計測自動制御学会東北支部 第261回研究集会(2010.11.17)
資料番号 261-7

強化学習における統制的探査法

A New Method of Directed Exploration in Reinforcement Learning

○阿部健一*，藤野晃弘**

○Kenichi Abe*, Akihiro Hujino**

＊日本大学工学部，＊＊日本大学大学院工学研究科

*College of Engineering, Nihon University

**Graduate School of Engineering, Nihon University

キーワード: 学習オートマトン(learning automaton)，探査と知識利用(exploration/exploitation)，統制的探査(directed exploration)，縮退指数(degeneracy index)，マルチ学習オートマトン(multi-learning automata)

連絡先: 〒963-8642 郡山市田村町徳定字中河原1 日本大学工学部情報工学科
阿部 健一, TEL: 024-956-8827, E-mail: abe@cs.ce.nihon-u .ac.jp

1. はじめに

強化学習問題[1), 2)]は，許容される政策集合からもっとも環境に適合する政策―最適政策―を選択する問題として定式化される．学習システムがその最適政策を見出すためには，その政策の導出に必要な環境に関する知識を手に入れる必要がある．ここに，学習問題における環境の探査(exploration)と知識利用(exploitation)のトレードオフの問題が生ずる[3)]．この環境探査と知識利用とをうまく按分することが有効な学習アルゴリズムの構築に不可欠である．

強化学習の一つである学習オートマトン[4)]の多くは非常にシンプルで，あらかじめ設定しなければならない2，3のパラメータをもつ．それらのパラメータが上記のトレードオフに深く関わる．たとえば，学習オートマトンL_{R-I}の場合，ステップサイズパラメータを大きく設定すると，十分な探査を行うことなく早く収束してしまい，最適行動(学習オートマトンの場合，最適行動を選ぶことが最適政策である)の獲得率が減少する．しかし，ステップサイズパラメータをあまり小さく設定すると，最適行動の獲得率は上昇するが収束が遅くなる．

多くの場合，学習オートマトンのパラメータは一定値に設定され，学習の途中で変更されることはない．そのためには，あらかじめ類似の問題についてシミュレーション実験を行って，適切なパラメータ値を探っておく必要がある．これに対し，学習の進行状態に応じてパラメータ値をオンラインで変更する方法が考えられる．この方法をここでは統制的探査(directed exploration)ということにする．

本稿では，モンテカルロ法における確率分布の縮退指数(degeneracy index)の考え方を利用することで一種の統制的探査が行えることを示す．まず，複数学習オートマトンを用意しておき，各時点において，その一つをsoft-max法によって選択し起動して行動選択を行う．この学習システムをマルチ学習オートマトン(MLA: multi-learning automata)と名付ける．soft-max

法では温度が上記のトレードオフに関わるパラメータである. 各学習オートマトンの学習状態を縮退指数によって評価し, それらの一つがある閾値より小さくなったら, 温度を下げる. この方法で, 一種の統制的探査法(directed exploration)を実現する. この方法の有効性をシミュレーションによって検証する.

2. 学習オートマトンの概要

図1は学習オートマトン(LA)とその環境との相互関係を表している.

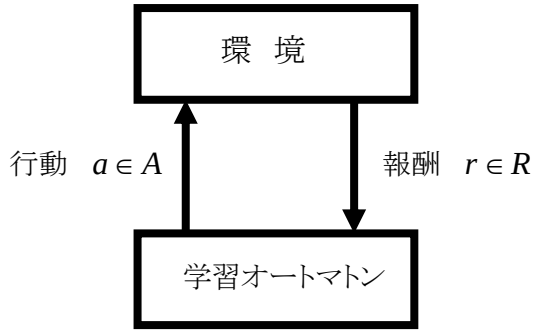


図1 学習オートマトンの概念図

Fig. 1 Conceptual Scheme of Learning Automaton

ある時点において学習オートマトンが行動 $a \in A = \{a_1, a_2, \ldots, a_n\}$ を選択し, それを実行すると, 環境から報酬(reward) $r \in R$ (実数)を受け取る. ただし, 各行動に対して報酬は未知の確率分布にしたがってランダムに発生する. 学習オートマトンは行動と報酬に基づいて, 自身が持つ行動選択確率を更新する. これを繰り返すことによって, 報酬の期待値を最大にする行動(最適行動)を獲得することが学習オートマトンの目的である.

報酬 r は, ある未知の関数 f によって次のように生成される.

$$r = f(a) + \varsigma \qquad (1)$$

ここで, ς は平均値ゼロのnoiseである. f を最大にする行動が最適行動である.

学習オートマトンが行動 $a_i \in A$ を選択する確率を π_i とする.

$$\sum_{i=1}^{n} \pi_i = 1$$

ベクトル $\boldsymbol{\pi} = (\pi_1, \pi_2, \cdots, \pi_n)$ を行動確率ベクトルという. 学習オートマトンは, この行動確率ベクトルを更新するアルゴリズムによって特徴付けられる. ここでは次のアルゴリズムを取り上げる.

行動 a に対して報酬 r を受け取ったとき, 最適行動の価値の推定値 v を次式で更新する.

$$\begin{aligned} v &\leftarrow v + \beta\delta \\ \delta &= r - v \end{aligned} \qquad (2)$$

δ を時間的差分(TD:temporal difference)誤差という. このTD誤差を用いて, 行動確率ベクトルを次の式で更新する.

$$\pi_i \leftarrow \pi_i + \alpha I(\delta > 0)(I(a = a_i) - \pi_i) \qquad i = 1,2,\cdots,n \qquad (3)$$

ここで, $I(\bullet)$ は, 事象 $\bullet$ が真なら1, 偽なら0をとる関数(インジケータ)である. α, β をステップサイズパラメータという.

この学習アルゴリズムに従うと, 行動確率ベクトルは最適行動に対応する要素が1に近づき, v は最適行動に対する平均報酬に漸近する(ただし, 学習オートマトン理論における ε 最適の意味で).

3. 行動確率ベクトルの縮退と探査能力

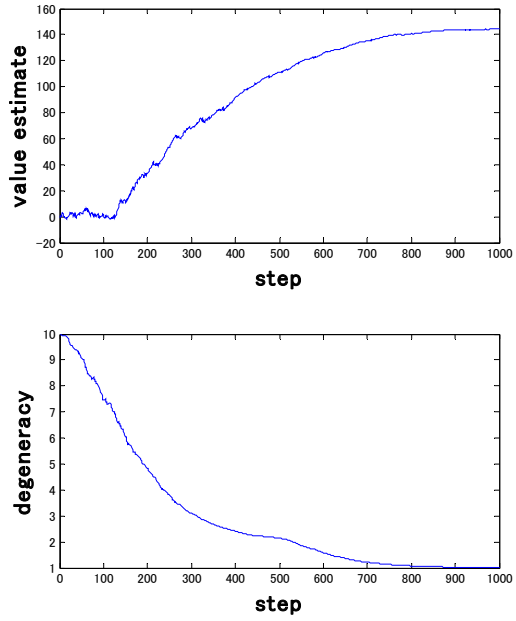
行動確率ベクトルのある要素に着目したとき, それがゼロであれば, その要素に対応する行動は選択されない. ゼロに近い要素が多いほど, 環境の探査(exploration)は限定的となる. 行動確率ベクトルのこのような現象を次の式で評価することができる. 行動確率ベクトル $\boldsymbol{\pi} = (\pi_1, \pi_2, \cdots, \pi_n)$ について

$$\hat{N} = \frac{1}{\sum_{i=1}^{n} (\pi_i)^2}$$

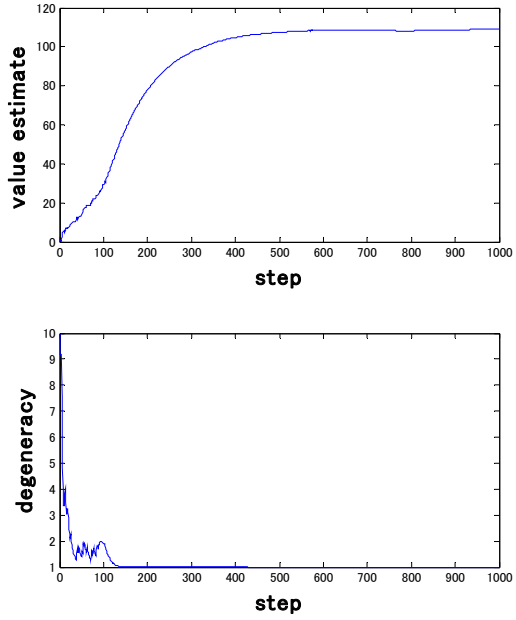
これは, モンテカルロ法における, 確率分布の縮退指数(degeneracy index)[5]として知られる量である. $1 \le \hat{N} \le n$ を満たす. 1に近いほど, 分布は縮退していることを意味する.

シミュレーション例 1 図2は, 最適行動における報酬平均が145.33の $n = 10$ の環境に対する

学習結果で，vと$\hat{N}$の変化を示した．ただし，アルゴリズムにおけるパラメータα, βを0.01, 0.1とした場合(a)と0.1, 0.1の場合(b)を示した．$\alpha = 0.1$のときは約60ステップ以降で，$\hat{N} < 1.1$となり，約100ステップ以降で$\hat{N} < 1.01$である．一方，$\alpha = 0.01$のとき，約



(a)　$\alpha = 0.01$



(b)　$\alpha = 0.1$

図2　価値推定と縮退指数の推移

Fig. 2　Change of Value Estimate and Degeneracy Index

700ステップ以降で$\hat{N} < 1.1$となるが，1000ステップでも$\hat{N} > 1.01$である．αが大きくなるにつれ，探査能力が減少する．また，αが大きいほど，最適行動への収束割合が減少する．シミュレーションによれば，たとえば，$\alpha = 0.01$のとき，ほぼ100パーセントで最適行動に収束するが，$\alpha = 0.1$のとき約44パーセントである(いずれも1000試行における割合)．

4. マルチ学習オートマトン

行動集合Aの学習オートマトンを考えたとき，オートマトンの価値推定vと行動確率ベクトル$\boldsymbol{\pi}$でそのオートマトンの状態が表せる．つまり

$$\mathscr{A} = \{v, \boldsymbol{\pi}\}$$

いまcl個の学習オートマトン

$$\mathscr{A}_j = \{v_j, \boldsymbol{\pi}_j\}, \quad j = 1, 2, \cdots, cl$$

を用意しておき，それらの一つをある方法で選択しながら学習を進めるマルチ学習オートマトン(MLA)を考えてみよう．ここでは，選択を次のBoltzmann分布で行う．

$$p_j = \frac{e^{v_j/\tau}}{\sum_{i=1}^{cl} e^{v_i/\tau}} \quad \text{for} \quad j = 1,2,\ldots,cl \quad (3)$$

この分布による行動選択法をsoft-max法ともいう．ここで，τは温度と呼ばれる正定数である．$\tau \to 0$の極限でグリーディ法と一致する．

マルチ学習オートマトン(MLA)アルゴリズム

ステップ1　各学習オートマトンと価値推定vとを初期化する．温度τを設定する．

ステップ2　soft-max法により学習オートマトンの一つを選ぶ．それを$\mathscr{A}_j$とする．

ステップ3　行動確率ベクトル$\boldsymbol{\pi}_j$により行動選択を行う．選択された行動とその行動に対する報酬rによって，(2)式でv_jを，(3)式で$\boldsymbol{\pi}_j$を更新する．

ステップ4　報酬rを用いて，(2)式でMLAの価値推定vを更新する．

ステップ5　ステップ2，3，4を与えられたステップ数(maxstep)だけ繰り返す．

シミュレーション例 2

$n=10$とし，シミュレーション例1と同じ環境に対し，$\alpha=0.1$，maxstep=1,000，$\tau=50$，$cl=6$のMLAによる学習を100試行行う．図3は報酬(MLAの価値推定ではなく)の100試行の平均をプロットしたものである．平均的にかなり良い結果になっている．100試行のうちの60試行において，最終ステップまでに6個のオートマトンのうち1個以上のオートマトンが最適行動の発見に成功している．ただし，ある1回の試行結果を見ると(図4)，報酬の低い学習オートマトンもある割合で選択されていることが分かる．これは温度が高いことによる．

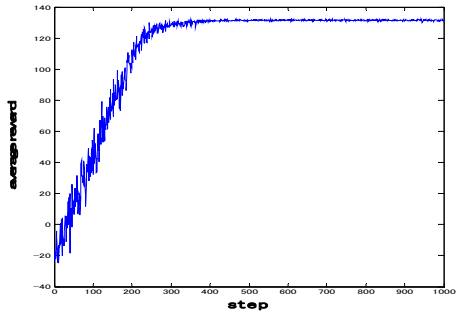


図3　MLAによる平均報酬の推移(100試行平均)

Fig. 3 Change of Average Reward by MLA (Average of 100 trials)

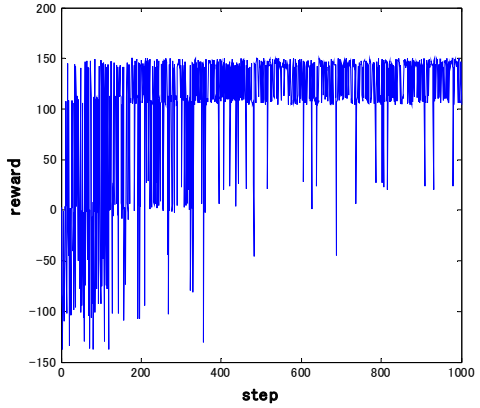


図4　MLAによる報酬の推移(1試行)

Fig. 3 Change of Reward (1 trial)

一方，各試行において，degeneracy $\hat{N}$ に着目すると，MLAのどれか一つはかなり早いステップで1に漸近している．このことを温度の更新に利用してみよう．

5. 縮退指数利用による統制的探査法

MLAアルゴリズムのステップ2を次のように書き換える．

統制的探査法を組み込んだMLAアルゴリズム

ステップ2’　もし

$$\hat{N} < \hat{N}_0$$

ならば，温度τをτ_0($<\tau$)に変更し，soft-max法により学習オートマトンの一つを選ぶ．それを$\mathcal{A}_j$とする．

シミュレーション例 3

$n=10$とし，シミュレーション例1と同じ環境に対し，$\alpha=0.1$，maxstep=1,000，$\tau=50$，$\tau_0=10$，$cl=6$，$\hat{N}_0=1.001$として，100試行行う．図5は報酬の100試行の平均をプロットしたものである．約70%で最適行動に収束する．

図6に1試行の結果を示す．ステップ2’の操作によって，温度が低くなった時点からほぼ最適行動を選択している様子が分かる．

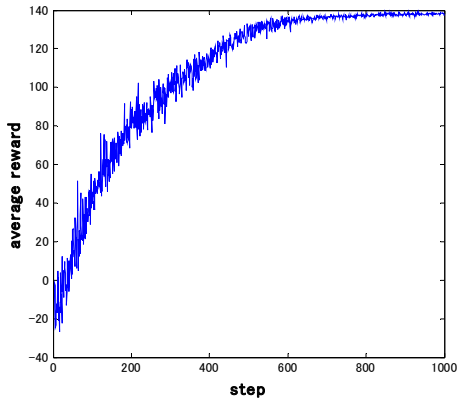


図5　統制的探査法を組み込んだMLAによる平均報酬の推移(100試行平均)

Fig. 5 Change of Average Reward by MLA with Directed Exploration (Average of 100 trials)

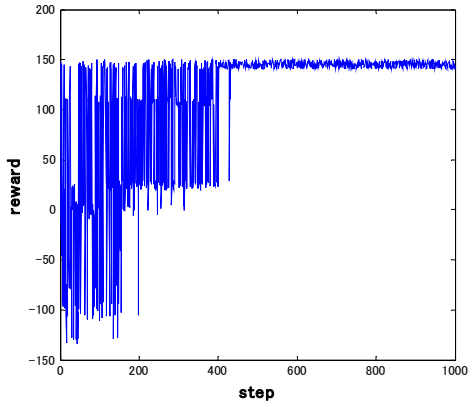


図6　統制的探査法を組み込んだMLAによる報酬の推移(1試行)

Fig. 6 Change of Reward by MLA with Directed Exploration(1 trial)

閾値 $\hat{N}_0$ を大きくすると，低い温度に早めに切り替わる．図7は $\hat{N}_0 = 1.5$ とする場合の結果である(他の実験条件は図5の場合と同じ)．図5と比較すると，平均報酬が少し低く，最適行動への収束が約5%ほど落ち込んでいる．

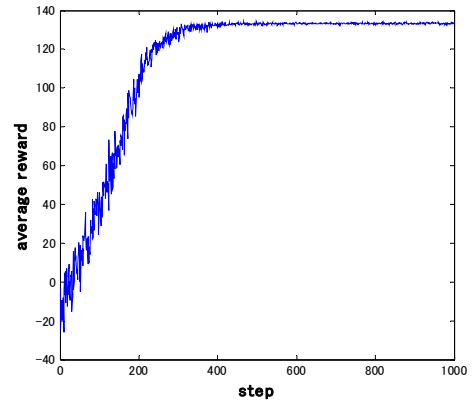


図7　$\hat{N}_0 = 1.5$ のMLAによる平均報酬の推移(100試行平均)

Fig. 7 Change of Average Reward by MLA with $\hat{N}_0 = 1.5$ (Average of 100 trials)

6. おわりに

本稿では，行動数が少ない場合を扱った．行動数が少ないと，単独の学習オートマトンでもいい学習性能が容易に達成できる．たとえばステップサイズパラメータをステップ数が増えるに従い減少させる，などの方法をとればよい．

一方，実応用問題の多くは行動が連続値をとるので，学習オートマトンを適用する場合，行動を離散化することになる．離散化は行動空間の次元が大きくなると，行動数が指数的に増大し，いわゆる規模の障害に陥る．この問題は，いわゆるマルコフ環境における強化学習(Q学習やSARSA[1),2)])では，状態，行動対を扱うため，より一層厄介な問題である．

著者らはさきに，Actor-Critic法と呼ぶ強化学習アルゴリズムにおいて，Actorの並列化を導入することで規模の問題の解決を試みた[6)]学習オートマトン問題の場合，この並列アクターの働きは，並列学習オートマトン[4)]にほかならない．この並列学習オートマトンは計算量の点で有効な方法であるが，いわゆるNash解に陥り，最適行動に至らないことがある．

そこで，マルチ学習オートマトンと同様に，並列学習オートマトンのマルチシステムを構成し，本稿の方法を適用することにより，学習性能の向上が望める．その検討については今後の課題としたい．

参考文献

1) R. S. Sutton and A. G. Barto: Reinforcement Learning-An Introduction, The MIT Press, 1998.

2) 阿部：強化学習―価値関数推定と政策探索，計測と制御，**41**-9, 2002.

3) 阿部：強化学習における二つのジレンマ，SICE東北支部40周年記念学術講演会，2004.

4) M. A. L. Thathachar and P. S. Sastry: Varieties of Learning Automata: An Overview, IEEE Trans. SMC-B, **32**-6, 2002

5) M. S. Arulampalam, S. Maskell, N. Gordon, N.; and T. Clapp: A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, IEEE Trans. Signal Process., **50**-2, 2002.

6) 藤野・佐藤・阿部：並列結合アクターによる強化学習，平成22年度電気関係学会東北支部連合大会，2010.