システム/制御/情報，Vol. 60, No. 6，pp. 238–243，2016

解　　説

強化学習における効率的な計算法

泉田　　啓*

1.　はじめに

強化学習手法は有望な機械学習であるが，計算量の多さが制御などへの適用を阻んでいる．そこで本稿では，強化学習手法の基本動作と計算量の少ない効率的な計算法について説明する．強化学習についてはすでに成書や解説も多く，全般的な学習には [1,2] などの教科書，最近の研究動向については [3,4] などが参考になる．

1.1　強化学習とは

強化学習では，数値化された評価を最適化するように方策（状況に行動をどのように対応づけるか）を学習する．教師なし学習で，どの行動をとれば最適な評価が得られるかを試行により見つけ出す必要がある [1].

学習システムが方策（制御）を習得し，制御対象（プラント）を制御する．制御対象として，時不変で離散時間の有限 MDP (Markov Decision Process) でモデル化される系を考えることが多い．離散時間の有限 MDP では，状態と行動（操作量）は有限集合の元であり，現在の状態と行動の組（状態行動対）の関数として，つぎの時刻に遷移する状態の状態遷移確率が定まる．なお，現在の状態行動対と遷移先の状態の組を状態遷移という．

方策（政策）は，現在の状態の関数として，どの行動を選択するかの行動選択確率を与える．ただし，後で述べるように，適当な問題設定のもと，時不変で離散時間の有限 MDP に対する最適方策は，定常な（時不変な）確定的方策になる．制御目標は状態を構成する量と考えることができるので，学習の結果得られる最適方策は最適状態フィードバックである．また，状態が完全に観測できない場合 POMDP (Partially Observable MDP) とよばれ，その強化学習は現在も研究されている [4].

1.2　強化学習システムの基本動作

上述のように，強化学習システムは試行錯誤的サンプリングに基づき最適方策を求める．これは制御対象のシステム同定と最適制御の習得に対応する．ここで，サンプリングとは学習システムによる状態遷移の観測である．

サンプリングに基づき陽にモデル（一般的には状態遷移確率のテーブル）を推定することをモデル学習とよぶ．推定されたモデルを用いて最適方策を求めることをプランニングという．陽にモデルをもち，モデル学習とプランニングを行う場合をモデルベース強化学習という．陽にモデルをもたず，サンプリングに基づき直接に最適方策を求めることを直接強化学習またはモデルレス強化学習という．一般にモデルレス強化学習は，同じ学習結果を得るモデルベース強化学習か，その近似に書き直すことができる．そのため，モデルレス強化学習はモデル学習とプランニングの機能を合わせたものと捉えられる．

正確なモデルが得られていない学習の初期においてはモデル学習に重点を置く必要がある．全状態行動対のサンプルを得る探査のために，一般的に確率的方策を採る．このようにサンプルを得ることも含め，学習中の行動を決定するための方策を挙動方策とよぶ．他方，学習の過程では，最適方策の候補を推定し，それを評価して改善することを反復して最適方策を得る．このように評価・改善される方策を推定方策とよぶ．なお，推定方策を用いる場合の状態の評価値を状態価値関数（J 値），ある行動をした後は推定方策に従う場合の状態行動対の評価値を行動価値関数（Q 値）という．逆に，これらの値は方策を与える．そのため，最適方策を与える J 値や Q 値を獲得することが学習の目的になる．

プランニングは，基本的に DP (Dynamic Programming) の解法に基づいて行われる．モデル学習の結果，全状態行動対に対する状態遷移確率（モデル）が推定されると，方策評価と方策改善を繰り返す DP 反復により効率的に最適方策を得ることができる．どの程度正確に方策評価を行うかによって，方策反復や価値反復などの方法がある．TD (Temporal Difference) 学習 [5] の一種である TD(λ)[6] は，推定モデルを用いたプランニングの手法を与え [2]，$\lambda=1$ では方策反復，$\lambda=0$ では価値反復になり，$0<\lambda<1$ ではそれらの中間的アルゴリズムになる．モデルレス強化学習として知られる Q 学習 [7] は価値反復の近似アルゴリズムと解釈できる．その他の多くの手法も方策評価と方策改善を基本動作とする [2,8].

プランニングにおいて，DP 反復により推定方策が最適方策に収束するためには，プロパー方策から開始してプロパー方策の列を生じる必要がある [2]．ここで，プロパー方策とは有限回の状態遷移でゴール（制御目標を達成する状態）に到達する確率が正の方策である．もし

* 京都大学　工学研究科
Key Words: reinforcement learning, policy evaluation, iterative method.

推定モデルがプロパー方策をもつとすると，ランダム方策やそれを含む確率的方策（ε-グリーディ方策やソフトマックス行動選択）はプロパー方策になる．先に，モデル学習のために，学習の初期の挙動方策には確率的方策が適していると述べた．他方で，DP 反復により最適方策を求めるためにも確率的方策が適している．

1.3 強化学習手法の効率化に関連する話題

状態数を 1000，行動数を 10 とし，各状態行動対に対して 100 サンプルを取るとする．これらの数は，制御問題において必ずしも多い数ではないが，それでもサンプル数は 100 万になる．1 サンプルを得る時間を 1 秒とすると，サンプルを得るために 11.5 日以上必要である．これに比べ，プランニングに要する時間ははるかに短い．そのため，より少ないサンプルで効率的にプラント推定することは重要である [9] が，現在も研究課題である．

学習中の強化学習システムでは，モデル学習やプランニングが未収束な方策が使われる．推定モデルがあればプランニングを収束させられるので，モデルレス強化学習よりモデルベース強化学習の方が少ないサンプルで同等の方策を習得できる．ただし，モデルベース強化学習では，より多くのメモリが必要になる．

サンプル数の低減に関連して，状態遷移確率モデルを関数近似し，少数のパラメータで表現する方法がある．この少数のパラメータを決定するために必要なサンプル数は，元の離散的な状態遷移確率をすべて精度よく推定するサンプル数より，一般に少なくて済む．ただし，その精度は近似表現の良し悪しに依存する．この関数近似は，モデル表現のみならず，Q 値，J 値，方策の近似表現に用いられ，それらを記憶するメモリを低減する効果もある [2,10–15]．ただし，そのような近似構造の導入により，DP 解法が適用できなくなったり，収束の理論的保証が失われたり，収束しなくなったりする．

必要なサンプルを得る時間が現実的でない場合，実プラントに代わりシミュレーションを用いることが多い．推定モデルは実プラントに対して誤差（変動）をもつので，方策のロバスト性が重要になる．変動にロバストな方策を学習する方法も研究されている [16,17]．

1.4 強化学習における効率的な計算法

以下では，モデルベース強化学習により，DP 反復で効率的に Q 値を改善して最適方策を得ることを考える．それでも，一般的な解法では，膨大な計算量が問題になることが多い．そこで本稿では，おもに方策評価を効率化する方法を考える．これまで定常反復法 (Stationary Iterative Method: SIM) に基づく効率的アルゴリズムが検討されている [2,8] が，非定常反復法に分類される共役勾配法の発展であるクリロフ部分空間法 (Krylov Subspace Method: KSM) 型アルゴリズムの応用 [18] についても説明する．なお，**5.2** で定義する反復行列が一定の反復解法を定常反復法という．

方策評価のアルゴリズムには直接法型と反復法型がある．前者の演算は有限回であるが，全演算終了まで方策を評価できる Q 値が得られず，一般に直接法型は用いられない．後者の誤差は反復回数に対して指数関数的に減少し，反復を途中で打ち切ることができるので，方策評価には反復法型が用いられる．SIM 型と KSM 型はともに反復法に区分されるが，KSM 型は有限回で Q 値を得る直接法の性質も有している．

KSM 型と SIM 型の効率を一般的な代数問題で比較すると大差がない [18]．そのため，KSM 型を強化学習問題に適用しても方策評価の効率が改善されるとは考えにくく，注目されてこなかった．しかし，本稿では学習問題においては KSM 型が SIM 型と比較して数十〜数百倍の効率で方策評価ができることを示す．学習全体も同様に効率化されるが，これらは KSM に一般に期待される効率よりはるかに高い効率である．これらの利点は，これまで系統的に調べられていない．また，なぜこのように効率的で優れた方策評価法になるのかも説明する．

本稿の構成として，**2.** で強化学習問題を定式化し，**3.** で解法の基本的な仮定と構成を述べる．**4.** で方策評価のアルゴリズムを示し，**5.** では解法の効率を検討する．**6.** を本稿の結びとする．

2. 強化学習問題

一般的な DP の定式に従い，離散時間の動的システムを考える [2]．状態 s_i と行動 u_k は，有限集合 $\mathcal{S}$ と $\mathcal{U}$ の元で，離散的な変数とする．状態 $\mathcal{S}$ の元として $s_1,s_2,\ldots,s_N$ の N 個を考え，問題に応じて終端の状態 s_0 を付け加える．行動 $\mathcal{U}$ の元として $u_1,u_2,\ldots,u_K$ の K 個を考える．状態 s_i で行動 u_k を選ぶと状態 s_j に移る場合を遷移 (s_i,u_k,s_j) と書く．このときの状態遷移確率は $p_{ij}(u_k)$ で，コスト $g(s_i,u_k,s_j)$ を生じる．

離散時間の有限 MDP を扱う．すなわち $p_{ij}(u_k)$ は現在の状態 s_i と行動 u_k のみによって決定される．また，系は時刻に陽には依存しない．状態を行動に写像する方策として時刻に陽に依存しない定常方策 μ を考え，行動を $u_k=\mu(s_i)\in\mathcal{U}$ のように選択し，行動選択確率 $\pi(s_i,u_k)$ は時刻によらない．なお，本稿を通じて使う記号などは**付録**に示し，定式などの詳細は文献 [8] に従う．

DP 問題は，有限ステージにわたってコストが蓄積する finite horizon 問題と，期間不定でコストが蓄積する infinite horizon 問題に区別できる．本稿では後者を扱うが，時刻を特別な状態量とみなせば，前者を後者に変換できるので，一般性を欠かない．また，infinite horizon 問題で求める方策は一般的に定常方策になる [2]．

初期状態 $s^{(0)}=s_i$，初期行動 $\mu(s^{(0)})=u_k$ から始まり，第 m 回目の遷移で状態 $s^{(m)}$ を訪れ，蓄積するコストを生じるとき，定常方策 μ の全コストの期待値（Q 値）は

$$Q^{\mu}(s_i,u_k)=E\left[\sum_{m=0}^{\infty}\alpha^m g\left(s^{(m)},\mu\left(s^{(m)}\right),s^{(m+1)}\right)\right]$$

ただし，α は $0<\alpha\leq 1$ で割引率とよぶ．最適 Q 値は $Q^*(s_i,u_k)=\min_\mu Q^\mu(s_i,u_k)$ で，すべての状態と行動に対し $Q^\mu(s_i,u_k)=Q^*(s_i,u_k)$ なら方策 μ は最適である．

本稿では，infinite horizon 問題の一種である，以下の確率的最短経路問題を考える．割引なし $(\alpha=1)$ とし，コストを生じない終端（ゴール）状態 s_0 が存在し，すべての行動 u_k に対して $p_{00}(u_k)=1$, $g(s_0,u_k,s_0)=0$, $Q(s_0,u_k)=0$ とする．これらの条件下において，コスト総和の期待値最小で終端状態に達すること，すなわち，$Q^\mu(s_i,u_k)$ を最小化する最適方策を見つけることが，この問題の目的である．この問題の解法は，非常に多くの強化学習問題に適用できる一般的解法であることが知られている [2].

3. 強化学習の解法

3.1 解法の基本的仮定

多くの強化学習の解法と同様，プロパー方策を用い，Q 値を改善することで最適方策を得る．プロパー方策は，任意の初期状態 s_i に対して，最大 M ステージ $(0<M<\infty)$ で終端状態に至る確率が正の定常方策である．学習過程で挙動方策に用いられる ϵ-greedy 方策やソフトマックス行動選択則などもプロパー方策である．推定プラントを用いて Q 値を算出するが，推定誤差が検討に影響しないよう，正確にプラントが推定されているものとする．

3.2 解法の基本的構成

この仮定のもとに最適 Q 値 $\boldsymbol{Q}^*$ と最適方策を求める多くの解法がある．本稿では方策反復を考え，プロパー方策 μ_0 から開始し，新しいプロパー方策 $\mu_1,\mu_2,\ldots$ の列を生じる．まず，方策評価として，行動選択確率 $\pi_l(s_i,u_k)$ の方策 μ_l に対し，$p_{ij}(u_k)$ を既知とする線形代数方程式

$$Q^{\mu_l}(s_i,u_k)=\sum_{j=0}^{N}p_{ij}(u_k)\left\{g(s_i,u_k,s_j)+\sum_{\ell=1}^{K}\pi_l(s_j,u_\ell)Q^{\mu_l}(s_j,u_\ell)\right\}\quad(1)$$

を満たす解 $\boldsymbol{Q}^{\mu_l}$ を求める．方策 μ_l が最適方策 μ^* であるとき，(1) 式は特に Bellman 方程式とよばれる．つぎに方策改善として，次式で greedy 方策として新しい方策 μ_{l+1} を求める．得られる方策は確定的である．

$$\mu_{l+1}(s_i)=\arg\min_{u_k}Q^{\mu_l}(s_i,u_k),\ \forall s_i\quad(2)$$

$\boldsymbol{Q}^{\mu_{l+1}}=\boldsymbol{Q}^{\mu_l}$ を満たす方策 μ_l が得られるまで，μ_l に μ_{l+1} を代入して方策評価と方策改善を繰り返す．方策反復は最適方策 μ^* を得て有限回の方策改善で終了する．

方策評価には，次節以降の反復アルゴリズムを用いる．反復 1 回のみで打ち切って方策改善を行う方法は価値反復である．同様に，多くの解法は方策評価と方策改善を基本動作とするため，次節以降では方策評価の効率化に重点をおいて検討する．

Q 値ではなく J 値に基づく計算方法もあるが，方策改善の際に J 値から Q 値を計算する必要がある．また，計算コストなどの評価が容易なため，本稿では Q 値を直接計算する．なお，行動数を K として，以下に示す全アルゴリズムにおいて，J 値に基づく方法の計算量は，直接 Q 値を計算する方法に比べ $1/K$ 倍のオーダになる．

4. 方策評価の各種アルゴリズム

4.1 SIM 型の方策評価アルゴリズム

既存の方策評価アルゴリズムは SIM と考えられる．文献 [2,8] に従い，その概要を示す．行動選択確率 $\pi(s_i,u_k)$ のプロパー方策 μ に対して (1) 式を満たす $\boldsymbol{Q}^\mu$ を求めるものとする．アルゴリズムによって収束の速さが異なるが，いずれも反復回数 $m\to\infty$ での収束が保証されている．現実には必要な精度の解が得られたら反復を打ち切る．これらがどのように収束するかは，**5.2** で説明する．

4.1.1 PEI（方策評価反復）型アルゴリズム

つぎの写像 H_μ を反復する．

$$Q^{(m+1)}(s_i,u_k)=H_\mu(\boldsymbol{Q}^{(m)})\equiv\sum_{j=0}^{N}p_{ij}(u_k)\left\{g(s_i,u_k,s_j)+\sum_{\ell=1}^{K}\pi(s_j,u_\ell)Q^{(m)}(s_j,u_\ell)\right\}\quad(3)$$

4.1.2 Jacobi 型アルゴリズム

つぎの写像 F_μ を反復する．

$$Q^{(m+1)}(s_i,u_k)=F_\mu(\boldsymbol{Q}^{(m)})\equiv\frac{1}{1-p_{ii}(u_k)\pi(s_i,u_k)}\sum_{j=0}^{N}p_{ij}(u_k)\left\{g(s_i,u_k,s_j)+\sum_{\ell=1}^{K}\pi(s_j,u_\ell)Q^{(m)}(s_j,u_\ell)(1-\delta_{ij}\delta_{k\ell})\right\}\quad(4)$$

ここで，δ_{ij} は一致関数で，$i=j$ ならば 1，$i\neq j$ ならば 0 である．(3) 式，(4) 式の右辺で使われる Q 値の更新回数がすべて m であるため，これらは同期型更新則である．

4.1.3 Gauss-Seidel(GS) 型アルゴリズム

つぎの非同期型更新則を用いる．

$$Q^{(m+1)}(s_i,u_k)=F_\mu(\hat{\boldsymbol{Q}})\quad(5)$$

ここで $\hat{\boldsymbol{Q}}$ は，それまでに計算されている最も新しい Q 値からなるベクトルである．

4.1.4 SOR 型アルゴリズム

つぎの非同期型更新則を用いる．

$$Q^{(m+1)}(s_i,u_k)=(1-\omega)Q^{(m)}(s_i,u_k)+\omega F_\mu(\hat{\boldsymbol{Q}})\quad(6)$$

ただし，ω は加速係数であり，収束の必要条件は $0<\omega<2$ である．適切な ω に対し，SOR 型は収束が保証され，かつ Gauss-Seidel 型より加速される．

4.2 KSM 型の方策評価アルゴリズム

既存の方策評価法とは異なる KSM[19–21] に基づいた方策評価アルゴリズムを考える．ここでは，基礎となる共役勾配法 (Conjugate Gradient: CG) 型 [22] のみアルゴリズムを示す．その拡張である双共役勾配法 (Bi-Conjugate Gradient: BCG) 型 [23]，積型 KSM (Block Product-type Krylov Subspace Method: BPKSM) 型 [24] の CGS 型と BiCGSTAB 型については文献 [18] にまとめられている．これらがどのように収束するかは，**5.3** で説明する．なお，$\boldsymbol{A}\in R^{NK\times NK}$ は付録の記号を用いて $\boldsymbol{A}=\boldsymbol{I}-\boldsymbol{P\Pi}$ と定義される係数行列である．また，$(\boldsymbol{a},\boldsymbol{b})$ はベクトル $\boldsymbol{a}$ と $\boldsymbol{b}$ の内積を表す．

CG 型アルゴリズム

1) $m=0$ とし，初期ベクトル $\boldsymbol{Q}^{(0)}$ と以下を用意する．

$$\boldsymbol{r}^{(0)}=\bar{\boldsymbol{g}}-\boldsymbol{A}\boldsymbol{Q}^{(0)},\ \boldsymbol{c}^{(0)}=\boldsymbol{r}^{(0)}$$

2) 以下のように上から順に各値を計算する．

$$\alpha^{(m)}=\frac{\left(\boldsymbol{r}^{(m)},\boldsymbol{r}^{(m)}\right)}{\left(\boldsymbol{c}^{(m)},\boldsymbol{A}\boldsymbol{c}^{(m)}\right)}$$
$$\boldsymbol{Q}^{(m+1)}=\boldsymbol{Q}^{(m)}+\alpha^{(m)}\boldsymbol{c}^{(m)}$$
$$\boldsymbol{r}^{(m+1)}=\boldsymbol{r}^{(m)}-\alpha^{(m)}\boldsymbol{A}\boldsymbol{c}^{(m)}$$
$$\beta^{(m)}=\frac{\left(\boldsymbol{r}^{(m+1)},\boldsymbol{r}^{(m+1)}\right)}{\left(\boldsymbol{r}^{(m)},\boldsymbol{r}^{(m)}\right)}$$
$$\boldsymbol{c}^{(m+1)}=\boldsymbol{r}^{(m+1)}+\beta^{(m)}\boldsymbol{c}^{(m)}$$

3) 終了条件を満たせば終了し，満たさなければ m に $m+1$ を代入して 2) へ戻る．

5. 方策評価の効率に関する理論的検討

5.1 方策評価法の代数的解釈

付録のように，状態遷移確率 $\boldsymbol{p}$ により $\boldsymbol{P}$，方策 μ により $\boldsymbol{\Pi}$，$\boldsymbol{p}$ とコスト $\boldsymbol{g}$ により $\bar{\boldsymbol{g}}$ を与えるとき，(1) 式は

$$\boldsymbol{Q}^{\mu}=\bar{\boldsymbol{g}}+\boldsymbol{P\Pi}\boldsymbol{Q}^{\mu} \tag{7}$$

と表すことができる．(7) 式は $\boldsymbol{Q}^{\mu}$ を未知とした大規模な代数方程式とみなすことができ，次式が解となる．

$$\boldsymbol{Q}^{\mu}=(\boldsymbol{I}-\boldsymbol{P\Pi})^{-1}\bar{\boldsymbol{g}}=\boldsymbol{A}^{-1}\bar{\boldsymbol{g}} \tag{8}$$

方策評価を (8) 式の代数問題と捉えると，アルゴリズムは直接法型と反復法型に区分できる．前者の演算は有限回であるが，演算終了まで方策を評価できる Q 値が得られないため，一般に用いられない．後者の誤差は反復回数に対して指数関数的に減少するので，反復を途中で打ち切ることができる．さらに，学習過程では厳密に方策評価しないことが多く，より少ない反復で打ち切ることができるため反復法型が用いられる．上述の SIM 型と KSM 型のアルゴリズムはともに反復法型に区分されるが，KSM 型は有限回で解を得る直接法の性質も有している．数値計算上の誤差を考慮せず厳密に解を得るとすると，SIM 型では無限回の反復が必要であるが，KSM 型の反復回数は問題のサイズに応じて有限回に定まり，LU 分解などの直接法と同程度と見積もられる [19].

第 1 表　近似解ベクトル $\boldsymbol{Q}^{(m)}$ 更新の計算コスト

アルゴリズム	乗除算の回数
PEI	N^2K^2
Jacobi	N^2K^2+NK
Gauss-Seidel	N^2K^2+NK
SOR	N^2K^2+3NK
BCG	$2N^2K^2+7NK+2$
CGS	$2N^2K^2+12NK+5$
BiCGSTAB	$2N^2K^2+14NK+4$

5.2 SIM 型の特徴と効率評価指標

方策評価を表す代数方程式 (7) の反復解法の一般形は

$$\boldsymbol{Q}^{(m+1)}=\boldsymbol{b}+\boldsymbol{C}\boldsymbol{Q}^{(m)} \tag{9}$$

で与えられる．ここで，反復行列 $\boldsymbol{C}$ が一定の反復が SIM 型である．また，$\boldsymbol{A}$ はつぎのように一意に分解できる．

$$\boldsymbol{A}=\boldsymbol{L}+\boldsymbol{D}+\boldsymbol{U} \tag{10}$$

$\boldsymbol{D}$ は対角成分が 0 でない対角行列，$\boldsymbol{L}$ と $\boldsymbol{U}$ は対角成分が 0 の下三角と上三角の行列である．**4.1** の PEI 型，Jacobi 型，Gauss-Seidel 型，SOR 型の各反復行列は

$$\boldsymbol{C}_{\mathrm{PEI}}=\boldsymbol{P\Pi},\ \boldsymbol{C}_{\mathrm{Jacobi}}=-\boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{U}),$$
$$\boldsymbol{C}_{\mathrm{Gauss-Seidel}}=-(\boldsymbol{D}+\boldsymbol{L})^{-1}\boldsymbol{U},$$
$$\boldsymbol{C}_{\mathrm{SOR}}=(\boldsymbol{D}+\omega\boldsymbol{L})^{-1}\{(1-\omega)\boldsymbol{D}-\omega\boldsymbol{U}\}$$

である [8]．近似解ベクトル 1 回の更新に必要な乗除算の回数を第 1 表に示す．(9) 式は真値誤差 $\Delta\boldsymbol{Q}^{(m)}\equiv\boldsymbol{Q}^{(m)}-\boldsymbol{Q}^{\mu}$ により

$$\Delta\boldsymbol{Q}^{(m)}=\boldsymbol{C}^{m}\Delta\boldsymbol{Q}^{(0)} \tag{11}$$

と変形でき，スペクトル半径 $\rho(\boldsymbol{C})$(固有値の絶対値の最大値) が $\rho(\boldsymbol{C})<1$ ならば $\Delta\boldsymbol{Q}^{(m)}$ と $\boldsymbol{Q}^{(m)}$ は収束する．また，$\|\Delta\boldsymbol{Q}^{(m)}\|_2\propto\rho(\boldsymbol{C})^m$ となり，指数関数的に減少する．そのため，$\rho(\boldsymbol{C})$ を収束速度の指標とする．$\|\cdot\|_p$ は p ノルムであり，2 ノルムはユークリッドノルムを表す．

5.3 KSM 型の特徴と効率評価指標

4.2 の KSM 型は，反復法の特徴とともに，プロパー方策の方策評価において，分母が 0 となる破綻が起きない限り，最大 $n=NK$ 回の反復で正しい解を得る [19].

CG 型は，係数行列 $\boldsymbol{A}$ が正定値対称であれば，正解への収束が保証される．BCG 型と BPKSM 型は $\boldsymbol{A}$ が非対称でも正解への収束が保証される．学習問題の $\boldsymbol{A}$ は一般に正定値対称ではないが，BCG 型は係数行列を

$$\tilde{\boldsymbol{F}}\tilde{\boldsymbol{A}}\equiv\begin{bmatrix}\boldsymbol{0} & \boldsymbol{I}\\ \boldsymbol{I} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{A} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{A}^T\end{bmatrix}=\begin{bmatrix}\boldsymbol{0} & \boldsymbol{A}^T\\ \boldsymbol{A} & \boldsymbol{0}\end{bmatrix} \tag{12}$$

として CG 型を適用する場合と同等であり [25]，BPKSM 型は BCG 型の修正アルゴリズムなので，いずれも CG

型と同様に動作する．第 1 表より，1 回の近似解ベクトルの更新に必要な計算量は，従来の SIM 型に比べ，CG 型は同等，BCG 型と BPKSM 型はおよそ倍である．

CG 型は，サイズ n の正定値対称係数行列 $\boldsymbol{A}$ に対し評価値 $J(\boldsymbol{Q}^{(m)}) \equiv \frac{1}{2}(\boldsymbol{Q}^{(m)},\boldsymbol{A}\boldsymbol{Q}^{(m)}) - (\bar{\boldsymbol{g}},\boldsymbol{Q}^{(m)})$ を反復ごとに単調減少させる．それゆえ誤差の重みつきノルム $\|\Delta\boldsymbol{Q}^{(m)}\|_{\boldsymbol{A}} \equiv \sqrt{(\Delta\boldsymbol{Q}^{(m)},\boldsymbol{A}\Delta\boldsymbol{Q}^{(m)})} = \sqrt{2J(\boldsymbol{Q}^{(m)})}$ も単調減少するので KSM 型ではこれを効率評価に用いる．さらに $\boldsymbol{A}$ の固有値 $\lambda_i (0 < \lambda_1 \leq \cdots \leq \lambda_n)$ と対応する固有ベクトルを $\boldsymbol{v}_i$，$\boldsymbol{V}$ を直交行列とし，変換行列 $\boldsymbol{U} \equiv \sqrt{\boldsymbol{\Lambda}}\boldsymbol{V}^T$, $\sqrt{\boldsymbol{\Lambda}} \equiv \mathrm{diag}\left[\ \sqrt{\lambda_1},\ldots,\sqrt{\lambda_n}\ \right]$, $\boldsymbol{V} \equiv [\ \boldsymbol{v}_1,\ldots,\boldsymbol{v}_n\ ]$ による変換 $\hat{\boldsymbol{Q}}^{(m)} = \boldsymbol{U}\boldsymbol{Q}^{(m)}$, $\hat{\boldsymbol{Q}}^{\mu} = \boldsymbol{U}\boldsymbol{Q}^{\mu}$ を導入すれば，$\|\Delta\boldsymbol{Q}^{(m)}\|_{\boldsymbol{A}} = \|\hat{\boldsymbol{Q}}^{(m)} - \hat{\boldsymbol{Q}}^{\mu}\|_2$ を得るため変換後の $\hat{\boldsymbol{Q}}^{(m)}$, $\hat{\boldsymbol{Q}}^{\mu}$ に対しては通常のユークリッドノルムになる．

この変換を用いると CG 型に対する収束定理 [26] から次式が成り立ち，ν が反復回数 m における収束率の上限を与え，SIM 型のスペクトル半径に相当する．

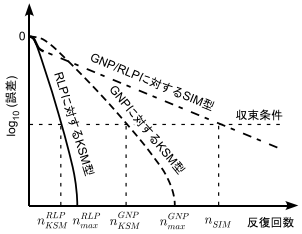
$$\left\|\hat{\boldsymbol{Q}}^{(m)} - \hat{\boldsymbol{Q}}^{\mu}\right\|_2 \leq 2\nu^m \left\|\hat{\boldsymbol{Q}}^{(0)} - \hat{\boldsymbol{Q}}^{\mu}\right\|_2 \qquad (13)$$
$$\nu \equiv \left(\sqrt{\lambda_n} - \sqrt{\lambda_1}\right) \Big/ \left(\sqrt{\lambda_n} + \sqrt{\lambda_1}\right)$$

5.4 強化学習での KSM 型の速い収束の理由

これまでの検討と数値例 [18] に基づいて，強化学習問題で提案する KSM 型が速く収束する理由を説明する．第 1 図は，横軸を反復回数，縦軸を誤差とした片対数グラフである．誤差としては，相対残差 $\|\bar{\boldsymbol{g}} - \boldsymbol{A}\boldsymbol{Q}^{(m)}\|_2/\|\bar{\boldsymbol{g}}\|_2$ または正規化された誤差の重み付きノルム $\|\Delta\boldsymbol{Q}^{(m)}\|_{\boldsymbol{A}}/\|\Delta\boldsymbol{Q}^{(0)}\|_{\boldsymbol{A}}$ を考える．数値計算上の誤差は考慮しないものとして，SIM 型と KSM 型のアルゴリズムを一般的な数値計算問題と強化学習問題に適用した場合のグラフを模式図で示す．各グラフの傾きは反復回数あたりの収束率を表し，傾きが急であるほど収束率が小さく，誤差が速く零に収束する．

まず，SIM 型について考える．**5.2** と数値例 [18] のように，SIM 型の最終的な収束率はスペクトル半径 $\rho(\boldsymbol{C})$ により定まり，傾き一定の直線状に誤差が零に収束していく．そのため，SIM 型で厳密に解を得るためには，無限回の反復が必要になる．なお，一般的な数値計算問題と学習問題の SIM 型のスペクトル半径は同じとする．

つぎに，KSM 型について考える．**5.2** で述べたように，KSM 型で厳密に解を得るために必要な反復回数が有限に定まる．反復回数は最大でも $n_{max} = n - \sum_i (n_i - 1)$ である．ただし，$n = NK$ は行列 $\boldsymbol{A}$ のサイズ，n_i は行列 $\boldsymbol{A}$ の固有値 λ_i の重複度である．KSM 型を一般的な数値計算に適用する場合，固有値がほとんど重複していないとすると，その最大反復回数は $n_{max}^{GNP} \simeq NK$ である．他方，学習問題に適用する場合，固有値 1 の重複度が少なくとも $N(K-1)$ なので [18]，その最大反復回数は $n_{max}^{RLP} \leq N+1$ である．他方，**5.3** のように誤差の重みつきノルムは単調に減少する．また，KSM 型の収束



第 1 図 SIM 型と KSM 型を GNP と RLP に適用する場合の収束特性（GNP: 一般的数値計算問題, RLP: 強化学習問題）

率は徐々に小さくなり，反復回数 n_{max} では誤差が零なので，収束率は最終的に零になる．

以上より，SIM 型と KSM 型のアルゴリズムを一般的な数値計算問題と学習問題に適用した場合のグラフは第 1 図のようになることがわかる．

いずれの数値計算でも，誤差がある値になると解が収束したとみなす．図中，SIM 型の反復回数は n_{SIM}，KSM 型を一般的な数値計算に適用すると n_{KSM}^{GNP}，KSM 型を学習問題に適用すると n_{KSM}^{RLP} である．KSM 型と SIM 型を一般的な数値計算に適用する例 [18] では，$n_{KSM}^{GNP} \simeq n_{SIM}/2$ である．KSM 型の反復あたりの計算量は SIM 型の約 2 倍であった．それゆえ，一般的な数値計算例で KSM 型が収束に要する計算量は SIM 型とほとんど同じである．他方，KSM 型と SIM 型を強化学習問題に適用する例 [18] では，$n_{KSM}^{RLP} \ll n_{SIM}/2$ である．それゆえ，強化学習問題例では，KSM 型が解の収束に要する計算量は SIM 型に比べてはるかに少ない．

このように，強化学習問題では KSM 型が SIM 型より効率的になる．

6. おわりに

方策評価を効率化し学習を加速するため，KSM 型のアルゴリズムを検討した．KSM 型は既存の SIM 型に比べ格段に計算効率が優れることが示された．その要因を問題と解法の性質に基づいて説明した．係数行列 $\boldsymbol{A} = \boldsymbol{I} - \boldsymbol{P}\boldsymbol{\Pi}$ はサイズ $n = NK$ であるが，学習問題の特性により固有値 1 の重複に起因して KSM 型アルゴリズムで最大 NK 回必要な反復が最大 $N+1$ 回に減ることが示された．また，SIM 型は収束率が一定であるが，KSM 型は反復に従って収束率が向上することが示された．この 2 点により SIM 型より反復回数が少なく，強化学習問題に対し KSM 型が効率的である．それゆえ，強化学習問題に対して従前の SIM 型よりも，KSM 型に基づくアルゴリズムが望ましい．

(2016 年 1 月 5 日受付)

参 考 文 献

[1] R. S. Sutton and A. G. Barto: *Reinforcement Learning, An Introduction*, MIT Press (1998)（三上・皆川訳: 強化学習, 森北出版 (2000)）

[2] D. P. Bertsekas and J. N. Tsitsiklis: *Neuro-Dynamic Programming*, Athena Scientific (1996)

[3] リレー解説: 強化学習の最近の発展; 計測と制御, Vol. 52, No. 1–12 (2013)

[4] M. Wiering and M. van Otteerlo (Eds.): *Reinforcement Learning*, Springer (2012)

[5] J. H. Holland: Escaping brittleness, the possibility of general-purpose learning algorithms applied to rule-based systems; *Machine Learning: An Artificial Intelligence Approach*, Vol. 2, pp. 593–632 (1986)

[6] R. S. Sutton: Learning to predict by the methods of temporal differences; *Machine Learning*, Vol. 3, pp. 9–44 (1988)

[7] C. J. C. H. Watkins and P. Dayan: Q-learning; *Machine Learning*, Vol. 8, pp. 279–292 (1992)

[8] 藤井, 泉田, 真野: 状態遷移モデルを逐次推定する強化学習の学習加速; 計測自動制御学会論文集, Vol. 42, No. 1, pp. 47–53 (2006)

[9] 泉田, 岩崎, 藤井: 強化学習における最適政策の信頼性を保証するサンプリング政策; 計測自動制御学会論文集, Vol. 46, No. 5, pp. 274–280 (2010)

[10] Cs. Szepesvári: *Algorithms for Reinforcement Learning*, Morgan and Claypool (2010)

[11] W. B. Powell: Approximate dynamic programming: *Solving the Curses of Dimensionality*, Wiley (2007)

[12] M. G. Lagoudakis and R. Parr: Least-squares policy iteration; *J. Machine Learning Research*, Vol. 4, pp. 1107–1149 (2003)

[13] X. Xu, D. Hu and X. Lu: Kernel based least squares policy iteration for reinforcement learning; *IEEE Transactions on Neural Networks*, Vol. 18, No. 4, pp. 973–992 (2007)

[14] J. Boyan: Technical update: Least-squares temporal difference learning; *Machine Learning, Special Issue on Reinforcement Learning*, Vol. 49, No. 2–3, pp. 233–246 (2002)

[15] S. Mahadevan and M. Maggioni: Proto-value functions: A Laplacian framework for learning representation and control in Markov decision processes; *J. Machine Learning Research*, Vol. 8, pp. 2169–2231 (2007)

[16] J. Morimoto and K. Doya: Robust reinforcement learning; *Neural Computation*, Vol. 17, No. 2, pp. 335–359 (2005)

[17] K. Senda and Y. Tani: Autonomous robust skill generation using reinforcement learning with plant variation; *Advances in Mechanical Engineering*, Vol. 2014, 276264, 12 pages (2014)

[18] 泉田, 服部, 幸田: 強化学習における方策評価の効率化による学習の加速; 計測自動制御学会論文集, Vol. 49, No. 7, pp. 696–702 (2013)

[19] 広中（編）: 第 2 版 現代数理科学辞典, 丸善 (2009)

[20] 森, 杉原, 室田: 線形計算, 岩波書店 (1994)

[21] C. T. Kelley: *Iterative Methods for Linear and Nonlinear Equations*, Society for Industrial and Applied Mathematics (1995)

[22] M. R. Hestenes and E. Stiefel: Methods of conjugate gradients for solving linear systems; *J. Research of the National Bureau of Standards*, Vol. 49, No. 6, pp. 409–436 (1952)

[23] R. Fletcher: Conjugate gradient methods for indefinite system; *Lecture notes in Mathematics*, Vol. 506, pp. 73–89 (1976)

[24] H. A. van der Vorst: Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems; *SIAM Journal on Scientific and Statistical Computing*, Vol. 13, No. 2, pp. 631–644 (1992)

[25] 名取: BCG 法と CGS 法; 数理解析研究所講義録, Vol. 613, pp. 135–143 (1987)

[26] 山本: 行列解析の基礎, サイエンス社 (2010)

付　　録

$\boldsymbol{e}_i$ は i 番目成分が 1 で残りが 0 の列ベクトルである.

$$
\begin{aligned}
[ik] &\equiv (i-1)\times K+k \\
\boldsymbol{\pi}(s_i) &\equiv [\ \pi(s_i,u_1),\pi(s_i,u_2),\ldots,\pi(s_i,u_K)\ ]^T \\
\boldsymbol{\Pi}(s_i) &\equiv \boldsymbol{e}_i\boldsymbol{\pi}^T(s_i) \\
\boldsymbol{\Pi} &\equiv [\ \boldsymbol{\Pi}(s_1),\boldsymbol{\Pi}(s_2),\ldots,\boldsymbol{\Pi}(s_N)\ ] \\
\boldsymbol{p}_i(u_k) &\equiv [\ p_{i0}(u_k),p_{i1}(u_k),\ldots,p_{iN}(u_k)\ ]^T \\
\boldsymbol{p}_i &\equiv [\ \boldsymbol{p}_i^T(u_1),\boldsymbol{p}_i^T(u_2),\ldots,\boldsymbol{p}_i^T(u_K)\ ]^T \\
\boldsymbol{p} &\equiv [\ \boldsymbol{p}_1^T,\boldsymbol{p}_2^T,\ldots,\boldsymbol{p}_N^T\ ]^T \\
\boldsymbol{p}_{i|j\neq 0}(u_k) &\equiv [\ p_{i1}(u_k),p_{i2}(u_k),\ldots,p_{iN}(u_k)\ ]^T \\
\boldsymbol{P}_i &\equiv [\ \boldsymbol{p}_{i|j\neq 0}(u_1),\boldsymbol{p}_{i|j\neq 0}(u_2),\ldots,\boldsymbol{p}_{i|j\neq 0}(u_K)\ ]^T \\
\boldsymbol{P} &\equiv [\ \boldsymbol{P}_1^T,\boldsymbol{P}_2^T,\ldots,\boldsymbol{P}_N^T\ ]^T \\
\boldsymbol{g}(s_i,u_k) &\equiv [\ g(s_i,u_k,s_0),\ldots,g(s_i,u_k,s_N)\ ]^T \\
\bar{\boldsymbol{G}}(s_i,u_k) &\equiv \boldsymbol{e}_{[ik]}\boldsymbol{g}^T(s_i,u_k) \\
\bar{\boldsymbol{G}}(s_i) &\equiv [\ \bar{\boldsymbol{G}}(s_i,u_1),\bar{\boldsymbol{G}}(s_i,u_2),\ldots,\bar{\boldsymbol{G}}(s_i,u_K)\ ] \\
\bar{\boldsymbol{G}} &\equiv [\ \bar{\boldsymbol{G}}(s_1),\bar{\boldsymbol{G}}(s_2),\ldots,\bar{\boldsymbol{G}}(s_N)\ ] \\
\bar{\boldsymbol{g}} &\equiv \bar{\boldsymbol{G}}\boldsymbol{p} \\
\boldsymbol{Q}(s_i) &\equiv [\ Q(s_i,u_1),Q(s_i,u_2),\ldots,Q(s_i,u_K)\ ]^T \\
\boldsymbol{Q} &\equiv [\ \boldsymbol{Q}^T(s_1),\boldsymbol{Q}^T(s_2),\ldots,\boldsymbol{Q}^T(s_N)\ ]^T
\end{aligned}
$$

著 者 略 歴

泉田　啓（正会員）

1988 年大阪府立大学大学院博士前期課程修了．同年大阪府立大学工学部助手，同大学助教授，金沢大学助教授，同大学教授を経て，2008 年京都大学工学研究科教授となり現在に至る．1996〜97 年ミシガン州立大学客員教授など兼任．動物の運動知能などの研究に従事．博士（工学）．1994 年システム制御情報学会賞論文賞など受賞．計測自動制御学会などの会員．