知能と複雑系　125－5
（2001. 7. 23）

マルコフゲームにおける環境モデルの推定を利用したマルチエージェント強化学習法

長行 康男　　伊藤 実

奈良先端科学技術大学院大学
〒630-0101 奈良県生駒市高山町 8916-5
E-mail : {yasuo-n, ito}@is.aist-nara.ac.jp

あらまし

本稿では，マルコフゲームにおける新たなマルチエージェント強化学習法を提案する．本稿で提案するマルチエージェント強化学習法では，エージェントが，環境モデル（環境内に存在する他エージェントの政策と，環境の状態遷移関数）を推定し，その推定した環境モデルを利用して，（エージェントが）どの行動を実行すればどの環境状態に遷移するかを予測する．そして，その予測した環境状態における価値関数（V 関数）を基に，どの行動を実行すればよいかを決定し，強化学習を進行する．提案したマルチエージェント強化学習法をマルコフゲームの枠組みでモデル化した追跡問題に適用し，実験を行った結果，その有効性が示される．

キーワード：マルチエージェント強化学習，TD 学習，環境モデル，マルコフゲーム，追跡問題

Multi-agent reinforcement learning method for Markov games : An approach based on the estimation of the environmental model

Yasuo Nagayuki　　Minoru Ito

Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-0101, Japan
E-mail : {yasuo-n, ito}@is.aist-nara.ac.jp

Abstract

In this article, we propose a multi-agent reinforcement learning method for Markov games. In our multi-agent reinforcement learning method, each agent infers the environmental model which consists of the other agents' policies and the state transition function, and estimates the future states by using the inferred environmental model. Each agent conducts its reinforcement learning based on the estimated future states. In order to evaluate our multi-agent reinforcement learning method, we employ the variant of the pursuit problem as a task. Through experiments, we demonstrate that our multi-agent reinforcement learning method is effective.

keywords: multi-agent reinforcement learning, TD learning, environmental model, Markov game, pursuit problem

1 まえがき

複数のエージェントが存在する環境（マルチエージェント環境）における，エージェントの適応行動の実現は，工学及び認知科学の観点から興味深い課題である．その中でも，エージェントが思考錯誤の経験を基に適応行動を自律的に獲得するマルチエージェント強化学習の研究が，強化学習 [1] の発展を契機として近年注目を集めている．

マルチエージェント強化学習の一つの接近法として，環境内に存在する他エージェントの政策（行動決定関数）を推定しながら学習を進行する手法が提案されている [2, 3, 4]．これらの3つのマルチエージェント強化学習法は，2体エージェント問題（環境内に存在するエージェント数が2体である問題）において，その有効性が示されている．これらのマルチエージェント強化学習法はすべて，シングルエージェント環境を基に提案された強化学習法である Q 学習 [5] を基盤にしている．そして，学習時に他エージェントの政策（を基に実行される行動）を考慮に入れるため，学習関数である Q 関数を，環境状態 s と行動 a の関数 $Q(s,a)$ から，環境状態 s と全てのエージェントの行動 $a^1,\cdots,a^n$ の関数 $Q(s,a^1,\cdots,a^n)$ に拡張している．しかしながら，このように Q 関数を拡張した場合，学習空間は（すべてのエージェントの行動に依存するため）エージェント数の増加に対して指数関数的に増加する．強化学習において，学習空間の指数関数的増加は，『次元の呪い』と呼ばれ，学習を大幅に遅らせる原因となる．したがって，学習関数を上記のように拡張することは，あまり相応しいことではない．

本稿では，エージェントの行動に依存しない学習関数を用いた，他エージェントの政策推定を利用した新たなマルチエージェント強化学習法を提案する．本稿で提案するマルチエージェント強化学習法は，TD 学習 [6] を基盤にしたもので，学習関数として，Q 関数ではなく，V 関数 $V(s)$ を用いる．

本研究では，マルチエージェント環境として，先行研究 [2, 3, 4] と同様，マルコフゲームの枠組みを採用する．マルコフゲームの枠組みでモデル化した追跡問題 [7] に，本稿で提案するマルチエージェント強化学習法を適用し，実験を行った結果，Q 学習に基づいた手法 [3] より高速に学習が行われることが示された．

2 マルコフゲーム

マルコフゲームは，マルチエージェント環境における行動決定問題のモデルで，以下の組で定義される．

$$\langle n, S, A^1,\cdots,A^n, T, R^1,\cdots,R^n\rangle \tag{1}$$

ここで，n は環境内に存在するエージェントの数，S は環境状態の有限集合，A^k（$k=1,\cdots,n$）はエージェント k の行動の有限集合，T は環境状態の遷移関数，R^k はエージェント k の報酬関数である．

各離散時間ステップ $t=0,1,2,\cdots$ において，すべてのエージェントは，現在の環境状態 $s_t\in S$ を観測し，行動 $a_t^k\in A^k$ を実行する．そして，環境状態は $s_{t+1}\in S$ に遷移し，環境から直接報酬 r_{t+1}^k を受け取る．環境状態の遷移は状態遷移関数 T に従う．この状態遷移関数は時不変な状態遷移確率

$$T(s,a^1,\cdots,a^n,s') = \Pr(s_{t+1}=s'|s_t=s,a_t^1=a^1,\cdots,a_t^n=a^n) \tag{2}$$

の集合で表される．ここで，$\Pr(s'|s,a^1,\cdots,a^n)$ は，環境状態 s でそれぞれのエージェントが行動 $a^1,\cdots,a^n$ を実行したときに環境状態が s' へ遷移する確率を表す．エージェントが環境から受け取る直接報酬 r_{t+1}^k も確率的で，その期待値は報酬関数

$$R^k(s,a^1,\cdots,a^n) = E\{r_{t+1}^k=r^k|s_t=s,a_t^1=a^1,\cdots,a_t^n=a^n\} \tag{3}$$

で表される．ここで，$E\{r^k|s,a^1,\cdots,a^n\}$ は，環境状態 s でそれぞれのエージェントが行動 $a^1,\cdots,a^n$ を実行したときにエージェント k が受け取る直接報酬 r^k の期待値を表す．

マルコフゲームにおいて，個々のエージェントの目的は，式 (4) で表される関数を最大にするような政策 π^k を見つけることである．

$$V^{\pi^k}(s) = \sum_{n=0}^{\infty} E\left\{\gamma^n r_{t+n+1}^k|s_t=s,\pi^1,\cdots,\pi^n\right\} \tag{4}$$

ここで，政策 π^k は各環境状態において各行動を選択する確率への写像，$\pi^k: S\times A^k\to[0,1]$ を表す．$\gamma\in[0,1]$ は割引率と呼ばれるパラメータである．$\pi^1,\cdots,\pi^n$ は，それぞれエージェント $1,\cdots,n$ の政策を表す．式 (4) の右辺は，それぞれのエージェントが時刻 t 以降の行動を政策 $\pi^1,\cdots,\pi^n$ に従って選択したときに，エージェント k が受け取る割引報酬の和の期待値を表す．$V^k(=V^{\pi^k})$ はエージェント k の V 関数と呼ばれる．

3 マルチエージェント強化学習

本稿では，マルコフゲームにおける新たなマルチエージェント強化学習法を提案する．

上述したように，マルコフゲームにおける個々のエージェントの目的は，式(4)で表されるV関数を最大にするような政策を見つけることである．ここで，個々のエージェントのV関数は，他エージェント（環境内に存在する自分以外のエージェント）の政策に依存していることに注意する．この点に着目したマルチエージェント強化学習法がいくつか報告されている[2, 3, 4]．これらのマルチエージェント強化学習法では，個々のエージェントが，他エージェントの政策を推定し，その推定した政策を利用して他エージェントの未来の行動を予測する．そして，その予測行動を基に行動選択を行い，強化学習を進行する．これら3つのマルチエージェント強化学習法は，2体エージェントマルコフゲーム（式(1)において，$n=2$ であるマルコフゲーム）において，その有効性が示されている．これら3つのマルチエージェント強化学習法はすべて，シングルエージェント環境を基に提案されたQ学習[5]を基盤にしている．そして，学習時に他エージェントの政策（を基に実行される行動）を考慮に入れるため，学習関数であるQ関数を，環境状態 s と行動 a の関数 $Q(s,a)$ から，環境状態 s と全てのエージェントの行動 $a^1,\cdots,a^n$ の関数 $Q(s,a^1,\cdots,a^n)$ に拡張している．この拡張した学習関数 $Q(s,a^1,\cdots,a^n)$ は，環境内に存在する全てのエージェントの行動に依存しており，学習空間の数は，

$$|S|\times|A^1|\times\cdots\times|A^n| \tag{5}$$

となる．ここで，$|S|,|A^1|,\cdots,|A^n|$ は，それぞれ集合 $S,A^1,\cdots,A^n$ の要素数を表す．式(5)は，Q関数を $Q(s,a^1,\cdots,a^n)$ に拡張することにより，学習空間がエージェント数の増加に対して指数関数的に増加することを示している．強化学習において，学習空間の指数関数的増加は，『次元の呪い』と呼ばれ，学習を大幅に遅らせる原因となる．したがって，学習関数を上記のように拡張することは，あまり相応しいことではない．

本稿では，エージェントの行動に依存しない学習関数を用いた，他エージェントの政策推定を利用した新たなマルチエージェント強化学習法を提案する．本稿で提案するマルチエージェント強化学習法は，TD学習[6]を基盤にしたもので，学習関数としてV関数 $V(s)$ を用いる．ここで，学習空間の数は，エージェント数に関わらず，$|S|$ である．

本稿で提案するマルチエージェント強化学習法では，他エージェントの政策推定に加えて，状態遷移関数 T の推定を行う（本稿では，他エージェントの政策と，状態遷移関数を合わせて『環境モデル』と呼ぶことにする）．そして，推定した環境モデルを基に，自分（エージェント k とする）がどの行動を実行すれば，未来にどの環境状態に遷移するかを式(6)で与えられる $\hat{P}^k$ を利用して予測する．

$$\hat{P}^k(s,a^k,s')=\sum_{a^{o_1}}\cdots\sum_{a^{o_{n-1}}}\hat{\pi}_k^{o_1}(s,a^{o_1})\cdots\hat{\pi}_k^{o_{n-1}}(s,a^{o_{n-1}})\hat{T}^k(s,a^1,\cdots,a^n,s') \tag{6}$$

ここで，$\hat{P}^k(s,a^k,s')$ は，エージェント k の観点から，自分（エージェント k）が環境状態 s で行動 a^k を実行したときに環境状態が s' へ遷移すると予想される確率を表す．$\hat{T}^k(s,a^1,\cdots,a^n,s')$ は，エージェント k が推定した状態遷移関数で，環境状態 s でそれぞれのエージェントが行動 $a^1,\cdots,a^n$ を実行したときに環境状態が s' へ遷移すると予想される確率を表す．$\hat{\pi}_k^{o_1},\cdots,\hat{\pi}_k^{o_{n-1}}$ は，それぞれ，エージェント k が推定した他エージェントの政策を表す．ここで，$o_1,\cdots,o_{n-1}$ は，それぞれ，k 以外のエージェント（他エージェント）のいずれか1体のエージェントに対応するものとする．$\hat{\pi}^o(s,a^o)$（という表記）は，他エージェント o が環境状態 s で行動 a^o を選択すると予想される確率を表す．本稿で提案するマルチエージェント強化学習法では，式(6)の $\hat{P}^k$ で予測した未来の環境状態 s' に対するV関数値 $V^k(s')$ を考慮しながら行動選択を行い，強化学習を進行する．

以下で，本研究で採用する環境モデル推定法と，提案する環境モデル推定を利用したマルチエージェント強化学習法を示す．

3.1 環境モデル推定法

3.1.1 状態遷移関数の推定法

状態遷移関数 T の推定には，式(7)を利用する．

$$\hat{T}^k(s,a^1,\cdots,a^n,s^*)=\frac{c(s,a^1,\cdots,a^n,s^*)}{n(s,a^1,\cdots,a^n)} \tag{7}$$

ここで，$n(s,a^1,\cdots,a^n)$ は，環境状態 s でそれぞれのエージェントが行動 $a^1,\cdots,a^n$ を実行した回数，$c(s,a^1,\cdots,a^n,s^*)$ は，環境状態 s でそれぞれのエージェントが行動 $a^1,\cdots,a^n$ を実行したときに環境状態が s^* へ遷移した回数を表す．

すべての環境状態 s において，すべてのエージェントが，すべての行動を無限回実行した場合，式 (7) の推定法により，$\hat{T}^k$ は T に収束する．

3.1.2 他エージェントの政策の推定法

他エージェントの政策の推定には，我々が以前提案した手法 [3] を採用する．以下に，その推定法を示す．

時刻 t において，他エージェント o が環境状態 s_t で行動 a_t^o を実行したとする．そのとき，環境状態 $s = s_t$ で実行可能な，すべての行動 $a^o \in A^o$ に対して，式 (8) に従って $\hat{\pi}_k^o$ を更新する．

$$\hat{\pi}_k^o(s,a^o) \leftarrow (1-\theta)\hat{\pi}_k^o(s,a^o) + \begin{cases} \theta & (a^o = a_t^o) \\ 0 & (\text{otherwise}) \end{cases} \tag{8}$$

ここで，$\theta \in [0,1]$ は観測した行動を将来の行動予測時にどれくらい考慮するかを決定するパラメータである．式 (8) の更新則によって $\sum_{a^o \in A^o} \hat{\pi}_k^o(s,a^o) = 1$ が保たれることに注意する．本稿では，それぞれのエージェントが他エージェントの行動集合 A^o について予め知っていることを仮定する．式 (8) の推定法が強化学習エージェントの政策の推定に適していることが実験的に示されている [3].

3.2 環境モデルの推定を利用したマルチエージェント強化学習法

以下に，本稿で提案する環境モデルの推定を利用したマルチエージェント強化学習法の流れを示す．

1. 現在（時刻 t とする）の環境状態 $s_t \in S$ において，エージェント k（$k = 1,\cdots,n$）は，式 (9) のボルツマン分布で与えられる政策 π^k に従って行動 a^k を選択する．

$$\pi^k(s_t,a^k) = \frac{e^{J(s_t,a^k)/\tau}}{\sum_{b \in A^k} e^{J(s_t,b)/\tau}} \tag{9}$$

ここで，政策 $\pi^k(s,a^k)$ は環境状態 s で行動 a^k を選択する確率を表す．τ は温度パラメータと呼ばれ，行動選択のランダムさを調整するパラメータである．$J(s,a^k)$ は，エージェント k の V 関数 V^k の P^k（式 (6)）に関する期待値で，式 (10) で与えられる．

$$J(s,a^k) = \sum_{s^*} P^k(s,a^k,s^*)V^k(s^*) \tag{10}$$

2. エージェント k は，手続き 1 で選択した行動 a_t^k を実行する（ここで，他エージェントも同期して行動 $a_t^{o_1},\cdots,a_t^{o_{n-1}}$ を実行する．環境状態は，式 (2) の状態遷移関数に従って s_t から s_{t+1} へ遷移する）．エージェント k は，他エージェントの行動 $a_t^{o_1},\cdots,a_t^{o_{n-1}}$ を観測する．また，環境から直接報酬 r_{t+1}^k を受け取る．エージェント k は，関数 n，関数 c をそれぞれ式 (11)，式 (12) に従って更新する．

$$n(s_t,a_t^1,\cdots,a_t^n) \leftarrow n(s_t,a_t^1,\cdots,a_t^n) + 1 \tag{11}$$

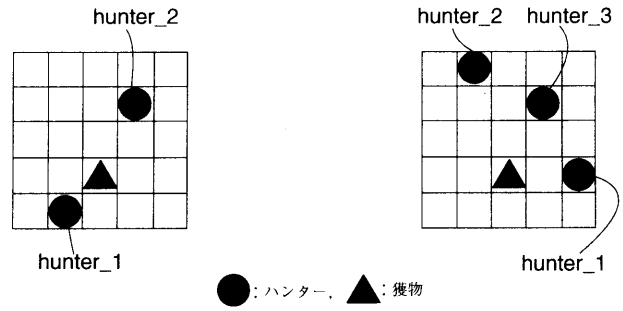
$$c(s_t,a_t^1,\cdots,a_t^n,s_{t+1}) \leftarrow c(s_t,a_t^1,\cdots,a_t^n,s_{t+1}) + 1 \tag{12}$$

エージェント k は，他エージェントの政策 $\hat{\pi}_k^o$（$o = o_1,\cdots,o_{n-1}$）を式 (8) に従って推定（更新）する．そして，環境状態 s_t における V 関数値を式 (13) に従って更新する．

$$V^k(s_t) \leftarrow (1-\alpha)V^k(s_t) + \alpha(r_{t+1}^k + \gamma V^k(s_{t+1})) \tag{13}$$

ここで，$\alpha \in (0,1]$ は学習率と呼ばれるパラメータである．（式 (13) は TD 学習 [6] における更新式と同じものである．）

3. 学習の終了条件を満たしていれば学習終了．そうでなければ t に 1 を加えて，手続き 1 に戻る．



（a）ハンター数が2体の追跡問題　　（b）ハンター数が3体の追跡問題

図 1: 追跡問題のグリッド空間

4 実験

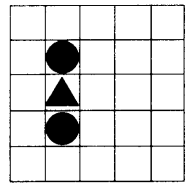
4.1 追跡問題

本研究では，実験に使用するタスクとして追跡問題 [7] を取り上げる．追跡問題は，複数のハンターが獲物を追いかけて捕獲する課題である．以下に，本研究における追跡問題の問題設定を示す．

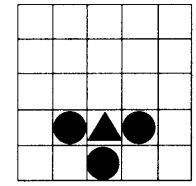
- 2次元（5 × 5）のグリッド空間中に，複数のハンターと1体の獲物が存在する（図 1）．ここで，グリッド空間の上と下，左と右の境界は繋がっているものとする．
- 本研究では，ハンターを『エージェント』と定義する．本研究では，ハンター数が2体の場合（図 1（a））と，3体の場合（図 1（b））の2つの実験を行う．
- 各時間ステップ毎に，ハンターと獲物は，それぞれ1つの行動を同期して実行する．ここで，ハンターが実行可能な行動は，隣接する上下左右のグリッドへ移動する，現在位置に留まる，の5通りとする．また，獲物が実行可能な行動は，隣接する右のグリッドへ移動する，現在位置に留まる，の2通りとする．
- ハンターの目標は獲物を捕獲することである．ここで，捕獲の定義は，
  - ハンター数が2体の実験では，『2体のハンターが獲物を上下，あるいは左右から挟んだ状態』（図 2（a））
  - ハンター数が3体の実験では，『獲物に隣接する4近傍のグリッドうち，いずれか3つのグリッドを3体のハンターが占有している状態』（図 2（b））

  とする．
- 初期配置から獲物が捕獲されるまでを『1エピソード』とする．獲物が捕獲されると，ハンターと獲物はグリッド空間中にランダムに初期配置され，新たなエピソードを開始する．
- 環境状態は，それぞれのハンターと獲物の相対位置の組合せ $s = (p^1, \cdots, p^n)$ とする．ここで，p^i $(i = 1, \cdots, n)$ は hunter_i と獲物の相対位置を表す．例えば，図1（a）では $s = ([1,1],[-1,-2])$，図1（b）では $s = ([-2,0],[1,2],[-1,-2])$ である．
- 獲物は学習を行なわず，2通りの行動の中から1つの行動を確率的に選択する．実験では，右へ移動する確率を $\frac{2}{3}$，現在位置に留まる確率を $\frac{1}{3}$ としている．この行動選択確率は時不変とする．

以上の問題設定では，報酬関数について言及していないが，報酬関数が式 (3) を満たすように設定された場合，この追跡問題はマルコフゲームの条件を満たす．



（a）ハンター数が2体の追跡問題



（b）ハンター数が3体の追跡問題

図 2: 捕獲状態の例

4.2 他エージェントの政策推定を利用したマルチエージェント Q 学習法

提案したマルチエージェント強化学習法の性能を評価するための比較対象として，本稿では，他エージェントの政策推定を利用したマルチエージェント Q 学習法 [3]（以下 MAQL と書く）を取り上げる．MAQL は，学習関数として Q 関数 $Q(s,a^1,\cdots,a^n)$ を用いたもので，3章で述べたように，学習空間（$|S|\times|A^1|\times\cdots\times|A^n|$）は，エージェント数の増加に対して指数関数的に増加する．MAQL の学習の流れを以下に示す．

1. 現在（時刻 t とする）の環境状態 $s_t\in S$ において，エージェント k（$k=1,\cdots,n$）は，式 (14) で与えられる政策 π^k に従って行動 a^k を選択する．

$$\pi^k(s_t,a^k)=\frac{e^{\bar{Q}(s_t,a^k)/\tau}}{\sum_{b\in A^k}e^{\bar{Q}(s_t,b)/\tau}} \tag{14}$$

ここで，$\bar{Q}(s,a^k)$ は式 (15) で与えられる関数である．

$$\bar{Q}(s,a^k)=\sum_{a^{o_1}}\cdots\sum_{a^{o_{n-1}}}\hat{\pi}_k^{o_1}(s,a^{o_1})\cdots\hat{\pi}_k^{o_{n-1}}(s,a^{o_{n-1}})Q^k(s,a^1,\cdots,a^n) \tag{15}$$

2. エージェント k は，手続き 1 で選択した行動 a_t^k を実行する（ここで，他エージェントも同期して行動 $a_t^{o_1},\cdots,a_t^{o_{n-1}}$ を実行する．環境状態は，式 (2) の状態遷移関数に従って s_t から s_{t+1} へ遷移する）．エージェント k は，他エージェントの行動 $a_t^{o_1},\cdots,a_t^{o_{n-1}}$ を観測する．また，環境から直接報酬 r_{t+1}^k を受け取る．エージェント k は，他エージェントの政策 $\hat{\pi}_k^o$（$o=o_1,\cdots,o_{n-1}$）を式 (8) に従って推定（更新）する．そして，環境状態 s_t，行動 $a_t^1,\cdots,a_t^n$ における Q 関数値を式 (16) に従って更新する．

$$Q^k(s_t,a_t^1,\cdots,a_t^n)\leftarrow(1-\alpha)Q^k(s_t,a_t^1,\cdots,a_t^n)+\alpha(r_{t+1}^k+\gamma\max_{a^k}\bar{Q}(s_{t+1},a^k)) \tag{16}$$

3. 学習の終了条件を満たしていれば学習終了．そうでなければ t に 1 を加えて，手続き 1 に戻る．

4.3 実験結果

提案した環境モデルの推定を利用したマルチエージェント強化学習法（以下，MATDL と書く）と MAQL を追跡問題に適用した．ハンター数が2体の場合の実験結果を図 3 に，3体の場合の実験結果を図 4 に示す．図の横軸は学習エピソード数，縦軸は 1 エピソード中で獲物捕獲までに費やした平均時間ステップ数を表す．図の結果は，10 学習エピソード毎に，そのときまでの学習性能を評価するため，初期配置を変えた 100 評価エピソード（このエピソードでは学習を行なわない）の実験を行ない，その平均時間ステップ数を示したものである．2つの学習法（MATDL と MAQL）で使用した学習パラメータは $\alpha=0.3\times decay^{num\_ep}$，$\gamma=0.9$，$\tau=0.1\times decay^{num\_ep}$，$\theta=0.5\times decay^{num\_ep}$ である．ここで，$decay$ は減衰係数，num_ep は学習エピソード数を表す．減衰系数 $decay$ は，ハンター数が2体の実験では $decay=0.9977$，3体の実験では $decay=0.99977$ としている．減衰係数 0.9977，0.99977 は，それぞれ $0.9977^{1000}\approx 0.1$，$0.99977^{10000}\approx 0.1$ となるように選ばれた値である．エージェントが環境から受け取る直接報酬 r^k は，獲物捕獲時に $r^k=1.0$，それ以外の時に $r^k=-0.05$ としている．すべての $s\in S$，$s^*\in S$，$a^1\in A^1,\cdots,a^n\in A^n$ に対して，V 関数，Q 関数，関数 n，関数 c のそれぞれの初期値は，$V^k(s)=0.0$，$Q^k(s,a^1,\cdots,a^n)=0.0$，$n(s,a^1,\cdots,a^n)=0$，$c(s,a^1,\cdots,a^n,s^*)=0$ としている．また，すべての $s\in S$，$a^o\in A^o$（$o=o_1,\cdots,o_{n-1}$）に対して，関数 $\hat{\pi}_k^o$ の初期値は $\pi_k^o(s,a^o)=0.2$ としている．

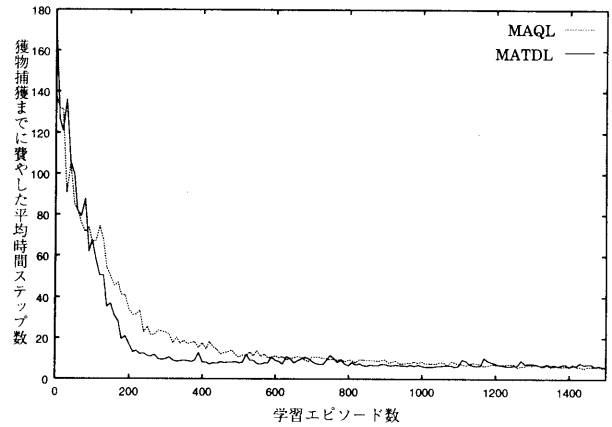


図 3: 獲物捕獲までに費やした平均時間ステップ数：ハンター数が 2 体の場合.

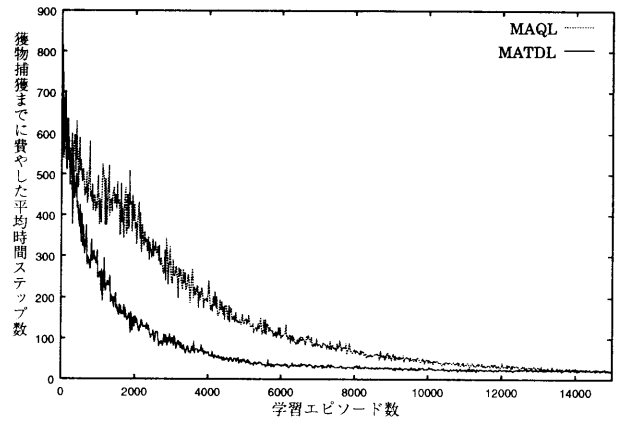


図 4: 獲物捕獲までに費やした平均時間ステップ数：ハンター数が 3 体の場合.

図 3，図 4 の結果は，ハンター数が 2 体の場合，3 体の場合の両方で，MATDL が MAQL よりも学習が速いことを示している．また，獲物捕獲までに費やした平均時間ステップ数の差（MATDL−MAQL）と比率（MATDL:MAQL）は，ハンター数が 3 体の場合の方が，2 体の場合より大きいことを示している．

4.4 考察と今後の課題

2 つの学習法 MATDL と MAQL の主な相違点は以下の 3 つである．

1. 学習関数として，MATDL では V 関数 $V(s)$，MAQL では Q 関数 $Q(s, a^1, \cdots, a^n)$ を用いている．
2. MAQL の学習空間は，ハンター数が 2 体の場合，MATDL の 25（5^2）倍，3 体の場合，MATDL の 125（5^3）倍である．
3. MATDL では状態遷移関数 T を式 (7) により明に推定しているのに対して，MAQL では学習（思考錯誤の経験）を通じて，Q 関数中で暗に推定される．

MATDL が MAQL よりも学習が高速である原因は，これら 3 つの違いが影響していると考えられるが，獲物捕獲までに費やした平均時間ステップ数の比率（MATDL:MAQL）が，ハンター数が 2 体の場合より 3 体の場合の方が大きいこ

とより，項目2の学習空間の増加が大きく寄与していることが予想される．より詳細な調査は今後の課題である．

本稿では，学習空間の指数関数的増加の要因としてエージェントの行動集合 $A^1, \cdots, A^n$ に着目した．しかしながら，一般に，エージェント数の増加に対して $|S|$ も指数関数的に増加する（本稿における追跡問題もその一例である）．したがって，学習の高速化が，主に上記の相違点の項目2に依存している場合は，$\sum_i |A^i| \ll |S|$ であるような問題（タスク）に対して，MATDL による学習の高速化はあまり期待できない．より詳細な調査は今後の課題である．

実験結果より，MATDL は（MAQL も）学習が収束している．MATDL と MAQL の両学習法とも学習の収束性は保証されていない．強化学習において，学習の収束性を保証することは重要である．MATDL と MAQL の学習の収束性の解析は今後の課題である．

5 あとがき

本研究では，マルコフゲームにおける環境モデルの推定を利用したマルチエージェント強化学習法を提案した．提案したマルチエージェント強化学習法では，エージェントが，環境モデル（他エージェントの政策と，状態遷移関数）を推定し，その推定した環境モデルを利用して，どの行動を実行すればどの環境状態に遷移するかを予測した．そして，その予測した環境状態における V 関数値を基に，どの行動を実行すればよいかを決定し，強化学習を進行した．提案したマルチエージェント強化学習法では学習関数として V 関数を採用した．提案したマルチエージェント強化学習法をマルコフゲームの枠組みでモデル化した追跡問題に適用し，実験を行った結果，ハンター数が2体の場合，3体の場合の両方の実験で，他エージェントの政策推定を利用したマルチエージェント Q 学習法 [3] より学習が高速になった．学習が高速になった原因の詳細な調査は今後の課題である．

参考文献

[1] R. S. Sutton, and A. G. Barto, Reinforcement Learning : An Introduction, MIT Press, Cambridge, Massachusetts, 1998.

[2] M. L. Littman, "Markov games as framework for multi-agent reinforcement learning," Proc. 11th International Conference on Machine Learning, pp.157-163, New Brunswick, New Jersey, USA, July 1994.

[3] Y. Nagayuki, S. Ishii, and K. Doya, "Multi-agent reinforcement learning : An approach based on the other agent's internal model," Proc. 4th International Conference on Malti-Agent Systems, pp.215-221, Boston, Massachusetts, USA, July 2000.

[4] J. Hu, and M. P. Wellman, "Multiagent reinforcement learning: theoretical framework and an algorithm," Proc. 15th International Conference on Machine Learning, pp.242-250, Madison, Wisconsin USA, July 1998.

[5] C. J. C. H. Watkins, and P. Dayan, "Technical Note Q-Learning," Machine Learning, vol.8, no.3, pp.279-292, 1992.

[6] R. S. Sutton, "Learning to predict by the methods of temporal differences," Machine Learning, vol.3, pp.9-44, 1988.

[7] M. Benda, V. Jagannathan, and R. Dodhiawalla. "On optimal cooperation of knowledge sources". Technical Report BCS-G2010-28, Boeing AI Center, 1985.