九州大学 工学部地球環境工学科
船舶海洋システム工学コース

システム設計工学 （担当：木村）

（14） マルコフ決定過程２

場所：船１講義室

http://sysplan.nams.kyushu-u.ac.jp/gen/index.html

MDPの割引報酬評価における最適性原理

「最適な政策」とは、一連の意思決定において、
各状態で常に最適な行動を選択すること
→　局所的な最適化の集まりによって全体が最適化、すなわち

最適な価値関数 $\mathbf{V}^* = \begin{bmatrix} V^*(s_1) \\ V^*(s_2) \\ \vdots \\ V^*(s_n) \end{bmatrix}$ とおくと、全ての状態 s について

以下のn個の非線形連立方程式が成り立つ

Bellman方程式

$$V^*(s) = \max_{a \in A} \sum_{s' \in S} P(s'|s,a)\left(R(s'|s,a) + \gamma V^*(s')\right)$$

割引率

状態遷移確率　　直接報酬　　遷移先での状態価値

MDPの割引報酬評価では、最適政策π*は複数存在できるが
最適価値関数V*はただ一つである

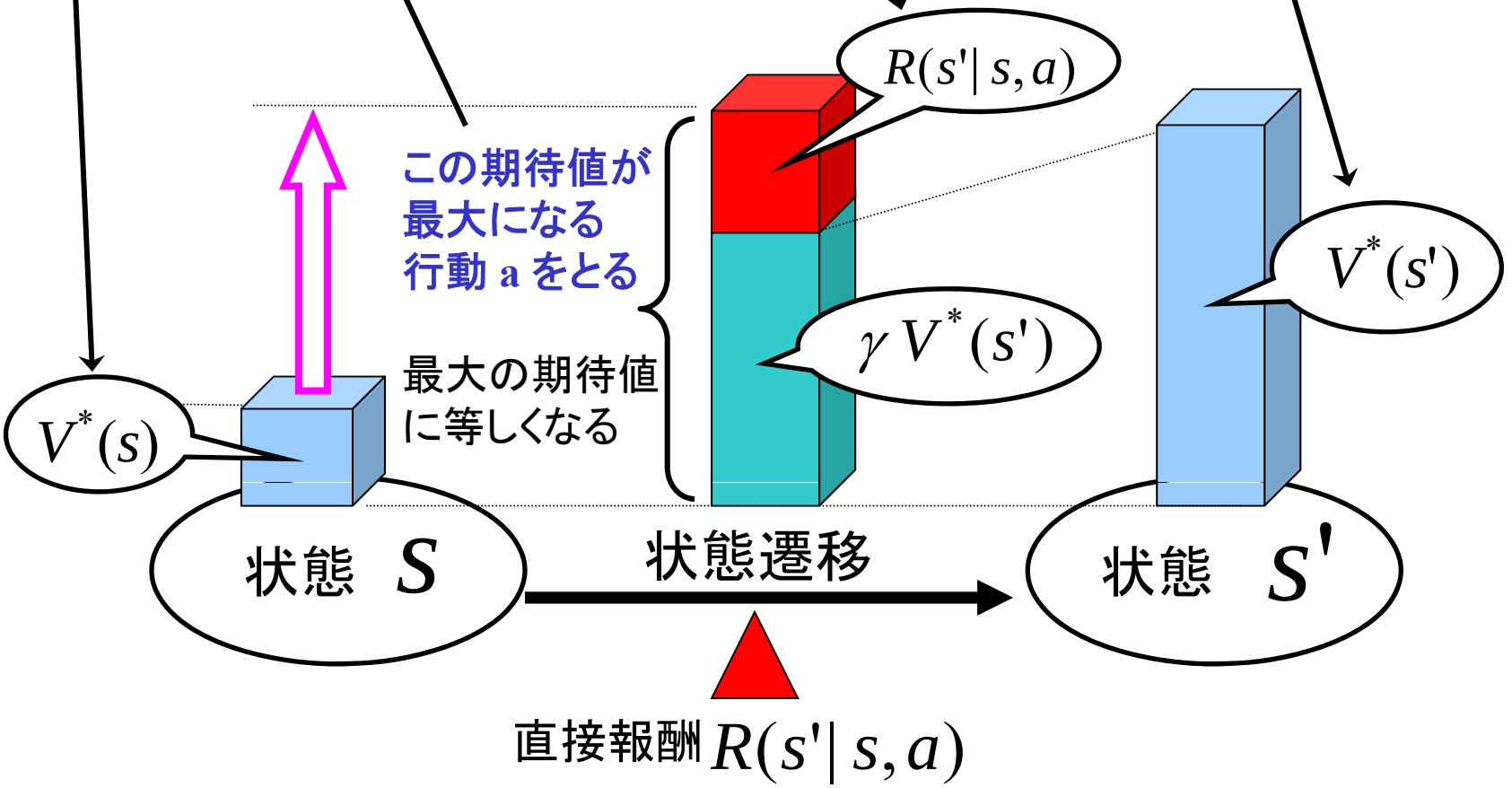
よって、この方程式を解いて最適価値関数を求めてから最適政策を得ればよい

Bellman最適方程式の意味は？

$$V^*(s) = \max_{a \in A} \sum_{s' \in S} P(s'|s,a)\left(R(s'|s,a) + \gamma V^*(s')\right)$$

割引率

状態遷移確率　直接報酬　遷移先での最適状態価値



ダイナミックプログラミングによるMDPの解法：

Bellman方程式は**非線形**のため、**単純な行列演算では解けない**

解法1

Howard の政策反復法（Policy Iteration Method）

- **「政策の改善」と「価値関数の計算」を交互に繰返す**
- 政策の改善ができなくなると、それが最適政策
  そのときの価値関数が最適価値関数

解法2

価値反復法（Value Iteration Method）

- 非線形なBellman方程式を
  反復法（数値計算）によって強引に解く
- 同じ状態で価値関数が最大となる行動が最適行動
  全状態で最適行動をとる政策が最適政策

ダイナミックプログラミングによるMDPの解法：

Bellman方程式は非線形のため、単純な行列演算では解けない

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

政策改善の基本原理

ある定常政策πをとると遷移行列　$\mathbf{P}^{\pi}$と報酬行列　$\mathbf{R}^{\pi}$が決まり、
状態評価値 $\mathbf{V}^{\pi} = \left(\mathbf{I} - \gamma \mathbf{P}^{\pi}\right)^{-1} \mathbf{R}^{\pi}$ が計算できる

ある状態 s のみにおいてπとは別の政策π'に従って行動をとるとき

$$R^{\pi'}(s) + \gamma \sum_{s' \in S} P^{\pi'}(s' \mid s) V^{\pi}(s') \quad > \quad V^{\pi}(s)$$

現在の政策πの
評価値で代用

ならば、その状態 s だけ政策π'に従う新しい政策の状態評価値がもとの政策πより改善されることが証明されている。

→　この原理を利用して政策を改善していく

ダイナミックプログラミングによるMDPの解法：
Howard の政策反復法（Policy Iteration Method）

処理手順：

（1）ある確定的な（確率的でない）定常政策πについて、
以下の方程式 $\forall s \in S$

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

を解き、状態評価値 $\mathbf{V}^{\pi}$ を計算する。

（2）全状態において $R^{\pi'}(s) + \gamma \sum_{s' \in S} P^{\pi'}(s' | s) V^{\pi}(s')$

が最大になるよう政策πをπ’へ変更する

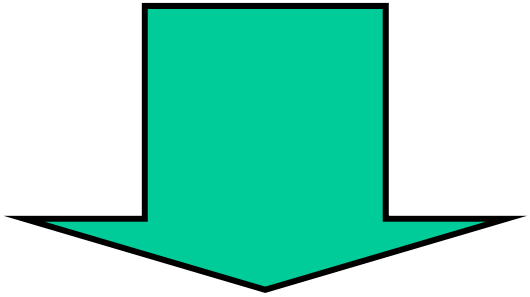
現在の政策πの評価値で代用

（3）π’=π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π←π’として手順（1）より繰り返す。

Howardの政策反復法は「厳密解法」:

1） 最適政策を1つだけ見つける
2） 状態評価値を厳密に計算できる

だけど…

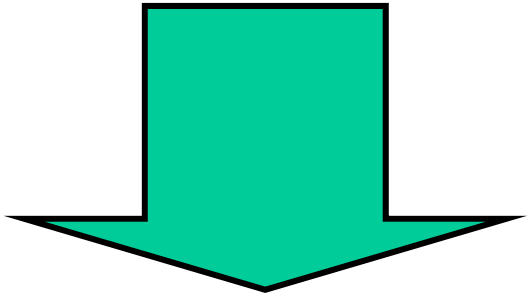


・最適政策さえ見つかれば良い
・状態評価値は**数値計算**でそれなりの精度があれば良い
・もっと簡便な処理手順は無いの？

Howardの政策反復法は「厳密解法」:

1） 最適政策を1つだけ見つける
2） 状態評価値を厳密に計算できる

だけど…



・最適政策さえ見つかれば良い
・状態評価値は**数値計算**でそれなりの精度があれば良い
・もっと簡便な処理手順は無いの？

Bellman方程式の反復解法:
価値反復法（Value iteration method）

Bellman方程式はMax演算があるため逆行列計算で解けないが、
[　　] を用いると簡単に**数値計算で**解ける

Bellman方程式の [　　]

1） 全ての状態 s について V*(s) を適当な値（ゼロなど）に初期化

2） 全ての V*(s) について以下の式を計算して値を代入

$$V^*(s) = \max_{a \in A} \sum_{s' \in S} P(s'|s,a)\left(R(s'|s,a) + \gamma V^*(s')\right)$$

新しい価値関数

遷移先での状態価値関数
古い関数の値を使う

等式として見ればBellman方程式そのもの

3） 処理手順（2）を関数の値が収束するまで繰り返す

これもダイナミックプログラミングの一種

Bellman方程式はMax演算があるため逆行列計算で解けないが、
反復解法 を用いると簡単に**数値計算で**解ける

Bellman方程式の **反復解法：価値反復法（Value iteration method）**

1） 全ての状態 s について V*(s) を適当な値（ゼロなど）に初期化

2） 全ての V*(s) について以下の式を計算して値を代入

$$V^*(s) = \max_{a \in A} \sum_{s' \in S} P(s'|s,a)\left(R(s'|s,a) + \gamma V^*(s')\right)$$

新しい価値関数

遷移先での状態価値関数
古い関数の値を使う

等式として見ればBellman方程式そのもの

3） 処理手順（2）を関数の値が収束するまで繰り返す

これもダイナミックプログラミングの一種

例： 製品製造業者の意思決定問題

$$\begin{cases} V^*(s_1) = \max\Big(\big(0.5\big(9+\gamma V^*(s_1)\big)+0.5\big(3+\gamma V^*(s_2)\big)\big), \\ \qquad\qquad \big(0.8\big(4+\gamma V^*(s_1)\big)+0.2\big(4+\gamma V^*(s_2)\big)\big)\Big) \\ V^*(s_2) = \max\Big(\big(0.4\big(3+\gamma V^*(s_1)\big)+0.6\big(-7+\gamma V^*(s_2)\big)\big), \\ \qquad\qquad \big(0.7\big(1+\gamma V^*(s_1)\big)+0.3\big(-19+\gamma V^*(s_2)\big)\big)\Big) \end{cases}$$

Bellman方程式

← 行動a1をとった場合の状態価値

← 行動a2をとった場合の状態価値

← 行動a1をとった場合の状態価値

← 行動a2をとった場合の状態価値

（1） まず $V^*(s_1)$ と $V^*(s_2)$ の値をゼロとして計算してみる

$$\begin{cases} V^*(s_1) = \max\Big(\big(0.5\big(9+\gamma\times 0\big)+0.5\big(3+\gamma\times 0\big)\big), \quad \big(0.8\big(4+\gamma\times 0\big)+0.2\big(4+\gamma\times 0\big)\big)\Big) \\ \qquad\quad = \max\big(6,4\big) = 6 \\ V^*(s_2) = \max\Big(\big(0.4\big(3+\gamma\times 0\big)+0.6\big(-7+\gamma\times 0\big)\big), \quad \big(0.7\big(1+\gamma\times 0\big)+0.3\big(-19+\gamma\times 0\big)\big)\Big) \\ \qquad\quad = \max\big(-3,-5\big) = -3 \end{cases}$$

（2） 上の計算結果 $V^*(s_1)=6$ $V^*(s_2)=-3$ の値を代入して再計算（γ=0.8）

$$\begin{cases} V^*(s_1) = \max\Big(\big(0.5\big(9+\gamma\times 6\big)+0.5\big(3+\gamma\times(-3)\big)\big), \quad \big(0.8\big(4+\gamma\times 6\big)+0.2\big(4+\gamma\times(-3)\big)\big)\Big) \\ \qquad\quad = \max\big(6+1.5\gamma, 4+4.2\gamma\big) = 7.36 \\ V^*(s_2) = \max\Big(\big(0.4\big(3+\gamma\times 6\big)+0.6\big(-7+\gamma\times(-3)\big)\big), \quad \big(0.7\big(1+\gamma\times 6\big)+0.3\big(-19+\gamma\times(-3)\big)\big)\Big) \\ \qquad\quad = \max\big(-3+0.6\gamma, -5+3.3\gamma\big) = -2.36 \end{cases}$$

（3） 先の計算結果 $V^*(s_1)=7.36$ $V^*(s_2)=-2.36$ の値を代入して再計算（γ=0.8）

$$\begin{cases} V^*(s_1)=\max\left(\left(0.5\left(9+\gamma\times 7.36\right)+0.5\left(3+\gamma\times(-2.36)\right)\right),\ \left(0.8\left(4+\gamma\times 7.36\right)+0.2\left(4+\gamma\times(-2.36)\right)\right)\right) \\ \qquad =\max\left(6+2.5\gamma,\ 4+5.416\gamma\right)=8.3328 \\ V^*(s_2)=\max\left(\left(0.4\left(3+\gamma\times 7.36\right)+0.6\left(-7+\gamma\times(-2.36)\right)\right),\ \left(0.7\left(1+\gamma\times 7.36\right)+0.3\left(-19+\gamma\times(-2.36)\right)\right)\right) \\ \qquad =\max\left(-3+1.528\gamma,\ -5+4.444\gamma\right)=-1.4448 \end{cases}$$

（以後繰り返し）

$$\begin{cases} V^*(s_1)=\max\left(\left(0.5\left(9+\gamma\times 12.17\right)+0.5\left(3+\gamma\times(-2.391)\right)\right),\ \left(0.8\left(4+\gamma\times 12.17\right)+0.2\left(4+\gamma\times(-2.391)\right)\right)\right) \\ \qquad =12.17 \\ V^*(s_2)=\max\left(\left(0.4\left(3+\gamma\times 12.17\right)+0.6\left(-7+\gamma\times(-2.391)\right)\right),\ \left(0.7\left(1+\gamma\times 12.17\right)+0.3\left(-19+\gamma\times(-2.391)\right)\right)\right) \\ \qquad =-2.391 \end{cases}$$

ある値へ収束

このときS1で行動a2、S2で行動a2 をとる政策が最適

割引報酬を最大化する政策は、

Howardの政策反復法や価値反復法で効率良く見つかる。

では、平均報酬を最大化する政策を効率良く探すには？
以下の2通りの方法がある

1） 割引率γを1に近づけていくと、ある割引率の値以上では全て同じ最適政策を共有する。
この最適政策は、平均報酬に関する最適政策でもある

→ 割引率 γ を変えて政策反復法や価値反復法で最適政策を何度も計算

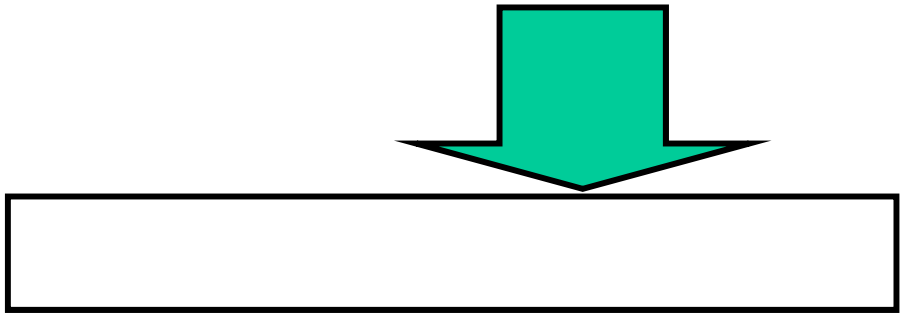
2） 平均報酬に関する最適方程式を利用して
Howardの政策反復法を用いる。

Bellman方程式の反復解法の特徴

- [　　] **が不要** → 状態数が多い場合でも計算できる
  ただしその代償として繰り返し計算を要する

- 古い価値関数値と新しい価値関数値との誤差（Bellman残差）の大きさで計算を打ち切るかどうかを判断

- max計算をする部分で、最大値となる行動 a を記録しておく必要
  → 最適価値関数V*だけでは最適政策（行動）は分かりにくい



[　　] の導入へ

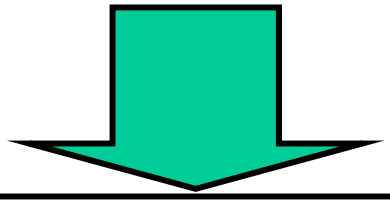
Q*(s,a) ：状態 s で行動 a を選択後は、ずっと最適政策を取り続ける場合の報酬合計の期待値（ただし割引率γで割引く）

Bellman方程式の反復解法の特徴

- **逆行列計算　が不要**　→　状態数が多い場合でも計算できる
  ただしその代償として繰り返し計算を要する

- 古い価値関数値と新しい価値関数値との誤差（Bellman残差）
  の大きさで計算を打ち切るかどうかを判断

- max計算をする部分で、最大値となる行動 a を記録しておく必要
  →　最適価値関数V*だけでは最適政策（行動）は分かりにくい



最適行動価値関数（Q関数）　の導入へ

Q*(s,a)　：状態 s で行動 a を選択後は、ずっと最適政策を取り続ける
場合の報酬合計の期待値（ただし割引率γで割引く）

Q関数を用いたBellman方程式

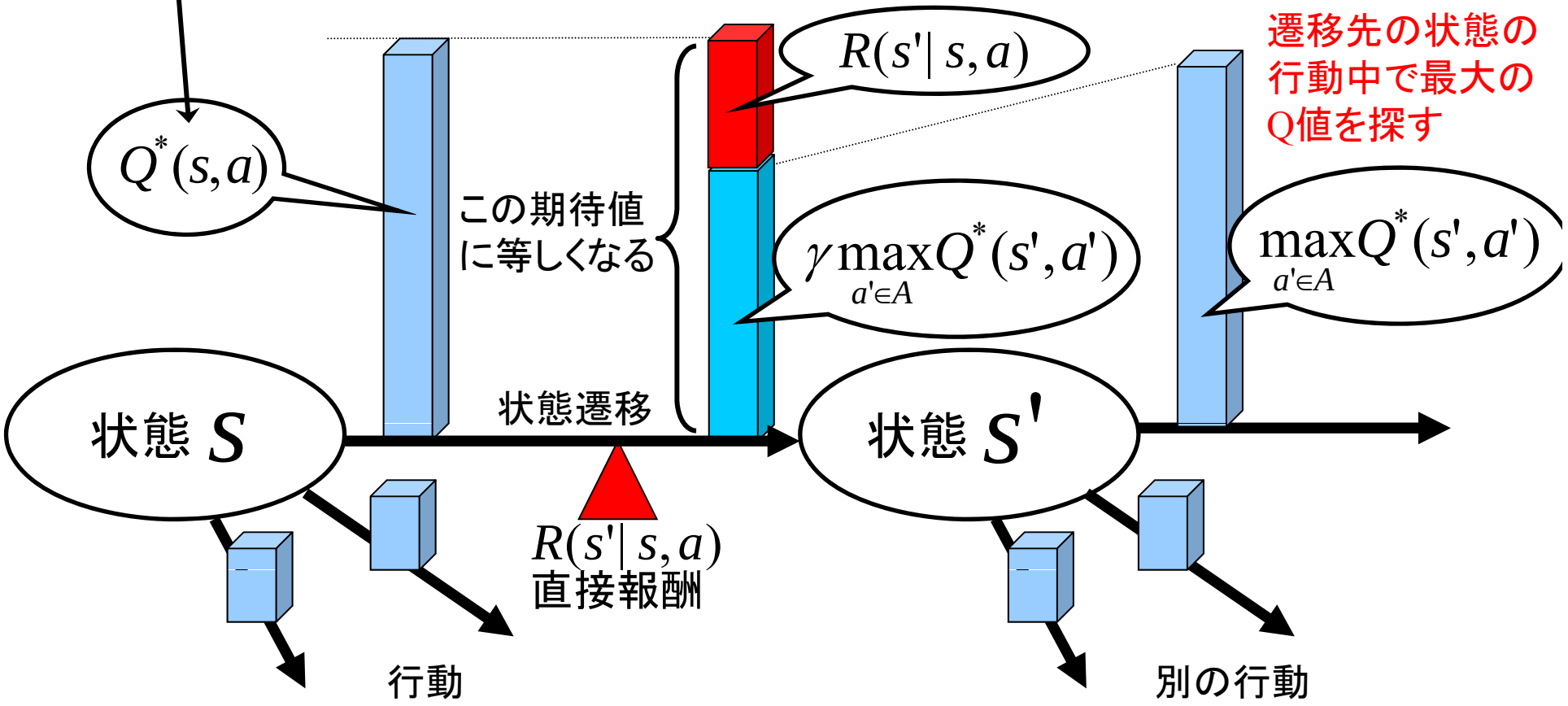
全ての状態 s について以下のn個の
非線形連立方程式が成立

$$V^*(s) = \max_{a \in A} Q^*(s,a)$$

$$Q^*(s,a) = \sum_{s' \in S} P(s'|s,a)\left(R(s'|s,a) + \gamma \max_{a' \in A} Q^*(s',a')\right)$$

直接報酬　割引率　遷移先での状態-行動価値

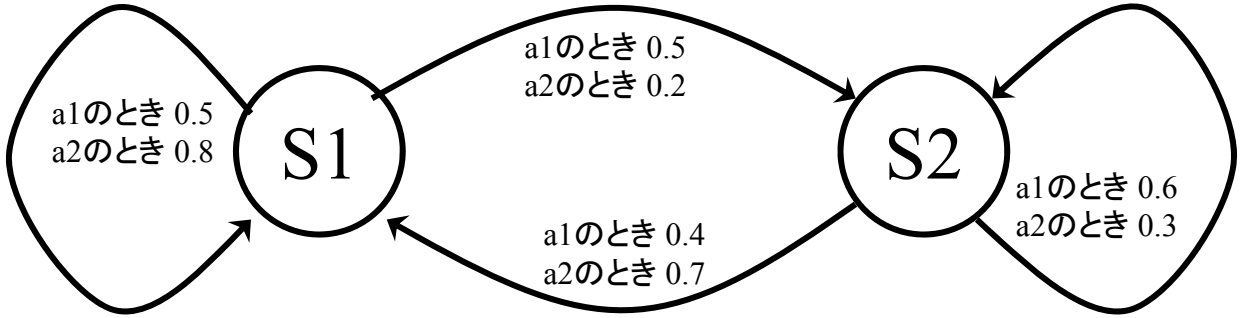
状態遷移確率



例題： 製品製造業者の意思決定問題

ある製品の製造業者が、市場における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 S1：市場において製品の人気がある
- 状態 S2：市場において製品の人気がない
- 状態 S1 における行動 a1 ：広告をしない、行動 a2：広告を出す
- 状態 S2 における行動 a2：何もしない、行動 a2：新製品を開発
- 報酬：各期の売り上げ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	$-5 = 0.7 \times 1 + 0.3 \times (-19)$

製品製造業者の例題の補足説明:

全ての状態で行動a1をとる場合の状態遷移行列と報酬行列

$$\mathbf{P}^{a1} = \begin{bmatrix} P(S_1 \mid S_1, a_1) & P(S_2 \mid S_1, a_1) \\ P(S_1 \mid S_2, a_1) & P(S_2 \mid S_2, a_1) \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \\ 0.4 & 0.6 \end{bmatrix}$$

$$R(S_1 \mid S_1, a_1) = 9, \quad R(S_2 \mid S_1, a_1) = 3$$
$$R(S_1 \mid S_2, a_1) = 3 \quad R(S_2 \mid S_2, a_1) = -7$$

全ての状態で行動a2をとる場合の状態遷移行列と報酬行列

$$\mathbf{P}^{a2} = \begin{bmatrix} P(S_1 \mid S_1, a_2) & P(S_2 \mid S_1, a_2) \\ P(S_1 \mid S_2, a_2) & P(S_2 \mid S_2, a_2) \end{bmatrix} = \begin{bmatrix} 0.8 & 0.2 \\ 0.7 & 0.3 \end{bmatrix}$$

$$R(S_1 \mid S_1, a_2) = 4, \quad R(S_2 \mid S_1, a_2) = 4$$
$$R(S_1 \mid S_2, a_2) = 1 \quad R(S_2 \mid S_2, a_2) = -19$$

製品製造業者の例題のBellman方程式

$$Q^*(s,a) = \sum_{s' \in S} P(s'|s,a)\left(R(s'|s,a) + \gamma \max_{a' \in A} Q^*(s',a')\right)$$

$$Q^*(s_1,a_1) = P(s_1|s_1,a_1)\left(R(s_1|s_1,a_1) + \gamma \max_{a' \in A} Q^*(s_1,a')\right)$$
$$+ P(s_2|s_1,a_1)\left(R(s_2|s_1,a_1) + \gamma \max_{a' \in A} Q^*(s_2,a')\right)$$

S_1へ遷移

S_2へ遷移

$$= \square\left(\square + \gamma \max_{a' \in A} Q^*(s_1,a')\right)$$
$$+ \square\left(\square + \gamma \max_{a' \in A} Q^*(s_2,a')\right)$$

$Q^*(S_1,a_1)$ と $Q^*(S_1,A_2)$ のうちの大きい方

$Q^*(S_2,a_1)$ と $Q^*(S_2,A_2)$ のうちの大きい方

製品製造業者の例題のBellman方程式

$$Q^*(s,a) = \sum_{s' \in S} P(s'|s,a)\left(R(s'|s,a) + \gamma \max_{a' \in A} Q^*(s',a')\right)$$

$$Q^*(s_1,a_1) = P(s_1|s_1,a_1)\left(R(s_1|s_1,a_1) + \gamma \max_{a' \in A} Q^*(s_1,a')\right)$$
$$+ P(s_2|s_1,a_1)\left(R(s_2|s_1,a_1) + \gamma \max_{a' \in A} Q^*(s_2,a')\right)$$

S_1へ遷移

S_2へ遷移

$$= 0.5\left(9 + \gamma \max_{a' \in A} Q^*(s_1,a')\right)$$
$$+ 0.5\left(4 + \gamma \max_{a' \in A} Q^*(s_2,a')\right)$$

$Q^*(S_1,a_1)$ と $Q^*(S_1,A_2)$ のうちの大きい方

$Q^*(S_2,a_1)$ と $Q^*(S_2,A_2)$ のうちの大きい方

$$Q^*(s_1,a_1) = P(s_1 \mid s_1,a_1)\left(R(s_1 \mid s_1,a_1) + \gamma \max_{a' \in A} Q^*(s_1,a')\right)$$

S_1へ遷移

$$+ P(s_2 \mid s_1,a_1)\left(R(s_2 \mid s_1,a_1) + \gamma \max_{a' \in A} Q^*(s_2,a')\right)$$

S_2へ遷移

$$= \square \left(\square + \gamma \max_{a' \in A} Q^*(s_1,a')\right)$$

$$+ \square \left(\square + \gamma \max_{a' \in A} Q^*(s_2,a')\right)$$

$$Q^*(s_1,a_2) = P(s_1 \mid s_1,a_2)\left(R(s_1 \mid s_1,a_2) + \gamma \max_{a' \in A} Q^*(s_1,a')\right)$$

S_1へ遷移

$$+ P(s_2 \mid s_1,a_2)\left(R(s_2 \mid s_1,a_2) + \gamma \max_{a' \in A} Q^*(s_2,a')\right)$$

S_2へ遷移

$$= \square \left(\square + \gamma \max_{a' \in A} Q^*(s_1,a')\right)$$

$$+ \square \left(\square + \gamma \max_{a' \in A} Q^*(s_2,a')\right)$$

$$Q^*(s_1,a_1) = P(s_1 \mid s_1,a_1)\left(R(s_1 \mid s_1,a_1) + \gamma \max_{a' \in A} Q^*(s_1,a')\right)$$

S_1へ遷移

$$+ P(s_2 \mid s_1,a_1)\left(R(s_2 \mid s_1,a_1) + \gamma \max_{a' \in A} Q^*(s_2,a')\right)$$

S_2へ遷移

$$= \boxed{0.5}\left(\boxed{9} + \gamma \max_{a' \in A} Q^*(s_1,a')\right)$$

$$+ \boxed{0.5}\left(\boxed{4} + \gamma \max_{a' \in A} Q^*(s_2,a')\right)$$

$$Q^*(s_1,a_2) = P(s_1 \mid s_1,a_2)\left(R(s_1 \mid s_1,a_2) + \gamma \max_{a' \in A} Q^*(s_1,a')\right)$$

S_1へ遷移

$$+ P(s_2 \mid s_1,a_2)\left(R(s_2 \mid s_1,a_2) + \gamma \max_{a' \in A} Q^*(s_2,a')\right)$$

S_2へ遷移

$$= \boxed{0.8}\left(\boxed{4} + \gamma \max_{a' \in A} Q^*(s_1,a')\right)$$

$$+ \boxed{0.2}\left(\boxed{4} + \gamma \max_{a' \in A} Q^*(s_2,a')\right)$$

$$Q^*(s_2,a_1) = P(s_1 \mid s_2,a_1)\left(R(s_1 \mid s_2,a_1) + \gamma \max_{a' \in A} Q^*(s_1,a')\right)$$

S_1へ遷移

$$+ P(s_2 \mid s_2,a_1)\left(R(s_2 \mid s_2,a_1) + \gamma \max_{a' \in A} Q^*(s_2,a')\right)$$

S_2へ遷移

$$= \square \left(\square + \gamma \max_{a' \in A} Q^*(s_1,a')\right)$$

$$+ \square \left(\square + \gamma \max_{a' \in A} Q^*(s_2,a')\right)$$

$$Q^*(s_2,a_2) = P(s_1 \mid s_2,a_2)\left(R(s_1 \mid s_2,a_2) + \gamma \max_{a' \in A} Q^*(s_1,a')\right)$$

S_1へ遷移

$$+ P(s_2 \mid s_2,a_2)\left(R(s_2 \mid s_2,a_2) + \gamma \max_{a' \in A} Q^*(s_2,a')\right)$$

S_2へ遷移

$$= \square \left(\square + \gamma \max_{a' \in A} Q^*(s_1,a')\right)$$

$$+ \square \left(\square + \gamma \max_{a' \in A} Q^*(s_2,a')\right)$$

$$Q^*(s_2,a_1) = P(s_1 \mid s_2,a_1)\left(R(s_1 \mid s_2,a_1) + \gamma \max_{a' \in A} Q^*(s_1,a')\right)$$

S_1へ遷移

$$+ P(s_2 \mid s_2,a_1)\left(R(s_2 \mid s_2,a_1) + \gamma \max_{a' \in A} Q^*(s_2,a')\right)$$

S_2へ遷移

$$= \boxed{0.4}\left(\boxed{3} + \gamma \max_{a' \in A} Q^*(s_1,a')\right)$$

$$+ \boxed{0.6}\left(\boxed{-7} + \gamma \max_{a' \in A} Q^*(s_2,a')\right)$$

$$Q^*(s_2,a_2) = P(s_1 \mid s_2,a_2)\left(R(s_1 \mid s_2,a_2) + \gamma \max_{a' \in A} Q^*(s_1,a')\right)$$

S_1へ遷移

$$+ P(s_2 \mid s_2,a_2)\left(R(s_2 \mid s_2,a_2) + \gamma \max_{a' \in A} Q^*(s_2,a')\right)$$

S_2へ遷移

$$= \boxed{0.7}\left(\boxed{1} + \gamma \max_{a' \in A} Q^*(s_1,a')\right)$$

$$+ \boxed{0.3}\left(\boxed{-19} + \gamma \max_{a' \in A} Q^*(s_2,a')\right)$$

Q関数を用いたBellman方程式

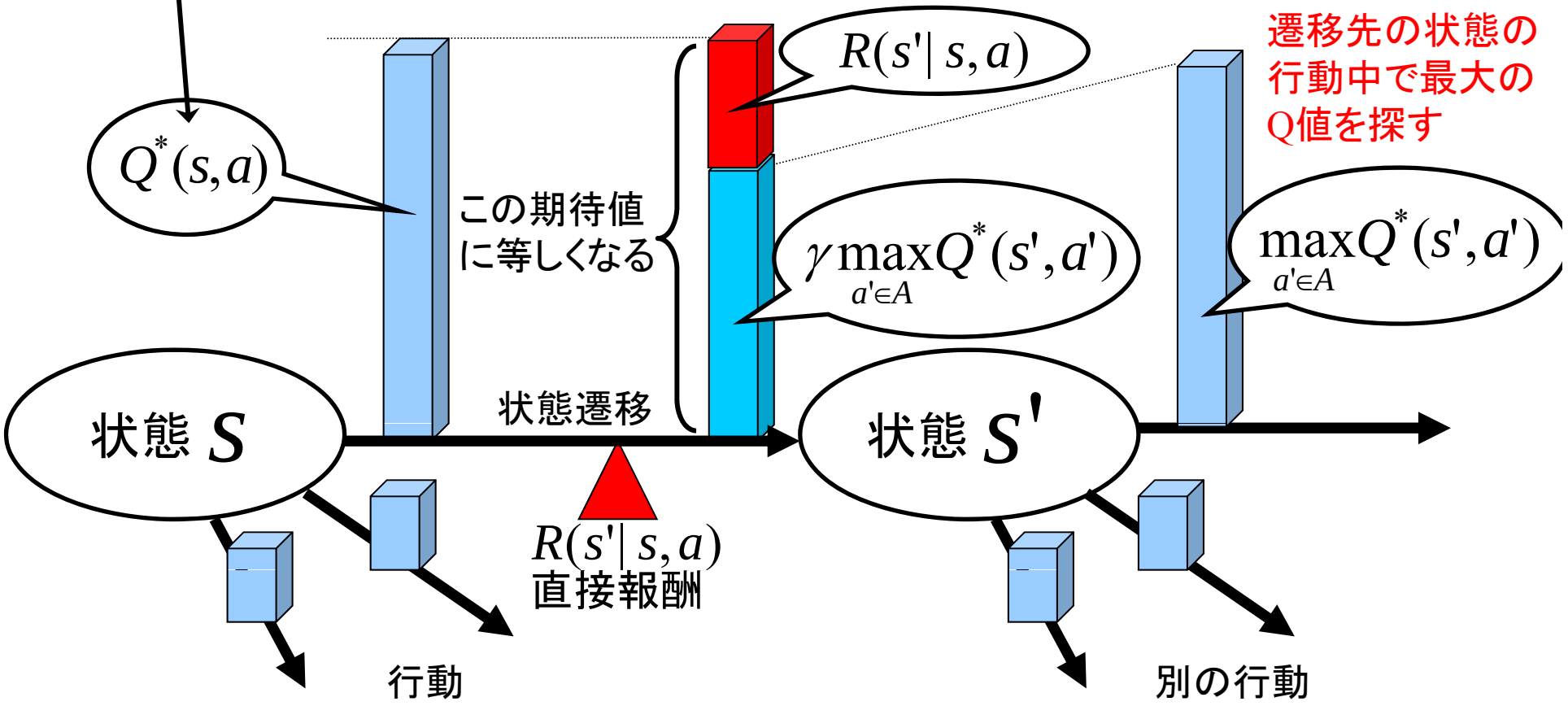
全ての状態 s について以下のn個の
非線形連立方程式が成立

$$V^*(s) = \max_{a \in A} Q^*(s,a)$$

$$Q^*(s,a) = \sum_{s' \in S} P(s'|s,a)\left(R(s'|s,a) + \gamma \max_{a' \in A} Q^*(s',a')\right)$$

直接報酬　割引率　遷移先での状態-行動価値

状態遷移確率



Q値を用いたBellman方程式の［　　］

1） 全状態-行動について Q*(s,a) を適当な値（ゼロなど）に初期化

2） 全ての Q*(s,a) について以下のような式の値を代入

$$Q^*(s,a) = \sum_{s' \in S} P(s'|s,a)\left(R(s'|s,a) + \gamma \max_{a' \in A} Q^*(s',a')\right)$$

新しい価値関数

遷移先での状態価値関数
古い関数の値を使う

3） 処理手順（2）を関数の値が収束するまで繰り返す

○ 各状態 s において、［　　］が
最適な行動として得られる

× 計算に必要な記憶容量が（状態）×（行動）の個数分
必要になり、大きな問題では計算不能に

Q値を用いたBellman方程式の反復解法（Value Iteration Method）:

1） 全状態-行動について Q*(s,a) を適当な値（ゼロなど）に初期化

2） 全ての Q*(s,a) について以下のような式の値を代入

$$Q^*(s,a) = \sum_{s' \in S} P(s'|s,a)\left(R(s'|s,a) + \gamma \max_{a' \in A} Q^*(s',a')\right)$$

新しい価値関数

遷移先での状態価値関数
古い関数の値を使う

3） 処理手順（2）を関数の値が収束するまで繰り返す

○ 各状態 s において、最大のQ*(s,a) を持つ行動 a が最適な行動として得られる

× 計算に必要な記憶容量が（状態）×（行動）の個数分必要になり、大きな問題では計算不能に

プランニング：

状態遷移マトリクスが予め与えられている場合に，
マトリクスの計算により最適value関数や最適政策を求める

強化学習：

状態遷移マトリクスが未知の場合，環境との試行錯誤的な
インタラクションを通じて最適value関数や最適政策を求める
マトリクス計算のモンテカルロ近似

【例】 政策πのもとで状態S1から1ステップ遷移して得る報酬の期待値を求める

・プランニングによる計算：

政策πにおいて状態sで行動aを選ぶ確率

状態遷移確率

状態s1で行動aを選んで状態s'へ遷移した場合の報酬の期待値

$$R^{\pi}(s_1) = \sum_{s'} \sum_{a} \pi(s_1, a) \Pr(s' | s_1, a) R^{a}(s_1, s')$$

・モンテカルロ近似による計算： 状態S1で得る報酬を実際の環境中で観測して平均する

【準備】 平均値を逐次的に計算するには？

ある確率変数 X についての n-1 個の標本 $x_1, x_2, \cdots, x_{n-1}$ があるとき、

確率変数 X の平均値は $\overline{X}_{n-1} = \dfrac{x_1 + x_2 + \cdots + x_{n-1}}{n-1}$

いま、$\overline{X}_{n-1}$ の値が既知の状況において、n 個目の新たな標本 x_n を得たとき、

新たな平均値 $\overline{X}_n$ は次のように計算できる:

$$\overline{X}_n = \left(1 - \frac{1}{n}\right)\overline{X}_{n-1} + \frac{1}{n}x_n$$

Robbins-Monro の
逐次更新アルゴリズム

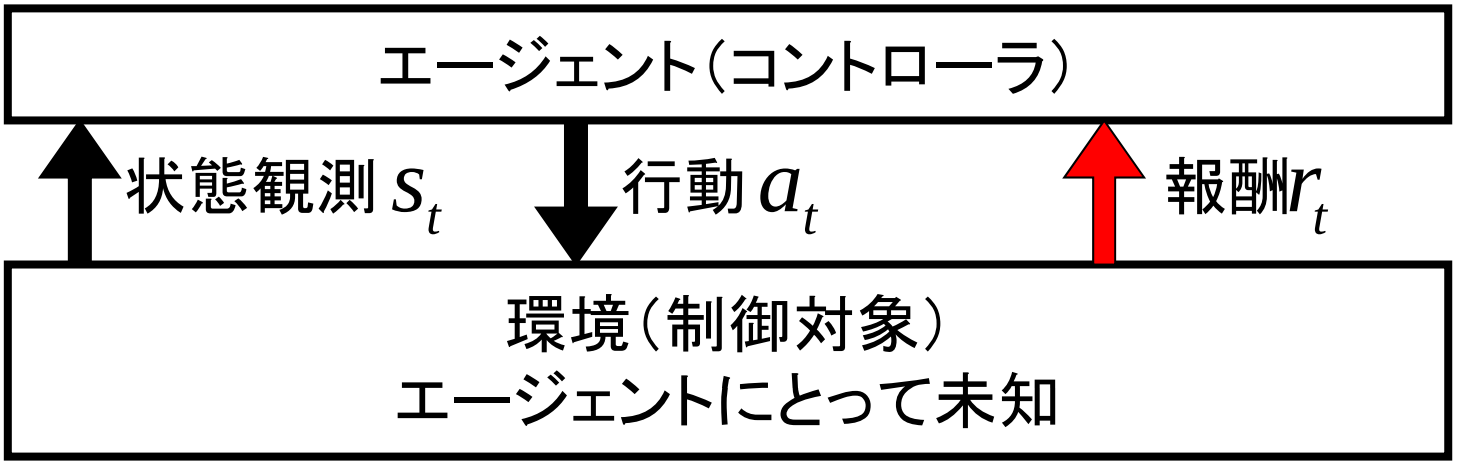
$$= \left(\frac{n-1}{n}\right)\frac{x_1 + x_2 + \cdots + x_{n-1}}{n-1} + \frac{1}{n}x_n$$

$$= \frac{x_1 + x_2 + \cdots + x_{n-1} + x_n}{n}$$

標本 $x_1, x_2, \cdots, x_{n-1}$ が順に
得られるとき、標本の値を記憶せずに
平均値を計算できる

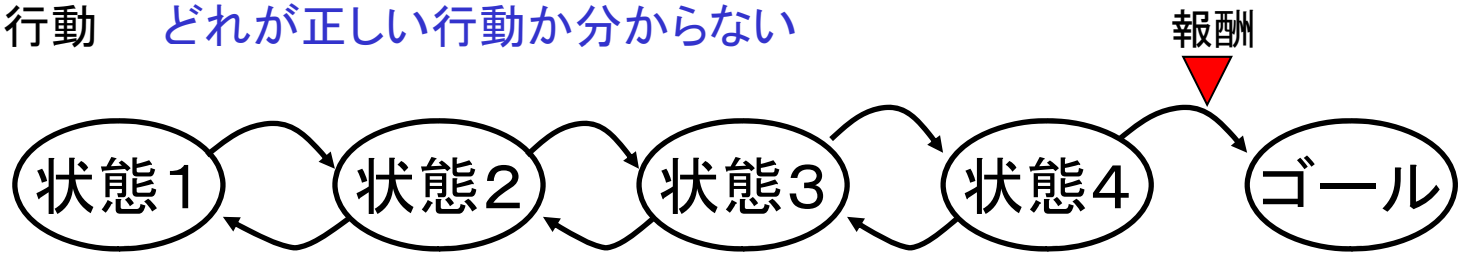
平均値は標本を多数得れば得るほど
精度が高くなる

どのような仕組みで学習するのか？



【例】

下図の環境では 状態4→5 に遷移すると正の報酬
矢印は行動　どれが正しい行動か分からない



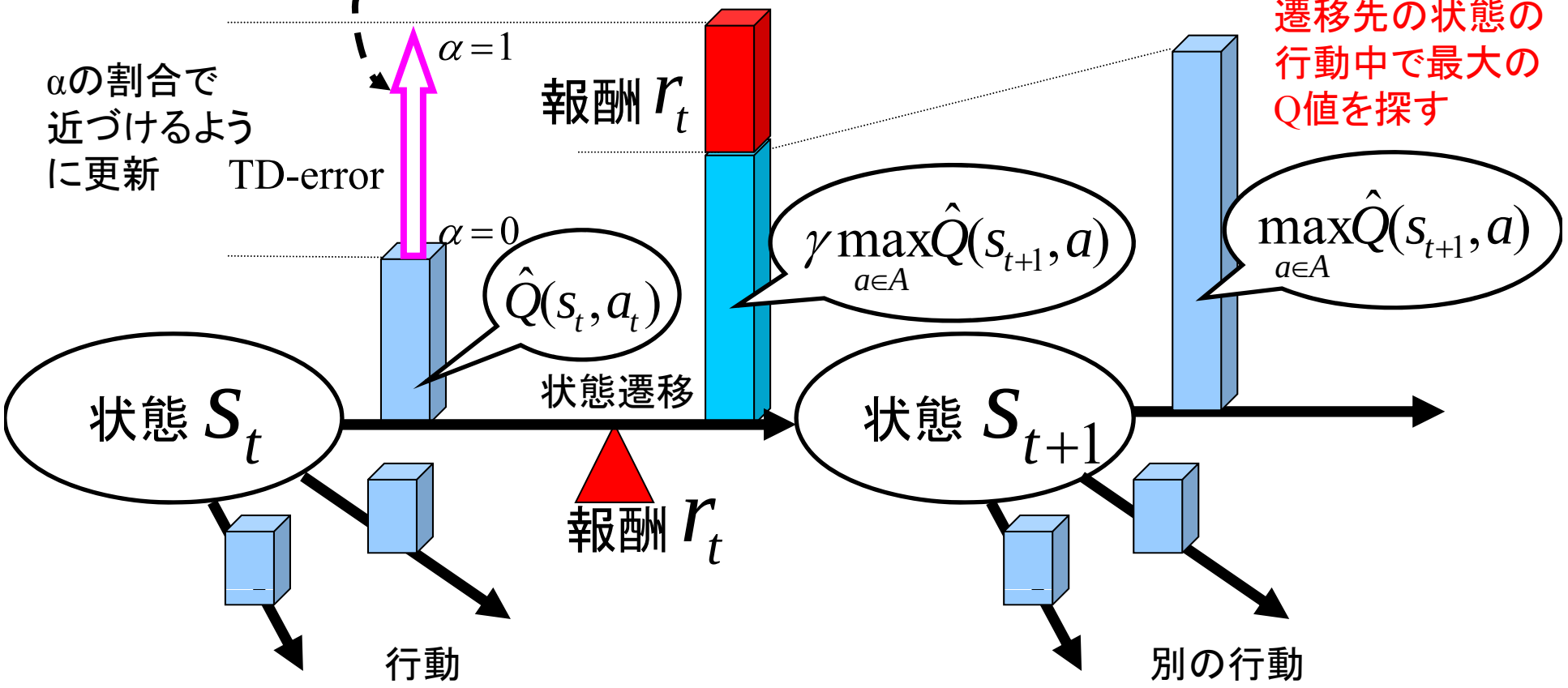
例えばエージェントが状態3に
あるとき、環境を左図のように観測

強化学習アルゴリズムQ-learning:　以下の式で逐次更新

$$\hat{Q}(s_t,a_t) \leftarrow \hat{Q}(s_t,a_t) + \alpha \left[r_t + \gamma \max_{a \in A} \hat{Q}(s_{t+1},a) - \hat{Q}(s_t,a_t) \right]$$

状態Stで実行した
行動のQ値を更新

γは割引率（$0 \leqq \gamma \leqq 1$）
αは学習率（$0 < \alpha \leqq 1$）
＝（1／行動をとった回数）



Q値を用いたBellman方程式の反復解法との比較

1） 全状態-行動について Q*(s,a) を適当な値（ゼロなど）に初期化

2） 全ての Q*(s,a) について以下のような式の値を代入

$$Q^*(s,a) = \sum_{s' \in S} P(s'|s,a)\left(R(s'|s,a) + \gamma \max_{a' \in A} Q^*(s',a')\right)$$

新しい価値関数

遷移確率で重み付け平均する部分
→強化学習では試行を繰返して得る

遷移先での状態価値関数
古い関数の値を使う

3） 処理手順（2）を関数の値が収束するまで繰り返す

各状態 s において、最大のQ*(s,a) を持つ行動 a が
最適な行動として得られる

Q-learningの収束定理（Watkins92）

行動選択において全行動を十分な回数選択し，かつ学習率αが

$$\sum_{t=0}^{\infty} \alpha'(t) \to \infty \qquad \text{and} \qquad \sum_{t=0}^{\infty} \alpha(t)^2 < \infty$$

を満たす時間 t の関数になっているとき，アルゴリズムで得るQ値は確率1で最適政策の評価値に概収束する。
ただし環境はエルゴート性を有するMDP。

Q-learningの行動選択方法

上記の定理は条件を満たせばどんな行動選択方法でも成り立つ。
学習とともになるべく行動を改善していく行動選択が望ましい

1）ε-greedy選択：
　εの確率でランダム，それ以外は最大Q値の行動

2）ボルツマン選択：
exp(Q(s,a)/T)に比例した割合で行動選択
　ただし温度パラメータTは時間とともにT→0へ

その他：OIV　初期Q値大きめに設定→未経験の行動とりやすく
Optimistic Initial Value:「楽天的な初期値」

強化学習の理論的特徴

- ●状態遷移に不確実性を伴う制御問題を理論的に扱う
- ●離散的な状態遷移も含んだ段取り的な制御も理論的に扱う

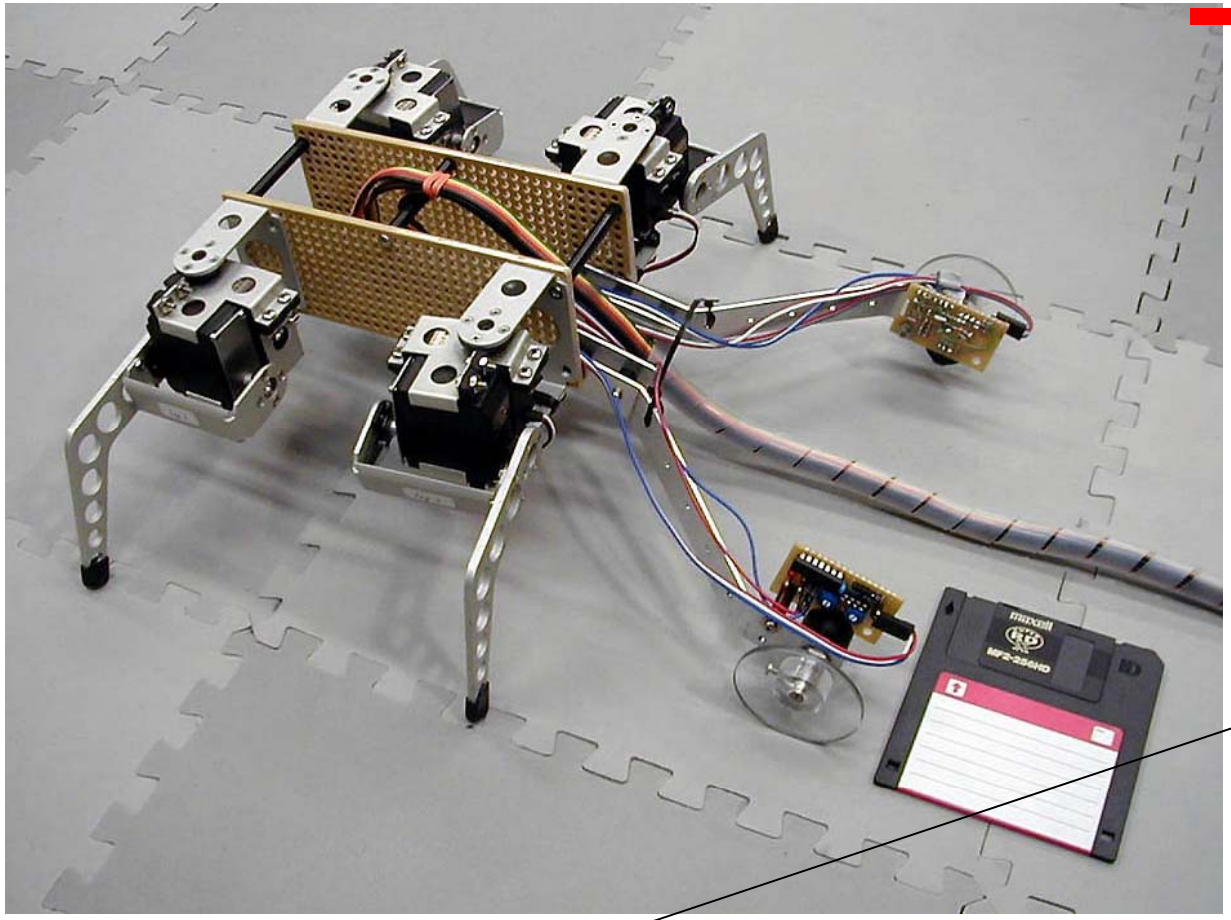
→　環境を確率過程（マルコフ決定過程）でモデル化

応用上の特徴

「何をすべきか」を「報酬」によって簡単に指示するだけで
「どのように実現するか」という制御規則を学習により自動的に獲得

1）制御プログラミングの自動化・省力化
2）ハンドコーディングよりも優れた解：
　特に不確実な要素（摩擦やガタ，振動，誤差など）や計測困難な未知パラメータが多い場合に有利
3）自律性と想定外の環境変化への対応：
　通信が物理的に困難だったり現象のダイナミクスが人間にとって早過ぎる場合や，機械故障など急激な変化やプラント経年変化など予め想定しておくことが困難な環境の変化に対し自動的に追従

強化学習の適用例：ロボットの歩行動作獲得(1)

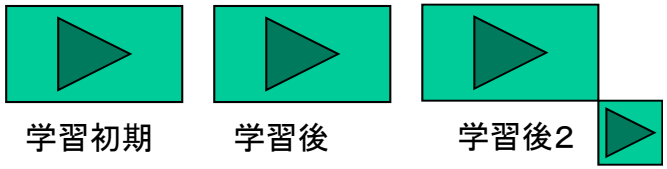


前進するために歩行

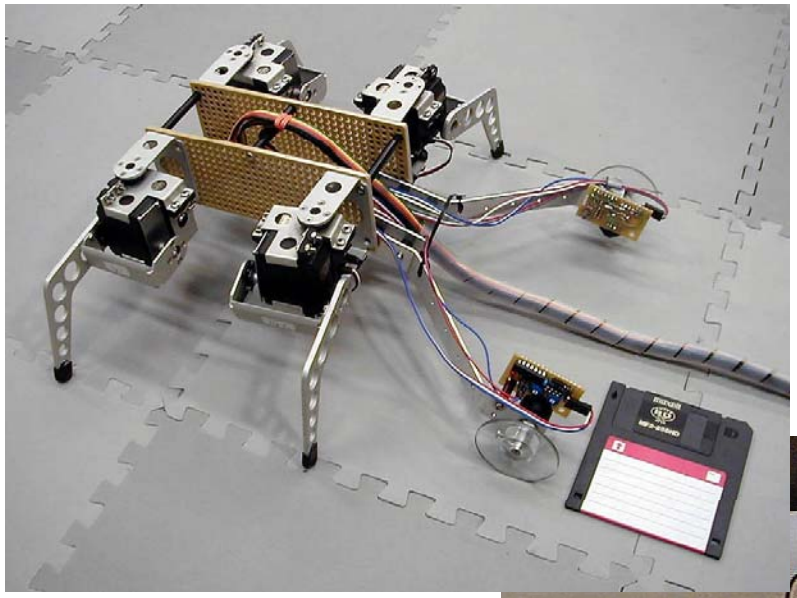
まっすぐ前進したら
最大の報酬を与える
ように設定

最大の報酬を得る
制御則を強化学習で
自動的に獲得

簡単な報酬設定から複雑な制御規則を
自動的に獲得
→　制御プログラミング自動化

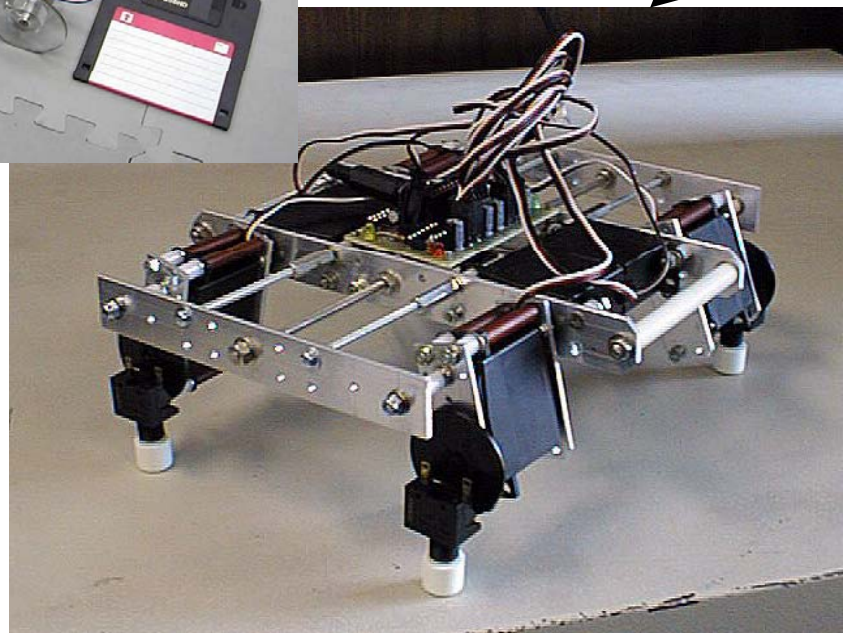


強化学習の適用例：ロボットの歩行動作獲得(2)
同一の学習器で異なるロボットを学習

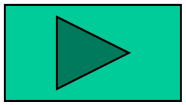
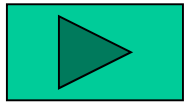


感覚入力：関節の角度（8次元連続値）
行動出力：関節モータの角度（8次元連続値）
報酬：　　毎ステップの移動距離　（移動速度）

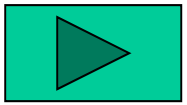
コントローラから見ると
同一の環境に見える

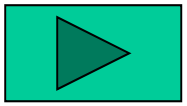


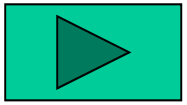
ダイナミクスは異なる

学習初期　　学習途中


学習後1

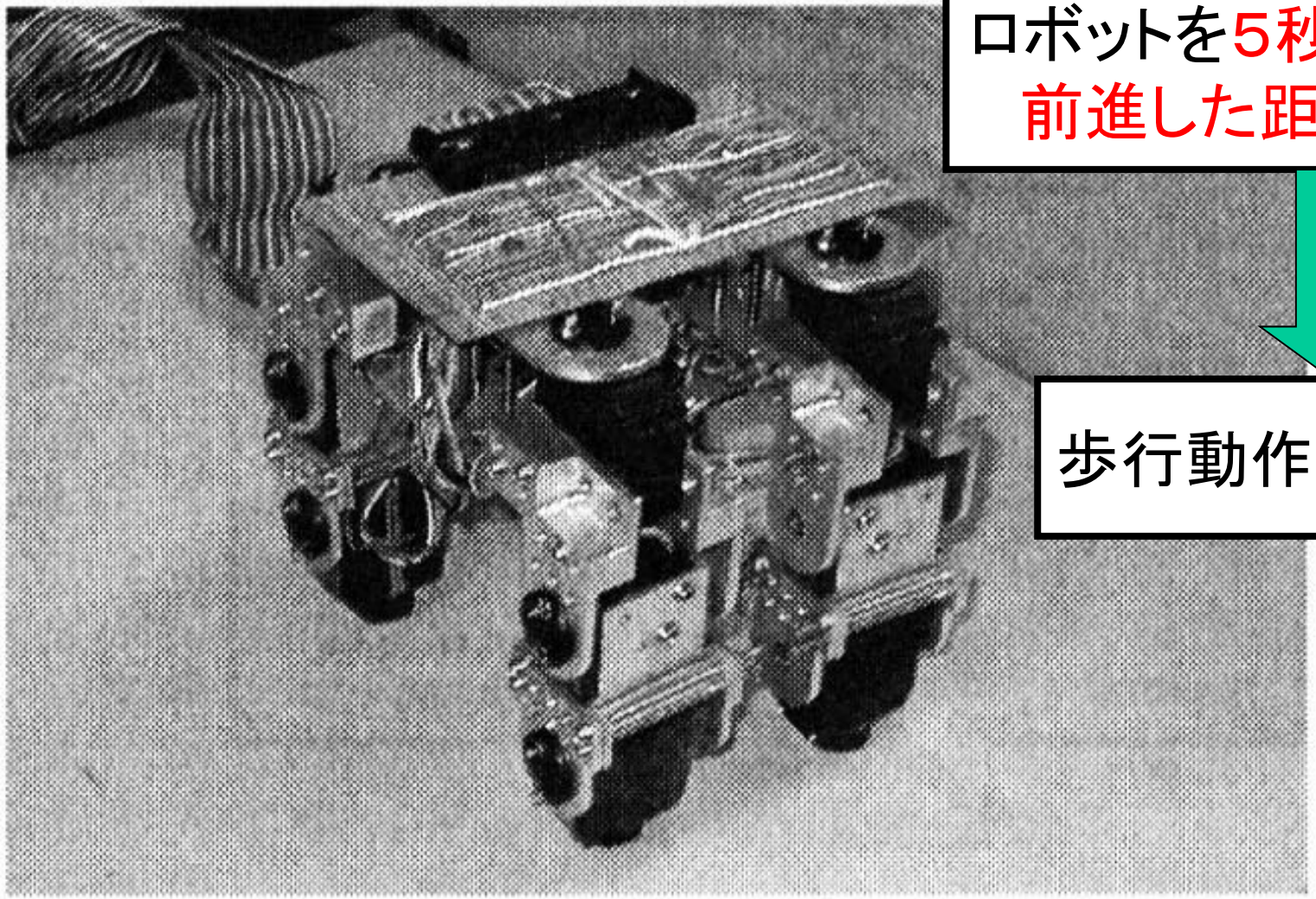

学習後2


学習後3

同一の学習器で、各ロボットに適した
複数の制御プログラムを自動的に獲得

強化学習の失敗例：期待する動作を獲得しない場合

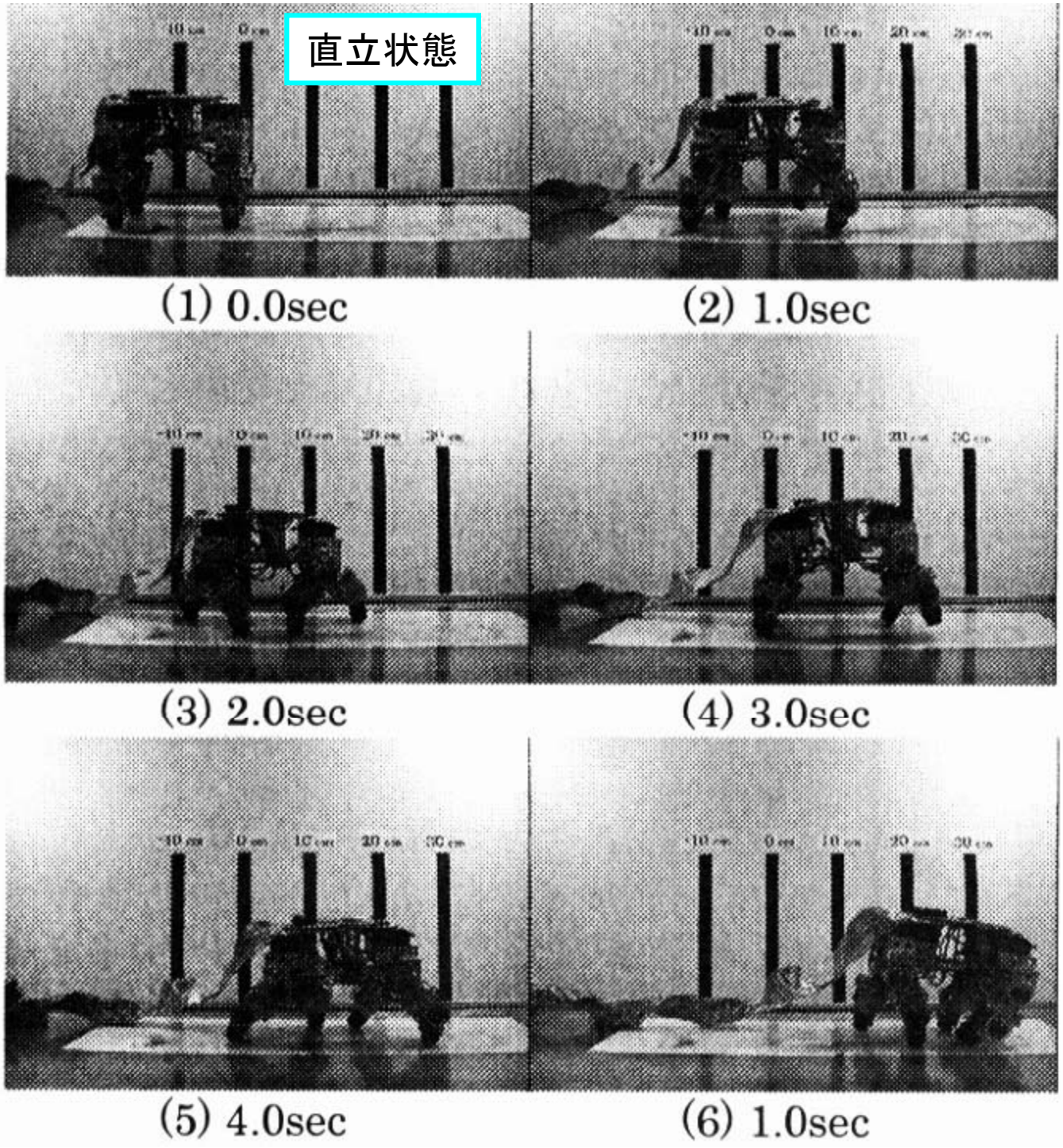
〔神戸大学：森，北村，村尾　2001〕



ロボットを5秒間動かし，
前進した距離を報酬

歩行動作獲得を期待

Figure 1 Real Robot ： 総重量約 1.45kg，
全長約 0.18m，全高約 0.15m



学習結果

最初の数歩は
歩くが…

最後に
ヘッド
スライディング

5秒間の移動距離は
最大になる

ロボット以外の応用例

大規模プランニング問題における
シミュレータベースの探索手法として

- ネットワークルーティング
  (Boyan et al. 1993, Subramanian et al. 1997, Kumer et al. 1999)
- セルラー通信システム（PHS）におけるバンド割当て
  (Singh et al. 1997)
- 生産システム管理（在庫管理） (Wang and Mahadevan 1999)
  トヨタのKANBAN方式よりもコスト削減
- エレベータ群制御 (Crites and Barto 1996)
- 株式売買エージェント，Finance関連 (Neuneier 1998)
- データベースシステムにおけるタイムアウト時間間隔制御
  (後藤，木村，小林 2001)
- 分散型強化学習による上下水道系制御（青木，木村，小林 2002)

「環境」のモデル化

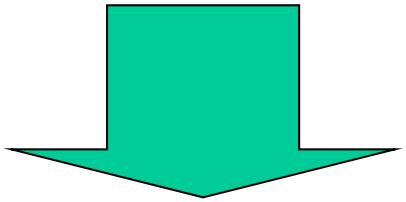
迷路問題の学習で成功した強化学習方法が4足ロボットや
ネットワークルーティングの学習問題で成功するとは限らない

そこで‥

- 迷路問題
- 4足ロボット
- ネットワーク問題
- 在庫管理問題
- Etc..

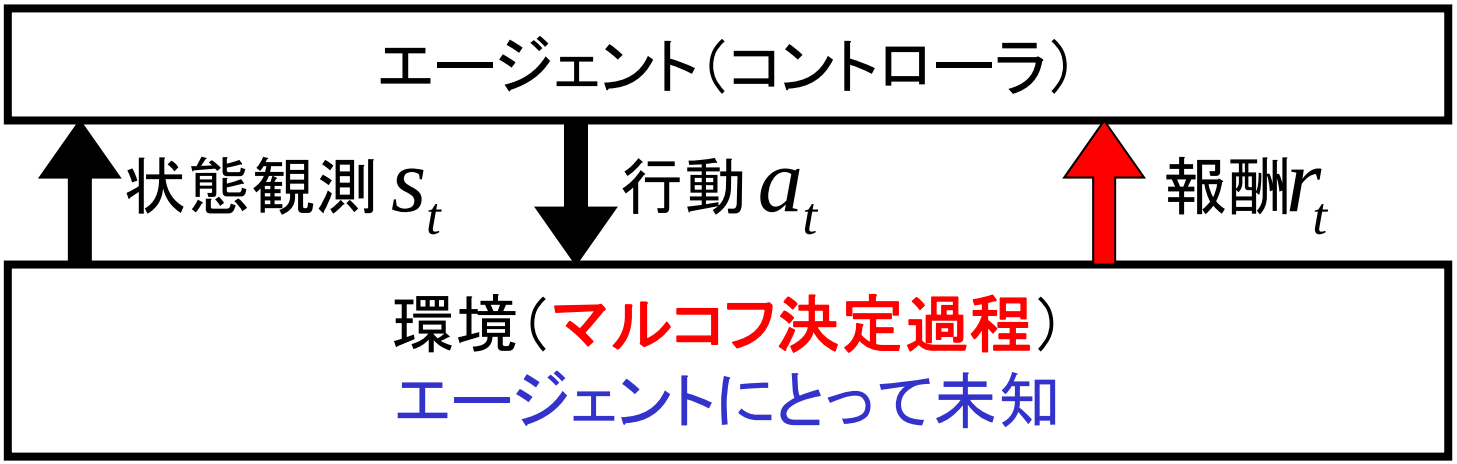
これらに共通する問題の特徴を
マルコフ決定過程という数理モデル
によって表現（モデル化）

マルコフ決定過程において学習することが示されたアルゴリズム



マルコフ決定過程で表現される全ての問題において学習が保証

マルコフ決定過程の強化学習問題定式化



マルコフ決定過程において学習することが示されたアルゴリズム
（Q-learning等）

環境がマルコフ決定過程で表現できることさえ分かれば、
それをどう表現するかが未知であっても
全ての問題において学習が保証

強化学習における先進的話題：

１）セミマルコフ決定過程における学習
　時間間隔が乱数

２）状態空間が連続な場合
３）行動空間が連続な場合
　} 実用上はこれらが最も重要

４）マルチエージェント
　マルコフゲーム

５）階層化

６）状態観測が不完全な場合：隠れマルコフ問題

http://sysplan.nams.kyushu-u.ac.jp/gen/edu/RL_intro.html

【演習問題】

学籍番号＿＿＿＿＿＿＿＿＿＿氏名＿＿＿＿＿＿＿＿＿＿

（1） マルコフ決定過程において割引報酬を最大化するためのアルゴリズムとして政策反復法や価値反復法を用いるが、平均報酬を最大化する政策を求める方法は主に2種類ある。それらの方法を説明せよ。

（2）最適行動価値関数（Q関数）を導入してBellman方程式をたて、価値反復法で解く場合の利点・欠点を説明せよ。