1 Introduction

The purpose of this document is to detail the architecture of the universal pricing machine in the case of the local volatility model.

2 Local volatility model in a nutshell

Assuming that it pays no dividend and that the interest rates are zero, we find the following dynamics under the risk neutral probability measure $\mathbb{P}^*$:

$$\frac{dS_t}{S_t} = \sigma(S_t, t) dW_t \tag{1}$$

where W_t is the standard Brownian motion.

What matters for us is that, in that general diffusive model, one can relate the model parameter $\sigma(S, t)$ to the prices $C(K, T)$ of the call options of maturity T and strike price K:

$$\Lambda(K, T) = \frac{K}{2}\sigma^2(K, T) = \frac{\frac{\partial C(T,K)}{\partial T}}{K\frac{\partial^2 C(T,K)}{\partial K^2}} \tag{2}$$

That is going to be the corner stone of our machine learning approach.

3 Challenge of the pricing machine

The goal of the machine is to determine the probability density function of the stock $p_S(s, t)$ under the measure $\mathbb{P}^*$ i.e. the probability of ending up in the vicinity of s at time t given that the stock was at S_0 at time 0.

That density is related to the option price by:

$$p_S(K, T) = \frac{\partial^2 C(K, T)}{\partial K^2} \tag{3}$$

For that, we must use as input the model parameter $\nu(K, T)$ and determine from it a solution meeting the Dupire's formula Eq.2

To complete the picture, we are going to consider that the solution can be represented by some Gaussian mixture. That is not really restrictive since any function can be decomposed as a series of Gaussian radial basis functions.

4 Gaussian radial basis functions

4.1 Expression

Gaussian radial basis functions are natural interpolating functions and, in the world of information theory, they correspond to a mixture of Gaussian variables. As the stock price S is log-normal, it is natural to introduce the log price X_t as:

$$X_t \triangleq \ln(S_t/S_0) \tag{4}$$

And we have:

$$p_X(x,T) = p_S(S_0 e^x, T) S_0 e^x \tag{5}$$

$$p_S(S,T) = p_X(\ln(S/S_0), T)/S \tag{6}$$

The Gaussian mixture model reads:

$$p_X(x,t) = \sum_{i=0}^{I} a_i(t) g(x - c_i(t), b_i(t)) \tag{7}$$

where

$$g(x,b) = \frac{1}{\sqrt{2\pi b}} e^{-\frac{x^2}{2b}} \tag{8}$$

The parameters $a_i(t) \geq 0, c_i(t), b_i(t) \geq 0$ are in turn functions of the time and $T_i(t)$ must be an increasing function. Those functions are typically going to be splines or coefficients of polynomials. We will see later on how to can formulated more explicitly in terms of neural network architecture.

4.2 Rationales of the choice

The reasons for that choice are manifold:

- If the local volatility surface ν is interpolated from a Gaussian radial basis function then it is clear from the Fokker Planck equation that once the solution is in that class it is going to stay within it (if we accept the idea that the number of centroids may grow over time). Clearly, the initial condition being a Delta distribution, only one single centroid is enough for small maturities.

- From the Chapman Kolmogorov identity, we can consider that if the probability measure at time T is approximated by some discretized measure (which all amounts to saying that the solutions can be quantized and remarking that discrete measures are degenerated cases of the Gaussian mixture), the probability density at the next step is of the form 7.

- It is conditionally Gaussian just as the local volatility model.

The interpretation of 7 in terms of financial market model is clear. The probabilities a_i are the discrete big picture scenarios under which the market behaves like a Black Scholes model of predefined volatility. Or, instead of saying that there is a single constant volatility coefficient as we do when plugging a Black Scholes model in, it all amounts to saying that there are I economics experts giving their views on the market and the model is the combination of those alternative market views, weighted by the credibility of each expert.

4.3 Black Scholes connection

The Gaussian mixture being a mixture of Black Scholes models, we get for free the European option prices:

$$
\begin{aligned}
C(T,K) &= \int_{\mathbb{R}} (S_0 e^{\xi} - K)_+ p_X(\xi, T) d\xi && (9)\\
&= \sum_{i=0}^{I} a_i(T) \int_{\mathbb{R}} (S_0 e^{\xi} - K)_+ g(\xi - c_i, b_i(T)) d\xi && (10)\\
&= S_0 \sum_{i=0}^{I} a_i(T) \psi(c_i(T), b_i(T), K/S_0) && (11)
\end{aligned}
$$

where

$$
\begin{aligned}
&\psi(c_i, b_i, K) = \Phi_i \mathrm{N}(d_+(K)) - K\mathrm{N}(d_-(K)) && (12)\\
&\Phi_i = e^{c_i + \frac{1}{2} b_i} && (13)\\
&d_- = \frac{c_i - \ln(K)}{\sqrt{b_i}} && (14)\\
&d_+ = d_- + \sqrt{b_i} && (15)
\end{aligned}
$$

4.4 Network architecture

Crucially, $a_i(t), b_i(t), c_i(t)$ must be the outputs of the neural network and we must have more explicitly:

$$
\begin{aligned}
a_i(t) &= A_i(t, \mathcal{W}_a(\mathbf{\Lambda}_{0,t})) && (16)\\
b_i(t) &= B_i(t, \mathcal{W}_b(\mathbf{\Lambda}_{0,t})) && (17)\\
c_i(t) &= C_i(t, \mathcal{W}_c(\mathbf{\Lambda}_{0,t})) && (18)
\end{aligned}
$$

where $\mathbf{\Lambda}_{t_1,t_2}$ is the collection of all the local volatilities $\Lambda(K,\tau)$ with τ between t_1 and t_2.

It means that the probability of the stock to be at some value at some date t must not depend on the volatility of the stock after t.

The general idea would be then to find the parameters of the networks $\mathcal{W}_a, \mathcal{W}_b, \mathcal{W}_c$ minimizing the gap to Dupire identity:

$$
E^2_{\mathcal{W}_{a,b,c}} = \int_0^T \int_{K_{\min}}^{K_{\min}} \varepsilon(k,t)^2 dk dt \qquad (19)
$$

where

$$
\varepsilon(k,t) = \frac{\partial C(k,t)}{\partial t} - \Lambda(k,t) k p_S(k,t) \qquad (20)
$$

and

$$
k p_S(k,t) = \sum_{i=0}^{I} a_i(t) g(\ln(k/S_0) - c_i(t), b_i(t)) \qquad (21)
$$

and

$$\frac{\partial C(k,t)}{\partial t} = \sum_{i=0}^{I} \frac{da_i(t)}{dt}\psi_i(k,t) + a_i(t)\frac{d\psi_i(k,t)}{dt} \tag{22}$$

with

$$\psi_i(k,t) = \psi(c_i(t), b_i(t), k/S_0)$$

Hence:

$$\frac{d\psi_i(k,t)}{dt} = \frac{\partial\psi(c_i(t), b_i(t), k/S_0)}{\partial c}\frac{dc_i(t)}{dt} + \frac{\partial\psi(c_i(t), b_i(t), k/S_0)}{\partial b}\frac{db_i(t)}{dt}$$

With

$$\frac{\partial\psi(c,b,k)}{\partial b} = g(c+b-\ln(k), b) \tag{23}$$

$$\frac{\partial\psi(c,b,k)}{\partial c} = N\left(\frac{c+b-\ln(k)}{\sqrt{b}}\right)e^{c+b/2} \tag{24}$$

5 Time Block approach

For clarification sake, we are going to slice the time domain into blocks and demand that the time dependence of each parameter be piecewise linear.

6 First Time block

To make it more concrete, let us now detail the first time block of the neural network architecture.

That is going to correspond to times between 0 and T_1. T_1 could typically be 1 year.

The initial condition is a Dirac distribution in $X_0 = 1$.

$$b_i(0) = 0 \tag{25}$$

And

$$b_i(t) = tb_i(1) \tag{26}$$

The centroids are going to be extrapolated flat for the first block:

$$c_i(0) = c_i(1) \tag{27}$$

So are the weights:

$$a_i(0) = a_i(1) \tag{28}$$

We are then left with the following 3 vectors to predict:

$$\boldsymbol{a}(1) = [a_0(1), \cdots, a_I(1)] \tag{29}$$

$$\boldsymbol{b}(1) = [b_0(1), \cdots, b_I(1)] \tag{30}$$

$$\boldsymbol{c}(1) = [c_0(1), \cdots, c_I(1)] \tag{31}$$

So the only thing we have now to determine is:

$$\boldsymbol{a}(1) = \boldsymbol{A}(1, \mathcal{W}_a(\boldsymbol{\Lambda}_{0,T_1})) \triangleq \mathcal{W}_a^1(\boldsymbol{\Lambda}_{0,T_1}) \tag{32}$$

$$\boldsymbol{b}(1) = \boldsymbol{B}(1, \mathcal{W}_b(\boldsymbol{\Lambda}_{0,T_1})) \triangleq \mathcal{W}_b^1(\boldsymbol{\Lambda}_{0,T_1}) \tag{33}$$

$$\boldsymbol{c}(1) = \boldsymbol{C}(1, \mathcal{W}_c(\boldsymbol{\Lambda}_{0,T_1})) \triangleq \mathcal{W}_c^1(\boldsymbol{\Lambda}_{0,T_1}) \tag{34}$$

with

$$\mathcal{W}_a^1(\boldsymbol{\Lambda}_{0,T_1}) \geq 0 \tag{35}$$

$$\sum_{i=0}^{I} \mathcal{W}_a^1(\boldsymbol{\Lambda}_{0,T_1})_i = 1 \tag{36}$$

Let us write:

$$\boldsymbol{\Lambda}_{0,T_1} = [\Lambda(k_m, t_n)]_{m\in\{0,\cdots,M\},n\in\{0,\cdots,N\}} \tag{37}$$

The matching error given by Eq.19 now reads:

$$E^2_{\mathcal{W}^1_{a,b,c}} = \sum_{m=1}^{M-1} \sum_{n=1}^{N} \varepsilon(k_m, t_n)^2 \Delta k_m \Delta t_n \tag{38}$$

$$\Delta k_m \triangleq k_m - k_{m-1} \tag{39}$$

$$\Delta t_n \triangleq t_n - t_{n-1} \tag{40}$$

In that case, we also have a simple expression for $\varepsilon(k_m, t_n)$:

$$\varepsilon(k,t) \triangleq \sum_{i=0}^{I} \varepsilon_i(k,t) \tag{41}$$

$$\varepsilon_i(k,t) = b_i(1)\frac{\partial\psi(c_i(1), tb_i(1), k/S_0)}{\partial b} - \Lambda(k,t)g(\ln(k/S_0), c_i(1), tb_i(1)) \tag{42}$$

7 Next time blocks

For $t \in [T_Y, T_{Y+1}]$, let us posit $\Delta_Y t = t - t_Y$. We generalize inductively as follows:

$$\boldsymbol{a}(t) = \boldsymbol{a}(T_Y) + \Delta_Y t(\boldsymbol{a}(T_{Y+1}) - \boldsymbol{a}(T_Y)) \tag{43}$$

$$\boldsymbol{b}(t) = \boldsymbol{b}(T_Y) + \Delta_Y t\boldsymbol{b}'(T_{Y+1}) \tag{44}$$

$$\boldsymbol{c}(t) = \boldsymbol{c}(T_Y) + \Delta_Y t(\boldsymbol{c}(T_{Y+1}) - \boldsymbol{c}(T_Y)) \tag{45}$$

We must determine

$$\boldsymbol{a}(T_{Y+1}) = \mathcal{W}_a^{Y+1}(\boldsymbol{\Lambda}_{T_Y,T_{Y+1}}, \boldsymbol{\pi}_{T_Y}) \tag{46}$$

$$\boldsymbol{b}'(T_{Y+1}) = \mathcal{W}_b^{Y+1}(\boldsymbol{\Lambda}_{T_Y,T_{Y+1}}, \boldsymbol{\pi}_{T_Y}) \tag{47}$$

$$\boldsymbol{c}(T_{Y+1}) = \mathcal{W}_c^{Y+1}(\boldsymbol{\Lambda}_{T_Y,T_{Y+1}}, \boldsymbol{\pi}_{T_Y}) \tag{48}$$

where $\boldsymbol{\pi}_{T_Y}$ is the representation of the Gaussian mixture at time T_Y:

$$\boldsymbol{\pi}_{T_Y} = [\boldsymbol{a}(T_Y), \boldsymbol{b}(T_Y), \boldsymbol{b}(T_Y)] \tag{49}$$

We must have:

$$\mathcal{W}_a^{Y+1}(\boldsymbol{\Lambda}_{T_Y,T_{Y+1}}, \boldsymbol{\pi}_{T_Y}) \geq 0 \tag{50}$$

$$\sum_{i=0}^{I} \mathcal{W}_a^{Y+1}(\boldsymbol{\Lambda}_{T_Y,T_{Y+1}}, \boldsymbol{\pi}_{T_Y})_i = 1 \tag{51}$$

$$\mathcal{W}_b^{Y+1}(\boldsymbol{\Lambda}_{T_Y,T_{Y+1}}, \boldsymbol{\pi}_{T_Y}) \geq 0 \tag{52}$$

As for the first block, we need to determine the parameters of each neural network to minimize Eq. 2 encoding the matching gap to Dupire's formula.

8 Remarks

- The time step within each block can be taken equal to one month.
- We might have to do something to prevent centroids c_i from cycling while moving from one time block to the next one. Penalizing large variations could then be considered.
- Topology matters for both that problem the concept of distance in the local volatility surface must be kept. The parameter b_i is both a time and the typical radius of interaction of a centroid. Said differently, I would expect the update of the parameters to be essentially driven by the local volatility in the region located within a distance b_i of the centroid c_i. I would even expect more precisely that b_i must depend on the average level of the local volatility while the change in a_i and c_i must rather be sensitive to the gradient of the local volatility.