Artificial Intelligence in Fintech Project (2)[1]

[1] You can pick one project

Implied volatility Pricing (180 points)

1. Determine European options from Option2017_2_Clean.csv by using Brent or bisection method and write all European options as EuropeanOption2017.csv

   (a) I already did this step for you. If you need more accuracy, you can redo it by yourself

2. Apply the following methods to estimate the implied volatility for the put and call option data you get by using tolerance 10^{-12} in iteration and interest rate 0.03.

   (a) the classic Bisection method (*you need to code it,* but you can refer to the bisection method in scipy)

   (b) Brent method (you don't need to code it, you can only use the implementation from scipy)

   (c) Muller-Bisection method (*you need to code it*)

   (d) Newton method (*you* need to code it but you can refer to the Newton method in scipy)

   (e) New newton: use brent as the fill-in method to compute initial points

   (f) New Halley: use brent as the fill-in method to compute initial points for Halley

   (g) Halley's irrational formula (Note: you can only pick plus sign in your implementation)

   You need to use

$$vomma = \frac{\partial^2 f}{\partial \sigma^2} = vega \times \frac{d_1 d_2}{\sigma}$$

$$d_1 = \frac{\ln(S/K) + (r + \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_2 = d_1 - \sigma\sqrt{T}$$

Note: σ is actually unknown, which is just the x_n in your iteration scheme.

3. Evaluate your implied volatility calculation by using the following measure for each method:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\sigma_{imp,i} - \sigma^*_{imp,i})^2$$

Where $\sigma_{imp,i}$ is the implied volatility you calculate for the i^{th} option and $\sigma^*_{imp,i}$ is the true implied volatility.

- You need to compare MSE plots for each method
- You also need to calculate the average iteration number to converge for each method

4. Compare the efficiency of the methods in implied volatility pricing.

- Efficiency aims to answer the query: how efficient the method Θ can be when it is convergent? It is defined as the following ratio for a given method Θ,

$$\eta = \frac{1}{(1 + mse)\log_2(1 + E(t))} \tag{1}$$

where mse represents the mean square error (MSE) of the method for all convergent cases and $E(t)$ is its corresponding time expectation (average iteration number to converge).

- A few average iteration steps and small MSE values under convergence both contribute to a good efficiency. The closer the efficiency to 1, the more efficient the method. In fact, $\eta = 1$ means the method achieves 100% efficiency, the ideal state of pricing. That is, $mse = 0$ and the method only takes one iteration step averagely to reach the true implied volatility value.

5. Draw your conclusion about accuracy and time for the four methods in implied volatility prediction.

B) Build a deep-learning trading system (200 points)

- A good *automated trading system* (we always assume it is just HFT though there are some subtle difference) can fully take advantage of the temporary inefficiencies in market to get profit.
- Suppose we want to build an automated trading system for S&P500 index. The idea is really easy: if stock price varies more than a cutoff, say $p\%$, (25% means price change is the current price's 25%) we believe it is worthwhile to trade.

> Simply, we can say the trading system will do the following operators automatically,
>
> if price increases $p\%$ with respect to the current price, buy, if price decreases $p\%$, sell.

- However, what we really concern is if we can make some profit by doing trading in the following k days.
- The automated trading system needs to predict the general tendency of stock prices in the following k days
- We know data before time t, we want to know what may happen in the next k days : $t+1$,$t+2$,...,$t+k$

HFT Data

- Please use the provided HFT datasets provided to build and test your trading machines.
- Please pay attention their time orders are inverse

Predict stock market by using deep learning methods

- 1. Classification is a way to predict unknown information by using our current/prior knowledge.
- 2. It has two data sets: training data that contains known knowledge and testing data that is the data we need to predict
  - Example:
  - Three month stock price and correct action we need to take are our known knowledge (you know it because it happened!)
  - Each training data entry has a label: here we have three choices: buy (+1), sell (-1), hold (0).
  - The price can be a vector includes all kinds of prices or some measures derived from them.

Finish the following problems

You need at least use

- LSTM, CNN, DNN, Extra Trees
- You can use other DL models such as Transformer, Capsule-NET
- Your trading model should predict sell, hold, and buy trading actions
- You need to mark your training data to get the labels and visualize it
- You need to show your confusion matrix in prediction.
- You need to calculate the d-index along with all other classification measures.
- Draw you conclusion

A.1: A Simple Trading Model for your reference

We need to find measures to indicate the next k days' stock tendency. Generally speaking, a good measure will be really helpful

People use the following measures to do their modeling

- 1. daily average price

$$\bar{P_i} = \frac{C_i + H_i + L_i}{3}$$

C_i, H_i and L_i are the close, high and low prices on day i

- 2. Arithmetic return daily on the following k days

$$V_i = \frac{P^{-}_{i+j} - C_i}{C_i}$$

where $j = 1, 2, ...k$

- 3. Sum of returns: add all returns whose absolute value $>$ p%

$$T_i = \sum_v \{v \in V_i if \; |v| > p\%\}$$

Large positive values of T means “buy” because the daily prices $>$ the targeted variation value. On the other hand, large negative values of T mean “sell”

-
  ```
  if T value > =0.15:
  Action: buy
  if T value < 0.15 and T value > 0.15
  Action: hold
  if T value < -0.15
  Action: sell
  ```

What should you turn in?

- 1. A folder that contains
  - A ppt to show details of your analytics (at LEAST 100 pages)
  - your data
  - source files
  - corresponding related output.
  - A link to your presentation video
- 2. Submit the zipped file (.zip instead of ,rar) of your folder to Canvas