MATH4900 Report

Chan Chun Pang

November 26, 2023

1 Strategy to Test

There are many technical analyses in stock trading. Most of the common trading rules are based on some indicators, for example moving average (MA), resistance and support level (RS-level), RSI etc. In reality, most traders will use a mixture of these analyses to decide whether to buy or sell. Here I want to investigate how we should combine those strategies to improve the performance in trading.

1.1 Q-Learning

A model-free machine learning algorithm is implemented since it is hard to obtain the price distribution. Here, we implemented the so-called Q-learning algorithm.

The standard setup of Q-learning is as follows: suppose we have

1. a finite set of states S
2. a finite set of actions A. Note that a action $a \in A$ is the mapping $a : S \to S$
3. a reward function $r : S \times A \to \mathbb{R}$

We want to construct $Q_{opt} : S \times A \to \mathbb{R}$, indicating the expected reward by applying action a at state s. Therefore an agent can based on the function Q_{opt} choose the best action that leads to a larger Q_{opt}: the deterministic policy $\pi_Q : S \to A$ (chosen action) is

$$\pi_Q(s) = \arg\max_a Q_{opt}(s, a), \quad \forall s \in S \tag{1.1}$$

1.2 General Procedure on Machine Training

In general, it is difficult to directly find Q_{opt}. However, there are some recursive rules allowing us to approximate Q_{opt}. In our case, the situation is slightly different. r and all $a \in A$ is not deterministic. For instance, $r(s, a)$ can be different each time even if you plug in the same value of s and a. Hence we have to use a slightly different recursive rule to approximate Q_{opt}:

$$Q_{n+1}(s, a) = (1 - \alpha_n)Q_n(s, a) + \alpha_n(r(s, a) + \gamma \max_{a'} Q_n(a(s), a')) \tag{1.2}$$

$$\alpha_n = \frac{1}{1 + visits(s, a)} \tag{1.3}$$

$$Q_0(s, a) = c, \quad \forall s \in S, a \in A \tag{1.4}$$

Here $visits(s, a)$ is the number of visits (number of trials) during the training when applying a on s, $c \in \mathbb{R}, \gamma \in [0, 1)$ are arbitrary numbers. The γ is called the discount rate of the estimated future reward $\max_{a'} Q_n(a(s), a')$.

This algorithm is guaranteed to converge i.e. $Q_n \to Q_{opt}$. Therefore we can stop training whenever the error is less than tolerance ϵ. We calculate the error as follows: let $\Delta Q(s,a)$ be the different of the current Q estimation and the pervious Q estimation. Then the error is

$$\text{error} = \langle |\Delta Q| \rangle \tag{1.5}$$

To reduce the training time, a better training policy is needed such that we spend more time on exploiting locally optimized actions, and remaining fair enough time to explore other actions to seek global optimum. Here we use the so-called Boltzmann distribution policy:

$$\pi_{training}(a'|s) = \Pr\big(\text{choosing action } a'|s\big) = \frac{\exp(Q_n(s,a')/\tau)}{\sum_a \exp(Q_n(s,a)/\tau)}, \quad \forall s \in S, a \in A \tag{1.6}$$

Here $\tau > 0$ is a number that stands for temperature, controlling how greedy the policy is (the lower value of τ is, the policy behaves more likely to the greedy policy). Moreover, the action with higher Q_n will have a higher chance of being selected, while there is still a chance that the action with lower Q_n to be selected.

1.3 Specification of the Model

Let $\mathbb{N} = \{1,2,3,\dots\}$, $I_n = \{1,2,\dots,n\} = \mathbb{N} \cap [1,n]$. Given the daily price (closing price) data $X : I_n \subset \mathbb{N} \to \mathbb{R}$, (set $X(1)$ be the first data), we define the following functions based on the common trading strategies:

1.3.1 MA State

The common trading strategy relating to MA is called moving average crossover, which stated as follows:

1. Let $d \in \mathbb{N}$. Then define MA_d to be

$$MA_d(t;X) = \begin{cases} \dfrac{1}{d}\displaystyle\sum_{t-d+1}^{t} X(t), & \forall t \geq d \\ \text{undefined}, & \forall t < d \end{cases} \tag{1.7}$$

2. Let $d_1, d_2 \in \mathbb{N}$ with $d_1 < d_2$ (here I take $d_1 = 5, d_2 = 150$, which is one of the common pairs of moving average crossover strategy). Then

$$\begin{cases} \text{buy}, & \text{when } MA_{d_1}(t-1;X) < MA_{d_2}(t-1;X) \text{ and } MA_{d_1}(t) \geq MA_{d_2}(t;X) \\ \text{sell}, & \text{when } MA_{d_1}(t-1;X) > MA_{d_2}(t-1;X) \text{ and } MA_{d_1}(t) \leq MA_{d_2}(t;X) \end{cases} \tag{1.8}$$

To capture this strategy, we can define the MA state as

$$S_{MA}(t;X) = \begin{cases} 1, & \text{if } MA_{d_1}(t-1;X) < MA_{d_2}(t-1;X) \text{ and } MA_{d_1}(t;X) \geq MA_{d_2}(t;X) \\ 2, & \text{if } MA_{d_1}(t-1;X) > MA_{d_2}(t-1;X) \text{ and } MA_{d_1}(t;X) \leq MA_{d_2}(t;X) \\ 3, & \text{if } MA_{d_1}(t-1;X) \geq MA_{d_2}(t-1;X) \text{ and } MA_{d_1}(t;X) \geq MA_{d_2}(t;X) \\ 4, & \text{if } MA_{d_1}(t-1;X) \leq MA_{d_2}(t-1;X) \text{ and } MA_{d_1}(t;X) < MA_{d_2}(t;X) \end{cases} \tag{1.9}$$

1.3.2 RS-Level State

The common trading strategy relating to RS-level is called trading range break, which stated as follows:

1. In general a resistance level is defined to be the last local maximum and a support level is defined to be the last local minimum. Since the price data is noisy, to accurate find the local extremum, we compare all $d = 5$ data on each side.

2. Let RL and the SL be the resistance level and support level based on the price X. Then we

$$\begin{cases} \text{buy}, & \text{when } X(t) > RL(t;X) \\ \text{sell}, & \text{when } X(t) < SL(t;X) \end{cases} \tag{1.10}$$

To capture this strategy, we can define the RS-Level state as

$$S_{RS}(t;X) = \begin{cases} 1, & \text{if } X(t) > RL(t;X) \\ 2, & \text{if } X(t) < SL(t;X) \\ 3, & \text{if } SL(t;X) \leq X(t) \leq RL(t;X) \end{cases} \tag{1.11}$$

1.3.3 RSI State

The common trading strategy relating to RSI is as follows:

1. Let $d \in \mathbb{N}$ (here I take $d = 14$). Then define RS_d to be

$$RS_d(t;X) = \begin{cases} \dfrac{\text{average gain during time interval } d}{\text{average loss during time interval } d}, & \forall t \geq d \\ \text{undefined}, & \forall t < d \end{cases} \tag{1.12}$$

   Then define $RSI_d : \mathbb{N} \to \mathbb{R}$ to be

$$RSI_d(t;X) = 100 - \frac{100}{1 + RS_d(t;X)} \tag{1.13}$$

2. Let $Z_1, Z_2 \in [0, 100]$ with $Z_1 < Z_2$ (typically $Z_1 = 20, Z_2 = 80$). Then we

$$\begin{cases} \text{buy}, & \text{when } RSI_d(t;X) < Z_1 \\ \text{sell}, & \text{when } RSI_d(t;X) > Z_2 \end{cases} \tag{1.14}$$

To capture this strategy, we can define the RSI state as

$$S_{RSI}(t;X) = \begin{cases} 1, & \text{if } RSI_d(t;X) < Z_1 \\ 2, & \text{if } RSI_d(t;X) > Z_2 \\ 3, & \text{if } Z_1 \leq RSI_d(t;X) \leq Z_2 \end{cases} \tag{1.15}$$

1.3.4 Stock-Holding State

To simplify the transaction, we only allow to hold one share of stock. Hence we need to store this information. This state also affects what action we can take. For instance,

- if the agent is not holding a share of stock, then it can either do nothing or buy one
- if the agent is holding a share of stock, then it can either do nothing or sell it

We define the stock-holding state as

$$S_{stock\_holding} = \begin{cases} 1, & \text{if we are not holding the stock} \\ 2, & \text{if we are holding the stock} \end{cases} \tag{1.16}$$

1.3.5 Reward Function and State Transaction

In conclusion, a state in my model is a vector $[s_{MA}, s_{RS}, s_{RSI}, s_{stock\_holding}]$. The first 3 elements can be obtained from the price data X, and the last one is the internal state of the agent. Now we define the rule of the reward function and state transaction. Firstly, since we need lots of data for training, we need to have price data for more than one stock. Define $X_i : \mathbb{N} \to \mathbb{R}$ be the price with the stock index i. Here are the rules of reward function and state transaction: let $c(X(t))$ be the transaction cost at time t for price data X. Then

1. For the state $[s_{MA}, s_{RS}, s_{RSI}, 1], s_{MA} \in I_4, s_{RS}, s_{RSI} \in I_3$, the possible actions are $\{\text{do nothing}, \text{buy}\}$.
   (a) If the agent decides to do nothing, then we randomly pick a stock index i_0, and a time t_0. The next state will be
   $$[S_{MA}(t_0; X_{i_0}), S_{RS}(t_0; X_{i_0}), S_{RSI}(t_0; X_{i_0}), 1] \tag{1.17}$$
   The reward function will be 0 since you do nothing.
   (b) If the agent decides to buy, it means the agent wants to enter the market and track the stock. Suppose the values of the current state s_{MA}, s_{RS}, s_{RSI} are generated from a price data $X_{i_0}(t_0)$. The next state will be
   $$[S_{MA}(t_0 + 1; X_{i_0}), S_{RS}(t_0 + 1; X_{i_0}), S_{RSI}(t_0 + 1; X_{i_0}), 2] \tag{1.18}$$
   The reward function will be $[X_{i_0}(t_0 + 1) - X_{i_0}(t_0) - c(X_{i_0}(t_0))]/X_{i_0}(t_0)$.
2. For the state $[s_{MA}, s_{RS}, s_{RSI}, 2], s_{MA} \in I_4, s_{RS}, s_{RSI} \in I_3$, the possible actions are $\{\text{do nothing}, \text{sell}\}$.
   (a) If the agent decides to do nothing, suppose the values of the current state s_{MA}, s_{RS}, s_{RSI} are generated from a price data $X_{i_0}(t_0)$. The next state will be
   $$[S_{MA}(t_0 + 1; X_{i_0}), S_{RS}(t_0 + 1; X_{i_0}), S_{RSI}(t_0 + 1; X_{i_0}), 2] \tag{1.19}$$
   The reward function will be $[X_{i_0}(t_0 + 1) - X_{i_0}(t_0)]/X_{i_0}(t_0)$.
   (b) If the agent decides to sell, it means the agent wants to leave the market and try other stock. We randomly pick a stock index i_1, and a time t_1. The next state will be
   $$[S_{MA}(t_1; X_{i_1}), S_{RS}(t_1; X_{i_1}), S_{RSI}(t_1; X_{i_1}), 1] \tag{1.20}$$
   Suppose the values of the current state s_{MA}, s_{RS}, s_{RSI} are generated from a price data $X_{i_0}(t_0)$. Then the reward function will be $-c(X_{i_0}(t_0))/X_{i_0}(t_0)$.

Here I adopt a common way to calculate the transaction cost which is

$$c(X) = 0.001 * X \tag{1.21}$$

2 Data Preparation

Since machine learning requires lots of data, we can obtain the stock price using web scraping. Here I use the stock screener from NASDAQ to obtain all current stock tickers. Then I download the stock data between 2013-10-01 to 2023-10-01 from Yahoo Finance by looping over the list of tickers.

To test the predictive power, the dataset will be split into two parts by time. The dataset where the date is before 2021-11-11 will be the training dataset and another will be the testing dataset.

Additionally the pre-calculation of S_{MA}, S_{RS}, S_{RSI} should be done before training to reduce computational cost during training.

3 Hypothesis Tests and Results

To test the performance of the q-learning strategy, we construct some benchmark cases by re-constructing those simple strategies related to moving average, resistance and support level and RSI. Let π_{MA} be the policy of moving average crossover strategy, π_{RS} be the policy of trading range break strategy and π_{RSI} be the policy of RSI-strategy. They are defined as follows

$$\pi_{MA}(\vec{s}) = \begin{cases} \text{buy}, & \text{if } s_{MA} = 1 \text{ and } s_{stock\_holding} = 1 \\ \text{sell}, & \text{if } s_{MA} = 2 \text{ and } s_{stock\_holding} = 2 \\ \text{do nothing}, & \text{otherwise} \end{cases} \tag{3.1}$$

$$\pi_{RS}(\vec{s}) = \begin{cases} \text{buy}, & \text{if } s_{RS} = 1 \text{ and } s_{stock\_holding} = 1 \\ \text{sell}, & \text{if } s_{RS} = 2 \text{ and } s_{stock\_holding} = 2 \\ \text{do nothing}, & \text{otherwise} \end{cases} \tag{3.2}$$

$$\pi_{RSI}(\vec{s}) = \begin{cases} \text{buy}, & \text{if } s_{RSI} = 1 \text{ and } s_{stock\_holding} = 1 \\ \text{sell}, & \text{if } s_{RSI} = 2 \text{ and } s_{stock\_holding} = 2 \\ \text{do nothing}, & \text{otherwise} \end{cases} \tag{3.3}$$

To test the profitability, a t-test can be implemented for comparing the unconditional daily return to the daily return with buy / sell signal, similar to William Brock et al. Consider the stock index $i \in I$, all the stock index we tested, $X_i(t)$ be the test price at time t, where $t \in T_i$, the test period. The daily return r_i is defined by

$$r_i(t) = \log X_i(t+1) - \log X_i(t) \tag{3.4}$$

Let $S_{i,\pi}$ be the trading signal generated by policy π:

$$S_{i,\pi}(t) = \begin{cases} 1, & \text{if buy signal is triggered} \\ -1, & \text{if sell signal is triggered} \\ 0, & \text{otherwise} \end{cases} \tag{3.5}$$

Then the unconditional daily returns will be the set $\{r_i(t)\}$, the daily returns with buy signal will be $\{r_i(t) : S_{i,\pi}(t) = 1\}$ and the daily returns with sell signal will be $\{r_i(t) : S_{i,\pi}(t) = -1\}$. Now we can compare the unconditional daily returns and the daily returns with buy/sell signal by t-test for each policy. Overall, we have 1065222 unconditional data.

1. The first hypothesis test is testing whether we enter the market at correct timing. The null hypothesis and the alterative hypothesis can be formulated as follows: let $\mu_{b,\pi}, \mu$ be the mean daily returns with buy signal under policy π, and the unconditional daily returns. Then the null hypothesis and alterative hypothesis are

   $$H_0 : \mu_{b,\pi} = \mu, \quad H_1 : \mu_{b,\pi} > \mu \tag{3.6}$$

   The table of t-statistics are shown in table 1. Under 95% significant level, we conclude that the null hypothesis is not rejected for all strategies, meaning that we cannot conclude that all the strategies are profitable.

2. The second hypothesis test is testing whether we leave the market at correct timing. The null hypothesis and the alterative hypothesis can be formulated as follows: let $\mu_{s,\pi}$ be the mean daily returns with buy signal under policy π. Then the null hypothesis and alterative hypothesis are

   $$H_0 : \mu_{s,\pi} = \mu, \quad H_1 : \mu_{s,\pi} < \mu \tag{3.7}$$

   The table of t-statistics are shown in table 2. Under 95% significant level, we conclude that the null hypothesis is rejected for RSI strategy and Q-Learning strategy, meaning that both RSI strategy and Q-Learning strategy can leave the market to prevent loss.

Strategy	t-statistics	Number of buys	p-value
Simple Moving Average Crossover	-3.3314	9901	0.99957
Trading Range Break	-3.7774	16952	0.99992
RSI Strategy	0.56895	7016	0.28470
Q-Learning Strategy	2.68855	41222	3.5896e-3

Table 1: t-statistics for buy-unconditional sample among all the strategies. The p-value is calculated using the hypotheses given in (3.6).

Strategy	t-statistics	Number of Sells	p-value
Simple Moving Average Crossover	4.4475	8945	0.99996
Trading Range Break	-0.83956	15869	0.20058
RSI Strategy	-3.3934	5054	3.4778e-4
Q-Learning Strategy	-3.2847	39501	5.1082e-4

Table 2: t-statistics for sell-unconditional sample among all the strategies. The p-value is calculated using the hypotheses given in (3.7)

3. To compare the performance gain between Q-learning and other simple strategies, we can compare both buys and sells. Let π be the strategies we want to compare. The null hypothesis and alterative hypothesis for buys can be formulated as

$$H_0 : \mu_{b,\pi_Q} = \mu_{b,\pi}, \quad H_1 : \mu_{b,\pi_Q} > \mu_{b,\pi} \tag{3.8}$$

The table of t-statistics are shown in table 3. Under 95% significant level, we conclude that the null hypothesis is rejected for Simple Moving Average Crossover and Trading Range Break, meaning that Q-Learning strategy performs significantly better on buys than these two strategies. Similar argument can be applied to sells. The null hypothesis and the alterative hypothesis for sells can be formulated as

$$H_0 : \mu_{s,\pi_Q} = \mu_{s,\pi}, \quad H_1 : \mu_{s,\pi_Q} < \mu_{s,\pi} \tag{3.9}$$

The table of t-statistics are shown in table 3. Under 95% significant level, we conclude that the null hypothesis is rejected for Simple Moving Average Crossover, meaning that Q-Learning strategy performs significantly better on sells than that strategy.

Overall, Q-learning strategy shows significant improvement in performance when comparing to simple moving average crossover, significant improvement only on buys but not sells when comparing to trading range break. Q-learning strategy does not show significant improvement when comparing to RSI strategy. However, all these strategies does not show profitable on buys in test dataset. Only RSI strategy and Q-learning strategy generate returns which are lower than normal returns.

Strategy to compare	t-statistics	degree of freedom	p-value
Simple Moving Average Crossover	3.418	11370	0.00031579
Trading Range Break	3.8454	19988	6.0374e-5
RSI Strategy	-0.44172	7403.0	0.67065

Table 3: t-statistics for buys when compared Q-learning strategy to other simple trading strategies

Strategy to compare	t-statistics	degree of freedom	p-value
Simple Moving Average Crossover	-5.1821	10860	1.12e-7
Trading Range Break	-0.22869	20103	0.40956
RSI Strategy	2.9665	5231.4	0.99849

Table 4: t-statistics for sells when compared Q-learning strategy to other simple trading strategies