2A1-1

Q値累積型モンテカルロ法に関する一考察

About Q-values of Monte Carlo method

植村　渉[*1]
Wataru UEMURA

[*1]龍谷大学　理工学部
Ryukoku University

Profit Sharing method is one of the reinforcement learning methods. Profit Sharing can work well on the Partially Observable Markov Decision Processes(POMDPs). Because it is the typical non-bootstrap method, and it's Q-value is usually handled accumulative. Profit Sharing, however, does not work well on the probabilistic state transition. This paper we propose the novel learning method which can work well on the probabilistic state transition. It is similar to the Monte Carlo method. So we discuss about Q-values of our proposed method. In the environment with deterministic state transitions, we show the same performance both the conventional Profit Sharing and proposed method. And show the good performance of proposed method against the conventional Profit Sharing.

1. はじめに

強化学習 [8] の枠組みでは，学習者である**エージェント**が目標状態に到達したときに，エージェントに報酬が与えられる．エージェントは試行錯誤を繰り返し，その報酬情報を基にして，よりよい政策を学習する必要がある．**時刻** t において，環境の**状態** s_t をエージェントは**知覚** o_t として観測する．このとき，エージェントは**政策**にしたがって，選択できる行動群 A の中から**行動** a_t を選択し実行する．状態 s_t における行動 a_t のことを**ルール** (s_t, a_t) と言う．行動 a_t を実行した結果，環境は状態 s_t から次状態 s_{t+1} へと変化する．もし，次状態 s_{t+1} が目標状態であれば，エージェントは**報酬** r_{t+1}（> 0）を受け取る．目標状態でなければ，$r_{t+1} = 0$ である．

環境がマルコフ決定過程（**MDPs**:Markov Decision Processes）のクラスに属する場合，状態 s_t から状態 s_{t+1} へ遷移する確率 $P^{a_t}_{s_t,s_{t+1}}$ は，状態 s_t と行動 a_t によって決まる．もし，エージェントの知覚能力に制限がある場合，いくつかの状態 s_t を同じ観測として知覚する場合がある．このとき，部分観測可能マルコフ決定過程（**POMDPs**:Partially Observable Markov Decision Processes）のクラスとなり，本来別の行動をとる必要がある複数の状態を同じ観測として知覚した場合，不完全知覚問題が生じる [4, 6, 7, 13].

Profit Sharing 法 [2, 5] は，POMDPs 環境に対しても頑強であり [1]，また学習の立ち上がりも早いことから，実世界への適用が期待されている．

本研究では，Profit Sharing 法が確率的な状態遷移下で価値を適切に扱えないことを指摘し，それを克服した新たな学習方法を提案する．POMDPs 環境への適用も議論し，Profit Sharing 法が持つ特長を失わず，学習性能を向上させる．確率的な状態遷移を含む簡単な迷路環境で実験を行い，提案手法の有効性を確認する．

2. Profit Sharing 法

この節では，強化学習の一つである Profit Sharing 法について紹介する．Profit Sharing 法では，エージェントが目標状態に到達したときに，それまでの行動系列（**エピソード**と呼ぶ）のルールの評価値に報酬を分配し，強化する方法である．報酬の分配関数 $f(x)$ を**強化関数**と呼び，MDPs 環境では，

$$f(x) = 1/L^x, \tag{1}$$

が提案されている [5]．ここで，x は目標状態からさかのぼったステップ数で，L は各状態での行動の数である．また，一部の POMDPs 環境に対しては，

$$f(x) = 1/L^w, \tag{2}$$

が有効であることが報告されている [10, 11]．ここで，w はエピソードの長さである．報酬は上記強化関数を用いて，ルールの評価値へ累積されることが多い．評価値は過去のルールの価値の累積値となるため，Profit Sharing 法では，評価値の大きさに応じた割合でルールを選択する**ソフトマックス行動選択**を用いることが多い．例えば，ルーレット選択やボルツマン分布に従った選択などである．

POMDPs 環境では，別の行動を選択する必要がある異なる状態を，同じ観測として知覚する場合がある．つまり，ある観測において，複数の行動を選択できる必要がある [3].

Profit Sharing 法は，POMDPs 環境に対して以下の２つの点で頑強である．１つ目の理由は，評価値の更新が**非ブートストラップ型**だからである．非ブートストラップ型とは，評価値の更新において他のルールの評価値を用いないことである．隣の観測のルールの評価値には，複数の状態のルールの価値が含まれているため本来の価値と異なる場合があり，悪影響を受ける．非ブートストラップ型の更新方法では，その影響を受けない分頑強である．Profit Sharing 法の更新式は，

$$\omega(s_x, a_x) \quad \leftarrow \quad \omega(s_x, a_x) + f(x), \quad \text{for all } x \text{ in the episode}, \tag{3}$$

である．ここで，$\omega(s_x, a_x)$ は，ルール (s_x, a_x) の評価値である．式 (3) は，累積型の更新式として提案されており [5]，他のルールの評価値を必要としない．

２つ目の理由は，Profit Sharing 法の行動選択方法がソフトマックス行動選択だからである．不完全知覚問題が生じる環境では，１つの観測に対して複数の行動を選択する必要がある．

連絡先: 植村　渉，龍谷大学　理工学部，滋賀県大津市瀬田大江町横谷１－５，077-543-7416，077-543-7428(FAX)，wataru@rins.ryukoku.ac.jp

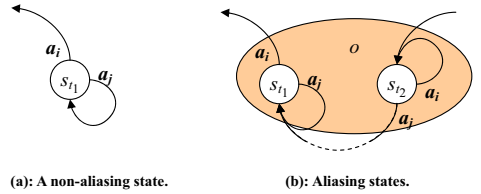

図 1: POMDPs 環境において不完全知覚問題が生じる場合

5	11	14	20	26	31	37		G
4	10		19	25	30	36		45
S	9		18	24	29	35		44
3	8		17	23	28	34	40	43
2	7	13	16	22		33	39	42
1	6	12	15	21	27	32	38	41

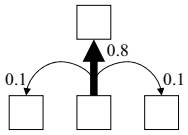

図 2: 迷路環境と確率的な状態遷移

例えば図 1 のように目標状態へ近づくために，行動 a_j を必要とする状態 s_{t_2} と，行動 a_i を必要とする状態 s_{t_1} の両方を観測 o と知覚する場合を考える．このとき，エージェントが観測 o を知覚したときに，行動 a_i のみしか実行できないと，目標状態へ到達することができない．最も高い評価値を持つルールを選択するグリーディ選択方法では，このような問題に直面する．ソフトマックス行動選択であれば，両方のルールを均一に強化することで，この問題を解決できる．そのための強化関数が，式（2）である．

従来の強化学習法では，グリーディ行動選択を用いることが多い．グリーディ行動選択では，各状態において最大の評価値（Q値）を持つルールを選択する．評価値がルールの報酬獲得期待値に収束する場合，最大の値を選び続けることで最適解を選択し続けることが期待できる．しかしこの方法では，二番目に大きな評価値を持つルールは，決して選択されない．そのため，従来の強化学習法は POMDPs 環境ではうまく行動を選択できない．累積型 Profit Sharing 法で用いる式（3）では，評価値の値そのものに意味はなくその大きさの比率が意味を持つ．言い換えると，グリーディ行動選択を用いる場合は，ルールの評価値がそれぞれ理論値に収束する必要があり，ソフトマックス行動選択を用いる場合は，評価値に応じたルールの選択確率が理論値に収束する必要がある．累積型 Profit Sharing 法において，行動選択方法としてソフトマックス行動選択を用いることでエージェントは同じ観測に対して複数の行動を選択できるため，POMDPs 環境に頑強である．

しかしながら，従来の累積型 Profit Sharing 法では，確率的な状態遷移を考慮していない問題点がある [12]．例えば，状態遷移確率がいくつであれ，同じ報酬を分配する．本来，期待値とは報酬 R と確率 P の乗算値を意味するはずである．そこで，報酬分配値を，その期待値に近づける必要がある．

状態遷移確率を知るためには多数の試行が必要であるため，一つのエピソード情報による強化では，うまく扱うことができない．従来の強化学習法では，試行錯誤中に評価値を更新することで，確率的状態遷移に対する期待値を扱ってきた．これは**オンライン型**の強化学習と呼ばれる．Profit Sharing 法は，行動選択時には評価値を更新しないため**オフライン型**の強化学習である．

3. 新しい評価値更新方法の提案

この節では，確率的な状態遷移を考慮した新しい評価値更新方法を提案する．

従来の累積型 Profit Sharing 法では，一つのエピソード内で確率的な状態遷移を経験しても，そのルールに対しては決定的な状態遷移の場合と同じ価値を分配する．そのため，状態遷移確率の情報は，報酬分配量に反映されない．

同じルールを複数回選択した場合，価値の割引を行わなければならない．そこで，ルールの新しい評価値 $Q(s,a)$ として，

$$Q(s,a) \quad \leftarrow \quad n_{r(s,a)}/n_{a(s,a)} \times \omega(s,a), \tag{4}$$

を用いることを提案する．ここで，$n_{r(s,a)}$ はルール (s,a) によって報酬を得た回数であり，$n_a v(s,a)$ は，ルール (s,a) を選択した回数である．

ルール (s,a) の状態遷移が決定的な場合，ループを経験しない限りはルールを選択した回数 $n_a v(s,a)$ と報酬を得た回数 $n_{r(s,a)}$ は等しくなり，新しい評価値 $Q(s,a)$ は従来の評価値 $\omega(s,a)$ と等しくなる．ルール (s,a) の状態遷移が確率的な場合，評価値 $\omega(s,a)$ をルール (s,a) の選択回数で割ることは，期待値を意味する．

これに類する評価値の更新方法として，モンテカルロ法がある．モンテカルロ法では，評価値として獲得報酬の平均値を用いる．

$$Q(s,a) \quad \leftarrow \quad \omega(s,a)/n_{a(s,a)}. \tag{5}$$

この更新式により，評価値 $Q(s,a)$ は，獲得報酬の平均値に近づく．しかし，この評価値は累積型ではなく更新型のため，モンテカルロ法はグリーディ行動選択を用いることが多い．本論文で提案する更新式（4）は，分配報酬値を累積する．そのため，ソフトマックス行動選択法を用いることができる．よって，POMDPs 環境に対する頑強性は失われない．獲得報酬の平均値を扱いながら累積型の評価値の更新を行うため，**Q 値累積型モンテカルロ法**と呼ぶことにする．

4. 実験

強化学習でよく用いられる迷路環境 [9] にて実験を行った（図 2）．エージェントは，状態 S からスタートし，東西南北の 4 方向への移動を行動として選択する．状態 G に到着すると，報酬 $R = 10$ を得て状態 S に遷移する．1 ステップあたりの獲得報酬量を性能として評価する．状態遷移確率は，図 2 右に示すとおり，どの行動選択も正しい方向へ移動するのは 8 割の確率である．

状態遷移が決定的な場合は，提案手法も従来法も，ほぼ同じ性能を示した．強化方法に違いが出るのは，学習の初期段階においてエージェントがループを作った際に，提案手法の方が分配量が若干少なくなるため，わずかな差異が現れた．状態遷移が確率的な場合は，図 3 のように提案手法のほうが若干性能向上が確認できた．これは評価値が期待値に近づくため，学習の効率が向上していると考えられる．

5. おわりに

本論文では，状態遷移確率を考慮した新しい評価値更新方法を提案した．従来の累積型 Profit Sharing 法は，評価値の更

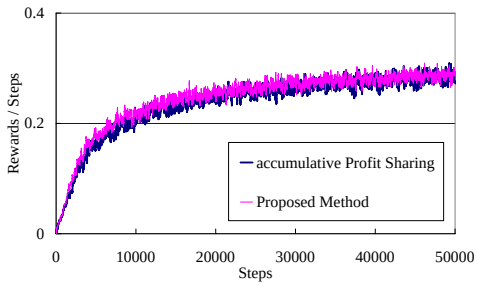

図 3: 累積型の Profit Sharing 法と，提案する Q 値累積型モンテカルロ法の性能比較

新が累積型であるため POMDPs 環境に頑強である反面，報酬を獲得するまで評価値を更新しないため同じルールを何度経験しても価値を割り引かず，確率的な状態遷移を持つ環境では適切な価値の更新ができなかった．そこで，評価値の算出式にルールの選択回数を組み込み，オンライン型とすることで確率的な状態遷移を持つ環境においても，適切な価値の更新を行う更新式を提案した．従来のモンテカルロ法と累積型 Profit Sharing 法の両方の長所を併せ持つ手法となった．性能評価においては，若干の向上が見られた．

今回は，MDPs 環境において考察したため，今後は POMDPs 環境における考察を進めていく必要がある．

参考文献

[1] 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 (1998)

[2] Grefenstette, J., “Credit assignment in rule discovery systems based on genetic algorithms,” *Machine Learning*, Vol. 3, pp. 225 – 243, 1988.

[3] 木村 元, 山村 雅幸, 小林 重信, “部分観測マルコフ決定過程下での強化学習”, 人工知能誌, Vol. 11, No. 5, pp. 761–768 (1996)

[4] McCallum, R., “Instance-based utile distinctions for reinforcement learning with hidden state”, in *Proc. of the 12th International Conference on Machine Learning*, pp. 387–395 (1995)

[5] 宮崎 和光, 山村 雅幸, 小林 重信, “強化学習における報酬割当ての理論的考察”, 人工知能誌, Vol. 9, No. 4, pp. 580–587 (1994)

[6] Miyazaki, K. and Kobayashi, S., “Learning Deterministic Policies in Partially Observable Markov Decision Processes”, *in Proceedings of International Conference on Intelligent Autonomous System 5*, pp. 250 – 257, 1998.

[7] 宮崎 和光, 荒井 幸代, 小林 重信, “POMDPs 環境下での決定的政策の学習”, 人工知能誌, Vol. 14, No. 1, pp. 148–156 (1999)

[8] Sutton, R., “Integrated Arcitecture for learning, planning, and reacting based on approximating dynamic programming”, *in Proc. of the 7th International Conference on Machine Learning*, pp. 216 – 224, 1990.

[9] Sutton, R., “Reinforcement learning – an introduction – ,” *the MIT Press*, 1998.

[10] Uemura, W., Ueno A. and Tatsumi, S., “The exploitation reinforcement learning method on POMDPs”, *in Joint 2nd International Conference on Soft Computing and Intelligent Systems*, TUE – 1 – 3, 2004.

[11] 植村 渉, 上野 敦志, 辰巳 昭治, “POMDPs 環境のためのエピソード強化型強化学習法”, 信学会論文誌, Vol. J88–A, No. 6, pp. 761–774 (2005)

[12] Uemura, W., “About distributing rewards to a rule with probabilistic state transition”, *in the SICE Annual Conference 2007, International Conference on Instrumentation, Control and Information Technology*, pp. 2762 – 2765, (2007).

[13] Whitehead, S. and Balland, D., “Active perception and reinforcement learning”, *in Proc. of the 7th International Conference on Machine Learning*, pp.162 – 169, 1990.