Reinforcement Learning Project

Raphael Monges

Topic: Stock Trading - We choose a commodity with volatility and seasonality: Gaz.

Motivation :
As being a student who specialized in financial markets and who is doing his internship in it, I am motivated to develop tools with reinforcement learning to improve algorithmic trading strategies through reinforcement learning and if time available, reinforcement learning with human feedbacks.

Description of the environment:
State space:

The state space in a reinforcement learning framework for stock trading can be mathematically defined as a vector of features $st \in S$ representing the market and internal state at time t.

This includes:
1. Price vectors P(t,i) that represents close price at time t of stock i.
2. Technical indicators I(t,i) that represents an indicator I at time t for stock i.
3. Other features At, such as CPI, interest rates from central bank, positions, at time t

Moreover, the state at time t can be represented as : st = [Pt, It, At]

Action space:
All the possible actions at ∈ A that the agent can take at each timestep are =
1. Buy x shares of stock i such as a(t,i) = x >0
2. Sell x shares of stock i such as a(t,i) = x < 0
3. Hold: s(t,i) = 0

(We do not take into account hedging here with options for example)

Reward space:
The reward function R(st, a(t,i)) provides feedback to the agent based on the outcome of its actions guiding the learning process.

R(st, a(t,i)) = VariationPV(t) – lambda where PV(t) is the change in portfolio value resulting from action a(t) at state s(t) and lambda is a risk aversion coefficient representing volatility or drawdown.

Description of the implemented agent:
As RL algorithm, at first we'll develop a Q-learning and we are looking forward to use in a continuous space the Deep Q-Networks. We want to learn a policy π* that max expected returns. Then we would have:
Q-learning : Q*(s,a) = E[R(s,a) + gamma*max(Q*(s', a')]
DQN: an approximation of Q*(s,a) using a neural network

If time available we would be able to make the agent learn from Human Feedback, it implies modifying the reward function adding a term:
R'(st, at) = R(st, at) + n*H(st, at)

Conclusion:
Our goal is to optimize the policy π* such as it maximizes the expected cumulative reward, takin, into account market returns and market risk. The challenge will be to include a stochastic model such as [P (st+1|st, at)] that generalizes well to unseen market conditions.