1D4-01

株式取引エージェントへの強化学習の応用

A Reinforcement Learning Agent for Stock Trading

松井 藤五郎　大和田 勇人
Tohgoroh Matsui　Hayato Ohwada

東京理科大学 理工学部 経営工学科
Department of Industrial Administration, Faculty of Science and Technology, Tokyo University of Science

This paper describes an application of reinforcement learning to stock-trading agent. Since the tasks in reinforcement learning strongly depend on the states, actions, and rewards, we show how to design them for Kaburobo which is a programming contest for stock-trading software agents. We then describe how to modify on-line profit-sharing (OnPS), which is an reinforcement learning algorithm, for Kaburobo.

1. はじめに

強化学習 [5] は試行錯誤に基づいた機械学習の手法であり，自律型エージェントの行動学習に適している．本論文では，強化学習を株式取引エージェントの行動学習に応用する方法について述べる．

強化学習のタスクは，エージェントが置かれうる状態，取りうる行動，そして与えられる報酬によって定義される．強化学習を行う際には，これらをどのように設計するかが非常に重要なポイントとなる．また，一般的な強化学習アルゴリズムはいずれもエージェント中心に設計されており，自由に環境と相互作用できることを前提としている．しかし，実際には，環境との相互作用には様々な制約があってエージェントの自由にはならないため，強化学習エージェントを実装する際にその制約に基づいてアルゴリズムを修正しなければならない．

そこで，本論文では，株式取引を行うソフトウェア・プログラミング・コンテストである**カブロボ・プログラミング・コンテスト** [6] の仮想証券会社との取引を対象として，強化学習タスクの設計方法と**オンライン型 profit sharing**（**OnPS**）の実装方法を示す．

ソフトウェアによる競技大会へ強化学習を応用する研究としては，Stone らの RoboCup サッカーへの適用例がある [3, 4]. 彼らは，サッカー環境シミュレータと相互作用して強化学習を行うために，強化学習のタスクの設計方法とサッカー環境シミュレータの仕様に合わせた Sarsa(λ) アルゴリズムの実装方法を示している．しかしながら，RoboCup のサッカー環境シミュレータは実際の環境をかなり簡素化しているにも関わらず，強化学習を適用するのが困難なため，彼らはキープ・アウェイ[*1]と呼ばれる独自の部分問題を定義し，この問題に強化学習を適用している．

このように，強化学習の実世界問題への応用には課題が山積しているのが現状である．本研究は，カブロボ・プログラミング・コンテストに参加する強化学習エージェントを作成することにより，トイ・プロブレムでないタスクへ強化学習を応用する際の課題を明らかにし，その解決方法を探ることを目的としている．

連絡先: 松井 藤五郎 (matsui@ia.noda.tus.ac.jp)

∗1　二人の敵プレイヤにボールを奪われないようにしながら，できる限り長く三人でパスを回すという問題．

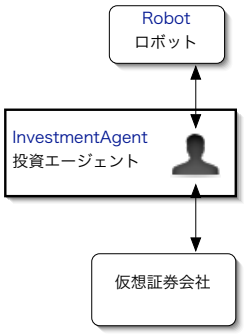

図 1: カブロボにおけるロボットと仮想証券会社の関係.

2. カブロボ・コンテストの概要

カブロボ・プログラミング・コンテスト（略称：**カブロボ・コンテスト**．以下，**カブロボ**）[6] は，株式取引を対象としたソフトウェア・プログラミング・コンテストである．2005 年 1 月に第 1 回のコンテストが開催され，2,400 を超えるチームが参加した．

コンテストの参加者は，株式取引を行う**ソフトウェア・ロボット**（エージェント）を作成する．ロボットは，日経 225 銘柄の中から主催者が選んだ 40 銘柄[*2]を対象として**仮想証券会社**と取引し，所持金 1,000 万円からの運用成績を競う．

カブロボにおけるロボットと仮想証券会社との関係を図 1 に示す．参加者が作成したロボットは，主催者が用意する**投資エージェント**を介して仮想証券会社との取引を行う．具体的には，株価や銘柄に関する情報はすべて投資エージェントから取得し，その情報に基づいてロボットが注文を決定し，投資エージェントに注文を依頼する．そこで，参加者は，投資エージェントから提供される情報に基づいて注文を決定するプログラムを，主催者から提供されている Java API を用いて作成しなければならない．また，コンテスト期間中に外部からロボットに命令することができないため，ロボットは**完全自律型**でなければならない．ただし，主催者が用意したロボットのパラメータを変更しただけのロボットでもコンテストに参加でき，プログラミング初心者でも参加できるように配慮されている．

∗2　参加者には知らされない．

3. カブロボにおける強化学習タスクの設計

強化学習の**タスク**は，エージェントが置かれうる**状態**，取りうる**行動**，そして与えられる**報酬**によって定義される．したがって，強化学習を適切に行うには状態・行動・報酬を適切に設計しなければならない．本研究では，カブロボのルールや仕様を考慮して，次のように設計した．

状態を表す変数としては，各銘柄について，**始値・終値・最高値・最安値**などの値があり，これに加えて**株価収益率**（PER）・**株価純資産倍率**（PBR）・**純資産利益率**（ROA）・**株主資本利益率**（ROE）などの指標がある．また，**出来高**など市場全体の状況を表す指標や**移動平均**などテクニカル指標も利用可能である．このように，利用可能なデータは非常に多くあり，それらをすべて使用して状態を表現することは困難である．

行動については，各銘柄について**買い・売り**という 2 種類の行動を取ることができ，それぞれに**成り行き・指し値**という 2 種類の注文方法がある．成り行き注文が市場の終値で売買を行うのに対し，指し値注文は取引価格を指定して売買を行う．注文する際には量を決める必要があるため，行動は自ずと連続的なパラメータを持つ．さらに，指し値注文の場合は取引価格も決めなければならない．このように，銘柄・売買・量・価格の組み合わせによる注文パターンは非常に多く，それらをすべて行動とすることは困難である．

そこで，本研究では，関連のある 2 銘柄の関係に着目して売買する**ペア取引**を行う．ペア取引は，ヘッジ・ファンドなども用いている基本的な投資手法の一つである．1 組だけのペア取引，かつ，一定量の成り行き注文だけを行うことによって，**行動**を「買い」と「売り」だけにすることができる．本研究では，これに「様子見」を加えた 3 種類としている．取引する銘柄は，ROE が最も高い銘柄とその同業で 2 番目に ROE が高い銘柄とした[*3]．

ペア取引では主に価格差（スプレッド）に着目したスプレッド取引が用いられるが，本研究では市場全体の価格変動に影響を受けない価格比（レシオ）に着目した**レシオ取引**を提案する．銘柄 *main* と銘柄 *sub* のレシオ $ratio(main, sub)$ は次のように定義される．

$$ratio(main, sub) = \frac{price(sub)}{price(main)} \tag{1}$$

ここで，$price(s)$ は銘柄 s の株価を表す．レシオ $ratio(main, sub)$ が大きくなると予測されるときは *sub* を買い *main* を売り，逆に価格比が小さくなると予測されるときは *main* を買い *sub* を売る．

本研究では，次の関数を用いて式 1 で定義したレシオを補正している．

$$\overline{x} = \begin{cases} x - 1 & \text{if } x \leq 1 \\ 1 - \dfrac{1}{x} & \text{otherwise.} \end{cases}$$

これにより，$\overline{ratio}$ は任意の銘柄 s, s' に対して次の特徴を持つ．

$$\overline{ratio(s, s')} = -\overline{ratio(s', s)} \tag{2}$$

$$-1 \leq \overline{ratio(s, s')} \leq 1 \tag{3}$$

式 2 がスプレッドと共通の特徴であるのに対し，式 3 はレシオだけが持つ特徴である．

本研究では，レシオのトレンドを分析するために，レシオの**割引ゴールデン・クロス**を用いる．割引ゴールデン・クロスの値は，ゴールデン・クロス[*4]になると 1，デッド・クロス[*4]になると -1 となり，その後 1 ステップごとに割引率 γ が乗じられて減衰する．レシオと割引ゴールデン・クロスの値は，格子状に配置した動径基底関数（RBF）を用いた関数近似 [5] によって表す．

株式取引エージェントの目的は運用成績を最大化することであり，運用成績は保有株式をすべて現金化したときの額と所持金の合計額である**評価額**によって決定される．したがって，**報酬**は評価額に基づいて決定することが望ましい．そこで，本研究では，評価額の前日比を求め，シグモイド関数を適用した -1 から 1 の値を報酬とした．

*3 複数の銘柄をもつ業種に限定している．

4. カブロボに強化学習を応用する際の問題点

強化学習を応用するにあたって，次のようなカブロボの仕様が問題となる．

- **プログラムが 1 日 1 回実行・終了される**
- **ファイルの入出力ができない**
- **学習に使用できるデータが少ない**

カブロボでは，参加者が作成したプログラムが 1 日 1 回実行される．しかしながら，通常の強化学習アルゴリズムは，1 ステップごとにプログラムが終了することを想定していない．したがって，1 回の実行につき 1 ステップ分だけ処理するようにアルゴリズムを修正する必要がある．

また，1 ステップごとにプログラムが終了するため，強化学習エージェントが内部で保持しているパラメータの書き出しと読み込みが必要となる．しかしながら，カブロボではファイル入出力を禁止しており，その替わりに Memo と呼ばれる最大 100 KB のテキスト・オブジェクトを用意している．強化学習では状態数や行動数が大きくなると内部パラメータの数も大きくなるため，状態数と行動数を小さくしなければならない．

一般に，強化学習には非常に多くの試行錯誤が必要であるが，カブロボが公式に用意しているデータは 3ヶ月分と非常に少ない．したがって，少ないステップ数で学習できる強化学習アルゴリズムを用いなければならない．

そこで，本研究では，筆者が提案した**オンライン型 profit sharing**（**OnPS**） [1, 2] を用いる．OnPS は行動優先度学習型のアルゴリズムであり，Sarsa(λ) など行動価値推定型のアルゴリズムに比べて早く学習できるという特徴を持っている．また，従来の（オフライン型）profit sharing は目標状態が定義可能なエピソード型タスクにしか適用できないが，OnPS は今回のような非エピソード型タスクにも適用可能である．これらの点で，OnPS はカブロボへの応用に適している．

5. 強化学習を用いた株式取引エージェント

上で述べた手法に基づき，本研究では，**強化学習エージェント**（ReinforcementLearningAgent）と**ペア取引エージェント**（PairTradingAgent）を実装した．これらのエージェントの関係を図 2 に示す．株式に関する情報の分析はペア取引エージェントが行い，強化学習エージェントは注文を決定する．1 日に 1 回実行されるという仕様に合わせて，OnPS アルゴリズムを図 3 のように修正した．

まず，プログラムが毎ステップ起動・終了され，内部パラメータを保持することができないことから，内部パラメータ $\vec{\theta}$,

*4 短期的な移動平均線が長期的な移動平均線を越えるときの交点を**ゴールデン・クロス**といい，その逆の交点を**デッド・クロス**という．

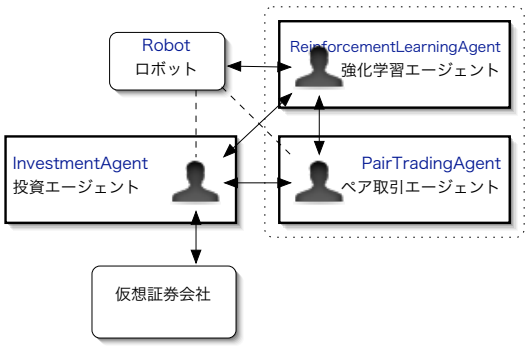

図 2: 本研究で実装したロボットと各種エージェントの関係.

1. もし Memo が空ならば
   すべての i（$i = 1, \ldots, n$）について：
   $\theta(i) \leftarrow \frac{1}{|\mathcal{A}||\mathcal{T}|}$
   $c(i) \leftarrow 0$
   そうでないなら，Memo から $\vec{\theta}$ と $\vec{c}$ を読み込む
2. 前日の行動（注文）に対する報酬 r を観測する
3. $\vec{\theta} \leftarrow \vec{\theta} + \alpha r \vec{c}$
4. $\vec{c} \leftarrow \gamma \vec{c}$
5. 状態 s を観測する
6. すべての i（$i = 1, \ldots, n$）について：
   $\phi_{s,a}(i) \leftarrow \exp\left(-\frac{||s - o_{a,i}||^2}{2\sigma_{a,i}{}^2}\right)$
7. $P(s,a) \leftarrow \sum_{i=1}^{n} \theta(i)\phi_{s,a}(i)$
8. P から導かれる確率分布に従って s での行動 a を選択する
9. すべての i（$i = 1, \ldots, n$）について：
   $c(i) \leftarrow c(i) + \phi_{s,a}(i)$
10. 行動 a を取る（注文する）
11. $\vec{\theta}$ と $\vec{c}$ を Memo に書き込む

図 3: カブロボ用 OnPS アルゴリズム．ここで，n は特徴数，$\mathcal{A}$ は行動の集合，$\mathcal{T}$ は格子の集合，α はステップ・サイズ・パラメータ，γ は割引率パラメータ，$o_{a,i}$ と $\sigma_{a,i}$ はそれぞれ動径基底関数 i の中心と幅を表す.

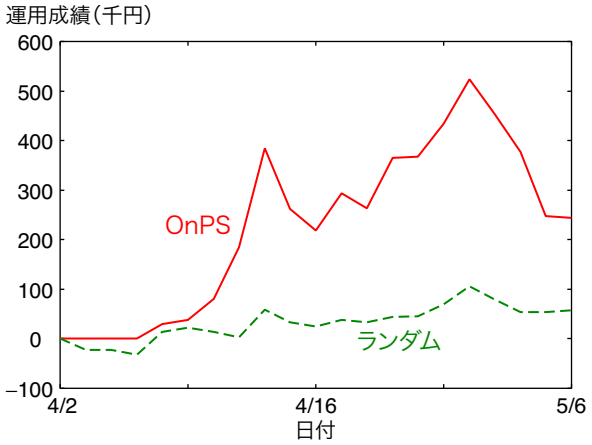

図 4: 実行例.

$\vec{c}$ をすべて Memo へ書き出し，実行開始時に読み込むようにした[*5]（1, 11 行目）.

また，k 日目の状態行動対 s_k, a_k に対するパラメータの更新に必要な報酬 r_{k+1} は翌日にならないと観測できず，k 日目のうちにパラメータを更新することができない．そこで，本アルゴリズムでは，その日の状態 s_k を観測する前に報酬 r_k を求め，前日までの状態行動対に対する内部パラメータの更新を行う（2–3 行目）.

次に，強化学習エージェントはペア取引エージェントから情報を受け取り，状態 s_k を観測する（5 行目）．行動 a_k の選択には Gibbs 分布によるソフト・マックス選択を用い，学習効果を大きく反映させるため温度パラメータを $\tau = 0.1$ とし，わずかな優先度の違いでも行動選択確率が大きく変わるようにした（8 行目）．また，ステップ・サイズは重ねた格子の枚数から $\alpha = 0.1$ とし，割引率は早く学習するよう $\gamma = 0.9$ と比較的大きい値にした.

6. 実行例

本論文で提案した強化学習エージェントの効果を確認するため，カブロボ・シミュレータを用いて簡単な実験を行った．特徴量 $\vec{\phi}$ を決める動径基底関数は，タイル・コーディングと同様に格子状に配置した．幅が 0.25 である 9×9 の格子を 10 枚用意し，原点を基点として幅内でランダムにずらして重ねて配置した[*6].

実験では，2004 年 4 月のデータを用いて一ヶ月間の運用成績を求め，一様なランダムに行動を選択した場合と比較した．2004 年 4 月は「株価の方向性がない相場」であり，日経平均株価の始値と終値の差はほとんどない．エージェントが選択した銘柄はイオンとセブン-イレブン・ジャパンだった．ここでは，売買の量は 100 万円とした.

結果を図 4 に示す．強化学習エージェントの運用成績は 10 回繰り返し学習したときの 10 回目のものである．グラフは乱数シードを変えて 10 回行った実験の中央値を表している．グラフに示された通り，強化学習エージェントはランダムに売買するよりも良い運用成績を示した.

*5 このとき，double 型の値を文字列に変換し再び double 型の値に変換すると誤差が生じるが，本研究ではまだ具体的な対策を講じていない.

*6 同じ状態空間を表現するために，このときのランダム・シードには毎回同じものを用いている.

7. おわりに

本論文では，カブロボ・プログラミング・コンテストにおいて株式取引を行うソフトウェア・エージェントに強化学習を適用する際の問題点を挙げ，強化学習タスクの設計方法とオンライン型 profit sharing（OnPS）の実装方法を示した．また，実験により本論文で示した強化学習エージェントが株式取引の運用成績を伸ばすことができることを確認した．

今後は，実際のコンテストの環境に近づけるために，学習に用いた期間とは異なる期間に運用したときの運用成績を評価する必要がある．また，今年の秋に予定されている第 2 回コンテストに参加し，その結果を分析したい．

参考文献

[1] 松井藤五郎, 犬塚信博, 世木博久. 線形関数近似を用いた profit sharing 強化学習法. 2002 年度人工知能学会全国大会（第 16 回）論文集, pp. 2D3–03, 2002.

[2] Tohgoroh Matsui, Nobuhiro Inuzuka, and Hirohisa Seki. Online profit sharing works efficiently. In V. Palate, R. J. Howlett, and L. Jain, editors, *7th International Conference on Knowledge-Based Intelligent Information & Engineering Systems*, Vol. 2773 of *Lecture Notes in Artificial Intelligence*, pp. 317–324, 2003.

[3] Peter Stone and Richard S. Sutton. Scaling reinforcement learning toward RoboCup soccer. In *Proceedings of the 18th International Conference on Machine Learning (ICML-01)*, pp. 537–544. Morgan Kaufmann Publishers, 2001.

[4] Peter Stone, Richard S. Sutton, and Gregory Kuhlmann. Reinforcement learning for RoboCup-soccer keepaway. *Adaptive Behavior*, 2005. To appear.

[5] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 1998. 三上貞芳, 皆川雅章 共訳. 強化学習. 森北出版, 2000.

[6] カブロボ・コンテスト, 2004. http://kaburobo.jp/.