計測自動制御学会東北支部 第280回研究集会 (2013年5月29日)
資料番号 280-6

じゃんけんゲームの戦略学習

Strategy Learning for the Game of Scissors-paper-rock

○佐藤 隆雅（岩手大・院），西山 清（岩手大）

○Takao Satou, Kiyoshi Nishiyama

岩手大学, Iwate University

キーワード： 強化学習(Reinforcement Learning) 環境(Enviroment) 戦略(Strategy) Q-leaning 価値関数(Value Functions)

連絡先： 〒020-8551 盛岡市上田4-3-5 岩手大学 工学部 情報システム工学科 西山研究室
西山清, Tel.: (019)621-6475, Fax.: (019)621-6475, E-mail: nisiyama@cis.iwate-u.ac.jp

1. はじめに

強化学習とは、環境に対する試行錯誤的なインタラクションを通じて環境に適応する学習制御の枠組みである[1)]。学習の主体であるエージェントは環境に関する知識を持たない。また、環境は状態遷移及び報酬の与えられ方は確率的であるものが想定される。このような環境において、エージェントは試行錯誤により適切な行動規則を獲得していく。「何をすべきか」をエージェントに報酬という形で指示しておくだけで「どのように実現するか」をエージェントが学習によって自動的に獲得する枠組みとなっている。環境に不確実性や計測不能な未知のパラメータが存在すると、タスクの達成方法やゴールへの到達方法は設計者にとって自明ではない。よって、ロボットへタスクを遂行するための制御規則をプログラムすることは設計者にとって重労働である。ところが、達成すべき目標を報酬によって指示することは前記に比べれば遥かに簡単である。そのため、タスク遂行のためのプログラミングを強化学習で自動化することにより、設計者の負荷軽減が期待できる。

「じゃんけん」における最適政策はグー・チョキ・パーをそれぞれ1/3の確率で出すことであることが知られている。しかし、実際に人間同士が対戦した際に出された手を調べると、グー・チョキ・パーのそれぞれが出される確率はほぼ1/3ずつであったが、一手前との相関を見るとある程度のばらつきが見受けられた。本研究ではじゃんけんゲームの戦略学習に強化学習を適用し、このような「癖」に対する戦略の学習を行えるかどうか検証し、またその学習性能について評価する。

2. 強化学習

2.1 概要

強化学習では、まず学習の主体であるエージェントとそれをとりまく環境を定義する。エージェ

ントは図1で表されるような環境との試行錯誤的な相互作用により、状態から行動への写像を学習する。

エージェントは

(1) 環境から状態sを観測

(2) 状態sに基づき行動aを決定

(3) 行動aを実行することで環境は状態s'へ移行し、その状態遷移に応じた報酬rを得る

(4) 学習が終了するまで(1) ~ (3)のサイクルを繰り返す

という一連の環境との相互作用により最も効率の良い行動の仕方(戦略)を環境から学んでいく。(3)によってエージェントが得る報酬は、0や負の数、すなわちペナルティに相当する場合もある。

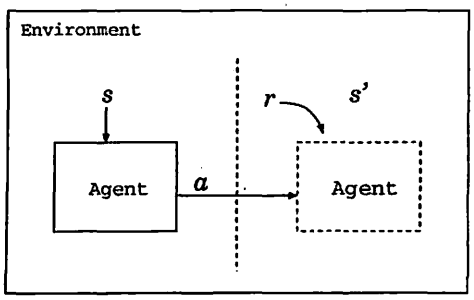

Fig. 1　環境とエージェントの相互作用

2.2　環境モデル

強化学習の多くは、扱う環境がマルコフ決定過程(Markov Decision Process: MDP)としてモデル化できることを仮定している。環境のとりうる状態の集合を$S=\{s_1,s_2,\cdots,s_n\}$、エージェントがとりうる行動の集合を$A=\{a_1,a_2,\cdots,a_m\}$、報酬の集合を$W=\{r_1,r_2,\cdots,r_l\}$とそれぞれ表す。ここで、nは状態集合の要素数、mは行動集合の要素数、lは報酬集合の要素数である。時刻tで環境がある状態$s\in S$にあるとき、エージェントがある行動aを実行すると、次の時刻に環境は確率的に状態$s'\in S$へ遷移する。定常性を仮定し、その遷移確率を$P^a(s,s')$と表す。このとき確率$\Pr(r|s')$で環境からエージェントへ報酬rが与えられるが、その期待値を$R^a(s,s')$により表す。エージェントにおける状態集合から行動集合への確率分布を政策と呼び、$\pi(s,a)$と表す。

$$\pi(s,a) = \Pr(a_t=a|s_t=s)=\Pr(a|s) \quad (1)$$

$$\begin{aligned} P^a(s,s') &= \Pr(s_{t+1}=s'|s_t=s,a_t=a) \\ &= \Pr(s'|s,a) \quad (2) \end{aligned}$$

$$\begin{aligned} R^a(s,s') &= E^{s'}\{r_t|s_t=s,a_t=a,s_{t+1}=s'\} \\ &= \sum_{r_t\in W} r_t \Pr(r_t|s_t=s,a_t=a,s_{t+1}=s') \\ &= \sum_{r_t\in W} r_t \Pr(r_t|s')\Pr(s'|s,a) \\ &= \sum_{r_t\in W} r_t \Pr(r_t|s')P^a(s,s') \quad (3) \end{aligned}$$

ここで、s_t、a_t、r_tはそれぞれ時刻tにおけるエージェントの状態、行動、報酬である。

2.3　最適政策

エージェントは多くの場合、以下のような割引報酬の期待合計を最大化することを目的とする。

$$R_t = r_t+\gamma r_{t+1}+\gamma^2 r_{t+2}+\cdots=\sum_{k=0}^{\infty}\gamma^k r_{t+k} \quad (4)$$

ここで、$\gamma(0\leq\gamma\leq 1)$は割引率であり、将来得られる報酬を重視するか、短期間に得られる報酬を重視するかを調節するパラメータである。

ある政策πを用いたときの状態sからs'への状態遷移確率$P^\pi(s,s')$は以下の式で表される。

$$\begin{aligned} P^\pi(s,s') &= \Pr(s'|s)=\frac{\Pr(s',s)}{\Pr(s)} \\ &= \sum_{a\in A}\frac{\Pr(s',s,a)}{\Pr(s)} \\ &= \sum_{a\in A}\frac{\Pr(s',s,a)}{\Pr(s)}\frac{\Pr(s,a)}{\Pr(s,a)} \\ &= \sum_{a\in A}\frac{\Pr(s',s,a)}{\Pr(s,a)}\frac{\Pr(a,s)}{\Pr(s)} \end{aligned}$$

$$= \sum_{a\in A} \Pr(s'|s,a)\Pr(a|s)$$
$$= \sum_{a\in A} P^a(s,s')\pi(s,a) \quad (5)$$

ただし、Prは政策πに対応する確率分布を表す。

また、状態sにおいて、ある政策πに従った行動を行うときの報酬の期待値$R^\pi(s,s')$は以下の式で表される。

$$R^\pi(s,s') = E^\pi\{r_t|s_t=s, s_{t+1}=s'\} \quad (6)$$
$$= \sum_{r_t\in W} r_t \Pr(r_t|s')P^\pi(s,s')$$
$$= \sum_{r_t\in W} r_t \sum_{a\in A} \Pr(r_t|s')P^a(s,s')\pi(s,a)$$
$$= \sum_{a\in A} \pi(s,a) \sum_{r_t\in W} r_t \Pr(r_t|s')P^a(s,s')$$
$$= \sum_{a\in A} \pi(s,a)R^a(s,s') \quad (7)$$

環境がMDPであるということは、将来の状態は現在の状態とそのときとる行動にのみ依存し、過去の状態や行動の系列には依存しないことを意味する。MDPにおいてエージェントが定常な政策πをとるとき、割引報酬の期待合計は時間に関係せず環境の状態のみに依存するため、状態sの関数として表すことができる。これを状態価値関数と呼び、$V^\pi(s)$で表す。

$$V^\pi(s) = E^\pi\{R_t|s_t=s\}$$
$$= \sum_{r\in W} R_t \sum_{s'\in S} \Pr(r|s')P^\pi(s,s') \quad (8)$$

ここで、1次のマルコフ性$\Pr(a|s',s) = \Pr(a|s')$より

$$\Pr(a|s) = \frac{\Pr(a,s)}{\Pr(s)}$$
$$= \frac{\sum_{s'\in S}\Pr(a,s',s)}{\Pr(s)}$$
$$= \sum_{s'\in S} \frac{\Pr(a,s',s)}{\Pr(s',s)} \cdot \frac{\Pr(s',s)}{\Pr(s)}$$
$$= \sum_{s'\in S} \Pr(a|s',s)\Pr(s'|s)$$
$$= \sum_{s'\in S} \Pr(a|s')\Pr(s'|s)$$
$$= \sum_{s'\in S} \Pr(a|s')P^\pi(s,s') \quad (9)$$

また、

$$P^\pi(s_t,s_{t+2}) = \sum_{s_{t+1}\in S} P^\pi(s_t,s_{t+1})P^\pi(s_{t+1},s_{t+2}) \quad (10)$$

これより、$V^\pi(s)$は以下のように表される。

$$V^\pi(s) = E^\pi\{R_t|s_t=s\}$$
$$= E^\pi\{\sum_{k=0}^{\infty}\gamma^k r_{t+k}|s_t=s\}$$
$$= E^\pi\{r_t+\gamma\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|s_t=s\}$$
$$= E^\pi\{r_t|s_t=s\}+\gamma E^\pi\{\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|s_t=s\}$$
$$= R^\pi(s)$$
$$+\gamma\sum_{r\in W}\sum_{s''\in S} R_{t+1}\Pr(r|s'')P^\pi(s,s'')$$
$$= R^\pi(s)+\gamma\sum_{r\in W}\sum_{s''\in S} R_{t+1}\Pr(r|s')$$
$$\cdot\sum_{s'\in S} P^\pi(s,s')P^\pi(s',s'')$$
$$= R^\pi(s)+\gamma\sum_{s'\in S} P^\pi(s,s')$$
$$\cdot\left(\sum_{r\in W} R_{t+1}\sum_{s''\in S} Pr(r|s'')P^\pi(s',s'')\right)$$
$$= R^\pi(s)+\gamma\sum_{s'\in S} P^\pi(s,s')V^\pi(s') \quad (11)$$

ここで、定常性より$V^\pi(s) = E^\pi\{R_t|s_t=s\} = E^\pi\{R_{t+1}|s_{t+1}=s\}$を用いた。

全ての状態sにおいて$V^\pi(s) \geqq V^{\pi'}(s)$となるとき、政策$\pi$は政策$\pi'$より優れているといえる。MDPにおいては他のいかなる政策よりも優れた、もしくは同等な政策が少なくとも1つ存在し、これを最適政策π^*と呼ぶ。最適政策をとるときの状態価値関数は以下のようになる。

$$V^{\pi^*}(s) = \max_\pi V^\pi(s) \text{ for all } s\in S \quad (12)$$

2.4 政策反復法

状態遷移確率$P^a(s,s')$と報酬の与えられ方$R^a(s,s')$が与えられているとき最適政策を求める手法として政策反復法がある。以下にこれを説明する。

政策πに従うときの各状態における状態遷移確率

と報酬の与えられ方を以下の$\boldsymbol{P}^\pi$と$\boldsymbol{R}^\pi$で定義する。

$$\boldsymbol{P}^\pi = \begin{bmatrix} P^\pi(s_1,s_1) & P^\pi(s_1,s_2) & \cdots & P^\pi(s_1,s_n) \\ P^\pi(s_2,s_1) & P^\pi(s_2,s_2) & \cdots & P^\pi(s_2,s_n) \\ \vdots & \vdots & \ddots & \vdots \\ P^\pi(s_n,s_1) & P^\pi(s_n,s_2) & \cdots & P^\pi(s_n,s_n) \end{bmatrix} \tag{13}$$

$$\boldsymbol{R}^\pi = \begin{bmatrix} R^\pi(s_1) \\ R^\pi(s_2) \\ \vdots \\ R^\pi(s_n) \end{bmatrix} \tag{14}$$

式(11)より

$$\begin{aligned} V^\pi(s) &= R^\pi(s) + \gamma \sum_{s' \in S} P^\pi(s,s') V^\pi(s') \\ &= R^\pi(s) \\ &+\gamma \sum_{s' \in S} P^\pi(s,s') \left[R^\pi(s') + \gamma \sum_{s'' \in S} P^\pi(s',s'') V^\pi(s'') \right] \\ &= R^\pi(s) + \gamma \sum_{s' \in S} P^\pi(s,s') R^\pi(s') \\ &\quad +\gamma^2 \sum_{s' \in S} P^\pi(s,s') \sum_{s'' \in S} P^\pi(s',s'') V^\pi(s'') \\ &= R^\pi(s) + \gamma \sum_{s' \in S} P^\pi(s,s') R^\pi(s') \\ &\quad +\gamma^2 \sum_{s' \in S} P^\pi(s,s') \sum_{s'' \in S} P^\pi(s',s'') R^\pi(s'') + \cdots \end{aligned} \tag{15}$$

であるから、政策πに従うときの各状態における状態価値$\boldsymbol{V}^\pi$は以下のように表される。

$$\begin{aligned} \boldsymbol{V}^\pi &= \begin{bmatrix} V^\pi(s_1) \\ V^\pi(s_2) \\ \vdots \\ V^\pi(s_n) \end{bmatrix} \\ &= \begin{bmatrix} R^\pi(s_1) \\ R^\pi(s_2) \\ \vdots \\ R^\pi(s_n) \end{bmatrix} + \gamma \begin{bmatrix} \sum_{s' \in S} P^\pi(s_1,s') R^\pi(s') \\ \sum_{s' \in S} P^\pi(s_2,s') R^\pi(s') \\ \vdots \\ \sum_{s' \in S} P^\pi(s_n,s') R^\pi(s') \end{bmatrix} \\ &+\gamma^2 \begin{bmatrix} \sum_{s' \in S} P^\pi(s_1,s') \sum_{s'' \in S} P^\pi(s',s'') R^\pi(s'') \\ \sum_{s' \in S} P^\pi(s_2,s') \sum_{s'' \in S} P^\pi(s',s'') R^\pi(s'') \\ \vdots \\ \sum_{s' \in S} P^\pi(s_n,s') \sum_{s'' \in S} P^\pi(s',s'') R^\pi(s'') \end{bmatrix} + \cdots \\ &= \boldsymbol{R}^\pi + \gamma \boldsymbol{P}^\pi \boldsymbol{R}^\pi + \gamma^2 (\boldsymbol{P}^\pi)^2 \boldsymbol{R}^\pi + \cdots \\ &= \sum_{t=0}^{\infty} \gamma^t (\boldsymbol{P}^\pi)^t \boldsymbol{R}^\pi \end{aligned} \tag{16}$$

ここで$\boldsymbol{V}_k^\pi = \sum_{t=0}^{k} \gamma^t (\boldsymbol{P}^\pi)^t \boldsymbol{R}^\pi$ とすると、

$$\begin{aligned} \boldsymbol{V}_{k+1}^\pi &= \sum_{t=0}^{k+1} \gamma^t (\boldsymbol{P}^\pi)^t \boldsymbol{R}^\pi \\ &= \boldsymbol{R}^\pi + \sum_{t=1}^{k+1} \gamma^t (\boldsymbol{P}^\pi)^t \boldsymbol{R}^\pi \\ &= \boldsymbol{R}^\pi + (\gamma \boldsymbol{P}^\pi \boldsymbol{R}^\pi + \gamma^2 (\boldsymbol{P}^\pi)^2 \boldsymbol{R}^\pi + \gamma^3 (\boldsymbol{P}^\pi)^3 \boldsymbol{R}^\pi + \cdots \\ &= \boldsymbol{R}^\pi + \gamma \boldsymbol{P}^\pi (\boldsymbol{R}^\pi + \gamma \boldsymbol{P}^\pi \boldsymbol{R}^\pi + \gamma^2 (\boldsymbol{P}^\pi)^2 \boldsymbol{R}^\pi + \cdots \\ &= \boldsymbol{R}^\pi + \gamma \boldsymbol{P}^\pi \sum_{t=0}^{k} \gamma^t (\boldsymbol{P}^\pi)^t \boldsymbol{R}^\pi \\ &= \boldsymbol{R}^\pi + \gamma \boldsymbol{P}^\pi \boldsymbol{V}_k^\pi \end{aligned} \tag{17}$$

ここで、$\boldsymbol{V}_k^\pi \to \boldsymbol{V}^\pi$ as $k \to \infty$、すなわち定常状態が存在するとき、

$$\begin{aligned} \boldsymbol{V}^\pi &= \boldsymbol{R}^\pi + \gamma \boldsymbol{P}^\pi \boldsymbol{V}^\pi \\ \boldsymbol{V}^\pi - \gamma \boldsymbol{P}^\pi \boldsymbol{V}^\pi &= \boldsymbol{R}^\pi \\ (\boldsymbol{I} - \gamma \boldsymbol{P}^\pi) \boldsymbol{V}^\pi &= \boldsymbol{R}^\pi \\ \boldsymbol{V}^\pi &= (\boldsymbol{I} - \gamma \boldsymbol{P}^\pi)^{-1} \boldsymbol{R}^\pi \end{aligned} \tag{18}$$

となる。

ここで、ある状態sにおいてのみ政策π'に従い、それ以降は政策πに従って行動するときの状態価値を

$$V^{\pi,\pi'}(s) = R^{\pi'}(s) + \gamma \sum_{s' \in S} P^{\pi'}(s,s') V^\pi(s') \tag{19}$$

とすると、

$$V^{\pi,\pi'}(s) > V^\pi(s)$$

であれば、π'に従う政策の価値はπの価値よりも改善されることが証明されている。よって、以下の手順により最適政策を得ることができる。

1) 確定的な政策πについて$\boldsymbol{V}^\pi$を計算する。

2) すべてのsにおいて$V^{\pi,\pi'}(s)$が最大となるようなπ'を得る。

3) ここで$\pi' \approx \pi$のとき、πは最適政策なので処理を打ち切る。そうでなければ$\pi \leftarrow \pi'$として手順1より繰り返す。

3. Q-learning

3.1 価値関数

状態sにおいて行動aを行い、その後は政策πに従った行動をとるときの割引報酬の期待合計を行動価値関数と呼び、$Q^\pi(s,a)$と表す。

$$Q^\pi(s,a) = \sum_{s' \in S} P^a(s,s')(R^a(s,s') + \gamma V^\pi(s')) \quad (20)$$

最適な行動価値関数は以下のようになる。

$$Q^*(s,a) = \max_\pi Q^\pi(s,a), \ ^\forall s \in S, \ ^\forall a \in A \quad (21)$$

これは次の方程式(Bellman方程式)の解である。

$$\begin{aligned} Q^*(s,a) &= E\{r_t + \gamma \max_{a' \in A} Q^*(s_{t+1}, a') | s_t = s, a_t = a\} \\ &= \sum_{s' \in S} P^a(s,s') \left[R^a(s,s') + \gamma \max_{a' \in A} Q^*(s', a') \right] \end{aligned} \quad (22)$$

状態遷移確率$P^a(s,s')$と報酬の与えられ方$R^a(s,s')$が与えられれば価値関数の値を計算により求めることができるが、実環境においては環境モデルが予め与えられるとは限らず、$P^a(s,s')$や$R^a(s,s')$は通常未知である。そのため、エージェントはなんらかの方法で「価値」を推定しなければならない。

最適なQ関数が与えられれば、状態sにおいてQ関数の値が最大となる行動aを行うことで最適に行動することができる。

強化学習においては、環境との相互作用の試行錯誤により価値関数を推定していく。

Q-learningは最適な行動価値関数$Q^*(s_t,a_t)$を試行錯誤により推定するものである。以下に$Q^*(s_t,a_t)$の推定値である$Q(s_t,a_t)$の更新式を示す[2]。

$$\begin{aligned} Q(s_t,a_t) \leftarrow \ & (1-\alpha) Q(s_t,a_t) \\ & + \alpha (r_t + \gamma \max_{a' \in A} Q(s_{t+1}, a')) \end{aligned} \quad (23)$$

ここで、αは学習率、γは割引率であり、s_{t+1}は状態s_tで行動a_tをとったときの遷移先の状態を表す。図2にQ-learningのアルゴリズムの概要を示す。

1. 環境の状態s_tを観測
2. 任意の行動選択法に従い行動a_tを実行
3. 環境より報酬r_tを受け取る
4. 状態遷移後の環境の状態s_{t+1}を観測
5. 式(23)により行動価値関数を更新
6. 時間ステップをtから$t+1$に進め、手順1へ戻る

Fig. 2 Q-learningアルゴリズム

このQ-learningには次の収束定理が知られている[1]。

「エージェントの行動選択において、全ての行動を十分な回数選択し、かつ学習率αが$\sum_{t=0}^{\infty} \alpha(t) \to \infty$かつ$\sum_{t=0}^{\infty} \alpha(t)^2 < \infty$を満たす時間$t$の関数となっているとき、Q-learningのアルゴリズムで得るQ値は確率1で最適なQ値に収束する(概収束)。ただし、環境はエルゴート性を有する離散有限マルコフ決定過程であることを仮定する。」

3.2 行動選択法

上記の収束定理は、全ての行動を十分な回数選択しさえすれば行動選択方法には依存せずに成り立つ。よって行動選択はランダムでもよい。しかし、強化学習ではまだQ値が収束していない学習の途中においてもなるべく多くの報酬を得るような行動選択を求められることが多い。学習に応じて徐々に挙動を改善していくような行動選択方法として、

- ϵ-greedy 選択:
  ϵの確率でランダム、それ以外は最大のQ値を持つ行動を選択する。

- ボルツマン選択:
  $e^{Q(s,a)/T}$に比例した割合で行動選択する。ただし、Tは時間とともにゼロに近付く。

などの手法が提案されている。

なお、本研究ではボルツマン選択(式(24))を用いる。

$$\pi(s,a) = \frac{e^{Q(s,a)/T}}{\sum_{b\in A} e^{Q(s,b)/T}} \tag{24}$$

4. じゃんけんの戦略学習

4.1 Q-learningの適用

じゃんけんにおける戦略をQ-learningを用いて学習させることを考える。まず、じゃんけんにおける環境を定義した。環境の状態はエージェントの対戦相手の出した手の組み合わせとした。例えば1手前までの手を見る場合は3通り、2手前までの手を見る場合は9通りの状態が存在する。本研究においては1手前に出された手を環境として用いた。

ゲームが始まると、まずエージェントは現在の環境の状態sを観測する。次にQ値から得られる政策πにより出す手aを決定する。ここで、エージェントの対戦相手の手はある一定の戦略から決定するものとする。エージェントとその対戦相手の出す手が決定すると、その行動を行い、環境の状態が変化する。遷移後の状態をs'とし、環境の変化によって報酬rを得る。以上のようにして得られたs, a, s', rと現在のQ値を用いて、Q値をより最適な値へと更新していくことでじゃんけんの戦略を学習する。

4.2 じゃんけんゲームにおける環境

じゃんけんゲームにおいて、環境は以下の通りとなる。

- 状態集合$S = \{s_1, s_2, s_3\}$
- 行動集合$A = \{a_1, a_2, a_3\}$
- 状態s ··· 対戦相手によって出された現在の手
- 状態s' ··· 対戦相手の次の手
- 行動a ··· エージェントの次の手

なお、ここでじゃんけんの手と行動及び状態は以下の通りとする。

Table 1　状態及び行動と手の対応

s	a	手
s_1	a_1	グー
s_2	a_2	チョキ
s_3	a_3	パー

ゲームの流れは以下のようになる。

1) 対戦相手の現在の手sを観測

2) エージェントが次の手aを決定する

3) 対戦相手の手s'が決定される

じゃんけんゲームにおいては環境の状態遷移確率はエージェントの行動aには依存せず、状態sと遷移先の状態s'にのみ依存する。

$$\Pr(s'|s,a) = \Pr(s'|s) \tag{25}$$

また、報酬はエージェントの行動aと遷移先の状態s'によって決まり、状態sには依存しない。

$$R^a(s,s') = R^a(s') \tag{26}$$

式(5),(25)より、環境の状態遷移確率は以下の式であり、これは政策πに依存しない。

$$\begin{aligned} P^\pi(s,s') &= \sum_{a\in A} \Pr(s'|s,a)\pi(s,a) \\ &= \Pr(s'|s)\sum_{a\in A}\pi(s,a) \\ &= \Pr(s'|s) \end{aligned} \tag{27}$$

式(7),(25),(26)より、

$$\begin{aligned} R^\pi(s) &= \sum_{s'\in S}\sum_{a\in A}\pi(s,a)R^a(s,s')\Pr(s'|s,a) \\ &= \sum_{s'\in S}\Pr(s'|s)\sum_{a\in A}\pi(s,a)R^a(s') \end{aligned} \tag{28}$$

$$V^\pi(s) = R^\pi(s) + \gamma\sum_{s'\in S} P^\pi(s,s')V^\pi(s') \tag{29}$$

5. 実験

5.1 学習に用いたデータ

まず、学習に用いるデータに関して説明する。被験者AとBが80回じゃんけんの対戦を行い、その際にAが出した手における一手前の手とその次の手の関係から、ある手が出された次に出された手の頻度を表2に、表3にAが出したグー・チョキ・パーそれぞれの数を示した。

Table 2 被験者Aにおけるある手が出された後の次の手の数とその確率

現在の手	次の手					
	グー		チョキ		パー	
	数	確率	数	確率	数	確率
グー	9	0.333	11	0.407	7	0.259
チョキ	8	0.296	4	0.148	15	0.556
パー	10	0.385	12	0.462	4	0.154

Table 3 出された各手の総数

グー	チョキ	パー
27	27	26

5.2 政策反復法による解

MDPにおいて、状態遷移確率と報酬が既知であれば、政策反復法を用いて最適な政策を得ることができる。今回実装したじゃんけんゲームにおいては、対戦相手の一手前の手を状態として用いているため、表2の一手前の手をs、次の手をs'とすると、表2における確率は状態遷移確率$P^a(s,s')$とみなすことができる。また、$P^a(s,s')$はエージェントの行動aにかかわらず一定である。

また、じゃんけんにおいて勝った時の報酬を1、あいこの時の報酬を0、負けた時の報酬を-1と設定する。割引率γは0.2とする。

このゲームにおいては報酬関数の値は遷移前の状態sには依存しない。よって以下のようになる。

Table 4 $R^a(s,s') = R^a(s')$

	s'_1	s'_2	s'_3
a_1	0.0	1.0	-1.0
a_2	-1.0	0.0	1.0
a_3	1.0	-1.0	0.0

2.4章に示した政策反復法を用い、以上の$P^a(s,s')$及び$R^a(s,s')$において最適政策π^*を求めた結果を以下に示す。

Table 5 得られた最適政策$\pi^*(s,a)$

	a_1	a_2	a_3
s_1	1.0	0.0	0.0
s_2	0.0	1.0	0.0
s_3	1.0	0.0	0.0

Table 6 最適政策に従うときの状態価値$V^{\pi^*}(s)$

s_1	0.207
s_2	0.322
s_3	0.365

Table 7 最適政策を用いた場合のエージェントの勝敗; 試合数 = 80

	勝ち	負け	分け
数	38	19	23
%	47.50	23.75	28.75

よって、被検者Aの出した手の通りに手を出す対戦相手を用いてエージェントに強化学習による学習を行わせた場合、表5のような戦略に収束することが期待される。

5.3 Q-learningによる学習結果

実際にAが出した手のデータを用いてQ-learningによる戦略学習を行った。学習の際に用いたパラメータは、学習率の初期値$\alpha_0 = 0.5$、割引率$\gamma = 0.2$、行動選択法はボルツマン選択、温度パラメータの初期値$T_0 = 1.0$、エージェントに対する報酬は、勝った際1.0、負けた際-1.0、あいこの際0.0である。

80ステップ、400ステップ、800ステップの学習を

行い、それぞれ学習率の減衰率α_d、温度パラメータの減衰率T_dは学習終了時に学習率$\alpha \cong 0.01$、温度パラメータ$T \cong 0.1$となるような値を設定している。また、乱数の種はいずれも1234とした。

学習後の$Q(s,a)$,$\pi(s,a)$の値及び得られたπを用いて被験者Aの手と80試合の対戦を行った結果を以下に示す。

学習ステップ数 = 80

$\alpha_d = 0.95228$、$T_d = 0.97163$

Table 8　80ステップ時点での$Q(s,a)$,$\pi(s,a)$; $T = 0.100$

	$Q(s,a)$			$\pi(s,a)$		
	a_1	a_2	a_3	a_1	a_2	a_3
s_1	0.340	-0.286	-0.096	0.985	0.002	0.013
s_2	-0.681	-0.329	0.179	0.000	0.006	0.994
s_3	0.380	0.073	-0.281	0.955	0.044	0.001

Table 9　πを用いたときの勝敗数

	勝ち	負け	分け
数	30	14	36
%	37.50	17.50	45.00

学習ステップ数 = 400

$\alpha_d = 0.99027$、$T_d = 0.99426$

Table 10　400ステップ時点での$Q(s,a)$,$\pi(s,a)$; $T = 0.100$

	$Q(s,a)$			$\pi(s,a)$		
	a_1	a_2	a_3	a_1	a_2	a_3
s_1	0.138	-0.175	-0.495	0.957	0.042	0.002
s_2	-0.473	0.340	0.113	0.000	0.906	0.094
s_3	0.328	-0.331	-0.119	0.987	0.001	0.011

Table 11　πを用いたときの勝敗数

	勝ち	負け	分け
数	36	18	26
%	45.00	22.50	32.50

学習ステップ数 = 800

$\alpha_d = 0.99512$、$T_d = 0.99713$

Table 12　800ステップ時点での$Q(s,a)$,$\pi(s,a)$; $T = 0.100$

	$Q(s,a)$			$\pi(s,a)$		
	a_1	a_2	a_3	a_1	a_2	a_3
s_1	0.133	-0.223	-0.501	0.970	0.028	0.002
s_2	-0.656	0.315	0.150	0.000	0.838	0.162
s_3	0.332	-0.225	-0.279	0.994	0.004	0.002

Table 13　πを用いたときの勝敗数

	勝ち	負け	分け
数	34	19	27
%	42.50	23.75	33.75

6. 考察

政策反復法による解(表5)と400及び800ステップのQ-learningによる学習結果を比較すると、最終的に同じ政策が得られており、十分なステップ数の学習を行うことで最適な政策を得ることができることがわかる。

実際に人間との対戦中に学習を行うことを考えると、少ない対戦数で最適な選択を行えるようになることが望ましい。しかし80ステップの学習においては最適政策と異なる政策が学習されており、少ないステップ数で最適政策を得るための学習パラメータや手法の検討が必要である。

参考文献

1) 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, Vol.38, No.10, pp.618-623, 1999.

2) 長行 康男, 伊藤 実: "2体エージェント確率ゲームにおける他エージェントの政策推定を利用した強化学習法", 電子情報通信学会論文誌, Vol.J86-D-I, No.11, pp.821-829, 2003.

3) Michael L. Littman: "Markov games as a framework for multi-agent reinforcement learning", Proc. 11th International Conference on Machine Learning, pp.157-163, USA, July 1994.

4) 山田 知明, 西山 清: "1プレーヤーサッカーゲームにおける戦略学習", 計測自動制御学会 東北支部 第228回研究集会, 228-1, 2006.