Multi-Agent Reinforcement Learning for Stock trading

Objective:

The purpose of this project is to implement a Single agent stock market trading strategy using Reinforcement Learning and extend it to Multi-Agent settings by replicating the dynamics of real world to the possible extent. Studying the behavior of emergent behavior of these agents in the Multi Agent environment.

Introduction:

FinRL is an open-source library that provides practitioners a unified framework for pipeline strategy development. In reinforcement learning (or Deep RL), an agent learns by continuously interacting with an environment, in a trial-and-error manner, making sequential decisions under uncertainty and achieving a balance between exploration and exploitation.

Liquidation is the process of selling a large number of shares of one stock sequentially within a given time frame, taking into consideration the costs arising from market impact and a trader’s risk aversion. we propose to use multiagent deep reinforcement learning model.



Challenges:

* Liquidation of large number of stock shares would have huge impact on the market making the environment difficult to predict.
* Current methods for static environment ignore the dynamics and interactive nature of stock market.
* Inability to collect enough historical events data to obtain practical trading insights.

Implementation:

Single Agent –

Multi Agent

We model the liquidation process as a Markov decision process (MDP), and then formulate the multi-agent setting we used to resolve the liquidation problem. Our goal is to minimize the expected implementation shortfall. We used DDPG based Multi Agent Training and trained the model for different shares ratio an risk aversion levels.

Environment settings:

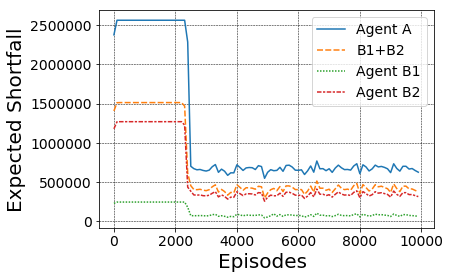
Number of agents: 2

Total number of stocks: 1,000,000

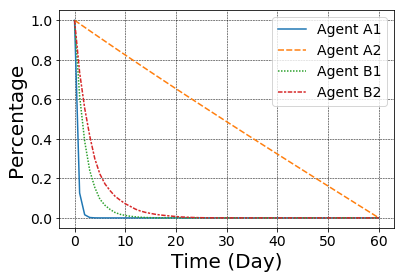
Time frame: 60days

Number of trades: 60

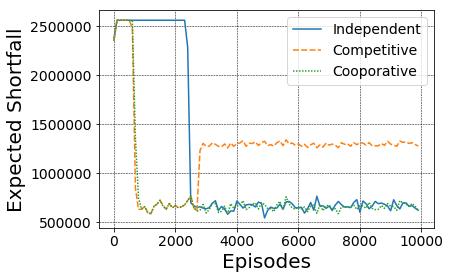
Analysis:



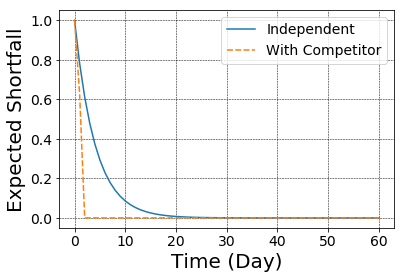
Comparison of expected implementation shortfalls: there are three agents $A, B1$ and $B2$. The expected shortfall of agent A is higher than the sum of two expected shortfalls $B\_1$ and $B\_2$



comparing to their original trading trajectories, their current trading trajectories are closer to each other when they are trained in a multi-agent environment.



Cooperative and competitive relationships: if two agents are in cooperative relationship, the total expected shortfall is not better than training with independent reward functions. If two agents are in a competitive relationship, they would first learn to minimize expected shortfall, and then malignant competition leads to significant implementation shortfall increment.



comparing to independent training, introducing a competitor makes the host agent learn to adapt to new environment and sell all shares of stock in the first two days.

Future Scope:

* Adding more parameters to State space to match the complexities of real world dynamics
* Analysis of emergent behavior of Multiple agents in complex environment.
* Validation of these behaviors with respect to real world stock market.

References: