Learning Optimal Hedging for Options

Marcos Costa Santos Carreira

École Polytechnique - CMAP

Introduction

Our goal is to learn a pricing and hedging policy for vanilla options. In order to learn from history, we must first understand the limitations of this approach by examining how learning works with known models. The project is to investigate:

- Black and Scholes (GBM)
- Power law distributions for the tails
- Local volatility
- Heston-Nandi
- Heston
- Rough Heston
- Rough Bergomi

In doing so, we expect to retrieve theoretical results, investigate the need for recalibration within the models and classify the optimal hedging of historical returns as a mixture of the models above, while questioning to ability to learn from an ensemble of paths.

Metrics

The final P&L of a hedged portfolio of a vanilla option and its underlying asset can be written (with interest rates, dividends and repo rates equal to zero) as:

$$\Pi_T = max\left(\phi(S_T - K, 0)\right) - P_{K,T,0} + \sum_{t=0}^{T-dt}\left(w_{K,T,t}(S_{t+dt} - S_t)\right)$$

Where:

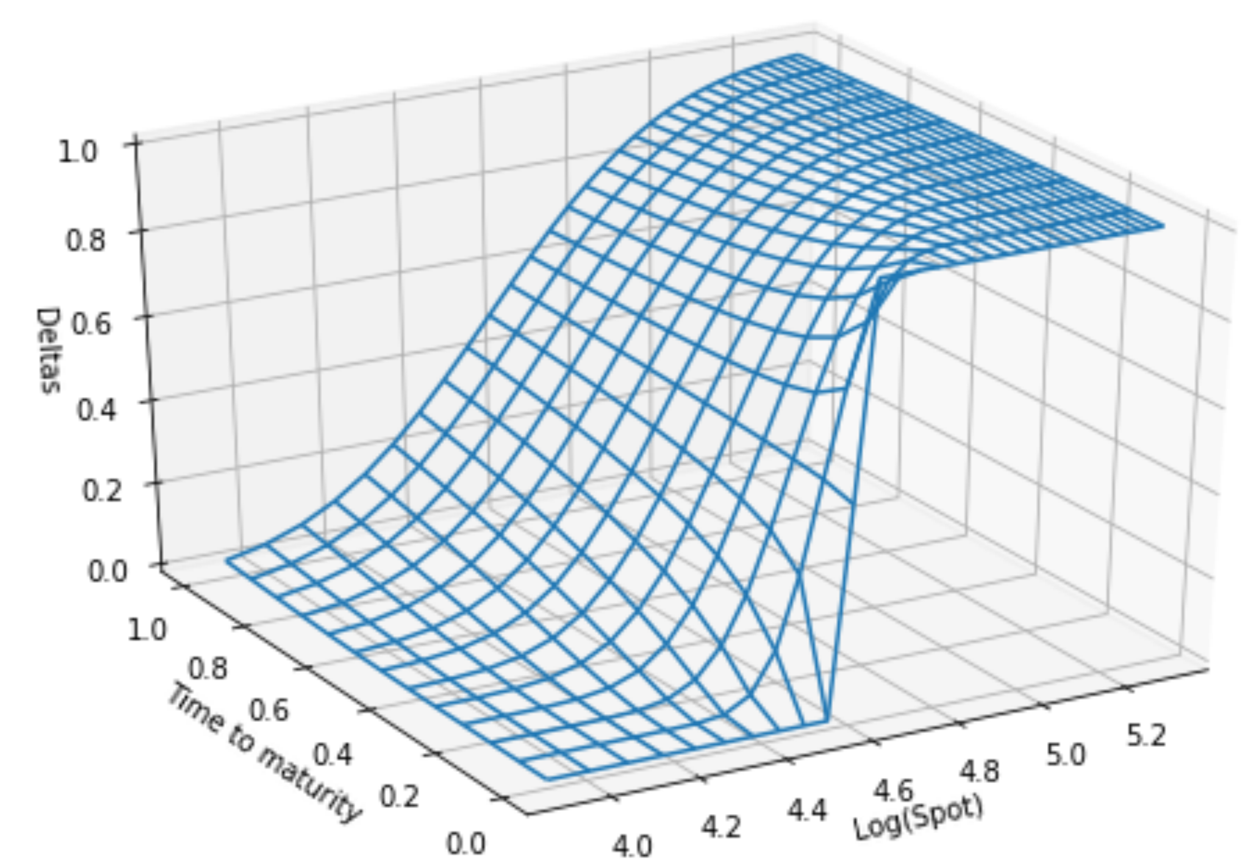
- $w_{K,T,t}$ is the weight of the asset used in the hedging portfolio at time t (the delta of the option)
- S_t is the price of the asset at time t
- K is the strike / exercise price of the option
- $P_{K,T,0}$ is the price at time 0 of the option with strike K and maturity at T

Without knowing anything about the process or the history of the data, what can we know about the pricing function P and the associated weights w?

We can start by assuming that:

- $w_{K,T,t}$ goes from 0 to ±1 continuously and monotonically for each pair T,t
- $w_{K,T,t}$ becomes steeper the closer t gets to T

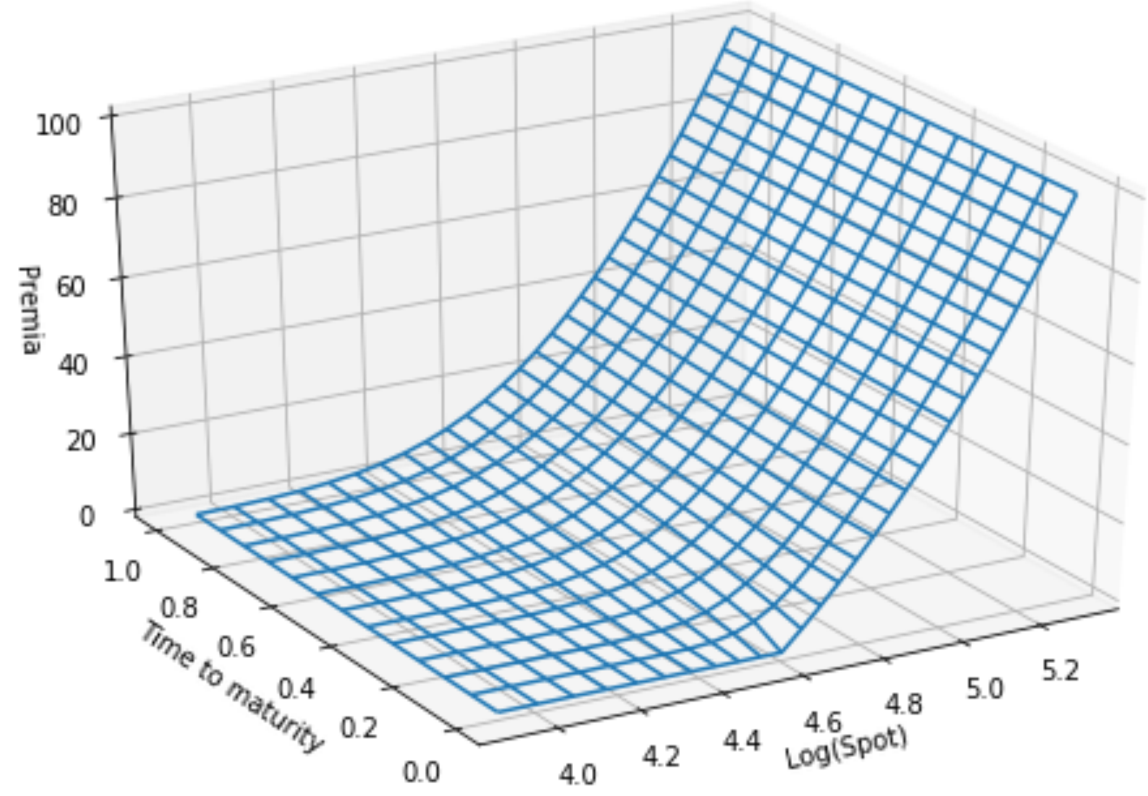
A sigmoid is a natural candidate for this function:



And we can cheat by using the usual $N(d_1)$ Black and Scholes function as our sigmoid template, with the implied volatility scaled by $\sqrt{T-t}$ as our input to control the slope of the sigmoid.

As Rebonato once said: "The implied volatility is the wrong number to put on the wrong formula to arrive at the same price".

Here the price is going to be used only at the start of the period:



Learning

At time t=0:

- We define different maturities T_m
- For each maturity T_m we define strikes $K_{m,j}$

For each time t from 0 to T:

- The volatility "surface" $\sigma_{m,j}$ is defined such that the deltas corresponding to $w_{K,T,t}$ are calculated with it

For t=0 the premia are calculated as well

For each maturity T the payoffs are calculated

We end up with a series of P&Ls for each set of strikes $K_{m,j}$ for each path (where the same random numbers are used on the spot process),

For the Black and Scholes paths, the lack of features to distinguish one path from another enables an efficient learning with an ensemble of paths; by selecting paths that not only have the same constant volatility input but also the same realized volatility we can minimize the variance of the results at each maturity / strike pair.

On the other hand, paths where something different or unique happens are less amenable for this method. If the volatility process is on a high vol state in one path but in a low vol state in 99 other paths averaging or using the variance should lose information.

For that, we are going to simply add the absolute values of the P&L of each option, which ends up as a kind of vega-weighted calibration, while being able to capture the damage of a tail event.

This will be done on a path-by-path basis, and we will look at the worst cases.

Another way to put it is:

- Even though different models might be calibrated to the same initial smile that price options equally for everyone
- i.e. the implied volatility surface represents the initial market prices, which all models must fit
- The delta hedges that each model prescribes are not the same
- Some models will require vega or forward variance hedges
- But we want to know what delta-hedging only can (or cannot) achieve (i.e. how far from idealized models the real world is)

With that said, we should be able to see whether recalibration of model parameters to generate a "new" implied volatility surface is necessary for optimal hedging, so we can understand better the results for historical data.

Technology

To do so we'll need to run several simulations and optimizations, and we're glad to have the support of Techila Technologies, who are helping us not only in setting up efficient computation routines that can be parallelized and run (on Python) at Google Cloud but with credits to do it.

Conclusions / next steps

- Working with leading academics (Mathieu Rosenbaum, Elisa Alós) to understand better the theoretical results on optimal hedging
- Working with Techila Technologies to create reproducible research in Python
- Calibrate model results
- Learn optimal hedges on historical data
- Test ability to transfer results within and between asset classes

References

- "Learning Interest Rate Interpolation", M. Carreira
- "Stochastic Volatility Modeling", L. Bergomi
- "Roughening Heston", O. El Euch, J. Gatheral, M. Rosenbaum
- "Regimes of Volatility: Some Observations on the Variation of S&P 500 Implied Volatilities", E. Derman
- "Delta Hedging Vega Risk?", S. Crépey
- "QLBS: Q-Learner in the Black-Scholes(-Merton) Worlds" I. Halperin
- "Learning minimum variance discrete hedging directly from the market", K. Nian et al
- "Quantifying Model Performance", A. Antonov et al
- "Deep Hedging", H. Buehler et al
- "Option Pricing under Power Laws: A Robust Heuristic", N. Taleb et al

Contact

My profile: https://quantreg.com/people/marcos-carreira/

Contact: marcos.costa-santos-carreira@polytechnique.edu

Link to paper and Jupyter notebook on the page soon