TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN TOÁN ỨNG DỤNG VÀ TIN HỌC

BÀI TẬP LỚN:

ỨNG DỤNG HỌC TĂNG CƯỜNG VÀO ĐẦU TƯ CHỨNG KHOÁN

Giáng viên hướng dẫn: TS. Nguyễn Thị Ngọc Anh

Nhóm Sinh viên thực hiện:

Vũ Thị Ngọc

Trịnh Hoàng Đức

Đặng Đình Trung

Lớp: KSTN Toán Tin K60

HÀ NỘI - 06/2018

Mục lục

Chương 1

Giới thiệu bài toán

Ngày nay, việc đầu tư chứng khoán là lĩnh vực đầu tư phát triển rất mạnh, thu hút sự quan tâm của không ít các nhà đầu tư trên khắp thế giới. Nhà đầu tư tốt sẽ thu được lợi nhuận kể cả khi giá cổ phiếu có chiều hướng tăng hoặc giảm. Cụ thể:

- Khi cổ phiếu đang có xu hướng tăng, nhà đầu tư sẽ mua cổ phiếu đó trước khi giá tiếp tục tăng và bán trước khi giá sẽ giảm
- Khi cổ phiếu đang có xu hướng giảm, nhà đầu tư sẽ đi vay cổ phiếu và lập tức bán đi. Đợi khi giá cổ phiếu giảm đến một mức nào đó, nhà đầu tư sẽ đi mua lại đủ cổ phiếu và trả cho người đã cho vay

Chiến lược đầu tư rất thông minh phải không? Tuy nhiên vấn đề ở đây là ta biết khi nào giá cổ phiếu đang có xu hướng tăng và khi nào giảm.

Dựa trên suy nghĩ rằng giá cổ phiếu thay đổi là có quy luật, người ta xây dựng một cách tiếp cận tương đối mới với giao dịch tài chính là sử dụng thuật toán học máy để dự đoán giá cổ phiếu tăng hoặc giảm trước khi nó xảy ra.

Trong bài này, nhóm em xin trình bày một cách tiếp cận như vậy với phương pháp học tăng cường và sử dụng ý tưởng cốt lõi để tối ưu là phương pháp hướng tăng Gradient. Cụ thể:

- *Chương 2: Các khái niệm cơ bản về học tăng cường và phương pháp hướng tăng Gradient*
- *Chương 3: Cách ứng dụng vào đầu tư chứng khoán*
- *Chương 4: Kết luận và một số kết quả được nhóm rút ra trong quá trình làm*

Chương 2

Phương pháp

2.1 Giới thiệu

Trong ngành khoa học máy tính, học tăng cường (Reinforcement Learning)là một lĩnh vực con của học máy (Machine Learning), nghiên cứu cách thức một tác tử (agent) trong một môi tường nên chọn thực hiện các hành động nào để có được phần thưởng có giá trị lớn nhất về lâu về dài. Các thuật toán học tăng cường cố gắng tìm một chiến lược ánh xạ từ không gian trạng thái tới không gian hành động mà agent nên chọn trong các trạng thái đó.

2.2 Quá trình quyết định Markov

Quá trình quyết định Markov (Markov Decision Processes, ký hiệu là MDP)cung cấp một nền tảng toán học cho việc mô hình hóa việc ra quyết định trong các tình huống mà kết quả là một phần ngẫu niên, một phần dưới sự điều khiển của một người ra quyết định. MDP rất hữu dụng trong việc học một loạt các bài toán tối ưu hóa được giải quyết thông qua quy hoạch động và hocn tăng cường. MDP được biết đến sớm nhất vào những năm 1950 (cf. Bellman 1957). Một cốt lõi của nghiên cứu về quá trình ra quyết định Markov là từ kết quả của cuốn sách của Ronald A. Howard xuất bản năm 1960, Quy hoạch động và quá trình Markov. Chúng được sử dụng trong rất nhiều các lĩnh vực khác nhau, bao gồm robot, điều khiển tự động,kinh tế, và chế tạo.

Chính xác hơn, một quá trình quyết định Markov là một quá trình điều khiển ngẫu nhiên thời gian rời rạc. Tại mỗi bước thời gian, quá trình này trong một vài trạng thái s, và người ra quyết định có thể chọn bất kỳ hành động a nào có hiệu lực trong trạng thái s. Quá trình này đáp ứng tại bước thời gian tiếp theo bằng cách di chuyển ngẫu nhiên vào một trạng thái mới s', và đưa ra cho người ra quyết định một phần thưởng tương ứng $R_a(s, s')$

Xác suất mà quá trình di chuyển vào trạng thái mới của nó s' bị ảnh hưởng bởi hành động được chọn. Đặc biệt, nó được đưa ra bởi hàm chuyển tiếp trạng thái $P_a(s, a')$. Do đó, trạng thái kế tiếp s' phụ thuộc vào trạng thái hiện tại s và a đã cho, lại độc lập có điều kiện với toàn bộ trạng thái và hành động trước đó. Nói cách khác, các trạng thái chuyển tiếp của một quá trình MDP thỏa mãn thuộc tính Markov.

Quá trình quyết định Markov là một phần mở rộng của chuỗi Markov; khác biệt là ở sự bổ sung của các hành động (cho phép lựa chọn) và phần thưởng (cho động cơ). Ngược lại, nếu chỉ có một hành động tồn tại cho mỗi trạng thái và tất cả các phần thưởng là giống nhau (ví dụ: zero), một quá trình quyết định Markov làm giảm một chuỗi Markov.

MDP là một quá trình điều khiển ngẫu nhiên thời gian rời rạc. MDP là một tập 5 dữ liệu $< \mathbb{S}, \mathbb{A}, P, R, \gamma >$. Trong đó:

- $\mathbb{S}$: không gian các trạng thái
- $\mathbb{A}$: không gian các hành động
- $P : \mathbb{S} \times \mathbb{A} \times S \to \mathbb{R}$: hạt nhân chuyển tiếp Markov
- $R : \mathbb{S} \times \mathbb{A} \to \mathbb{R}$: hàm phần thưởng, $0 < \gamma < 1$ là hệ số chiết khấu

Giả sử rằng, tahi thời điểm t, trạng thái $S_t = s$ và agent có hành động $A_t = a$. Khi đó, xác suất của trạng thái $B \in \mathbb{S}$ tại thời điểm $t + 1$ được cho bởi công thức:

$$P(s, a, B) = \mathbb{P}(S_{t+1} \in B | S_t = s, A_t = a) \text{ (1)}$$

Sau quá trình này, agent nhận được một phần thưởng ngẫu nhiên là $R_(t + 1)$. Hàm thưởng $R(s, a)$ là phần thưởng thu được khi thực hiện hành động a ở trang thái s

$$R(s, a) = \mathbb{E}[R_(t + 1) | S_t = s, A_t = a] \text{ (2)}$$

Tại bất kì bước thời gian nào, agent chọn hành động của nó theo một chính sách $\pi : \mathbb{S} \times A \to \mathbb{R}$ sao cho với mỗi $s \in \mathbb{S}, C \to \pi(s, C)$ là xác suất phân phối trên $(\mathbb{A}, A)$. Do đó, chính sách π và trạng thái ban đầu

$s_0 \in \mathbb{S}$ xác định chuỗi trạng thái - hành đồng - phần thưởng ngẫu nhiên $\{(S_t, A_t, R_{t+1})\}_{t\geq 0}$ với giá trị trên $\mathbb{S} \times \mathbb{A} \times \mathbb{R}$. Trong một không gian vô hạn,hiệu suất của agent thường được tính bằng tổng phần thưởng chiếu khấu thu được sau một chính sách là

$$G_t = \sum_{t=0}^{\infty} \gamma^t R_{t+k+1} \ (3)$$

Vì phần thưởng này là ngẫu nhiên, agent xem xét giá trị kì vọng của nó, thưởng được gọi là hàm giá trị trạng thái

$$V_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s] \ (4)$$

Trong đó, chỉ số $\mathbb{E}_\pi$ chỉ ra rằng xá hành đồng được chọn theo chính sách π. Hàm giá trị trạng thái được đo tốt khi agent ở trong một trạng thái nhất định và tuân theo một chính sách nhất định. Tương tự, ta có hàm giá trị hành động

$$Q_\pi(s,a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a] \ (5)$$

Ta có mối liên hệ giữa V_π và Q_π

$$V_\pi(s) = \int_A \pi(s,a) Q_\pi(s,a) da \ (6)$$

Hầu như tất cả các thuật toán học tăng cường được thiết kế để tính các hàm giá trị này dựa trên các phương trình Bellman

$$V_\pi(s) = R_\pi(s) + \gamma T_\pi V_\pi(s) \ (7)$$
$$Q_\pi(s,a) = R(s,a) + \gamma T_a V_\pi(s) \ (8)$$

Khi chúng ta biểu diễn bở T_a thì toán tử chuyển với hành đồng a (chính sách π)

$$T_sF(s) = \mathbb{E}[F(S_{t+1}|S_t = s, A_t = a] = \int_S P(s, a, s')F(s')ds' \ (9)$$
$$T_\pi F(s) = \mathbb{E}_p i[F(S_{t+1}|S_t = s] = \int_{\mathbb{A}} \pi(s, a) \int_{\mathbb{S}} P(s, a, a')F(s')F(s')ds'da \ (10)$$

Các phương trình này có thể được viết dưới dạng phương trình điểm cố định tho một số giả thuyết về hàm thưởng, thừa nhận một giải pháp duy nhất theo định lý ánh xạ co. Mục tiêu của agent là chọn một chính sách π_* mà tối đa hóa được lợi nhuận ở tất cả các trạng thái có thể. Chính sách như vậy được gọi là tối ưu và các hàm giá trị (sửa lỗi) được gọi là hàm giá trị trạng thái tối ưu

$$V_*(s) = \sup_\pi V_\pi(s) \ (11)$$

và hàm tối ưu giá trị-hành động

$$Q_*(s, a) = \sup_\pi Q_\pi(s, a) \ (12)$$

Các hàm giá trị tối ưu thỏa mãn phương trình Bellman sau

$$V_*(s) = \sup_a Q_*(s, a) = \sup_a R(s, a) + \gamma T_a V_*(s) \ (13)$$
$$Q_*(s, a) = R(s, a) + \gamma T_a V_*(s) \ (14)$$
$$= R(s, a) + \gamma \int_{\mathbb{S}} P(s, a, s') \sup_{a'} Q_*(s', a')ds$$

Một lần nữa, đây là những phương trình điểm cố định mà luôn tồn tại và duy nhất một giải pháp do định lý ánh xạ co. Với hàm giá trị trạng thái tối ưu Q_*. Một chính sách tối ưu thu được bằng cách chọn mỗi trạng thái hành động với giá trị lớn nhất Q_*

$$a_* = argsup_a \; Q_*(s,a) \; (15)$$

Chính sách tham lam này được xác định và chỉ phụ thuộc vào trạng thái hiện tại của hệ thống

2.3 Phương pháp Gradient Ascent

Trong Machine Learning nói riêng và Toán Tối Ưu nói chung, chúng ta thường xuyên phải tìm giá trị lớn nhất (hoặc đôi khi là nhỏ nhất) của một hàm số nào đó.

Nhìn chung, việc tìm giá trị tối ưu toàn cục của các hàm mục tiêu trong Machine Learning là rất phức tạp, thậm chí là bất khả thi. Thay vào đó, người ta sẽ cố gắng tìm các điểm tối ưu địa phương của bài toán.

Ta đã biết, x^* là nghiệm tối ưu địa phương bài toán max f(x) v.đ.k x $\in \mathbb{R}^n$ thì x^* là nghiệm của phương trình $\nabla f(x) = 0$. Để tính được nghiệm tối ưu toàn cục, ta có thể tính tất cả các nghiệm của phương trình $\nabla f(x) = 0$. Tuy nhiên, việc này đa số cũng là bất khả thi vì có thể sự phức tạp của đạo hàm hoặc dữ liệu có số chiều lớn làm ta không thể tìm được nghiệm.

Hướng tiếp cận phổ biến nhất là xuất phát từ một điểm mà chúng ta coi là gần với nghiệm của bài toán, sau đó dùng một phép toán lặp để tiến dần đến điểm cần tìm, tức đến khi đạo hàm gần với 0

Theo lí thuyết qui hoạch phi tuyến trong Toán Tối ưu, ta biết rằng từ một điểm $u \in \mathbb{R}^n$ thì $\nabla f(u)$ chính là hướng tăng của f(x) tại u. Tức là, $\exists \epsilon > 0$ thỏa mãn: f(u) $\leq$ f(u+$\delta . \nabla f(u)$), $\forall \delta \in (0, \epsilon)$

Xuất phát từ ý tưởng này, ta sẽ triển khai để xây dựng một dãy $(x^n)_{n=0}^{\infty}$ sao cho $lim\ x^n \to x^*$ khi n $\to \infty$

B0: Chọn $x^0 \in \mathbb{R}^n$, $\epsilon > 0$, $\eta > 0$, gán k := 0, N > 1, n = 1

B1: Nếu $\nabla f(x^k) \approx 0$. Chọn x đủ gần x^k và kiểm tra f(x) - f(x^k).

- Nếu f(x) - f(x^k) ≤ 0. Gán $x^* = x^k$ chuyển sang B3
- Ngược lại. Chuyển sang B2

B2: n := n+1.

- Nếu n = N hoặc $|f(x^k + \eta.\nabla f(x^k)) - f(x^k)| < \epsilon$ thì gán $x^* = x^k$ và chuyển sang B3.
- Ngược lại, gán $x^k = x^k + \eta.f(x^k)$ rồi quay lại B1

B3: Thông báo $x*$ và kết thúc chương trình

Chương 3

Ứng dụng

3.1 Hai khái niệm cơ bản

• **Long**: Ta nói rằng ta **long** n cổ phiếu nghĩa là ta sẽ mua n cổ phiếu và hi vọng giá cổ phiếu này sẽ tăng

• **Short**: Ta nói rằng ta **short** n cổ phiếu nghĩa là ta sẽ đi vay n cổ phiếu và bán ngay. Với hi vọng giá cổ phiếu sẽ giảm, ta sẽ đợi đến khi nó giảm và mua lại rồi trả cho cổ phiếu cho người cho vay

3.2 Hàm mục tiêu: SHARPE

Với mỗi thời điểm của lợi nhuận đầu tư, Sharpe được xác định bởi công thức

$$S_T = \frac{Average(R_t)}{Standard\ Deviation(R_t)} \text{ với } t = 1, 2, ..., \text{T}$$

Trong đó, R_t là lợi nhuận đầu tư cho giao dịch tại thời điểm t so với thời điểm t-1. Bằng trực giác, Sharpe thưởng cho các chiến lược đầu tư dựa vào xu hướng ít biến động để tạo ra lợi nhuận.

3.3 Hàm đầu tư

Nhà đầu tư sẽ cố gắng sao cho Sharpe đạt giá trị lớn nhất cho mỗi chuỗi thời gian nhất định. Với bài báo cáo này, hàm đầu tư có công thức là:

$$F_t = tanh(w^T x_t)$$

trong đó M là số chuỗi thời gian đầu vào cho nhà đầu tư, tham số $w \in R^{M+2}$, vector đầu vào $x_t = [1, r_t, ..., r_{t-M}, F_{t-1}]$, và $r_t = p_t - p_{t-1}$, với p_t là giá của cổ phiếu ở thời điểm t

Để ý rằng r_t là sai khác giá trị của cổ phiếu giữa thời điểm hiện tại t và thời điểm trước đó. Vì vậy, r_t là lợi nhuận trên một cổ phiếu được mua ở thời gian $t - 1$

Ngoài ra, hàm $F_t \in [-1, 1]$ cho biết trường hợp đầu tư tại thời điểm t. Có 3 trường hợp có thể có: long, short, neutral

- Long khi $F_t > 0$. Trong trường hợp này, nhà đầu tư mua một cổ phiếu với giá p_t và hi vọng rằng nó tăng giá ở thời điểm $t + 1$

- Short khi $F_t < 0$. Ở trường hợp này, nhà đầu tư bán một cổ phiếu mà họ không sở hữu với giá p_t, với kì vọng nó có thể thực hiện giao dịch tại thời điểm $t + 1$. Nếu giá cao hơn tại thời điểm $t + 1$ thì nhà đầu tư bắt buộc phải mua ở giá cao hơn thời điểm $t+1$ để hoàn thành hợp đồng. Nếu giá ở thời điểm $t+1$ thấp hơm thì nhà đầu tư thu được lợi nhuận.

- Neutral khi $F_t = 0$. Trong trường hợp này nhà đầu tư không được cũng không mất lợi nhuận.

Như vậy, F_t cho biết cổ phiếu tại thời điểm t. Đó là $n_t = \mu.F_t$, cổ phiếu được mua (long) hoặc bán (short) với μ là số lượng cổ phiếu tối đa cho mỗi giao dịch. Lợi nhuận ở thời điểm t phụ thuộc vào F_{t-1}:

$$R_t = \mu(F_{t-1}.r_t - \delta|F_t - F_{t-1}|$$

trong đó, δ là chi phí giao dịch tại thời điểm t. Nếu $F_t = F_{t-1}$ (không thay đổi đàu tư trong thời điểm này) thì sẽ không mất phí giao dịch. Nếu không sẽ mất phí tỷ lệ thuận với sự chênh lệch trong cổ phiếu nắm giữ.

Đầu tiên, $(\mu.F_{t-1}.r_t)$ là kết quả trả về từ quyết định đầu tư tại thời điểm $t-1$. Ví dụ nếu $\mu = 20$ cổ phiếu, quyết định mua một nửa mức tối đa cho phép ($F_{t-1} = 5$) và mỗi cổ phiếu tăng $r_t = 8$ đơn vị giá, kì hạn này sẽ là 80. tổng lợi nhuận thu được (bỏ qua các khoản phí giao dịch phát sinh ở thời điểm t)

3.4 Hướng tăng

Tối đa hóa Sharpe tuân theo hướng tăng. Đầu tiên chúng ta xác định hàm thưởng bằng cách sử dụng công thức cơ bản của thống kê cho kì vọng và phương sai.

Ta có:

$$S_T = \frac{E[R_t]}{\sqrt{E[R_t^2] - (E[R_t])^2}} = \frac{A}{\sqrt{B - A^2}}$$

,với $A = \frac{1}{T} \sum_{t=1}^{T} R_t$ và $B = \frac{1}{T} \sum_{t=1}^{T} R_t^2$

Sau đó chúng ta lấy đạo hàm của S_T sử dụng chuỗi:

$$\frac{\partial S_T}{\partial w} = \frac{\partial}{\partial w}\left\{\frac{A}{\sqrt{B - A^2}}\right\} = \frac{dS_T}{dA}.\frac{\partial A}{\partial w} + \frac{dS_T}{dB}.\frac{\partial B}{\partial w}$$

$$= \sum_{t=1}^{T}\left\{\frac{dS_T}{dA}.\frac{dA}{dR_t} + \frac{dS_T}{dB}.\frac{dB}{dR_t}\right\}.\frac{\partial R_t}{\partial w}$$

$$= \sum_{t=1}^{T}\left\{\frac{dS_T}{dA}.\frac{dA}{dR_t} + \frac{dS_T}{dB}.\frac{dB}{dR_t}\right\}.\left\{\frac{dR_t}{dF_t}.\frac{\partial F_t}{\partial w} + \frac{dR_t}{dF_{t-1}}\frac{\partial F_{t-1}}{\partial w}\right\}$$

Đạo hàm từng phần, kết quả trả về được hàm:

- $\frac{dR_t}{dF_t} = \frac{d}{dF_t}\ \{\mu\,(F_{t-1}.r_t - \delta\,|F_t - F_{t-1}|)\}$

$$= \frac{d}{dF_t}\ \{-\mu.\delta.\,|F_t - F_{t-1}|\}$$

$$= \begin{cases} -\mu.\delta & F_t - F_{t-1} > 0 \\ \mu.\delta & F_t - F_{t-1} < 0 \end{cases}$$

$$= -\mu\delta.sign\,(F_t - F_{t-1})$$

- $\frac{dR_t}{dF_{t-1}} = \frac{d}{dF_{t-1}}\ \{\mu\,(F_{t-1}.r_t - \delta\,|F_t - F_{t-1}|)\}$

$$= \mu.r_t - \frac{d}{dF_{t-1}}\{-\mu.\delta.\,|F_t - F_{t-1}|\}$$

$$= \begin{cases} \mu.\delta & F_t - F_{t-1} > 0 \\ -\mu.\delta & F_t - F_{t-1} < 0 \end{cases}$$

$$= \mu.r_t + \mu\delta.sign\,(F_t - F_{t-1})$$

Khi đó, đạo hàm từng phần dF_t/dw và dF_{t-1}/dw được tính như sau:

$$\bullet \frac{\partial F_t}{\partial w} = \frac{\partial}{\partial w} \left\{tanh(w^T x_t)\right\}$$
$$= \left(1 - tanh\left(w^T x_t\right)^2\right) . \frac{\partial}{\partial w} . \left\{w^T . x_t\right\}$$
$$= \left(1 - tanh\left(w^T x_t\right)^2\right) \left\{x_t + w_{M+2} \frac{\partial F_{t-1}}{\partial w}\right\}$$

Lưu ý rằng đạo hàm $\partial F_t/\partial w$ có tính hồi quy và phụ thuộc vào tất cả các giá trị trước đó của nó. Điều này nghĩa là để tính được tham số thì ta phải có giá trị $\partial F_t/\partial w$ từ chuỗi thời gian ban đầu. Bởi vì dữ liệu cổ phiếu trong khoảng 1000 - 2000 mẫu, điều này làm chậm hướng tăng nhưng vẫn tính toán được.

Khi điều kiện $\partial S_t/\partial w$ được tính toán, khối lượng ta cập nhật tuân theo nguyên tắc hướng tăng: $w_{t+1} = w_t + \rho.\partial S_T/\partial w$. Quá trình đươc lặp lại với số lần lặp tối đa N_e.

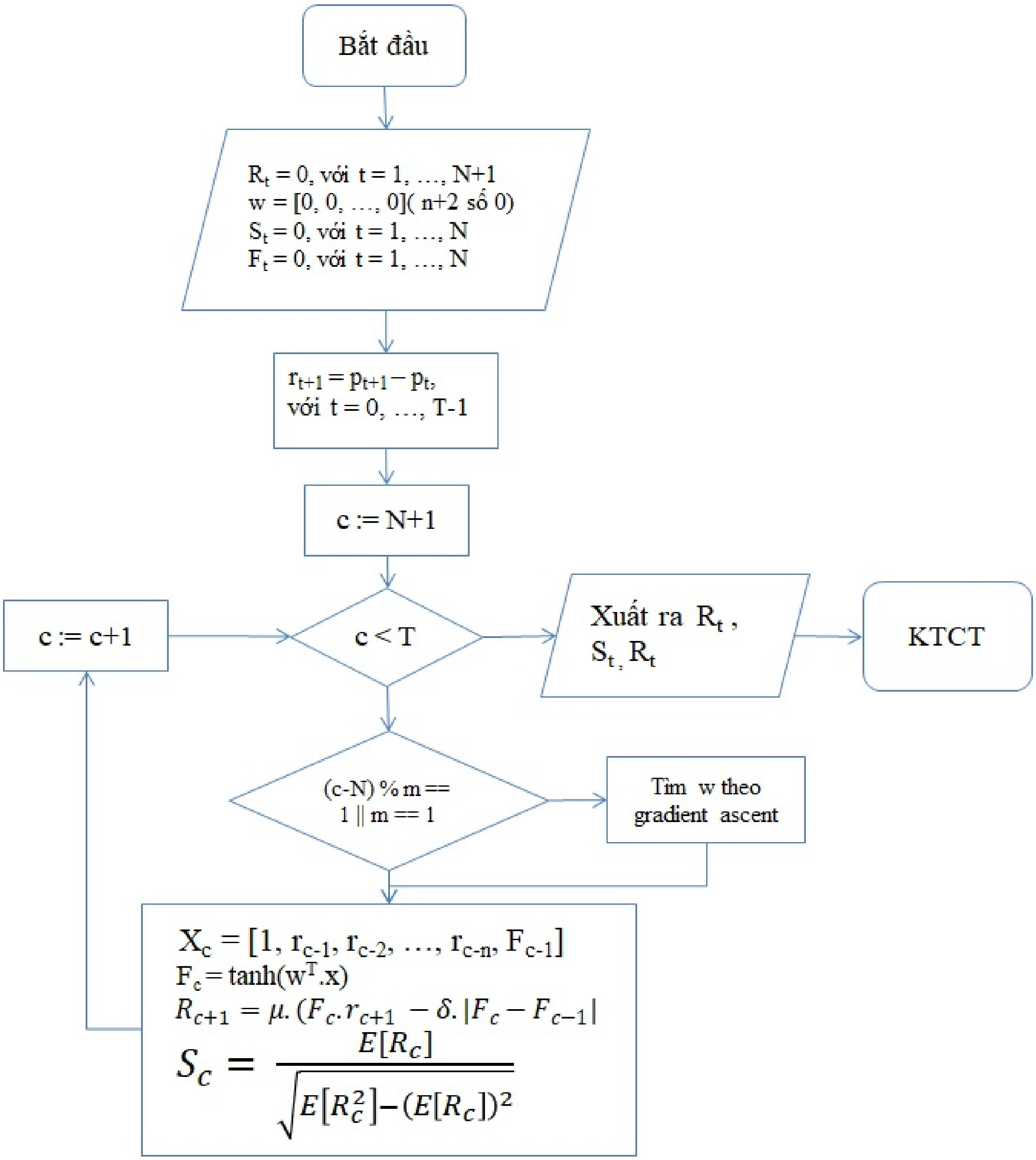
3.5 Thuật toán

***Input:**

- Kho dữ liệu giá $\mathbb{P}$ trong T ngày
- n là số ngày trong trạng thái(không gian trạng thái là $\mathbb{R}^{n+2}$)
- N là số ngày dùng để luyện
- m là số ngày sẽ áp dụng cùng một chiến thuật
- μ là số cổ phiếu tối đa **long** hoặc **short**
- δ là tỉ lệ phần trăm chi phí giao dịch trên 1 cổ phiếu
- N_e là số lần lặp tối đa dùng trong phương pháp hướng tăng gradient
- ϵ là sai số chấp nhận trong phương pháp hướng tăng gradient

***Output:** Lợi nhuận R, shapre S và phương án đầu tư F

***Thuật toán:**



Bắt đầu
$R_t = 0$, với $t = 1, \ldots, N+1$
$w = [0, 0, \ldots, 0]$(n+2 số 0)
$S_t = 0$, với $t = 1, \ldots, N$
$F_t = 0$, với $t = 1, \ldots, N$
$r_{t+1} = p_{t+1} - p_t$,
với $t = 0, \ldots, T-1$
c := N+1
c < T
Xuất ra R_t ,
S_t, R_t
KTCT
c := c+1
(c-N) % m ==
1 || m == 1
Tìm w theo
gradient ascent
$X_c = [1, r_{c-1}, r_{c-2}, \ldots, r_{c-n}, F_{c-1}]$
$F_c = \tanh(w^T.x)$
$R_{c+1} = \mu.(F_c.r_{c+1} - \delta.|F_c - F_{c-1}|$
$S_c = \frac{E[R_c]}{\sqrt{E[R_c^2] - (E[R_c])^2}}$

Chương 4

Kết luận

Dưới đây là một số kết quả khi chạy thuật toán để tìm ra phương án đầu tư cho mã cổ phiếu MSFT

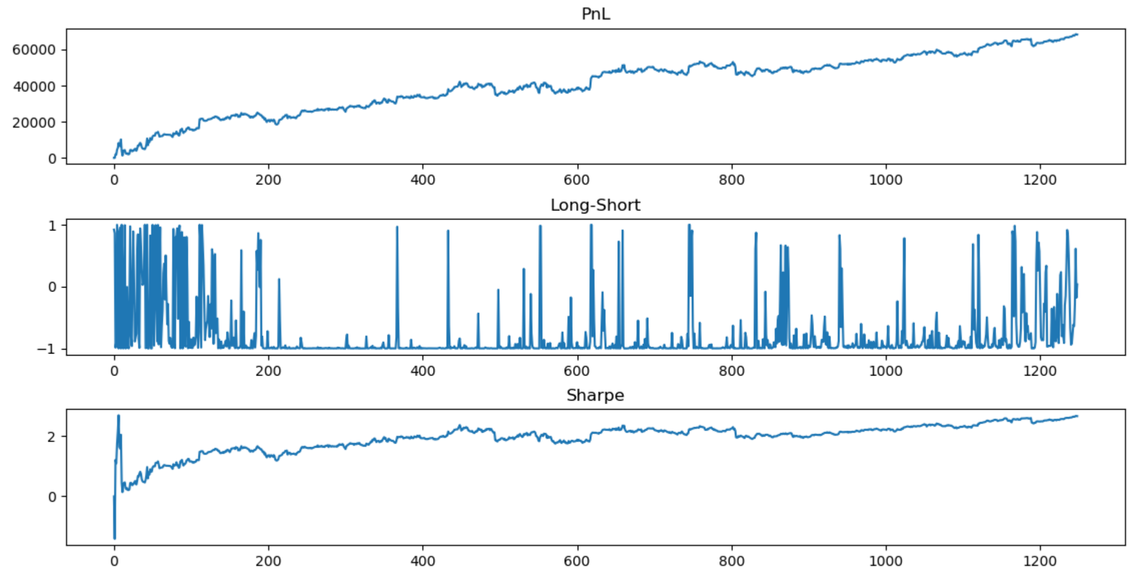
4.1 Không nên quá tập trung vào việc luyện để tìm chiến lược w

Nhìn vào hình 4.1 và 4.2, ta có nhận xét rằng để chạy được ra kết quả hình 4.2 thì ta sẽ mất nhiều thời gian hơn(vì cứ 2 ngày ta lại phải luyện lại để tìm w) nhưng lại không cho kết quả kém hơn.

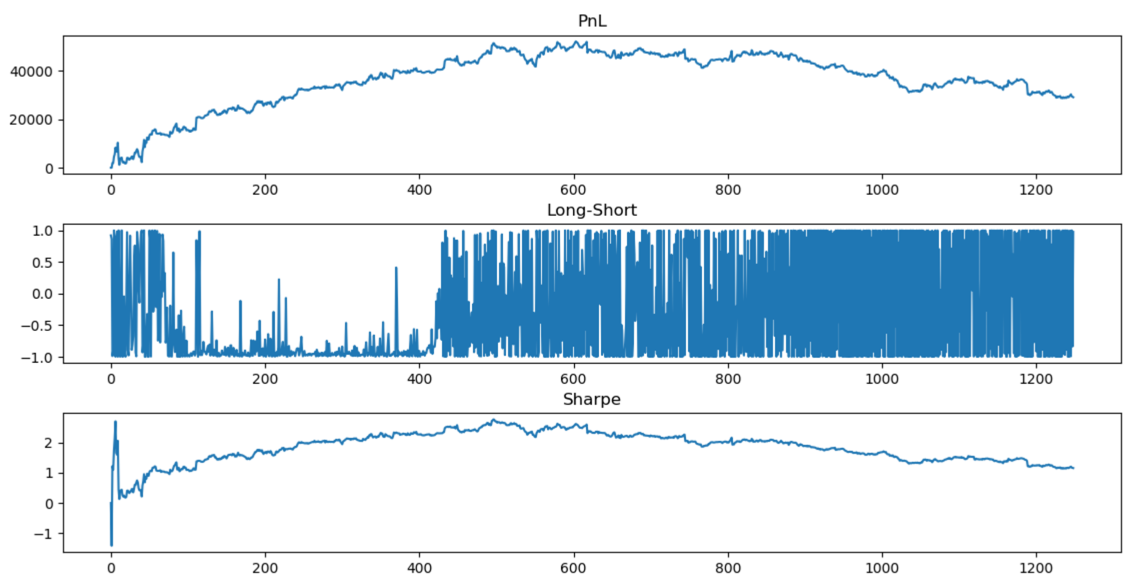
Giải thích cho trường hợp này, ta hiểu rằng mô hình ta đang làm là tại mỗi thời điểm luyện, ta sẽ tìm w để sao cho nếu hôm trước ta dùng w này để đưa ra chiến lược đầu tư thì sẽ có sharpe cao nhất. Và ta sẽ dùng w đã tìm được để đưa ra chiến lược đầu tư cho các ngày tiếp theo với hi vọng là xu hướng các ngày sau tiếp theo không thay đổi nhiều về xu hướng.

Tuy nhiên, việc này lại thường xuyên xảy ra vì giá của một cổ phiếu còn phụ thuộc vào rất nhiều nguyên nhân. Nếu ta cứ cố tối ưu tại mỗi thời điểm thì sẽ dễ dẫn đến hiện trạng overfit.

Ngoài ra, ta có thể thấy việc chọn các giá trị n, N nên là bội của 5. Bởi lẽ, thị trường chứng khoán hoạt động 5 ngày/tuần nên việc sử dụng những con số này sẽ rất có ý nghĩa về mặt thực tế để tìm qui luật của giá chứng khoán.



Hình 4.1: T = 1280, n = 5, N = 30, m = 5



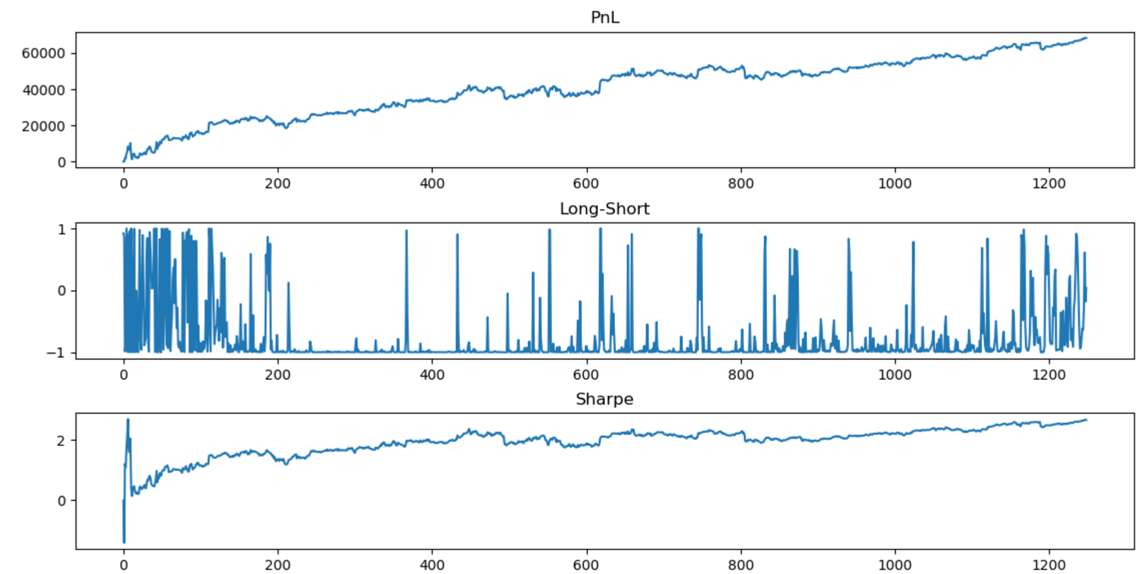
Hình 4.2: T = 1280, n = 5, N = 30, m = 2

4.2 Việc luyện nhiều bằng nhiều dữ liệu chưa hẳn đưa chiến lược w tốt

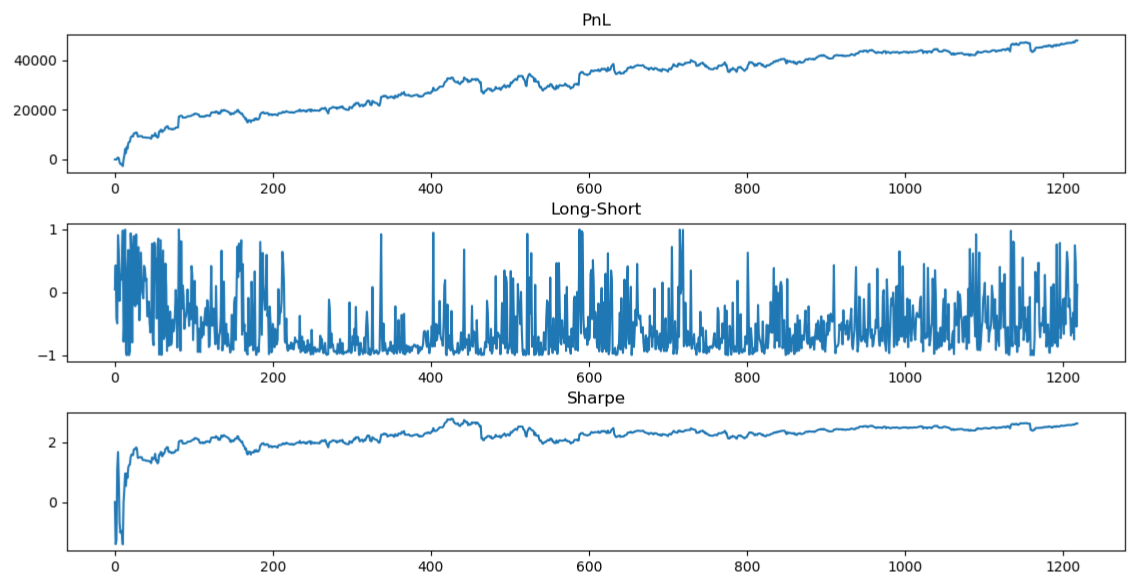
Nhìn vào hình 4.3 và 4.4, ta có nhận xét rằng shapre trong 2 hình không có nhiều thay đổi nhưng lợi nhuận thì có(dao động 60.000 và 40.000)

Nguyên nhân dẫn đến sự sai khác này là do thay đổi về N(số dữ liệu dùng để luyện). Có thể giải thích trường hợp này là thông tin về giá vì thay đổi phụ thuộc vào nhiều yếu tố nên qui luật của nó thường xảy ra trong một khoảng thời gian không dài. Vì thế nếu ta chọn khoảng thời gian hơi dài một chút thì sẽ là sự pha tạp của nhiều qui luật.

Từ nhiều qui luật mà ta lại đúc rút thành một qui luật để áp dụng thì việc đầu tư sẽ không còn đúng đắn nữa.



Hình 4.3: T = 1280, n = 5, N = 30, m = 5



Hình 4.4: T = 1280, n = 5, N = 60, m = 5

Tiệu tham khảo

[1] Pierpaolo G. Necchi: *Reinforcement Learning For Automated Trading*

[2] CS229 Application Project Gabriel Molina, SUID 5055783: *Stock Trading with Recurrent Reinforcement Learning (RRL)*