Introduction to Artificial Intelligence Writing 4

YiFan Li
ZeYuan Yang

November 16, 2019

1 Introduction

On today's financial market, increasing number of financial indexes are being created. Take the stock market as an example. Up till now, there have been over 200 generic rules that can suggest a buying or selling or selling decision based on the historical stock price time series [3]. Some of the most prevailing ones include moving average, relative strength indicator, stochastic oscillator, etc. However, these rules have at least three flaws. Firstly, all of these rules seems somewhat arbitrary, and are highly unstable. They could perform well on some stocks, but might fail on others. Secondly, it's impossible for a human trader to integrate all these 200 rules in his mind and make a trading decision. Thirdly, most of the indexes have some parameters as output, for example the time window for moving average, the choice of what number should be used for these parameters are still empirical.

Both stock market practitioner and academic researchers have devoted huge effort to determine good interpreters that can help make profitable decisions in the financial market. Additionally, with the rise of quantitative trading, which means trading stocks by computer program instead of a human trader, the need for algorithmically select good financial indexes have become more important.

To be more specific, the problem setting can be summarized as follows:

- Given a historical stock price time series in the past n periods, $P_t = [p_{t-n}, p_{t-n+1}, \cdots, p_{t-1}]$, where t represents the present,
- and a set of k trading rules suggesting "buy", "sell" or "hold" action based on the time series $R = \{r_1, r_2, \cdots, r_k\}, \forall r_i(P_t) \to s_i \in \{\text{sell}, \text{buy}, \text{hold}\}$,
- design an agent A who can integrate all the rules to make a final decision $A(R(P_t)) \to d_t \in \{\text{sell}, \text{buy}, \text{hold}\}$,
- that can maximize the profit in the following m period, which can be calculated as $\sum_{t=0}^{m} p_t^{\text{sell}} - p_t^{\text{buy}}$,

$$p_t^{\text{sell}} = \begin{cases} p_t & \text{if} A(R(P_t)) = \text{sell} \\ 0 & \text{otherwise} \end{cases}, p_t^{\text{buy}} = \begin{cases} p_t & \text{if} A(R(P_t)) = \text{buy} \\ 0 & \text{otherwise} \end{cases}$$

2 Approach

We are aiming to solve this problem with genetic algorithm. On one hand, the problem setting can be considered as a search problem, which tries to find the best combination of trading rules from all possibilities. The evaluation function can also be very easily defined. On the other hand, there are already some previous studies using the same approach [1] [2]. We can based our project on their findings and dig deeper.

3 Implementation Plan

For the experiment, we will have to gather data and implement the algorithm.

3.1 Data

We can find the historical stock data for free on Alpha Vantage. It provides a lot of APIs and we can get the data via HTTP requests. The data pipeline, however, will have to be built by ourselves. We will use Python to grab data and save it as csv files for further use.

We plan to gather 3 years' data and use the first two years as the train set, and the last one year as test set. With this architecture, our agent does not only find the optimal strategy for the history, but also have some generalization ability and therefore have the potential to be applied on the market in practice.

3.2 Algorithm

References

[1] F. A. Badawy, H. Y. Abdelazim, and M. G. Darwish. Genetic algorithms for predicting the egyptian stock market. In *2005 International Conference on Information and Communication Technology*, pages 109–122, Dec 2005.

[2] D. Fuente, A. Garrido, J. Laviada, and A. Gomez. Genetic algorithms to optimise the time to make stock market investment. pages 1857–1858, 01 2006.

[3] J. Korczak and P. Roger. Stock timing using genetic algorithms. *Applied Stochastic Models in Business and Industry*, 18(2):121–134, 2002.