論　文

2体エージェント確率ゲームにおける他エージェントの政策推定を利用した強化学習法

長行　康男[†]　　伊藤　実[†]

A Reinforcement Learning Method with the Inference of the Other Agent's Policy for 2-Player Stochastic Games

Yasuo NAGAYUKI[†] and Minoru ITO[†]

あらまし　本論文では，2体エージェント確率ゲームにおける新たな強化学習法を提案する．提案する手法では，他エージェントが実際に実行した行動の観測情報をもとに他エージェントの政策（行動決定関数）を推定し，その推定した政策を利用して他エージェントが未来に実行する行動を予測する．そして，その予測行動を利用しながら強化学習を進行する．提案した手法を2体エージェント確率ゲームの枠組みでモデル化した追跡問題に適用し，実験を行い，提案手法の有効性を示す．

キーワード　マルチエージェント強化学習，Q学習，2体エージェント確率ゲーム，政策推定，行動予測

1. まえがき

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，学習による適応行動の自律的獲得に関する研究が，強化学習（RL）[1]の発展を契機として近年注目を集めている．RLのマルチエージェント環境への適用例は，サッカー[2],[3]，追跡問題[4]〜[7]，囚人のジレンマ[8]，共同ゲーム[9],[10]などが挙げられる．

RLは，もともとシングルエージェント環境を対象として発展してきた．そして，その研究の多くは，エージェントが対峙する環境の状態遷移が時不変な関数で記述される，マルコフ決定過程（MDP）[11]を基盤としている．ここで，学習主体としてのエージェントが複数存在するマルチエージェント環境を仮定し，個々のエージェントと環境の間の相互作用をMDPの枠組みでモデル化することを考える．MDPの枠組みでは，環境内に存在する自分以外のエージェント（以下，他エージェント）を環境の一部として扱うことになる．しかしながら，他エージェントを環境の一部として扱った場合，他エージェントの政策（行動決定関数）は学習に応じて時間とともに変化するため，個々のエージェントが認識する環境の状態遷移は時不変な関数で記述できない．すなわち，個々のエージェントと環境の間の相互作用はMDPの枠組みでモデル化できない．しかしながら，これまでに報告されているマルチエージェント強化学習（MARL：学習主体としてのエージェントが複数存在する環境における個々のエージェントのRL）に関する研究の多くでは，MDPを基盤として定式化された伝統的なRL法（Q学習[12]など）が，改良されることなくそのまま適用されている．言い換えると，これまでに報告されているMARL法の多くは，学習により時間とともに変化する他エージェントの政策を時不変な関数とみなして学習を行っている．

以上のような考察から，Littman [2]，Huら[13]は，動的に変化する他エージェントの政策（をもとに実行される他エージェントの行動）を考慮に入れられるようにQ学習を改良した，新たなMARL法を提案した．Littman [2]は，2体エージェント零和ゲーム（あるエージェントの利益が常に他エージェントの損失となるゲーム）を対象としたMARL法，mini-max Q学習法を提案した．mini-max Q学習法では，他エージェントが自分（エージェント）の利益を最小にする

[†] 奈良先端科学技術大学院大学情報科学研究科，生駒市
Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma-shi, 630–0101 Japan

行動を実行するという仮定のもと，そのような行動を自分の政策をもとに予測し，その予測行動を利用しながら RL を進行する．mini-max Q 学習は，2 体エージェントの利害関係が正反対となる環境においてのみ有効な手法である．これに対して，Hu ら [13] は，2 体エージェント間の利害関係が正反対になるとは限定されない環境における MARL 法を提案した．この MARL 法では，自分（エージェント）が Q 関数の学習を行うのと並行して，他エージェントの Q 関数も獲得する（両エージェントが Q 関数の学習を行い，お互いのエージェントがそのことを知っていると仮定している）．そして，獲得した他エージェントの Q 関数から他エージェントの政策を算出し（それぞれのエージェントが，お互いの政策算出法を知っていると仮定している），その算出した政策をもとに他エージェントが実行する行動を予測する．そして，その予測行動を利用しながら RL を進行する．この MARL 法では，他エージェントの Q 関数を獲得することで，他エージェントの政策を知ることを可能にしている．この MARL 法では，他エージェントの Q 関数を獲得するために，他エージェントが実際に実行した行動と，その行動を実行した後に他エージェントが環境から受け取る報酬を観測できることを仮定している．また，他エージェントが Q 関数の学習で使用するすべての学習パラメータを知っていることも仮定している．しかしながら，MARL において，他エージェントの行動，報酬が観測でき，更に他エージェントの（学習パラメータを含めた）学習法，政策算出法も知ることができると仮定することは，他エージェントの内部モデルを完全に知ることができることを意味し，マルチエージェント問題としてあまり現実的ではない．

そこで本論文では，他エージェントの報酬，学習パラメータ，学習法，政策算出法を知ることのできる情報と仮定せず，他エージェントが実際に実行した行動のみを知ることができる（観測できる）情報と仮定し(注1)，その観測情報のみをもとに他エージェントの政策を直接（Q 関数を獲得することなしに）推定する政策推定法を提案する．そして，その推定した政策を利用しながら RL を進行する，Q 学習に基づいた新たな MARL 法を提案する．

本研究では，学習により最適な政策を獲得することは目的とせず，他エージェントが存在することが原因で時変となる環境に対して，その環境に適応した有効な政策を獲得（学習）することを目的とする．

提案する MARL 法の性能を評価するためのマルチエージェント環境のモデルとして，本研究では，Littman [2]，Hu ら [13] と同様，2 体エージェント確率ゲーム（2-player stochastic game）[14] の枠組みを採用する．

以下，**2.** で，MDP と Q 学習について概説する．そして，**3.** で，2 体エージェント確率ゲームについて説明した後，提案する MARL 法について述べる．**4.** では，提案する MARL 法の性能を評価するための実験タスク，及び実験方法について説明し，実験結果を示す．そして，最後に **5.** でまとめる．

2. マルコフ決定過程と Q 学習

2.1 マルコフ決定過程（MDP）

MDP [11] は，シングルエージェント環境における行動決定問題のモデルで，4 項組 $\langle S, A, P, R\rangle$ で表される．ここで，S は環境状態の有限集合，A はエージェントの行動の有限集合，P は環境状態の遷移確率関数，R は報酬関数である．

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

$$P(s, a, s') = \Pr(s_{t+1} = s' | s_t = s, a_t = a) \qquad (1)$$

の集合で表される．ここで，$\Pr(s'|s, a)$ は，状態 s で行動 a を実行したときに状態が 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報酬 r_{t+1} も確率的で，その期待値は報酬関数

$$R(s, a) = E\{r_{t+1} = r | s_t = s, a_t = a\} \qquad (2)$$

で表される．ここで，$E\{r|s, a\}$ は，エージェントが状態 s で行動 a を実行したときに受け取る報酬 r の期待値を表す．

2.2 Q 学習

Q 学習 [12] は，MDP を対象として提案された RL 法である．Q 学習では，Q 関数と呼ばれる関数 $Q(s, a)$ をもとに行動選択，行動学習を行う．ここで，$Q(s, a)$ は状態 s で行動 a を実行する価値を表す関数で，値

(注1)：他エージェントが実行した行動を観測できるという仮定は，ゲーム理論 [14] の研究領域において現実的な仮定である．

が大きいほど，その状態でその行動を実行することが効果的である（将来に多くの報酬が得られることが期待できる）ことを表す．

Q 学習では，すべての $s \in S$，$a \in A$ に対する Q 関数値 $Q(s,a)$ を任意の初期値に設定して学習を開始し，エージェントが試行錯誤の経験を通して Q 関数を更新することにより学習を進行する．Q 学習における学習の流れは以下である．

（1） 現在（時刻 t とする）の環境状態 $s_t \in S$ において，エージェントは Q 関数から算出される政策 $\pi(s_t,a)$ に従って行動を確率的に選択する．ここで，政策 $\pi(s,a)$ は状態 s で行動 a を選択する確率を表す．政策算出法（以下，行動選択法と書く場合もある）には，これまでにいくつかの手法が提案されているが，本論文では，その中でも代表的なボルツマン選択法（式 (3)）と ε-greedy 選択法（式 (4)）を採用する．

$$\pi(s_t,a) = \frac{e^{Q(s_t,a)/T}}{\sum_{b\in A} e^{Q(s_t,b)/T}} \tag{3}$$

$$\pi(s_t,a) = \begin{cases} \dfrac{1-\varepsilon}{h} + \dfrac{\varepsilon}{|A|} & (a = \arg\max_b Q(s_t,b)) \\ \dfrac{\varepsilon}{|A|} & (\text{otherwise}) \end{cases} \tag{4}$$

ここで，式 (3) 中の T 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．また，式 (4) 中の $|A|$ は行動集合 A の要素数，h は，状態 s_t における Q 関数 $Q(s_t,b)$ の値が最大となる行動 $b \in A$ の数，$\varepsilon \in [0,1]$ はパラメータである．

（2） エージェントは手続き（1）で選択した行動 a_t を実行する．環境状態は s_t から s_{t+1} へ遷移し，エージェントは環境から報酬 r_{t+1} を受け取る．エージェントは，状態 s_t，行動 a_t に対する Q 関数値を式 (5) に従って更新（学習）する．

$$Q(s_t,a_t) \leftarrow (1-\alpha)Q(s_t,a_t) + \alpha(r_{t+1} + \gamma \max_{a'\in A} Q(s_{t+1},a')) \tag{5}$$

ここで，$\alpha \in (0,1]$，$\gamma \in [0,1]$ は，それぞれ学習率，割引率と呼ばれるパラメータである．

（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．

3. 2 体エージェント確率ゲームとマルチエージェント強化学習

前章で説明した MDP は，シングルエージェント環境における行動決定問題のモデルである．MDP を 2 体エージェント環境における行動決定問題のモデルに拡張したものとして，2 体エージェント確率ゲーム（2-player stochastic game）[14] がある．

3.1 2 体エージェント確率ゲーム

2 体エージェント確率ゲームは，2 体のエージェント（agent_1，agent_2 とする）と環境の間の相互作用をモデル化したもので，6 項組 $\langle S, A^1, A^2, P, R^1, R^2\rangle$ で表される．ここで，S は環境状態の有限集合，A^k $(k=1,2)$ は agent_k の行動の有限集合，P は環境状態の遷移確率関数，R^k は agent_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，agent_k $(k=1,2)$ 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，agent_k は環境から報酬 r_{t+1}^k を受け取る．環境状態の遷移は状態遷移確率関数 P に従う．この関数は時不変な状態遷移確率

$$P(s,a^1,a^2,s') = \Pr(s_{t+1}=s' | s_t=s, a_t^1=a^1, a_t^2=a^2) \tag{6}$$

の集合で表される．ここで，$\Pr(s'|s,a^1,a^2)$ は，状態 s でそれぞれのエージェントが行動 a^1，a^2 を実行したときに状態が s' へ遷移する確率を表す．agent_k が環境から受け取る報酬 r_{t+1}^k も確率的で，その期待値は報酬関数

$$R^k(s,a^1,a^2) = E\{r_{t+1}^k = r^k | s_t=s, a_t^1=a^1, a_t^2=a^2\} \tag{7}$$

で表される．ここで，$E\{r^k|s,a^1,a^2\}$ は，状態 s でそれぞれのエージェントが行動 a^1，a^2 を実行したときに agent_k が受け取る報酬 r^k の期待値を表す．

3.2 マルチエージェント強化学習

本研究では，2 体エージェント確率ゲームにおいて，それぞれのエージェントが自律的に RL する場合を考える．ここで，本研究で取り扱う 2 体エージェント確率ゲームは，不完全情報ゲーム [14] を仮定し，次の三つの条件を満たすものとする．

- 両エージェントが同期して行動を実行する．
- 両エージェントがお互いの行動を観測できる．

● エージェント間のコミュニケーションは存在しない．

このようなマルチエージェント環境において，個々のエージェントと環境の間の相互作用は，他エージェントの政策が学習に応じて時間とともに変化するため，MDP の枠組みでモデル化することができない（式 (6) の状態遷移確率関数，式 (7) の報酬関数のそれぞれが，2 体のエージェントの両方の政策（をもとに実行される行動）に依存することに注意する）．したがって，MDP を対象として定式化された Q 学習をそのまま適用することは適切ではない．

Littman [2]，Hu ら [13] は，2 体エージェント確率ゲームにおいて，時間とともに変化する他エージェントの政策を考慮に入れられるように Q 学習を改良した MARL 法を提案した．これらの MARL 法では，学習時に他エージェントの政策（をもとに実行される行動）を考慮に入れるため，(agent_k の) Q 関数を $Q^k(s, a^k, a^o)$ と書き換えている．ここで，a^o は他エージェントの行動を表す（$k = 1$ のとき $o = 2$，$k = 2$ のとき $o = 1$ に対応する）．このように Q 関数を書き換えた場合，agent_k が Q 関数 $Q^k(s, a^k, a^o)$ を利用して行動選択をするとき（a^k を決定するとき）に，他エージェントの行動 a^o が未知変数となる．上述の条件（不完全情報ゲームの条件）より，a^k の決定時には a^o は観測できないことに注意する．したがって，何らかの方法でこの未知変数 a^o の値を予測しなければならない．

Littman [2] は，2 体エージェント確率ゲームの中でも，更に，2 体のエージェントの利害関係が正反対となる 2 体エージェント零和ゲームに限定し，未知変数 a^o は自分（エージェント）の利益を最小にする行動であるという仮定のもと，そのような行動を自分の政策をもとに予測した．この予測方法は，2 体エージェント零和ゲームに対してのみの有効な手法である．一方，Hu ら [13] は，上述の不完全情報ゲームの三つの条件に，「他エージェントが環境から受け取る報酬を観測できる」という条件を加え，更に，他エージェントの正確な政策を知るために必要な他エージェントに関する情報（具体的には，他エージェントの学習法（学習パラメータの値も含む），政策算出法）を知ることができると仮定し，それらの情報を利用して他エージェントの正確な政策を獲得し，その政策を用いて a^o を予測した．この手法は，2 体エージェント零和ゲーム以外の確率ゲームにも適用可能である．しかしながら，他エージェントの報酬，学習法，政策算出法を知ることができると仮定することは，マルチエージェント問題としてあまり現実的ではない．

そこで本論文では，他エージェントの報酬，学習法，政策算出法を知ることができると仮定せず，他エージェントが実際に実行した行動のみを知ることができる（観測できる）情報と仮定し（上述の不完全情報ゲームの条件の二つ目），その観測情報をもとに他エージェントの政策を推定する政策推定法を提案する．そして，その推定した政策を基に変数 a^o を予測しながら学習を進行する新たな MARL 法を提案する．

3.2.1 他エージェントの政策推定法

以下に，本論文で提案する他エージェントの政策推定法を示す．ここで，agent_k が推定した他エージェントの政策を $I^k(s, a^o)$ とし，他エージェントが状態 s で行動 a^o を実行する（と予想される）確率を表すものとする．

時刻 t において，他エージェントが状態 s_t で行動 a^o_t を実行したとする．そのとき，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) に従って関数 I^k を更新する．

$$I^k(s, a^o) \leftarrow (1 - \theta) I^k(s, a^o) + \begin{cases} \theta & (a^o = a^o_t) \\ 0 & (\text{otherwise}) \end{cases} \tag{8}$$

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 $\sum_{a^o \in A^o} I^k(s, a^o) = 1$ が保たれることに注意する．本研究では，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 についてはあらかじめ知っていることを仮定する(注2)．

他エージェントの行動選択確率は，学習を通して，実行することが有効である（将来に多くの報酬が得られることが期待できる）行動に対して増加し，それ以外の行動に対して減少する．そして，他エージェントは，それらの行動選択確率（政策）をもとに行動を実行する．式 (8) の政策推定法は，他エージェントが実際に実行した行動に対する確率を少し増加させ，それ以外の行動に対する確率を少し減少させるものであり，他エージェントの政策を追従できると考えられる．

(注2)：A^o は，学習関数として $Q^k(s, a^k, a^o)$ を利用した時点で必要な情報である．

3.2.2 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強化学習法

以下に，本論文で提案する MARL 法の学習の流れを示す．

（1） 現在（時刻 t とする）の環境状態 $s_t \in S$ において，agent_k は，式 (9) で与えられる関数 $\bar{Q}^k(s_t, a^k)$ から算出される政策 $\pi^k(s_t, a^k)$ に従って行動を確率的に選択する．ここで，政策 $\pi^k(s, a^k)$ は agent_k が状態 s で行動 a^k を選択する確率を表す．

$$\bar{Q}^k(s_t, a^k) = \sum_{a^o \in A^o} I^k(s_t, a^o) Q^k(s_t, a^k, a^o) \quad (9)$$

行動選択法としてボルツマン選択法を用いる場合は，式 (3) 中の π を π^k，Q を $\bar{Q}^k$，A を A^k，a を a^k で置き換えたものを利用する．ε-greedy 選択法を用いる場合は，式 (4) 中で同様の置換えをしたものを利用する．

（2） agent_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 を実行する）．agent_k は他エージェントの行動 a_t^o を観測する．両エージェントの行動により，環境状態は s_t から s_{t+1} へ遷移し，agent_k は環境から報酬 r_{t+1}^k を受け取る．agent_k は状態 s_t，行動 a_t^k，a_t^o における Q 関数値を式 (10) に従って更新（学習）し，状態 s_t における関数 I^k を式 (8) に従って更新する．

$$Q^k(s_t, a_t^k, a_t^o) \leftarrow (1-\alpha) Q^k(s_t, a_t^k, a_t^o) + \alpha(r_{t+1}^k + \gamma_k \max_{a^k} \bar{Q}^k(s_{t+1}, a^k)) \quad (10)$$

（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．

以上の学習法を 2 体エージェントの両方が自律的に行う．

4. 評価実験

4.1 タスク（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 [15] 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（7×7）のグリッド空間中に，2 体のハンター（hunter）と 1 体の獲物（prey）が存在する（図 1）．ここで，グリッド空間の上と下，左と右の境界はつながっているものとする．

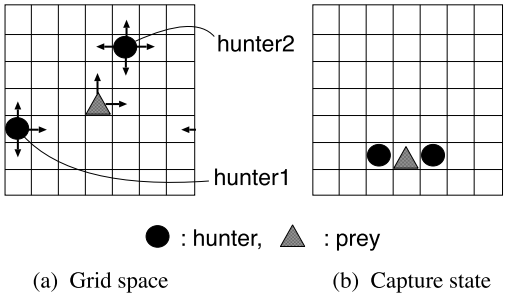

図 1 追跡問題のグリッド空間
Fig. 1 Grid space of pursuit problem.

- 本研究では，ハンターを『エージェント』と定義する．
- 各時間ステップごとに，ハンターと獲物は，それぞれ一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（図 1 (a) の矢印），現在位置にとどまる，の 5 通りである．獲物の実行可能な行動は，隣接する上，右のグリッドへ移動する（図 1 (a) の矢印），現在位置にとどまる，の 3 通りである．
- 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，「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」とする（図 1 (b) は左右から挟んだ状態の例）．
- 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．
- ハンターが環境から受け取る報酬は，獲物捕獲時に 1.0，それ以外のときに -0.05 とする．
- 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，$s = (p^1, p^2)$ とする．ここで，p^1，p^2 は，それぞれ hunter_1 と獲物の相対位置，hunter_2 と獲物の相対位置を表す．例えば，図 1 (a) における環境状態 s は $([3,1],[-1,-2])$ である．
- 獲物は学習を行わず，3 通りの行動の中から一つの行動を確率的に選択する．実験では，上，右へ移動する確率をそれぞれ $\frac{2}{5}$，現在位置にとどまる確率を $\frac{1}{5}$ とする．この行動選択確率は時不変とする．

以上の問題設定によって定義される追跡問題は，2 体エージェント確率ゲームの条件を満たす．

4.2 中央集権的学習器による Q 学習

ここで，提案した MARL 法を評価するための

比較対象として，中央集権的学習器による Q 学習（Centralized Q-Learning．以下，CQL）を定義する．CQL では，両ハンターの行動が完全に中央集権的学習器の制御下にあるものとし，中央集権的学習器は両ハンターの行動の対 $\boldsymbol{a} = (a^1, a^2) \in A$ を行動とみなした通常の Q 学習をする．ここで，a^1，a^2 は，それぞれ hunter_1，hunter_2 の行動である．CQL 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，中央集権的学習器は Q 関数 $Q(s_t, \boldsymbol{a})$ から算出される政策 $\pi(s_t, \boldsymbol{a})$ に従って行動対 $\boldsymbol{a} = (a^1, a^2)$ を確率的に選択する．ここで，行動選択法としてボルツマン選択法を用いる場合は，式 (3) 中の a を $\boldsymbol{a}$，b を $\boldsymbol{b}$ で置き換えたものを利用する．ε-greedy 選択法を用いる場合は，式 (4) 中で同様の置換えをしたものを利用する．

（2） 中央集権的学習器が手続き（1）で選択した行動対 $\boldsymbol{a}_t = (a_t^1, a_t^2)$ に従い，hunter_1，hunter_2 は，それぞれ行動 a_t^1，a_t^2 を実行する．環境状態は状態遷移確率 $P(s_t, a_t^1, a_t^2, s_{t+1})$ に従って s_t から s_{t+1} へ遷移し，中央集権的学習器は環境から報酬 r_{t+1} を受け取る．ここで，この報酬 r_{t+1} は，獲物捕獲時に 1.0，それ以外のときに -0.05 とする．中央集権的学習器は，状態 s_t，行動対 $\boldsymbol{a}_t$ に対する Q 関数値を式 (11) に従って更新（学習）する．

$$Q(s_t, \boldsymbol{a}_t) \leftarrow (1-\alpha) Q(s_t, \boldsymbol{a}_t) + \alpha (r_{t+1} + \gamma \max_{\boldsymbol{a}'} Q(s_{t+1}, \boldsymbol{a}')) \quad (11)$$

（3） 捕獲状態を満たしていなければ，t に 1 を加えて手続き（1）に戻る．捕獲状態を満たしていれば，t を 0 として手続き（1）に戻り，新たなエピソードを開始する．

CQL において，ハンターは，もはやエージェントではなく行動実行器にすぎず，CQL は，中央集権的学習器をエージェントとみなしたシングルエージェント RL であることに注意する．本研究の追跡問題では，獲物の行動選択確率は時不変であるため，中央集権的学習器から見た環境状態の遷移は，時不変な状態遷移確率関数 $P(s, \boldsymbol{a}, s')$ $(= P(s, a^1, a^2, s'))$ で記述できる．また，中央集権的学習器に与えられる報酬は式 (2) の報酬関数を満たす．したがって，CQL は，両ハンターの行動の対を中央集権的学習器の行動とみなした，MDP における通常の Q 学習である．

4.3 実験結果

4.3.1 実験 1：行動選択法としてボルツマン選択法を利用した場合

提案した MARL 法（Multi-agent Q-Learning with Policy Estimation．以下，MQLwPE）と CQL を追跡問題に適用した．図 2 に，行動選択法としてボルツマン選択法を利用した場合の実験結果を示す．図 2 中の MQLwoPE（Multi-agent Q-Learning without Policy Estimation）は，MQLwPE において，関数 I の更新を行わず，すべての $s \in S$，$a^o \in A^o$ に対して，$I^k(s, a^o) = 0.2$ に固定して学習を進行したものである．これは，他エージェントの行動をランダムに予測しながら学習を進行することを意味する．図 2 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 2 の結果は，10 学習エピソードごとに，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わない）の実験を行い，その平均時間ステップ数を示したものである．三つの学習法で使用した学習パラメータの値は，$\alpha = 0.3 \times 0.998849^{num\_ep}$，$\gamma_k = \gamma = 0.9$，$T = 0.1 \times 0.998849^{num\_ep}$，$\theta = 0.5 \times 0.998849^{num\_ep}$ である．ここで，num_ep は学習エピソード数を表す．減衰係数 0.998849 は，$0.998849^{2000} \approx 0.1$ となるように選ばれた値である．すべての $s \in S$，$a^k \in A^k$，$a^o \in A^o$，$\boldsymbol{a} \in A$ に対して，Q 関数，関数 I の初期値は，それぞれ $Q^k(s, a^k, a^o) = 0.0$，$Q(s, \boldsymbol{a}) = 0.0$，

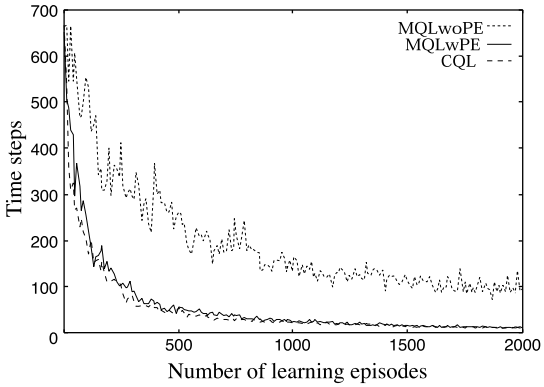

図 2 獲物捕獲までに費やした平均時間ステップ数：行動選択法としてボルツマン選択法を利用した場合
Fig. 2 Average time steps needed to capture the prey: The experiments in which the Boltzmann selection is employed as an action-selection method.

$I^k(s, a^o) = 0.2$ としている．図 2 の結果は，CQL と MQLwPE では，捕獲までに費やした平均時間ステップ数の値が収束し，安定した捕獲行動が獲得できていることを示している．一方，MQLwoPE では，安定した捕獲行動が獲得できず，獲物捕獲までに費やした時間ステップ数は，他の二つの学習法より多いことを示している．

図 2 中の MQLwPE と MQLwoPE を比較することにより，提案した行動観測に基づく政策推定法が Q 関数の学習に効果的に働いていることがわかる．本研究の追跡問題において，ハンターの学習には，「獲物に接近する行動の学習」と「獲物を捕獲する行動の学習」の二つの過程があると考えられる．前者は，他ハンターの位置や行動にあまり依存しないのに対して，後者は，他ハンターとの協調が必要で，他ハンターの実行する行動の予測が重要となってくる．MQLwoPE は，他ハンターが 5 通りの行動を等確率で選択すると予測しながら学習を進行するため，前者の獲物に接近する行動を学習することはできるが，後者の獲物を捕獲する行動を学習することはできず，捕獲は確率的である．

図 2 中の CQL と MQLwPE を比較することにより，この二つの学習法の学習性能は，学習初期段階で MQLwPE の方が幾分劣るが，ほぼ同等であることがわかる．CQL は，中央集権的学習器が 2 体のハンターの行動を完全に制御するもので，マルチエージェント系の特長である「個々のエージェントの自律性」は失われるが，効率的な学習が可能である．一方，MQLwPE は，「個々のエージェントの自律性」が保たれており，学習進行時に未知変数（他エージェントの行動）が存在する．この未知変数の予測がうまくいかない場合，学習はうまくいかないことが予想される（MQLwoPE の結果は，その一例である）．CQL と MQLwPE の学習性能がほぼ同等であることは，提案した政策推定法が効果的で，MQLwPE が有効な MARL 法であることを示している．

4.3.2 実験 2：行動選択法として ε-greedy 選択法を利用した場合

実験 1 では，行動選択法としてボルツマン選択法を採用した場合，提案した MARL 法が有効であること示した．ここでは，行動選択法として ε-greedy 選択法を利用した場合の実験を行う．実験結果を図 3 に示す．実験で使用したパラメータ ε の値は，$\varepsilon = 0.3 \times 0.998849^{num-ep}$ である．その他のパラメータの値，報酬の値，Q 関数，関数 I の初期値は実験 1 と同じである．図 3 の結果は，図 2 の結果と同様，CQL と MQLwPE の両学習法で安定した捕獲行動が獲得できており，両学習法の獲物捕獲までに費やした時間ステップ数はほぼ同じであることを示している．また，MQLwoPE では，獲物捕獲までに費やした時間ステップ数が他の二つの学習法より多いことを示している．これは，行動選択に ε-greedy 選択法を利用した場合でも，提案した他エージェントの政策推定法が Q 関数の学習に効果的に働いていることを示している．提案した他エージェントの政策推定法は，他エージェントが実際に実行した行動に対する推定確率を少し増加させ，それ以外の行動に対する推定確率を少し減少させるものである．ε-greedy 選択法では，$\bar{Q}$ 関数値が最大となる行動が変化すると，新たに最大となった行動と，それまで最大だった行動の行動選択確率が大きく変化するが，学習の進行とともに，$\bar{Q}$ 関数値が最大となる行動の変化は少なくなるため，提案した政策推定法でも他エージェントの政策が追従できているものと考えられる．

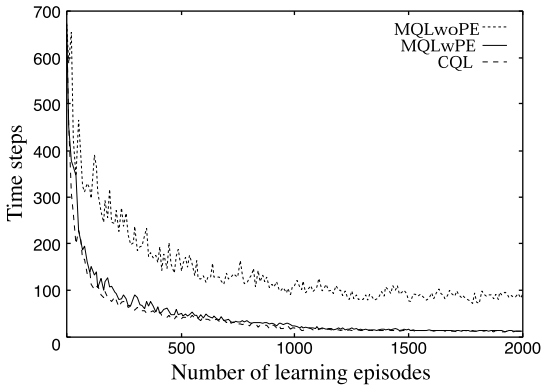

図 3 獲物捕獲までに費やした平均時間ステップ数：行動選択法として ε-greedy 選択法を利用した場合

Fig. 3 Average time steps needed to capture the prey: The experiments in which the ε-greedy method is employed as an action-selection method.

4.3.3 実験 3：片方のハンターの報酬関数を途中で変更した場合

ここでは，片方のハンターの報酬関数を学習途中で変更した場合の実験を行う（報酬関数が途中で変わるため，確率ゲームの枠組みからは逸脱する）．この実験は，学習によるエージェントのパフォーマンスがある程度収束してきたあたりで急に他エージェントの政

策が変化した場合，提案した MARL 法は対応できるかどうかを調査するためのものである．実験では，学習を通して $\theta = 0.1$ で固定とする以外は，学習エピソード数が 500 までは実験 1 と全く同じ条件で学習を行う．そして，501 学習エピソードから，hunter_2 の報酬 r^2 のみを変更する（hunter_1 の報酬 r^1 はそのままとする．すなわち，獲物捕獲時に $r^1 = 1.0$，それ以外のときに $r^1 = -0.05$ である）．501 学習エピソード以降の，hunter_2 の報酬は次のようなものである．hunter_2 は，捕獲状態を満たし，かつ，獲物の右側に位置した場合にのみ成功報酬 $r^2 = 1.0$ を得る．そして，捕獲状態を満たし，かつ，獲物の上，下，左側のいずれかに位置した場合には，失敗報酬 $r^2 = -1.0$ を得る．それ以外の場合（捕獲状態を満たしていない場合）の報酬は $r^2 = 0.0$ とする(注3)．このような報酬関数の設定においても，学習による協調解は，「hunter_1 が獲物の左側，hunter_2 が右側から挟む」として存在することに注意する．実験結果を図 4 に示す．図 4 の結果は，MQLwPE では 501 学習エピソード以降，捕獲までに費やす平均時間ステップ数はいったん増加するが，しばらくするとまた減少し，そのまま収束していっていることを示している．これは，学習によるエージェントのパフォーマンスがある程度収束してきたあたりで急に他エージェントの政策が変化した場合でも，提案した MARL 法は対応できることを示している．ただし，ここでは $\theta = 0.1$ で固定していることに注意する．実験 1，実験 2 では，θ の値をやや高め

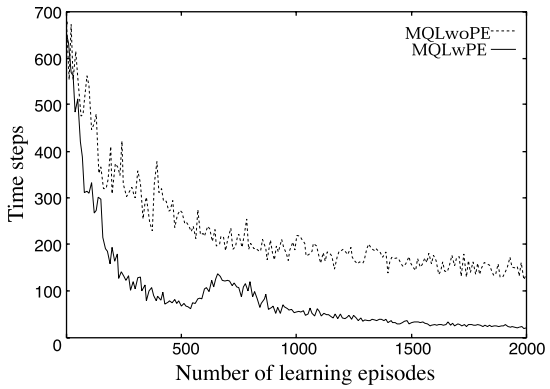

図 4　獲物捕獲までに費やした平均時間ステップ数：片方のハンターの報酬関数を途中で変更する場合
Fig. 4　Average time steps needed to capture the prey: An experiment in which one hunter's reward function changes during the learning.

（0.5）に設定し，そこからその値を減衰をさせることにより効果的な学習が行えた．それに対し，実験 3 では，やや低めの値を初期値とし，減衰を掛けずにそのまま値を固定している方が良い結果が得られた．これは，θ の値を固定することにより他エージェントの政策が途中で変化しても，その変化を追従することが可能であるためと考えられる．実験結果は提示していないが，θ の値をいろいろと変化させて実験を行ってみたところ，エージェントのパフォーマンスがある程度収束してきたあたりで他エージェントの政策が急に変化する場合は θ の値に減衰を掛けない方がよく，他エージェントの政策が急に変化しない場合は減衰を掛けた方がよいという結果が出ている．

5. むすび

本論文では，2 体エージェント確率ゲームにおける他エージェントの政策推定を利用した新たな MARL 法を提案した．提案した MARL 法では，他エージェントが過去に実行した行動の観測情報のみを利用して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した．そして，その予測行動を利用しながら RL を進行した．提案した MARL 法を 2 体エージェント確率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，行動選択法として，RL における代表的な行動選択法である，ボルツマン選択法と ε-greedy 選択法のいずれを利用した場合でも有効であることを示した．また，一方のエージェントの報酬関数が学習途中で変更されるような場合においても提案した MARL 法は有効であることを示した．

マルチエージェント環境における学習問題は，環境のダイナミックスが時間とともに変化するため，一般に難しい問題とされている．本論文では，2 体エージェ

(注3)：このような報酬関数の設定において，hunter_1 は，捕獲状態を満たさない限りずっと負の報酬 $r^1 = -0.05$ を得ることになるため，できるだけ早く捕獲状態を満たそうとする．一方，hunter_2 も 500 学習エピソードまでは，hunter_1 と同様にできるだけ早く捕獲状態を満たそうとするが，501 学習エピソード以降は，捕獲状態を満たした場合でも，そのときに獲物の右側に位置していない限り成功報酬 $r^2 = 1.0$ を得ることができず，獲物の右側以外（上，下，左側）に位置した場合には，失敗報酬 $r^2 = -1.0$ を得ることになってしまう．この失敗報酬は捕獲状態を満たしていない場合の報酬 $r^2 = 0.0$ よりも少ないため，捕獲状態において獲物の右側に位置できないのであれば，捕獲状態を満たさない（動き回っている）方がよくなる．ここで，このような報酬関数の違いを表現するような中央集権的学習器に対する単一の報酬関数を設計することは難しく，このような非均質エージェントの環境は CQL では取り扱うことが難しいことに注意する．

ント確率ゲームという限定された問題に対して，時間とともに変化する環境のダイナミックス（他エージェントの政策）を推定し，その推定を学習に利用した．そして，有効な実験結果を得た．しかしながら，本論文で示した実験結果は，タスク（追跡問題）に依存していることは否定できず，別のタスクに対しても検証する必要があると思われる．マルチエージェントシステムにおける行動決定問題には，追跡問題のような協調問題の他に競合問題（サッカーなど）と非協調・非競合問題（囚人のジレンマなど）がある．今後の課題は，これらの問題に提案した学習法を適用して評価することである．また，3 体以上の問題に適用できるように拡張し，実験及び評価をする予定である．

謝辞　本研究を行うにあたって，貴重な意見を頂戴した，奈良先端科学技術大学院大学の石井信教授，ATR 人間情報科学研究所の銅谷賢治主任研究員に厚く感謝致します．

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] R. Salustowicz, M. Wiering, and J. Schmidhuber, “Learning team strategies: Soccer case studies,” Machine Learning, vol.33, no.2, pp.263–282, 1998.

[4] 荒井幸代，宮崎和光，小林重信，“マルチエージェント強化学習の方法論–Q-Learning と Profit Sharing による接近，” 人工知能誌，vol.13, no.4, pp.609–618, July 1998.

[5] T. Kohri, K. Matsubayashi, and M. Tokoro, “An adaptive architecture for modular Q-learning,” Proc. 15th International Joint Conference on Artificial Intelligence, pp.820–825, Nagoya, Japan, Aug. 1997.

[6] N. Ono and K. Fukumoto, “Multi-agent reinforcement learning: A modular approach,” Proc. 2nd International Conference on Multi-Agent Systems, pp.252–258, Kyoto, Japan, Dec. 1996.

[7] M. Tan, “Multi-agent reinforcement learning: Independent vs. cooperative agents,” Proc. 10th International Conference on Machine Learning, pp.330–337, Amherst, Massachusetts, USA, June 1993.

[8] T.W. Sandholm and R.H. Crites, “Multiagent reinforcement learning in the iterated prisoner’s dilemma,” Biosystems, vol.37, no.1-2, pp.147–166, 1996.

[9] C. Claus and C. Boutilier, “The dynamics of reinforcement learning in cooperative multiagent systems,” Proc. 15th National Conference on Artificial Intelligence, pp.746–752, Madison, Wisconsin, USA, July 1998.

[10] S. Sen, M. Sekaran, and J. Hale, “Learning to coordinate without sharing information,” Proc. 12th National Conference on Artificial Intelligence, pp.426–431, Seattle, Washington, USA, July-Aug. 1994.

[11] M.L. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming, John Wiley & Sons, New York, 1994

[12] C.J.C.H. Watkins and P. Dayan, “Technical note Q-learning,” Machine Learning, vol.8, no.3, pp.279–292, 1992.

[13] 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.

[14] G. Owen, Game Theory, Third edition, Academic Press, San Diego, California, 1995.

[15] L. Gasser, N.F. Rouquette, R.W. Hill, and J. Lieb, “Representing and using organizational knowledge in distributed AI systems,” in Distributed Artificial Intelligence, ed. L. Gasser and M.N. Huhns, vol.2, pp.55–78, Morgan Kaufmann, Los Altos, California, USA, 1989.

（平成 13 年 11 月 12 日受付，14 年 12 月 3 日再受付）

長行　康男　（学生員）

平 8 愛媛大・工・情報卒．現在，奈良先端科学技術大学院大学情報科学研究科博士後期課程に在学中．マルチエージェントシステム，強化学習の研究に従事．

伊藤　　実　（正員）

昭 52 阪大・基礎工・情報卒．昭 54 同大大学院修士課程了．同年同大・基礎工・情報助手．昭 61 同講師，平元同助教授，平 5 奈良先端科学技術大学院大学情報科学研究科教授，現在に至る．平 3 ウォータールー大（カナダ）客員準教授．工博．