強化学習による高頻度取引戦略の構築

The Construction of High Frequency Trading Strategy via Reinforcement Learning

小林弘幸[1*] 和泉潔[1] 松島裕康[1] 坂地泰紀[1] 島田尚[1]
Hiroyuki Kobayashi[1] Kiyoshi Izumi[1] Hiroyasu Matsushima[1] Hiroki Sakaji[1] Takashi Shimada[1]

[1] 東京大学大学院工学系研究科　システム創成学専攻
[1] Department of Systems Innovation, School of Engineering, the University of Tokyo

Abstract: 近年ゲーム AI の領域の成功により強化学習の研究が活性化し, 金融市場においても将来価格の予測に留まらず取引戦略をシステマティックに開発するための枠組みとして強化学習のアプローチに注目が集まっている. 本研究では東京証券取引所に上場する銘柄のティックデータから高頻度取引戦略を構築するための強化学習アルゴリズムを提案する. ニューラルネットワークを関数近似器として利用した強化学習により日本株市場のトレーディング戦略の構築を行い, 戦略の収益性をバックテストで確認する.

1 はじめに

今日の自動取引システムの一般的な開発プロセスでは, 取引戦略を過去の一定期間において稼働させた場合に発生する仮想的な損益やシャープレシオ等の評価尺度を元に, 開発者が人手によって取引モデルのロジックや閾値などのパラメータのチューニングを行っている. あるいは投資尺度を目的関数としてモデルのパラメータを調整する探索ベースの最適化手法が採用されることもある.

このバックテストを繰り返すことで取引戦略を構築する方法は取引スパンの長短を問わず広く採用されているが, このアプローチには固有の限界も存在する.

第一に, 行動判断の良し悪しの判定及びそのフィードバックが不十分な点が指摘される. 人間のトレーダーであれば取引判断の可否は即座に判定され, その事実を学習することができるが, バックテストにおいては取引戦略に従って選択した注文行動の帰結は全期間の平均的なパフォーマンスによってのみ評価され, 各々の取引や注文送出結果を学習材料として利用することは想定されていない.

第二に, エージェント（＝取引アルゴリズム）が取り得た他の状態を考慮しないという点が挙げられる. エージェントの発注判断材料として, 他の市場参加者も観測可能な「現在及び過去の市場状況」（Market State）に加えて, 「ポジション残高」や「既存注文状況」等のエージェント固有の状況（Agent State）も利用されるのが通常であるが, ある時点において取り得る無数の可能性の中で, （それまでの取引履歴に依存して）偶然にも経由することとなったただ一つの Agent State と, 行った発注判断の結果のみがパフォーマンス評価対象となり, 他の全ての起こり得る Agent State と注文送出の可能性は考慮されることがない.

そこで本研究ではこの限界を克服するために強化学習アプローチを提案する. 強化学習は価値ベースの手法と方策ベースの手法に大別されるが, 本研究では価値ベースを採用する. [状態 s, 行動 a] のペアに対する価値関数 Q(s,a) という形で現在のエージェントが置かれている状態の下で選択し得る行動の価値を推定し, 価値が最大となる行動を選択する. この方法によりエージェントの行動の成否は毎回独立したデータセットとして直接学習対象とすることができる.

加えて, 強化学習においては価値関数の推定に必要なデータを生成するために, エージェントに最適でない行動も確率的に選択させる. エージェントの行動にこのランダム性を付与することで Agent State の系列に幅を持たせることができ, 取引戦略の構築という視点からは現実の限られたマーケットデータを余すことなく利用できる手法であると考えている.

2 関連研究

強化学習を日中の取引戦略の構築に用いる研究の歴史は長く,2001 年には既にマーケットメイク戦略を Sarsa や Actor-Critic の手法で学習する手法が提案されている (Nicholas(2001)[1]).

*連絡先：東京大学工学系研究科,
〒 113-8656 東京都文京区本郷 7-3-1
E-mail: d2019hkobayashi@socsim.org

実際の株式の取引データを用いた研究としては米国株の執行のための最適指値戦略を Q 学習ベースの手法で構築した Nevmyvaka(2006)[2] が挙げられる.

強化学習が再注目された 2010 年代中盤以降の研究としてはタイルコーディングベースの線形関数近似と適格度トレースを用いて欧州株式市場のマーケットメイク戦略の学習を行った Spooner(2018)[3] や高頻度の逆張り戦略を MLP を用いてオンライン学習で構築した Ganesh(2019)[4] が挙げられる.

本研究はこれらの先行研究の設定を参考に, Atari などのゲーム AI 領域を中心に近年急速に研究が進む深層強化学習の手法を取り入れたものとなっている.

3 提案方法

3.1 学習ターゲット

従来の金融市場研究においては将来価格の予測に焦点を当てる文献が主流であるが, イントラデイのトレーディングにおいては「指値注文を提示し約定を待つ」戦略が多く観測され, 予測精度に加えて, 予測値の利用側である注文メッセージの送信タイミングを動的に決定するための発注戦略の重要性も高い.

予測と戦略決定は開発プロセス上分離可能なので前者を前工程, 後者を後工程と捉え, 本研究では後工程である「発注戦略の学習」にフォーカスする. 前工程のアウトプットである予測値に関しては実際の将来リターンにノイズを載せた数値で代用する.

3.2 環境シミュレータ

強化学習においてはエージェントの行動の結果として生ずる「状態遷移」及び「獲得する報酬」を計算するための環境シミュレータが必要となる. この環境シミュレータとして, 実際の板情報と歩み値データを用いてティックデータが変わる毎に注文の約定を判定する. このシミュレーションはエージェント自身の注文が市場に与えるマーケットインパクトは考慮できない性質のものである.

3.3 強化学習アルゴリズム

本研究が想定する取引エージェントは連続値の状態空間と, 有限個で少数（7 個）の行動空間を取る. そこで Deep Q Network（DQN)[5] をベースとした強化学習アルゴリズムを採用する. [5] の研究は画像データが入力値のために CNN を用いているが, 本研究では少数（6 個）の「状態変数」を入力値とし, 全結合型ニューラルネットワークを関数近似器として採用する.

加えて, 現在 Atari のプレイスコアが最も高く SoTA とされる R2D2(2019)[6],Ape-X(2018)[7] の強化学習モデルが取り入れている初代 DQN に対する 3 つの拡張

- Double Q Nework [8]
- Prioritized Experience Replay [9]
- Dueling Network [10]

を適用したものを本研究の強化学習アルゴリズムとする.

3.4 状態

以下の 6 個の状態変数を定義する.[1]

1. 現在ポジション株数/最大保有株数
2. 既存買注文の反対気配との乖離率 (逆数)
3. 既存売注文の反対気配との乖離率 (逆数)
4. Bid-Ask スプレッド
5. 短期（5 秒）リターン予測値
6. 長期（60 秒）リターン予測値

最大保有株数は 100 万円相当の株数（最小取引単位の 100 株に満たない場合は 100 株）で, ショートポジションも取り得るものとしている.

3.5 行動

以下の 7 種類の行動を可能な行動集合とする.

1. 何もしない
2. 最良気配への新規買注文送出/既存買注文訂正
3. 最良気配への新規売注文送出/既存売注文訂正
4. 反対気配への新規買注文送出/既存買注文訂正
5. 反対気配への新規売注文送出/既存売注文訂正
6. 既存買注文の取消
7. 既存売注文の取消

取引エージェントの行動タイミングは各銘柄のティック更新時に, 前回行動から 1 秒以上経過している場合に発注判断を行うものとする (各発注判断を 1 ステップとする）. ある時点において市場に出すことのできる注文は買い, 売りのそれぞれに対して 1 注文のみとし, 最大ポジションを超える新規注文は行わない. 各時点で選ぶことのできない行動は選択肢から排除している.

[1] 状態 2, 状態 3 は既存注文が無い状態の数値を 0（乖離率は∞）として扱うために逆数を取ったものを状態変数としている.

3.6 報酬

1 ステップごとに発生する損益額に加え, ポジションが増大することによるペナルティ項とキャンセル注文送出によるペナルティ項を付加したものを報酬 r(t) として次の式を用いて算出する.

$r(t) = \Delta PnL(t+1) - \alpha(\Delta Pos(t+1))^2 - \beta Fo(t)$

$\Delta PnL(t+1)$: (累積) 損益額の変化幅（円)

$\Delta Pos(t+1)$: ポジション金額の変化幅 (円)

$$Fo(t) = \begin{cases} 1 & (cancel\ order\ was\ submitted\ at\ t) \\ 0 & (otherwise) \end{cases}$$

損益額のみを報酬とする場合, ポジションが必要以上に増加する傾向にあり [3], また不必要な注文の送信, 取消を繰り返してしまう. そこで過度なポジショニングと注文取消を抑制するために調整項を加えている.

3.7 ニューラルネット

本研究で関数近似器として用いるニューラルネットの入力層ユニット数は強化学習の「状態変数」の数の 6 , 出力層ユニット数は強化学習の「行動」の数の 7 となる.

Dueling Network[10] のアイディアを取り入れ,Q(s,a) を状態価値関数 Value(=V(s)) と行動による価値の増分 Advantage(=A(s,a)) に分解する.

$Q(s,a) = V(s) + A(s,a) - 1/n\Sigma^{n}_{a'=1}A(s,a')$

n : 行動の数 $(= 7)$

この分解で全てのサンプルデータを状態価値関数の学習に用いることができるようになり学習が安定する.

Dueling Network を実装するために 2 層の中間層を設定する (図 1). 第 1 中間層は全結合層とし, ユニット数は 20 とする. 第 2 中間層は V(s) の計算用と A(s,a) の計算用に分岐し, それぞれユニット数 10 に設定する. 活性化関数には $\tanh x = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$ を採用する.

3.8 パラメータの学習

行動価値関数 Q の値が最大となる行動のみを選択する評価用エージェント（Greedy 方策）1 体と, 確率 ε で取り得る行動からランダムに選択する探索用エージェント（ε - Greedy 方策）を 5 体用意する. [2]

探索用エージェント i の行動履歴 $Experience_i[s,a,r,s']$ は Replay Buffer に蓄えられ, 1 日（1 エピソード）の終了後にバックプロパゲーションによりニューラルネット・パラメータ（ウェイト）の学習が行われる. [3] 次式で定義される TD 誤差 $\delta(\theta)$ の二乗和を減少させる方向に SGD でパラメータ θ の学習が行われる.

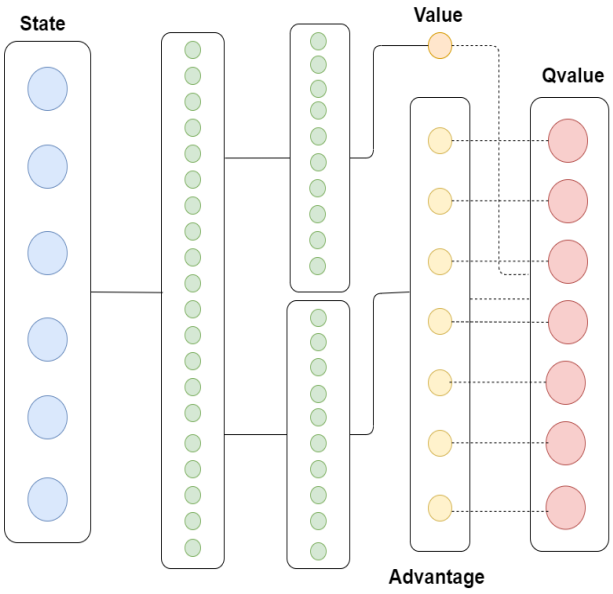

図 1: 本研究の関数近似器である Dueling Network.

$\delta(\theta) = (r + \gamma\, Q(s', a^*|\theta^*)) - Q(s,a|\theta)$

$a^* = argmax_{a'}\ Q(s', a'|\theta)$

r : 報酬

γ : 割引率 (0.999)

s, a : 状態, 行動

s', a' : 次ステップ（遷移先）の状態, 行動

θ : ニューラルネット・パラメータ

θ^* : 過去の時点のニューラルネット・パラメータ

$\theta \leftarrow \theta - \alpha\nabla_\theta\delta(\theta)^2/2 = \theta + \alpha\ \delta(\theta)\nabla_\theta Q(s,a|\theta)$

α : 学習率

この学習に用いるデータ (Experience) は Replay Buffer からランダムに取り出されるが, 本研究では Schaul(2016)[9] に倣い各 Experience の TD 誤差の絶対値をサプライズの大きさと捉え, 選択される確率を TD 誤差の絶対値に比例するように設定している.

学習の対象となる期間全体を 1 エポックとして日付順にシミュレーションと学習を行う. 各 Experience は次エポックの当該営業日に Replay Buffer から除去される. すなわち第 2 エポック以降は常に直近の 1 エポック分の記憶から学習データがサンプリングされることとなる. 平均を取ると 1 つの Experience が 8 回程度学習用にピックアップされるように SGD の繰り返し回数を調整している.

[2] 本研究はシングルエージェントシミュレーションで各エージェントの取引は他のエージェントが参照する板情報に影響を与えることは無い.

[3] 学習はエージェント毎ではなく, 銘柄毎に単独の学習器が全ての探索用エージェントの行動履歴を利用して行う.

4 実験と考察

4.1 データセットと取引対象

データは東京証券取引所のヒストリカルデータ（Flex Full) を利用し, 上下 10 本の板情報を再現する. 取引対象は東京証券取引所に上場する主要企業から 5 銘柄（武田製薬 (4502), ソニー (6758), トヨタ自動車 (7203), 三菱 UFJ フィナンシャルグループ (8306), ソフトバンクグループ (9984)）を選定する.

4.2 期間及び取引時間

学習期間は 2018 年 7〜12 月の 124 営業日を 1 エポックとして 20 エポック繰り返して学習を行う. 学習で得られたニューラルネット・パラメータを用いて 2019 年 1〜6 月のデータでバックテストを行い強化学習で獲得した戦略の収益性を確認する（テスト中と最終エポックはニューラルネット・パラメータの更新は行わない）.

東京証券取引所の取引時間の 9:00〜11:30（前場）,12:30〜15:00（後場）の中で各セッションの開始後 10 分と最後の 5 分は取引を行わず, 翌日にポジションを持越さないように 14:55 に成行でポジションを解消する. [4]

4.3 前提条件と評価方法

発注タイムラグに関してはゼロ（注文送出後次のティック時刻までに自注文が取引所に到達）と仮定している. 取引に要する取引料, アクセス料, 委託手数料, 品貸料等の諸経費は考慮しない.

評価は学習したエージェントの行動選択をグリーディ選択に固定して行う. 取引のシャープレシオは日次ベースで計算し年間平均営業日数 (246) の平方根を掛けて年率換算する.

予測精度がどの程度必要かを把握するために将来リターン予測値に載せるノイズの大きさを変えて評価を行う. ノイズは (-N, N) の一様分布とし, 区間長 N を調整することでノイズの大きさを調整する. 実際のノイズの大きさは短期, 長期のそれぞれの将来リターンの標準偏差を参考に 2N(bp)，5N(bp) としている.

4.4 学習過程

図 2 はソフトバンク株式（9984) に関して, ノイズ N=0（完全予見ベース）とした時の学習の様子を確認したものである. 横軸は経過エピソード数, 縦軸は各エピソードの損益と累計報酬を基に直近 1 周分（124 エピソード）のデータからシャープレシオを算出している.

[4]特別気配が出ている時間やセッションをまたぐ状態遷移は学習対象から外している.

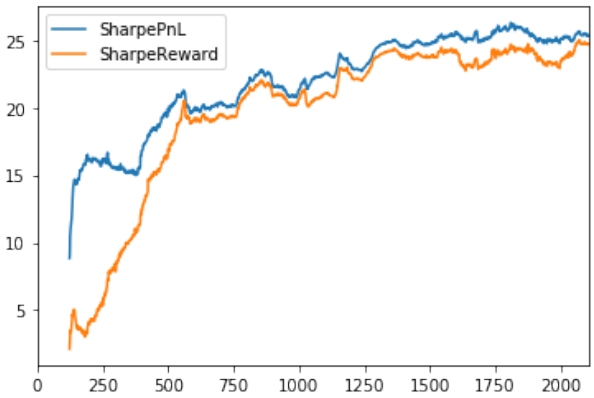

図 2: 学習過程（9984・完全予見ベース）

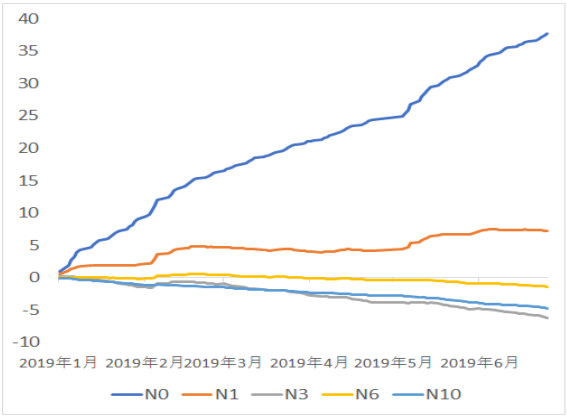

図 3: バックテスト期間中の累積損益

4.5 実験結果

学習期間で得られたニューラルネット・パラメータを用いてテスト期間でバックテストを行った結果を図 3, 表 1 に示す. 表 1 中の数値は 6 か月間 (117 営業日）の平均値となっている. また, 銘柄ごとにシャープレシオを算出した表を表 2 に示す.

4.6 考察

完全予見の設定ではエピソードを重ねるごとに学習が進んでおりアウトオブサンプルのテストセットにおいても収益性の高い行動選択が可能であった. 一方で, ノイズを付加していくにつれて戦略の構築に失敗する銘柄の割合が増加し, それに伴いポートフォリオ全体の収益性も低下する. ノイズ率とパフォーマンスの関係が逆転するケース (4502）も観測され, 学習が安定していない可能性が示唆される.

表 1: バックテスト結果.

ノイズ N	売買代金（百万円）	発注件数（件）	損益（円）	シャープレシオ
0	3,104	6,752	322,045	27.21
1	6,591	13,398	61,411	6.59
3	4,889	9,452	-53,556	-10.53
6	1,562	2,979	-11,965	-4.23
10	1,374	2,090	-40,500	-27.30

表 2: 銘柄ごとのシャープレシオ.

ノイズ	4502	6758	7203	8306	9984
0	21.18	20.44	19.37	29.03	17.60
1	-6.62	12.64	-29.34	-20.70	15.38
3	16.25	-8.68	-18.90	-70.88	7.97
6	-17.78	-18.38	-21.19	-12.00	9.81
10	-2.33	-17.38	-20.22	-40.16	-9.20

5 むすびと今後の展望

本研究では関数近似器にニューラルネットワークを用いた強化学習により日本株式の高頻度取引戦略の構築を試みた.

高精度の将来予測値を利用可能な状況化においては有効な取引戦略を獲得できることが確認できたが, 予測値のノイズレシオを増加させた時は収益性の低下が見られ, 現実的な予測精度の環境において適用するためには強化学習アルゴリズムの細部を調整する必要があると考えている.

また現在の 5 銘柄,6 エージェント,20 エポックの強化学習シミュレーションにおいて, ワークステーションのシングルスレッドで 1 日以上の学習時間を要すことから, 学習をスケールさせるために今後は並列計算に取り組むことを予定している.

参考文献

[1] Nicholas T. Chan and Christian R. Shelton. 2001. An Electronic Market-Maker. AI Memo 2001-005. MIT AI Lab.

[2] Yuriy Nevmyvaka et al.: Reinforcement Learning for Optimized Trade Execution. Proceedings of the 23 rd International Conference on Machine Learning, Pittsburgh, PA, (2006)

[3] Thomas Spooner et al.: Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. Andre, S. Koenig (eds.), July 10-15, 2018, Stockholm, Sweden.

[4] Ganesh, Prakhar & Rakheja, Puneet. (2018). Deep Reinforcement Learning in High Frequency Trading.

[5] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. Nature, 518(7540):529, 2015.

[6] Kapturowski, Steven, Georg Ostrovski, John Quan, Remi Munos, and Will Dabney. "Recurrent experience replay in distributed reinforcement learning." (2018).

[7] Horgan, Dan, John Quan, David Budden, Gabriel Barth-Maron, Matteo Hessel, Hado Van Hasselt, and David Silver. "Distributed prioritized experience replay." arXiv preprint arXiv:1803.00933 (2018).

[8] Van Hasselt, Hado, Arthur Guez, and David Silver. "Deep reinforcement learning with double q-learning." In Thirtieth AAAI conference on artificial intelligence. 2016.

[9] Schaul, Tom, John Quan, Ioannis Antonoglou, and David Silver. "Prioritized experience replay." arXiv preprint arXiv:1511.05952 (2015).

[10] Wang, Ziyu, et al. "Dueling network architectures for deep reinforcement learning." arXiv preprint arXiv:1511.06581 (2015).