Implied Stock Probability Mass Function from Market European Option Prices Methodology

Philip Felizarta

February 2022

1 Motivation

The purpose of this project is to retrieve a probability mass function from market call and put european option prices. I use tensorflow and tensorflow probability to learn the implied PMF.

2 Option Price and Stock Price Probability Density

$$u_{call} = \int_{K}^{\infty} (x-k)s(x)dx$$

$$u_{put} = \int_{0}^{K} (K-x)s(x)dx$$

$$L_{model} = \frac{1}{2M}\sum_{j}^{M}\frac{1}{2}(ln(u_{model,j}) - ln(u_{true,j}))^2 + \frac{1}{2}(u_{model,j} - u_{true,j})^2$$

3 Parameterizing $s(.)$ and Discretizing the Optimization Problem

$$s = softmax(z)$$

Where z will be a learned vector, s_i will represent:

$$P(S_T = x_i)$$

x_i will is an element of the mesh spanning the moneyness of strike prices. Thus, the option prices can be estimated like such (N=1000 in the code)...

$$u_{call}(K) = \sum_{i}^{N} s_i(x_i - K)^+$$

$$u_{put}(K) = \sum_i^N s_i(K - x_i)^+$$

4 Continuity Regularization

To achieve "smooth" functions I utilized a neat regularization trick where I sum the square distances between neighboring parameters.

$$\min_z \gamma \sum_i^{N-1} \frac{(z_i - z_{i+1})^2}{|x_i - x_{i+1}|}$$

I chose to weigh the square difference between points by the inverse of the absolute difference between corrosponding mesh points because the mesh is not linear (its geometric so that I may concentrate on PMF on values near at-the-money).

5 Optimization Problem

$$\min_z L_{call}(z) + L_{put}(z) + \gamma \sum_i^{N-1} \frac{(z_i - z_{i+1})^2}{|x_i - x_{i+1}|}$$

I use tensorflow probability to optimize this equation with L-BFGS.