状態遷移に不確実性を伴う環境下での意志決定問題

1　はじめに

実問題における多くの場面において，状態遷移に不確実性を伴うダイナミクスのもとでの意志決定が求められている．例えば，在庫管理や生産システム管理，計算機システムにおける資源割り当て，ロボットの制御など，その分野は多岐にわたる．そのような問題を定式化し，洗練された方法論を与えるモデルの一つにマルコフ決定過程 (Markov decision process: MDP) がある．状態遷移のダイナミクスが既知である場合において，最適な意志決定を求めることは「プラニング」と呼ばれる．

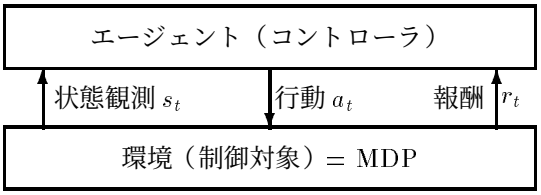


図 1: MDP におけるプラニングの枠組．行為主体『エージェント』の報酬獲得が最大になる制御規則を求める．

MDP におけるプラニングの枠組は，図 1に示すように最適制御問題になっている．行為主体『エージェント』と制御対象『環境』は，以下のやりとりを繰り返す．

1. エージェントは時刻 t において環境の状態観測 s_tに基づいて意志決定を行い，行動 a_tを出力
2. エージェントの行動により，環境は s_{t+1}へ状態遷移し，その遷移に応じた報酬 r_tをエージェントに与える．
3. 時刻 t を $t+1$ に進めてステップ 1 へ戻る

エージェントの利得（return: 最も単純な場合，報酬の総計）が最大となるような制御規則，すなわち状態観測から行動出力へのマッピング（政策（policy）と呼ばれる）を求める．

【プラニングの例：在庫管理問題】

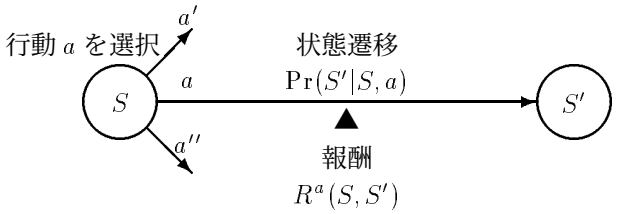
・状態：在庫量×市場動向
・行動：品物を仕入れる数量
・状態遷移：（次の日の在庫量）＝（現在の在庫量）＋（仕入れた数量）－（注文を受けた数量）
　ただし在庫はゼロより小さくなることはない
　市場動向に応じて注文を受ける数量が確率的に変化する．
・報酬＝（注文を受けた利益）－（在庫量に応じた管理費）－（仕入れ費用）
　ただし在庫がゼロの場合，注文はキャンセルされる
・利益を最大にするには，どのような制御規則で品物を仕入れたらよいか？

2　マルコフ決定過程 (MDP) とは?

●マルコフ性 :とは？
状態 s'への遷移が，そのときの状態 s と行動 a にのみ依存し，それ以前の状態や行動には関係ないこと.

●エルゴート性 :とは？
任意の状態 s からスタートし，無限時間経過した後の状態分布確率は最初の状態とは無関係になること.

- 環境のとりうる状態の集合 $\mathcal{S} = \{s_1, s_2, \cdots, s_n\}$
- エージェントが選択可能な行動の集合 $\mathcal{A} = \{a_1, a_2, \cdots, a_\ell\}$
- 状態遷移のダイナミクス：
  状態 s において行動 a を実行したときの状態 s'へ遷移する確率 $\Pr(s'|s,a)$ により記述.
  状態遷移は離散時間ステップで発生.
- 報酬のダイナミクス：
  状態 s において行動 a を実行した後，状態 s'へ遷移した場合の報酬の期待値 $R^a(s,s')$ により記述.



【MDP によるモデル化の例：迷路問題】

s_0	s_1	s_2	s_3	s_4
s_5	s_6	s_7	s_8	s_9
s_{10}	s_{11}	s_{12}	s_{13}	s_{14}
s_{15}	s_{16}	s_{17}	s_{18}	s_{19}
s_{20}	s_{21}	s_{22}	s_{23}	Goal

状態 S : (goal を除く) 迷路のマス目に相当, 24 状態

行動 A : 上下左右への移動 (4 種類)

遷移規則 $\Pr(s'|s,a)$: 壁が存在する方向へは移動できない等を記述
ゴールと同時に一様分布で他のマスへ飛ぶ

報酬 $R^a(s,s')$: ゴールした時 100，それ以外 0

図中の○が現在のロボットの位置．すなわち $s_t = s_{12}$.

図 2: 迷路問題を MDP で表現した例.

- エージェントは現在ロボットがいる場所の番号だけを観測する.
- 迷路中での移動が確率 $\Pr(s'|s,a)$ によって表されているため，行動を選択してもある確率で失敗したり別の方向へ移動してしまうなどのダイナミクスも表現できる.
- 報酬の設定を変えれば，壁に衝突したら罰 (負の報酬) を与えたり，移動コストを考慮する問題も表現可能.
- 状態遷移の不確実性や移動コスト等の扱いが求められると，単なる経路探索では解けない難しい問題となる.

3　マルコフ解析

● 政策πは，エージェントの制御規則である．
状態 s で行動 a をとる確率を表す関数$\pi(s,a) = \Pr(a_t = a|s_t = s)$ として定義されることが多い．
● 定常政策とは，時間とともに変化することのない政策．
ある定常政策πをとり続けるとき，状態 s において状態 s'へ遷移する確率を $P^\pi(s,s')$ で表すと
$P^\pi(s,s') = \sum_a \Pr(s'|s,a)\pi(s,a)$ である．よって定常政策πのもとでは状態遷移規則は以下の $n \times n$ の正方行列 P^πで表現される．同様に報酬の期待値 R^πは以下のように $1 \times n$ の行列で表される．

$$P^\pi = \begin{bmatrix} P^\pi(s_1,s_1) & P^\pi(s_1,s_2) & \cdots & P^\pi(s_1,s_n) \\ P^\pi(s_2,s_1) & P^\pi(s_2,s_2) & \cdots & P^\pi(s_2,s_n) \\ \vdots & \vdots & \ddots & \vdots \\ P^\pi(s_n,s_1) & P^\pi(s_n,s_2) & \cdots & P^\pi(s_n,s_n) \end{bmatrix} \tag{1}$$

$$R^\pi = \begin{bmatrix} R^\pi(s_1) \\ R^\pi(s_2) \\ \vdots \\ R^\pi(s_n) \end{bmatrix} = \begin{bmatrix} \sum_{s'\in\mathcal{S}} \sum_{a\in\mathcal{A}} \pi(a,s_1)\, R^a(s_1,s')\, \Pr(s'|s_1,a) \\ \sum_{s'\in\mathcal{S}} \sum_{a\in\mathcal{A}} \pi(a,s_2)\, R^a(s_2,s')\, \Pr(s'|s_2,a) \\ \vdots \\ \sum_{s'\in\mathcal{S}} \sum_{a\in\mathcal{A}} \pi(a,s_n)\, R^a(s_n,s')\, \Pr(s'|s_n,a) \end{bmatrix} \begin{matrix} \leftarrow\text{状態 } s_1\text{の報酬の期待値} \\ \leftarrow\text{状態 } s_2\text{の報酬の期待値} \\ \\ \leftarrow\text{状態 } s_n\text{の報酬の期待値} \end{matrix} \tag{2}$$

このように環境の性質をマトリクスの数学的性質へ帰着して考えるのがマルコフ解析である．

●kステップ後にエージェントが状態 s に存在する確率の計算方法：
エージェントが時刻 kに状態 s に存在する確率を U_k^πとして以下のようにマトリクス表現する．

$$U_k^\pi = \begin{bmatrix} U_k^\pi(s_1) \\ U_k^\pi(s_2) \\ \vdots \\ U_k^\pi(s_n) \end{bmatrix} \begin{matrix} \leftarrow\text{エージェントが状態 } s_1\text{に存在する確率} \\ \leftarrow\text{エージェントが状態 } s_2\text{に存在する確率} \\ \\ \leftarrow\text{エージェントが状態 } s_n\text{に存在する確率} \end{matrix} \tag{3}$$

初期状態分布が U_0^πのとき，kステップ後の状態分布 U_k^πは，式 1，3 を用いて以下のように計算される：

$${}^t(U_k^\pi) \;=\; {}^t(U_0^\pi)(P^\pi)^k \tag{4}$$

ただし記号 t はマトリクスの転置を表す．エルゴート的な MDP では無限ステップ経過後の状態分布 U_∞^πが存在する．これを定常分布または平衡分布と呼ぶ．定常分布は以下の平衡方程式の解である：

$${}^t(U_\infty^\pi) \;=\; {}^t(U_\infty^\pi)P^\pi \tag{5}$$

この方程式だけでは未知数の数に対して式が１つ足りない．U_∞^πが確率分布で，要素の合計が１になる条件を用いる．

●kステップ後に得る直接報酬の期待値の計算方法：
初期状態分布が U_0^πのとき，直接報酬の期待値 $E\{r_0\}$ は以下の式でスカラーの値として計算される：

$$E\{r_0\} \;=\; {}^t(U_0^\pi)R^\pi \tag{6}$$

よって，初期状態分布が U_0^πのとき，エージェントが kステップ後に得る直接報酬の期待値 $E\{r_k\}$ は，式 1，3より

$$E\{r_k\} \;=\; {}^t(U_k^\pi)R^\pi \;=\; {}^t(U_0^\pi)(P^\pi)^k R^\pi \tag{7}$$

定常分布における直接報酬の期待値${}^tU_\infty^\pi R^\pi$は，１ステップあたりの平均報酬を表す．

【マルコフ解析の例：オモチャ製造業者の問題】[3]

- 玩具の製造業者が，市場における彼の製品の人気の状態を観測し，各状態に応じて適切な決定を下す.
- 状態 s_1：製品が市場に人気がある
- 状態 s_2：製品が市場に人気なし
- 状態 s_1における行動 a_1：広告をしない，行動 a_2：広告を出す
- 状態 s_2における行動 a_1：研究をしない，行動 a_2：研究する
- 報酬：売上から経費を差し引いた金額

状態 s	行動 a	推移確率		報酬		直接報酬の期待値
		$\Pr(s_1\|s,a)$	$\Pr(s_2\|s,a)$	$R^a(s,s_1)$	$R^a(s,s_2)$	
s_1: 人気あり	a_1: 広告なし	0.5	0.5	9	3	$6 = 0.5\times 9 + 0.5\times 3$
	a_2: 広告あり	0.8	0.2	4	4	$4 = 0.8\times 4 + 0.2\times 4$
s_2: 人気なし	a_1: 研究なし	0.4	0.6	3	-7	$-3 = 0.4\times 3 + 0.6\times(-7)$
	a_2: 研究あり	0.7	0.3	1	-19	$-5 = 0.7\times 1 + 0.3\times(-19)$

状態 s_1では常に行動 a_1，状態 s_2でも常に行動 a_1をとる政策をπ_{11}とすると，状態遷移マトリクス $P^{\pi_{11}}$および報酬の期待値マトリクス $R^{\pi_{11}}$は

$$P^{\pi_{11}} = \begin{bmatrix} 0.5 & 0.5 \\ 0.4 & 0.6 \end{bmatrix}, \qquad R^{\pi_{11}} = \begin{bmatrix} 0.5\times 9 + 0.5\times 3 \\ 0.4\times 3 + 0.6\times(-7) \end{bmatrix} = \begin{bmatrix} 6 \\ -3 \end{bmatrix} \tag{8}$$

エージェントが時刻 $t=0$ において状態 s_1にいるとき，状態分布マトリクスは $U_0^{\pi_{11}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ である.

よって政策π_{11}をとる場合，1 ステップ後の状態分布 $U_1^{\pi_{11}}$は以下のとおり

$${}^tU_1^{\pi_{11}} = {}^tU_0^{\pi_{11}} P^{\pi_{11}} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 0.5 & 0.5 \\ 0.4 & 0.6 \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix} \tag{9}$$

同様に kステップ後の状態分布 $U_k^{\pi_{11}}$は以下のようになる：

$${}^tU_k^{\pi_{11}} = {}^tU_0^{\pi_{11}} (P^{\pi_{11}})^k = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 0.5 & 0.5 \\ 0.4 & 0.6 \end{bmatrix}^k \tag{10}$$

政策π_{11}をとり，初期状態が s_1のとき，ステップ $t=0$ における直接報酬の期待値 $E\{r_0\}$ は以下のように計算する：

$$E\{r_0\} = {}^t(U_0^{\pi}) R^{\pi} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 6 \\ -3 \end{bmatrix} = 6 \tag{11}$$

同様に kステップ後に得る直接報酬の期待値 $E\{r_k\}$ は以下のとおり：

$$E\{r_k\} = {}^t(U_0^{\pi}) (P^{\pi_{11}})^k R^{\pi} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 0.5 & 0.5 \\ 0.4 & 0.6 \end{bmatrix}^k \begin{bmatrix} 6 \\ -3 \end{bmatrix} \tag{12}$$

4　割引報酬による評価：最適な政策とは？

●プラニングの目的は，ある評価値が最大になる政策を見付けること

●1 ステップあたり $1-\gamma$の確率で活動停止するエージェントが得る報酬の合計を利得 (return) と呼ぶ．
ただしγは割引率と呼ばれ，$0 \leq \gamma \leq 1$である．この評価は，割引報酬による評価と呼ばれる．

●評価値はベクトル表現：マルコフ決定過程においてエージェントが定常政策πをとるとき，利得の期待値は，時間には関係なく状態 s だけに依存する．利得の期待値は状態 s の value と呼ばれ，定常政策πのもとで状態 s の関数として表される value を state-value 関数と呼び $V^\pi(s)$ で表す．式 1,2を用いると，この $V^\pi(s)$ を要素としてマトリクス表現すると，以下の方程式が成り立つ．

$$
\begin{aligned}
V^\pi &= \begin{bmatrix} V^\pi(s_1) \\ V^\pi(s_2) \\ \vdots \\ V^\pi(s_n) \end{bmatrix} \begin{matrix} \leftarrow \text{状態 } s_1 \text{からスタートした場合の報酬合計の期待値} \\ \leftarrow \text{状態 } s_2 \text{からスタートした場合の報酬合計の期待値} \\ \vdots \\ \leftarrow \text{状態 } s_n \text{からスタートした場合の報酬合計の期待値} \end{matrix} \\
&= R^\pi + \gamma P^\pi R^\pi + \gamma^2 (P^\pi)^2 R^\pi + \cdots = \sum_{t=0}^{\infty} \gamma^t (P^\pi)^t R^\pi \\
&= R^\pi + \gamma P^\pi V^\pi \\
&= (I - \gamma P^\pi)^{-1} R^\pi \qquad (13)
\end{aligned}
$$

● 最適政策と最適 state-value 関数：

- 2つの政策πとπ'およびそれぞれの政策の評価関数 V^πおよび $V^{\pi'}$を考える．
  全ての状態 s において $V^\pi(s) \geq V^{\pi'}(s)$ となるとき，政策πはπ'より優れているという．
- マルコフ決定過程では，他のどんな政策よりも優れた，あるいは同等な政策が確定的 (deterministic) な政策の集合中に少なくとも 1 つ存在する．これを最適政策π^*という．
- 最適政策は複数存在することもあるが，
  全ての最適政策は唯一の state-value 関数を共有する．これは最適な state-value 関数 V^*と呼ばれる．

【迷路問題においてランダム政策をとる場合の value 関数の値】

0.58	0.81	1.39	0.74	0.42
0.61	1.12	3.24	6.54	0.29
0.91	1.62	1.12	12.7	24.6
2.12	4.06	10.2	20.9	47.4
1.13	0.65	0.45	14.5	Goal

●割引率$\gamma = 0.9$ として value 関数 $V^\pi(s)$ を計算

●関数 $V^\pi(s)$ は，ゴールから遠いほど低い地形となる

●遷移規則 $\Pr(s'|s,a)$ ：ゴール時にランダム遷移する以外は
状態遷移に不確実性無し

迷路問題においてエージェントがランダムに行動選択する場合の value 関数 V^πの値．DP の反復計算を 100 回行って求めた．これは，1 ステップあたり 0.1 の確率で故障するロボットが任意の場所からスタートしてランダムに行動選択をとり続けた場合において未来に得る報酬合計の期待値と等価．

【迷路問題における最適 value 関数の値】

115	128	142	128	115
128	142	158	176	104
142	158	142	195	217
158	176	195	217	241
142	128	115	195	Goal

●割引率$\gamma = 0.9$ として最適 value 関数 $V^*(s)$ を計算

●関数 $V^*(s)$ は，ゴール状態が最も高く，遠いほど 低い地形となる

●遷移規則 $\Pr(s'|s,a)$ ：ゴール時にランダム遷移する以外は
状態遷移に不確実性無し

●最も高い方向へと移動することがゴールへの最短パス＝最適政策

迷路問題における最適 value 関数 V^*の値．DP の反復計算を 100 回行って求めた．これは，1 ステップあたり 0.1 の確率で故障するロボットが任意の場所からスタートして最適政策をとり続けた場合において未来に得る報酬合計の期待値と等価.

【演習問題】

オモチャ製造業者の問題において，
状態 s_1では常に行動 a_1，状態 s_2では常に行動 a_2をとる政策をπ_{12}，
状態 s_1では行動 a_2，状態 s_2で行動 a_1をとる政策を π_{21}，
状態 s_1でも状態 s_2でも行動 a_2をとる政策を π_{22}とする．
以下を計算せよ．

1. それぞれの政策π_{12}, π_{21}, π_{22}における遷移確率 $P^{\pi_{12}}$, $P^{\pi_{21}}$, $P^{\pi_{22}}$
   および直接報酬の期待値 $R^{\pi_{12}}$, $R^{\pi_{21}}$, $R^{\pi_{22}}$を求めよ．
2. 割引率$\gamma = 0.5$ のとき，各政策π_{11}, π_{12}, π_{21}, π_{22}における value 関数 $V^{\pi_{11}}$, $V^{\pi_{12}}$, $V^{\pi_{21}}$, $V^{\pi_{22}}$ を求めよ．
3. 割引率$\gamma = 0.5$ のとき，最適な政策はπ_{11}, π_{12}, π_{21}, π_{22}のどれであるか答えよ．
4. 割引率$\gamma = 0.9$ のとき，各政策π_{11}, π_{12}, π_{21}, π_{22}における value 関数 $V^{\pi_{11}}$, $V^{\pi_{12}}$, $V^{\pi_{21}}$, $V^{\pi_{22}}$ を求めよ．
5. 割引率$\gamma = 0.9$ のとき，最適な政策はπ_{11}, π_{12}, π_{21}, π_{22}のどれであるか答えよ．

5 プラニング：最適政策を求める方法は？

割引報酬による評価では，最適な政策は確定的 (deterministic) な (確率的ではない) 政策集合中に存在する．確定的な政策集合は離散 MDP では有限なので，確定的な全政策について value を計算し，最大の value を持つ政策を探せばよいのだが，探索空間は状態数の指数オーダーになってしまう．そこで下記のように少ない計算量で済む方法を用いて最適政策を探す．
また，状態数が大きくなるとマトリクスの逆行列演算が大変なので以下の DP のテクニックを用いる．

5.1 Howard の政策反復法 (Policy Iteration Method)

環境の状態遷移確率 $\Pr(s'|s,a)$ および報酬関数 $R^a(s,s')$ が与えられたとき，ある政策πを決めると，式 1,2より P^π と R^πがすぐに決まり，方程式 13の解は $V^\pi = (I-\gamma P^\pi)^{-1}R^\pi$ を計算することにより得る．ここで，ある状態 s においてπとは別の政策π'に従って行動をとるとき

$$R^{\pi'}(s) + \gamma \sum_{s' \in \mathcal{S}} P^{\pi'}(s,s')V^\pi(s') > V^\pi(s) \tag{14}$$

ならば，その状態 s だけは政策π'に従う新しい政策の value はπの value よりも改善されることが証明されている [3]．式 14はπの value だけを用いて簡単に計算できる．よって全状態において $R^{\pi'}(s) + \gamma \sum_{s' \in \mathcal{S}} P^{\pi'}(s,s')V^\pi(s')$ が最大になるように政策πをπ'へ改善し，$\pi \leftarrow \pi'$として再び $V^\pi = (I-\gamma P^\pi)^{-1}R^\pi$ を計算して同様の操作を繰り返せばよい．確定的な政策集合は離散 MDP では有限であるから，上記の政策改善を確定的な政策集合中で行うと，わずかな有限回数で最適政策を得る．政策反復法の手順を以下にまとめる：

1. 確定的 (deterministic) な政策πについて，$V^\pi = (I-\gamma P^\pi)^{-1}R^\pi$ を計算する．
2. 全状態において $R^{\pi'}(s) + \gamma \sum_{s' \in \mathcal{S}} P^{\pi'}(s,s')V^\pi(s')$ が最大になるように政策πを確定的政策π'へ改善する．
3. $\pi' \equiv \pi$のとき，πは最適政策なので処理を打ち切る．さもなければ$\pi \leftarrow \pi'$として手順 1 より繰り返す．

5.2 動的計画法 (Dynamic Programming: DP)

DP は概念が広すぎて説明が困難だが，ここでは式 13 を解くための逆行列演算などを繰り返し処理によって数値的に計算する方法だと考えればよい [1] [3]．特に行列がまばら (sparse) だと効率が良い．

●【例１】 式 13の $V^\pi = (I-\gamma P^\pi)^{-1}R^\pi$ を DP で計算する場合：

1. V^πの全要素を適当な値 (ゼロなど) に初期化 (Xとおく)
2. V^πに $R^\pi + \gamma P^\pi V^\pi$の値を代入
3. 処理手順 2 を収束するまで繰り返す

【解説】上記の処理は，初期化直後は $V^\pi = X$だが，手順 2 を 1 回実行すると，$V^\pi = R^\pi + \gamma P^\pi X$，
2 回実行すると $V^\pi = R^\pi + \gamma P^\pi R^\pi + \gamma^2 (P^\pi)^2 X$，
n 回実行すると $V^\pi = R^\pi + \gamma P^\pi R^\pi + \gamma^2 (P^\pi)^2 R^\pi + \cdots + \gamma^{n-1}(P^\pi)^{n-1}R^\pi + \gamma^n (P^\pi)^n X$となる．
式 13より $V^\pi = R^\pi + \gamma P^\pi R^\pi + \gamma^2 (P^\pi)^2 R^\pi + \cdots$なので，処理 2 の繰返しは高次の項まで近似していくのと等価．

レポート 課題

右図のように，2 台の機械 M1，M2 より構成される生産ラインがある．
●各機械は「稼働」と「休止」の２状態．よってライン全体で４状態
◎機械M２は，M１が休止していたら次のステップでは稼働できない
●機械M２が稼働しているときは常に 1.0 の報酬を得る．それ以外の場合，報酬は 0.
●若い作業員と年配の作業員がそれぞれ機械を操作
●若い作業員が稼働中の機械を操作すると，◎の場合を除き確率 0.8 で稼働状態を継続
●若い作業員が休止中の機械を操作すると，◎の場合を除き確率 0.5 で稼働状態へ
●年配の作業員が稼働中の機械を操作すると，◎の場合を除き確率 0.9 で稼働状態を継続
●年配の作業員が休止中の機械を操作すると，◎の場合を除き確率 0.4 で稼働状態へ
●M１休止かつM２休止の状態を s_1，M１休止かつM２稼働の状態を s_2，
M１稼働かつM２休止の状態を s_3，M１稼働かつM２稼働の状態を s_4とする
●若い作業員をM１へ，年配の作業員をM２へ配置する行動を a_1
年配の作業員をM１へ，若い作業員をM２へ配置する行動を a_2とする

【問１】全ての状態で行動 a_1をとる政策をπ_{1111}，全ての状態で行動 a_2をとる政策をπ_{2222}とする．
このとき $P^{\pi_{1111}}$，$P^{\pi_{2222}}$，$V^{\pi_{1111}}$，$V^{\pi_{2222}}$を求めよ．ただし割引率$\gamma = 0.9$ とする．

【問２】割引率$\gamma = 0.9$ のときの最適な政策π^*および最適 valueV^*を求めよ．

レポート提出期限：平成 12 年 9 月 21 日 (木) 午後 5 時 必着
レポート提出先　：学務部教務掛 (西８Ｅ号館 101 号室)
レポートは A4 判で作成すること
１枚目は表紙として以下の事項を明記すること
・人工知能基礎 (O)
・レポート (９月１日出題分)
・学籍番号
・氏名

参考文献

[1] Barto, A. G., Bradtke, S. J. and Singh, S. P.: Learning to act using real-time dynamic programming, *Artificial Intelligence 72 (1995)*, 81-138.

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

[3] 小笠原正巳，坂本武司 著，北川 敏男 編: 情報科学講座 (全 62 巻)A・5・1　マルコフ過程，共立出版 (1967).

[4] Sutton, R. S. & Barto, A.: Reinforcement Learning: An Introduction, *A Bradford Book*, The MIT Press (1998).

本演習問題に関する質問等はメールにて担当の木村へ行うこと
gen@fe.dis.titech.ac.jp
http://www.fe.dis.titech.ac.jp/ gen/indexj.html