Contents

Proposal 1

Parameter Learning in a Local Volatility Model Using a Particle Filter

In this project we use underlying prices and observed option data to make *online* inference about a local volatility model. We use a parametric model for the local volatility function, and apply a particle filter to construct iterative approximations of the posterior distribution of the parameters on arrival of new underlying and option data. This work is perhaps closest in spirit to Hamida & Cont (2005), but it differs in the sense that they solve a pure calibration problem rather than approximate the unknown parameter's posterior.

Problem Outline

Part 1: Background

Not Included.

Part 2: Definitions and General Implementation Ideas

To make inference about the parameters θ in $\sigma_{\mathrm{LV}}(S,t;\theta)$, we apply a particle filter to iteratively construct an approximation for the posterior distribution of θ on arrival of new underlying and options data.

Define $c_t^M(K,T;\theta)$ to be a model generated call price at time t with strike K, expiry T, and a particular set of local volatility parameters θ. These model prices can be computed by solving either the Dupire Equation, or the Black-Scholes Equation, and are comparable to market observed prices $c_t(K,T)$ also at time t. To construct a dynamic system define $s_{1:t}=\{s_1,s_2,\ldots,s_t\}$ to be a set of observed underlying prices which are assumed to have been generated from

$$dS/S = rSdt + \sigma_{\mathrm{LV}}(S,t;\theta)\,dB. \tag{1}$$

At each t we also have a set of N observed call prices $\{c_t(K_i,T_i)\}_{i=1}^N$, where K_i and T_i are the strike and expiry of each call in the market (ie $N=K\times T$ where we have K strikes and T expiries).

It is conveniant to package the observed quantities into the set $x_{1:t}$, where at each time t

$$x_t = [s_t;\theta, c_t(K_1,T_1;\theta),\ldots,c_t(K_N,T_N;\theta)] \tag{2}$$

The objective of the project is to compute the posterior $p(\theta|x_{1:t})$. To do so we construct a particle approximation for $p(\theta|x_{1:t+1})$ where the weight $\pi_{t+1}^j = p(\theta^j|x_{1:t+1})$ is updated according to the standard rule

$$\pi_{t+1}^j = \frac{p(x_{t+1}|x_t,\theta^j)\pi_t^j}{\sum_j p(x_{t+1}|x_t,\theta^j)\pi_t^j}. \tag{3}$$

and $j=1,2,\ldots,J$ denotes a suitably large number of particles in the approximation. The design of the filter can follow Gellert & Schlogl 2018.

To compute $p(x_{t+1}|x_t,\theta^j)$ for each particle we follow Hurn Et Al 2012 and note that the likelihood $p(x_{t+1}|x_t,\theta)$ can be written

$$p(x_{t+1}|x_t,\theta^j) = f(s_{t+1}|s_t,\theta^j)\prod_{i=1}^N g\left(c_{t+1}(K_i,T_i)|c_{t+1}^M(K_i,T_i),\theta^j\right) \tag{4}$$

as x_t is fully observed and each option privides independant piece of information. The distribution $g(.)$ is discussed below.

Note that $s_{t+1}|s_t$ is also conditional on θ via the local volatility function $\sigma_{\text{LV}}(S,t;\theta)$ and is distributed according to Equation (1). The transition pdf required to compute the likelihood $f(s_{t+1}|s_t;\theta)$ satisfies a Fokker-Planck Equation

$$\frac{\partial f}{\partial t} = \frac{\partial}{\partial s}\left[\frac{1}{2}\frac{\partial}{\partial s}\left(\sigma_{\text{LV}}^2(s,t;\theta)f\right) - rsf\right] \tag{5}$$

with initial condition $f(s,t_0;\theta) = \delta(s - s_0)$ and relevant boundary conditions. In general this PDE will not have an analytic solution but can be approximated to high accuracy (insert econometric references here).

The distribution $g\left(c_{t+1}(K,T)|c_{t+1}^M(K,T),\theta\right)$ is problematic. Since all option prices are known when conditioned on the underlying and local volatility parameters θ, the problem is overspecified. This motivates the use of a *pricing error* model, which might be either multiplicative or additive (insert references here). For example we might take

$$g\left(c_{t+1}(K,T)|c_{t+1}^M(K,T),\theta\right) \sim \mathbf{N}(c_{t+1}^M(K,T),\sigma_M^2) \tag{6}$$

where σ_M is the spread for each option. Alternatively, one could include σ_M in the paramter set θ and inferred from the data.

This fully specifies a procedure for computing $p(\theta|x_{1:t})$ via a particle filter, where $x_{1:t}$ includes both underlying and option prices.

Part 3: Reformulation Using Arbitrage Free Implied Volatility Surfaces

A slight reformulation of the problem allows us to use volatility surfaces that are guaranteed to be free from static arbitrage by construction. Previously, we have discussed making inference about the parameters θ in a local volatility function $\sigma_{\text{LV}}(S,t;\theta)$. Instead we can make inference about the implied volatility surface while restricting attention to surfaces free from static arbitrage. From such a surface, we can compute $\sigma_{\text{LV}}(K,T)$ via Dupire's Formula (in the original form using market call prices, or the form using implied volatility).

Define $\Sigma_t(K_i,T_i)$ to be the Black-Scholes implied volatility of a European call option with price $c_t(K_i,T_i)$ at time t. The collection of $\Sigma_t(K,T)$ for all options in the market defines the implied volatility surface. An equivalent formula to Dupire's Equation is available so that local volatilities $\sigma_{\text{LV}}(K,T;\theta)$ can be computed from implied volatilities $\Sigma(K,T)$. Previously we assumed that $\sigma_{\text{LV}}(S,t;\theta)$ was parameterised by a set θ, which we made inference about using a particle filter. Here, instead we parameterise the implied volatility surface with a set α in $\Sigma(K,T;\alpha)$. We will restrict attention to particular surfaces $\Sigma(K,T;\alpha)$ guaranteed to be free from static arbitrage by construction, such as those introduced in (insert Gatheral references). This is an easier way to enforce no-arbitrage requirements than more procedural methods (eg Fengler 2009). This formulation has a secondary benefit of exploiting rapid closed form solutions for model prices using the standard Black-Scholes formula.

Part 4: Implied Volatility Function

This project relies heavily on the Gatheral SVI and SSVI models. Here I include a couple of practical details. Given the importance of this function I should learn the complete model. The parameterisation of Σ from Gatheral and Jacquier (2014) and Gatheral (2013) used in this project is:

$$w(k,T) = \frac{1}{2}\theta_T \left(1 + \rho\varphi(\theta_T)k + \sqrt{(\varphi(\theta_T)k+\rho)^2 + (1-\rho)^2}\right) \tag{7}$$

where

$$w(k,T) = \Sigma^2(k,T)T$$

denotes the option total implied variance to expiry T, k is the log forward moneyness

$$\begin{aligned} k &= log(K/F_t) \\ &= log(K/S_t) \end{aligned}$$

when $r = q = 0$, the ATM variance in log moneyness (ie $k = 0$) is

$$\theta_T = \Sigma^2(0,T)T$$

and $\varphi(\theta)$ is

$$\varphi(\theta) = \frac{\eta}{\theta^{\gamma_1}(1+\beta_1\theta)^{\gamma_2}(1+\beta_2\theta)^{1-\gamma_1-\gamma_2}}$$

with

$$\begin{aligned} \gamma_1 &= 0.238 \\ \gamma_2 &= 0.253 \\ \beta_1 &= exp(5.18) \\ \beta_2 &= exp(-3.00) \\ \eta &= 2.016048\, exp(\varepsilon) \end{aligned}$$

and $\varepsilon \in (-1,1)$ and ρ are parameters.

I also note the Hendriks (2017) extension to this function, which retains the arbitrage free properties of the SSVI function whilst adding more degrees of freedom. This has been shown to be particularly useful for short expiries. We should keep this in mind for the future.

Simulation Experiment

The first requirement is a simulation experiment, with which we can:

1. Check for accuracy of the algorithm. That is, can we recover a known $\theta = \theta^*$ from option and underlying prices? What sample size is needed to recover θ^* with some certainty? Can we properly quantify this?
2. How does computation time scale with
   - Sample size
   - Particle number
3. How is accuracy and performance affected by various
   - error distribution for option prices
   - discretisation of underlying process
   - approximation method for Fokker-Planck Equation
   - implementation details in the filter
   - Implementation details for code, as it relates to performance

Pricing vs Historical Simulation

The proposal in this project is different to a pricing problem, which is the usual applications of local volatility models. In a Pricing Problem we have

$$\begin{aligned}
\text{initial conditions}: &\ s_0, t_0 \\
\text{expiries}: &\ \{T_0, T_1, \ldots\} \\
\text{strikes}: &\ \{K_0, K_1, \ldots\} \\
\text{local volatilities}: &\ \sigma_{\text{LV}}(K_i, T_j; \theta)
\end{aligned}$$

into the future $t < T$, and assume we are using a monte-carlo pricer to price a derivative expiring in the 'distant' future. The Monte-Carlo solver will simulate the risk-neutral dynamics of the underlying s_t into the distant future using the whole grid of $\sigma_{\text{LV}}(K_i, T_j; \theta)$. Here $\{s_0, t_0\}$ is static thoughout the whole process.

Consider now the 'Historical Simlulation' problem we are considering, which is subtely different. We are trying to recreate the experience of recieving a stream of underlying and option data over time. Here $\{s_0, t_0\}$ is no longer static. The *now* in the simulation is moving towards our first expiry.

This exposes the critical difference between historical simulations and pricing problems. In the pricing problem the underlying *experiences* the whole local volatility surface. In the historical simulation the underlying only experience parts of the local volatility close to the front month.

Complete Procedure Recipe

Below is the complete recipe. The objective is to create a simulaion as close to how we will experience live data as possible. Many of these settings have SPX in mind.

Fix Initial Conditions and Simulation Settings. Choose $s_0, t_0 = (4500, 20220103)$ and sample once daily at 3:00pm and skip weekends. Therefore both weekdaysand weekends will have $\Delta_t = 1$day though with actual data I will count weekends as eg $\Delta_t = 1/2$day.

Fix Implied Volatility Term Structure and Expiry Rules. In the Gatheral model the ATM volatility term structure is treated as an observed quantity determined exogenously to the model and as such we can choose it as we like. An initial proposal is a simple increasing function of time to expiry τ

$$\Sigma(\tau) = \Sigma_0 + \Sigma_1 \left(1 - exp(-\lambda_2 \tau)\right) \tag{8}$$

where eg we can λ_2 such that $\Sigma(\tau^*) = \Sigma^*$. With this function we build the future implied volatility surface. From the initial t_0 we use the above function to generate the ATM volatilities at each $T_i = t_0 + \tau_i$. In the Gatheral model, these become the

$$\varphi(\theta_{T_i}) = \Sigma^2(0, T_i) T_i \tag{9}$$

for our set of expiries $\{T_0, T_1, \ldots\}$. Definiting T_0 to be the current front month, we have $t < T_0$ at all times in the simulation. This means we will *always* be extrapolating implied volatilities *backwards* from T_0 to t. Note also that we need a roll rule - I will choose some rule such as $\tau > 5$ days to avoid being too close to expiry. The idea here is related to the previous discussion on **Historical** vs **Pricing** Simulations. Note that we require volatility to decay to zero at $t = T$, and Gatheral includes some discussion of this limit but we avoid it but rolling the front month.

Fix Strikes. I don't think this is particularly important, but we want to make sure we have enough strikes to sample the shape of each curve slice, we don't have too many strikes for efficiency reasons, and we dont want a set of strikes that become *all* in or out of the money. Given that we *do not* take any positions in any options I think we can determine which strikes we have available at any point in the simulation without affecting results. So we can choose strikes linear in k or K, and I choose the latter. An initial proposal is to take $K_{\min}, K_{\max}$ according to

$$K_{\min}, K_{\max} = F_t \cdot exp\left([-3, 3]\sqrt{\varphi_T}\right) \tag{10}$$

and linearly interpolate for the K_i, which in turns gives the k_i. Interestingly in a simulation where the volatility structure is definied in log moneyness, our underlying is always at the money (ie we will be using the front month ATM volatility for the whole simulation).

Fix Grid Size. Fix the number of expiries and strikes ie the size of the option grid. SPX monthlies list 12 expiries, and have very large numbers of strikes. Presumably the larger the grid the more volatility information we recieve. To begin I will take

$$\begin{aligned} \# T &= 12 \quad \text{ie monthly} \\ \# K &= 100 \end{aligned}$$

Fix Simulation Parameter Values Choose $\theta = \theta^*$ and generate the target simulation $s^*_{1:t}$. For diagnostic reasons we can generate the whole simulation at once but later generate it online. The rules are:

$$\begin{aligned}\text{Euler}: &\ s_{t+1}|s_t;\theta^* = s_t + s_t\,\sigma_{\mathrm{LV}}(s_t,t;\theta^*)\sqrt{\Delta_t}\varepsilon_t\\ \text{Euler}: &\ f(s_{t+1}|s_t,\theta^*) \sim \mathbf{N}(s_t,\sigma^2_{\mathrm{LV}}(s_t,t;\theta^*)\Delta_t)\end{aligned}$$

References

R Cont, S Ben Hamida: Recovering volatility from option prices by evolutionary optimization. Journal of Computational Finance, Vol. 8, No 3, 43-76.

Dupire, B.: Pricing With A Smile. Risk Magazine, January, pp 18-20, 1994.

Fengler, M.: Arbitrage-free smoothing of the implied volatility surface, Quantitative Finance, 2009

Gellert & Schlogl: Parameter Learning and Change Detection Using a Particle Filter With Accelerated Adaption, 2018.

Hurn Et Al.: Estimating the Parameters of Stochastic Volatility Models using Option Price Data, Working Paper #87 October 2012

MISC 1

Here I will outline the new procedure more clearly. Define $c_t^M(K,T;\alpha)$ to be the Black-Scholes price at time t of a European call with strike K and expiry T, whose Black-Scholes volatility σ is given by $\sigma = \Sigma(K,T;\alpha)$. These call prices are free of static arbitrage, since $\Sigma(K,T;\alpha)$ is free of static arbitrage by construction. In turn, inference about α defines $\Sigma(K,T;\alpha)$, from which $\sigma_{\text{LV}}(K,T)$ can be calculated easily via Dupire's formula. As above, we construct the filter

$$\pi_{t+1}^j = \frac{p(x_{t+1}|x_t,\alpha^j)\pi_t^j}{\sum_j p(x_{t+1}|x_t,\alpha^j)\pi_t^j} \tag{11}$$

following Gellert & Schlogl (2018). A secondary advantage is that by definition $c_t^M(K,T;\alpha)$ is available in closed form via

$$c_t^M(K,T;\alpha) = \text{Call}_{\text{BS}}\left(K,T,\sigma = \Sigma(K,T;\alpha)\right) \tag{12}$$

when computing the option observation term in the likelihood

$$p(x_{t+1}|x_t,\alpha) = f(s_{t+1}|s_t,\alpha)\prod_{i=1}^{N} g\left(c_{t+1}(K_i,T_i)|c_{t+1}^M(K_i,T_i),\alpha\right) \tag{13}$$

Some care is required when computing $f(s_{t+1}|s_t;\alpha^j)$ for the j^{th} particle. Remember that the α refers to parameters defining the implied volatility surface, which does not factor directly in the model for the underlying process

$$dS/S = rdt + \sigma_{\text{LV}}(S,t)\,dB \tag{14}$$

The way to handle this will depend on the approximation techique used, but I don't foresee a problem here. For example, if we exploit the Euler approximation

$$S_{t+1} = S_t + rS_t\Delta + \sigma_{\text{LV}}(S_t,t)\Delta\varepsilon_t \tag{15}$$

we only need a single value for $\sigma_{\text{LV}}(S_t,t)$ and this value can be computed using Dupire's equation applied at the point $\Sigma(K = S_t, T = t;\alpha^j)$.

MISC 2

The structures we need to compute relate to two quantities:

1. Simulation of the underlying $s_{1:t}^* = \{s_1, s_2, \ldots, s_t|\theta = \theta^*\}$

2. Calculation of the likelihood $p(x_t|x_{t-1}, \theta = \theta^*)$, which has two parts:

$$\begin{aligned} \text{underlying :} \quad & f(s_{t+1}|s_t, \theta = \theta^*) \\ \text{options :} \quad & g\left(c_{t+1}(K,T)|c^M_{t+1}(K,T), \theta = \theta^*\right) \end{aligned}$$

and both of the above require the volatility functions $\Sigma(K,T;\theta)$ and $\sigma_{\text{LV}}(S,T;\theta)$.

In this project we will specify $\Sigma(K,T;\theta)$, and we will get σ_{LV} from Dupire.
We need both, because the option component

$$c^M_t(K,T,\Sigma(K,T;\theta)) \tag{16}$$

is calculated directly with Σ, but the underlying component must be calculated with σ_{LV}, since we assume

$$dS_t/S_t = rdt + \sigma_{\text{LV}}(S_t,t;\theta)dB_t \tag{17}$$

We can calculate σ_{LV} from Σ using the Dupire Equation, but we will then still need to calculate $\partial w/\partial y$ and $\partial^2 w/\partial y^2$ where w is total variance and y is log-moneyness. This will mean using numeric derivatives or doing the maths by hand for each choice of Σ. To begin with, lets just code $\Sigma(K,T;\theta)$, and each time we need $\sigma_{\text{LV}}(S,T;\theta)$ compute it numerically via

$$\sigma^2_{\text{LV}}(K,T;\theta) = \frac{2\partial C(K,T;\Sigma(K,T;\theta))/\partial T}{K^2\partial^2 C(K,T;\Sigma(K,T;\theta))/\partial K^2} \tag{18}$$

using Black-Scholes formula, where we will begin with $r = 0$ for simplicity.

For any general choice $\Sigma(K,T;\theta)$ we will no longer have closed form expressions with which can simulate $s_{t+1}|s_t;\theta$, or calculate a likelihood $f(s_{t+1}|s_t;\theta)$. There are *many* choice we can use, but to begin with I suggest using:

1. Euler $o(\sqrt{\Delta t})$ Scheme: $s_{t+1} = s_t + s_t\sigma_{\text{LV}}(s_t,t;\theta)\Delta_t\varepsilon_t$
2. Euler $o(\sqrt{\Delta t})$ Approx: $f(s_{t+1}|s_t,\theta) \sim \mathbf{N}(s_t, \sigma^2_{\text{LV}}(s_t,t;\theta)\Delta_t)$

MISC 3

Here I just outline a few thoughts about what code we need to write. We require:

- A volatility function, which could be used for both σ_{LV} or Σ. It is just a callable object with two arguments of floating point type.
- An implementation for computing partial derivatives of two variable functions.
- Normal distribution sampling and likelihood calculations, ie
  - Simulate: $y \sim \mathbf{N}(\mu, \sigma^2)$
  - Likelihood: $f(y|\mu, \sigma^2)$
- Black-Scholes pricing for calls with standard inputs.
- Construction of options grids on $\{K \times T\}$ where K is a set of strikes and T a set of expiries. This requires keeping track of current time of the $s^*_{1:t}$ simulation, and the distance to each of T expiries. Also, when we approach the front month we must roll to the next expiry, and construct a new expiry at the end (ie we will always have T expiries in the data set). Idealy the required calculations for options are efficient over the grids. As an implementation detail, I will assume that $\#K$ remains fixed as s^*_t evolves. This may mean that *strike prices* may change, though the *number of strikes* will not.
- A vol and vol-time schedule for $s^*_{1:t}$. I want this simple but a setup that will be somewhat close to useable for live data. Simulations will be
  - Daily observations on weekdays.

- No observations on weekends, and assume no holidays.
- use $\Delta_t/2$ for weekends ie less variance on weekend days.