# MAFS 6010Y Assignment1

**Group Member:** **WANG Zihan(20801541)** **XIE Wenyu(12247636)**

## I. Problem Formulation

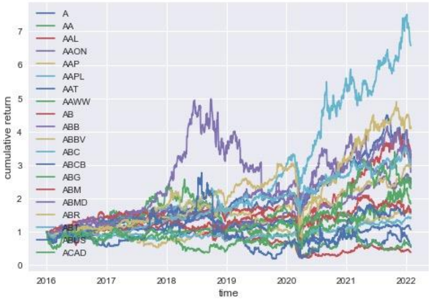
In this assignment we implement $\varepsilon$-greedy, Boltzmann and upper confidence bound algorithms to decide optimal stocks to invest on the daily basis. For the multi-armed bandit problem, each arm represents an individual stock and each round represents a trading day. Rewards are set to be the daily returns and action refers to pick one stock to invest in each day. Learning rules are the standard averaging rule under the stationary environment. The final goal is to maximize the total cumulative profit.

## II. Data Description

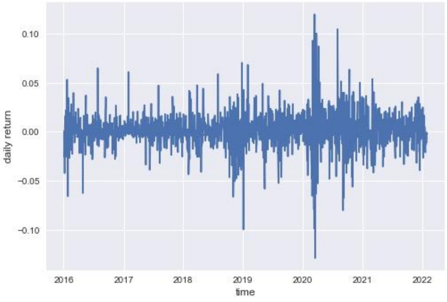
We use daily adjusted close price data downloaded from Yahoo Finance. The date range is from 2016-01-01 to 2022-01-28. These stocks are traded in NASDAQ or NYSE. We filter out stocks which have 40% total missing observations or 20% recent-one-year missing observations. Some of stocks that we choose are A, AA, AAL, AAON, AAP, AAPL, etc.

In our algorithm, the sample mean of the first one-month daily returns are used for the initialized reward estimates. The backtest period is from 2016-02-01 to 2022-01-28, with each trading day as a round. The initial cash is 1 dollar. 20 stocks are picked in total (I.e., 20 arms).

AAPL stands out with total return of 657%.



Graph1 20 stocks cumulative PnL

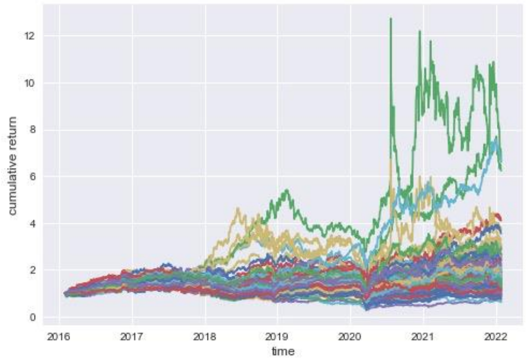


Graph2 AAPL daily return

## III. Experiment Results

### *ε-greedy*

We set the exploration rate to be 0.1 or 10% probability to explore.



Graph 3 $\varepsilon$-greedy, $\boldsymbol{\varepsilon = 0.1}$, path=100

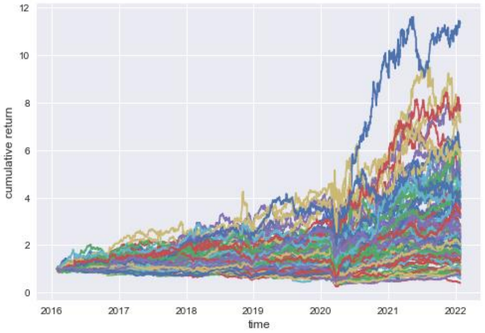
### *Boltzmann*

Sigma controls the probability of exploitation. With higher sigma, there are higher chances to do exploration. On each day best arm is chosen based on the Boltzmann distribution.

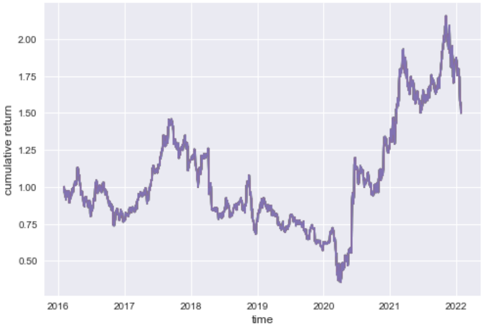
$$p_{(A_t=a)} = \frac{\exp\left(\frac{Q_t(a)}{\sigma}\right)}{\sum_{b=1}^{arms}\left(\frac{Q_t(b)}{\sigma}\right)}$$

### Upper confidence bound

This algorithm picks best arm by choosing highest upper confidence bound, controlled by the current estimation of reward and bound range. Arms that are selected more before have lower bound range so they are less likely to be chosen later on, fully utilizing the exploitation information. Parameter c controls the bound range.



Graph 4 Boltzmann, sigma=0.1



Graph5 UCB, c=2

### Summary

| | Mean regret | Std regret | Paths |
|---|---|---|---|
| ε-greedy | 4.72 | 1.09 | 100 |
| Boltzmann | 3.84 | 1.89 | 100 |
| UCB | 5.25 | Nil | 1 |

Table1 Regret Result

For ε-greedy and Boltzmann, because choosing whether to explore or not and which arm to explore are random, the strategy path varies drastically, which makes it unrealistic to use in the investment. Although UCB generates constant results since we do the backtest instead of simulating stock return, the highest regret and large drawdown are bad signals for investing either.

## IV. Analysis

The backtesting results of all the bandit algorithms works bad based on the result above. And we come up with a few possible explanations towards them.

### *I.I.D Assumptions*

The assumption that rewards for each action are identically and independently distributed is not realistic. Stocks returns usually hold trends and volatility clustering properties. Graph 2 above shows the daily return plot of AAPL stocks, and we can see the volatility clustering. Fixed mean and variance in a short period cannot fully characterize the distribution. Besides, the bandit ignores the fact that returns are serially correlated, which is useful information for exploration.

### *Bandit Feedback*

For a traditional multi-armed bandit problem, you can only get the reward information after you pull the lever. However, we know the past return data for all stocks in the selection pool, even ones without investing. This is called full feedback when the algorithm observes the rewards for all arms that could have been chosen; instead of bandit feedback, when the algorithm observes the reward for the chosen arm and no other feedback. For example, stocks are also cross-sectionally correlated with each other and such factors are quite necessary for investing. We can estimate easily from previous data for the correlation table and it’s usually stable, while bandit just ignores those kinds of information.

### *Unstable Backtesing Paths*

Considering the stock investing action in practice, it is hard to implement an algorithm which holds large variation for different backtesting paths. $\varepsilon$-greedy and Boltzmann methods are the typical ones since both of them contain random property- epsilon and sigma to control the exploration probability. So maybe upper confidence bound method is the best algorithm among the three in light of the stability.

## V. Project Code and Data Repository:

https://github.com/wangzh0912/RL-Projects/tree/master/HW1-MAB

**Contribution: 50% for each person**