(0, v)
\end{aligned}$$

Where

$$g(\theta_{t-1}, u_t, w_t) = \omega + \alpha u_t^2 + \beta\theta_{t-1} + w_t$$

is the GARCH(1,1) process. In other word, the GARCH(1,1) is representing the dynamics inside the EKF. And

$$f_{ti}(\theta_t, p_t, K_{ti}, T_{ti}) = BS(\theta_t, p_t, K_{ti}, T_{ti})$$

is the Black Scholes formula. Here, because on each day, the stock market has several several different Option choices. In order to incorporate all of their information, we have multiple outputs on each timestep as our target variable y_{it}, the option price on day t.

4.2 Forward Filtering

The theory of EKF lies in linearlizing the EKF at $E[\theta_{t-1}|\mathcal{D}_{t-1}] = m_{t-1}$ and $E[\theta_t|\mathcal{D}_{t-1}] = a_{t0}$ using Taylor first degree expansion. Therefore, we have

$$\begin{aligned}
\mathbb{P}(\theta_{t-1}|\mathcal{D}_{t-1}) &\sim \mathcal{N}(m_{t-1}, C_{t-1}) \\
\mathbb{P}(\theta_t|\mathcal{D}_{t-1}) &\sim \mathcal{N}(a_t, R_t) \\
a_t &= g_t(m_{t-1}) \\
R_t &= G_t C_{t-1} G_t^T + W \\
m_t &= \frac{R\sum_{i=1}^{n_t} F_{ti}\xi_{ti} + v a_t}{R\sum_{i=1}^{n_t} F_{ti}^2 + v} \\
C_t &= \frac{vR}{R\sum_{i=1}^{n_t} F_{ti}^2 + v}
\end{aligned}$$

Where

- $G_t = \frac{\partial g_t}{\partial \theta_{t-1}}\Big|_{\theta_{t-1}=m_{t-1}} = \beta$
- $F_{ti}^\intercal = \frac{\partial f_{ti}}{\partial \theta_t}\Big|_{\theta_t = a_t} = \frac{1}{2}\sqrt{\frac{T_{ti}}{a_t}}\left[\ln(\frac{p_t}{K_{ti}})a_t^{-1}T_{ti}^{-1} + r_t a_t^{-1}\right]\left[K_{ti}e^{-r_t T_{ti}}\phi(d_-) - p_t\phi(d_+)\right] +$ $\frac{1}{4}\sqrt{\frac{T_{ti}}{a_t}}\left[K_{ti}e^{-r_t T_{ti}}\phi(d_-) + p_t\phi(d_+)\right]$
- $\xi_{ti} = y_{ti} - h_{ti}(a_t)$
- $h_{ti}(a_t) = f_{ti}(a_t) - F_{ti}^T(a_t)$

(**Note:**) The update equation is not available for vector case as the closed form inverse matrix doesn't exist. While in our case, all the transitions are scalar, we're able to write out the correction step in closed form.

A detailed derivation can be found in Appendix.

4.3 Backward Smoothing

Backward Smoothing is particularly important when we're sampling the latent $\theta_{1:T}|\mathcal{D}_T$ for the MCMC parameter estimation part. Similar to forward filtering step, the backward smoothing also uses Taylor first degree expansion. Therefore we have:

$$\begin{aligned}\mathbb{P}(\theta_t|\theta_{t+1:T}\mathcal{D}_T) \sim \mathcal{N}(m_t^*, C_t^*)\\ m_{t-1}^* &= m_{t-1} + J_{t-1}(\theta_t - g_t(m_{t-1}))\\ C_{t-1}^* &= C_{t-1} - J_{t-1}R_tJ_{t-1}^\top\end{aligned}$$

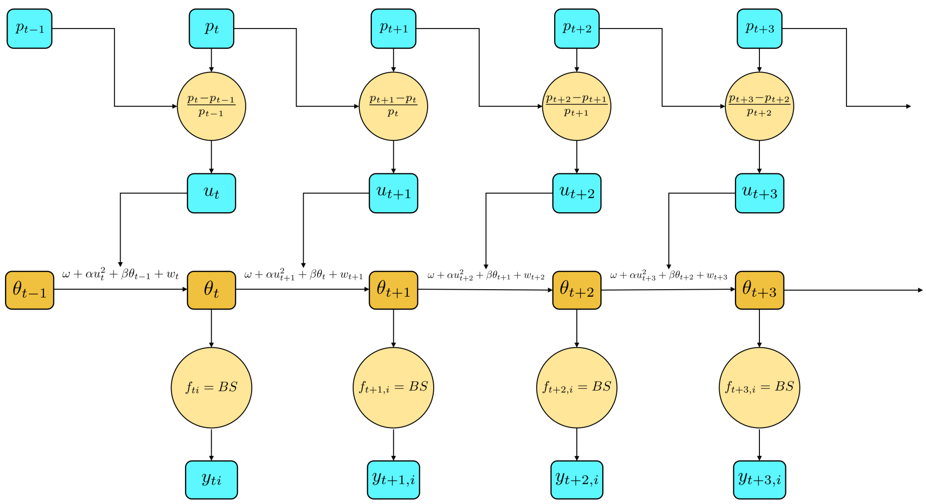
Where

- $m_{t-1} = \mathbb{E}[\theta_{t-1}|\mathcal{D}_{t-1}]$
- $C_{t-1} = \text{Var}[\theta_{t-1}|\mathcal{D}_{t-1}]$
- $R_t = \text{Var}[\theta_t|\mathcal{D}_t] = G_tC_{t-1}G_t^\top + W$
- $J_{t-1} = C_{t-1}G_t^\top R_t^{-1}$

A detailed derivation can be found in Appendix.

5 Model Illustration

Attached is a illustration for the model:



6 Parameter Inference

The Model requires statistical inference uppon the following parameters:

$$\{\omega, \alpha, \beta, W, v\}$$

We develop the MCMC algorithm that

1. Set priors $\mathbb{P}((\omega,\alpha,\beta)^\intercal), \mathbb{P}(W), \mathbb{P}(v)$. Random initialize.
2. Forward Filtering: Update all $m_t, C_t, \forall t \in 1:T$
3. Backward Smoothing: Sample $\theta_{1:T}$ from $\mathbb{P}(\theta_{1:T}|\mathcal{D}_T)$ by recursively sample θ_{t-1} from $\mathbb{P}(\theta_{t-1}|\theta_t, \mathcal{D}_T)$ using EKF smoothing.
4. Linear Regression with Constrains: Sample $(\omega,\alpha,\beta)^\intercal, W$ from $\mathbb{P}(\omega,\alpha,\beta,W|\theta_{1:T},\mathcal{D}_T,v) = \mathbb{P}(\omega,\alpha,\beta,|W,\theta_{1:T})\mathbb{P}(W|\theta_{1:T})$ with sequential inverse CDF method for constraints of equation (3) and (4).
5. Inverse Gamma: Sample v from $\mathbb{P}(v|\theta_{1:T},\mathcal{D}_T,W,\omega,\alpha,\beta) = \mathbb{P}(v|\theta_{1:T},\mathcal{D}_T)$
6. Repeat 2,3,4,5 until converges.

6.1 Sampling $(\omega,\alpha,\beta)^\intercal$, and $W = \phi^{-1}$

This is a simple linear regression $\boldsymbol{\theta} = \mathbf{X}\boldsymbol{\beta} + w_t$ with design matrices as

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \vdots \\ \theta_T \end{bmatrix} \qquad \mathbf{X} = \begin{bmatrix} 1 & u_1^2 & \theta_0 \\ 1 & u_2^2 & \theta_1 \\ 1 & u_3^2 & \theta_2 \\ 1 & u_4^2 & \theta_3 \\ 1 & u_5^2 & \theta_4 \\ 1 & \vdots & \vdots \\ 1 & u_T^2 & \theta_{T-1} \end{bmatrix} \qquad \boldsymbol{\beta} = \begin{bmatrix} \omega \\ \alpha \\ \beta \end{bmatrix}$$

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\beta};\boldsymbol{\theta},\mathbf{X}) &\propto \phi^{\frac{T}{2}} \exp\{-\frac{1}{2}\phi(\boldsymbol{\theta}-\mathbf{X}\boldsymbol{\beta})^\intercal(\boldsymbol{\theta}-\mathbf{X}\boldsymbol{\beta})\} \\
\boldsymbol{\beta}|\phi &\sim \mathcal{N}\left(\boldsymbol{\mu}_0, \boldsymbol{\Lambda}_0^{-1}/\phi\right) = \mathcal{N}((0.5,0.5,0.5)^T, 1/3\phi^{-1}\mathbf{I}) \\
\phi &\sim \mathbf{G}\left(a_0 = \frac{v_0}{2}, b_0 = \frac{v_0 s_0^2}{2}\right) = \mathbf{G}(1,1) \\
\boldsymbol{\mu}_n &= \left(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \boldsymbol{\Lambda}_0\right)^{-1}\left(\boldsymbol{\Lambda}_0\boldsymbol{\mu}_0 + \mathbf{X}^{\mathrm{T}}\boldsymbol{\theta}\right) \\
\boldsymbol{\Lambda}_n &= \left(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \boldsymbol{\Lambda}_0\right) \\
a_n &= a_0 + \frac{T}{2} \\
b_n &= b_0 + \frac{1}{2}(\boldsymbol{\theta}^\intercal\boldsymbol{\theta} + \boldsymbol{\mu}_0^\intercal\boldsymbol{\Lambda}_0\boldsymbol{\mu}_0 - \boldsymbol{\mu}_n^\intercal\boldsymbol{\Lambda}_n\boldsymbol{\mu}_n) \\
\boldsymbol{\beta}|\phi,\boldsymbol{\theta} &\sim \mathcal{N}\left(\boldsymbol{\mu}_n, \boldsymbol{\Lambda}_n^{-1}/\phi\right) \\
\phi|\boldsymbol{\theta} &\sim \mathbf{G}(a_n, b_n)
\end{aligned}$$

During sampling, we can use sequential inverse-CDF method rather than rejection sampling to boost up sampling efficiency. By multivariate normal theory:

$$\begin{aligned}
\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \end{bmatrix} &\sim N\left(\begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix}\right) \\
\mathbf{Y}_1|\mathbf{Y}_2 = \mathbf{y}_2 &\sim N\left(\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{y}_2 - \boldsymbol{\mu}_2), \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}\right)
\end{aligned}$$

All conditional distribution will be computable. We may use take the advantage and to sample $\mathbb{P}(\omega, \alpha, \beta|\boldsymbol{\theta}) = \mathbb{P}(\beta|\boldsymbol{\theta})\mathbb{P}(\alpha|\beta, \boldsymbol{\theta})\mathbb{P}(\omega|\alpha, \beta, \boldsymbol{\theta})$ sequentially. At each conditional normal, we can apply inverse-CDF methods for sampling. As ω has the least boundary condition (only requires positive value), it will be sampled lastly.

6.2 Sampling $\upsilon = \tau^{-1}$

This is a standard inverse gamma update

$$y_{ti} \sim \mathcal{N}(f_{ti}(\theta_t, p_t, K_{ti}, T_{ti}), \tau^{-1})$$

$$\mathcal{L}(\tau; \{y\}_{1:T}^{1:n_t}) \propto \tau^{\frac{\sum_{t=1}^{T} n_t}{2}} \exp\left\{-\frac{1}{2}\tau \sum_{t=1}^{T}\sum_{i=1}^{n_t}(y_{ti} - f_{ti})^2\right\}$$

$$\tau \sim \mathbf{G}(\delta, \epsilon) = \mathbf{G}(1, 1)$$

$$\tau|\{y\}_{1:T}^{1:n_t} \sim \mathbf{G}(\delta + \frac{\sum_{t=1}^{T} n_t}{2}, \epsilon + \frac{1}{2}\sum_{t=1}^{T}\sum_{i=1}^{n_t}(y_{ti} - f_{ti})^2)$$

7 Experiment

7.1 Data Sources

The data I used are the S&P 500 index option from year 2017.4.1 to 2020.4.1. The dataset was previously purchased from cboe.com. (If the dataset is bigger, it could be more interesting as it incorporates data from recessions. **If you happens to have related dataset, please LET ME KNOW as my continued research still need these data**).

symbol	code	OptionType	expirydate	date	close	change	bid	ask	volume	openinterest	strike	spotclose	close2
SPX	SPX1721D500	call	20170421	20170403	1761.1	0	1855.9	1860.2	0	14	500	2358.84	1761.1
SPX	SPX1721D1000	call	20170421	20170403	1355	0	1356.2	1360.7	0	6857	1000	2358.84	1355
SPX	SPX1721D1100	call	20170421	20170403	1279.8	0	1256.5	1260.7	0	20	1100	2358.84	1279.8
SPX	SPX1721D1200	call	20170421	20170403	0	0	1156.3	1160.8	0	0	1200	2358.84	1158.55
SPX	SPX1721D1225	call	20170421	20170403	0	0	1131.3	1135.6	0	0	1225	2358.84	1133.45
SPX	SPX1721D1275	call	20170421	20170403	0	0	1081.6	1085.8	0	0	1275	2358.84	1083.7
SPX	SPX1721D1300	call	20170421	20170403	0	0	1056.4	1060.9	0	0	1300	2358.84	1058.65
SPX	SPX1721D1325	call	20170421	20170403	0	0	1031.4	1035.7	0	0	1325	2358.84	1033.55
SPX	SPX1721D1350	call	20170421	20170403	0	0	1006.4	1010.9	0	0	1350	2358.84	1008.65
SPX	SPX1721D1375	call	20170421	20170403	0	0	981.4	985.7	0	0	1375	2358.84	983.55
SPX	SPX1721D1400	call	20170421	20170403	938	0	956.5	960.7	0	1	1400	2358.84	938
SPX	SPX1721D1425	call	20170421	20170403	0	0	931.7	935.9	0	0	1425	2358.84	933.8
SPX	SPX1721D1450	call	20170421	20170403	0	0	906.5	911	0	0	1450	2358.84	908.75
SPX	SPX1721D1475	call	20170421	20170403	0	0	881.6	885.8	0	0	1475	2358.84	883.7
SPX	SPX1721D1480	call	20170421	20170403	0	0	876.5	881	0	0	1480	2358.84	878.75
SPX	SPX1721D1490	call	20170421	20170403	0	0	866.5	871	0	0	1490	2358.84	868.75

Figure 1: SPX500 data, a portion of one day

The database is huge. For each day, it has option quotes for both call and put options. Then for each of them, there exist a number of different expiration dates. And for each given expiration date, there are a few possible strike prices. Then for the given expiration date and strike price, it includes the price quotes for ”ask”, ”bid”,”high”,”low”, ”closing”, and ”close2”. For each combination, a field called ”volume”, represents the amount of such configured option been traded on that date. The ”close2” records the mean of the traded value on each option type. In this experiment, I'm picking out the top 20 traded options on each day for the analysis (y_{ti} for each t).

Besides the option data, we also need the risk free interest rate data, another key component that is involved inside our Black-Scholes formula. Here, we used London InterBank Offer Rate (LIBOR) 3-month rate for U.S. dollar. The original link for the data is in LIBOR. However, the site requires subscription. As the data is very widely used, the same dataset is also attached on FRED Economic Data website. The dataset recorded a serie of risk-free interest rate for everyday, while few days have missed value. As U.S. dollar risk is reasonably stable, I interpolated the missed value by taking the mean of risks on the day before and the day after.

7.2 Results

Prior values have been shown in above in each section. We've run MCMC on a single chain for 10000 iterations. Figure 2 shows shows the sampled distribution of our MCMC samples for each parameter. And figure 3 the trajectory for MCMC has converged.

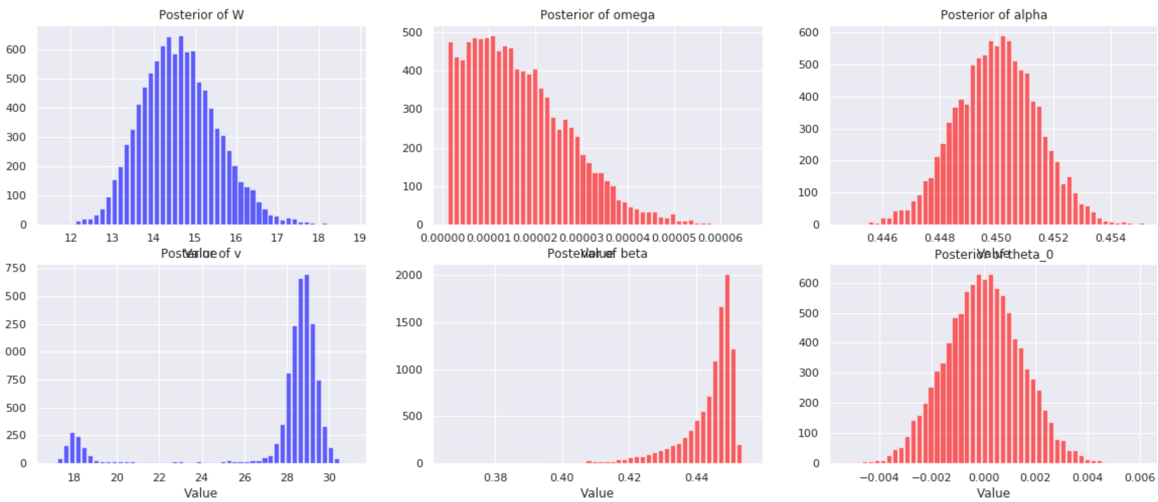


Figure 2: MCMC posterior empirical histogram for all parameters

We notice that $\mathbb{E}[\omega, \alpha, \beta] = [1.36 \times 10^{-5}, 4.51 \times 10^{-1}, 4.32 \times 10^{-1}]$. This shows that the contribution of volatility from observed variance at current time step and past volatility are nearly the same, current observation taking slightly higher weight. The ω, however, is very small, due to the fact that these volatility are all **day volatility on returns** which usually within ± 0.003.

Second, we notice that v has bi-model distribution. I guess this is because that our model takes into both longer-term options and shorter term options. However, the model assume same level of uncertainty of prices for these options. Intuitively, this should be false, because longer term options may have bigger risks. Therefore, the price should have larger variance. Therefore, a polarization of variance on predicted option prices occurs.

Lastly, we find that the transition variance W is 14.3. This is unacceptable for me as comparing to the daily volatility, which is at level 10^{-5} level, the uncertainty is too big to be acceptable.

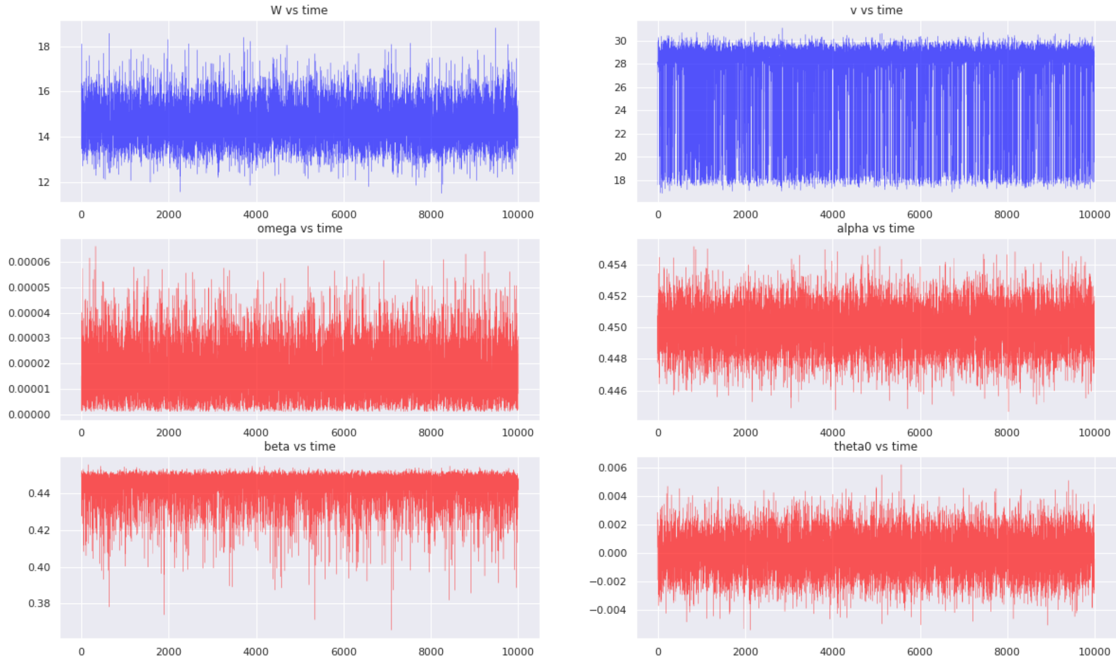


Figure 3: MCMC trajectories for all parameters

Finally, we append the backward sampled trajectory for volatility process throughout the time line in figure 4. Here, **volatility is represented as standard deviation $\sqrt{\theta_t}$ rather than variance,** as we aim to compare the trend and result with the dotted delta return: difference of consecutive 2 days' returns for each day.

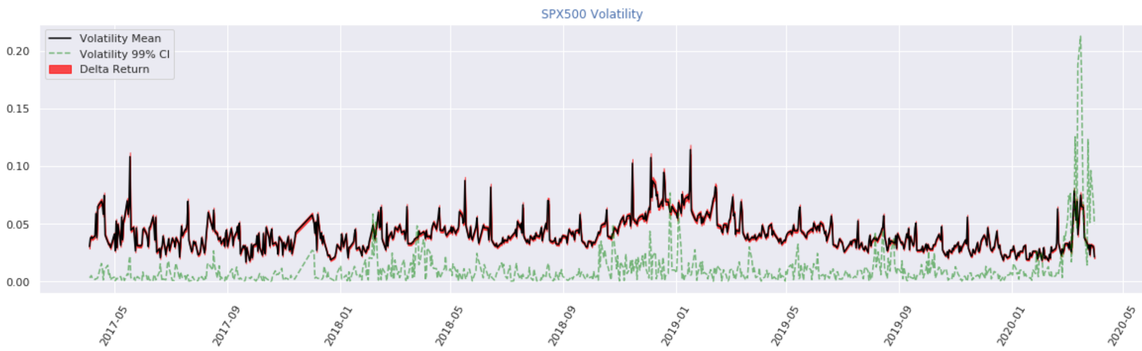


Figure 4: Implied volatility process

From the graph, we understand that the volatility process is capturing a reasonable trend: the volatility goes up in both early 2019, and also early 2020. However, it is obvious that the up-potential for 2020 is under-predicted. This may result from 2 reasons:

1. The backward smoothing smoothed the volatility out, and recent taking-off volatility hasn't

been fully captured by enough data points yet.

2. Our model takes in both stock price and return information and also option price information. We take in 20 different options at each day, weighing a lot in determining our model parameters. Therefore, if current market option prices have been systematically under-predicted, the modeled volatility process may also likely to be under-predicted.

7.3 Conclusions:

The following conclusions can be drawn from the modeling process:

1. It is possible to model latent volatility by combining modeling returns and also option data together.

2. Both in beginning of 2019 and 2020, a signal of increment in volatility occurs.

3. Recent volatility has been under-measured by market. A actual volatility is higher than that used for option pricing, leaving the SPX500 ETF European call options relatively under-priced .

8 Model Critics and Potential Further-works:

1. The model assume the volatility variance follows a Gaussian distribution. This is **false**, as the support doesn't allow negative value. Therefore, prior initialization needs to set negatively values 3 standard deviations from the mean.

2. Backward sampling requires sampling a series of $theta_t$. During which, negative value may also occur. Thus, computation of the model becomes highly unstable, and model needs to be re-run until a successful Backward smoothing. After that, the model becomes more stable.

3. Prior choosing takes much efforts. To better accommodate the issue and enable online learning, EM algorithm with boundary constrains can be proposed to better and faster deal with this task, And it doesn't require backward sampling a series of θ_t, as only $\mathbb{E}[\theta_{1:T}|\alpha, \beta, \omega]$, $\mathbb{E}[\theta_{1:T}^2|\alpha, \beta, \omega]$, $\mathbb{E}[\theta_t\theta_{t-1}|\alpha, \beta, \omega]$ are required.

4. Extend Kalman Filter uses 1st order Taylor approximate. However, the Black-Scholes formula is highly non-linear under some circumstances. In this case, error between approximated and correct output h_{ti} can be big, leading potential ξ_{ti} in forward filtering step, thus leads again to negative volatility.

5. The output variance v should not be constant. It's magnitude should be quadratic proportional to maturity time of the underlying option.

For further works, I will first implement EM algorithm instead of MCMC for parameter inference. To me, this sounds like the most promising task that will boost up inference efficiency and stability. Second, I'm seeking for potentially pushing EKF to second order Taylor approximation. Though the Kalman update won't be in close form, but at least numerical methods would be available, an approximation error can be highly reduced, furthermore boost up accuracy.

References

[1] G. Rigatos *A Kalman filtering approach for detection of option mispricing in the Black-Scholes PDE model.* 2014 IEEE Conference on Computational Intelligence for Financial Engineering Economics (CIFEr), London, 2014, pp. 378-383.

[2] K. Liu and X. Wang *A Pragmatical Option Pricing Method Combining Black-Scholes Formula, Time Series Analysis and Artificial Neural Network.* 2013 Ninth International Conference on Computational Intelligence and Security, Leshan, 2013, pp. 149-153.

[3] Introduction to S&P 500, http://www.cboe.com/micro/spx/

[4] London Interbank Offer Rate 3-month rate for U.S. dollars, http://www.economagic.com/em-cgi/data.exe/libor/day-us3m

[5] LIBOR 3 month U.S. data, *FRED Economic Data*, https://fred.stlouisfed.org/series/USD3MTD156N

Appendix

EKF Forward Filtering with Multiple Output Functions

$$
\begin{aligned}
h_{ti}(a_i) &= f_{ti}(a_t) - F_{ti}^T(a_t) \\
F_{ti}^T &= \frac{\partial f_{ti}}{\partial \theta_t}\Big|_{\theta_t = a_t} \\
d_t(m_{t-1}) &= g_t(m_{t-1}) - G_t m_{t-1} \\
G_t &= \frac{\partial g_t}{\partial \theta_{t-1}}\Big|_{\theta_{t-1} = m_{t-1}} \\
\xi_{ti} &= y_{ti} - h_{ti}(a_t) \\
a_t &= g_t(m_{t-1}) \\
R_t &= G_t C_{t-1} G_t^T + W_t \\
&\text{(Scalar Case)} \\
m_t &= \frac{R \sum_{i=1}^{n_t} F_{ti}\xi_{ti} + v a_t}{R \sum_{i=1}^{n_t} F_{ti}^2 + v} \\
C_t &= \frac{vR}{R \sum_{i=1}^{n_t} F_{ti}^2 + v}
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{P}(\theta_t|\mathcal{D}_t) &= \frac{\prod_i^n \mathbb{P}(y_{ti}|\theta_t, \mathcal{D}_{t-1})\mathbb{P}(\theta_t|\mathcal{D}_{t-1})}{\prod_i^n \mathbb{P}(y_{ti}|\mathcal{D}_{t-1})} \\
&\propto \prod_i^n \mathbb{P}(y_{ti}|\theta_t, \mathcal{D}_{t-1})\mathbb{P}(\theta_t|\mathcal{D}_{t-1}) \\
&= \prod_i^n N(f_{ti}(\theta_t), v) N(a_t, R_t) \approx \prod_i^n N(h_i(a_t) + F_{ti}^T\theta_t, v) N(a_t, R_t) \\
&\propto \exp\left\{-\frac{1}{2v}\sum_i^n ((y_{ti} - h_i(a_t)) - F_{ti}^T\theta_t))^2\right\} \exp\left\{-\frac{1}{2}(\theta_t - a_t)^T R_t^{-1}(\theta_t - a_t)\right\} \\
&\propto \exp\left\{-\frac{1}{2v}\sum_i^n (\xi_{ti} - F_{ti}^T\theta_t)^2\right\} \exp\left\{-\frac{1}{2}(\theta_t - a_t)^T R_t^{-1}(\theta_t - a_t)\right\} \\
&\propto \exp\left\{-\frac{1}{2}\left[\frac{1}{v}\sum_i^n (\xi_{ti} - F_{ti}^T\theta_t)^2 + (\theta_t - a_t)^T R_t^{-1}(\theta_t - a_t)\right]\right\} \\
&\propto \exp\left\{-\frac{1}{2}\left[\frac{1}{v}\theta_t^T(\sum_{i=1}^n F_{ti}F_{ti}^T)\theta_t - 2\theta_t^T(\sum_i^n F_{ti}\xi_{ti})(v)^{-1} + \theta_t^T R_t^{-1}\theta_t - 2\theta_t^T R_t^{-1} a_t\right]\right\} \\
&\propto \exp\left\{-\frac{1}{2}\left[\theta_t^T\left(\frac{1}{v}(\sum_{i=1}^n F_{ti}F_{ti}^T) + R_t^{-1}\right)\theta_t - 2\theta_t^T\left(\frac{1}{v}(\sum_i^n F_{ti}\xi_{ti}) + R_t^{-1}a_t\right)\right]\right\} \\
&\sim N\left(\left(\frac{1}{v}(\sum_{i=1}^n F_{ti}F_{ti}^T) + R_t^{-1}\right)^{-1}\left(\frac{1}{v}(\sum_i^n F_{ti}\xi_{ti}) + R_t^{-1}a_t\right), \left(\frac{1}{v}(\sum_{i=1}^n F_{ti}F_{ti}^T) + R_t^{-1}\right)^{-1}\right) \\
&= N(m_t, C_t)
\end{aligned}
$$

EKF Backward Smoothing

$$\begin{aligned}
\log P\left(\theta_{t+1}, \theta_t | \{y\}_1^T\right) &= \log P\left(\theta_{t+1}|\theta_t\right) + \log P\left(\theta_t|\{y\}_1^t\right) - \log P\left(\theta_{t+1}|\{y\}_1^t\right) + \log P\left(\theta_{t+1}|\{y\}_1^T\right) \\
&= -\frac{1}{2}\left(\theta_{t+1} - G_{t+1}\theta_t - d_{t+1}(m_t)\right)^\intercal W_{t+1}^{-1}\left(\theta_{t+1} - G_{t+1}\theta_t - d_{t+1}(m_t)\right) \\
&- \frac{1}{2}\left(\theta_t - m_t\right)^\intercal \left(C_t\right)^{-1}\left(\theta_t - m_t\right) + \frac{1}{2}\left(\theta_{t+1} - a_{t+1}\right)^\intercal \left(R_{t+1}\right)^{-1}\left(\theta_{t+1} - a_{t+1}\right) \\
&- \frac{1}{2}\left(\theta_{t+1} - m_{t+1}^*\right)^\intercal \left(C_{t+1}^*\right)^{-1}\left(\theta_{t+1} - m_{t+1}^*\right) + \dots \\
&= -\frac{1}{2}\theta_{t+1}^\intercal\left(W_{t+1}^{-1} - \left(R_{t+1}\right)^{-1} + \left(C_{t+1}^*\right)^{-1}\right)\theta_{t+1} \\
&- \frac{1}{2}\theta_{t+1}^\intercal\left(-W_{t+1}^{-1}G_{t+1}\right)\theta_t - \frac{1}{2}\theta_t^\intercal\left(-G_{t+1}^\intercal W_{t+1}^{-1}\right)\theta_{t+1} \\
&- \frac{1}{2}\theta_t^\intercal\left(G_{t+1}^\intercal W_{t+1}^{-1}G_{t+1} + \left(C_t\right)^{-1}\right)\theta_t + \theta_t^\intercal\left(-G_{t+1}^\intercal W_{t+1}^{-1}d_{t+1}(m_t) + \left(C_t\right)^{-1}m_t\right) + \dots
\end{aligned}$$

Therefore, we have

$$\begin{bmatrix}\theta_t \\ \theta_{t-1}\end{bmatrix}\Big|\mathcal{D}_T = \mathcal{N}\left(\begin{bmatrix}\mu_t \\ \mu_{t-1}\end{bmatrix}, \begin{bmatrix}\Phi_{t,t} & \Phi_{t,t-1} \\ \Phi_{t-1,t} & \Phi_{t-1,t-1}\end{bmatrix}^{-1}\right) = \mathcal{N}\left(\begin{bmatrix}\mu_t \\ \mu_{t-1}\end{bmatrix}, \begin{bmatrix}\Sigma_{t,t} & \Sigma_{t,t-1} \\ \Sigma_{t-1,t} & \Sigma_{t-1,t-1}\end{bmatrix}\right)$$

$$\begin{aligned}
\Phi_{t,t} &= W_t^{-1} - \left(R_t\right)^{-1} + \left(C_t^*\right)^{-1} \\
\Phi_{t,t-1} &= -W_t^{-1}G_t \\
\Phi_{t-1,t} &= -G_t^\intercal W_t^{-1} \\
\Phi_{t-1,t-1} &= G_t^\intercal W_t^{-1}G_t + \left(C_{t-1}\right)^{-1} \\
\Phi_{t-1,t-1}^{-1} &= C_{t-1} - J_{t-1}R_tJ_{t-1}^\intercal \\
J_{t-1} &= C_{t-1}G_t^\intercal R_t^{-1} \\
\Phi_{t-1,t-1}^{-1}\Phi_{t,t-1} &= -J_{t-1} \\
C_{t-1}^* &= C_{t-1} + J_{t-1}(C_t^* - R_t)J_{t-1}^\intercal \\
\Sigma_{t,t} &= C_t^* \\
\Sigma_{t,t-1} &= C_t^*J_{t-1}^\intercal \\
\Sigma_{t-1,t} &= J_{t-1}C_t^{*\intercal} \\
\Sigma_{t-1,t-1} &= C_{t-1} + J_{t-1}(C_t^* - R_t)J_{t-1}^\intercal \\
\mu_t &= m_t^* \\
\mu_{t-1} &= m_{t-1} + J_{t-1}(m_t^* - g_t(m_{t-1}))
\end{aligned}$$

So the conditional sampling distribution should be:

$$\begin{aligned}
\theta_{t-1}|\theta_t, \mathcal{D}_t &\sim \mathcal{N}(\mu_{t-1} + \Sigma_{t,t-1}^T\Sigma_{t,t}^{-1}(\theta_{t-1} - \mu_t), \Sigma_{t-1,t-1} - \Sigma_{t,t-1}^T\Sigma_{t,t}^{-1}\Sigma_{t-1,t}^T) \\
&= \mathcal{N}(m_{t-1} + J_{t-1}(m_t^* - g_t(m_{t-1})) + (C_t^*J_{t-1}^\intercal)^\intercal(C_t^*)^{-1}(\theta_{t-1} - m_t^*), \\
&\qquad C_{t-1} + J_{t-1}(C_t^* - R_t)J_{t-1}^\intercal - (C_t^*J_{t-1}^\intercal)^\intercal(C_t^*)^{-1}(J_{t-1}C_t^{*\intercal})^\intercal) \\
&= \mathcal{N}\Big(m_{t-1} + J_{t-1}(\theta_t - g_t(m_{t-1})), \quad C_{t-1} - J_{t-1}R_tJ_{t-1}^\intercal\Big)
\end{aligned}$$

Last Revised: April 20, 2020