強化学習の在庫問題への適用と方策オン型・方策オフ型学習の比較

A comparison between on-policy learning and off-policy one in reinforcement learning for an inventory problem

髙橋勇人，星野満博
秋田県立大学大学院システム科学技術研究科
Hayato Takahashi and Mitsuhiro Hoshino
Graduate School of Systems Science and Technology, Akita Prefectural University

1　はじめに

現在，画像認識や音声認識など様々な AI が開発され，第 3 次 AI ブームの到来と言われている．これは，2010 年代からの深層学習の発展が大きな影響を与えたとされている．機械学習のカテゴリのひとつである強化学習 (Reinforcement Learning) も深層学習の発展に伴い，ここ数年で研究が盛んになってきている．

強化学習はマルコフ決定過程 (Markov Decision Process) の枠組みでモデリングされており，マルコフ決定過程で定式化される問題に対して強化学習手法が代替手法となるか検証したい．本研究では，マルコフ決定過程で定式化される在庫管理問題を強化学習問題として定式化し，Python で実装する．扱う強化学習手法は方策オン型及び方策オフ型の TD 学習 (Temporal Difference Learning) の 2 種類とし，これらを比較する．

2　強化学習

強化学習のマルコフ決定過程による枠組みは，以下の図 1 のように表現される．エージェントと呼ばれる意思決定主体が行動によって環境に働きかけ，環境の変化として状態と報酬を観測する．環境とエージェント間の相互作用を繰り返し，期待報酬和などによって定義される価値を最大にする方策を求める．方策とは行動を決定するルールのことで，どのように行動を選択すれば価値が最大になるかを学習する．

強化学習には大きく，モデルベース手法，モデルフリー手法という 2 つの分類がある．モデルベース手法はマルコフ決定過程の推移確率などが既知で環境のモデルを構築できる場合に用いることができる手法で，動的計画法などが挙げられる．モデルフリー手法は環境のモデルが構築できなくても学習を行うことができる手法で，モンテカルロ法，TD 学習などが挙げられる．シミュレーション等で次の状態を観測しながら学習を行うため，学習にマルコフ決定過程の推移確率を必要としない．また，方策オン型学習，方策オフ型学習という分類

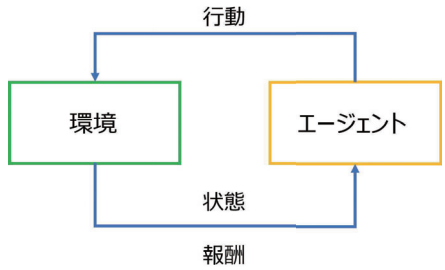

図 1　環境とエージェント間の相互作用

もあり，TD 学習では，方策オン型学習として SARSA，方策オフ型学習として Q-Learning が代表的である．

TD 学習はモンテカルロ法などのようにシミュレーションや実験などで環境とエージェントの相互作用を繰り返す手法である．TD(0) 法では動的計画法のように現在の価値を 1 ステップ先の価値と報酬で推定する．環境とエージェント間の相互作用を記述するマルコフ決定過程は以下の要素の組で表現される．

(i) 状態集合 $S = \{s^1, s^2, \ldots, s^N\}$
(ii) 行動集合 $A = \{a^1, a^2, \ldots, a^M\}$
(iii) 推移関数 $T : S \times A \times S \to [0, 1]$

$$T(s, a, s') = p(s_{t+1} = s \,|\, s_t = s, a_t = a) \tag{1}$$

(iv) 報酬関数 $R : S \times A \times S \to \mathbb{R}$

$$R(s, a, s') = \mathbb{E}[r_{t+1} \,|\, s_t = s, a_t = a, s_{t+1} = s] \tag{2}$$

また，方策を π とし，ある状態 s においてある行動 a が選択される確率を $\pi(a \,|\, s)$ と表す．TD 学習では動的計画法などと異なり状態価値 (3) と行動価値 (4) という 2 種類の価値関数を用いる．ここで，割引率 $\gamma \in [0, 1]$，$\pi \in \Pi_{\mathrm{S}}$(定常方策) とする．

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \,|\, s_0 = s\right], \forall s \in S \tag{3}$$

$$Q^{\pi}(s, a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \,|\, s_0 = s, a_0 = a\right], \forall s \in S, \forall a \in A \tag{4}$$

TD(0) 法での SARSA，Q-Learning のアルゴリズムを以下に示す．

SARSA，Q-Learning のアルゴリズム

1 行動価値 $Q(s,a)$ を初期化する．
2 各エピソードに対して以下を繰り返す．
　2.1 初期状態 s を決定する．
　2.2 ϵ-Greedy などによって初期状態 s での行動 a を決定する．
　2.3 エピソード中の各ステップに対して以下を繰り返す．
　　2.3.1 行動 a を実行し，報酬 r と次の状態 s' を観測する．
　　2.3.2 ϵ-Greedy などによって次の状態 s' での行動 a' を決定する．
　　2.3.3 学習式に従って行動価値 $Q(s,a)$ を更新する．
　　SARSA の場合
　　$Q^{\text{new}}(s,a) = (1-\alpha)Q(s,a) + \alpha(r + \gamma Q(s',a'))$
　　Q-Learning の場合
　　$Q^{\text{new}}(s,a) = (1-\alpha)Q(s,a) + \alpha(r + \gamma \max_{a' \in A}\{Q(s',a')\})$
　　2.3.4 $s \leftarrow s', a \leftarrow a'$ とする．
　　2.3.5 状態 s がエピソードの終端状態なら繰り返しを終了する．

この 2 つの違いは学習式にあり，SARSA では行動選択の方策が価値の更新に直接影響し，Q-Learning では最大値を用いるため直接影響しない．行動選択の方策として用いる ϵ-Greedy は，$[0,1]$ の一様乱数を発生させ，ϵ 以上なら価値が最大の行動を選ぶ (利用)，ϵ 未満ならランダムに行動を選ぶ (探索) 方策である．行動価値に関して Greedy な行動方策 $\arg\max_{a \in A}\{Q(s,a)\}$ だと局所解に陥る可能性があるため，一定の確率 ϵ でランダムに行動を選ぶ．

3　在庫問題の定式化

ここでは，Puterman[1] の定期発注方式の在庫管理問題を扱う．在庫管理の流れを図 2 に示す．

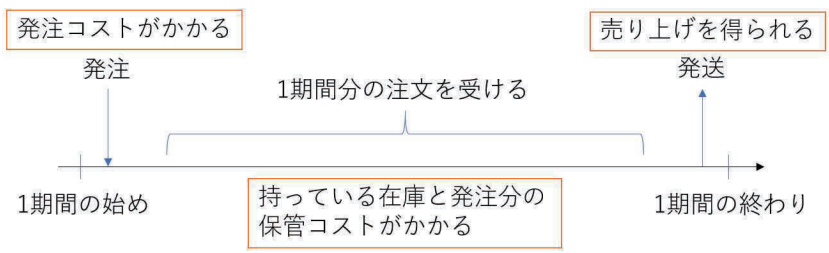

図 2　1 期間の在庫管理の流れ

この在庫管理問題について以下を仮定する．

(a) 発注量の決定は固定された各期間の初めに行われる．
(b) 需要に対する供給は固定された各期間の終わりに満たされる．
(c) リードタイムは 0 とする．
(d) 需要が在庫を超える場合，超過需要は失われる．
(e) 収益，費用，需要分布は定常とする．
(f) 製品は単位でのみ販売する．
(g) 在庫管理は先入れ先出しによって行う．
(h) 倉庫の容量は M 単位とする．（$M \in \mathbb{N}$）

この在庫管理問題を以下のように定式化する．

(i) 有限期間 $T = \{1, 2, \ldots, N\}, N < \infty$
(ii) 有限状態空間 $S = \{0, 1, 2, \ldots, M\}, s \in S$
(iii) 有限行動空間 $A_s = \{0, 1, 2, \ldots, M - s\}, a \in A_s$
(iv) 決定ルール $\delta_t : S \to \{0, 1, \ldots, M\}, \delta_t(s) \in A_s, t = 1, 2, \ldots, N-1$
(v) 政策 $\pi = (\delta_1, \delta_2, \ldots, \delta_{N-1})$
(vi) 需要分布 $p_d = P\{D = d\}, d = 0, 1, 2, \ldots$
(vii) 期待収益 (f: 単位当たり収益)

$$F(s+a) = \sum_{d=0}^{s+a-1} fdp_d + f(s+a) \sum_{d=s+a}^{\infty} p_d$$

(viii) 発注費 (c_1: 固定発注費，c_2: 単位当たり変動発注費)

$$O(a) = \begin{cases} c_1 + c_2 a & \text{if } a > 0 \\ 0 & \text{if } a = 0 \end{cases}$$

(ix) 在庫保管費 (c_3: 単位当たり在庫保管費)

$$H(s+a) = c_3(s+a)$$

(x) 報酬関数

$$r_t(s,a) = F(s+a) - O(a) - H(s+a), t = 1, 2, \ldots, N-1$$

(xi) 末期在庫価値

$$r_N(s) = g(s), t = N$$

(xii) 推移確率

$$p\left(s' \mid s, a\right) = \begin{cases} 0 & \text{if } s+a < s' \leq M \\ p_{s+a-s'} & \text{if } 0 < s' \leq s+a \leq M \\ \sum_{d=s+a}^{\infty} p_d & \text{if } s+a \leq M, s' = 0 \end{cases}$$

$v_N^{\pi}(s)$ を期待報酬和，Π_{MD} を確定的政策集合とする．ここで，$v_N^{\pi} = \left(\sum_{l=0}^{N-1} P^l\right) r$ である．

$$\begin{array}{ll} \text{maximize} & v_N^{\pi}(s) \\ \text{subject to} & \pi \in \Pi_{\mathrm{MD}} \end{array}$$

以下の再帰式に従って最大期待報酬和 $v_N^*(s)$ と最適政策 π^* を導出する．

$$u_t^*\left(s\right) = \max_{a \in A_s} \left\{ r\left(s,a\right) + \sum_{s' \in S} p\left(s' \mid s, a\right) u_{t+1}^*\left(s'\right) \right\},$$
$$u_N^*\left(s\right) = r_N\left(s\right)$$

このとき得られる最適政策は，非定常 (σ, Σ) 戦略と呼ばれる．

$$\delta_t^*(s) = \begin{cases} \Sigma_t - s & \text{if } s = \sigma_t \\ 0 & \text{if } s > \sigma_t \end{cases}$$

しかし，一般に強化学習では定常方策を扱うため，このまま適応できない．そのため，SARSA，Q-Learning で学習できるように以下のように定式化の一部を変更する．

(i) 有限状態空間 $S = \{(t,s) \mid t = 1, 2, \ldots, N, s = 0, 1, 2, \ldots, M\}, t \in T, s \in S$
(ii) 収益 $F(s+a-s') = f(s+a-s')$
(iii) 報酬関数 $r_t(s,a,s') = F(s+a-s') - O(a) - H(s+a), t = 1, 2, \ldots, N-1$

エピソード中のステップ数と期間を対応させ，1 エピソードが N ステップとなるようにすることで非定常方策が得られるようにする．また，モデルフリー手法で学習するため，推移確率は計算に用いない．

4 SARSA，Q-Learning による学習

4.1 数値例

Python3 で実装し，計算には Mac(チップ: Apple M1 8core, メモリ: 16GB) を用いた．数値例として，$N = 4, M = 3, f = 8, c_1 = 4, c_2 = 2, c_3 = 1, p_d = 0.25(\text{if } d = 0), 0.5(\text{if } d = 1), 0.25(\text{if } d = 2), 0(\text{if } d = 3), \gamma = 1, \alpha = 0.1, \epsilon = 0.1$ を用いた．表 1 に Q-Learning での結果を示す．価値の収束は得られなかったが，非定常 (σ, Σ) 戦略は得られた．

表 1　Q-Learning で得られる方策と期待報酬和

状態	行動	価値 (期待報酬和)
(1,0)	3	4.639
(1,1)	0	8.182
(1,2)	0	11.944
(1,3)	0	13.759
(2,0)	2	0.850
(2,1)	0	5.929
(2,2)	0	9.075
(2,3)	0	11.024
(3,0)	0	0.000
(3,1)	0	4.531
(3,2)	0	6.595
(3,3)	0	7.038

4.2　SARSA，Q-Learning の比較

数値計算では行動価値の収束が得られなかった．そのため，得られた価値と動的計画法によって導出した最適な価値の相対誤差が一定未満となるまでのエピソード数，計算時間を比較する．アルゴリズムは相対誤差の最大値

$$\max\left\{\frac{|(\text{TD での価値}) - (\text{DP での価値})|}{(\text{DP での価値})}\right\}$$

が 0.1 未満となったとき終了することとする．図 3，4 にそれぞれ SARSA，Q-Learning で学習した ($M = 10, N = 10$) 際の行動価値の最大更新量 (左図)，相対誤差 (右図) を示す．また，図 5，6 にそれぞれ倉庫容量 M を変えた際の終了時エピソード数と計算時間を示す．行動価値の最大更新量から収束に至っていないことが確認できる．相対誤差が 0.1 未満となったエピソード数は SARSA がおよそ 25000，Q-Learning がおよそ 12000 となり，Q-Learning のほうが少ないエピソード数で最適な価値の値に近づいている．一般に SARSA よりも Q-Learning のほうが収束が速く，このモデルでも同様のことが言えると考えられる．終了時エピソード数と計算時間ともに倉庫容量が多くなるほど差も大きくなっているが，動的計画法の計算時間のほうが短く，かなり計算に時間を要する実装になっている可能性がある．

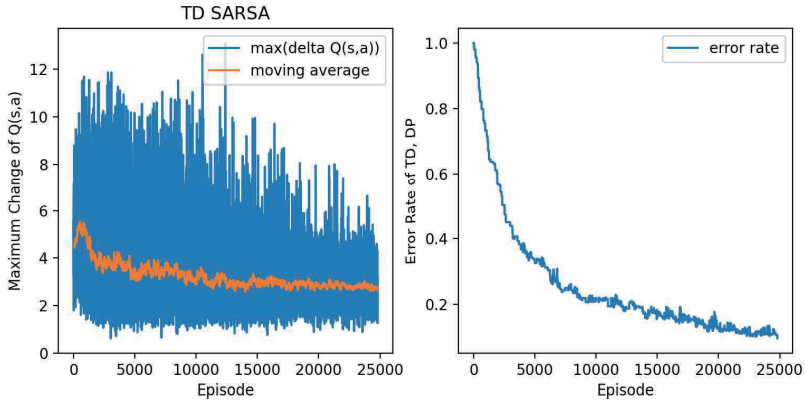

図 3　SARSA での相対誤差

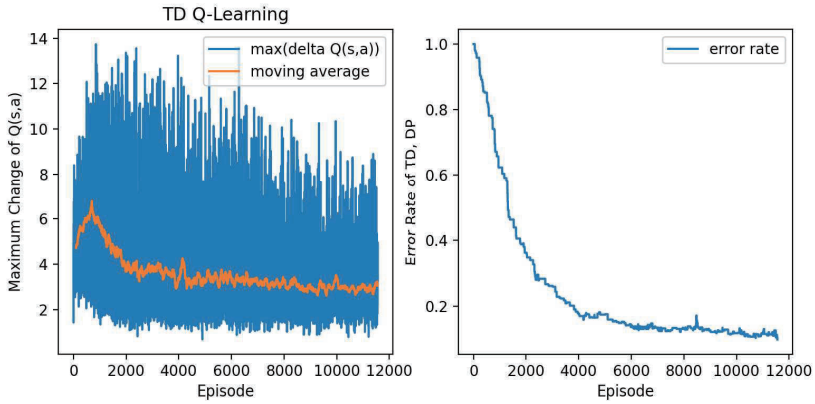

図 4　Q-Learning での相対誤差

5　まとめと今後の課題

Puterman[1] の在庫管理問題を強化学習問題として実装した．行動価値の収束は得られなかったが，動的計画法によって計算した最適解への近づき方で SARSA と Q-Learning を比較し，Q-Learning のほうが少ないエピソード数で最適解へ近づいていることがわかった．

今後は収束しなかった原因を検証し，定式化や実装等を見直す．また，状態や行動の数が大幅に増えるようなモデルを考える場合，ニューラルネットワークを用いた関数近似手法での実装を検討する必要がある．

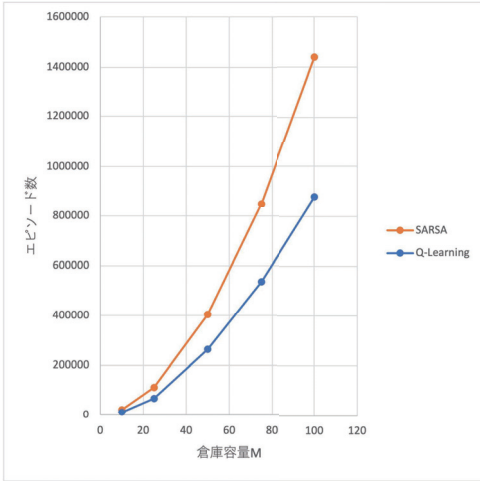

図 5　終了時のエピソード数

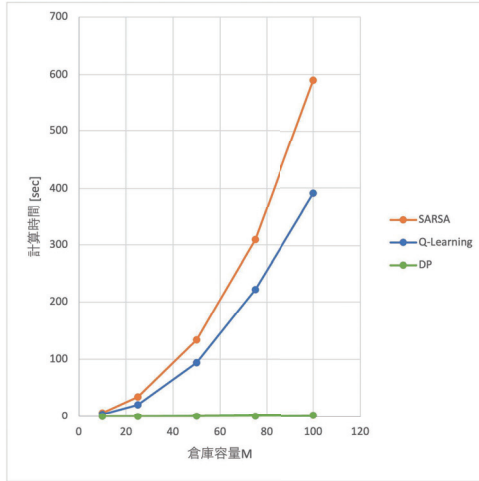

図 6　計算時間

参考文献

[1] M. L. Puterman, *Markov Decision Processes Discrete Stochastic Dynamic Programming*, Wiley-Interscience, 2005

[2] D. J. White, *Markov Decision Processes*, Wiley, 1993

[3] Richard S. Sutton, Andrew G. Barto, *Reinforcement Learning, second edition: An Introduction*, Bradford Books, 2018

[4] 牧野 貴樹, これからの強化学習, 森北出版, 2016

[5] 曽我部 東馬, 強化学習アルゴリズム入門: 「平均」からはじめる基礎と応用, オーム社, 2019