CS229 Project Milestone

Using reinforcement learning for algorithmic trading

Thanapat Worasaran
tworasar@stanford.edu

New project proposal

Algorithmic trading is a field that has been constantly researched on. As the name suggests, it uses algorithms to predict stock market price, and make an investment decision accordingly. There are two approaches on using artificial intelligence for algorithmic trading: supervised, and reinforcement learning [1]. Intuitively, supervised learning is a less viable approach to the algorithmic trading, since it lacks policy, i.e. the rules of which action to take in response to the external drive. Several past CS229 projects have used supervised learning algorithms to predict stock prices, but none has optimized the net profit on series of the price data. I would like to create a trading bot using reinforcement learning and supervised learning.

Problem statement

Given the historical data of a stock, a bond, or a cryptocurrency (i.e. prices and volumes), which action will maximize the profit. The set of actions are buy, sell, and hold (if possible, state the amount of money in the case of buy and sell). The profit can be calculated using the realized PnL (profit and loss). The reward is received when the position is closed, meaning the asset that is previously bought is sold off.

Data

I use Stock Market Dataset from Kaggle [2], which contains comprehensive daily stock data for all stock available in the market since around 1950.

Baselines

Two simple models are used for the baseline:

1. Buy, hold, and sale
Put all money into a stock at the start, and sell it at the end.

2. Moving Average Convergence Divergence (MACD)
MACD is commonly used by professional traders as buy and sell signals. It is calculated by subtracting the 26-day exponential moving average (EMA) out of the 12-day EMA. The signal line can be calculated using the 9-day EMA of the MACD line. When the MACD crosses above the signal line, it is the signal to buy, conversely when it crosses the signal line below, it is the signal to sell [3]. The baseline rule is as follow: starting from some initial amount of money, buy as much as possible at the buy signals and sell as much as possible at the sell signals.

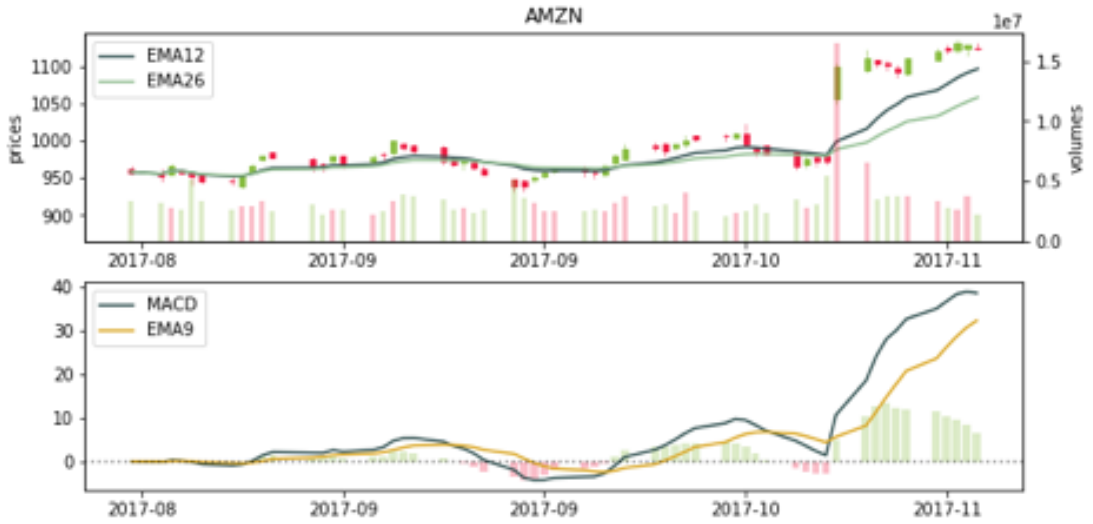

Fig 1 Above: The candlestick graph of amazon stock price and volume from August 2017 to November 2017 superimposed by exponential weighted average of 12 and 16 day. Below: the MACD and signal line. The different of MACD and signal line is indicated by the bar plot below

Three stocks, each representing most distinctive pricing trends: Amazon, Tesla, Groupon are sampled for the baseline model. The profit from (virtually) trading these three stocks in the last 600 days in the dataset using the two baseline models is shown below.

Stock	% change, buy hold and sell	%change, MACD signal
Amazon	251.19%	20.9%
Tesla	11.3%	66.5%
Groupon	3.46%	-13.1%

As we can see in the baseline here, the effectiveness of the MACD depends on the nature of the stocks. It does well in Tesla which has a clear cyclical behavior, but not as well in the other two.

Next steps

The price prediction, although can be used to help human decision, is inadequate for algorithmic trading. The main problem is it does not have policy i.e. even though it knows the future price, it does not know the optimal action i.e. it does not know whether to buy, sell or hold. Hence, reinforcement learning is highly desirable in this situation.

Machine learning packages

Keras and Pybrain are easy to use and flexible which make them very suitable for making prototypes

Time series forecasting model

Even though the reinforcement learning is a necessity, it still need a neural network on the inside of the model to be able to predict the price and act accordingly. From [4], using autoregressive integrated moving average (ARIMA) with artificial neural network is a feasible method to forecast the stock price.

Reinforcement Learning

A standard reinforcement learning process is Markov Decision Process (MDP). There were some works that use MDP for trading algorithm [5]. The starting point is to implement MDP for trading: defining states, actions, and reward. Then choosing gradient update algorithm, for example monte carlo policy gradient. Finally, the trading algorithm will be tested with different types of stocks.

Reference

[1] http://www.wildml.com/2018/02/introduction-to-learning-to-trade-with-reinforcement-learning/
[2] https://www.kaggle.com/borismarjanovic/price-volume-data-for-all-us-stocks-etfs
[3] https://www.investopedia.com/terms/m/macd.asp
[4] G. Peter Zhang, **Time series forecasting using a hybrid ARIMA and neural network model**, Neurocomputing 50 (2003) 159–175
[5] Pierpaolo G. Necchi, Thesis: Reinforcement Learning For Automated Trading, Mathematical Engineering Politecnico di Milano