Option Price

Yunshu Zhang

June 2021

1 Literature Review/Survey

According to machine learning and deep learning methods, most of the previous researchers focus especially on the neural networks method (NN), the other algorithms being unexplored. Ivascu [2] propose three three different decision tress algorithms: random forest, XGBoost, LightGBM. Jang [3] implements Generative Bayesian learning model with a prior incorporating a financial structure. Training the model using an artificial sample generated from a risk-neutral financial option model- CGMY. Arin [1] used DMLP based models are proposed namely, VS-BS/DMLP, VS-2DMLP, C-BS/DMLP, and C-2DMLP.

Another important motivation is that option data rare imbalanced because of the liquidity of options. Imbalanced data predominantly learns the patterns of classes that constitute the majority, whereas the patterns in small classes are insufficiently learned. Hence, Jang [4] trys to use various parametric methods to generate additional balanced training data, such as BI, BS, MC, FDM.

The data that most of the time used are all European call options that had as support the WTI oil futures contracts (Reuters Index Code: CL) traded on the Chicago Mercantile Exchange, SP 500 and 100, EuroStoxx50, Hang Seng, and TAIEX.

2 Option Data

```
# Stock
stock = 'IBM'
# Expriation date
expiry = '11-28-2021'
# Strike price
K = 150
# time to maturity
t = (datetime.strptime(expiry, "%m-%d-%Y") - datetime.utcnow()).days / 365

today = datetime.now()
one_month_ago = today.replace(month=today.month-1)

df = web.DataReader(stock, 'yahoo', one_month_ago, today)
```

This could give us one month IBM stock data from yahoo finance.

2.1 Underlying Price

Today's stock close price price

```
S = df['Close'].iloc[-1]
```

2.2 Sigma

We are able to calculate the sigma value by multiplying the standard deviation of the stock returns over the past year by the square root of 252 (number of days the market is open over a year).

```
# Previous day's price
df = df.assign(close_day_before=df.Close.shift(1))
# Stock Return
df['returns'] = ((df.Close - df.close_day_before)/df.close_day_before)

sigma = np.sqrt(252) * df['returns'].std()
```

2.3 Risk Free Rate

The 10-year U.S. treasury yield which you could get from ^TNX.

```
r = web.DataReader("^TNX", 'yahoo', one_day_ago, today)['Close'].iloc[-1]
```

All in all, we could get the option price, implied volatility and greeks for this specific example.

```
print('The Option Price for ' + stock + ' on ' + expiry + ' is: ', bs_call(S, K, t, r, sigma
print("Implied Volatility: " + str(100 * call_implied_volatility(bs_call(S, K, t, r, sigma
print("Delta: " + str(delta('c', S, K, t, r, sigma)))
print("Gamma: " + str(gamma('c', S, K, t, r, sigma)))
```

3 Binomial Options Model- American Option

- Input required stock and option parameters:
  - S_0: current or spot price of the underlying security
  - r: strike price of the option K; risk-free rate
  - T: time to maturity or expiration of the option
  - σ: annualized volatility of the return of the underlying security
- Calculate Tree parameters:
  - u and d: up and down size movements
  - p and 1-p: the probabilities of the up and down movements
- Generate the trees: three trees are generated:
  - Stock Tree
  - Probability Tree
  - Option Payoff Tree
- Discount backwards: the option is priced by obtaining the payoffs at the last time step, and then working backwards through the tree, discounting the value of the option by the risk free rate at each step, until you get to time zero. if the option value at a node is greater than the value obtained from discounting back to that node, then the greater early exercise value replaces the value discounted back to that node. The greater value at that node is then propagated back through the tree.

3.1 Cox Ross Rubinstein Binomial Tree

The stock moves up in increments of $u = e^{\sigma\sqrt{dt}}$ and down in increments of $d = \frac{1}{u}$ at each time step of length $dt = \frac{T}{n}$. The probability of an up move is $p = \frac{e^{r*dt}}{u-d}$

4 Black Scholes Model- European Option

The Black-Scholes formula, for call:

$$C = N(d_1)S_t - N(d_2)Ke^{-rt} \tag{1}$$

and for put:

$$P = Ke^{-rt}N(-d_2) - S_tN(-d_1) \tag{2}$$

where $d_1 = \frac{ln(\frac{S_t}{K})+(r+\frac{\sigma^2}{2})t}{\sigma\sqrt{t}}$ and $d_2 = d_1 - \sigma\sqrt{t}$

- C = call option price
- N = CDF of the normal distribution
- S_t= spot price of an asset
- K = strike price
- r = risk-free interest rate
- t = time to maturity
- σ = volatility of the asset

5 Greek Letters

Option Greeks measure the exposure of option price or option delta to movement of different factors such as the underlying price, time and volatility.

First Order Greeks: Delta, Vega, Theta, Rho **Second Order Greeks**: Gamma

5.1 Delta

Delta is the rate of change of the option price with respect to the price of the underlying asset. It measures the first-order sensitivity of the price to a movement in stock price S.

$$\Delta = \frac{\partial C}{\partial S} \tag{3}$$

For call: $\Delta = N(d_1)$ and for put: $\Delta = -N(-d_1)$

5.2 Gamma

Gamma is the rate of change of the portfolio's delta with respect to the underlying asset's price. It represents the second-order sensitivity of the option to a movement in the underlying asset's price.

$$\Gamma = \frac{\partial^2 C}{\partial S^2} \tag{4}$$

For call and put: $\Gamma = \frac{N^{'}(d_1)}{S\sigma\sqrt{t}}$.

5.3 Vega

The Vega is the rate of change in the value of the option with respect to the volatility of the underlying asset.

$$V = \frac{\partial C}{\partial \sigma} \tag{5}$$

For call and put: $V = SN^{'}(d_1)\sqrt{t}$.

5.4 Theta

Theta is the rate of change of the value of the option with respect to the passage of time. It is also referred to as the time decay of the portfolio.

$$\Theta = \frac{\partial C}{\partial t} \tag{6}$$

For call: $\Theta = -\frac{S\sigma N^{'}(d_1)}{s\sqrt{t}} - rKe^{-rt}N(d_2)$ and for put:$\Theta = -\frac{S\sigma N^{'}(d_1)}{s\sqrt{t}} + rKe^{-rt}N(d_2)$.

5.5 Rho

Rho is the rate of change of the value of a derivative with respect to the interest rate.

$$\rho = \frac{\partial C}{\partial r} \tag{7}$$

For call: $\rho = -tKe^{-rt}N(d_2)$ and for put:$\rho = -tKe^{-rt}N(-d_2)$.

5.6 Implied Volatility

It is defined as the expected future volatility of the stock over the life of the option. It is directly influenced by the supply and demand of the underlying option and the market's expectation of the stock price's direction.

References

[1] Efe Arin and A Murat Ozbayoglu. Deep learning based hybrid computational intelligence models for options pricing. *Computational Economics*, pages 1–20, 2020.

[2] Codrut-Florin Ivascu. Option pricing using machine learning. *Expert Systems with Applications*, 163:113799, 2021.

[3] Huisu Jang and Jaewook Lee. Generative bayesian neural network model for risk-neutral pricing of american index options. *Quantitative Finance*, 19(4):587–603, 2019.

[4] Ji Hyun Jang, Jisang Yoon, Jungeun Kim, Jinmo Gu, and Ha Young Kim. Deepoption: A novel option pricing framework based on deep learning with fused distilled data from multiple parametric methods. *Information Fusion*, 70:43–59, 2021.

References

[1] BLACK-SCHOLES FORMULA AND GREEKS: PYTHON IMPLEMENTATION, https://quantchronicles.blogspot.com/2018/11/black-scholes-formula-and-greeks-full.html/

[2] Black Scholes Model in Python for Predicting Options Premiums, https://medium.com/swlh/calculating-option-premiums-using-the-black-scholes-model-in-python-e9ed227afbee/

[3] Option Greeks– Delta, Gamma, Vega, Theta and Rho, https://financetrainingcourse.com/education/2012/09/option-greeks-delta-gamma-vega-theta-rho-a-quick-reference-guide/

[4] Python Implementation of Binomial Stock Option Pricing, https://www.linkedin.com/pulse/python-implementation-binomial-stock-option-pricing-sheikh-pancham/