計測自動制御学会東北支部 第179回研究集会 (1998.12.14)
資料番号 179-10

非マルコフ環境におけるラベリングQ-学習

Labeling Q-Learning For Partially Observable Markov Decision Process Environment

○李 海妍*,釜谷博行**,阿部健一*

○HaeYeon Lee*, Hiroyuki Kamaya**, Kenichi Abe*

*東北大学大学院工学研究科, **八戸高専

*School of Engineering, Tohoku University , **Hachinohe National College of Technology

キーワード： 強化学習(Reinforcement Learning), マルコフ環境(Markov Decision Process Environment), 非マルコフ環境(Partially Observable Markov Decision Process Environment), ラベリングQ-学習(Labeling Q-Learning),

連絡先： 〒980-8579 仙台市青葉区荒巻字青葉05 東北大学大学院工学研究科電気·通信工学専攻 阿部研究室
李 海妍, Tel.: (022)217-7074, Fax.: (022)263-9298, E-mail: yeon@abe.ecei.tohoku.ac.jp

1. はじめに

近年、規模の大きい複雑なシステムの計画、制御問題を、複数の処理要素(あるいは、それ自身ある程度自律したものの意味で、処理主体(AGENT)と呼ぶことができる)によって、分散的に解決する仕組みに多くの研究者が関心を寄せている. 本研究では、非マルコフ環境(POMDP:Partially Observable Markov Decision Process)における強化学習を処理要素とする分散学習アルゴリズムを提案し、その学習性能を実証する.

2. 強化学習

学習者(Agent)がある状態(状況,State)である行動をとると、環境からの刺激入力を受け取る. その刺激入力を報酬(Reward)と呼ぶ. Agentの目標は、そのRewardの累積値を最大にする事である. Agentは、目標達成のために、環境との相互作用に

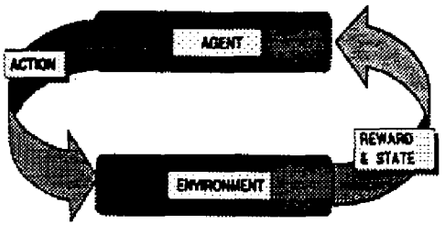

Fig. 1 強化学習の枠組

よって試行錯誤的に行動選択を改変する. この改変の仕組みに強化学習(RL:Reinforcement Learning)が用いられる. ここで、環境のある状態でのある行動に対する瞬時のRewardをもとに、その最終目標である、Rewardの累積値の最大化をはかるよう行動選択の仕方をうまく改変しなければならない.

3. MDPにおける強化学習

まず、離散マルコフ決定過程(Descrete Markov Decision Process:MDP、以下MDP)下の強化学習を考える．MDPでは入出力変数の値域には離散値、環境の性質にはマルコフ性を仮定する．時間は環境認識、行動選択、実行を一つのstepとして、離散化される．Rewardはscalar量であり、行動は離散的に選ばれる．Rewardに応じた実行可能な行動はルールとして記述される．各環境の観測に対し、選択すべきルールを与える関数を政策(Policy)と呼び、単位行動当りの期待獲得報酬を最大化するPolicyを最適政策(Optimal Policy)と呼ぶ．

3.1 MDPの学習問題

MDPは環境状態の有限集合S、行動の有限集合Aによって特徴付けられる．ある時点$t \in T = \{0,1,2,\ldots\}$において、環境状態が$s_t \in S$のとき、Agentが行動$a_t \in A$をとったとすると、Reward r_tを受け取り (Reward r_tは確率 $Pr(r_t|s_t,a_t)$で得られる)、環境状態は遷移確率$Pr(s_{t+1}|s_t,a_t)$ でつぎの状態 s_{t+1}に推移する.Agentの目標はRewardの割引期待利得$E\{\sum_{t=0}^{\infty}\gamma^t r_t\}$ $(0 \leqq \gamma \leqq 1)$を最大にすることである．環境が既知のときは、DP(Dynamic Programming)が適用できて、次式で解くことができる．

$V(s,a), Q(s,a)$を以下のように定義する．

$$V(s) = \max_{a\in A}[r_s^a + \gamma\sum_{s'} Pr(s'|s,a)V(s')] \quad (1)$$

$$Q(s,a) = r_s^a + \gamma\sum_{s'} Pr(s'|s,a)V(s') \quad (2)$$

$$r_s^a = E[r|s,a]$$

$$\Longrightarrow \quad V(s) = \max_a Q(s,a) \quad (3)$$

(3)式を満たすV、あるいはQを求めれば、最適政策が求まる．$Q(s,a)$をQ値と呼ぶことにする．

3.2 代表的な学習アルゴリズム

環境が未知のとき、学習によってQ値を逐次的に推定するアルゴリズムが必要である.代表的なアルゴリズムにはQ-learning[1]とその原型となるTemporal Difference法(TD法)[2]がある．TD法はMDP環境の各状態の評価を同定するのに対し、Q-learningは状態の評価だけではなく、状態と行動の対の評価を割引期待Rewardをもとに同定する．

1) Q-学習[1]

$$\begin{aligned} Q(s_t,a_t) &\leftarrow Q(s_t,a_t) + \alpha[r_t + \\ &\quad \gamma\max_a Q(s_{t+1},a) \\ &\quad -Q(s_t,a_t)] \quad (4) \\ Q(s,a) &\leftarrow Q(s,a) \\ &\quad (\text{for all } s \neq s_t \text{ or } a \neq a_t) \end{aligned}$$

2) TD(λ),Sarsa(λ)[2]

推定のみに使うのをTDといい、行動選択を伴う場合をSarsaと呼んでいる．

$$\begin{aligned} Q(s,a) &\leftarrow Q(s,a) + \alpha[r_t + \gamma Q(s_{t+1},a_{t+1}) \\ &\quad -Q(s_t,a_t)]e(s,a) \quad (5) \\ &\quad (\text{for all } s,a) \end{aligned}$$

ただし、$e(s,a)$:eligibility trace

eligibility traceの更新は次の通りである．

$$e(s_t,a_t) \leftarrow \gamma\lambda e(s_t,a_t) + 1 \quad (6)$$

$$e(s,t) \leftarrow \gamma\lambda e(s,a) \quad (7)$$

$$(\text{for all } s \neq s_t \text{ or } a \neq a_t)$$

また、eligibility traceの変形とも言えるreplacing traceが提案されている.[3),4)]

$$e(s_t,a_t) \leftarrow 1 \quad (8)$$

$$e(s,a) \leftarrow \gamma\lambda e(s,a) \quad (9)$$

$$(\text{for all } s \neq s_t \text{ or } a \neq a_t)$$

3.3 行動の選択

行動の選択のルールとしては次のようなものがある.

1) Greedy

$$a_t = \arg\max_a Q(s_t, a) \quad (10)$$

2) ε-Greedy

$$a_t = \arg\max_a Q(s_t, a) \ (1-\varepsilon\text{の確率}) \quad (11)$$

$$a_t : \text{random selection } (\varepsilon\text{の確率}) \quad (12)$$

3) Boltzman分布による方法

行動a_tを選択する確率を

$$\frac{e^{Q(s_t,a)/T}}{\sum_{a' \in A} e^{Q(s_t,a')/T}} \quad (13)$$

T:温度係数と呼ばれる正のパラメータ

4. POMDPにおける強化学習

4.1 部分観測マルコフ決定過程(POMDP)

部分観測マルコフ決定過程は、不十分なセンサのため状態観測に不確実性や不完全性の存在するシステムのモデルとして特に適している. POMDPにおける強化学習の厳密解法は、残念ながら極端に小さいかあるいは複雑さの小さい問題を除き、計算量的に実行不可能と考えられている. よって、POMDP環境下での強化学習問題は非常に意欲的な課題であると考えられる.

4.2 POMDPの学習問題

POMDPは環境を直接観測できないことでMDPと異なる. 環境状態の有限集合S、行動の有限集合Aとさらに観測の有限集合Oとによって特徴付けられる. ある時点tにおいて、環境状態が$s_t \in S$のとき、Agentは確率$Pr(o_t|s_t)$で観測$o_t \in O$を得る. このとき、Agentのa_tの行動によるReward r_t を受け取り、遷移確率で次の状態s_{t+1}に推移する.ここでのAgentの目標もRewardの割引期待利得の最大化である.ただし、Agentは各時点で、そのときの状態を知ることはできず、観測情報のみが与えられる.そのため、POMDPの問題状況をincomplete perception, hidden stateと呼ぶことがある.

4.3 代表的学習アルゴリズム

1) MDPのアルゴリズムをそのまま用いた方法[5]

通常のMDPのRLをPOMDPに適用する方法である. ある種のクラスのPOMDPに有効に適用できるが、問題によってはゴールへの到達を学習できないこともある.

2) Recurrent Q-learning[6]

Q-値をrecurrent neural networkによって学習する方法で、過去の観測、行動の系列でQ-値を予測する構造になっている.

3) HQ-Learning(Hierarchical Learning)[7]

POMDPな環境において、いくつかのsubgoalのためのQ-learningを用意し、当面のsubgoalに応じてQ-learningを切替える方法である.

4) Classifier system[8),9),10]

Hollandによって提唱されたclassifier system[8]をPOMDPに適用できるように拡張する研究が種々なされている. 代表的なものはWilson,S.W.が提案したZCS[9]がある. ZCSではQ-learningと類似のアルゴリズムを用いて、それにメモリを付け加えてPOMDPに対処する枠組を与えている.さらには、ZCSの改良バージョンのXCS[10]も提案されている.

5. ラベリングQ−学習

5.1 目的

以上の背景をもとに、POMDPにおける強化学習の一つのアルゴリズムとして既存のQ-learningをベースに、環境観測にラベルを付け加えたラベ

リングQ-learningを提案する. Agentが、現在おかれた環境を観測し、その上、その各状態にラベルを付け加えることにより、同じ状態観測値も区別できることになり、より有効な学習を目指す.

5.2 アルゴリズム

POMDPでは、MDPのRLにおける環境状態sの代りに観測$o \in \mathbf{O}$を考える. (Oはsの関数)その観測として、実際の観測値と新たにラベルθを結合させた$\tilde{o}$を最終的な観測と定義する.

$$\tilde{o} = (o,\theta) \in O \times \Theta \quad (14)$$

$$\text{ラベル} \quad \theta \in \Theta = \{1,2,\ldots,\theta_{max}\}$$

5.3 ラベリングのアルゴリズム

ラベリングは様々な方法が考えられるが、本研究では以下のように改変させる.
start時点で、すべての観測に対するラベルを0とする.

$$\theta'(o_{t+1}) \overleftarrow{\theta_{max}} \theta(o_{t+1})+1 \quad (15)$$
$$(\text{for } o_t \neq o_{t+1})$$

$$\theta'(o_{t+1}) \leftarrow \theta(o_t) \quad (16)$$
$$(\text{for } o_{t+1} = o_t)$$

$$\theta(o_{t+1}) \leftarrow \theta(o_t) \quad (17)$$
$$(\text{for } \theta'(o_{t+1}) < \theta(o_t))$$

$$\theta(o_{t+1}) \leftarrow \theta'(o_{t+1}) \quad (18)$$
$$(\text{for } \theta'(o_{t+1}) \geq \theta(o_t))$$

ただし、$\overleftarrow{\theta_{max}}$は右辺の値が$\theta_{max}$以上のとき、左辺の値を$\theta_{max}$と置くことを意味する.

5.3.1 Simulation

1) 環境

本研究のsimulationにはPOMDPの学習問題によく使われるGrid-Worldを用いた. Fig.2はGrid-Worldの一例である.

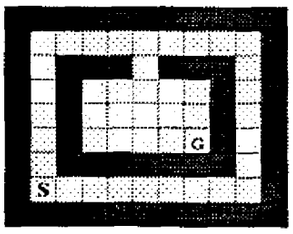

Fig. 2 Simulationに使用されたGrid

2) 観測

本simulationでは、N(North)、S(South)、E(East)、W(West)の4方向観測が可能であるよう設定した. それぞれをビット値に認識し、10進数に換算した値をその状態の値とする.

3) 行動の選択

観測した状態値により、壁が存在し行動が不可能な方向と行動可能な方向とに分け、行動不可能な方向は選択から取り除く. その後、行動可能な方向からε-greedyによって行動の選択を行う. 行動も最大で4方向のみ可能である.

6. Simulationの結果

simulation結果を Fig.3に示す.Fig.3で、実線は通常のQ-学習、一点斜線はラベリングQ-学習である.

ε-greedyに行動選択を持続させた場合、εの確率で、その影響が続くのに比べ、実行の途中から(学習が終了されたと思われる時点)その影響を除く事により、greedyに行動を選択する事になる(Fig.3の実線). その結果効率的な学習効果を得ることができた.

また、試行回数が増すにつれてεを減少させると、Fig.4の結果を得る.(ただし、30回のsimulationの平均である)

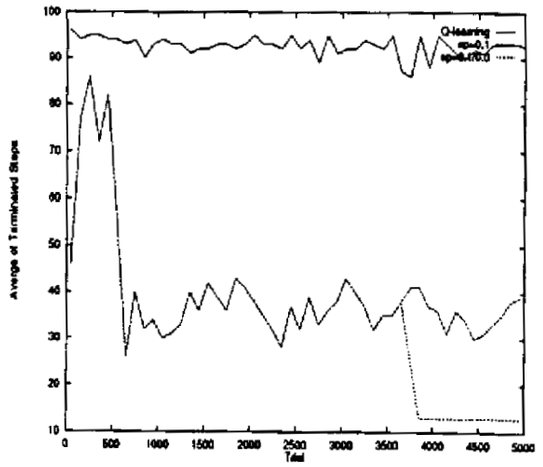

Fig. 3　ラベリングQ学習アルゴリズムのSimulation結果

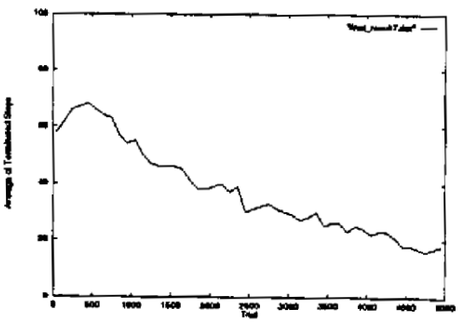

Fig. 4　学習結果

次に、学習の進みを止めた際の学習された行動の傾向はFig.5である.

Fig.5の→は学習された行動を示す.矢印の方向は各gridにおけるgreedy方向を意味する.

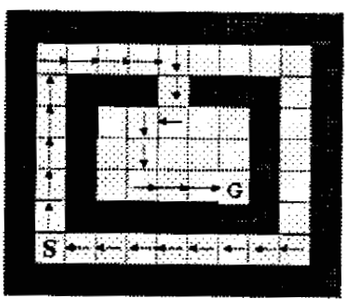

Fig. 5　学習後の行動の傾向

7.　まとめと今後の課題

実際には異なる状態であるが、環境認識能力の制限等によって区別する事ができない、POMDP環境においての強化学習のアルゴリズムとしてラベリングQ-学習を提案し、simulationを行った.

なお、今後の課題として

1) ラベリングQ-learningのアルゴリズムをより複雑で規模の大きい環境に適用できるようラベリングのアルゴリズムを改善する.
2) ラベリングQ-learningを実際の移動ロボットに用いて、その実用性を確認する.

参考文献

1) Watkins,C.J.C.H. and Dayan,P.: Q-learning, Machine Learning, 8,279/292 (1992)
2) Sutton,R.S: Learning to predict by the methods of temporal differences, Machine Learning, vol.3, 9/44 (1988)
3) Loch,J. and Singh,S : Using Eligibility Traces to Find the Best Memoryless Policy in Partially Observable Markov Decision Processes, ICML-98 (1998)
4) Singh,S.P. and Sutton,R,S. : Reinforcement Learning with Replacing Eligibility traces, Machine Learning,3,9/44 (1996)
5) Lin,L and Mitchell,T.M. : Reinforcement learning with hidden states, Proceeding of the Second International Conference on Simulation of Adaptive Behavior (1992)
6) Lin,L.: Reinforcement Learning for Robots Using Neural Networks, Ph.D thesis ,Carnegie Mellon University,Pittsburgh (1993)
7) Marco Wiering and Jürgen Schmidhuber:HQ-Learning,Adaptive Behavior 6:2 (1997)
8) Holland,J.H. : adaptation in Neural and Artificial Systems, University of Michigan Press,Ann Arbor (1975)
9) Wilson,S.W. : ZCS:a zeroth level order classifier system.Evolutionary Computation 1,2,1/18 (1994)
10) Lanzi,P.L. : Solving Problems in Partially Observable Environment with Classifier Systems;Experiments on Adding Memory to XCS,Technical report N (1997)