数理解析研究所講究録
1559巻 2007年 34-49

マルコフ決定過程におけるTD法による学習アルゴリズムについて
A learning algorithm of TD method for Markov decision processes

弓削商船高等専門学校・総合教育科　堀口 正之 (Masayuki HORIGUCHI)
General Education, Yuge National College of Maritime Technology
千葉大学・教育学部　蔵野 正美 (Masami KURANO)
Faculty of Education, Chiba University
千葉大学・理学部　安田 正實 (Masami YASUDA)
Faculty of Science, Chiba University

アブストラクト

マルコフ決定過程 (MDP) に関する最適化問題の解法として, ニューロ・ダイナミック・プログラミングを紹介し, TD(Temporal Difference) 法の収束性の結果と数値例として3状態のMDP問題を数値実験した結果およびHowardの自動車取替え問題への適用の考察を報告する.

1 はじめに

マルコフ決定過程という命名は, 1957年のR. Bellmanの論文が発端と思われるが, 同時に出たプリンストン大学出版の「動的計画法 (ダイナミック・プログラミング)」のほうが有名になってしまっている. したがって, 1960年R. A. Howardの「動的計画法とマルコフ過程」(ワイリー社) によるものがマルコフ決定過程の名前を喚起している. さらに何といっても統計学の大御所であるD.Blackwellが, 1962年のAnn. Math. Stat. に「離散的動的計画法」を発表したことが, 多くの研究者にかなりの驚きを与え, この方面の発展に極めて大きな役割をもたらした. その後, G. DeLeve, C. Derman, A. F. Veinott, D. J. White, R. E. Strauchなどの研究が知られている. 日本でも多くの研究者が取り組み, M. Ogawara, T. Sakamoto, T. Odanaka, N. Furukawa, M. Kurano, M. Mine, K. Ohno, S. Osaki, S. Iwamoto等々により, 今日の盛隆がもたらされた.

統計科学, 数理工学, 生産工学, 経営科学, 社会科学など制御確率システムに関する問題は, マルコフ決定過程としてモデル化されている. モデルの応用にはコンピュータの進歩とも結ばれている. 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. 理論的には有限集合の状態空間と決定空間から, 一般のボーランド空間に拡張しても, 現実の問題解法には直接関わることができない. 当初から, マルコフ決定問題または動的計画法についての応用に際しては, 計算困難が伴うことから理論研究が先行していた. またポントリヤーギンの最適制御問題も微分幾何学の変分法としてよく知られている.

一方, 学習アルゴリズムについては, 多くの手法が脳科学の進歩にも影響を受け, ロボットの知能研究などで近年進展されている. この不確実な現象の制御分野に関して, 動的計画法のブレイクスルーともPutermanが言う, 「Dynamic Programming and Optimal Control(動的計画法と最適制御)」(D. Bertsekas and J.N.Tsutsuklis アテネ出版) が1995年に出版された. いわゆるNeuro-Dynamic Programmingである. このニューロ・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の原理そのものである.

ニューロ・ダイナミック・プログラミングは人工知能分野の中で使用される用語で「強化学習 (Reinforcement Learning)」とよばれる. 有限状態数のマルコフ決定過程としてモデル化される環境下においてエージェントは現在の状態を観測し, それに応じた意思決定を行いその結果として状態推移が起こり環境から報酬を得る機械学習の一種である. 強化学習は, この一連のシステム変化に対する利得の最大化をもたらす行動を学習することである. 強化学習は, 教

師付き学習とは異なり未知の問題に対して, エージェントが探索 (exploration) と知識利用 (exploitation) の間のトレードオフをうまく実行していきながら最良の行動と報酬をもたらすことを目標としている. また, 強化学習の手法として動的計画法が取り入れられている.

ダイナミック・プログラミングの適用可能性については, 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) を使用することが特徴である. いわゆる複雑性に対して「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.

「価値関数 (value function) の近似」のアプローチでは, 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 [BerTsi96] 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] 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). 強化学習について検索できる多数の HP も参考になる (http://www.cs.ualberta.ca/~sutton/book/the-book.html).

日本語版の強化学習について, よくある質問と答えをまとめた HP も下記に掲げられている (http://nao.s164.xrea.com/RL-FAQ-j.html).

さらに強化学習の入門として [SutBar98] や survey として [KaeLit96, BarSutWat90] の文献を紹介する.

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 policy $\pi = \{\mu_0, \mu_1, \ldots\}$, $\mu_k(i) \in U(i)$(finite set) 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 0 が与えられているとする. この特別状態に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ことにする. 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

$$\begin{aligned} &J_N^\pi(i) \\ &:= E\left[\alpha^N G(i_N) + \sum_{k=0}^{N-1} \alpha^k g(i_k, \mu_k(i_k), i_{k+1}) \middle| i_0 = i\right], \end{aligned} \tag{2.1}$$

が定義される. 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

$$\rho_\mu = \max_{i=1,\ldots,n} P(i_n \neq 0 | i_0 = i, \mu) < 1. \tag{2.2}$$

定常政策が *proper* でない場合を *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, $TJ(i)$, $T_\mu J(i)$, $i = 1,\ldots,n$ はそれぞれ次式で定義する.

$$(TJ)(i) := \min_{u\in U(i)} \sum_{j=0}^{n} p_{ij}(u)(g(i,u,j)+\alpha J(j)), \tag{2.3}$$

$$(T_\mu J)(i) := \sum_{j=0}^{n} p_{ij}(\mu(i))(g(i,\mu(i),j)+\alpha J(j)). \tag{2.4}$$

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 $\alpha = 1$ かつ 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^*, μ^*) 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ものとする.

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

1. 初期値 (J_0, μ_0), $J_0 \in \mathbb{R}^n, \mu_0 \in \Pi_S$.
2. $k(\ge 0)$ 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
   $\Delta_k = (\Delta_k(1), \Delta_k(2), \cdots, \Delta_k(n)) \in \mathbb{R}$,
   $\Delta_k(i) =$
   $\sum_{m=0}^{\infty} E_{\mu_{k+1}}[(\alpha\lambda)^m d_k(i_m, i_{m+1}) \mid i_0 = i]$,
   $d_k(i,j)$
   $= g(i,\mu_{k+1}(i),j) + \alpha J_k(j) - J_k(i)$
   (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. さらにつぎの結果が得られている.

Theorem 2.1 (convergence, p.45). 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,

$$(J_k, \mu_k) \longrightarrow (J^*, \mu^*)\,(k \to \infty) \tag{2.5}$$

ある $\bar{k}$ が存在して, $k \ge \bar{k}$ では

$$\|J_{k+1} - J^*\|_\infty \le \frac{\alpha(1-\lambda)}{1-\alpha\lambda}\|J_k - J^*\|_\infty \tag{2.6}$$

(b) $\alpha = 1$ のとき, *Assumption 2.1* のもとで *(2.5)* 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([BerTsi96]) の結果を一部紹介する.

利得を意味する $r_t = (r_t(1), r_t(2), \cdots, r_t(n)) \in \mathbb{R}^n$ $(t \ge 0)$ 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) $\gamma_t(i) \ge 0$, $\sum_{t=0}^{\infty} \gamma_t(i) = \infty$, $\sum_{t=0}^{\infty} \gamma_t^2(i) < \infty$

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

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

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 $r \le \bar{r}$ ならば, $Hr \le H\bar{r}$.

(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: 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.

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

本文 246 ページの (5.60) 式ではつぎのような表現形式を用いる:

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

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.

次がこの 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 $Q_t(i, u) \to Q^*(i, u)$ *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, 仮定 2.1 のもとでの収束性は定理 (monotone case) を適用して証明されるが, これはそれを一般化したものである.

4 在庫問題の例題

この節では Actor-Critic Algorithm で取り上げられている例題を挙げる.

[KonTsi03] の例題 4.7(p.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].

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].

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] と [KonTsi99](前論文 [KonTsi03] の準備段階).

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 $\bar{\alpha}(\theta) = \sum_{x,u} c(x,u)\eta_\theta(x|u)$ と表される. 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:

$$\begin{aligned}&\bar{\alpha}(\theta) + V_\theta(x)\\ &= \textstyle\sum_u \mu_\theta(u|x)\left[c(x,u) + \sum_y p(y|x,u)V_\theta(y)\right]\end{aligned}$$

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

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

と定めると, つぎの結果が得られる:

Theorem 5.1.

$$\nabla\bar{\alpha}(\theta) = \sum_{x,u} \eta_\theta(x,u)Q_\theta(x,u)\psi_\theta(x,u) \qquad (5.1)$$

ただし $\quad \psi_\theta(x,u) := \nabla \ln \mu_\theta(u|x)$

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 $\phi_\theta^j, j = 1, 2, \ldots, m$ を用いて

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

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].

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].

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.

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

ただし TD d_k は

$$\begin{aligned} d_k &= c(\hat{X}_k, \hat{U}_k) - \alpha_k \\ &\quad + r'_k \phi_{\theta_k}(\hat{X}_{k+1}, \hat{U}_{k+1}) - r'_k \phi_{\theta_k}(\hat{X}_k, \hat{U}_k) \end{aligned}$$

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.

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

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:

$$\begin{aligned} \theta_{k+1} &= \theta_k - \beta_k \Gamma(r_k) r'_k \\ &\quad \times \phi_{\theta_k}(\hat{X}_{k+1}, \hat{U}_{k+1}) \psi_{\theta_k}(\hat{X}_{k+1}, \hat{U}_{k+1}) \end{aligned} \tag{5.3}$$

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)* $\sum_k \beta_k^2 < \infty$, $\sum_k \gamma_k^2 < \infty$ *(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

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

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 に近ければ,

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

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].

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:[Tsi02].

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].

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].

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].

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確率的近似理論であり, 次の文献で最初に発表された [Tsi94].

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].

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:

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:[KimMiyKob99].

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. 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.

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, Tes02]. 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である:

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

$\lambda^{t-k} \nabla_w Y_k$ 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) もよく知られたゲームである. 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. このゲームは確率最短経路問題の例として [BerTsi96]p.50 に挙げられている.

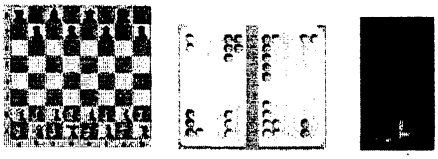


図 1: Chess, Backgammon, Tetris

最近日本の将棋についてもTD法を取り入れたアルゴリズムがあるという:[BeaSmi98].

7.2 事例研究など:

その他, 流通システムとして, 小売業者の問題として次の文献がある: [ManSimSunTsi04, SimSunTsi06].

また, 金融のアメリカン・オプションの価格評価は, 最適停止時間問題とみなすことができ, [RoyTsi01, TsiRoy99c] などに挙げられている.

在庫管理や生産管理についても幾多の論文が発表されている: [RoyBerLeeTsi96, WanMah99].

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]:この論文での負荷分散とは, 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.

さらにコミュニケーション・ネットワークに関してもつぎの論文で議論されている:[MarMihTsi00],[MarMihTsi98](前論文 [MarMihTsi00] の準備段階), [MarTsi97],[MarMihSchTsi97].

通信システムにおける PHS(主にウィルコムがサービスしている) では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 32〜128kbps と数年前の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). 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: [SinBer97].

8 数値実験

8.1 Optimistic TD(0) の実験

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擬似コードとして表すことが出来る (図 1). さらに, 表 1 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.

(文献 [IkiHorKura07] の数値例の一部を引用):

Step 1. $n = 0$ とせよ. Update function $J_n(\cdot)$ 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) $\mu_n(i_n) = \operatorname{argmax}_{v \in U(i_n)} \sum_{j=1}^{N} p_{i_n j}(v)(r(i_n, v) + \alpha J_n(j))$ 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) $J_{n+1}(\cdot)$ を次のように改定せよ.

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

Step 3. n を $n+1$ として Step 2 へ戻れ.

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

state	action	$p_{ij}(u)$			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

8.2 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図 4 に示す. step-size parameter $\gamma_n(i)$ については条件 (i)$\gamma_n(i) \geqq 0$,

(ii)$\sum_{n=0}^{\infty}\gamma_n(i)=\infty$, (iii)$\sum_{n=0}^{\infty}\gamma_n(i)^2<\infty$ 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. すなわち $\gamma_n(i)=1/([n/\mathrm{SP}]+5)$, ただし $SP=10000$, [ ] はガウス記号である. 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 (図 4). また, $\gamma_n(i)=1/([h_n(i)/\mathrm{SP}]+5)$(ただし, $h_n(i)$ は状態 i への n ステップ目までの訪問回数を表す) のように状態ごとにステップサイズを変更した場合についても調べた (図 5). さらに, greedy action を選択する際にそれまでの推移状態の情報をもとにした最尤推定値 ($\tilde{p}_{ij}(a)$) にした場合:

$$\mu_n(i_n)=\underset{v\in U(i_n)}{\operatorname{argmax}}\sum_{j=1}^{N}\tilde{p}_{i_nj}(v)(r(i_n,v)+\alpha J_n(j))$$

も調べた (図 6).

このモデルの optimal value(真の解) 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.

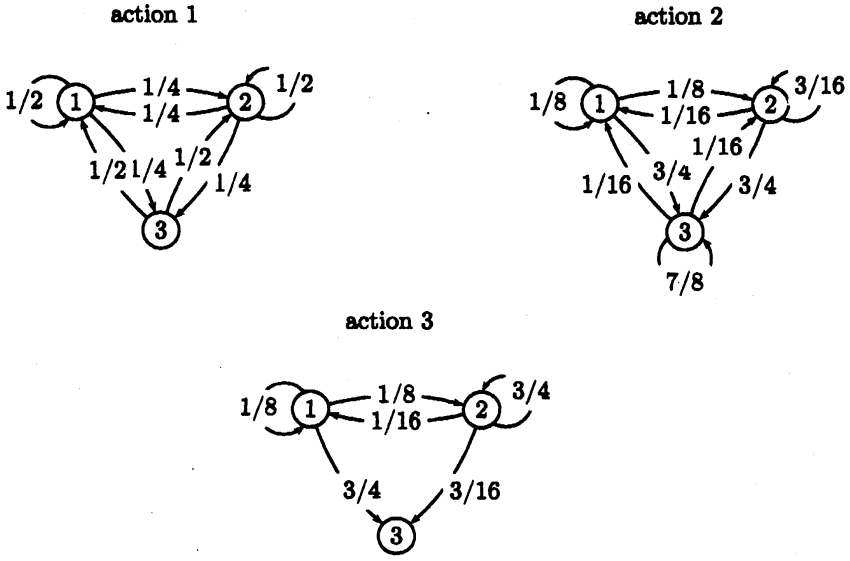


図 3: transition diagrams of numerical example.

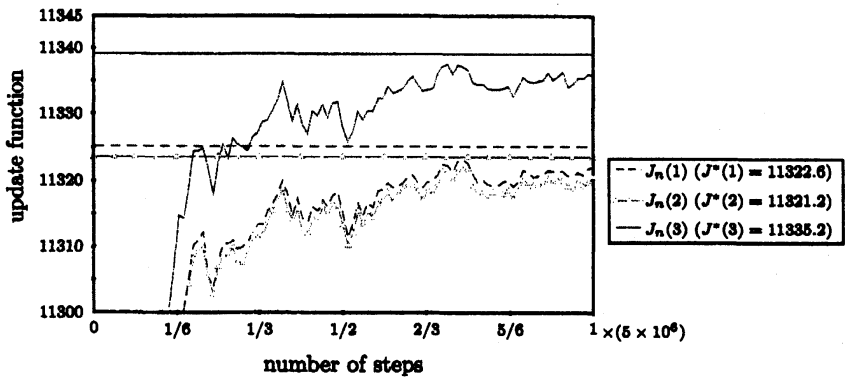
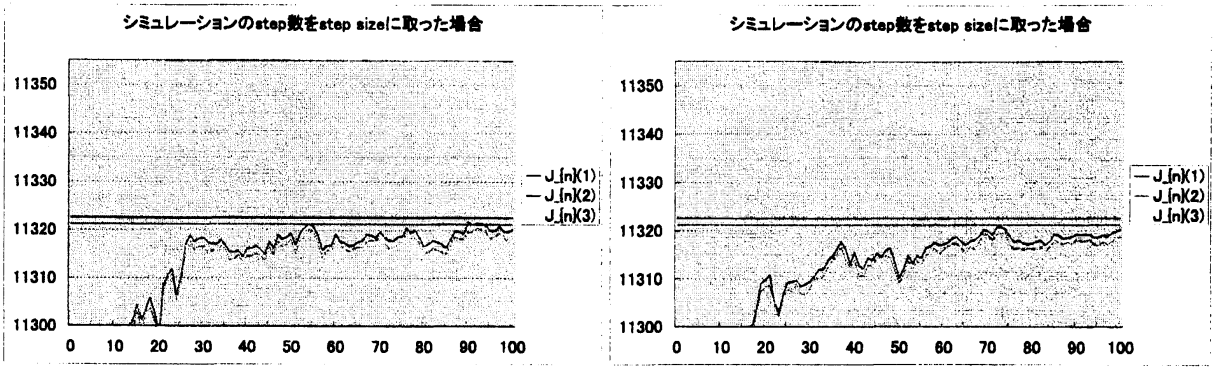
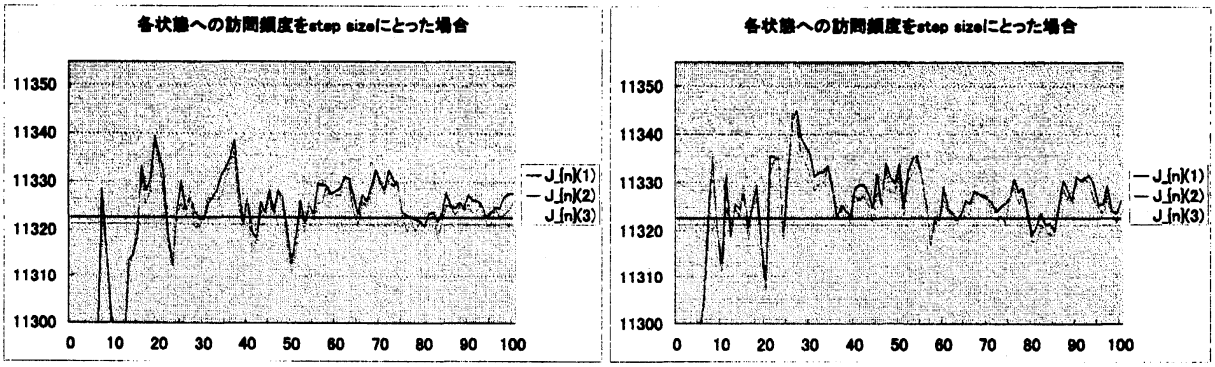


図 4: Numerical example ($\alpha=0.999$)



(a) (b)

図 5: Trajectories of $J_n(\cdot)$ with $\gamma_n(i) = \gamma_n$(using optimistic TD(0)(a) and updating with MLE(b))



(a) (b)

図 6: Trajectories of $J_n(\cdot)$ with $\gamma_n(i) = h_n(i)$(using optimistic TD(0)(a) and updating with MLE(b))

9 Howardの自動車取替え問題

Howard の自動車取替え問題 (cf.[Howd60]) では, それぞれの車の状態を 3ヶ月ごとに表し状態 0 から状態 40 までの観測状態をもつ自動車について考察する. 具体的には, 3ヶ月おきに現在所有している車の状態を調べる. 状態 '0' は新車を表し, 各 n 期の状態 i_n に対して "現在の車を次の期まで持ち続ける" 決定 $(a_n(i_n) = a = 1)$ を取ると自動車は生存確率 p_{i_n} で次の期に状態 i_{n+1} に推移して, 残りの確率 $1 - p_n$ で状態 40 に推移する. その時の所有車の維持費用は E_{i_n}(operating cost) で表される. 状態 '40' は, いわゆるポンコツ状態の車でこれ以上は悪くならないが維持費用ばかりがかかる. 他方, 決定 $a_n(i_n) = a > 1$ を選択することは, 手持ちの状態 i_n の車を売り払って (trade-in) 状態 $a_n - 2$ の車を購入することを表していて, その時の所有していた車の販売額が T_{i_n}, 購入費用が C_{a_n-2} であって, さらに決定 $a = 1$ を取ったときと同様に operating cost E_{a_n-2} がかかり次の期には生存確率 p_{a_n-2} で状態 $a_n - 1$ の車へと状態が推移するかまたは確率 $1 - p_{a_n-2}$ で状態 '40' に推移する.

MDP の構成 $(S, A, (p_{ij}(a)), (r(i,a)))$ は以下のようになっている.

$S = \{0, 1, 2, \ldots, 40\}$: state space

$A = \{1, 2, 3, \ldots, 41\}$: action space

$p_{ij}(a)$: transition prob. $r(i,a)$: immediate reward

if $a = 1$ (keep the present car),

$$p_{ij}(a) = \begin{cases} p_i, & j = i+1 \ (p_i : \text{survival prob.}) \\ 1 - p_i, & j = 40 \\ 0, & \text{other state } j \end{cases}$$

$r(i,a) = -E_i$ (operating cost)

if $a > 1$ (buy a car of age $0 \leqq a-2 \leqq 39$),

$$p_{ij}(a) = \begin{cases} p_{a-2}, & j = a-1 \\ 1-p_{a-2}, & j = 40 \\ 0, & \text{other state } j \end{cases}$$

$$r(i,a) = T_i - C_{a-2} - E_{a-2}$$

$$= (\textit{trade-in value}) - (\textit{buying cost}) - (\textit{operating cost})$$

C_i, T_i, E_i の具体的な数値は表2に示す. また, Howard 自身による最適解の結果を表 3(average case) と表 4(discount case$\alpha = 0.97$) に示す. 表の数字 '12' は状態 12 の車と買い換えること ($a = 14$) を表し, 'K' は車を次の期まで持ち続ける (Keep the present car, $a = 1$) を表している.

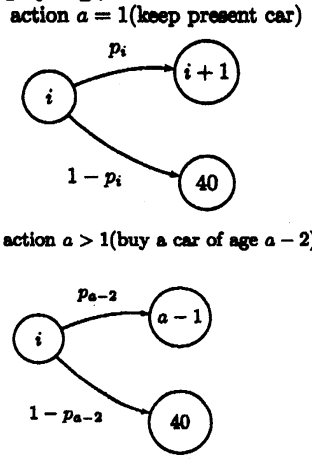


図 7: transition diagrams of Howard's automobile replacement problem.

割引き率 α のときの TD 法によるアルゴリズムは以下で与える:

Step 1. $n = 0$ とせよ. Update function $J_n(\cdot)$ 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) $\mu_n(i_n) = \underset{a \in A(i_n)}{\operatorname{argmax}} \sum_{j=1}^{N} p_{i_n j}(a)(r(i_n,a) + \alpha J_n(j))$ 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) $J_{n+1}(\cdot)$ を次のように改定せよ.

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

Step 3. n を $n+1$ として Step 2 へ戻れ.

図 8: Optimistic TD(0) Algorithm with discount factor α.

9.1 数値例 1

ここで, 我々はまず次のような update function $J_n(\cdot)$ の改定による TD 法の適用を行った.

Update function の決め方

$A_n(i_n) := a_n = 1$ のとき,

$$J_{n+1}(i_n) = J_n(i_n) + \frac{-E_i(i_n) + \alpha J_n(i_{n+1}) - J_n(i_n)}{[h_n(i_n)/\ \mathrm{SP}\ ] + 10}$$

$a_n > 1$ のとき,

$$J_{n+1}(i_n) = J_n(i_n) + \frac{\overline{R_n} + \alpha J_n(i_{n+1}) - J_n(i_n)}{[h_n(i_n)/\ \mathrm{SP}\ ] + 10}$$

ただし $\overline{R_n} = T_i(i_n) - C_i(a_n - 1) - E_i(a_n - 1)$ とした.

ここで, $J_n(i)$ は第 n 期の状態 i の update 関数の値を表し, [ ] はガウス記号, $h_n(\cdot)$ は第 n 期までの各状態の訪問頻度分布, SP は step-size parametar(を適当に調整するための変数) である.

このアルゴリズムによるシミュレーション結果を表 5 と表 6 にまとめる (SP=10000). いずれも初期状態 0(新車) から始めた 1 つの系列のシミュレーション結果である. シミュレーション回数 2 千万回までの状態 (行に相当) と決定 (列に相当) の特定の部分の頻度分布 (表 5) と同様の 5 千万回までの頻度分布 (表 6) である. ここでわかる事は, シミュレーション回数 2 千万回以降 5 千万回までは, 状態 '24' で決定 '19' を取る, すなわち "車の年齢がちょうど 6 年目 (状態 24) を迎えたら, 5 年 3 カ月目 (状態 17) の車を購入する" ことをずっと繰り返している. これは, Howard の discount case の結果の 6 年 9ヶ月目 (状態 27) で 3 年目 (状態 12) の車に買い換える最適解には一致しないが, 最適解の部分解になっている. しかし, 今回の実験では車が状態 1 から状態 3 までの新しいときと状態 27 から状態 40 までは状態 12 の車に買い換えて, 状態 4 から状態 26 までは現在の車を持ち続けるという決定の最適解は探しきれていない.

9.2 数値例 2

次に, 先に挙げた TD 法のアルゴリズムのうち Step 2 の (i) を次のような (i)' に置き換える.

Step 2.

(i)′ シミュレーションステップ数 n が 10^5 以下であるとき，または 10^3 で割り切れるとき $\mu_n(i_n)$ は $a \in A(i_n)$ のうちでそれまでの頻度の最小のものを選べ．それ以外は，greedy action $\mu_n(i_n) = \underset{a \in A(i_n)}{\operatorname{argmax}} \sum_{j=1}^{N} p_{i_n j}(a)(r(i_n, a) + \alpha J_n(j))$ とせよ．

すなわち，シミュレーション反復回数について始めの 10 万回と 1000 回目おきに現在の状態での決定の取り方の頻度分布を調べて，決定の頻度のうちで最小のものの決定 (複数ある場合は決定 "a" を数字とみなして一番小さいもの) から優先的にとる探索を入れてシミュレーションをしてみた結果が表 7 と表 8 である．これらは，探索を起こす割合 (学習確率) をそれぞれ 10% と 1% としその探索は車を買い替える決定 ($a > 2$) とすることにした．SP は 1000 としている．

表 7(学習確率 10%)，表 8(学習確率 1%) を見てみると，こちらの場合のほうが先の数値例 1 の結果に比べてループの間隔や時期が早まっているようにも見て取れる．　具体的には，表 7 では 状態 '22' で決定 '20' を取る ことと表 8 では 状態 '23' で決定 '18' を取って ループが起きている．ところで，一定の割合で探索を行った結果は表 7 で見れば各状態に対して決定 '14' を選択する頻度が他に比べてやや多いと見て取れるし，さらに状態 '1,2,3' のそれぞれにおいても決定 '14' を選択するような傾向が出ている．表 8 から状態 '27' の周辺で決定 '14' を選択する頻度が比較的多いようである．

数値実験のまとめ

今回は Howard の自動車取替え問題を例に TD 法の適用を考察してみた．初期状態が '0' の場合についてのみ考察している．このシミュレーション結果からは一つのパス (path) の流れで実験したためかループが発生した．その結果は，Howard による最適解の部分解となっていることがわかる．探索確率 (ここでは学習確率と呼んだ) をいろいろ変化させることで他の最適解を見つけられる可能性があることがわかった．誤差の伝播を考慮するとシミュレーション回数をこれ以上増やすことが難しそうであるので Q-learning など別の方法による探索を検討したい．

参考文献

[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.

[BeaSmi98] Beal,D.F., and Smith, M.C., First Results from Using Temporal Difference Learning in Shogi, Proceedings of Computers Games(CG) 1998, pp.113-125, 1998.

[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.

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

[Howd60] Howard, R.A., Dynamic Programming and Markov Processes, The Technology Press of M.I.T. 1960.

[IkiHorKura07] Iki,T, Horiguchi, M., Kurano, M., "A structured pattern algorithm for multichain Markov decision processes" To appear in Math. Meth. Oper., 2007.

[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.

[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.

[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).

[KonTsi99] 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.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[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.)

[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.

[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.

[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).

[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.

[SutBar98] Reinforcement Learning: An Introduction, by Richard S. Sutton and Andrew G. Barto. MIT Press 1998. Online version. (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.)

[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.

[Tsi94] J. N. Tsitsiklis, "Asynchronous Stochastic Approximation and Q-learning", Machine Learning, 16, 1994, pp. 185-202.

[Tsi02] J. N. Tsitsiklis, "On the Convergence of Optimistic Policy Iteration", Journal of Machine Learning Research, Vol. 3, July 2002, pp. 59-72.

[TsiRoy96] J. N. Tsitsiklis and B. Van Roy, "Feature-Based Methods for Large Scale Dynamic Programming", Machine Learning, Vol. 22, 1996, pp. 59-94.

[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.

[TsiRoy99] J. N. Tsitsiklis, and B. Van Roy, "Average Cost Temporal-Difference Learning", Automatica, Vol. 35, No. 11, November 1999, pp. 1799-1808.

[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.

[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.

[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).

表 2: Howard's Automobile Replacement

i	C_i	T_i	E_i	p_i
0	2000	1600	50	1
1	1840	1460	53	0.999
2	1680	1340	56	0.998
3	1560	1230	59	0.997
4	1300	1050	62	0.996
5	1220	980	65	0.994
6	1150	910	68	0.991
7	1080	840	71	0.988
8	900	710	75	0.985
9	840	650	78	0.983
10	780	600	81	0.98
11	730	550	84	0.975
12	600	480	87	0.97
13	560	430	90	0.965
14	520	390	93	0.96
15	480	360	96	0.955
16	440	330	100	0.95
17	420	310	103	0.945
18	400	290	106	0.94
19	380	270	109	0.935
20	360	255	112	0.93
21	345	240	115	0.925
22	330	225	118	0.919
23	315	210	121	0.91
24	300	200	125	0.9
25	290	190	129	0.89
26	280	180	133	0.88
27	265	170	137	0.865
28	250	160	141	0.85
29	240	150	145	0.82
30	230	145	150	0.79
31	220	140	155	0.76
32	210	135	160	0.73
33	200	130	167	0.66
34	190	120	175	0.59
35	180	115	182	0.51
36	170	110	190	0.43
37	160	105	205	0.3
38	150	95	220	0.2
39	140	87	235	0.1
40	130	80	250	0

表 3: Optimal policy and values for avarage case

	Iteration 7		
	gain: −$150.95		
i	a	Value	adjust value
1	12	1380	1460
2	12	1260	1340
3	K	1161	1241
4	K	1072	1152
5	K	987	1067
6	K	906	986
7	K	831	911
8	K	760	840
9	K	695	775
10	K	632	712
11	K	574	654
12	K	520	600
13	K	470	550
14	K	424	504
15	K	381	461
16	K	342	422
17	K	306	386
18	K	273	353
19	K	243	323
20	K	215	295
21	K	189	269
22	K	166	246
23	K	144	224
24	K	126	206
25	K	111	191
26	12	100	180
27	12	90	170
28	12	80	160
29	12	70	150
30	12	65	145
31	12	60	140
32	12	55	135
33	12	50	130
34	12	40	120
35	12	35	115
36	12	30	110
37	12	25	105
38	12	15	95
39	12	7	87
40	12	0	80

表 4: Optimal policy and present value for discount case $\alpha = 0.97$

i	a	Present value
1	12	−3925
2	12	−4045
3	12	−4155
4	K	−4332
5	K	−4398
6	K	−4462
7	K	−4523
8	K	−4581
9	K	−4635
10	K	−4688
11	K	−4738
12	K	−4785
13	K	−4829
14	K	−4870
15	K	−4909
16	K	−4946
17	K	−4979
18	K	−5011
19	K	−5041
20	K	−5069
21	K	−5096
22	K	−5121
23	K	−5145
24	K	−5167
25	K	−5186
26	K	−5202
27	12	−5215
28	12	−5225
29	12	−5235
30	12	−5240
31	12	−5245
32	12	−5250
33	12	−5255
34	12	−5265
35	12	−5270
36	12	−5275
37	12	−5280
38	12	−5290
39	12	−5298
40	12	−5305

表 5: Experiment by TD-method for discount case $\alpha = 0.97$(反復回数 2 千万回)

		action									
states freq.	state	1	···	13	14	15	16	17	18	19	20
1	17	34178		0	0	0	0	0	0	0	1
9	18	3018834		0	1	3	0	0	5	0	0
280	19	3092137		0	39	82	8	18	129	0	0
1428	20	2936937		0	15	14	50	65	1085	199	0
9807	21	2731559		0	27	25	89	167	5168	4174	157
46349	22	2480796		0	4	35	76	177	6894	32176	6760
200839	23	2079022		0	1	3	21	68	3267	139872	53873
1853151	24	38921		0	14	45	59	78	1959	1725456	107902
17687	25	17336		0	0	14	20	46	841	2158	4329

表 6: Experiment by TD-method for discount case $\alpha = 0.97$(反復回数 5 千万回)

		action									
state freq.	state	1	···	13	14	15	16	17	18	19	20
1	17	34178		0	0	0	0	0	0	0	1
9	18	7899531		0	1	3	0	0	5	0	0
280	19	7680570		0	39	82	8	18	129	0	0
1428	20	7227418		0	15	14	50	65	1085	199	0
9807	21	6722228		0	27	25	89	167	5168	4174	157
46349	22	6172249		0	4	35	76	177	6894	32176	6760
200839	23	5471836		0	1	3	21	68	3267	139872	53873
4942083	24	38921		0	14	45	59	78	1959	4814388	107902
17687	25	17336		0	0	14	20	46	841	2158	4329

表 7: Experiment by TD-method for discount case $\alpha = 0.97$(学習確率 10%)

		action												
freq. of states	state	1	···	11	12	13	14	15	16	17	18	19	20	21
711	18	6519		18	18	18	18	20	18	18	18	18	18	18
2168	19	12089299		54	54	54	54	54	54	57	58	54	54	54
2235	20	11304563		53	53	53	53	52	52	56	132	110	52	52
2844	21	10513506		49	49	49	49	49	49	49	146	70	810	49
6598848	22	3127197		47	47	47	47	47	47	47	113	180	6595642	1212
2862471	23	14028		23	23	23	23	23	23	52	104	74	2861414	23

表 8: Experiment by TD-method for discount case $\alpha = 0.97$(学習確率 1%)

		action										
frequancy of states	state	1	···	11	12	13	14	15	16	17	18	19
9	16	2902		1	1	1	1	1	1	1	1	1
30	17	8032963		4	4	4	3	3	3	3	3	3
36	18	7611390		4	4	4	4	4	4	4	4	4
36	19	7157777		4	4	4	4	4	4	4	4	4
100	20	6693079		5	5	5	5	5	5	5	60	5
1147	21	6223634		4	4	4	4	4	5	3	965	154
14351	22	5743331		4	4	4	44	28	40	82	12789	1356
5258471	23	19504		4	4	4	16	15	44	125	5251834	6425
11293	24	5675		2	2	2	8	23	2	45	6876	4333