UPPSALA
UNIVERSITET

Computational Finance

Filtering in Finance

Peili Guo
Peili.Guo.7645@student.uu.se

May 24, 2018

1 Introduction to filtering

Many financial instruments on the markets are modelled with Stochastic differential equations. When callabrating parameters, a long history data and a model was given, with the help of different computing methods, an optimisation problem was formed in order to find the best set of parameters. However, the above models does not include time series data for adaptive models that aims for a higher accuracy. Apart from the price of financial instruments, volatility is of great interest. For example volatilities across instruments need to be consistent to avoid arbitrage from the pricing perspective and use volatility to balance risk across assets and across time for portfolio management. The price of financial instruments can be observed directly on the market when trading. But volatility cannot be directly observed from trading. Therefore, filtering is introduced to handle time series data with noisy estimates of key quantities and trading indicators.

The famous Kalman filter comes from signal processing and control theory. Some of its application are in navigation and vehicle control, economics and finance, processing data from sensors with noises (rador, laser, camera, etc...), feature and cluster tracking in different computer applications. One of the advantage of Kalman filter is that it can handle both observable and hidden data using the state space model. In this report, different filters will be introduced and applied to the S&P 500 Index to model the volatility.

2 Filtering Methods

Kalman filters are based on state space model in discrete time domain. The general idea of filtering can be summaries in two steps:

1. Predict state i+1 using the updated data from state i with the state model.
   $x_{i+1} = f(x_i, w_{i+1})$

2. At state i+1, update the state i+1 with the observation at i+1 with the measuring model.
   $z_{i+1} = h(x_{i+1}, u_{i+1})$

w_{i+1} and u_{i+1} here are mutually uncorrelated normally distributed random variables. Below in figure 1 the filtering process is shown.

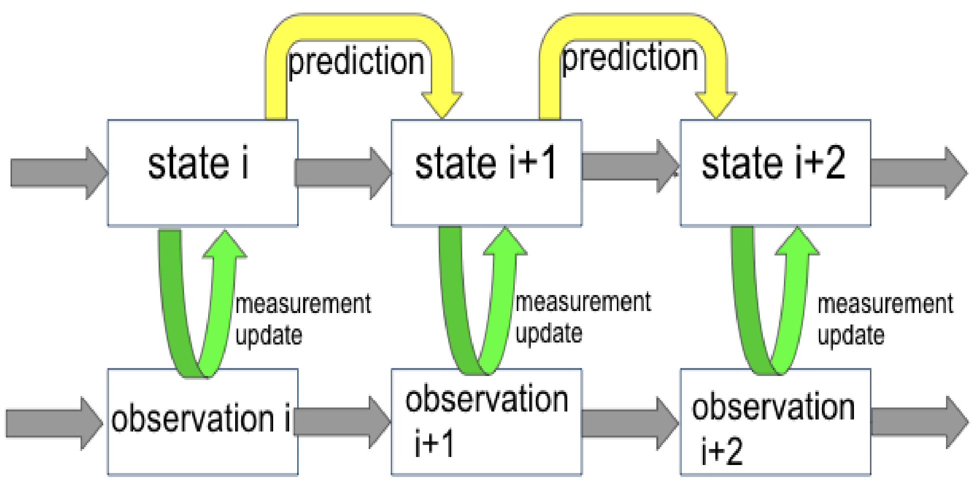

Figure 1: filtering process

2.1 Kalman Filter

The basic Kalman Filter has two assumptions: both the state transition and the measurement update model are linear models and the noise is Gaussian (normally distributed with mean 0). The algorithms is simple with easy matrix operations at each time step. The linear Kalman filter will work well if the data shows strong linear relation. The methods can be summarised in the following steps:

1. Predict state i+1 using the updated data from state i with the state model.
   $x_{i+1} = F * x_i$

2. At state i+1, update the state i+1 with the observation at i+1 with the measuring model.
   $\hat{x}_{i+1} = x_{i+1} + K * (z_{i+1} - H * x_i)$
   here the z_{i+1} is the observation at time i+1 and K is the Kalman gain that minimise the mean-square error estimate.

2.2 Extended Kalman Filter

The Kalman filter work well if the underlying model is linear. But in reality, many models are non linear. The assumption of linearity in Kalman filter might not hold when modelling financial derivatives especially the process is only partially observable. For example, volatility, which is the degree of variation of price over time measured by the standard deviation of logarithmic returns, is a hidden state that cannot be observed directly. A modified model that include non-linear transformation for the Kalman Filter is Extended Kalman Filter (EKF). In EKF, the state transition model and measurement update model can be non-linear but they should be differentiable. The covariance are computed using the Jacobian at each time step. In short, the process of EKF uses first-order linearisation technique for non-linear functions and the stochastic term in the state transition model and measurement update model are Gaussian. The methods can be summarised in the following steps, which is almost the same as the Kalman Filter:

1. Predict state i+1 using the updated data from state i with the state model.

   $x_{i+1} = F(x_i)$, F(x) is a non-linear function.

2. At state i+1, update the state i+1 with the observation at i+1 with the measuring model.

   $\hat{x}_{i+1} = x_{i+1} + K * (z_{i+1} - H(x_i))$

   here the z_{i+1} is the observation at time i+1 and K is the Kalman gain that minimise the mean-square error over all linear estimators.

2.3 Unscented Kalman Filter

Unscented Kalman Filter (UKF) was proposed by Julier and Uhlmann in 1997 that uses the true nonlinear model. A set of sample points called sigma points were generated using unscented transformation. These sigma points are propagated thought the non-linear state transition and measurement update model, and the covariance of both estimations is recovered. UKF has a higher order of accuracy than EKF with the same order of accuracy of covariance. UKF does not require to compute Jacobian, which can be both computational expensive and error prone for complex non-linear functions. The methods can be summarised in the following steps:

1. Start with an approximation and initialise state space augmentation.

$$x_i^a = \begin{bmatrix} x_i \\ w_i \\ u_i \end{bmatrix}$$

2. unscented transform of sigma points.

   For $j = 1...n_a$

   $\chi_i^a(j) = \hat{x}_i^a + (\sqrt{(n_a + \lambda) * P_i^a})_j$

For $j = n_a + 1 ... 2n_a$

$\chi_i^a(j) = \hat{x}_i^a - (\sqrt{(n_a + \lambda) * P_i^a})_{j-n_a}$

3. Time Update, use the transition function from state i transit to state i+1.

4. Measurement update, use observation at i+1 to update the predicted state i+1.

2.4 Particle Filter

Particle Filter (PF), also known as Sequential Monte Carlo, is a more recent filtering alternative that uses Markov-Chain Monte Carlo for simulations. PF uses discrete density function for $p(x_i|z_i)$ obtained from histogram via Monte Carlo simulation. This distinguish PF to KF and EKF as the later use Gaussian approximation for $p(x_i|z_i)$. Theoretically, PF is the closest for true non-linear stochastic model as it can use non linear state transition function and the noise are not limited to be assumed Gaussian. However, the PF is very sensitive to the initial state. If the initial state is very far away from the true state, it can take a long time for the filter to converge. And the computing of PF can be time consuming because of Monte Carlo simulation. The methods can be summarised in the following steps:

1. Start with an initial state x_0 and generate N numbers of particles around it using Monte Carlo.

   Each particle is computed with $x_0^j = x_0 + \sqrt{P_0} Z^j$, Z^j is a random number from standard normal distribution.

2. use state transition function from state i transit to state i+1 for each particle.

   $x_{i+1} = f(x_i)$

3. Resample the particles according to their importance at i+1.

   Once the observation at i+1 is arrived, the weight of each particle j can be computed using

$w_t^j = w_{t-1}^j \frac{p(y_t|v_t,v_{t-1},y_{t-1}p(v_t|v_{t-1})}{\pi(v_t|v_{t-1},y_t)}$

Each particles are then resampled with their weights that the particles with lower weight will be eliminated and regenerate with a particle of higher weight.

3 Results from different filters

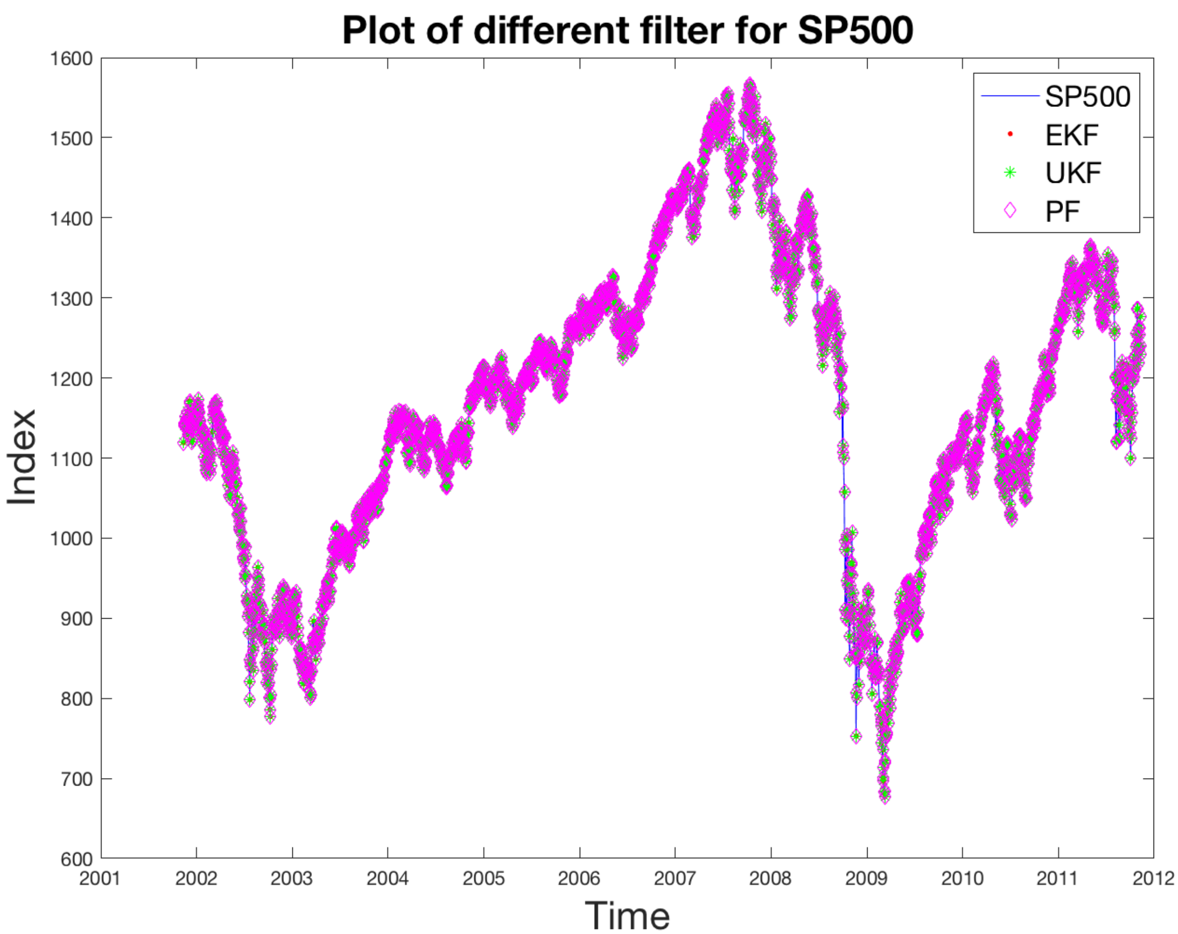

Figure 2: Results of EKF, UKF, PF for S&P 500 index

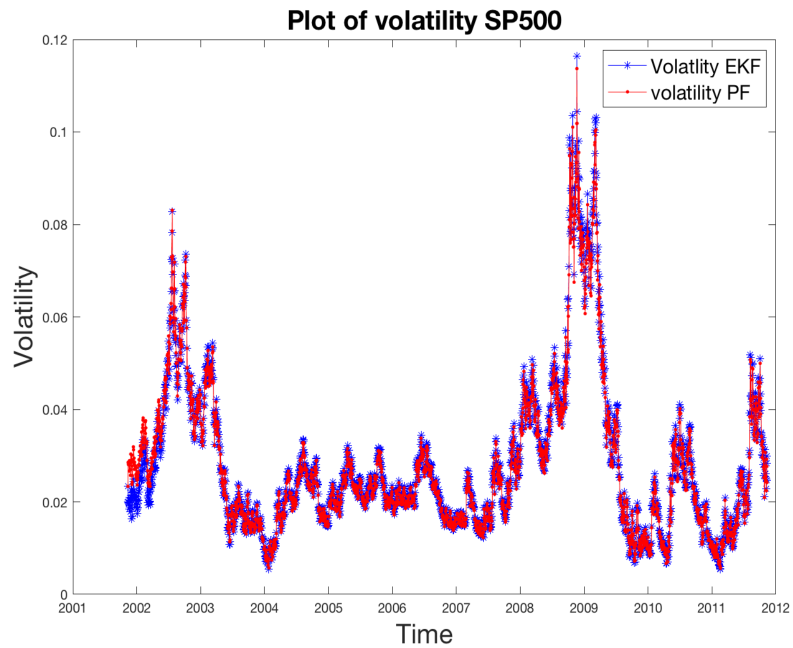

Figure 3: Volatility from EKF and PF for S&P 500 index

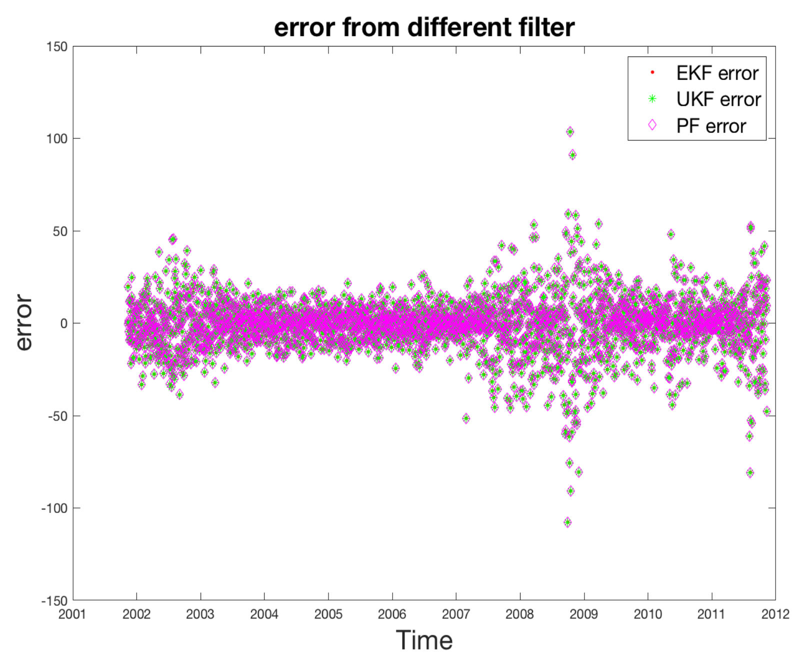

Figure 4: Residual from EKF, UKF and PF for S&P 500 index

Below is a short summary of EKF, UKF and PF. The residual is computed as the sum of absolute value of the difference between actual Index and the modelled Index from volatility:

Filter Type	Run Time (s)	Total Residual	RMSE
EKF	0.0103	25303.004	14.6389
UKF	0.2913	25308.064	14.6445
PF	10.6376	25302.779	14.6389

There is no significant difference in accuracy among the three filters. But the model simulation time for PF is 1000 times of that for EKF, which could potentially be a problem for real time data analysis for average desktops and laptops. The positive about PF is that overall, the residual is the smallest, which means that the results are more accurate than EKF and UKF. It is worth pointing out that for the PF applied on S&P data is using the same state transition function as EKF and we assumed the random number generator is good enough to represent the stochastic elements in the model. But the question arise from the S&P 500 data is that: is it worth to spend 1000 times time for a less than 0.001% improvement of accuracy?

The modelling with S&P 500 data has a unit in days, but to capture the true dynamic of the market, real time analysis shall be in favour. In this case, PF could potentially be problematic, as the heavy computing from simulating N particles at each time steps. In the simulation for S&P 500, the model parameters were given, and the time series data can be captures well with EKF. If the parameters are not given, an optimisation problem shall be formed searching for the set of parameters that will maximise the likelihood of the model. In my opinion, for the S&P 500 data, using EKF is sufficient. But, for some case, when the state transition model is highly non-linear or indifferentiable , computing the Jacobian can be either very complicated or computational expensive using numerical methods with risk of accumulated error. Then UKF or PF could be applied.

The volatility showed in Figure 3 showed the same trend as VIX. It confirmed that the filtering methods is applicable to financial data. But different methods has its own pros and cons and there is always the possibility to improve the state transition function. In the end, the choice of methods will need to be at a balanced point between accuracy and computing cost. To be able to perform real time analysis, there are much more to consider. For example, the observation data of Index/Price might come in with a big package with noises as actual transactions are executed, the parameters of the model can also be time depended. There are many uncertainties on the market, to get a good volatility is not an easy task. Volatility is important for a balanced portfolio in risk management across time for the buy side, and it is more important from the sell side on the pricing perspective. There are many financial derivatives (different options) available on the market that always desire for a more accurate volatility, because the volatility across instruments need to be consistent, to avoid arbitrage opportunities.

References

[1] A. Hirsa, *Computational Methods in Finance*, Chapman & Hall/CRC Financial mathematics series, 2013.

[2] A. Javaheri, D. Lautier and A. Galli, *Filtering in Finance*, Wilmott magazine, 2005.

[3] D. Creal, *A survey of sequential Monte Carlo Methods for economics and Finance*, Econometric Review, 31:3, 245-296, 2012.

[4] S.I. Aihara, A. Bagchi, S. Saha *On parameter estimation of stochastic volatility models from stock data using particle filter - application to AEX index*, ICIC, Vol 5, No. 1, 2009.