Research Proposal—Wang Xizhuo

Research title:

Data science application in finance: improving delta in hedging strategy

Introduction:

Practitioners often price options based on the Black-Scholes Model. However, the constant volatility assumption in the model can be easily violated in the real financial market. In practice, volatility can vary with time, underlying asset price or strike price. Since volatility is an important part in the calculation of delta, incorrect volatility will lead to inaccurate delta.

Various adjustments on delta have been proposed by previous research papers. They took volatility smile, stochastic volatility or other empirical results and statistic models into account. This research aims to test which model generates the delta that minimizes the variance of changes in the value of an accordingly delta-hedged position. Equipped with the functional programming language, Clojure, the research will test the improvement of various deltas using the Clojure back-testing library. It will compare reduction in variance to generate conclusion.

Data:

Daily prices of options on S&P 500 and the underlying

Hedging parameters based on Black-Scholes Model

Theory and Methodology:

Baseline Theory: Black -Scholes Model

$C = N(d_1)S_t - N(d_2)Ke^{-rt},\ where\ d_1 = \frac{\ln\frac{S_t}{K} + \left(r + \frac{\sigma^2}{2}\right)t}{\sigma\sqrt{t}}, d_2 = d_1 - \sigma\sqrt{t}$

$\Delta_{\mathrm{BS}} = N(d_1)$

Volatility smile-adjusted delta: $\delta_A = \Delta_{BS} + v_{BS}\left(\frac{\partial\sigma}{\partial K}\right)$;

Stochastic volatility delta: Heston Volatility Model and SABR Volatility Model;

Other deltas calculated based on empirical results.

Timeline:

Week 1: Conducting literature review for specifying the models; getting familiar with the domain specified language Clojure

Week 2: Collecting data from the related website; working out the hedging position outline; continuing with the learning process of Clojure

Week 3: Going through Clojure back-testing library and its application; staring the code of data analysis

Week 4: Continuing with Clojure back-testing library and the code

Week 5: Testing the hedging positions; plotting the value of the positions; generating results by making comparisons

Week 6: Writing report based on the comparison results; composing the poster for presentation