Contents

Below are some notes I wrote this afternoon after reflecting on our talk yesterday. I have a clear idea about some things, but sitting here this afternoon I am not sure about others. Let me write some pieces of information I think are relevant, and I will work towards the problem at hand and specify where exactly I am not as sure about the *project question*.

Introduction

We are investigating some projects related to local volatility models

$$dS/S = rdt + \sigma(S,t;\theta)dW \tag{1}$$

for the underlying S_t, where I have made explicit the dependance of the local volatility function on some parameter set θ. The model may be a parametric model like one of the Gatheral functions, or a non-parametrics model like one used in many papers. We may not even be interested in the functional form at all - instead with compute $\sigma_{ij} = \sigma(S = K_i, T = T_j)$ at specific discrete points in (S, T) space, and perhaps use this grid of local vols later in a pricing tree or finite difference grid.

Generally we are interested in

- Pricing options where the underlying satisfies Equation (1).
- Infering the values of the local function $\sigma(S,t;\theta)$ at specific strikes K and expiries T.

Actually the above points are closely related. The ability to price options with a given local vol is either useful or necessary.

- It is *necessary* in the sense that a specific local vol $\sigma(S,t;\theta)$ is not useful in and of itself - it is only useful if we can use it to price options or compute greeks. To do something useful with local volatilities we must be able to price options using it.
- It is *useful* because the ability to price options on arbitrary local volatility models allows us to infer a *true*, or *best* local volatility model that most closely matches market prices. This is the calibration approach to infering local volatilities from market prices. A more direct way is a standard manipulation of the Dupire PDE, as discussed later.

Computing Local Volatilities from Market Prices

There are two ways to infer local volatilities from observed market prices

- Infer $\sigma(S,t)$ from market prices (or market implied volatilies) using the Dupire Equation
- Calibrate $\sigma(S,t;\theta)$ from market prices using
  a. Numeric solutions of the Dupire PDE
  b. Numeric solutions of the Black-Scholes PDE
  c. Numeric solutions of the Fokker-Planck Equation

The Dupire PDE is

$$\frac{\partial C}{\partial T} = -rK\frac{\partial C}{\partial K} + \frac{1}{2}\sigma(K,T)\frac{\partial^2 C}{\partial K^2} \tag{2}$$

which is solved *from expiry* so the initial condition is the payoff function

$$c(t_0, S, t_0, K) = \max(S - K, 0) \tag{3}$$

and there is some subtlety as to the role of S and K (subtle to me anyway). Hereafter use the superscript c^M to denote an observed market price. The procedures are described below.

Infer Local Volatility from Observed Prices

This method is the most direct, and seems to be the most common approach.

1. Observe the set of market price $\{c^M(K_i, T_j)\}$ for all strikes and expiries.
2. Do the best of:
   a. Fit some smooth function to $\{c^M(K_i, T_j)\}$, then differentiate in T and twice in K. Plug these into the formula for σ_{ij} from Equation (2) after rearranging to make $\sigma(K,T)$ the subject.
   b. Do the same, except work in implied volatility instead of price space. There is an equivalent formula to use.
   c. Estimate the derivates using finite differences between expiries or across strikes. Plug these into the formula for σ_{ij} from Equation (2).

Calibrate Local Volatiliy to Observed Prices

This method is less direct, but it requires us to be able to price options with (relatively) arbitrary local volatilies. This is a necessary thing to be able to do to make this whole exercise useful, so we will have a quick think about it.

The calibration problem assumes that for each market price $c^M(K_i, T_j)$ we can compute a model generated equivalent $c(K_i, T_j; \theta)$ which in our case is parameterised by θ. The objective is to find $\widehat{\theta}$ where

$$\widehat{\theta} = \text{argmin } G(\theta), \text{ where}$$
$$G(\theta) = \sum_{ij} \text{cost}\left(c^M(K_i, T_j), c(K_i, T_j; \theta)\right)$$

This minimisation problem can be solved in a number of standard ways, including gradient methods with and without penalisation, or stochastic methods. It is interesting to see examples in Hamida & Cont (2013) of the challenges here. To solve this problem we must be able to compute a set of prices $\{c(K_i, T_j; \theta)\}$, which in almost every interesting case will not be available analytically. And as discussed previous, we will likely want to be able to do similar anyway after inferring the σ_{ij} directly using the Dupire Equation. Here are three general solutions.

1. Solve the Black-Scholes PDE using $\sigma(S, t; \theta)$ for each options $c(K_i, T_j)$. This is the obvious method and details are well understood. However this would be highly inefficient since we must compute $K \times T$ numerical PDE solutions *for each* θ required in the minimisation problem.

2. Solve the Dupire PDE (2) taking care to place every (K_i, T_j) on the grid. The solution at each grid point is the required price. This is very efficient as we only need to solve the PDE once per θ used in the minimisation.

3. But wait this feels like cheating! How is Dupire so efficient compared to Black-Scholes? Think about the use of the Fokker-Planck Equation in the Dupire Equation derivation. We can do similar here directly. The (risk neutral) density associated with the local vol process (1) satisfies the Fokker-Planck Equation

$$\frac{\partial f}{\partial t} = \frac{\partial}{\partial S}\left[\frac{1}{2}\frac{\partial}{\partial S}\left(\sigma^2(s,t;\theta)f\right) - rSf\right] \tag{4}$$

with a point source initial condition at $S = S_t$ and appropriate boundary conditions. Make sure that each time expiry on the way to the longest expiry T is included on the compute grid. After integrating forward we have the required density at each expiry we need. Then, we numerically evaluate the expectation of each option's payoff and discount appropriately. These weighted sums are essentially free. So this approach is very general *and* efficient.

Filtering Latent States in the Local Volatility Function

There are two components here:

1. Can we make inference about latent states in a local volatility model?
2. What does this say about local volatility surfaces?

I thought I understood the motivation here when we spoke, but now that I think about it again I'm not so sure. My memory was of a statement to the effect of:

```
Why do local volatility surfaces change over time?  Is this evidence
of latent states impacting the local volatility function?
```

If I have the question correct, I am now not sure why this is meaningful/interesting. But if it is, we have two related problems to look at:

- Can we make inference about the parameters θ which drive the latent stochastic factor?
- Assuming θ is known, can we just track the unobserved latent state v?

Let me quickly specify first thoughts about a filter to track a latent volatility state. Maybe this will clarify some of my uncertainty.

A Basic Filtering Setup

Assume local volatility is of the form $\sigma(S, t; v_t)$ where v_t is an unobserved random variable. Presumably information about v_t is contained in options prices. If we observe a sequence of volatility surfaces, can we make inference about v_t? Let's sketch a simple approach.

We observe a sequence of market price grids $\{C_1, C_2, \ldots, C_t\}$ where each C_t is a $K \times T$ discrete grid of market call prices at time t. From this (or from implied volatilities calculated from the C_t) we compute a similar sequence of grids $\{\Sigma_1, \Sigma_2, \ldots, \Sigma_t\}$ of local volatilities, where each element Σ_{ij} inside a single Σ_t might be computed from the Dupire Equation (2). A separate sequence $\{v_1, v_2, \ldots, v_t\}$ remains latent, but is driving volatilities via the local volatility function $\sigma(S, t; v)$ that is underlying the price C_t. This means each element in Σ_t is a function of v ie $\Sigma_{ij} = \Sigma(K_i, T_j, v_t)$.

To write a filter write the observed local volatilities at a specific time as a vector

$$\mathbf{\Sigma} = (\Sigma(K_1, T_1, S_t, v_t), \ldots, \Sigma(K_N, T_N, S_t, v_t))^T \tag{5}$$

where I have switched to using a single index for the $N = K \times T$ options on the grid. At time t all arguments except v_t are known (eg K, T etc) so I ignore. Thus, the observation vector at t is $\mathbf{\Sigma}_t = g(v_t)$, ie a vector valued non-linear function of the latent state v_t. We need to assume the Markov property for v_t with some known transition density. Then we have a standard filter

$$v_t | v_{t-1} \sim p(v_t | v_{t-1}) \quad \text{state equation}$$
$$\mathbf{\Sigma}_t = g(v_t) \quad \text{observation equation}$$

We need to specify some plausible model for $p(v_t|v_{t-1})$, and hope we can use the $\mathbf{\Sigma}_t$ to make inference about v_t. This includes two (possibly joint) problems:

- Track the latent variable v_t over time.
- Make inference about possible parameters governing the evolution of v_t.

An Example Setup

We should set up a working example where we understand the evolution of v_t, and see if any filter we setup can accurately make inference about it. Here is one possibility.

1. Take $\sigma(S, t, v_t) = \sqrt{v}$, so that the following model governs the underlying:

$$\begin{aligned} dS/S &= rdt + \sqrt{v}\, dB \\ dv &= \kappa(m - v)dt + \xi\sqrt{v}\, dB \end{aligned}$$

where we could assume parameters $\theta = (\kappa, m, \xi)$ are known, or unknown (harder). This is the Heston model.

2. Generate data (i). We do this by simulating underlying prices $\{S_1, S_2, \ldots, S_T\}$ and pricing corresponding grids of options $\{C_1, C_2, \ldots, C_t\}$ using the Heston model.
3. Generate data (ii). Compute local vols from the above price grids.
4. Work on a particle filter for this problem, which is now in standard form. A model is described previously. Our filter assumes no observation error (but we could add that), and we know the transitions $p(v_t|v_{t-1})$ are distributed as non-central chi-squared.

What does this mean?

What would this process show? I keep thinking of the comment:

```
Why do local volatility surfaces change over time?  Is this evidence
of latent states impacting the local volatility function?
```

I have a couple of thoughts on this.

1. Maybe it is an unobserved latent state. Traders try to capture the state of the local volatility surface, and as the latent stochastic volatility process evolves they modify their estimates. Here a particle filter would be ideal to uncover this process.
2. Local volatility has no latent stochastic factors, but its functional form is determined by a set of unknown parameters whose values traders disagree on or are unable to accurately estimate. Or traders disagree on the functional form itself. Some think it is a non-parametric surface of a certain shape, others think it is a Gatheral-like surface. This might be considered more of a calibration problem - as we are not considering underlying stochasticity.
3. Observed prices, and therefore local volatility include non-market factors that mean that the interpretation of local volatility as typical vol conditions into expiry is polluted by considerations such as (i) large fund managers taking small losses in expectancy to buy insurance (ii) regulatory risk requirements etc etc (iii) other technical, microstructure, psychological factors etc etc.