11.Reinforcement Learning
強化学習

60040130
渡辺秀臣

目次

目次

キーワード

エージェント (agent)
環境 (world)
状態価値関数 (State-Value function)
行動価値関数 (Action-Value function)
報酬 (reward)
収益 (return)
方策 (policy)

11.1 紹介

前章までのマルコフモデル
・環境が変化すると、行動のモデル化は不可能

環境変化対応型に拡張

MDP(Markov Decision Process
マルコフ決定過程)

行動についての3要素

1. 行動結果は演繹的にわからないかもしれない
2. 行動価値は実験からわかる
3. 行動価値を求めるには報酬遅延により困難

ポールバランシング

1. モデル（運動方程式）はわからないが
   加える力とその結果を観測する事で学習可能

2. 行動の有用性は試行と観測により学習可能

3. 制御方策の成功と失敗は直ぐにはわからない
   （失敗）ポールが平行か台車がぶつかる
   （成功）ポールが垂直になる

強化学習

特徴

- 遅延報酬を扱う
- 離散的な状態空間（環境モデル）を作る
- ある状態から行動し、その報酬の可能性を反映する各状態の有用性を関連する

有用性（価値）の計算について

ヒューリステック関数との違いは
強化学習では、経験ではなく繰り返し試行により計算できると仮定している

これからのあらすじ

１１．２

強化学習の設計方法と
迷路探索とポールバランシングの例を使い
そのアプローチについて

１１．３

強化学習の数学的基礎と方策改善について

１１．４

Ｑ学習について

１１．５

TD-gammon について

これからのあらすじ

１１．６

学習に教師をつけることで計算時間を
短縮できる事について

（１）批評家 (critic)
エージェントの行為を見て
重要な接点で報酬を補充する

（２）実演者 (demonstrator)
個々のステップに実行する事で
問題解決を説明する

これからのあらすじ

11．7

HMM を扱う

Partially Observable MDP

（部分観測マルコフ決定過程）について

決定が曖昧になる状態空間の困難な部分を避ける事について（例：カラーブロック問題）

行動と報酬の履歴による学習

Nearest Sequence Memory

Utile Distinction Memory

11.2 MDP(マルコフ決定過程)

特徴

- 決定が現状態のみ依存
- 離散時間
- 報酬または罰を受ける世界

X_t : ランダムに選ばれるシステム上で可能な状態

x_t : 実際の状態

R_t : ランダムに選ばれるシステム上で可能な報酬

r_t : 実際の報酬

u_t : エージェントが選んだ行動

状態遷移

$$T(x_t, u_t) = X_{t+1}$$

$$P_{xy} = Pr[T(x, u) = y]$$

各状態と行動のペアの価値＝報酬の期待値

$$p(x_t, u_t) = E(R_t)$$

この価値関数から
価値が最大となる行動を選ぶ事で
合理的な方策が導かれる

例　迷路探索

目的
　各状態においてエージェントがゴールに向けて
一番良い行動を特定するような方策を学習する事

$$R(x_t)=\begin{cases}100 & x_t=10\\ 0 & otherwise\end{cases}$$

$$T(x_t,u_t)$$

	1	2	3	4	5	6	7	8	9	10
N	2	2	3	4	5	4	6	8	9	8
E	1	3	4	5	5	6	8	9	9	10
S	1	2	3	6	5	7	7	10	9	10
W	1	2	2	3	4	6	7	7	8	10

例　ポールバランシング

11.3 方策改善

強化学習が動く理由
　居所的に改善可能

局所的な改善
　マルコフ特性により
　各状態での行動は現状態と
　その状態から有効な状態へ遷移できる状態の
　知識に依存して選択される

DPと強化学習

DP

- 状態空間の中を一度逆に戻る
- 入力は最終状態と制御
- ダイナミクスは直前の状態で行う事を返す

強化学習

- ダイナミクスはわからない
- 前方への試行でダイナミクスを学習する
- 方策を学習するための反復が重要になる

方策改善定義

f : 方策(policy)

$Q(x, u)$: 行動価値関数

状態価値関数は次のように表される

$$V_f(x) = \max_u [Q(x, u)]$$

方策改善定義は

$$Q[x, g(x)] \geq V_f(x) \qquad then$$

$$V_g(x) \geq V_f(x)$$

11.4 Q 学習 (Q-Learning)

未来に得られる報酬を割り引いた収益

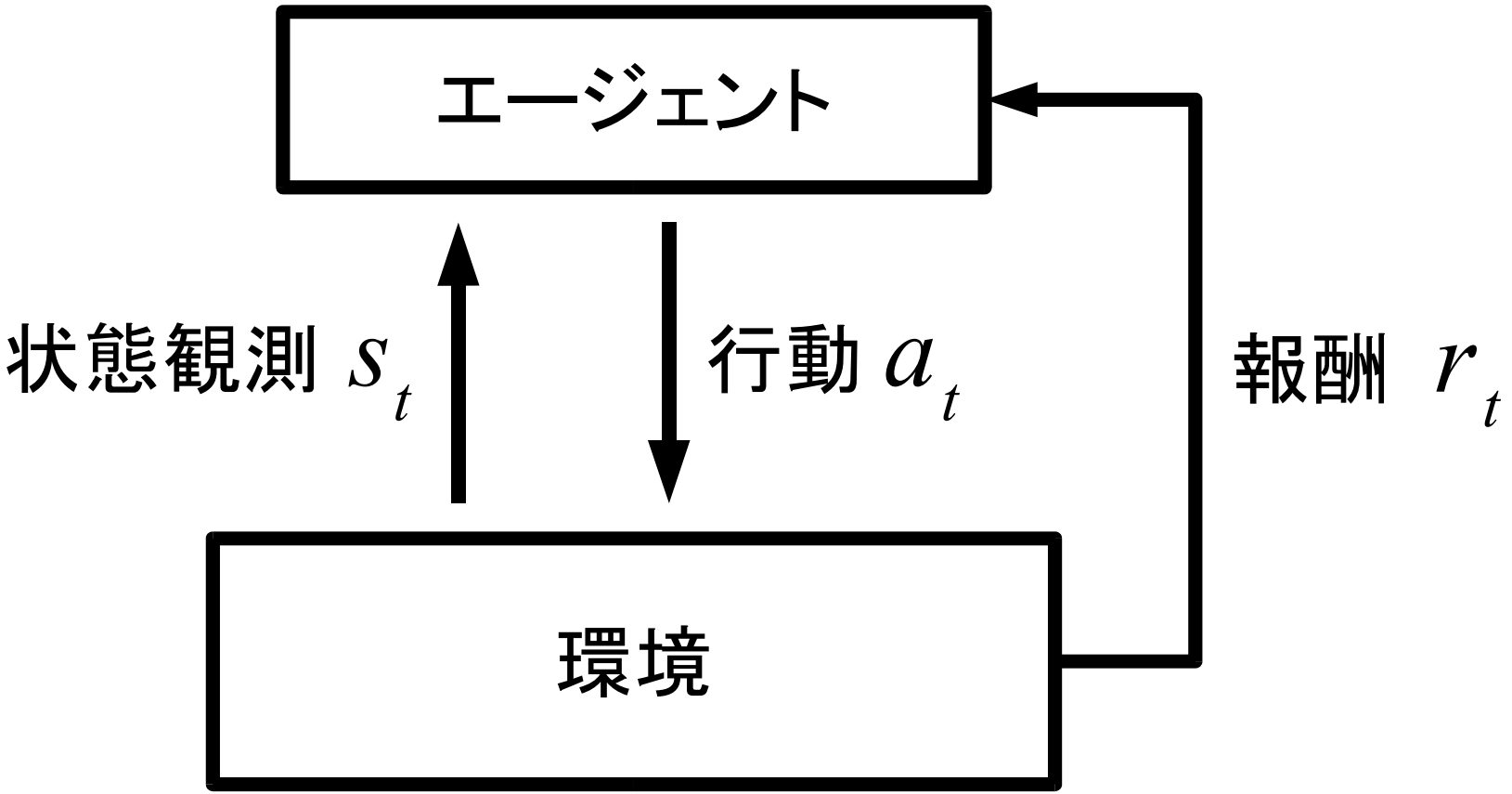
$$\sum_{t=0}^{n} \gamma^t r_t \qquad 0 \le \gamma \le 1$$

行動価値関数の更新
（ n=1:One-Step Q 学習）

$$Q^{new}(x,u) = (1-\eta)Q^{old}(x,u) + \eta\{r + \gamma Q^{old}[x', f(x')]\}$$

One-Step Q 学習アルゴリズム

強化学習の枠組み



Q 学習

収益（報酬の期待値）

$$R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots = \sum_{\tau=0}^{\infty} \gamma^{\tau} r_{t+\tau}$$

$\gamma\left(0 \le \gamma \le 1\right)$: 割引率

行動価値関数の更新
（1ステップ :Q 学習）

$$R_t = r_t + \gamma r_{t+1}$$

$$Q^{new} \leftarrow Q^{old} + \eta\left\{R_t - Q^{old}\right\}$$

$\eta\left(0 \le \eta \le 1\right)$: 学習率

行動価値関数の更新

$$R_t = r_t + \gamma r_{t+1}$$

$$f(x) = \arg\max_a Q(x, u)$$

$$r_{t+1} = Q[x', f(x')]$$

$$Q^{new}(x, u)$$

$$\leftarrow Q^{old}(x, u) + \eta\{r + \gamma Q^{old}[x', f(x')] - Q^{old}(x, u)\}$$

$$= (1-\eta) Q^{old}(x, u) + \eta\{r + \gamma Q^{old}[x', f(x')]\}$$

Q 学習のアルゴリズム

1. 状態観測を x に代入

2. 状態 x からの行動を方策 f に従ってを選ぶか
   または、ランダムに選ぶ

3. 行動結果を観測して x' に代入し、その時の報酬を r
   に代入

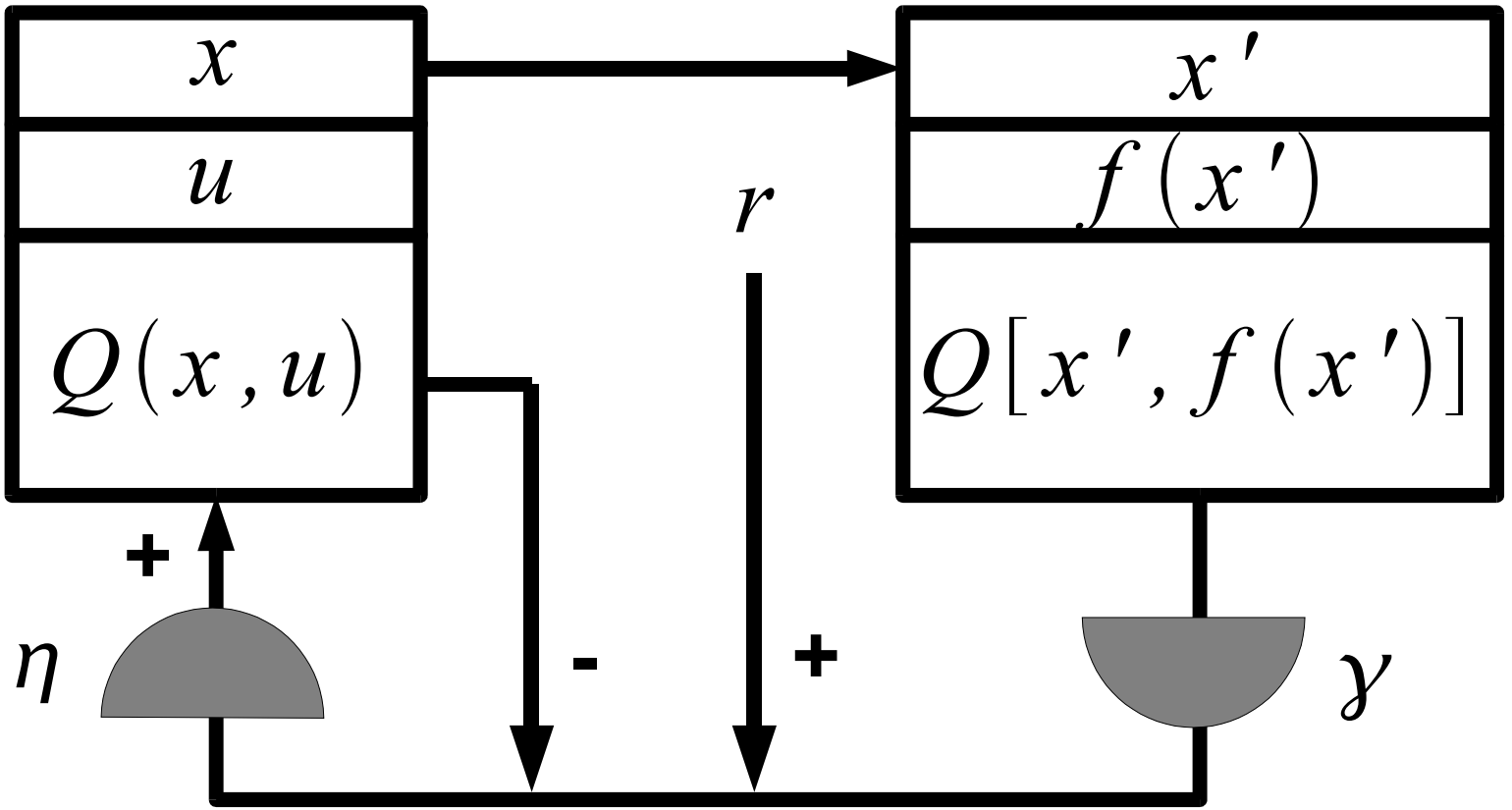
4. 行動価値関数の更新

$$Q^{new}(x,u) = (1-\eta)Q^{old}(x,u)+\eta\{r+\gamma Q^{old}[x', f(x')]\}$$

5. 方策の更新

$$f(x)=\arg\max_{a} Q(x,u)$$

更新の様子



$$Q^{new}(x,u)$$

$$\leftarrow\ Q^{old}(x,u)+\eta\{r+\gamma Q^{old}[x', f(x')]-Q^{old}(x,u)\}$$

$$=\ (1-\eta)Q^{old}(x,u)+\eta\{r+\gamma Q^{old}[x', f(x')]\}$$

11.5 TD 学習 (Temporal-Difference Learning)

Q 学習の問題点

・初めの方の実験が無駄になる
・最後から2番目の実験の影響が大きい

TD 学習の目的

無矛盾な予測を作ろうとするシステムを持つ事

本書での TD 学習アルゴリズム

本書での TD 学習アルゴリズムは
TD(λ) アルゴリズムの後方観測的な見方で
多層ニューラルネットを適応させて
誤差逆伝播法（関数近似は最急降下法）で
ステップ事に重みを更新して
状態価値関数Ｖの最適解を学習する事

TD(λ) 学習

収益

$$R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots = \sum_{\tau=0}^{\infty} \gamma^{\tau} r_{t+\tau}$$

n ステップ収益

$$R_t^{(n)} = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^{n-1} r_{t+n-1} + \gamma^n V(s_{t+n})$$

λ 収益

$$R_t^{\lambda} = (1-\lambda) \sum_{n=1}^{\infty} \lambda^{n-1} R_t^{(n)}$$

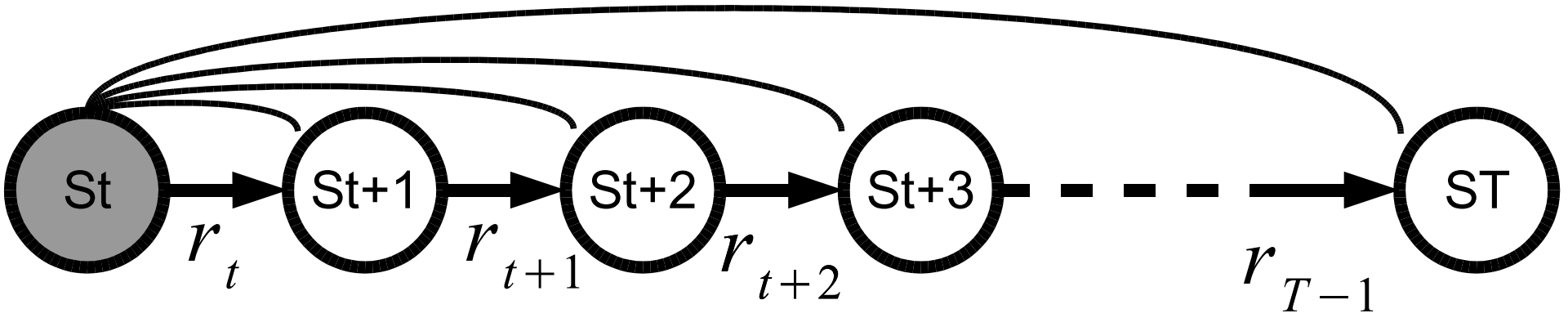
$$= (1-\lambda) \sum_{n=1}^{T-t-1} \lambda^{n-1} R_t^{(n)} + \lambda^{T-t-1} R_t$$

前方観測的な見方

（理想的）

更新

$$V(s_t) \leftarrow V(s_t) + \eta\left[R_t^{\lambda} - V(s_t)\right]$$



後方観測的な見方

（技法的）

適格度トレース

$$e_t(s)=\begin{cases} \gamma\lambda e_{t-1} & if\ s\neq s' \\ \gamma\lambda e_{t-1}+1 & if\ s=s' \end{cases}$$

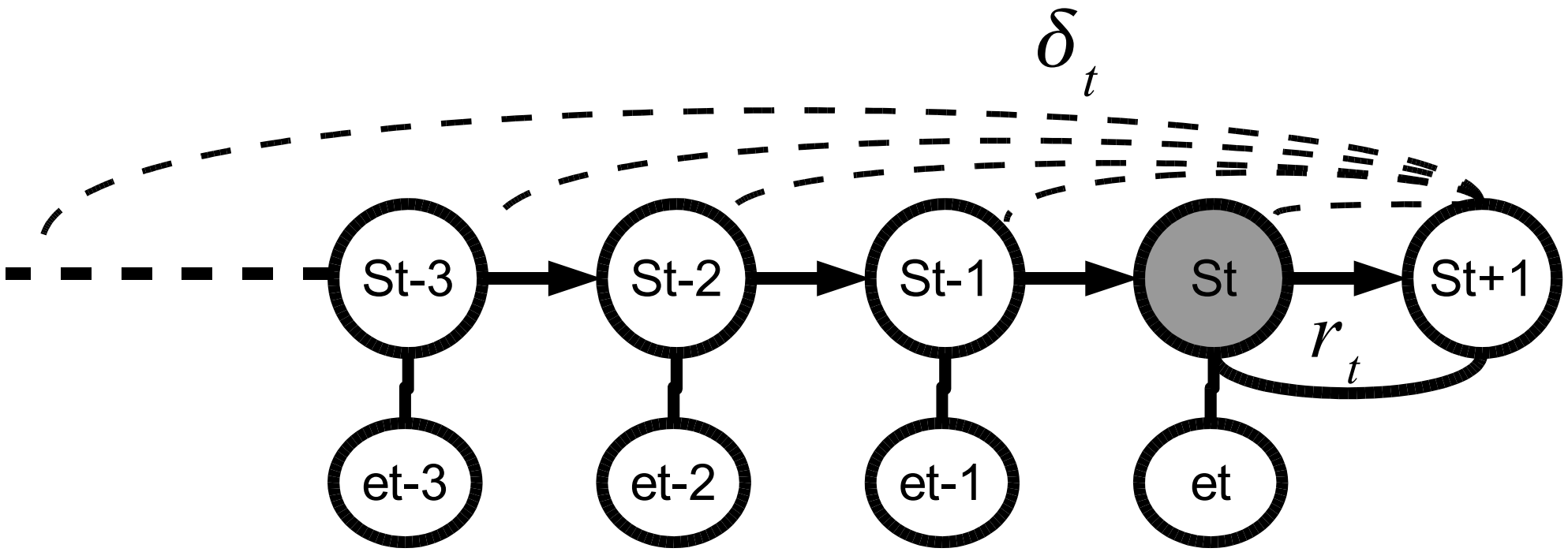
TD 誤差

$$\delta_t=r_t+\gamma V(s_{t+1})-V(s_t)$$

更新

$$V(s)\leftarrow V(s)+\eta\delta_t e_t(s) \qquad for\ \forall s\in S$$

後方観測的見方の様子



TD-gammon

バックギャモンのネットワークについて

入力ユニット数：１９８

盤面の各ポイントに対して
４個ユニットを割り当て
白の駒がなければ4個全ての値は0
白の駒が１個あると最初のユニットの値が1
白の駒が2個あると最初と2番目のユニットの値が1
白の駒が3個あると最初の3個のユニットの値は1
n>3 の時は4番目のユニットに (n-3)/2 の値が入る

２４個のポイントに対して白と黒で4ユニット＝１９２
付加ユニット（合計６個）
２個：白と黒でバーの上に n 個ある場合 n/2 の値が入る
２個：白と黒で盤面外に n 個ある場合 n/15 の値が入る
２個：どちらかの番かを2値化

関数近似による価値予測

目的

最適方策 π による状態価値関数 V^{π} を求める

注意

状態価値関数 V は ω をベクトルとする関数の形で新しく表現し、V は ω に依存する

ω を結合荷重とするニューラルネットで
表現すれば任意の V を求める事ができる

最急降下法

$$\omega' = \omega - \eta \frac{\partial J}{\partial \omega}$$

$$J = \frac{1}{2} \sum_{s} \left(V^{\pi}(s) - V(s) \right)^2$$

Jもωに依存した関数

$$\frac{\partial J}{\partial \omega} = -\left[V^{\pi} - V \right] \frac{\partial V}{\partial \omega}$$

$$\Delta\omega = \omega' - \omega = -\eta \frac{\partial J}{\partial \omega}$$

$$= \eta \left[V^{\pi} - V \right] \frac{\partial V}{\partial \omega}$$

最急降下法は勾配 $\frac{\partial V}{\partial \omega}$ を使ってωの更新を繰り返すことで状態間での誤差を均衡化し局所最適解 Vを求める

V^{π} が未知のままでは更新ができないので
真の価値を近似した v_t を使う

$$\Delta\omega = \eta[v_t - V_t]\frac{\partial V_t}{\partial \omega}$$

収束性
v_t に偏りがない場合
確率近似の条件に従って η を減少させる事で
ω は収束する事が保証される

このωの更新にTD(λ)の後方観測的見方を適応すると

$$\Delta\omega=\eta\,\delta_t e_t$$

$$\delta_t=r_t+\gamma V_{t+1}-V_t$$

$$e_t=\gamma\lambda e_{t-1}+\frac{\partial V_t}{\partial\omega}$$

（ここでのe_tはベクトル）

$$=\sum_{k=0}^{t}(\gamma\lambda)^{(t-k)}\frac{\partial V_t}{\partial\omega}$$

$$\Delta\omega=\eta\left(r_t+\gamma V_{t+1}-V_t\right)\sum_{k=0}^{t}(\gamma\lambda)^{(t-k)}\frac{\partial V_t}{\partial\omega}$$

１１．６　教師との学習

Learning with an External Critic (LEC)

　教師は生徒の行動を見て生徒を正すことができ、特別な外部報酬を与える

$$r_c(t)=\begin{cases} R_c & if\ YES \\ -R_c & if\quad NO \\ 0 & otherwise \end{cases}$$

もし、次のような事になると

$$Q(x,u)<0 \quad \text{and} \quad YES$$
or
$$Q(x,u)>0 \quad \text{and} \quad NO$$

その状態は矛盾した状態と考えられる

Learning by Watching (LBW)

生徒が教師の行動の効果を見ていて
その行動を理解するが報酬はでない

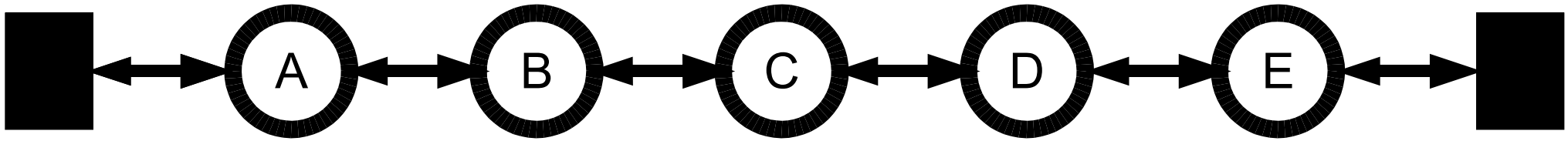
エージェントのスイッチによる動作

行動（act）
Q 学習

観測（observe）
教師の行動と状態のペアの流れを解釈

教師の決定と一致する場合　⇒　報酬増大
一致しない場合　⇒　報酬減少

An Abstract Model of Teaching



random walk

１１．７　部分観測マルコフ決定過程

状態観測

・ノイズやセンサの能力により不十分
・**MDP** は完全観測の数理モデル

PO-MDP（ Partially Observable MDP:
部分観測マルコフ決定過程）

・状態観測に不確実性を考慮した拡張 **MDP**

Perceptual Aliasing 問題

不確実な状態観測によって起こる問題

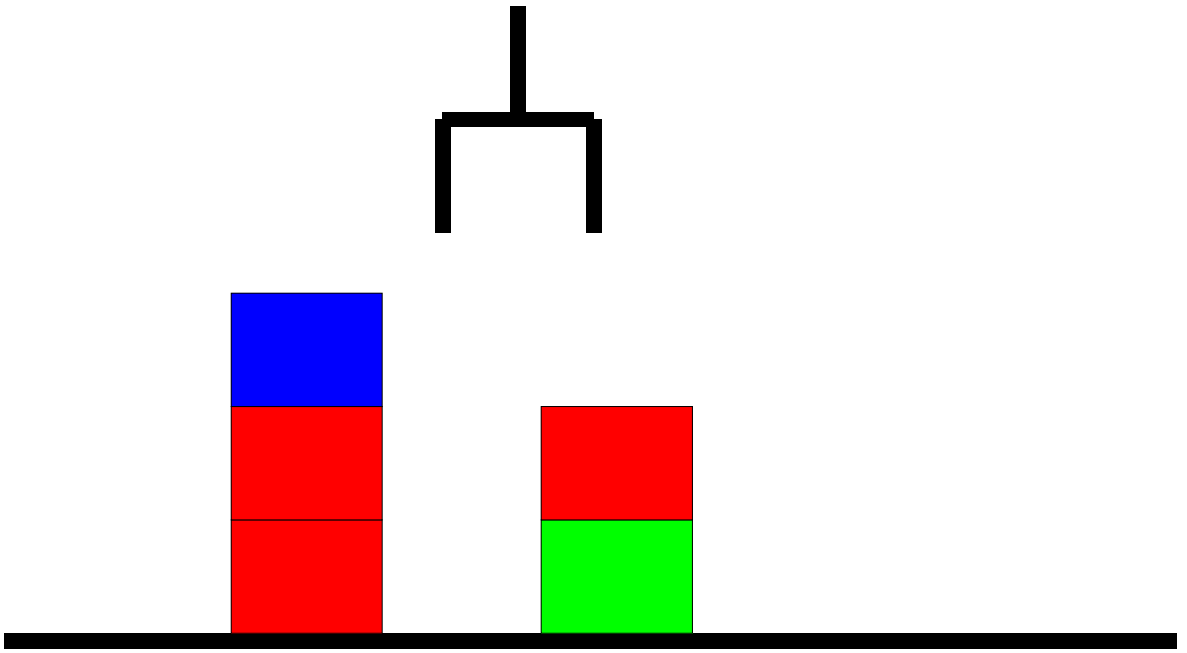
- 状態の混合
- 正しい方策を導く事ができない

解決案

- 状態を区別してモデルから削除
- 明白になるまで状態空間記述を拡張

１１．７．１　悪い状態を回避

例：カラーブロック問題



１１．７．２　時間順序からの学習

事例に基づく強化学習

・経験を事例として履歴に記録
・履歴から行動を決定

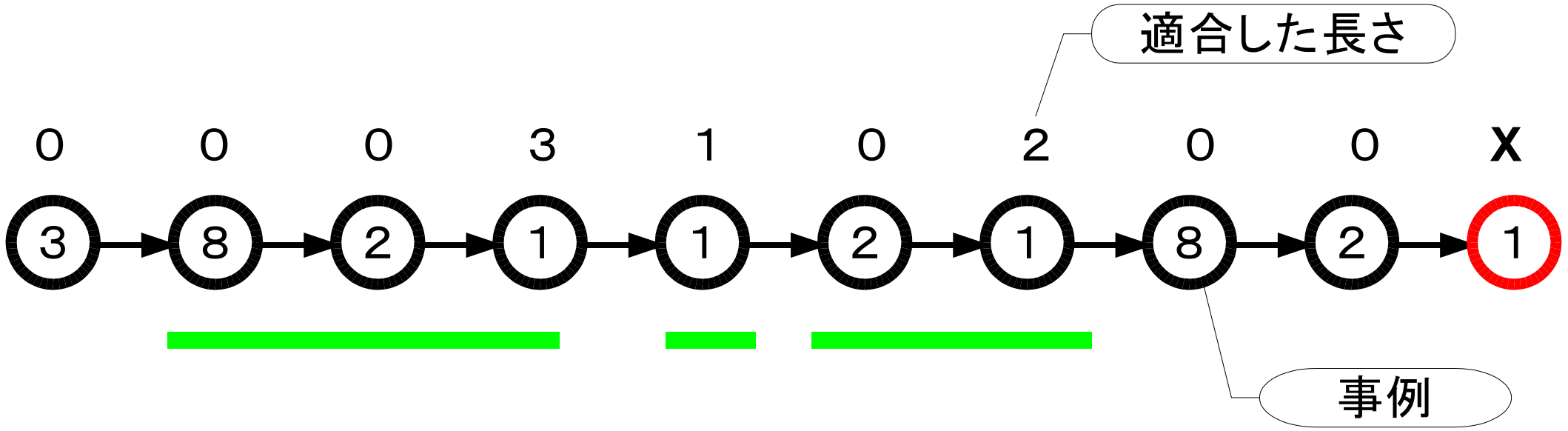
事例：x_t

o_t：観測された状態

a_t：行動

r_t：受け取った報酬

Nearest Sequence Memory



ここでは最近傍は適合した長さ

Nearest Sequence Memory

1. 履歴からもっとも似ている順序をk個見つける
2. 見つけた順序の最後の事例に投票
3. 投票者の価値を平均し、新しいＱ値を計算
4. 投票数がもっとも多い行動を選択するか
   （3で求めたＱ値が一番高い行動を選択）
   、またはランダムに選択する
5. その行動を実行し、新しい状況観測と報酬を記録する
6. Ｑ値の更新（Ｑ学習と同じ）

１１．７．３　状態価値を区別する

履歴の長さはどのぐらいがいいのか？

- 長いものより短くて有効性あるものが良い
- 上記の逆

ここでの目的

同程度の有効性を持つ区分に分ける

Utile Suffix Memory

特徴

- 履歴は木構造
- 枝の長さは必要に応じて可変長

事例: $T_t = \langle T_{t-1}, u_{t-1}, o_t, r_t \rangle$

葉ノード(状態): $x = L(T)$

事例参照: $\tau(x)$

Q の更新

$$Q(x,u) \leftarrow R(x,u) + \gamma P(x'|x,u) U(x')$$

収益

$$R(x,u) = \frac{\sum_{Ti \in \tau(x,u)} r_i}{|\tau(x,u)|}$$

確率

$$P(x'|x,u) = \frac{|\forall T_i \in \tau(x,u) \, s.t. \, L(T_{t+1}) = x'|}{|\tau(x,u)|}$$