ニューロ・ダイナミック・プログラミングとは
—— 動機づけ、解析、発展

蔵野正美 (千葉大学)、堀口正之 (弓削高専)、安田正實 (千葉大学)

1 はじめに

ニューロ・ダイナミック・プログラミングの主題は不確実性の下でおこなう逐次決定過程あるいは確率制御問題である。つまり、その過程の進展が決定（戦略）または制御によって影響を受け、コントロールされた動的システムをもっているものである。各々の時間で下された決定は一般にシステムの状態に依存し、目的は、ある定められた実行基準を最適化する意思決定規則 (フィードバック政策) を選択することである。このような問題はダイナミック・プログラミングの古典的方法を使用して、原理的なものとしては解決することができよう。

しかしながら、実際上、多くの重要な問題へのダイナミック・プログラミングの適用可能性は、対象とするな状態空間の巨大なサイズによって制限される。いわゆる R. Bellman 「次元の呪い (Curse of dimensionality)」あるいは「モデリングの呪い (Curse of modeling)」とよばれる。ニューロ・ダイナミック・プログラミング、あるいは人工知能分野の中で使用される用語では「強化学習（Reinforcement Learning)[*1] 」) とは、ダイナミック・プログラミングの適用可能性について、そのようなボトルネックを克服するためにニューラル・ネットアーキテクチャー (neural architecture) および近似アーキテクチャー (approximate architecture) を使用することである。複雑性に対して「近似」を提案する。方法論としては、システムがシミュレーションによってそれらの振る舞いを学習し、それらのパフォーマンス効率を反復される強化によって改善していこうと試みる。

1 つのアプローチ「価値関数 (value function) の近似」では、シミュレーションが状態空間の異なる状態の相対的な望ましさ（relative desirability）の量を計る「価値関数」のパラメータを調整するために用いられる。数学的な用語でいえば、目的は Bellman の方程式への近似解を計算することである。そして、それは近似最適政策 (sub-optimal policy) を構築するために使用される。このアプローチは 1996 年の本、ニューロ・ダイナミック・プログラミング

[BerTsi96] Neuro-Dynamic Programming, by Dimitri P. Bertsekas and John N. Tsitsiklis, 1996, Athena Scientific, Optimization adn Computation Series, ISBN 1-886529-10-8, 512 pages

[*1] 強化学習 (きょうかがくしゅう, Reinforcement Learning) とは、ある環境内におけるエージェントが、現在の状態を観測し、取るべき行動を決定する問題を扱う機械学習の一種。エージェントは行動を選択することで環境から報酬を得る。強化学習は一連の行動を通じて報酬が最も多く得られるような方策 (policy) を学習する。ここでの環境とは、有限状態数のマルコフ決定過程として定式化される。また、強化学習は動的計画法の一種としても位置づけられる。強化学習は、学習のための適切な入力データと出力データのペアが与えられることがない、という意味からすると、教師あり学習とは異なる学習手法である。また、未知の学習領域を開拓していく行動と、既知の学習領域を利用して行動とをバランス良く選択することができるという特徴も持っている。その性質から未知の環境下でのロボットの行動獲得に良く用いられる。(Wikipedia より)

の中で研究され、それは他の分野に利用されていない多くの結果を含んでいる。また別のアプローチ「政策空間 (policy space) の最適化」は、改良の方向に政策パラメータのチューニングを含むものである。

この問題領域のほとんどは当然、理論的なものであり、数種のアルゴリズムに対して、その劣最適性や収束性を明らかにすることを目的としているが、特定領域に関しては実際、この方法論によるさまざまな応用として研究されているものが多く知られている。ここでは、前述の [BerTsi96] Bertsekas and Tsitsiklis, 1996 をもとに、ニューロ・ダイナミック・プログラミングを紹介する。また同氏の HP には文献が多く掲載されていて有用である。http://web.mit.edu/dimitrib/www/home.html を参照されたい。

検索できる強化学習 HP：http://www.cs.ualberta.ca/ sutton/book/the-book.html

RL FAQ 日本語版: 強化学習についてのよくある質問と答え http://nao.s164.xrea.com/RL-FAQ-j.html

強化学習の入門としての資料：[SutBar98] Reinforcement Learning: An Introduction, by Richard S. Sutton and Andrew G. Barto. MIT Press 1998. Online version. [訳注] 日本語訳は以下の通り．強化学習， by Richard S. Sutton and Andrew G. Barto. 三上 貞芳・皆川 雅章 共訳. 森北出版 2000.

教科書レベルの長さの扱いをする時間がなければ，以下の 2 つの論文のどちらかが引用される:

[KaeLit96] Reinforcement learning: A survey, by Kaelbling, L.P., Littman, M.L., and Moore, A.W., in the Journal of Artificial Intelligence Research, 4:237–285, 1996.

[BarSutWat90] Learning and sequential decision making, by Barto, A.G., Sutton, R.S., & Watkins, C.J.C.H., in Learning and Computational Neuroscience, M. Gabriel and J.W. Moore (Eds.), pp. 539–602, 1990, MIT Press.

2 確率最短経路問題

はじめに掲げている動的計画法の問題は最短経路問題である。いわゆる動的計画問題としての一般的な定式化を述べる。離散的な dynamic system では $\pi = \{\mu_0, \mu_1, \ldots\}$, $\mu_k(i) \in U(i)$(finite set) policyπ が固定されると, i_k は次式の確率をもつ Markov chain になる。

$$P(i_{k+1} = j | i_k = i) = p_{ij}(\mu_k(i)).$$

状態空間は $\{1, 2, \ldots, n\}$ とし、特別に terminal state を 0 が与えられているとする。この特別状態は吸収壁の意味をもたせ、後ほどどんな定常政策をとっても、必ず終端することを仮定する。k 番目の推移において cost $\alpha g(i, u, j)$ が課せられる。$0 < \alpha \leq 1$ は割引率、 g は所与の利得関数とする。したがって有限計画期間問題は有限なある数 N と初期状態 i から始まる政策 π の期待利得としては

$$J_N^\pi(i) = E\left[\alpha^N G(i_N) + \sum_{k=0}^{N-1} \alpha^k g(i_k, \mu_k(i_k), i_{k+1}) \middle| i_0 = i\right],$$

ここで $\alpha^N G(i_N)$ は state i_N における費用で最適な N-stage cost-to-go は次式で定義される.

$$J_N^*(i) = \min_\pi J_N^\pi(i).$$

また無限計画期間問題の場合は

$$J^\pi(i) = \lim_{N\to\infty} E\left[\sum_{k=0}^{N-1} \alpha^k g(i_k, \mu_k(i_k), i_{k+1}) \middle| i_0 = i\right].$$

$$J^*(i) = \min_\pi J^\pi(i)$$

となる。

Definition 2.1 定常政策 π が *proper* であるとは, 初期状態に関わらず多くとも n 推移で *terminal state* に至る確率が正であること, つまり

$$\rho_\mu = \max_{i=1,\dots,n} P(i_n \neq 0 | i_0 = i, \mu) < 1. \tag{2.1}$$

定常政策が *proper* でない場合 *improper* という。

Assumption 2.1 (i)　少なくとも一つの *proper policy* が存在する.
(ii)　すべての *improper policy* μ に対して, $J^\mu(i)$ は少なくとも一つの状態 i で収束しない.

いま作用素 T を導入して、 $TJ(i), T_\mu J(i), i = 1, \dots, n$ はそれぞれ次式で定義する。

$$(TJ)(i) := \min_{u \in U(i)} \sum_{j=0}^{n} p_{ij}(u)(g(i,u,j) + \alpha J(j)), \tag{2.2}$$

$$(T_\mu J)(i) := \sum_{j=0}^{n} p_{ij}(\mu(i))(g(i,\mu(i),j) + \alpha J(j)). \tag{2.3}$$

行列 P_μ の成分を $p_{ij}(\mu(i))$ とおいて

$$T_\mu J = g_\mu + \alpha P_\mu J,$$

が成り立つ。定常政策 μ の期待利得 J^μ と最適利得 J^* は, $\alpha \in [0,1)$ のとき、あるいは $\alpha = 1$ で Assumption 2.1 のもとで、それぞれ T_μ, T の唯一の不動点になることが知られている。また $T_{\mu^*} J^* = TJ^*$ をみたす μ^* は最適政策である。この (J^*, u^*) を求めるアルゴリズムとして、政策改良法 (policy iteration) が知られているが、ここでは Neuro DP の中心的手法 TD 法 (Temporal-Difference method) の考え方を用いた $\lambda-$ 政策改良法を述べる。以下 Π_S を定常政策の全体を表す。

$\lambda-$ 政策改良法 $(0 \leq \lambda < 1)$:

1. 初期値 (J_0, μ_0), $J_0 \in \mathbb{R}^n, \mu_0 \in \Pi_S$.
2. $k(\geq 0)$ ステップ値 (J_k, μ_k) が与えられたとせよ.
   (a) $T_{\mu_{k+1}} = TJ_k$ を満たす $\mu_{k+1} \in \Pi_S$ を選べ.
   (b) $J_{k+1} := J_k + \Delta_k$ ただし $\Delta_k = (\Delta_k(1), \Delta_k(2), \cdots, \Delta_k(n)) \in \mathbb{R}$,
   $\Delta_k(i) = \sum_{m=0}^{\infty} E_{\mu_{k+1}}[(\alpha\lambda)^m d_k(i_m, i_{m+1}) \,|\, i_0 = i]$ $(i \in S)$,
   $d_k(i,j) = g(i, \mu_{k+1}(i), j) + \alpha J_k(j) - J_k(i)$ (Temporal Defference)
3. $k := k + 1$ として Step 2 へ。

この式において、もし $\lambda = 1$ とすれば、通常の政策改良法に帰着される。さらにつぎの結果 (p.45) が得られている。

Theorem 2.1 任意の $\lambda \in (0,1)$ に対して、つぎが成り立つ。

(a) $0 < \alpha < 1$ のとき、

$$(J_k, \mu_k) \longrightarrow (J^*, \mu^*)\,(k \to \infty) \tag{2.4}$$

ある $\overline{k}$ が存在して、$k \geq \overline{k}$ では

$$\|J_{k+1} - J^*\|_\infty \leq \frac{\alpha(1-\lambda)}{1-\alpha\lambda}\|J_k - J^*\|_\infty \tag{2.5}$$

(b) $\alpha = 1$ のとき、*Assumption 2.1* のもとで *(2.4)* が成り立つ。

3 収束性

作用素 $H : \mathbb{R}^n \to \mathbb{R}^n$ の不動点を求めるための確率近似 (逐次) 法 （Stochastic approximation, stochastic iterative method）を取り上げよう。つぎの 2 つの定理は TD 法による学習アルゴリズムの収束定理を証明する基礎的な道具を与える。この節では Bertsekas & Tsitsiklis([1]) の結果を一部紹介する。

利得を意味する $r_t = (r_t(1), r_t(2), \cdots, r_t(n)) \in \mathbb{R}^n\,(t \geq 0)$ はつぎの update equation によって生成される：

$$\begin{aligned} r_{t+1}(i) &= (1-\gamma_t(i))r_t(i) + \gamma_t(i)(Hr_t(i) + w_t(i)) \\ &= r_t(i) + \gamma_t(i)(Hr_t(i) - r_t(i) + w_t(i)) \end{aligned} \tag{3.1}$$

ただし $w_t = (w_t(1), w_t(2), \cdots, w_t(n)) \in \mathbb{R}^n$ は random vector であり、$\gamma_t(i)$ はステップサイズを表す。

つぎの 2 つの定理が martingale の収束定理を応用して証明されている。

Theorem 3.1 *(Contractive case, p.155, 157)* つぎの (a) ∼ (c) を仮定する。

(a) $\gamma_t(i) \geq 0,\ \sum_{t=0}^{\infty}\gamma_t(i) = \infty,\ \sum_{t=0}^{\infty}\gamma_t^2(i) < \infty$

(b) $E[W_t(i)\,|\,\mathcal{F}_t] = 0$ かつ、ある正の定数 A, B が存在して $E[W_t^2(i)\,|\,\mathcal{F}_t] \leq A + B\|r_t\|^2$ となる。ただし $\mathcal{F}_t = (r_\ell(i), \ell \leq t, W_\ell(i), \ell \leq t-1, \gamma_\ell(i), \ell \leq t-1, i = 1, 2, \cdots n)$

(c) $r^* \in \mathbb{R}^n$ と $\beta \in (0,1)$ が存在して $\|Hr_t - r^*\| \leq \beta\|r_t - r^*\|,\ \forall t$

これらの下では, *(3.1)* で定まる $\{r_t\}$ に関して、r_t は $t \to \infty$ で r^* へ確率 *1* で収束する。

Theorem 3.2 *(Monotone case, p.154)*

(a) 定理 *3.1* の (a), (b) が成り立つ。

(b) つぎの (i) ∼ (iii) が成り立つ。

(i) *H:monotone,* つまり $r \leq \overline{r}$ ならば、$Hr \leq H\overline{r}$.

(ii) $Hr^* = r^*$ なる r^*、 つまり不動点は一意に存在する。

(iii) $e = (1, 1, \cdots, 1) \in \mathbb{R}^n$ と任意の $\eta > 0$ に対して

$$Hr - \eta e \leq H(r - \eta e) \leq H(r + \eta e) \leq r\eta e,\ (r \in \mathbb{R}^n).$$

このとき r_t が一様有界ならば確率 1 で $r_t \to r^*\,(t \to \infty)$ が成り立つ。

定理 3.1, 定理 3.2 のマルコフ決定過程 (第 2 節の確率最短経路問題では吸収壁 terminal state 0 が必ずしも存在しない場合) への適用例をみてみよう。

Optimistic $TD(0)$: MDP の sample path $(i_0, i_1, \cdots\cdots)$ に対して、つぎの update equation を考える。

$$J_{t+1}(i_t) = (1-\gamma_t(i_t))J_t(i_t) + \gamma_t(i_t)\Big(g(i_t,\mu_t(i_t),i_{t+1}) + \alpha J_t(i_{t+1})\Big) \tag{3.2}$$

ただし μ_t は J_t に対する greedy policy で, $T_{\mu_t}J_t = TJ_t$ を満たすとし、さらに i_{t+1} は i_t が与えられたとき、P_{μ_t} による state transition の実現値とする。

上記の (3.2) は次のように書き換えられる。

$$J_{t+1}(i_t) = (1-\gamma_t(i_t))J_t(i_t) + \gamma_t(i_t)\Big(TJ_t(i_t) + W_t(i_t)\Big) \tag{3.3}$$

ただし

$$W_t(i_t) = g(i_t,\mu_t(i_t),i_{t+1}) + \alpha J_t(i_{t+1}) - TJ_t(i_t)$$

定理 3.2 の条件をチェックして、つぎを得る。

Proposition 3.1 $\alpha \in (0,1)$ とする。

(a) $\gamma_t(i) \geq 0,\ \sum_{t=0}^{\infty}\gamma_t(i) = \infty,\ \sum_{t=0}^{\infty}\gamma_t^2(i) < \infty$

(b) sample path $(i_0, i_1, \cdots)$ の中ですべての状態が確率 *1* で無限回生起する。

このとき、確率 *1* で $(J_t,\mu_t) \to (J^*,\mu^*)(t\to\infty)$ が成り立つ。

Q-factor による value iteration アルゴリズム:(p.245) よく知られた value iteration と同様な方法として

$$Q^*(i,u) = \sum_{j\in S} p_{ij}(u)\Big(g(i,u,j) + \alpha J^*(j)\Big) \tag{3.4}$$

が述べられている。ただし J^* は optimal value function とする。このとき、いわゆる Bellman's equation が得られることとなる。

$$J^* = \min_{u\in U(i)} Q^*(i,u) \tag{3.5}$$

この (3.4),(3.5) の式からはつぎが成立する。

$$Q^*(i,u) = \sum_{j\in S} p_{ij}(u)\Big(g(i,u,j) + \alpha \min_{v\in U(j)} Q^*(j,v)\Big) \tag{3.6}$$

このようにして得られた方程式 (3.6) に対する stochastic iteration アルゴリズム (Q-learning) はつぎで与えられる。

$$Q_{t+1}(i_t,u_t) = (1-\gamma_t(i_t,u_t))Q_t(i_t,u_t) + \gamma_t(i_t,u_t)\Big(g(i_t,u_t,j_{t+1}) + \alpha \min_{v\in U(j_t)} Q_t(j_{t+1},v)\Big) \tag{3.7}$$

本文２４６ページの（５．６０）式ではつぎのような表現形式をもちいる：

$$Q(i,u) := (1-\gamma)Q(i,u) + \gamma(i,u)\Big(g(i,u,j) + \alpha \min_{v\in U(j)} Q(j,v)\Big) \tag{3.8}$$

ただし (i_t,u_t) は simulated transition で、さらに i_{t+1} は (i_t,u_t) が与えられたときの $p_{i_t,\cdot}(u_t)$ による実現値を表す。

つぎがこの Q-learning の収束定理を表し、いままでの経験的な数値計算のみではなく、厳密な証明を与え、理論的に裏づけを与えたものとされる。

Theorem 3.3 *2つの仮定：*

(a) $\gamma_t(i) \geq 0,\ \sum_{t=0}^{\infty} \gamma_t(i) = \infty,\ \sum_{t=0}^{\infty} \gamma_t^2(i) < \infty,$

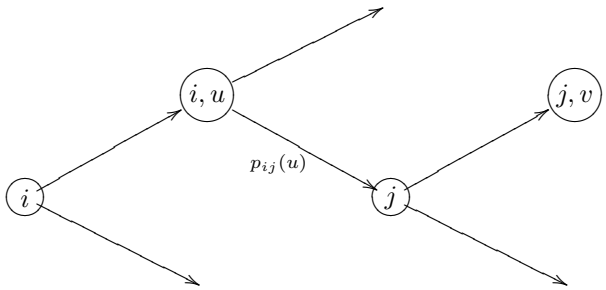
(b) Simulated transition $(i_t, u_t), (t = 0, 1, 2, \cdots)$ *において確率 1 で任意の* $(i, u)\,(i \in S, u \in U(i))$ *が無限回生起する,*

があれば、$\forall i \in S, \forall u \in U(i)$ について

$$Q_t(i,u) \to Q^*(i,u) \quad \textit{with probability 1 as } t \to \infty$$

の収束が成り立つ。

ここでの注意として、$\alpha = 1$ のとき、 Assumption 2.1 のもとでの収束性は定理 3.2(monotone case) を適用して証明されるが、これは一般化したものである。



4 例題

[KonTsi03] V. R. Konda and J. N. Tsitsiklis, "Actor-Critic Algorithms" , SIAM Journal on Control and Optimization, Vol. 42, No. 4, 2003, pp. 1143-1166. Appendix. に取り上げている例題 4.7 pp.1153 は在庫問題で、いわゆるステファン（自由境界）問題のタイプであり、政策の閾値を定める値も未知になる場合である。変分不等式の分野でもよく知られている典型的な問題である。

ある施設では期間 k では在庫 X_k を抱えており、不足の場合（負の在庫、欠品）のこともペナルティとして考慮して、バックログを許すことにする。D_k で k 期のランダムな在庫量を表し、問題は現在の在庫と直前の需要（注文量）から、どのくらいの量を注文して、次期の在庫とすべきかを考える。このための費用をつぎで定める：

$$c(X_k, U_k) = h \max(0, X_k) + b \max(0, -X_k) + pU_k$$

ここで p は単位あたりの材料購入費用,b はバックログにかかるコストで、h を在庫として保持しておくことにかかる費用である。在庫の変化は、動的なシステムとして、

$$X_{k+1} = X_k + U_k - D_k, \quad k = 0, 1, \ldots$$

ここでの確率変数 D_k は非負の独立同一分布に従い、有限な平均をもつとする。最適政策は、適当な S があって

$$\mu^*(x) = \max(S - x, 0)$$

で与えられる。この S も未知数であるから、自由境界問題に最適方程式が帰着される。特に状態空間は実数の連続値であるから、確率過程が一様な幾何的エルゴード性（正確には、終端期での分布 $X_N \in B,\ B \in \mathcal{B}(\mathcal{X})$

の値が下からある確率測度で抑えられていることと、変動の評価として確率版のリアプノフ関数が存在することを仮定している）のもとで、最適政策に対応したマルコフ連鎖が既約となる。

5 方法論

5.1 政策空間および actor-critic アルゴリズム

価値関数のパラメータを調整する代わりに、パラメータで政策のクラスが記述されるものとして、これを直接、政策パラメータで調整できるであろうか？推定 Q-因子の用語で解釈できるもののクラスに対しては研究されている：

[MarTsi01] P. Marbach and J. N. Tsitsiklis, "Simulation-Based Optimization of Markov Reward Processes," IEEE Transactions on Automatic Control, Vol. 46, No. 2, pp. 191-209, February 2001.

しかしこの方法では大きな分散や緩い収束に苦しむかも知れない。だが部分的な変形によって (例えば割引係数の導入によって) 緩和することができる:

[MarTsi03] P. Marbach and J. N. Tsitsiklis, "Approximate Gradient Methods in Policy-Space Optimization of Markov Reward Processes", Journal of Discrete Event Dynamical Systems, Vol. 13, pp. 111-148, 2003. (preliminary version: "Simulation-based optimization of Markov reward processes: implementation issues," in Proceedings of the 38th IEEE Conference on Decision and Control, December 1999, pp. 1769-1774.)

さらにより効率を求めるならば、価値関数近似と政策空間の学習を組み合わせることができるかという問題になろう。これは、すなわち俳優批評家アルゴリズム法が強調して目指すものである。その結果、一旦政策パラメータ化ができたならば、価値関数近似の中で使用される「特徴（features)」の自然な集まりが規定されることになり、また、1 つは、相応しい収束性を備えたアルゴリズムが得られる:

[KonTsi03] V. R. Konda and J. N. Tsitsiklis, "Actor-Critic Algorithms" , SIAM Journal on Control and Optimization, Vol. 42, No. 4, 2003, pp. 1143-1166. Appendix.

[KonTsi99] (前論文 [KonTsi03] の準備段階) V. R. Konda and J. N. Tsitsiklis, "Actor-Critic Algorithms" , in Advances in Neural Information Processing Systems 12, Denver, Colorado, November 1999, pp. 1008-1014.

上記の論文に用いられている MDP と政策のパラメータ化：

既約で非周期的なマルコフ連鎖に対して有限状態: X, 決定空間:U, 一期間費用関数:c、$p(y|x,u)$:推移確率, μ:政策とする。ここでベクトルパラメータ θ を導入し、$\eta_\theta(x,u) = \pi_\theta(x)\mu_\theta(u|x)$ 平均利得は極限確率を用いて：$\overline{\alpha}(\theta) = \sum_{x,u} c(x,u)\eta_\theta(x|u)$ と表され、また過平均利得 V_θ についてはポアソン方程式とよばれる方程式を満たす：

$$\overline{\alpha}(\theta) + V_\theta(x) = \sum_u \mu_\theta(u|x) \left[c(x,u) + \sum_y p(y|x,u) V_\theta(y) \right]$$

この文献では、これは将来における超過費用で、不利益なものとみなせると述べられている。このとき、Q 値関数を

$$Q_\theta(x,u) = c(x,u) - \overline{\alpha}(\theta) + \sum_y p(y|x,u) V_\theta(y)$$

と定めると、つぎの結果が得られる：

Theorem 5.1

$$\nabla\overline{\alpha}(\theta) = \sum_{x,u} \eta_\theta(x,u) Q_\theta(x,u) \psi_\theta(x,u) \quad \text{ただし} \quad \psi_\theta(x,u) := \nabla \ln \mu_\theta(u|x) \tag{5.1}$$

この値をゼロに収束させることができるように、「アメとムチ」を導入することが一つの提案アルゴリズムである。この論文では two actor-critic アルゴリズム として、critic ベクトル $r = (r^1, r^2, \ldots, r^m)$, 特徴 (feature) として $\phi_\theta^j, j = 1, 2, \ldots, m$ をもちいて

$$Q_\theta^r(x,u) = \sum_j r^j \phi_\theta^j(x,u)$$

とした。

actor-critic アルゴリズムにおける政策学習は、価値関数の近似より収束の速さは遅い。したがって、actor-critic アルゴリズムの収束分析は、確率近似アルゴリズムの 2 倍程度の規模の収束しか頼れない：
[KonTsi03b] V. R. Konda and J. N. Tsitsiklis, "Linear Stochastic Approximation Driven by Slowly Varying Markov Chains", Systems and Control Letters, Vol. 50, No. 2, 2003, pp. 95-102. といわれている。

5.2 平均コストの TD 法

Temporal Difference 法は平均コスト問題に適用することができる。収束および近似エラーの保証は本質的に割引率のある問題と同じ程度である。したがって、割引率のない問題に対する代用として、割引のある定式化をおこなう必要はない。
[TsiRoy99] J. N. Tsitsiklis, and B. Van Roy, "Average Cost Temporal-Difference Learning", Automatica, Vol. 35, No. 11, November 1999, pp. 1799-1808.

いま、ある時刻 k において、 $r_k, \hat{Z}_k, \alpha_k$ を critic パラメータとして、θ_k を actor パラメータ とする。$(\hat{X}_k, \hat{U}_k)$ を状態と決定の組から、新しくつぎの $\hat{X}_{k+1}$ を求める：つまり更新をつぎの関係式で求める。これを TD(1) critic とよぶ。

$$\begin{aligned} \alpha_{k+1} &= \alpha_k + \gamma_k \Big(c(\hat{X}_{k+1}, \hat{U}_{k+1}) - \alpha \Big) \\ r_{k+1} &= r_k + \gamma_k d_k \hat{Z}_k \end{aligned} \tag{5.2}$$

ただし TD d_k は

$$d_k = c(\hat{X}_k, \hat{U}_k) - \alpha_k + r_k' \phi_{\theta_k}(\hat{X}_{k+1}, \hat{U}_{k+1}) - r_k' \phi_{\theta_k}(\hat{X}_k, \hat{U}_k)$$

また γ_k は適当なステップサイズとする。
TD(1) critic: ある特別な状態 x^* は、推移が正の確率で到達できるもので、これを仮定して、つぎで定める.

$$\begin{aligned} \hat{Z}_{k+1} &= \hat{Z}_k + \phi_{\theta_k}(\hat{X}_{k+1}, \hat{U}_{k+1}) \quad \text{if} \quad \hat{X}_{k+1} \neq x^* \\ &= \phi_{\theta_k}(\hat{X}_{k+1}, \hat{U}_{k+1}) \qquad \text{otherwise} \end{aligned}$$

TD(λ) critic ($0 < \lambda < 1$):

$$\hat{Z}_{k+1} = \lambda \hat{Z}_k + \phi_{\theta_k}(\hat{X}_{k+1}, \hat{U}_{k+1})$$

Actor:

$$\theta_{k+1} = \theta_k - \beta_k \Gamma(r_k) r_k' \phi_{\theta_k}(\hat{X}_{k+1}, \hat{U}_{k+1}) \psi_{\theta_k}(\hat{X}_{k+1}, \hat{U}_{k+1}) \tag{5.3}$$

上記の改訂のアルゴリズムによって定められた列が、つぎの仮定を満たすならば、確率収束させることができる。

Theorem 5.2 条件：*(a)* $\sum_k \beta_k = \sum_k \gamma_k = \infty$ *(b)* $\sum_k \beta_k^2 < \infty, \quad \sum_k \gamma_k^2 < \infty$ *(c)* $\sum_k \left(\frac{\beta_k}{\gamma_k}\right)^d < \infty$ *for* $\exists d > 0,$ および追加の $\Gamma(r)$ に関する条件があれば, *TD(1)* アルゴリズムは

$$\liminf_k |\nabla\overline{\alpha}(\theta)| = 0, \quad w.p.1$$

なる収束が成り立つ。さらに *TD(λ) critic* アルゴリズムでは、$\forall\epsilon > 0$, λ が十分に値 1 へ近ければ、

$$\liminf_k |\nabla\overline{\alpha}(\theta)| < \epsilon, \quad w.p.1$$

を得ることができる。

平均基準および割引された基準の TD についての性質はつぎで詳細に比較されている:
[TsiRoy02] J. N. Tsitsiklis and B. Van Roy, "On Average Versus Discounted Reward Temporal-Difference Learning", Machine Learning, Vol. 49, No. 2, pp. 179-191, November 2002.

5.3 価値関数の学習に基づいた方法の収束性

貪欲な（グリーディ）政策を用いるシミュレーション、および単純な「モンテカルロ」(平均)、価値関数の学習のためのルックアップテーブル表現を使用する方法の収束性:
[TsiTsi02] J. N. Tsitsiklis, J. N. Tsitsiklis, "On the Convergence of Optimistic Policy Iteration", Journal of Machine Learning Research, Vol. 3, July 2002, pp. 59-72.

最適停止問題では、収束が保証されている唯一の既知問題のクラスである。Q 学習のような方法と同様であり、任意の線形化パラメータ化された価値関数近似をもち、ある固定された政策に制限をもたない:
[TsiRoy99c] J. N. Tsitsiklis and B. Van Roy, "Optimal Stopping of Markov Processes: Hilbert Space Theory, Approximation Algorithms, and an Application to Pricing Financial Derivatives", IEEE Transactions on Automatic Control, Vol. 44, No. 10, October 1999, pp. 1840-1851.

一時的差分法 (単一の政策の場合で、線形パラメータ化された関数近似) は、収束が保証されている。極限で得られる近似誤差は、特別な近似アーキテクチャのもとでは、最良からあまり遠くに外れるようなものではない：
[TsiRoy97] J. N. Tsitsiklis and B. Van Roy, "An Analysis of Temporal-Difference Learning with Function Approximation", IEEE Transactions on Automatic Control, Vol. 42, No. 5, May 1997, pp. 674-690.

ある特別なタイプの関数近似 (例えば、状態の集まり) に対する Q 学習に関する収束の結果および近似エラーの限界:
[TsiRoy96] J. N. Tsitsiklis and B. Van Roy, "Feature-Based Methods for Large Scale Dynamic Programming", Machine Learning, Vol. 22, 1996, pp. 59-94.

Q 学習および TD(0) の一時的差分法 (ルックアップ表表現を備えたもの) は、Bellman 方程式を解決する確率的な近似方法としてみなすことができる。それらの収束は、重みつき最大値ノルムに関して反復写像が縮約的である場合の、最初に開発され、確率的近似理論が確立されまた。
[Tsi94] J. N. Tsitsiklis, "Asynchronous Stochastic Approximation and Q-learning", Machine Learning, 16, 1994, pp. 185-202.

6 Rollout アルゴリズム

よいヒューリスティックおよび、本質的に単一の政策反復 (ダイナミック・プログラミング意味で) の実行から始めて、ヒューリスティックの性能を改善する系統的な方法を提供し、実際的なセッティングの中では大きな期待感をもつ。

[BerTsiWu97] D. P. Bertsekas, J. N. Tsitsiklis, and C. Wu, "Rollout Algorithms for Combinatorial Optimization", Journal of Heuristics, Vol. 3, 1997, pp. 245-262.

7 アプリケーションと事例研究

7.1 ロボット制御、盤ゲーム:

[KimMiyKob99] 木村元、宮崎和光、小林重信：強化学習システムの設計指針、計測と制御、38 巻、10 号、(1999)

ロボットの歩行動作獲得に、強化学習を適用した動作例を述べている。モータ 2 個搭載した 2 自由度のロボット A と B に対し、メカニズム的には全く異なるが、完全に同一の強化学習を適用し、効率よく前進する動作を獲得させている。

コンピュータによる知能を加えたボードゲームは、計算機の黎明期から行われていた。[*2]

バックギャモンとはいわゆる西洋双六（スゴロク）である。強化学習が注目を浴びるようになったのは、つぎの論文の成果が大きかったかもしれない。Sutton の TD(λ) アルゴリズムを実際の問題に適用できたという報告である。結論としては、従来、よく知られてきた定番よりこのアルゴリズムで新しい布石の戦略が得られたという。

[Tes92] G.Tesauro; "Pratical Issues in Temporal Difference Learning", Machine Learning, 8 (1992) 257-277.

[Tes02] G.Tesauro; "Programming backgammon using self-teaching newral nets", Artificial Intelligence, 134 (2002), 181-199.

ただしこの論文に述べられている数式はつぎの唯一つ TD(λ) アルゴリズムにおけるネットワークの重荷を変化させるものだけ：

$$w_{t+1} - w_t = \alpha(Y_{t+1} - Y_t) \sum_{k=1}^{t} \lambda^{t-k} \nabla_w Y_k$$

[*2] 1996 年に IBM のコンピュータであるディープ・ブルーがガルリ・カスパロフと対戦し、1 つのゲームとしては、初めて世界チャンピオンに勝利を収めた。ただし、これは 6 戦中の 1 勝に過ぎず、全体ではカスパロフの 3 勝 1 敗 2 引き分けであった。しかし、翌 1997 年に、ディープ・ブルーは、2 勝 1 敗 3 引き分けとカスパロフ相手に雪辱を果たした。現実的には、これだけの試合数で実力は評価できないが、世界チャンピオンと互角に戦えるだけの能力になったことは確かである。その後も人間の名人対コンピュータの対戦は行なわれ、2002 年の 10 月に行われたウラジミール・クラムニクとコンピュータソフトディープ・フリッツとのマッチでは、両者引き分け。2003 年 1 月 26 日から 2 月 7 日までニューヨークで行なわれた、カスパロフとディープ・ジュニアとのマッチも 1 勝 1 敗 4 引き分けで両者引き分けに終わっている。2003 年 11 月 11 日から 11 月 18 日に行なわれたカスパロフと X3D Fritz (英語) のマッチは 1 勝 1 敗 2 引き分けで両者引き分けに終わった。ディープ・ブルーの後は、PC で動くコンピュータソフトが主力であるが、ハードウェアを含めて最強のチェス・コンピュータを作る試みがヒドラプロジェクトで行われている。これは、64 ノードの Xeon プロセッサからなる。2005 年 11 月には、人間とコンピュータのチームによる対戦がスペインのピルバオで行われた。人間のチームは元世界チャンピオンの、カシムジャノフ、カリフマン、ポノマリョフの 3 人、コンピュータのチームは、ヒドラ、フリッツ (Fritz)、Junior の 3 種。結果は 8-4 でコンピュータの勝利となり、人間がコンピュータに勝つことは次第に難しくなってきた。wikipedia より

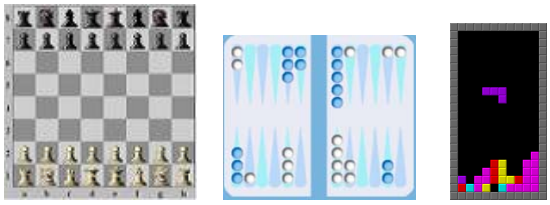


図 1　Chess, Backgammon, Tetris

$\lambda^{t-k}\nabla_w Y_k$ は重みに関するネット出力 Y_k の勾配を表す。

テトリス（tetris）[*3] のゲームは確率最短経路問題の例として p.50 に挙げられている。

7.2　小売り業:

[ManSimSunTsi04] S. Mannor, D. I. Simester, P. Sun, and J. N. Tsitsiklis, "Bias and Variance Approximation in Value Function Estimates," July 2004; to appear in Management Science.
[SimSunTsi06] D. I. Simester, P. Sun, and J. N. Tsitsiklis, "Dynamic Catalog Mailing Policies," Management Science, Vol. 52, No. 5, May 2006, pp. 683-696.

7.3　金融:(複雑なアメリカン・オプションの価格)

[RoyTsi01] B. Van Roy and J. N. Tsitsiklis, "Regression Methods for Pricing Complex American–Style Options," IEEE Trans. on Neural Networks, Vol. 12, No. 4, July 2001, pp. 694-703.
[TsiRoy99] J. N. Tsitsiklis and B. Van Roy, "Optimal Stopping of Markov Processes: Hilbert Space Theory, Approximation Algorithms, and an Application to Pricing Financial Derivatives", IEEE Transactions on Automatic Control; Vol. 44, No. 10, October 1999, pp. 1840-1851.

7.4　在庫管理, 生産管理:

[RoyBerLeeTsi96] B. Van Roy, D. P. Bertsekas, Y. Lee, and J. N. Tsitsiklis, "A Neuro-Dynamic Programming Approach to Retailer Inventory Management", November 1996. Short version in Proceedings of the 36th IEEE Conference on Decision and Control, San Diego, California, December 1997, pp. 4052-4057.
[WanMah99] Wang,G., Mahadevn,S.: "Hierarchical Optimization of Policy- Coupled Semi-Markov Decision Processes", Proceedins of the 16th International Conference on Machine Learning, pp.464-473 (1999).

複数の加工機械を直列に連結して構成された生産ラインにおいては、原料からある機械で初めの製品を作り、それを倉庫に保管し、つぎの機械ではこれを用いて別の製品を作る。最終的な製品に至るまでには在庫の管理が必要である。目的は在庫を最小化しつつ製品の需要を満たす最適な制御を学習する。製造機械の故障と修理など、機械の稼動・待機・メンテナンスのタイミングを制御する。この問題もセミマルコフ決定過程とし

[*3] 元々は旧ソ連の科学者アレクセイ・パジトノフ (en:Alexey Pajitnov) 英国名 Robert Richard Rutherfurd が教育用ソフトウェアとして開発したものである。その後ライセンス供給が様々なゲーム制作会社に対してなされ、各種のプラットフォーム上で乱立する状態になった。

て定式化される。上記の論文では各機械ごとにエージェントを割り当てるマルチエージェントシステムが用いられ、よく知られたトヨタのカンバン方式などと比較して、優れた制御規則を得ているという。
[InoOhn06] 井家敦、大野勝久；”ニューロ・ダイナミックプログラミングによる負荷分散システムの離散時間分散政策”、日本オペレーションズ・リサーチ学会和文論文誌、2006 年 49 巻、４６－６１頁.

負荷分散とは、システムの構成要素に負荷を与えることで与えられたシステムの性能を最大限発揮できるようにと意図するもの。コンピュータや通信、生産システムを主な対象とする。従来は静的な場合を待ち行列モデルでのレスポンス時間を最小化する非線形計画問題としているが、ここでは動的に変化するシステムの情報、たとえば各ノードのジョブ数を利用するなどして、モデルを平均利得のマルコフ決定過程として定式化し、従来の方式と、新しくニューロ・ダイナミックプログラミングアルゴリズムを適用して、有効性を評価している。

7.5 **コミュニケーション・ネットワーク**:

[MarMihTsi00] P. Marbach, O. Mihatsch, and J. N. Tsitsiklis, ”Call Admission Control and Routing in Integrated Service Networks Using Neuro-Dynamic Programming,” IEEE Journal on Selected Areas in Communications, Vol. 18, No. 2, February 2000, pp. 197-208.

前論文 [MarMihTsi00] の準備段階: [MarMihTsi98] P. Marbach, O. Mihatsch, and J. N. Tsitsiklis, ”Call Admission Control and Routing in Integrated Service Networks Using Reinforcement Learning,” in Proceedings of the 1998 IEEE CDC, Tampa, Florida.

[MarTsi97] P. Marbach, and J.N. Tsitsiklis, ”A Neuro-Dynamic Programming Approach to Call Admission Control in Integrated Service Networks: The Single Link Case,” Technical Report LIDS-P-2402, Laboratory for Information and Decision Systems, M.I.T., November 1997. Short version in Proceedings of the 2003 IEEE Conference on Decision and Control, Maui, Hawaii, December 2003.

[MarMihSchTsi97] P. Marbach, O. Mihatsch, M. Schulte, and J. N. Tsitsiklis, ”Reinforcement Learning for Call Admission Control and Routing in Integrated Service Networks,” presented at the Neural Information Processing Systems, Denver, Colorado, November 1997.

[SinBer97] Singh,S., Bertsekas,D.: ”Reinforcement Learning for Dynamic Channel Allocation in Celler Telephone Systems”, Advances in Neural Information Processing System 9, pp.974-980 (1997).

通信システムにおける PHS（NTT ドコモに代わり、ウィルコムがサービスしている）[*4]では、サービス地域はセルとよばれる地域に分割され、セル内の各通話者はそれぞれ異なる周波数帯を使う。近接するセルでは同一の周波数帯を使えないという制約がある。この限られたチャンネルで可能な通話数が最大となるよう周波数を割り当てる必要がセミ・マルコフ決定過程の強化学習にもとづく方法を提案し、既存のヒューリスティクスより効率のいい結果を得たという。

[*4] 設備や仕様を簡略化し、通話料を低く押さえた携帯電話の一種。一つの基地局がカバーする範囲が狭く、端末 1 台あたりの周波数帯域が携帯電話よりも広いため、データ通信の速度は 32〜64kbps と携帯電話に比べて極めて高速で、ISDN と遜色ない快適な通信環境を実現できる。音質も固定電話網並みに良い。また、基地局設備が簡易で安価な点を生かし、地下街や地下鉄駅などでの基地局設置がいち早く進み、都市部では携帯電話よりもつながりやすいという状況が生まれている。登場した当初は通話中の基地局の変更 (ハンドオーバー) ができず、高速移動中 (電車・自動車など) に通話ができないなどの欠点があったが、そうした欠点は現在ではほとんど解決され、「データ通信が高速な携帯電話」とも言える強力な通信システムに変身を遂げている。http://e-words.jp/w/PHS.html より

8 数値実験

この節では Optimistic TD(0) のアルゴリズムについて数値例を挙げる. この計算アルゴリズムは次のような擬似コードとして表すことが出来る.

Step 1. $n=0$ とせよ. J_n を初期化せよ. 初期状態 $i_0=i$ を選べ.

Step 2. 現在の状態 i_n と J_n を用いて,

(i) greedy policy(action) $\mu_n(i_n)=\underset{v\in U(i_n)}{\operatorname{argmax}}\sum_{j=1}^{N}p_{ij}(v)(r(i_n,v)+\alpha J_n(j))$ を決定せよ.

(ii) greedy action $\mu_n(i_n)$ から次の期の状態 $i_{n+1}:=j$ を simulation により観測せよ.$(i_{n+1}\leftarrow j)$

(iii) $J_n(i_n)$ を次のように改定せよ.

$$J_{n+1}(i)=\begin{cases} j_n(i), & (i\neq i_n)\\ j_n(i)+\gamma_n(r(i,\mu_n(i_n))+\alpha J_n(j)-J_n(i)), & (i=i_n)\end{cases}$$

Step 3. n を $n+1$ として Step 2 へ戻れ.

図 2　Optimistic TD(0) Algorithm with discount factor α.

次の Table1 のような MDP のモデルを考える. 状態空間は $S=\{1,2,3\}$, 各状態ごとの決定はそれぞれ $U(1)=\{1,2,3\},U(2)=\{1,2,3\},U(3)=\{1,2\}$ である. $p_{ij}(u)$ は推移確率行列を表し, $r(i,u)$ は immediate reward を表す. (Iki et.al.“A structured pattern algorithm for multichain Markov decision processes”(2007) の数値例の一部を引用)

state	action	$q_{ij}(u)$			reward
i	$u\in U(i)$	$j=1$	$j=2$	$j=3$	$r(i,u)$
1	1	1/2	1/4	1/4	5
	2	1/8	1/8	3/4	2
	3	3/16	3/4	1/16	2.5
2	1	1/4	1/2	1/4	6
	2	1/16	3/16	3/4	0.75
	3	5/8	1/4	1/8	2.25
3	1	1/2	1/2	0	14
	2	1/16	1/16	7/8	13

表 1　A numerical example

8.1 実験結果

ここでは割引率 $\alpha = 0.999$ として, Optimistic TD(0)-アルゴリズムにより update function J_n がどのように改定されていくかを Figure 8.1 に示す. stepsize parameter $\gamma_n(i)$ については条件 (i)$\gamma_n(i) \geqq 0$, (ii)$\sum_{n=0}^{\infty} \gamma_n(i) = 0$, (iii)$\sum_{n=0}^{\infty} \gamma_n^2(i) < \infty$ を満たすように選ぶため, 計算上は単純に $\gamma_n(i) = 1/n$ と取ることも出来るが, このように取ると収束が著しく遅くなるため, 始めの 1 万ステップでは 1/5 を取り, その後は 1 万ステップごとに分母が 1 ずつ増加するように工夫している. このときの実験結果では 500 万ステップ目で $J_n(1) = 11319.88, J_n(2) = 11318.53, J_n(3) = 11332.38$ という数値を得た. このモデルの optimal value は $J_n^*(1) = 11332.6, J_n^*(2) = 11321.2, J_n^*(3) = 11335.2$, optimal policy は $f^*(1) = f^*(2) = f^*(3) = 2$ である.

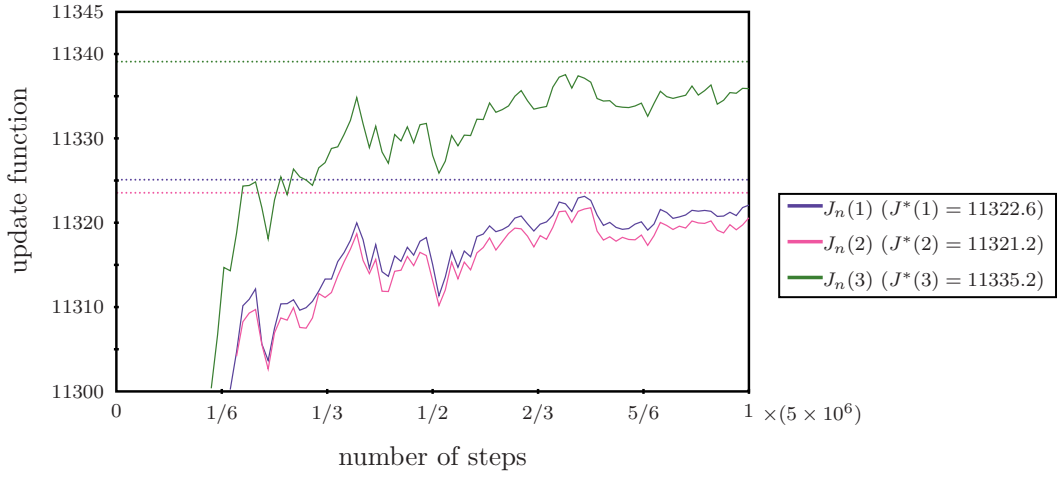


図 3　Numerical example ($\alpha = 0.999$)

参考文献

[1] Dimitri P. Bertsekas & John N. Tsitsiklis: ”Neuro-Dynamic Programming”, 1996, Athena Scientific, Optimization adn Computation Series, ISBN 1-886529-10-8, 512 pages