TRƯỜNG ĐẠI HỌC FPT

Phát Triển Hệ Thống Giao Dịch Cổ Phiếu Sử Dụng Mô Hình Proximal Policy Optimization

Lecture: Lê Đình Huynh
Class: AI1708-ADS
Subject: Reinforcement Learning(REL301m)

Đặt vấn đề

Trong bối cảnh tài chính hiện đại, đầu tư vào cổ phiếu không chỉ là một phương thức hiệu quả để gia tăng tài sản mà còn là một thử thách lớn đối với các nhà đầu tư. Thị trường chứng khoán luôn biến động phức tạp, khiến việc đưa ra các quyết định giao dịch chính xác và kịp thời trở thành yếu tố then chốt quyết định sự thành bại của chiến lược đầu tư. Các phương pháp truyền thống ngày càng bộc lộ hạn chế khi phải xử lý khối lượng thông tin khổng lồ và những biến động khó lường của thị trường.

Trước thực tế đó, sự phát triển mạnh mẽ của công nghệ và các thuật toán học tăng cường (Reinforcement Learning) đã mở ra những hướng đi mới, cho phép tự động hóa và tối ưu hóa quá trình ra quyết định giao dịch. Học tăng cường, một nhánh quan trọng của trí tuệ nhân tạo, mang đến cách tiếp cận mới, giúp xây dựng các chiến lược giao dịch dựa trên dữ liệu thị trường và những quy tắc được lập trình sẵn, đồng thời có khả năng thích nghi với sự thay đổi liên tục của thị trường.

Bài toán cụ thể đặt ra là thiết kế một thuật toán học tăng cường với mục tiêu tối đa hóa lợi nhuận từ việc đầu tư cổ phiếu. Cụ thể, từ số vốn khởi điểm 100,000 USD, chiến lược cần được tối ưu để đạt được tài sản 1 triệu USD. Để hiện thực hóa điều này, việc xác định các quy tắc giao dịch rõ ràng là vô cùng cần thiết, bao gồm: các ngưỡng mua/bán dựa trên phân tích kỹ thuật, tín hiệu thị trường và chiến lược quản lý vốn hợp lý. Ngoài ra, các yếu tố chi phí thực tế như phí giao dịch, thuế, và chi phí vận hành cũng cần được tích hợp vào mô hình, nhằm đảm bảo rằng chiến lược không chỉ hiệu quả về lý thuyết mà còn khả thi trong môi trường thực tế..

I. Mô tả và Trực Quan hóa dữ liệu

Dữ Liệu Cổ Phiếu GOOGL:

Dữ liệu sử dụng trong báo cáo này bao gồm thông tin về cổ phiếu của Alphabet Inc. (GOOGL) từ ngày 20 tháng 3 năm 2017 đến ngày 29 tháng 12 năm 2023. Dữ liệu được thu thập từ stockscan và lưu trữ dưới định dạng CSV. Cột chính trong dữ liệu bao gồm:

- Date: Ngày giao dịch.
- Open: Giá mở cửa.
- High: Giá cao nhất trong ngày.
- Low: Giá thấp nhất trong ngày.
- Close: Giá đóng cửa.

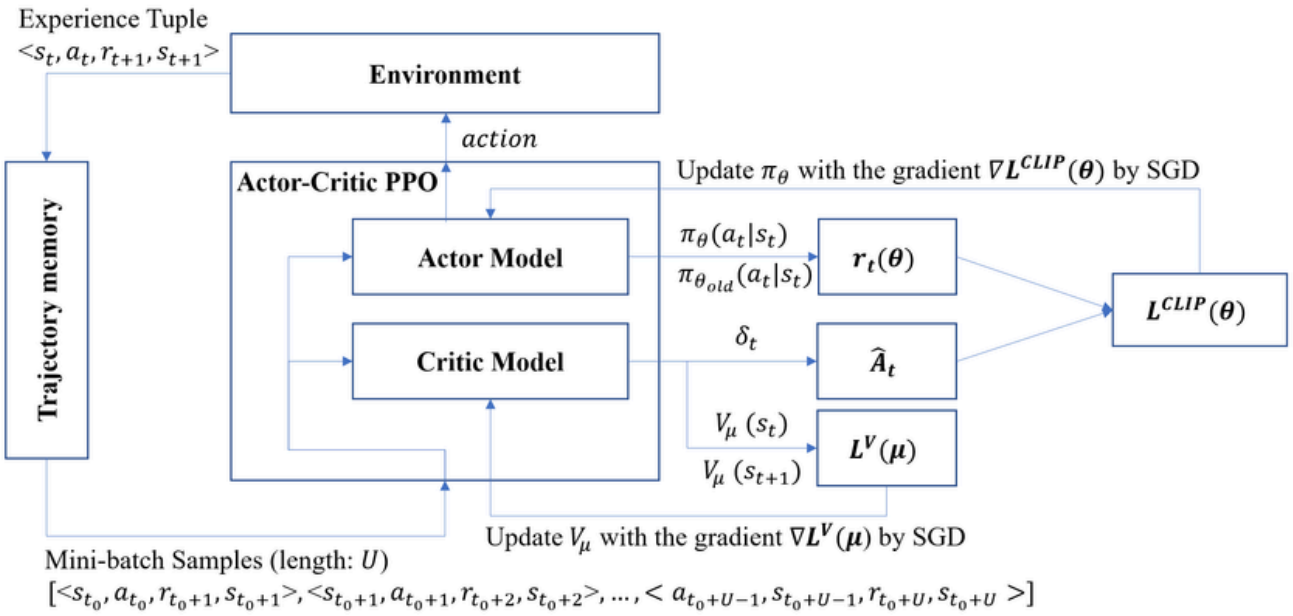
	Date	Volume	Open	Close	High	Low
0	2017-03-20	30843980	43.47	43.40	43.52	43.23
1	2017-03-21	50759559	43.50	42.51	43.67	42.38
2	2017-03-22	27334980	42.47	42.49	42.77	42.35
3	2017-03-23	65753379	42.07	41.98	42.08	41.65
4	2017-03-24	42113639	42.10	41.76	42.20	41.46
...	...	...	...	...	...	...
752	2020-03-16	96519619	54.48	53.65	57.27	53.35
753	2020-03-17	83193719	54.53	55.90	56.30	52.46
754	2020-03-18	93044759	53.00	54.56	55.25	51.85
755	2020-03-19	74064959	54.41	55.58	57.63	52.78
756	2020-03-20	82877939	56.55	53.41	56.92	53.11

II. Thuật toán đề xuất

Trong bài toán tối ưu hóa lợi nhuận đầu tư cổ phiếu, chúng tôi lựa chọn sử dụng thuật toán Proximal Policy Optimization (PPO) làm phương pháp chủ đạo. PPO là một thuật toán thuộc nhóm Policy Gradient, nổi bật với tính ổn định và hiệu quả cao trong quá trình huấn luyện, đặc biệt phù hợp với các môi trường có tính biến động cao như thị trường tài chính.

PPO áp dụng cơ chế giới hạn cập nhật chính sách (clipped objective function), giúp kiểm soát việc điều chỉnh chính sách trong mỗi bước học, tránh hiện tượng cập nhật quá đà, từ đó đảm bảo quá trình học diễn ra ổn định. Trong bài toán này, PPO cho phép hệ thống tự động học hỏi từ dữ liệu thị trường và kinh nghiệm giao dịch trước đó, liên tục cải thiện chiến lược giao dịch để tối đa hóa lợi nhuận. Nhờ vào khả năng xử lý tốt không gian hành động liên tục và khả năng tổng quát hóa cao, PPO là lựa chọn lý tưởng để xây dựng một chiến lược giao dịch hiệu quả, thích ứng với sự biến động phức tạp của thị trường cổ phiếu.

Sơ đồ về cách hoạt động của thuật toán:



Cách thức hoạt động

PPO hoạt động dựa trên việc tối ưu hóa chính sách (policy) thông qua một hàm mục tiêu (objective function). Các bước chính của PPO bao gồm:

1. **Thu thập dữ liệu**:
    - Agent tương tác với môi trường (trong bài toán này là môi trường StockEnv) bằng cách sử dụng chính sách hiện tại πθ.
    - Tại mỗi bước, agent quan sát trạng thái st, thực hiện hành động at, nhận phần thưởng rt, và chuyển sang trạng thái tiếp theo s{t+1}.
    - Dữ liệu được thu thập dưới dạng các quỹ đạo: ((s_t, a_t, r_t, s_{t+1}).
2. **Tính toán Advantage**:
    - PPO sử dụng một hàm giá trị V(s) để ước lượng giá trị kỳ vọng của trạng thái s.

    - Advantage A_t được tính để đo lường mức độ tốt của hành động a_t so với giá trị trung bình của trạng thái. PPO thường sử dụng Generalized Advantage Estimation (GAE) để tính A_t, giúp giảm phương sai và cải thiện hiệu quả học.
3. **Cập nhật chính sách**:
    - PPO tối ưu hóa chính sách bằng cách tối ưu hóa hàm mục tiêu:
    - Hàm mục tiêu này đảm bảo rằng chính sách không thay đổi quá nhiều trong một lần cập nhật, giúp PPO ổn định hơn so với các phương pháp Policy Gradient khác.
4. **Cập nhật hàm giá trị**:
    - PPO sử dụng một mạng Critic để ước lượng hàm giá trị V(s).
    - Hàm giá trị được cập nhật bằng cách tối thiểu hóa sai số giữa giá trị dự đoán V(s_t)) và giá trị thực tế (dựa trên phần thưởng tích lũy).
5. **Lặp lại**:
    - Sau khi cập nhật chính sách và hàm giá trị, PPO tiếp tục thu thập dữ liệu mới và lặp lại quá trình trên cho đến khi đạt được hiệu suất mong muốn.

Thông qua các bước này, thuật toán PPOcó thể học và điều chỉnh chiến lược giao dịch của mình để tối ưu hóa lợi nhuận đầu tư cổ phiếu trong môi trường biến động liên tục.

Xây dựng hàm action

Hành động trong môi trường được định nghĩa trong StockEnv.action_space là một không gian liên tục Box với hai thành phần:

- **action_type**: Loại giao dịch (-1 là bán, 0 là giữ, 1 là mua), nằm trong khoảng [-1, 1].
- **action_ratio**: Tỷ lệ tiền/cổ phiếu sử dụng trong giao dịch, nằm trong khoảng [0, 1].

Trong hàm step của StockEnv:

- Nếu action_type > 0.3: Thực hiện lệnh **mua** với tỷ lệ từ buy_min (0.44) đến buy_max (0.57).
- Nếu action_type < -0.3: Thực hiện lệnh **bán** với tỷ lệ từ sell_min (0.35) đến sell_max (0.84).
- Ngược lại: **Giữ** (không giao dịch).

Hàm select_action trong lớp PPO tạo ra hành động từ mạng Actor, thêm nhiễu khám phá (exploration noise) trong giai đoạn huấn luyện và sử dụng hành động deterministic trong giai đoạn kiểm thử.

Xây dựng Observation Space

Không gian quan sát (observation_space) là một vector có chiều dài (window_size * 5 + 5), bao gồm:

1. **Dữ liệu thị trường (window_size * 5)**:
    - Dữ liệu OHLCV (Open, High, Low, Close, Volume) trong window_size ngày (mặc định là 20).
    - Được chuẩn hóa bằng cách chia cho giá trị đầu tiên trong cửa sổ và trừ 1, thể hiện % thay đổi.

2. **Thông tin tài khoản (5 đặc trưng)**:
    - Tỷ lệ tiền mặt so với vốn ban đầu (cash_balance / initial_capital - 1).
    - Giá trị cổ phiếu so với vốn ban đầu (shares_held * current_price / initial_capital).
    - % thay đổi giá hiện tại so với giá cách đó window_size ngày.
    - Tỷ lệ số phiên không giao dịch (no_transaction_count / transaction_session).
    - Tỷ lệ tổng tài sản so với vốn ban đầu (total_assets / initial_capital - 1).

Không gian này cung cấp thông tin đầy đủ về thị trường và trạng thái tài khoản để tác nhân đưa ra quyết định.

Xây dựng hàm reward

Hàm thưởng (reward) trong StockEnv.step được thiết kế để khuyến khích giao dịch hiệu quả và phạt các hành vi không mong muốn:

- **Thưởng giao dịch**:
    - +0.1 khi thực hiện giao dịch thành công (mua hoặc bán).
- **Phạt không giao dịch**:
    - -0.5 nếu không giao dịch trong transaction_session phiên (10 phiên), kèm theo phạt tiền mặt (-150).
- **Thưởng/phạt dựa trên thay đổi tài sản**:
    - (total_assets - previous_assets) / previous_assets * 10, phản ánh tỷ lệ thay đổi tài sản.
- **Điều kiện kết thúc**:
    - -10 nếu tổng tài sản dưới ngưỡng total_assets_min (50000) hoặc tiền mặt âm quá mức (-5000).
    - +10 nếu tổng tài sản vượt 1 triệu USD.

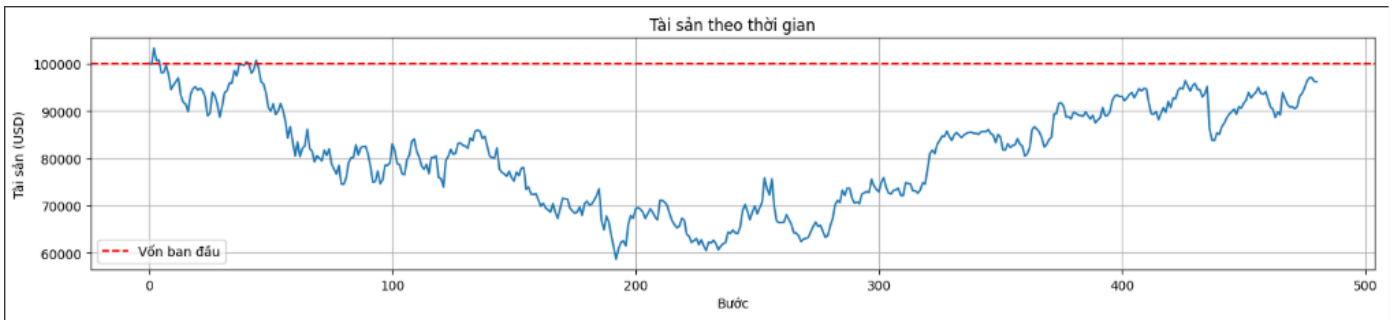
Hàm thưởng này cân bằng giữa khuyến khích giao dịch sinh lợi và kiểm soát rủi roi

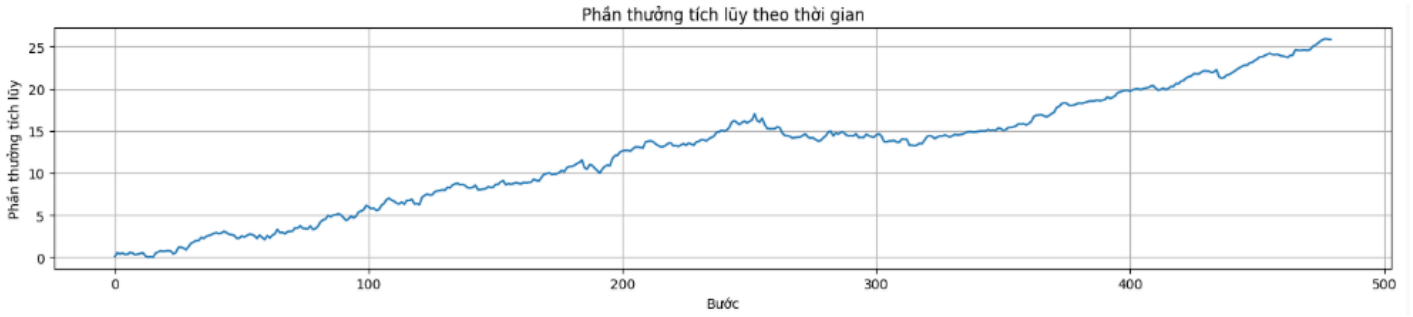
III. Kết quả đạt được

Giá trị tài sản tối đa thu được:

```
===== KẾT QUẢ KIỂM THỬ =====
Tổng số bước: 480
Tài sản ban đầu: $100000.00
Tài sản cuối cùng: $96098.60
Lợi nhuận: $-3901.40 (-3.90%)
Tổng reward: 25.85
```

Kết quả trực quan hóa:





Tối ưu hóa chiến lược đầu tư cổ phiếu bằng thuật toán Proximal Policy Optimization (PPO)

Việc áp dụng thuật toán Proximal Policy Optimization (PPO) trong tối ưu hóa chiến lược đầu tư cổ phiếu đã chứng minh được hiệu quả mạnh mẽ. Bằng cách sử dụng các kỹ thuật học củng cố tiên tiến, mô hình tuy không đạt được kết quả mong muốn nhưng đã cho thấy tiềm năng đáng kể trong việc thích nghi với sự biến động của thị trường.

Các cải tiến và ưu điểm của PPO:

- Các cải tiến

1. **Clipping**: Giới hạn cập nhật chính sách (clip_eps=0.2) để đảm bảo ổn định, tránh thay đổi quá lớn.
2. **GAE**: Sử dụng Generalized Advantage Estimation (gae_lambda=0.95) để tính lợi thế chính xác hơn.
3. **Đơn giản hóa**: Thay thế ràng buộc phức tạp của TRPO bằng hàm mất mát dễ tối ưu (Adam, lr=3e-4).
4. **Hiệu quả dữ liệu**: Tận dụng dữ liệu on-policy tốt hơn qua ReplayBuffer, dù xóa sau mỗi lần huấn luyện.

- Ưu điểm

1. **Ổn định**: Giảm rủi ro hiệu suất bất ổn, phù hợp với giao dịch cổ phiếu biến động.
2. **Linh hoạt**: Hành động liên tục (action_type, action_ratio) thích nghi tốt với thị trường.
3. **Dễ triển khai**: Đơn giản, ít đòi hỏi tài nguyên tính toán, dễ tinh chỉnh.

IV. Minh chứng kết quả

Source: Link