

# 強化学習を ベイズで理解する

牧野 貴樹<sup>†</sup>

東京大学 生産技術研究所

<sup>†</sup> 総合科学学術会議により制度設計された最先端研究開発支援プログラム(FIRST 合原最先端数理モデルプロジェクト)により、日本学術振興会を通じて助成を受けた。

# 概要

- 神経科学や機械学習の専門家でも、「強化学習は難しい」という人が少なくない
- 理由は、強化学習が何に着目しているのか、問題設定がとらえにくいから
- 強化学習の問題設定を丁寧に解説する
  - すでにご存じの部分もあるとは思いますが...

# 強化学習とは何か

## 教師つき学習

正解つきのデータをもとに、  
正解を導く規則を獲得する

## 強化学習

正解は与えられないが  
選んだ答えの「良さ」(報酬)が  
分かる時に最良解を探す

## 教師なし学習

正解のない大量データから、  
背後の規則を獲得する

- 機械学習の1分野であるが、  
「教師つき学習」とも「教師なし学習」とも異なる

# 強化学習とは何か

## 強化学習

正解は与えられないが  
選んだ答えの「良さ」(報酬)が  
分かる時に最良解を探す

データを自ら作る  
収集する対象を選びながら  
同時に分析する

牧野 貴樹: 強化学習をベイズで理解する

## 教師つき学習

正解付きのデータをもとに、  
正解を導く規則を獲得する

## 教師なし学習

正解のない大量データから、  
背後の規則を獲得する

すでに収集されたデータ  
どんどん入ってくるデータを  
分析する

2014/3/19

これから?

## 強化学習

正解は与えられないが  
選んだ答えの「良さ」(報酬)が  
分かる時に最良解を探す

バンディット問題

能動学習

調査計画

データを自ら作る  
収集する対象を選びながら  
同時に分析する

## ビッグ データ

### 教師つき学習

正解付きのデータをもとに、  
正解を導く規則を獲得する

半教師つき学習

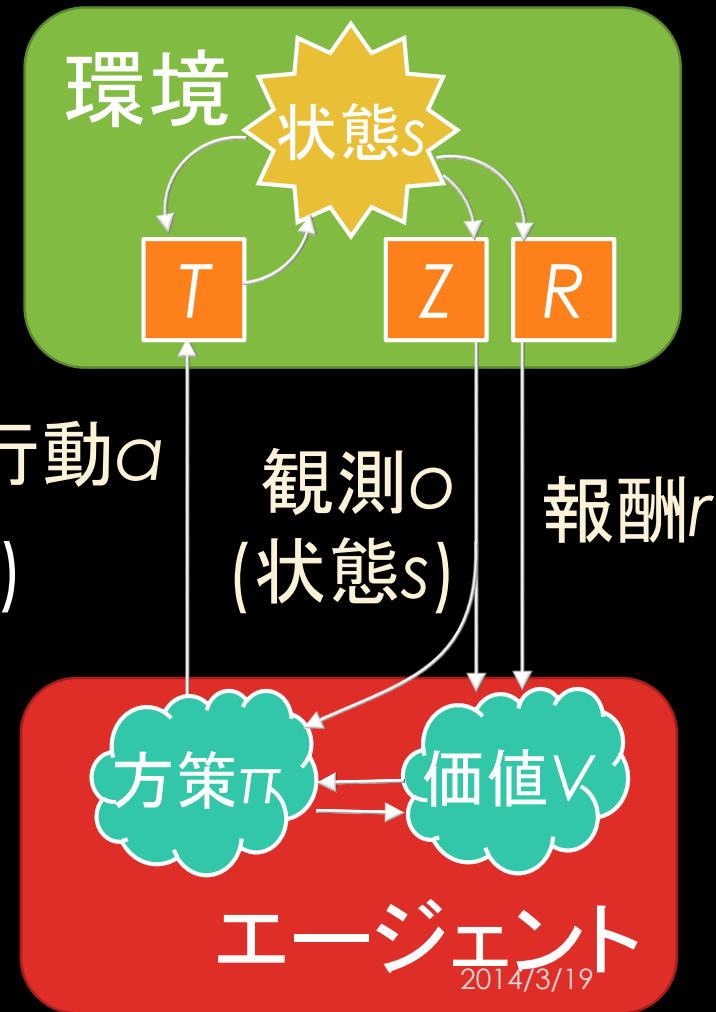
### 教師なし学習

正解のない大量データから、  
背後の規則を獲得する

すでに収集されたデータ  
どんどん入ってくるデータを  
分析する

# 強化学習とは？

- 未知の環境の中を探索しながら期待報酬和を最大化するためのエージェントの行動原理
- 「不確実である」ことは本質的ではない (が、問題を複雑にする)
  - 遷移が確率的 (ギャンブル)
  - 観測が不完全  
→ 現在の状態が不確実





# 試行錯誤しながら最適化



既知の知識の中で最適化  
データがない選択肢に  
よいものがあっても選べない

未知の選択肢に挑戦して  
データを収集する  
多くの場合、成績は下がる

- 『山のあなたの空遠く、「幸」住むと人の言ふ』(カール・ブッセ)  
「普段の日常を送る」→よく知っている、悪くない結果になる  
「山の向こうに行く」→データはない。知るには手間が要る
- 期待される結果を最適化するには、どう行動したらよいか？

# なぜ強化学習が重要か

- 自律システムの構築の基盤
  - ロボット、携帯基地局、エレベータ、...
- データ収集・行動計画のコスト最適化
  - A/Bテスト ... コスト=テスト実施に係る機会損失
- ヒトを説明するための論理的基盤
  - ヒトの行動・脳内の機構を、強化学習で説明する
- 応用例
  - 滞納税金取り立て最適化 (Abe+,2010)
  - ヘルプ文書理解 (Branavan+,2009)
  - 構文解析の高速化 (Neu&Szepevari, 2009)

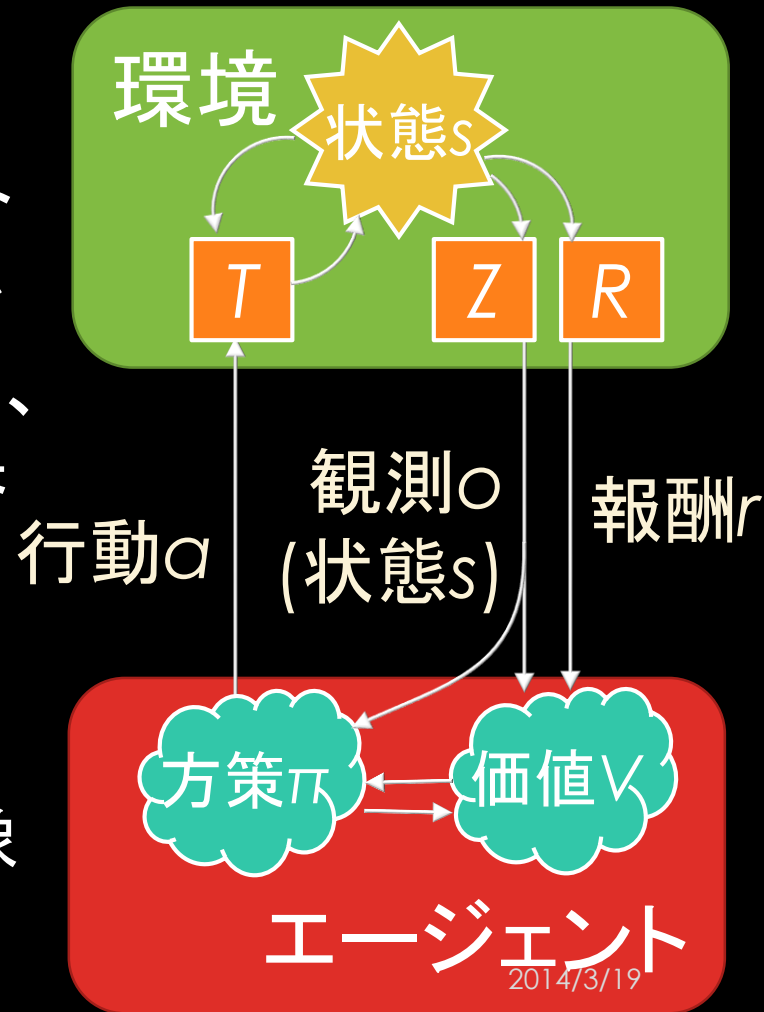


# 本 talk の概要

- Multi-armed Bandit Problem
  - 「不確かな時は楽観的に」原理
- 離散状態の強化学習問題
  - 環境モデルが既知の場合
  - Bayes-Adaptive Markov Decision Process
  - 近似戦略
- 問題設定の拡張
  - 報酬設定問題
  - 徒弟学習 (逆強化学習)

# 用語の定義 (1)

- 学習し、行動を選択する主体を「エージェント」と呼び、エージェントが働きかける対象を「環境」と呼ぶ
- エージェントが選択した行動により、環境の「状態」が変化し、その結果を「観測」できる
- 状態 (と行動) に応じて「報酬」というスカラー値が与えられる
- 「方策」は、状態から行動への写像 (エージェントが自由に設定できる)

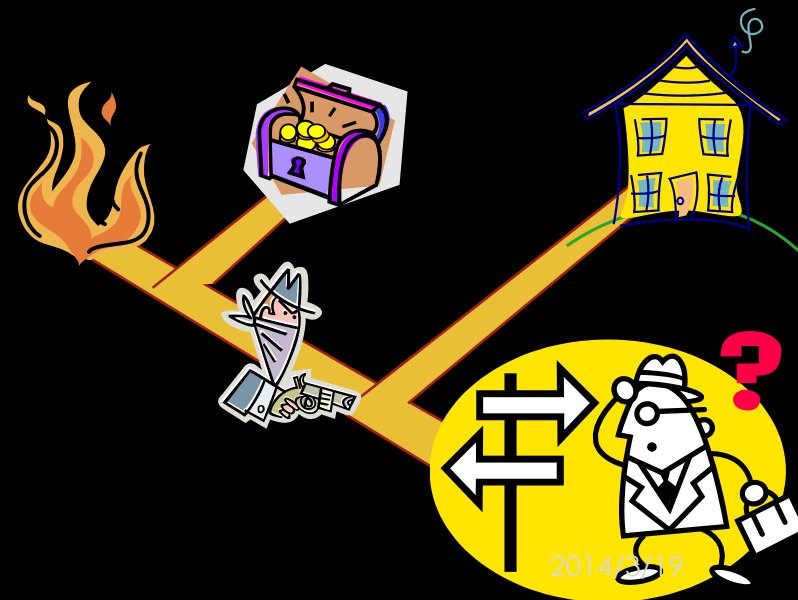


## 用語の定義 (2)

- 状態の「価値」とは、将来に期待される報酬の和 (未来の報酬を割引することが多い)

$$V(s) = \mathbb{E}[r_0 + \gamma r_1 + \gamma^2 r_2 + \gamma^3 r_3 + \dots]$$

- $\gamma$ : 割引率 ( $0 \leq \gamma < 1$ )
- 価値は方策に依存する
- 強化学習とは、ある環境に関して、開始状態の価値を最大にするような方策を決定する問題



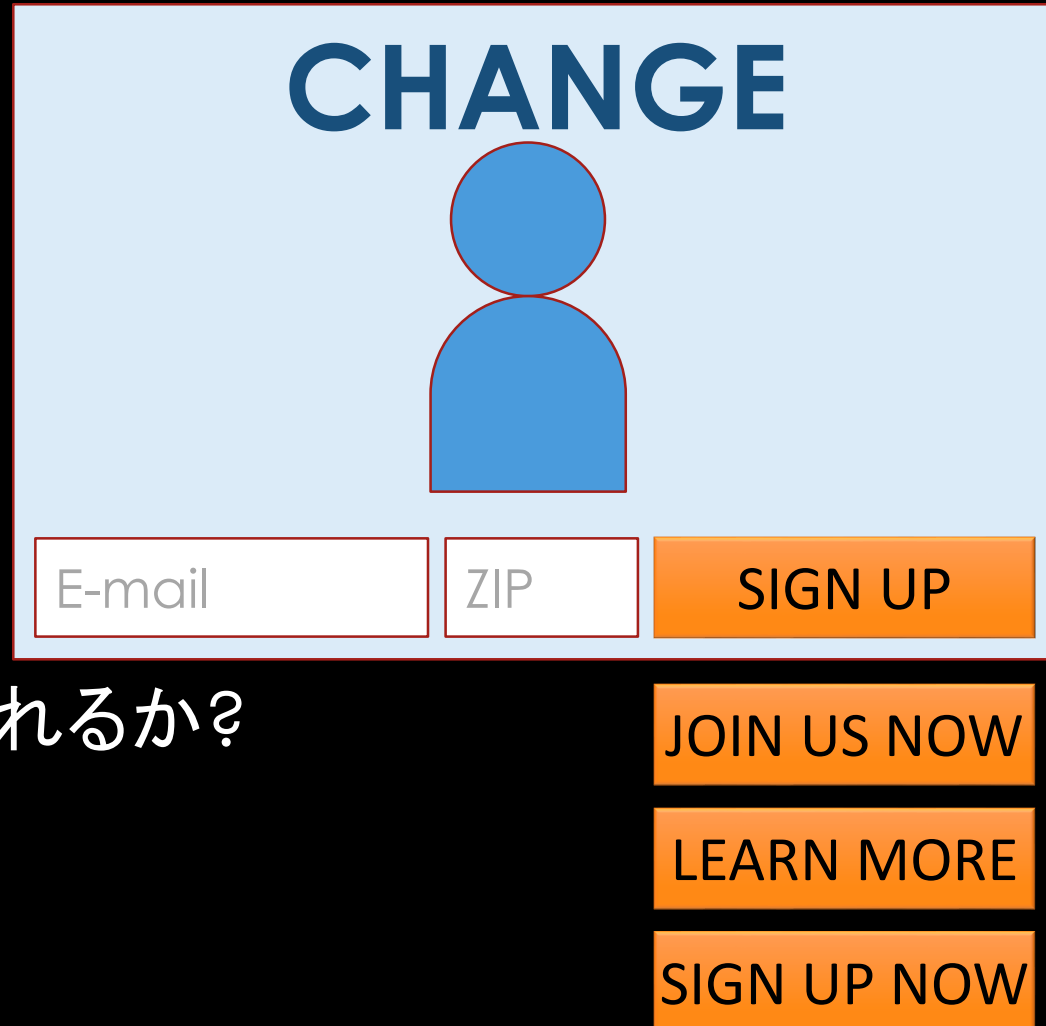
# Multi-armed Bandit (Stochastic)

- 腕がK本あるスロットマシン
  - 各々の腕が異なる報酬の確率分布をもつ (プレイヤーは知らない)
  - 結果の履歴から、次にどの腕に挑戦するかを決めたい
  - 簡単のため、報酬は  $[0, 1]$  区間におさまるものとする
- 強化学習においては、「状態」が変化しない、もっとも単純なケースに相当する



# 実問題の例 (1) (Siroker, 2010)

- 2008年大統領選挙  
オバマ候補の選挙  
キャンペーンweb  
サイトのトップページ  
のボタンのラベル
- 4種類のうち、どの  
ラベルを使うと、最も  
多くの人が登録してくれるか?



The image shows a sign-up form from the Obama 2008 campaign website. At the top, the word "CHANGE" is displayed in large blue letters. Below it is a blue silhouette of a person. The form includes two input fields: "E-mail" and "ZIP". To the right of these fields is an orange button labeled "SIGN UP". Below the "SIGN UP" button, there are three more orange buttons stacked vertically: "JOIN US NOW", "LEARN MORE", and "SIGN UP NOW".

# 実問題の例 (1) (Siroker, 2010)

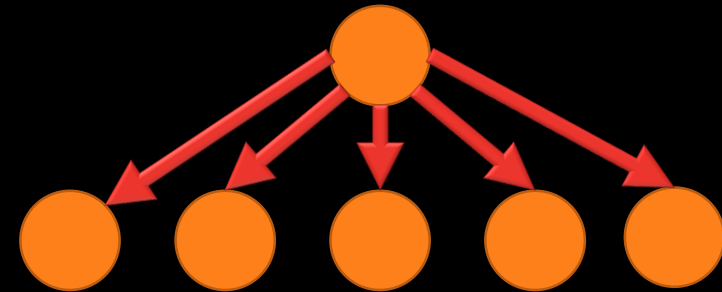
- コンバージョン(訪問した人のうち登録した人の割合) は、ボタンラベルによって大きく異なった
  - 単純に最終登録者数1000万人から計算すると約150万人分
- A/B テストによる調査結果
  - 約 31万人の訪問者に対してランダムに表示し情報収集
  - 調査対象者を増やせばより正確になるが、最適なラベルを表示できる回数が減る

ラベル	コンバージョン	増減
SIGN UP	7.51%	---
JOIN US NOW	7.62%	+1.4%
LEARN MORE	8.91%	+18.6%
SIGN UP NOW	7.34%	-2.4%



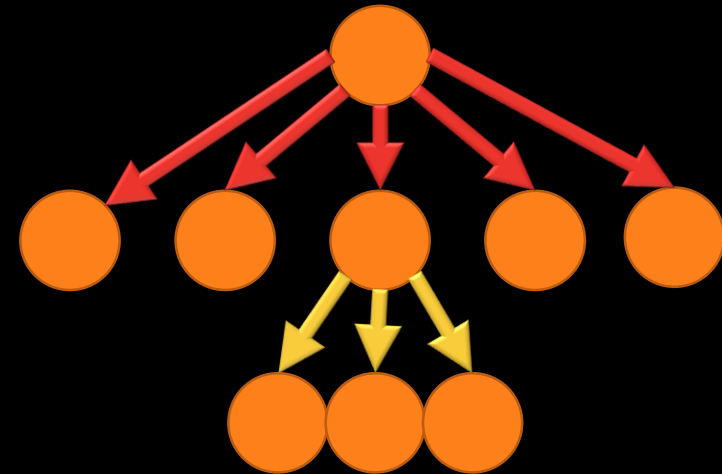
# 実問題の例 (2)

- 囲碁の盤面評価 (Gelly&Wang 2006)
  - 盤面の評価値をモンテカルロ法で計算 (プレイアウト)
  - 有望そうな手に対して多くのプレイアウトを割り当てたい  
→ Multi-armed Bandit の解法を使って木構造を構築



# 実問題の例 (2)

- 囲碁の盤面評価 (Gelly&Wang 2006)
  - 盤面の評価値をモンテカルロ法で計算 (プレイアウト)
  - 有望そうな手に対して多くのプレイアウトを割り当てたい  
→ Multi-armed Bandit の解法を使って木構造を構築



# 問題設定

- $K$ : 腕の本数
- $P_i$ : 腕  $i$  を選んだ時の報酬の確率分布 ( $[0,1]$  区間上)
  - $\mu_i$ : 腕  $i$  を選んだ時の報酬の期待値
  - $\mu^* = \max_i \mu_i$  (最適な選択で得られる報酬の期待値)
- プレイヤーは各時刻  $t$  で腕  $i_t$  を選び、結果  $x_t | t, i_t \sim P_{i_t}$  を観測する
- 評価基準: regret (最適な行動をした場合と比較した損失の期待値)

$$R(T) = \sum_{t=1}^T (\mu^* - \mu_{i_t})$$

# Greedy Algorithm

- 最初に各腕を一回ずつ選ぶ
- それ以降は期待値が最高の腕を選ぶ

$$i(t+1) = \underset{i}{\operatorname{argmax}} \bar{x}_i(t) \quad \bar{x}_i(t) = \frac{S_i(t)}{T_i(t)}$$

- $S_i(t) = \sum_{t:i_t=i} x_t$ : 時刻  $t$  までに腕  $i$  から得られた報酬の和
- $T_i(t) = \sum_{t:i_t=i} 1$ : 時刻  $t$  までに腕  $i$  を選んだ回数
- 「既存の知識の利用」に対応する

# Greedy Algorithm の例

腕1 (0.35)

腕2 (0.4)

腕3 (0.45)

腕4 (0.5)

腕1

期待値 ---

腕2

期待値 ---

腕3

期待値 ---

腕4

期待値 ---

# Greedy Algorithm の例

腕1 (0.35)

腕2 (0.4)

腕3 (0.45)

腕4 (0.5)

0

1

0

1

腕1

腕2

腕3

腕4

期待値 0

期待値 1

期待値 0

期待値 1



# Greedy Algorithm の例

腕1 (0.35)

腕2 (0.4)

腕3 (0.45)

腕4 (0.5)

0

10

0

10

腕1

腕2

腕3

腕4

期待値 0

期待値 0.5

期待値 0

期待値 0.5

# Greedy Algorithm の例

腕1 (0.35)

腕2 (0.4)

腕3 (0.45)

腕4 (0.5)

0

10100100

0

100

腕1

腕2

腕3

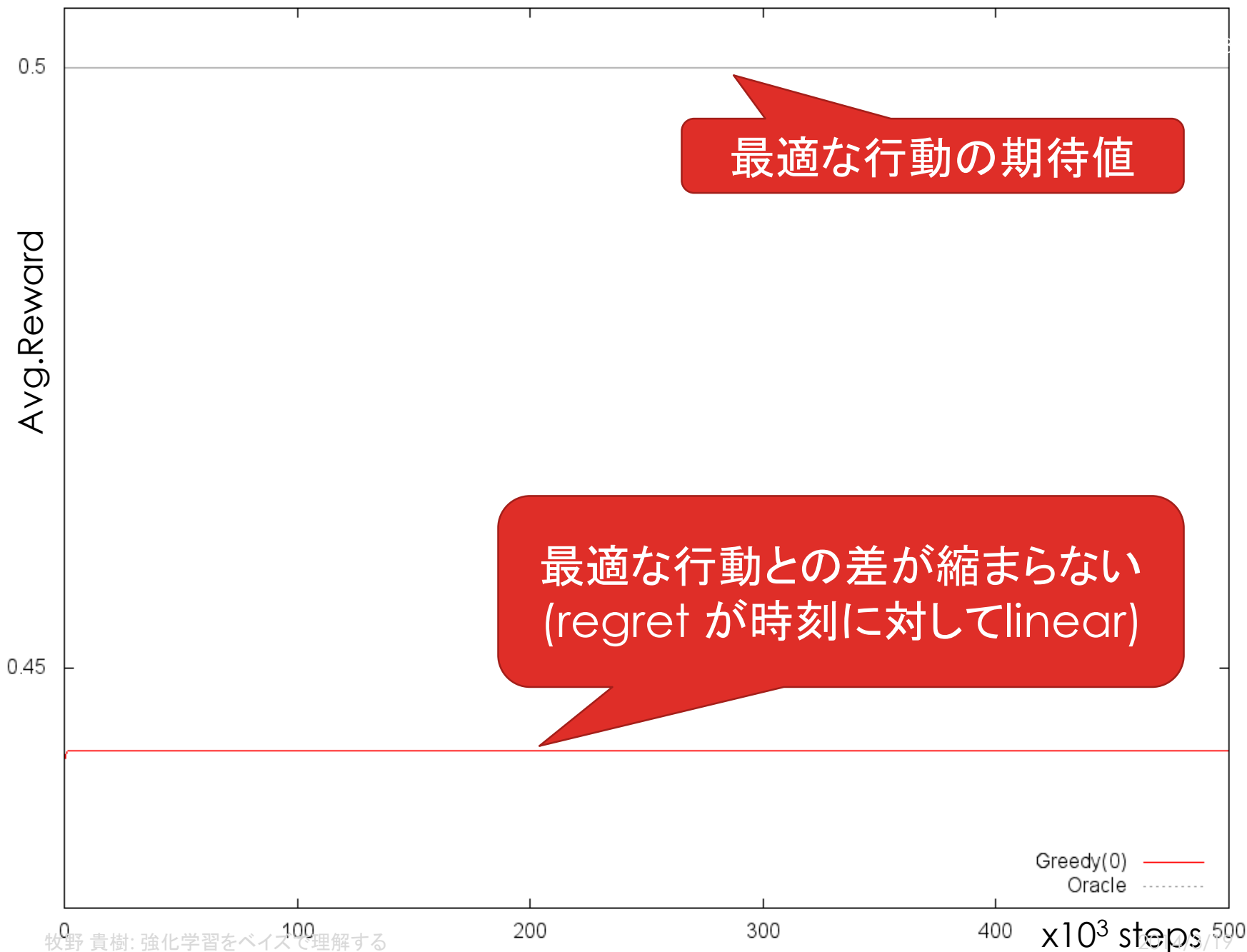
腕4

期待値 0

期待値 0.38

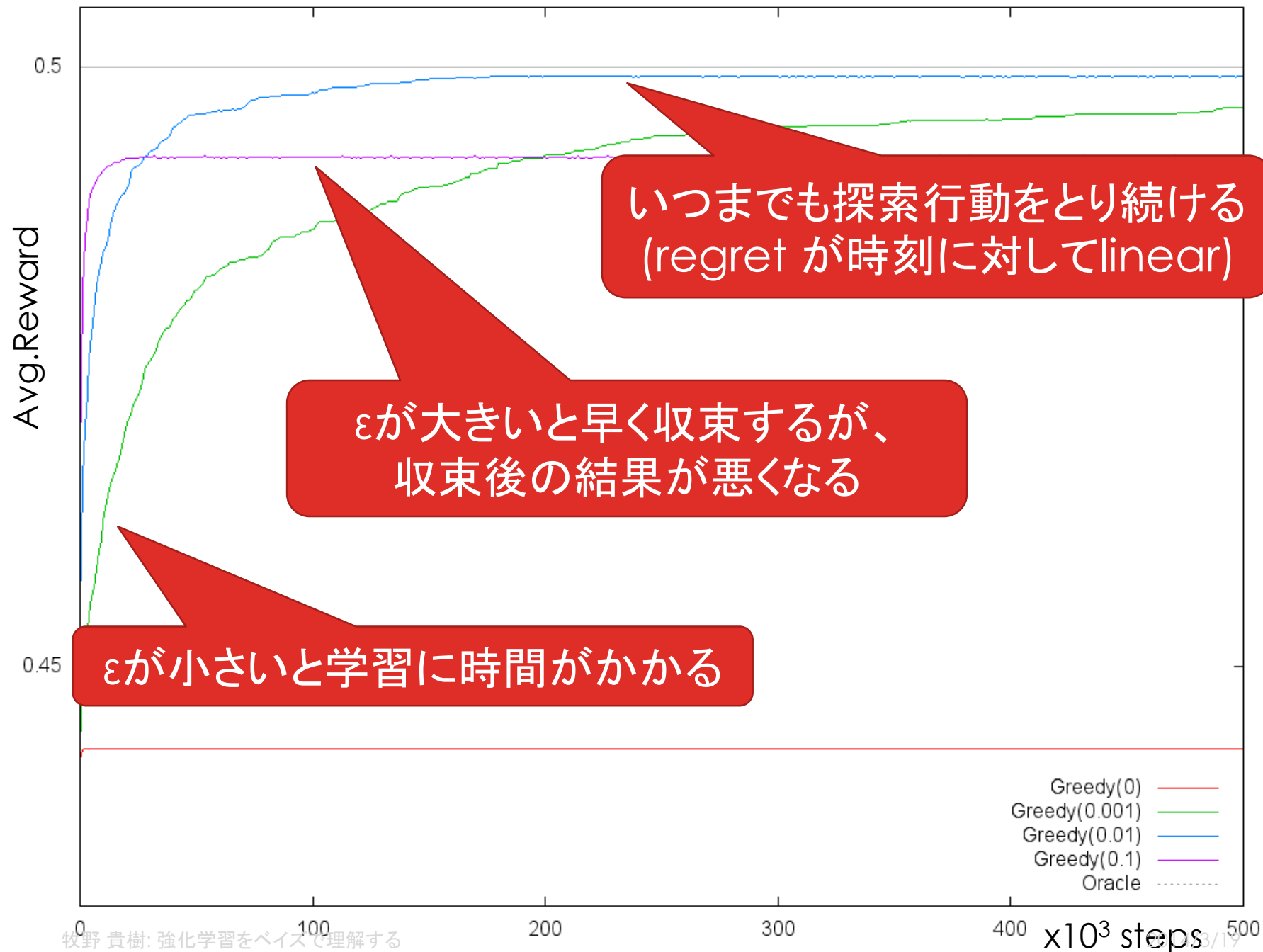
期待値 0

期待値 0.33



# $\epsilon$ -Greedy algorithm

- 確率  $\epsilon$  でランダムな腕を選択する (「探索」)
- 確率  $1 - \epsilon$  でGreedyに腕を選択する (「利用」)
- 多くの試行を繰り返していけば、いつかはすべての腕に関して十分なデータが蓄積され、最善な手を選択できる



# K-armed bandit における regret の性質

- どんなアルゴリズムを使ったとしても、少なくとも  $O(\log t)$  の regret は発生する (Lai&Robbins 1985)
  - 「これ以上探索は不要」という状態は永遠に來ない
  - 1時刻あたりの regret は、いくらでも0に近づけることが可能である
- $\varepsilon$ -Greedy の  $\varepsilon$  を  $O(1/t)$  で変化させることで、 $O(\log t)$  の regret は実現できる (Auer+, 2002)
  - ただし、係数がかなり大きく、あまり実用的ではない
- もっと賢い探索の方法はないか？



# Optimism in face of uncertainty

- 「不確かな時には楽観的に」
- より具体的な説明 (Bubeck & Cesa-Bianchi 2012):
  1. 現在の知識と整合性のある「想定しうる環境」の集合を想定
  2. その集合から「もっとも都合のよい」環境を選ぶ
  3. もっとも都合のよい環境における最適解を、次の行動とする
- 発見的な原理でしかない (証明はない) が、多くの場合に有効

# UCB1 algorithm (Auer+, 2002)

Upper Confidence Bound

- 最初に各腕を1回ずつ選ぶ
- それ以後は、各時刻  $t$  において、 $UCB_i(t)$ を最大にする腕  $i$ を選ぶ

$$UCB_i(t) = \bar{x}_i(t) + \sqrt{\frac{2 \ln t}{T_i(t)}}$$

- 定理: 時刻  $t$  におけるUCB1アルゴリズムのregretは

$$R(t) \leq \left\lceil 8 \sum_{i: \mu_i < \mu^*} \left( \frac{\ln t}{\mu^* - \mu_i} \right) \right\rceil + \left( 1 + \frac{\pi^2}{3} \right) \left( \sum_{j=1}^K (\mu^* - \mu_j) \right)$$

$O(\log t) + \text{定数}$

# 統計学的推論

- 時刻 $t$ までの結果から、各腕の報酬の期待値 $\mu_i$ の事後分布を考える
- Chernoff-Hoeffding Bound (Hoeffding, 1963)
  - $X_1, \dots, X_n$ を  $\mathbb{E}[X_t | X_1, \dots, X_{t-1}] = \mu$  を満たす範囲  $[0, 1]$  の random variable とすると、任意の  $a \geq 0$  に対し次が成り立つ

$$P(X_1 + \dots + X_n \geq n\mu + a) \leq \exp\left(-\frac{2a^2}{n}\right)$$

$$P(X_1 + \dots + X_n \leq n\mu - a) \leq \exp\left(-\frac{2a^2}{n}\right)$$

# UCB1の上限の証明

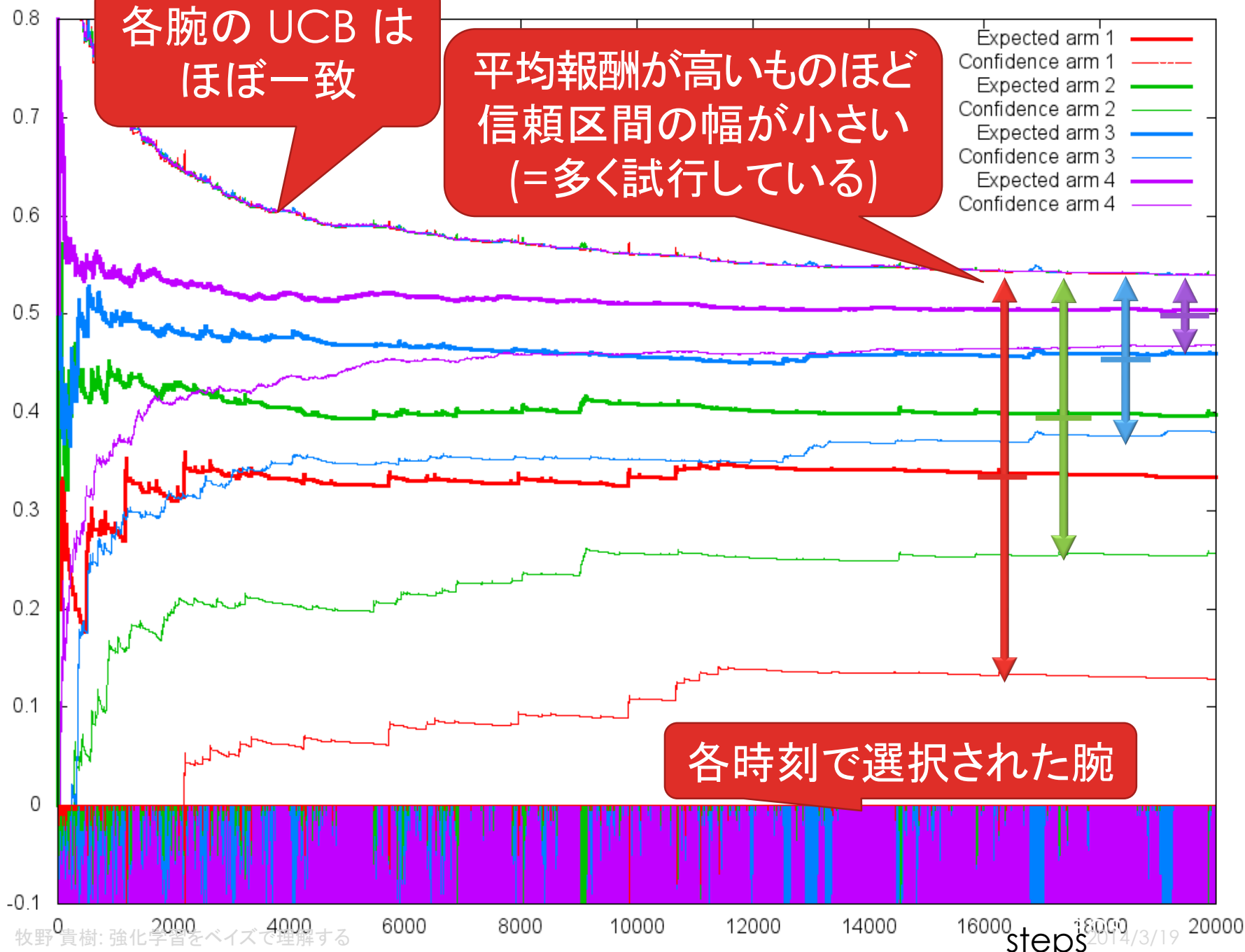
- 最適でない腕  $i$  を選ぶ回数  $T_i$  の期待値の上限を示す
- 腕  $i$  を選ぶ条件は  $\bar{x}^* + c^* < \bar{x}_i + c_i$
- $c_i = \sqrt{\frac{2 \ln t}{T_i}}$  とすると、Chernoff-Hoeffding bound より
 
$$P(\bar{x}^* \leq \mu^* - c^*) \leq t^{-4}, \quad P(\bar{x}_i \geq \mu_i + c_i) \leq t^{-4}$$
- 時刻  $t$  までにいずれか満たす回数の期待値は高々  $1 + \pi^2/3$  回
- $T_i \geq \frac{8 \ln t}{(\mu^* - \mu_i)^2}$  の時は、上の2条件以外で腕  $i$  は選ばれない
  - 必ず  $\mu^* \geq \mu_i + 2c_i$  を満たす
- よって  $\mathbb{E}[T_i] \leq \frac{8 \ln t}{(\mu^* - \mu_i)^2} + \left(1 + \frac{\pi^2}{3}\right)$

各腕の UCB は  
ほぼ一致

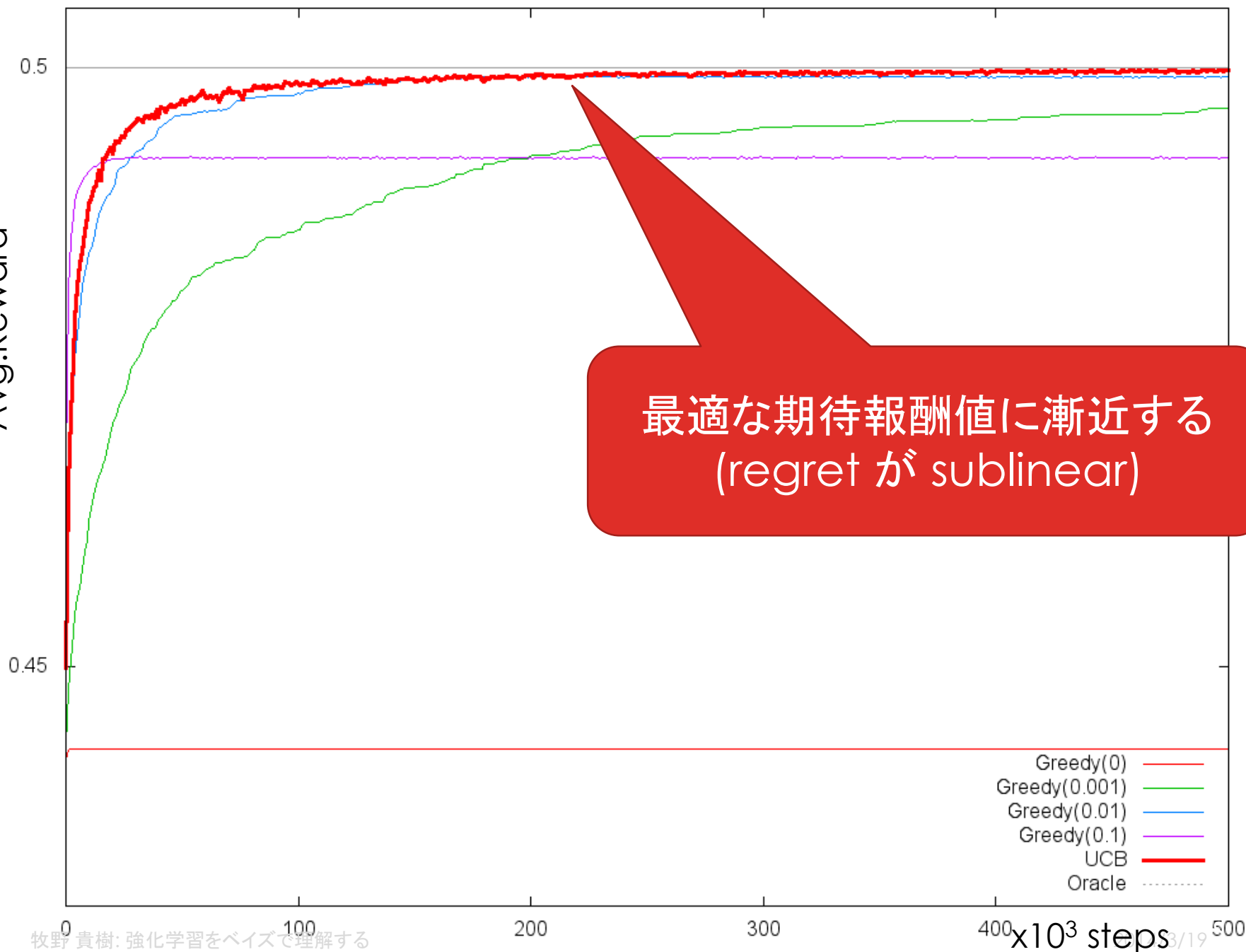
平均報酬が高いものほど  
信頼区間の幅が小さい  
(=多く試行している)

Expected arm 1  
Confidence arm 1  
Expected arm 2  
Confidence arm 2  
Expected arm 3  
Confidence arm 3  
Expected arm 4  
Confidence arm 4

各時刻で選択された腕



Avg. Reward



最適な期待報酬値に漸近する  
(regret が sublinear)



# UCB系アルゴリズム補足

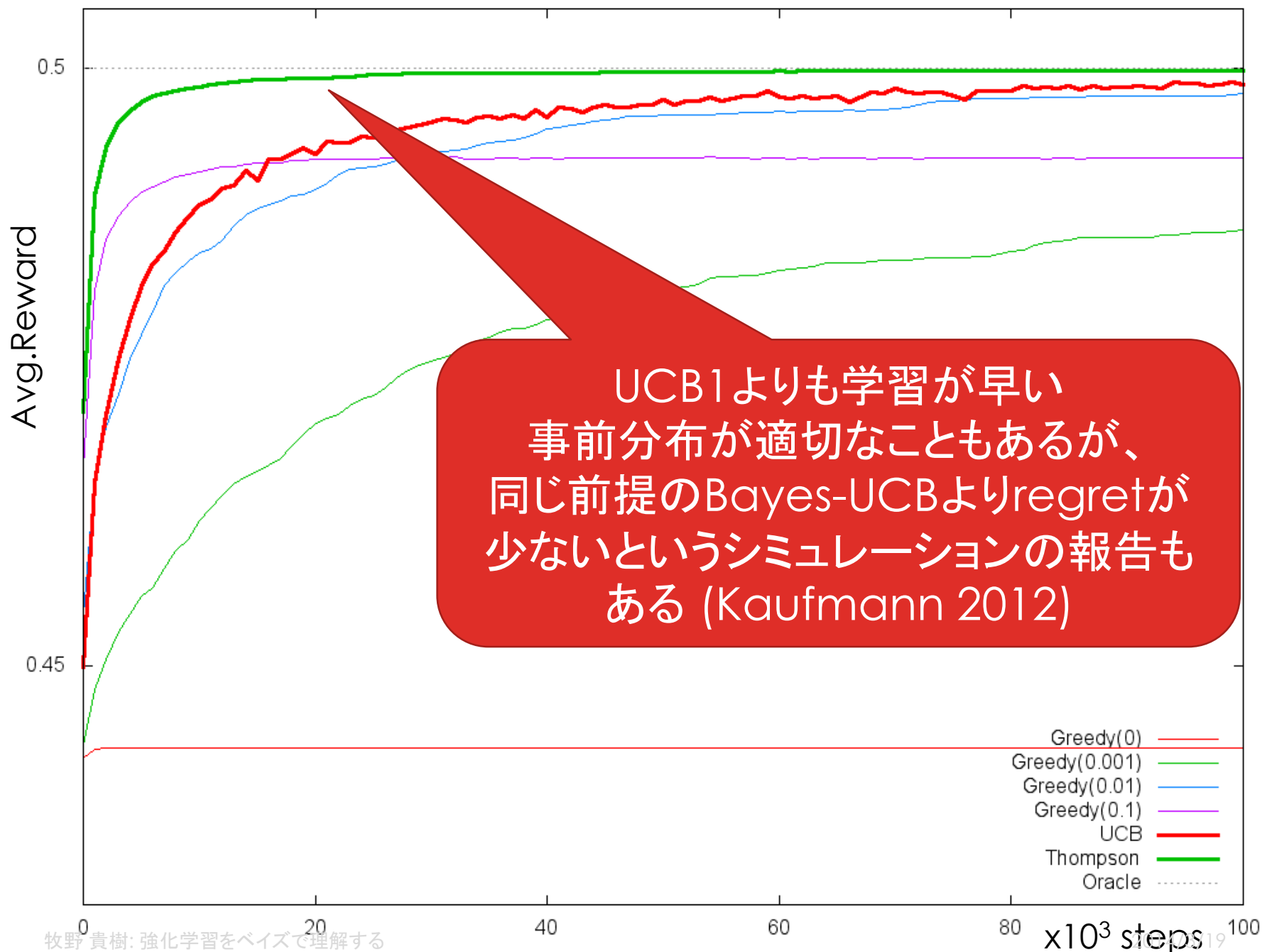
- UCB1は、有限時刻 $t$ におけるregretの上限が示されたはじめてのアルゴリズム
  - 前提が少なく、計算が簡単で使いやすい
- 報酬の事前分布や、比較指標の導出方法を変えた様々なバリエーションが研究されている
  - DMED (Honda&Takemura 2010): 報酬の経験分布とKL-divergence で指標を計算
  - Bayes-UCB (Kaufmann+ 2012): 各腕にベイズ事前分布を与え、quantileで信頼区間を計算
  - などなど...

# Thompson Sampling

(Thompson, 1933)

- 各腕の報酬がBernoulli分布の場合を仮定
  - 確率  $\mu_i$  で1、それ以外は0が出る
- 1.  $\mu_i$  の事前分布  $\pi_{i,0}$  を区間  $[0,1]$  の上の一様分布とする
- 2. 時刻  $t$  において:
  - 2-1. 各腕に対して  $\theta_{i,t} \sim \pi_{i,t-1}$  をサンプリング
  - 2-2.  $\theta_{i,t}$  が最大になる腕  $i_t$  を選んでプレイ
  - 2-3. 観測結果から事後分布  $\pi_{i,t}$  を計算
- 最近 Regret  $O(\log t)$  が証明された (Kaufmann+, 2012)

時刻  $t-1$  における  
 $\mu_i$  の事後分布



# Bandit 問題に関するその他の話題

- こちらの選んだ手に応じて報酬分布が変化する場合 (adversarial bandit)
- 腕が試しきれないくらい多い場合 ( $k \gg t$ )
- 腕が連続空間の場合
- 各腕の試行がi.i.d.ではない場合
  - 各々の腕にマルコフモデルが関連づけられている場合など
- 文脈によって分布が異なる場合

# ここまでのまとめ

- Multi-armed Bandit: 最も単純な強化学習の問題
  - 探索と利用のトレードオフ
  - 「不確かな時には楽観的に」原理
  - $\epsilon$ -Greedy: 乱数に基づく探索
  - UCB1: 信頼区間の上限の比較に基づく決定的アルゴリズム
  - Thompson Algorithm: ベイズ事後分布に基づく確率的アルゴリズム

# 一般の強化学習問題

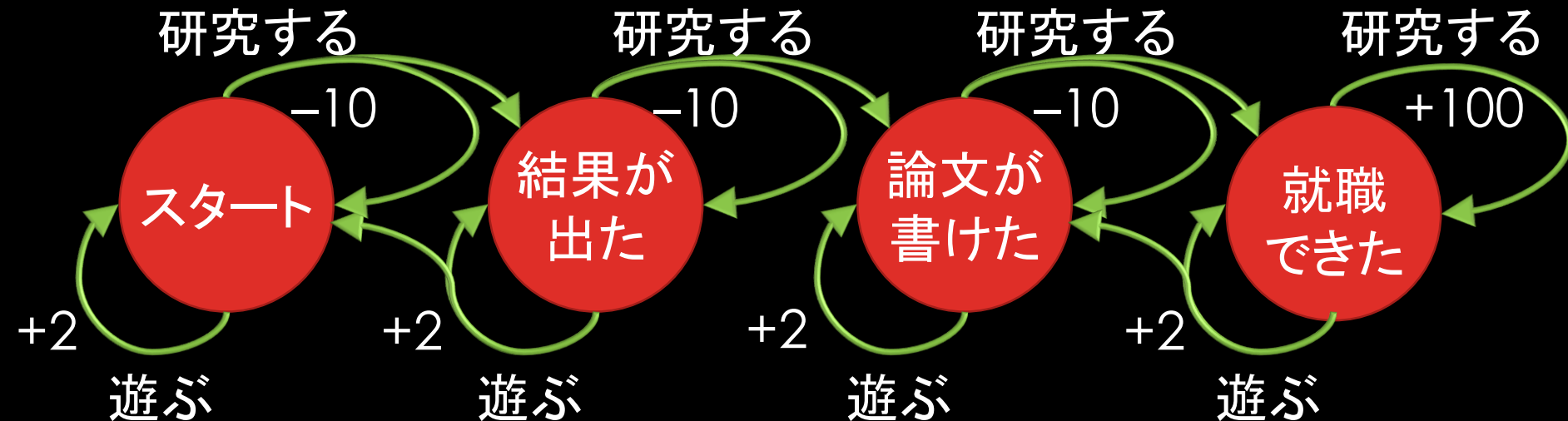
# はじめに

- 「状態」が入る
  - 行動によって報酬が与えられるだけでなく、環境の変化も起こす
  - 環境について知らない=ある状態に到達する方法も、試行錯誤しなければわからない
- 遅延報酬の問題
  - ある行動は、そのときの報酬としては悪いが、後でよい報酬を得るために必要かもしれない
- どのように未知の環境を探索すればよいか？



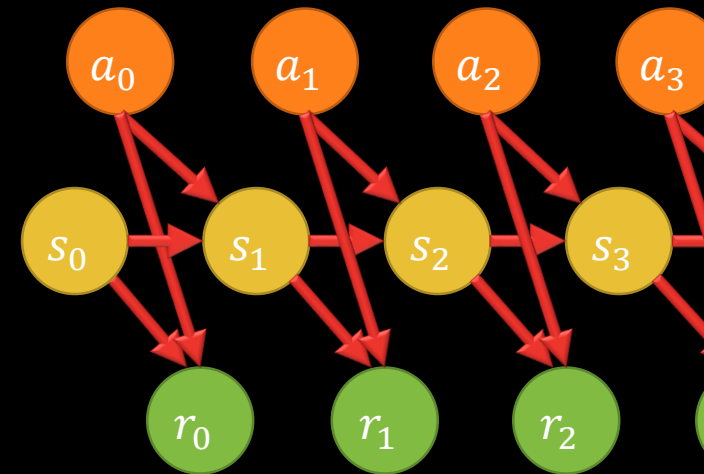
# 問題の例

- 行動: 「研究する」または「遊ぶ」
- エージェントは世界の構造を知らない
  - 即時報酬は、「研究する」より「遊ぶ」ほうが高い
  - 状態遷移は確率的で、その確率もわからない



# マルコフ決定過程 (MDP)

- 問題を  $\mathcal{P} = \langle S, A, T, R, \gamma, s_0 \rangle$  で定義する
  - $S$ : 状態の集合,  $A$ : 行動の集合
  - $T$ : 遷移確率関数
    - $T(s, a, s') = P(s' | s, a)$  : 状態  $s$  で行動  $a$  をとった時に
    - 状態が  $s'$  になる確率
  - $R$ : 報酬関数
    - $R(s, a)$ : 状態  $s$  で行動  $a$  をとったときの報酬値
  - $\gamma$ : 割引率 ( $0 \leq \gamma < 1$ )
  - $s_0$ : 初期状態



# 問題の定義

- 探索するものは「方策(policy)」
  - ある場面でどの行動がよいかは、ほかの場面での行動にも依存するため
  - 決定的方策  $\pi: S \rightarrow A$  (状態から行動への写像)
  - 確率の方策  $\pi(s, a) = P(a|s)$
- 方策が決まると、時刻 $t$ の状態 $s_t$ と行動 $a_t$ の確率分布が決まるので、状態の価値が計算できる
$$V^\pi(s) = \mathbb{E}[R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \dots]$$
- 初期状態の価値を最大化する方策を見つけたい

# 世界の仕組みを知っている場合

- 問題設定  $\mathcal{P} = \langle S, A, T, R, \gamma, s_0 \rangle$  が既知の時に最適  
方策の計算法 = Dynamic Programming
  - 問題設定が既知なので、試行錯誤の必要がない
  - その意味では、強化「学習」ではない...が、  
すべての強化学習の元になるアルゴリズム

# Bellman Equation

- 価値関数が満たす方程式 (Bellman, 1952)
- 方策  $\pi$  から価値  $V$  が計算できる

状態  $s$  の価値  
[状態価値関数]

状態  $s$  で方策  $\pi$  に従って  
行動を選んだときの価値

行動  $a$  をとった直後の報酬

$$V^\pi(s) = Q^\pi(s, \pi(s))$$

$$Q^\pi(s, a) = \sum_{s'} T(s, a, s') (R(s, a) + \gamma \cdot V^\pi(s'))$$

状態  $s$  で行動  $a$  をとる価値  
[行動価値関数]

次状態  $s'$  に関する期待値

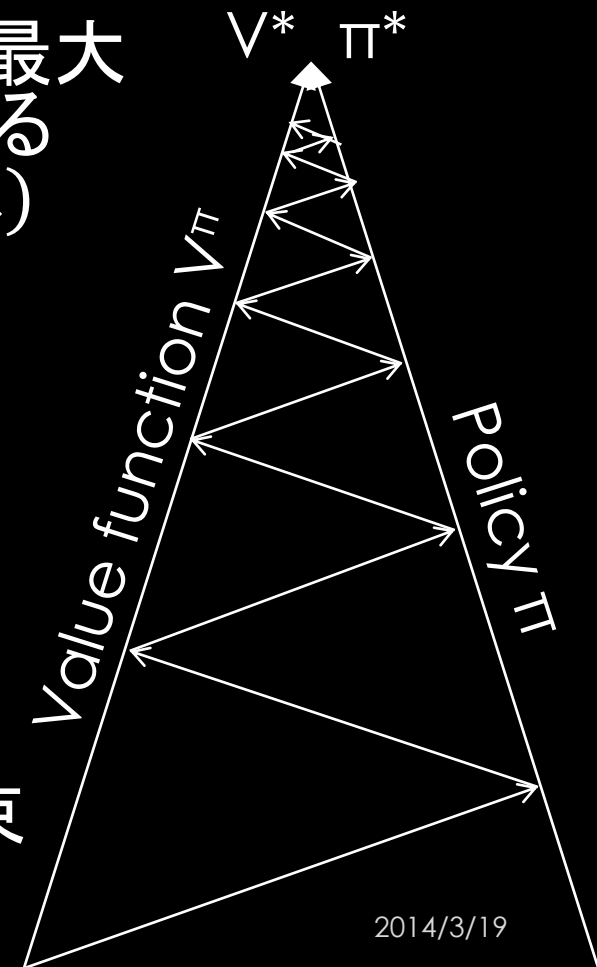
次状態  $s'$  の価値

# Dynamic Programming (DP)

- 価値関数 $V$ を固定すると、その上で最大の価値を得る方策 $\pi^V$ は簡単に求まる

$$\pi^V(s) := \underset{a}{\operatorname{argmax}} Q(s, a)$$

- 定理 (see Sutton&Barto 1998)
  - 任意の価値関数 $V$ に対して
 
$$V^{\pi^V}(s) \geq V(s)$$
  - 等号成立は、 $V$ が最適な価値関数 $V^*$ と等しいとき、またそのときに限る
  - →必ずglobal optimum に収束



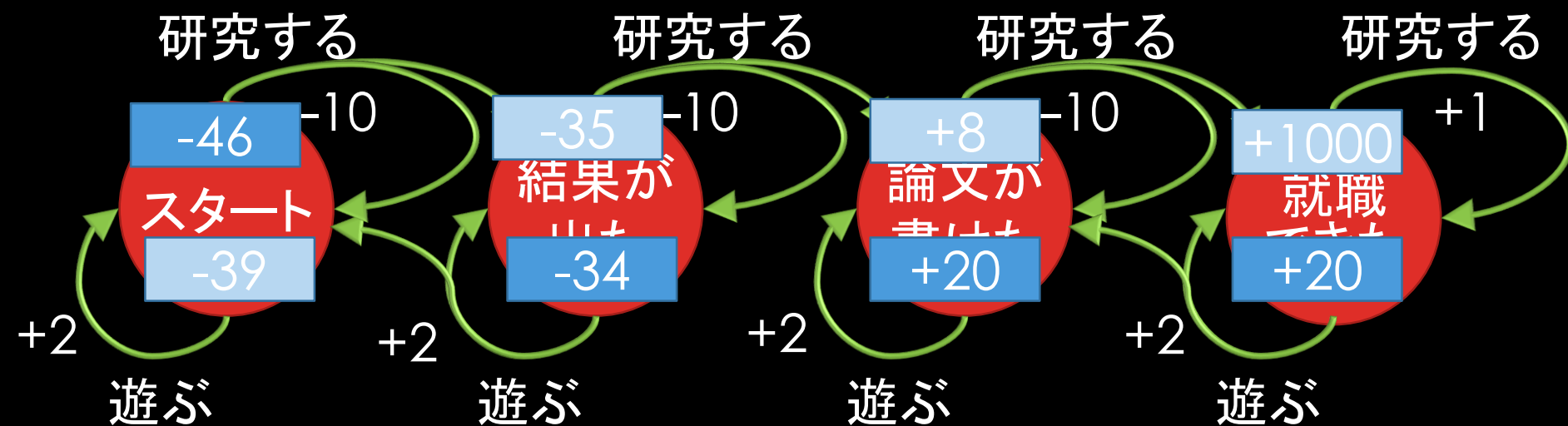
# Dynamic Programming の手法

- 価値反復 (Value Iteration):  $V$  を更新してゆく
  - $V \leftarrow V^{\pi^V}$
  - 変化が $\delta$ 以下になれば終了
  - $\pi$  を保持する必要がない ( $V$ 上の max 演算だけ)
- 方策反復 (Policy Iteration):  $\pi$  を更新していく
  - 1. 全状態について  $V^{\pi}(s)$  を計算する (方策評価)
  - 2.  $V^{\pi}$  上で最適となるように  $\pi$  を更新 (方策改善)
  - $\pi$  が変化しなくなれば終了



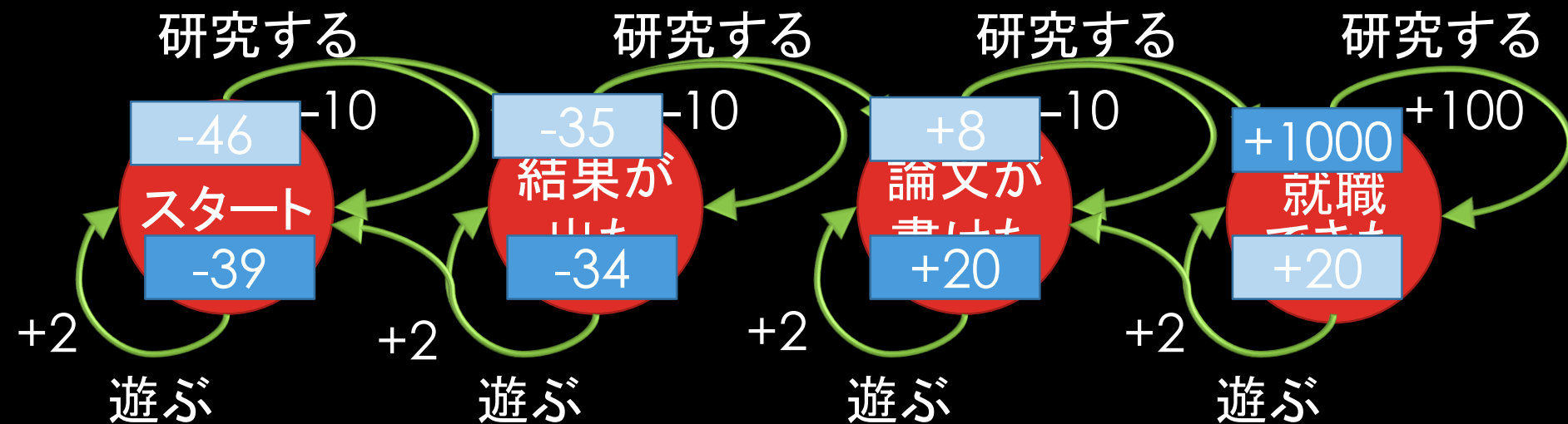
# Dynamic Programming の例

- 「結果が出るまで研究するが、それ以外は遊ぶ」  
方策からスタート (割引率=0.9)、方策反復法



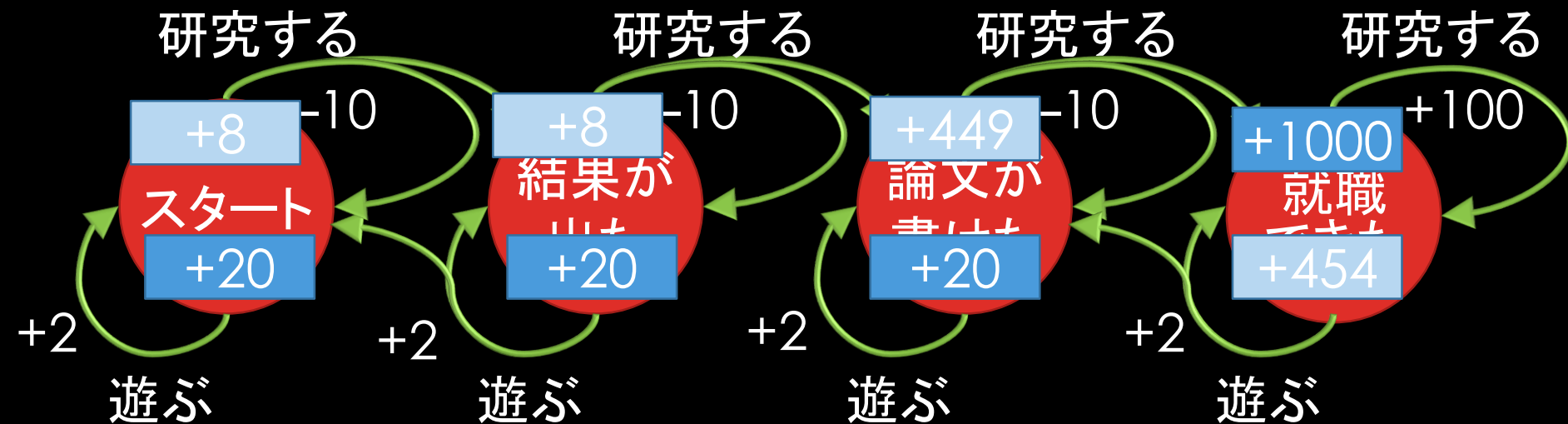
# Dynamic Programming の例

- 「結果が出るまで研究するが、それ以外は遊ぶ」  
方策からスタート (割引率=0.9)、方策反復法



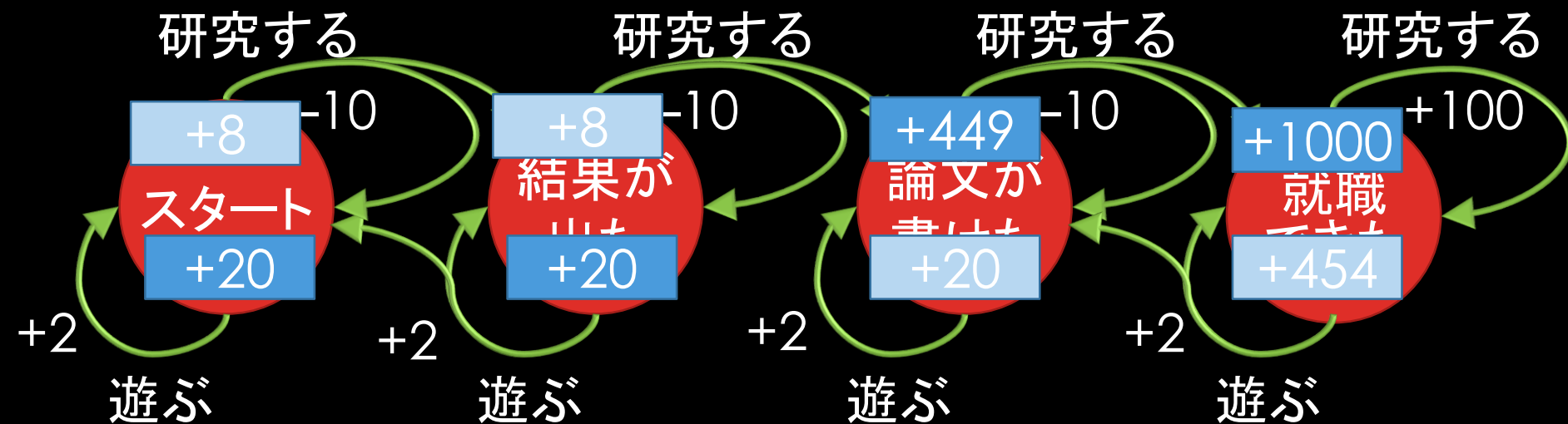
# Dynamic Programming の例

- 「結果が出るまで研究するが、それ以外は遊ぶ」  
方策からスタート (割引率=0.9)、方策反復法



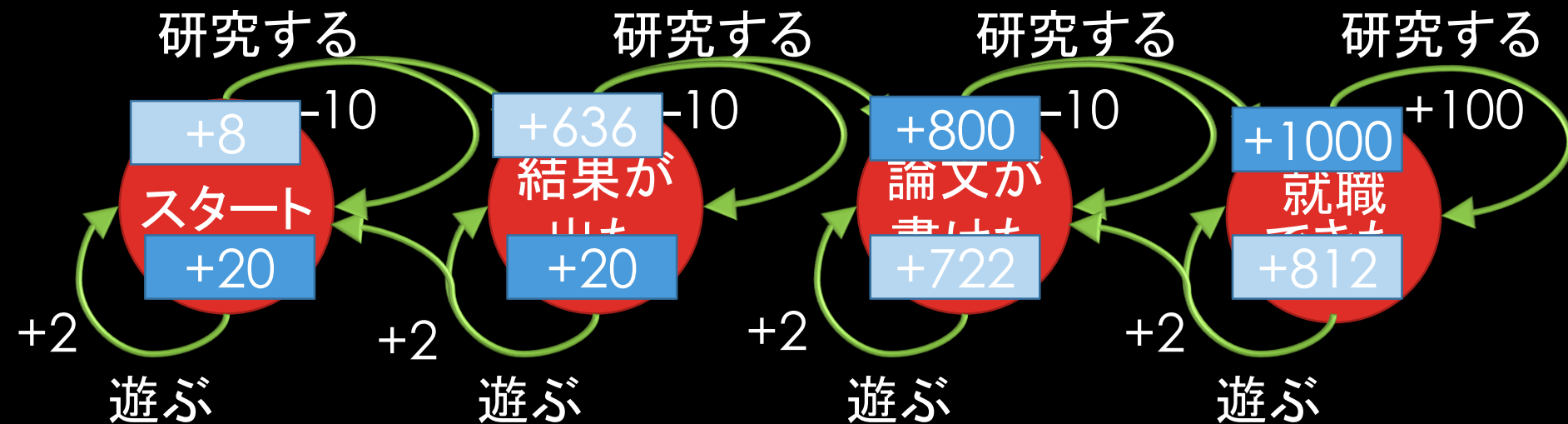
# Dynamic Programming の例

- 「結果が出るまで研究するが、それ以外は遊ぶ」  
方策からスタート (割引率=0.9)、方策反復法



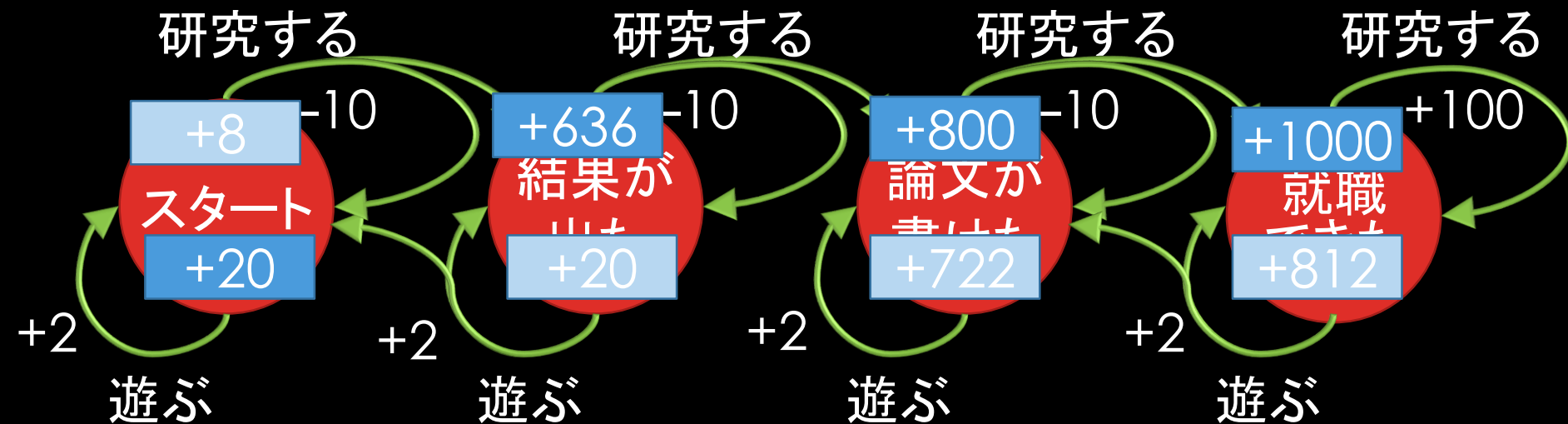
# Dynamic Programming の例

- 「結果が出るまで研究するが、それ以外は遊ぶ」  
方策からスタート (割引率=0.9)、方策反復法



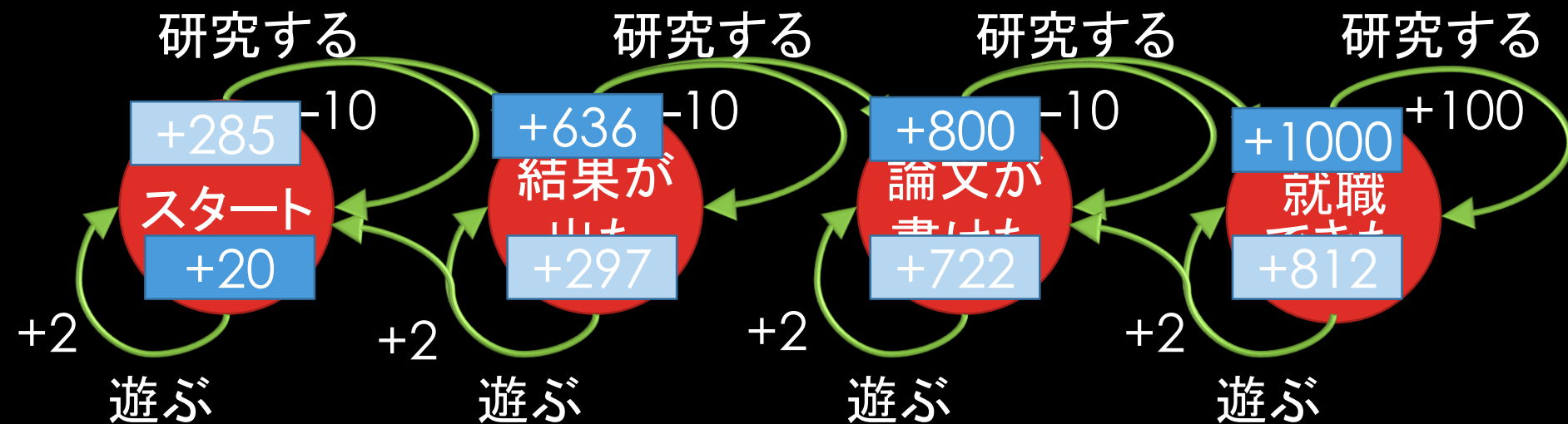
# Dynamic Programming の例

- 「結果が出るまで研究するが、それ以外は遊ぶ」  
方策からスタート (割引率=0.9)、方策反復法



# Dynamic Programming の例

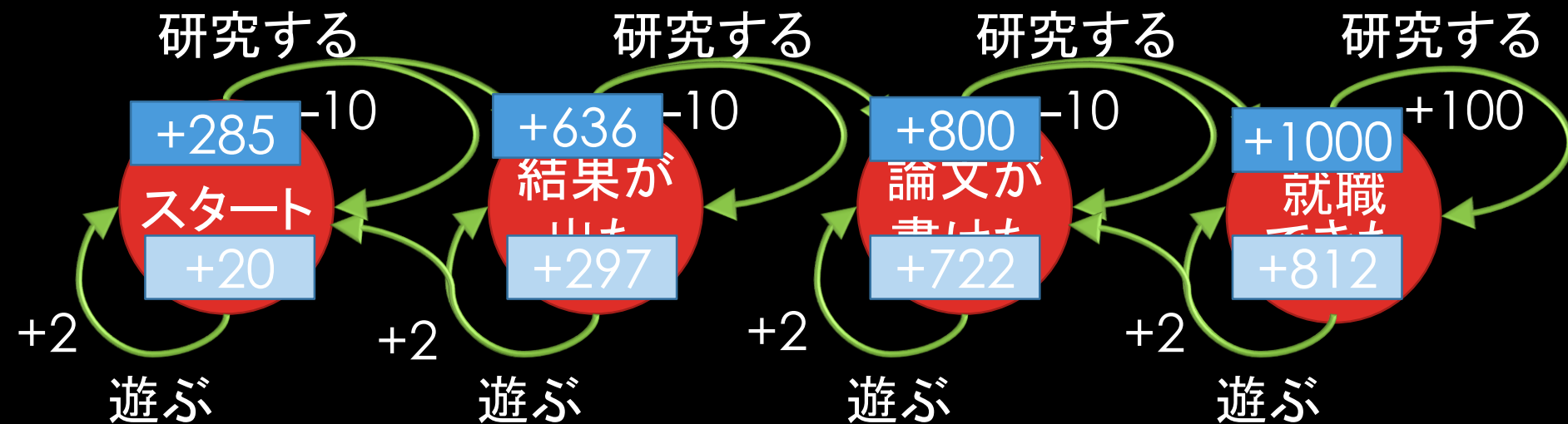
- 「結果が出るまで研究するが、それ以外は遊ぶ」  
方策からスタート (割引率=0.9)、方策反復法





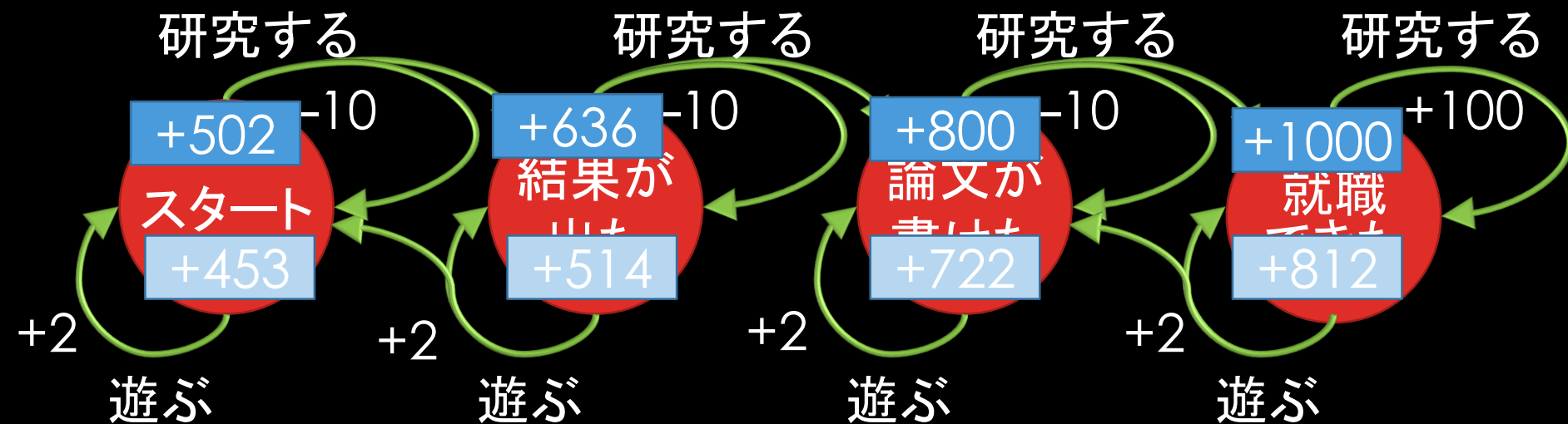
# Dynamic Programming の例

- 「結果が出るまで研究するが、それ以外は遊ぶ」  
方策からスタート (割引率=0.9)、方策反復法



# Dynamic Programming の例

- 「結果が出るまで研究するが、それ以外は遊ぶ」  
方策からスタート (割引率=0.9)、方策反復法



# 世界を知らない場合

- 遷移行列、報酬行列に関する情報がない
  - 遊んでいれば就職できるかもしれない...
- 情報を更新しながら行動を繰り返す (=試行錯誤)
  - 観測できるのは行動とその結果のみ
  - 問題は、この方法で十分な情報が収集できるか  
= 適切に問題が探索されるかどうか
- 収集した情報をもとに更新する部分は DP と同様
  - 価値反復法/方策反復法
  - 非同期的・確率的な更新となることが多い

# Q-learning (Watkins, 1989)

- 行動反復法を確率更新で求める手法
  - 状態 $s$ で行動 $a$ の結果、報酬 $r$ が得られ状態 $s'$ になったら
$$Q(s, a) \leftarrow Q(s, a) + \alpha(r + \gamma \max_{a'} Q(s', a') - Q(s, a))$$
- 学習者側は、 $Q(s, a)$ 表の更新だけで学習ができる
  - 環境モデルを学習する必要がない(モデルフリー)
- 全  $(s, a)$  ペアが無限回更新されるなら、最適価値関数への収束は保証される(Watkins&Dayan, 1992)
  - 適切に探索されることが必要条件、たとえば  $\epsilon$ -Greedy で行動する

# 強化学習と大脳生理学との関係

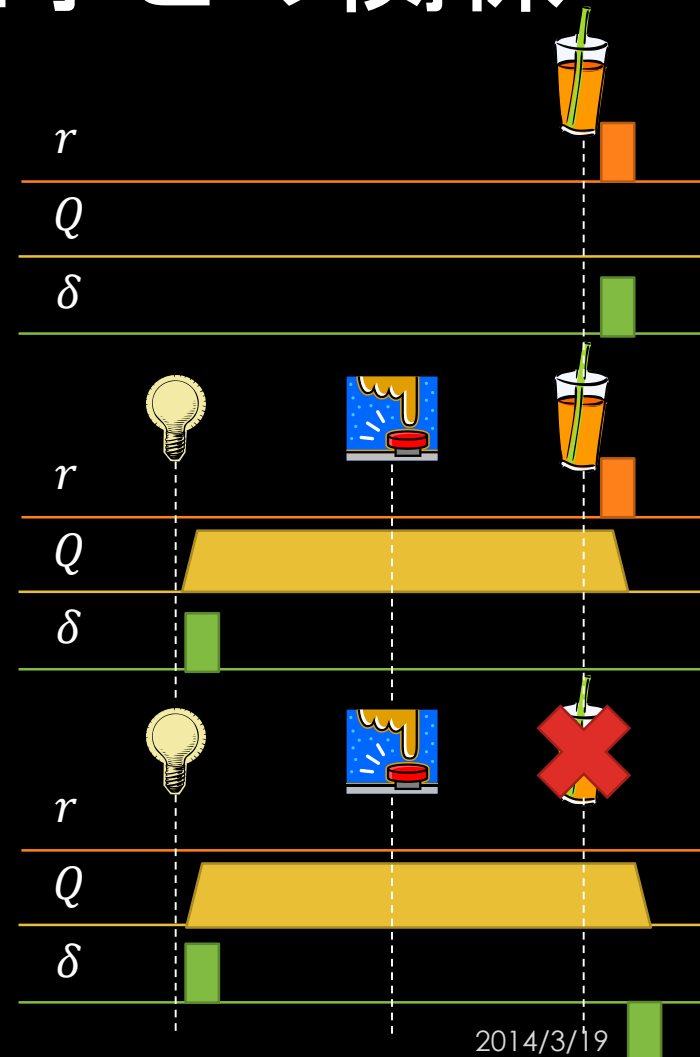
- $Q$  関数の更新量  $\propto$  TD error  

$$r + \gamma \max_{a'} Q(s', a') - Q(s, a)$$

状態  $s$  で行動  $a$  をとった  
結果がわかったときに  
計算できる価値

行動をとる  
前に思っ  
ていた価値

- サル大脳基底核のドーパミン  
ニューロンの活動が、TD error  
と一致 (Schultz 1997)

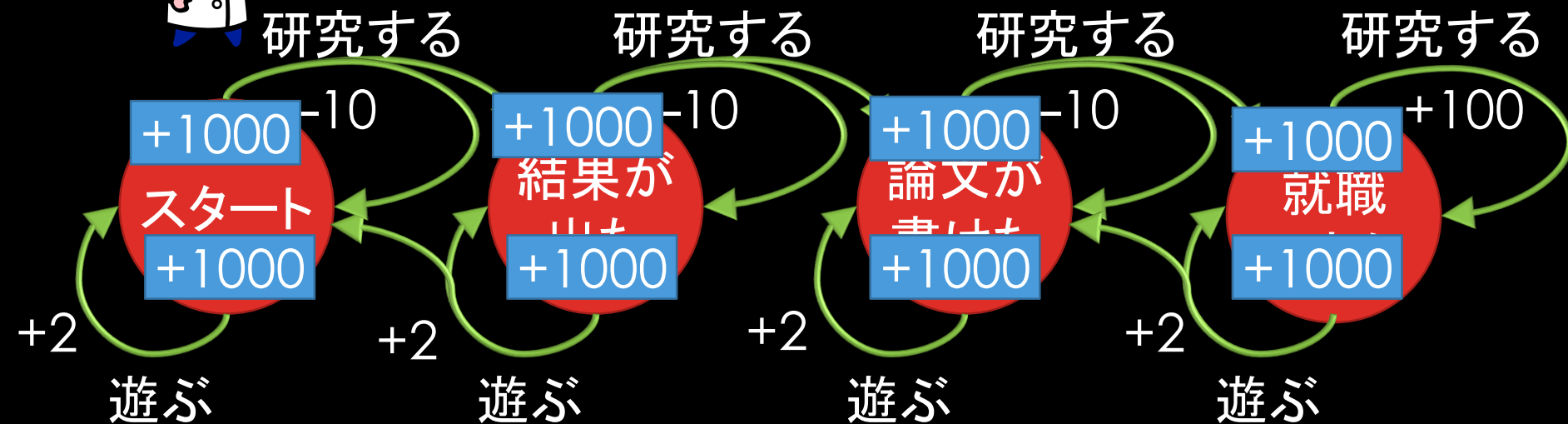


# 楽観的初期価値法

- Q-Learningなどのモデルフリー学習手法で、価値関数の初期値として、大きな値を設定しておくというヒューリスティクス
  - 経験の少ない行動の価値を楽観的に見積もることに相当
- 実装は非常に簡単だが、最適性は保障されない
  - 結局 $\epsilon$ -Greedyなどと組み合わせる必要がある

# Q-Learningの例

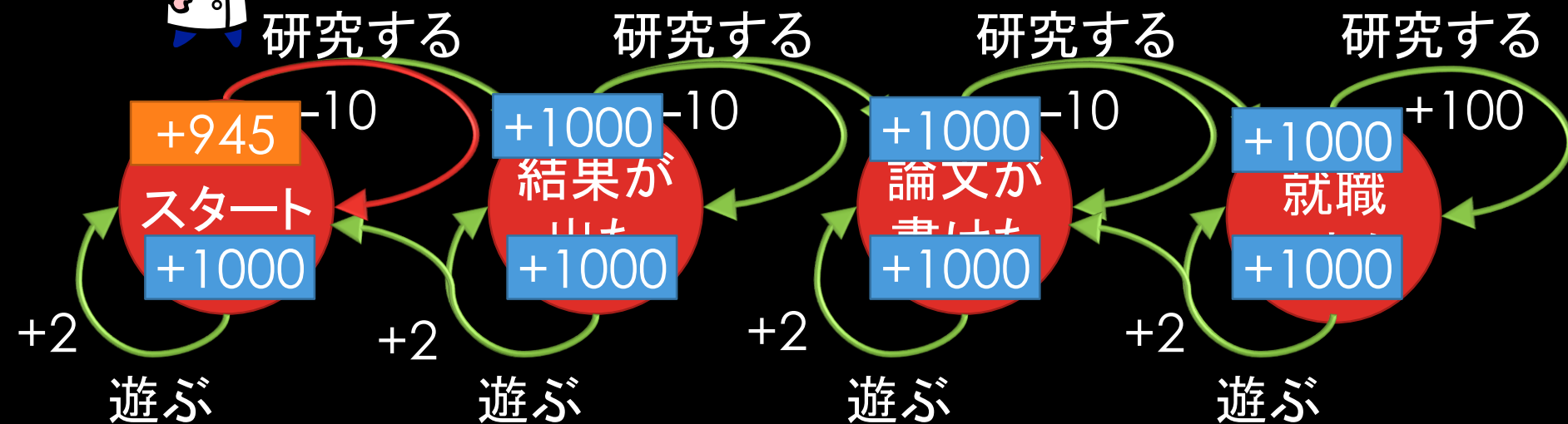
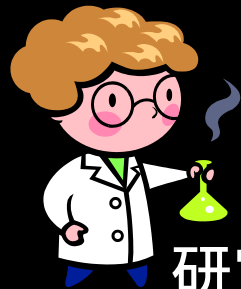
- 割引率 0.9, 学習率 0.5 の場合





# Q-Learningの例

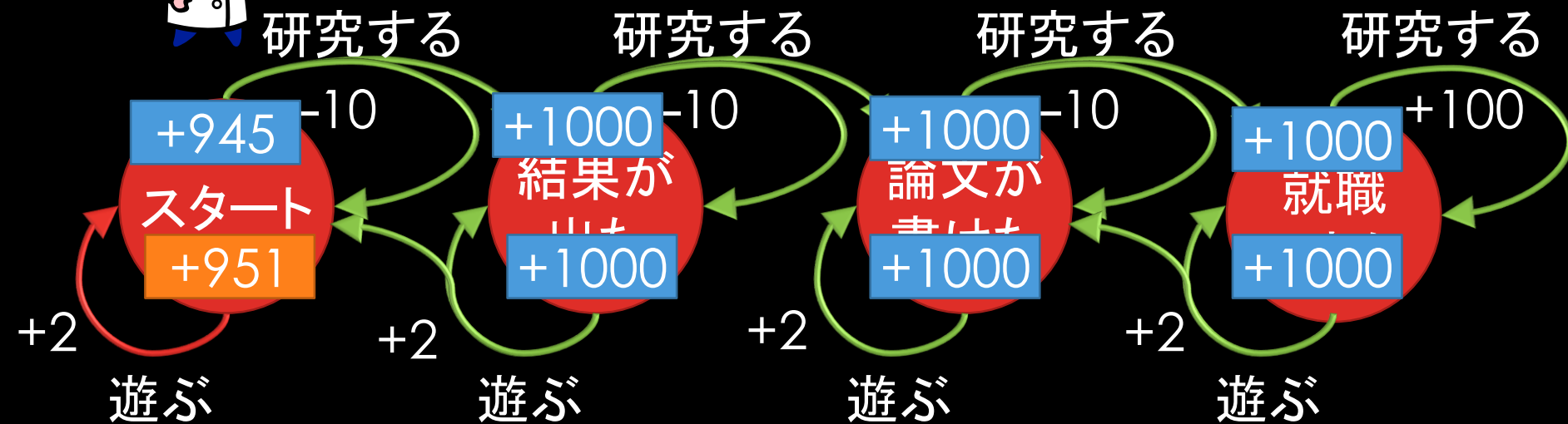
- 時刻1: 研究した。結果がでなかった





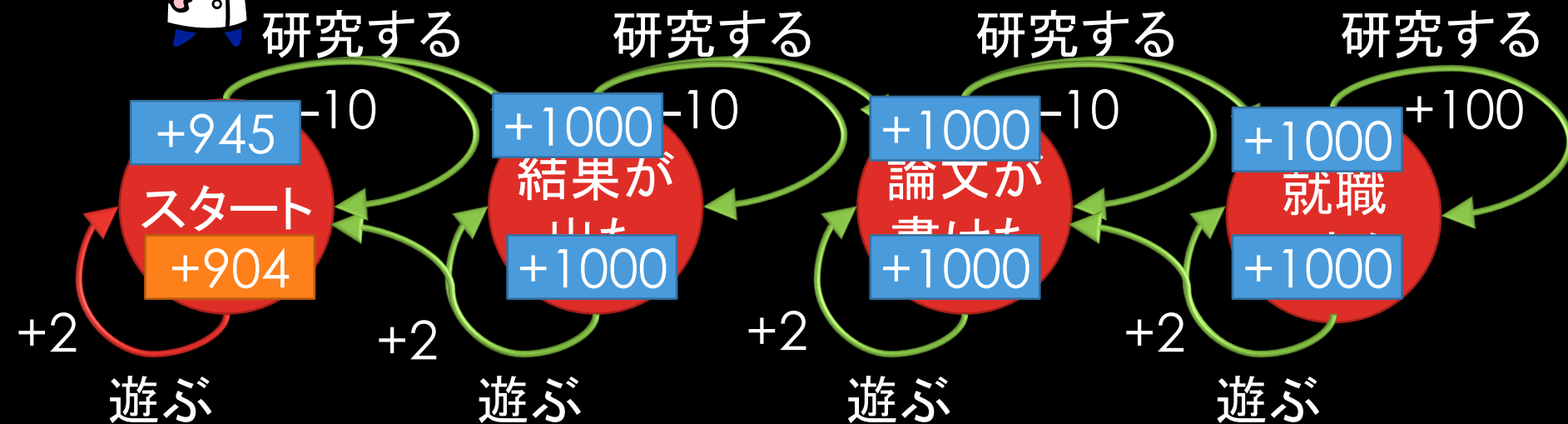
# Q-Learningの例

- 時刻2: 遊んだ。



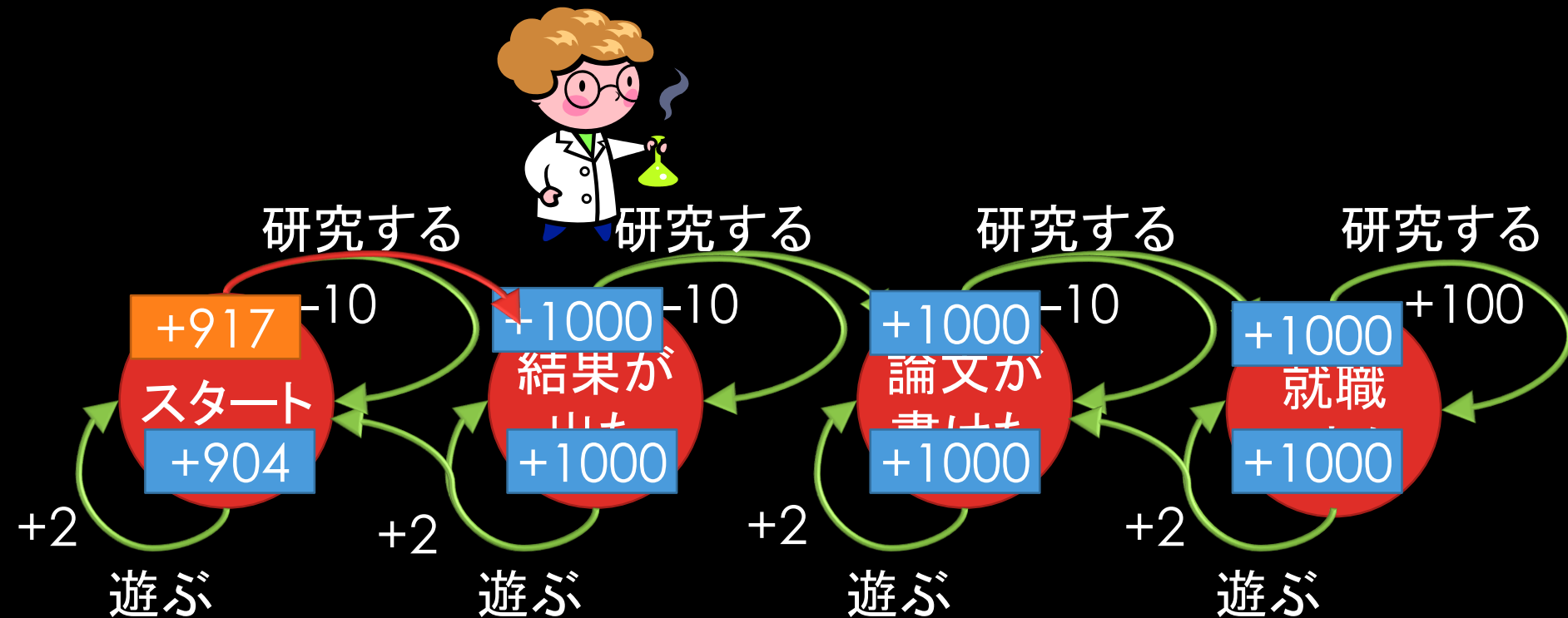
# Q-Learningの例

- 時刻3: 遊んだ。



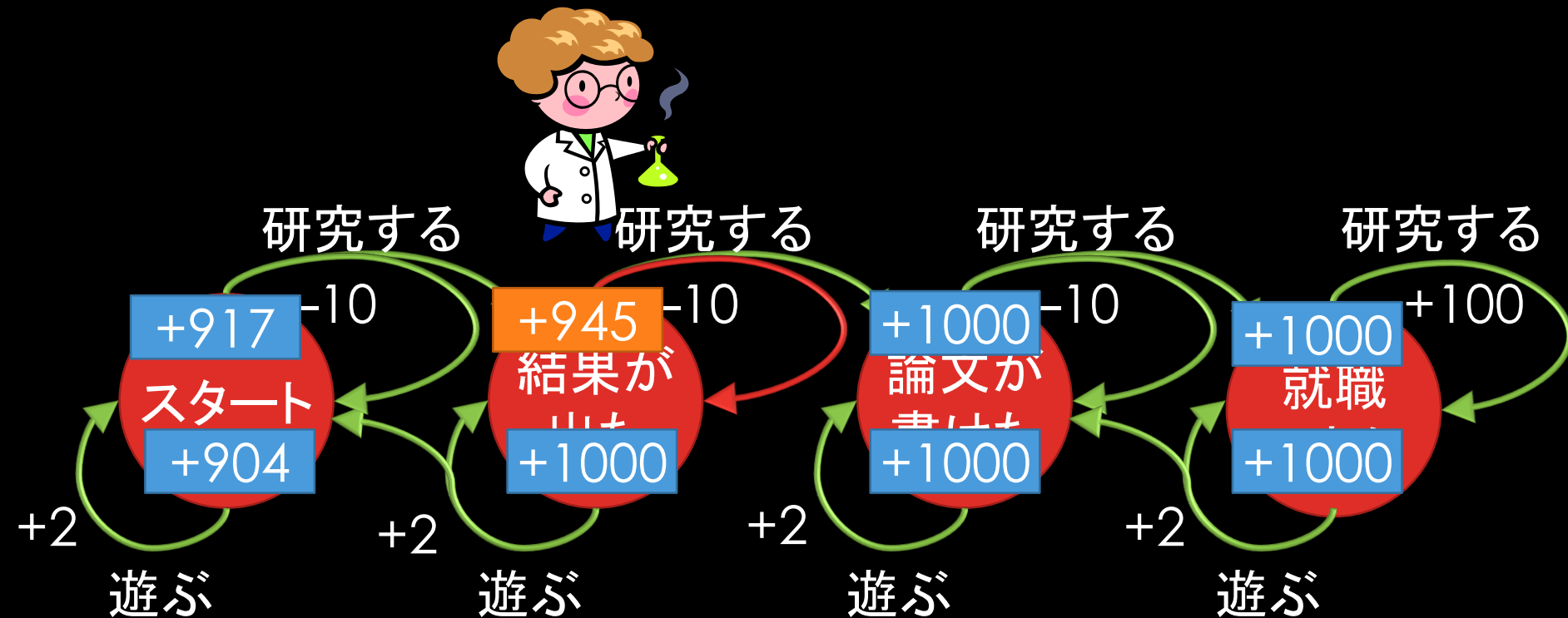
# Q-Learningの例

- 時刻4: 研究した。何か結果が出た



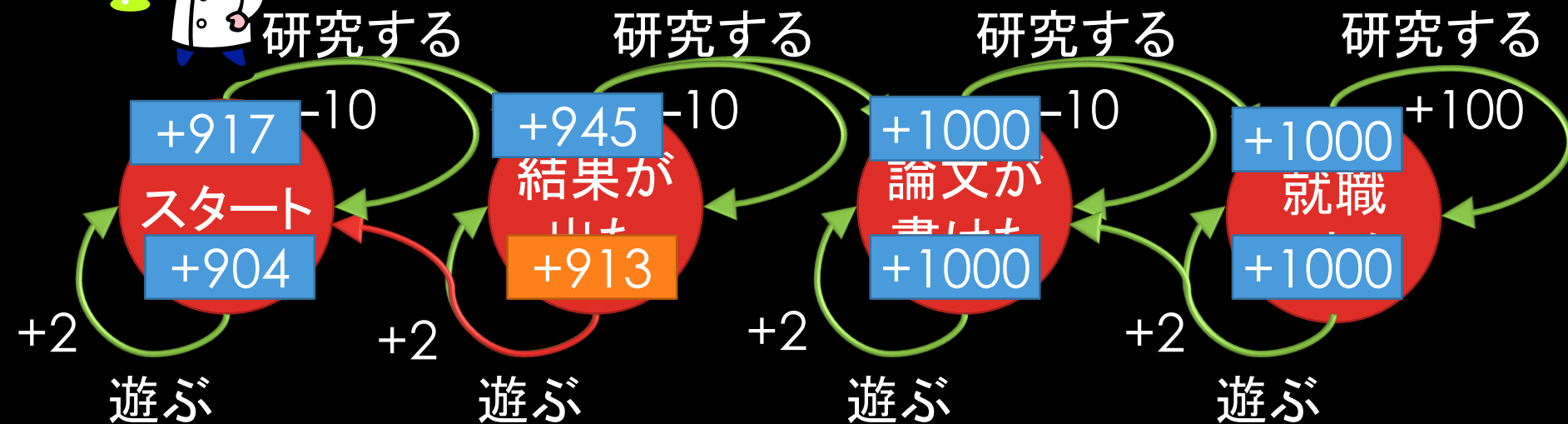
# Q-Learningの例

- 時刻5: 研究した。論文が書けなかった



# Q-Learningの例

- 時刻6: 遊んだ。  
他チームの論文が出て、結果が無意味になった



# 学習コストの評価指標

- サンプル複雑性 (Kearns&Singh 2002)
  - 最適方策から価値が  $\varepsilon \times \max |R|$  以上劣るような方策に基づいて行動する回数
  - PAC-MDP:  $(\frac{1}{\varepsilon}, \frac{1}{\delta}, \frac{1}{1-\gamma}, |S|, |A|)$  の多項式で表される上界に、確率  $1 - \delta$  でサンプル複雑性が抑えられる、というアルゴリズムの性質
- regret 上界 (Fiechter 1994)
  - 最適方策の報酬和の期待値と比べてどれだけ損をするか

# Q-learningの学習コスト

- 通常のQ-learningでは、PAC-MDPの証明はない
- Delayed Q-learning (Strehl+ 2006)
  - $(s, a)$  ペアをk回体験するごとに  $Q(s, a)$  を更新
  - 最初のk回以前は  $Q(s, a)$  は楽観的初期値
    - 本当はもう少し複雑(証明を成立させるためのトリック)
  - サンプル複雑性  $\tilde{O}\left(\frac{|S||A|}{\varepsilon^4(1-\gamma)^8}\right)$  が証明されている



# モデルベース強化学習

- モデルフリー強化学習は、実装はシンプルだが、学習コストは大きい
- 試行錯誤の結果から、環境のモデルを構成することで、学習コストを減らすことが可能になる  
→ モデルベース強化学習
- モデルの不確実性を考慮しながら次の行動を決定することで、探索と利用のトレードオフも解ける



# モデルベース区間推定 (MBIE)

(Strehl&Littman, 2004)

- これまでの履歴からモデルを作り、DP で解く
- ただし、遷移確率  $T(s, a, s')$  に関しては、これまで試した回数  $n(s, a)$  に応じて信頼区間を計算し、そのなかでもっとも「都合のよい」ものを選ぶ

$$Q(s, a) = \hat{R}(s, a) + \max_{\tilde{T} \in CI} \gamma \sum_{s'} \tilde{T}(s, a, s') \max_{a'} Q(s', a')$$

- 楽観的に見積もるため、「エデンの園」状態を仮定
  - 最大の報酬が得られることになっているが、実際は決して到達しない、仮想的状態
- サンプル複雑性  $\tilde{O}\left(\frac{|S||A|}{\varepsilon^3(1-\gamma)^6}\right)$

# 強化学習へのベイズ主義アプローチ

- 強化学習が難しいのは、環境が未知であるため
- ベイズ主義的アプローチでは、不確かさを確率分布を利用して表現する

→もし、ベイズ主義的アプローチに基づいて、環境モデルの事前分布が与えられるなら、強化学習の問題がより見通しよくなるのでは？

# ベイズ強化学習モデル

- 環境を、 $k$  次元のパラメータベクトル  $\theta \in \mathbb{R}^k$  で決まるMDP  $\mathcal{P}_\theta$  として定義する
- パラメータの事前分布  $p_0(\theta)$  が与えられている
- 問題は、事前分布の上での割引報酬和の期待値を最大化するような、行動列の決め方を考えること
$$\mathbb{E}_{p(\theta)}[r_0 + \gamma r_1 + \gamma^2 r_2 + \dots]$$

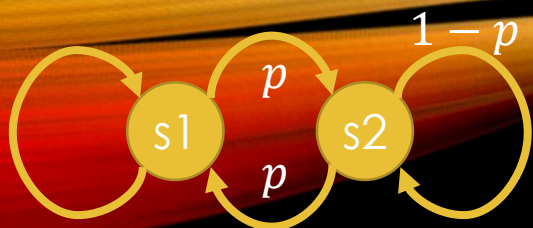
# ベイズ適応的マルコフ決定過程 (BAMDP) (Duff, 2002)

- パラメータ化されたMDPの集合をもとに、  
「拡張した部分観測MDP」を構成する
- $\mathcal{P}_\theta = \langle S, A, T_\theta, R_\theta, \gamma, s_0 \rangle \rightarrow \mathcal{P}^* = \langle S^*, A, T^*, R^*, \gamma, s_0^* \rangle$ 
  - $S^* = \mathbb{R}^k \times S$  — パラメータ空間と元の状態空間との直積
  - $T^*([\theta, s], a, [\theta', s']) = T_\theta(s, a, s') \cdot \delta(\theta, \theta')$
  - $R^*([\theta, s], a) = R_\theta(s, a)$
- エージェントは、 $\theta$ を観測できない  
→ 部分観測問題

クロネッカーのデルタ ( $\theta$  が  
変化しないという前提を表す)

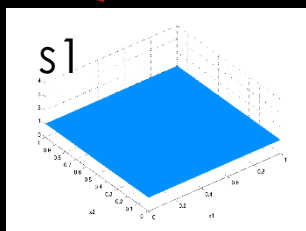
# BAMDPの何が重要か

- モデルだけから (=実際の試行錯誤をする前に) 最適解が求まる!
  - 拡張後の POMDP をDynamic Programming で解けばよい
  - 元の未知のMDPにおいて、出会いうるすべての環境とその確率を列挙して、各々の状況における最適行動を計算することに相当



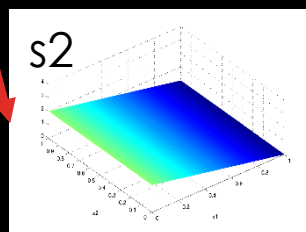
$1-p$  Action A

環境の状態と  
パラメータ値の  
事前分布のペア



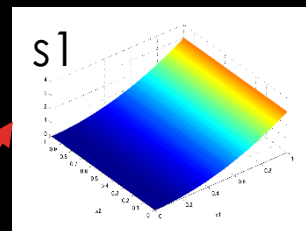
A

事前分布から  
遷移確率の期待  
値が計算できる

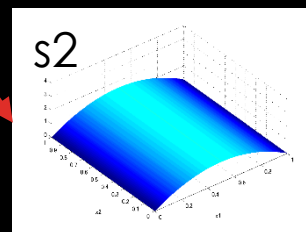


B

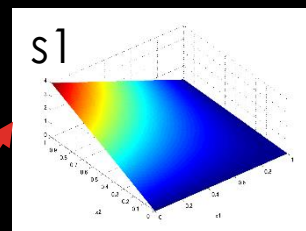
各遷移が起きた  
場合の事後分布  
も計算できる



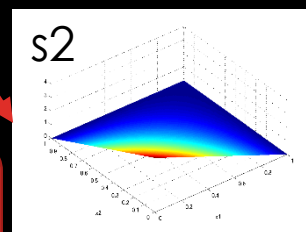
A



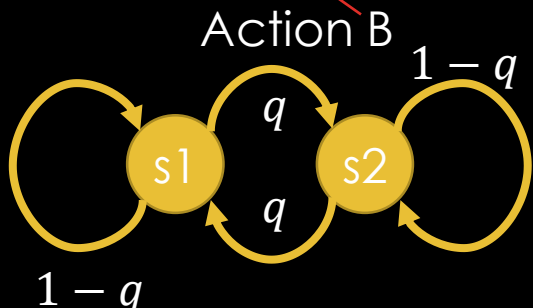
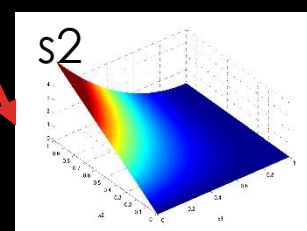
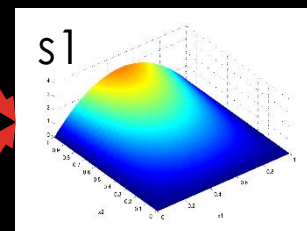
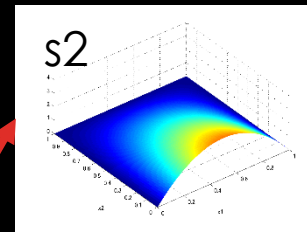
B



A



B



Action B

$1-q$

牧野 貴樹: 強化学習をベイズで理解する

# BAMDPによる強化学習の再定義

- BAMDP解=「探索コスト最小化」の解析解
  - 強化学習はすべてBAMDPに帰着させればよい
- おわり



# BAMDPによる強化学習の再定義

- BAMDP解=「探索コスト最小化」の解析解
  - 強化学習はすべてBAMDPに帰着させればよい
- ~~おわり~~
- BAMDPは、計算コストが intractable
  - 計算量が時間地平線の距離Hに対して指数関数的
  - Hが有限の場合、サンプル複雑性の意味で最適な探索とはならない
- BAMDP解をどう近似するかが重要

# ベイズ強化学習の近似戦略(1)

- BAMDPの近似問題を解く
  - BEETLE (Poupart+ 2006): 変形後のPOMDPを直接近似する
  - BOLT (Araya-López+ 2012): 探索回数が少ない状態に適切なボーナスを与えたアルゴリズム  
BAMDP に漸近すること(PAC-BAMDP) を証明
    - ベイズ的に遷移確率の信頼区間を計算し、MBIEと同様に解く
  - MC-BRL (Wang+ 2012):  $k$ 個のMDPをサンプリングして、 $n$ 個の世界の上の離散分布で信念を近似する

# ベイズ強化学習の近似戦略(2)

- 事前分布からサンプリングして作ったMDPを解く
  - Bayesian DP (Strens 2000): 一定時間毎に事後分布からMDPを1つサンプリングして、そのMDPの最適解を実行する ( $\equiv$  Thompson Algorithm)
  - BOSS (Asmuth+ 2009):  $k$ 個のMDPをサンプリングして、「都合のよい世界を自由に選べる」エージェントの最適解を実行する

# ベイズ強化学習の近似戦略(3)

- モンテカルロ木探索法
  - POMDPの木構造探索をサンプリングで代用
  - FSSS (Walsh+, 2010): POMDP の枝の選択にUCB1に基づく手法ではなく、多項式回で最適探索に漸近する手法を利用
  - BFS3 (Asmuth&Littman 2011): BAMDP を FSSS で解く→PAC-BAMDP 性が保障できる

# ここまでのまとめ

- 状態のある強化学習では、遅延報酬があるため、環境の探索はより難しくなる
- 環境が既知の場合は効率的に解ける
  - Dynamic Programming (DP)
- Q-learning などのモデルフリー強化学習は、DPの確率更新に相当
- モデルのベイズ事前分布を考えれば、最適探索の解析解が定義できる (BAMDP)
  - どう効率的に近似するかに研究の余地がある

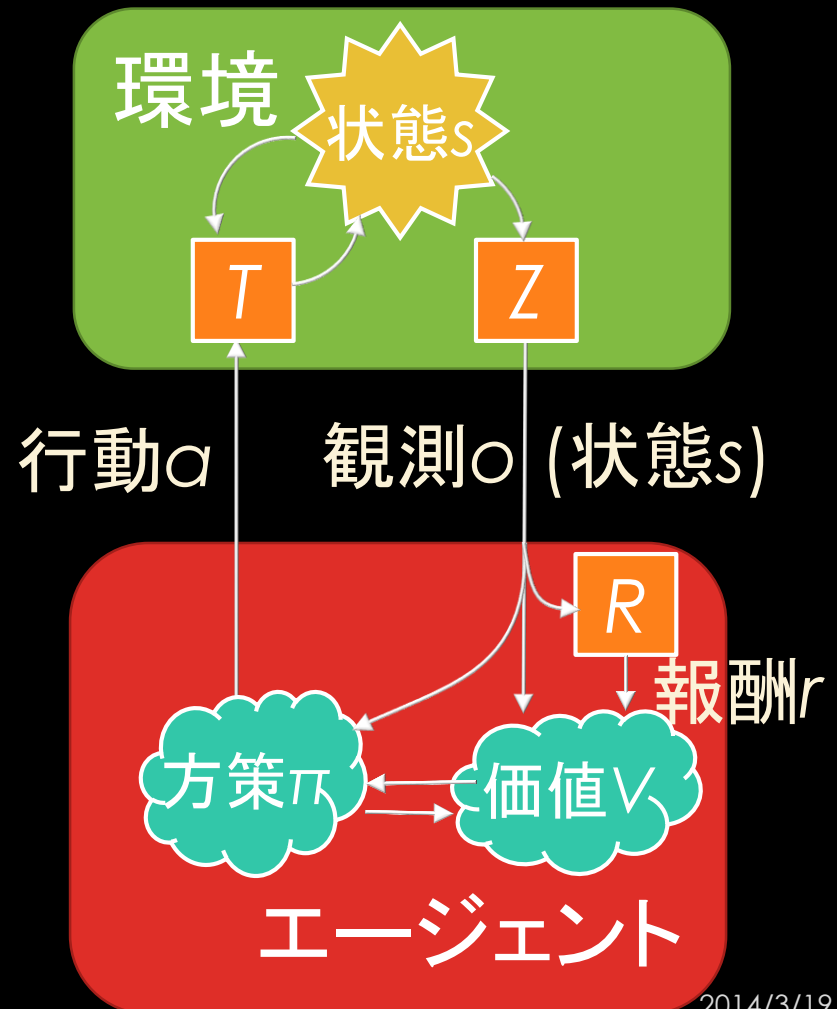
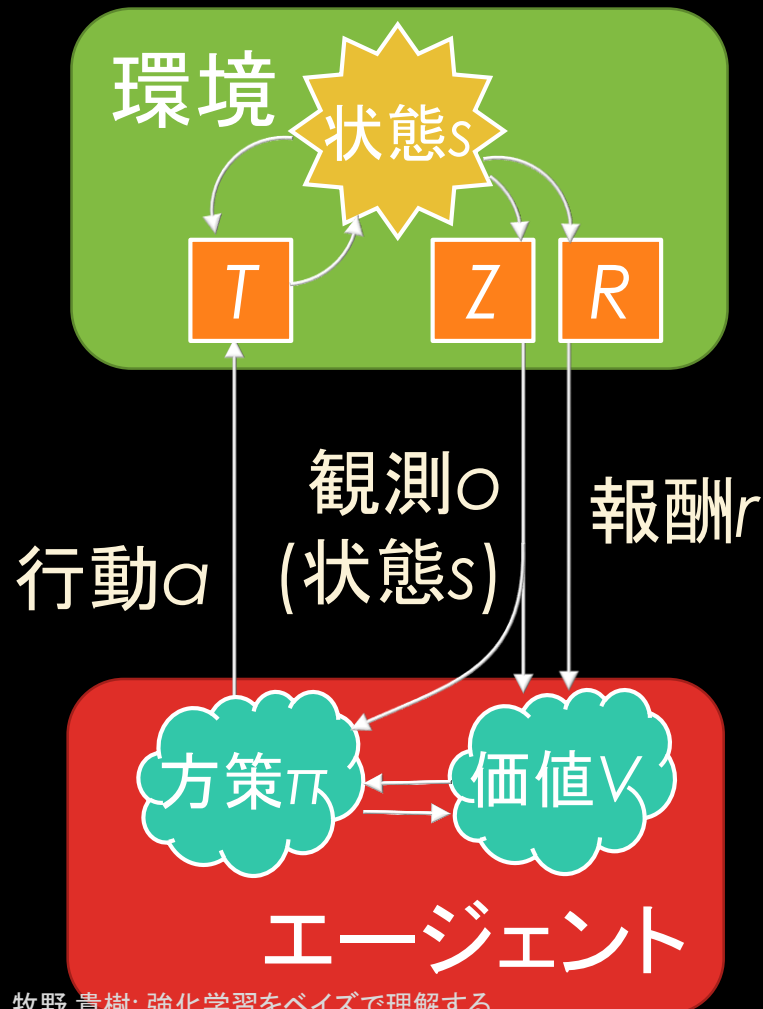
# 問題設定の拡張

# はじめに

- これまで、強化学習においては、報酬は環境から所与であると仮定していた
- 実際には、報酬をどう与えるか、が強化学習を適用するための本質的な問題である
  - 実際に受け取るものと効用関数との関係
- また、複雑な環境では、適切な環境知識も必要

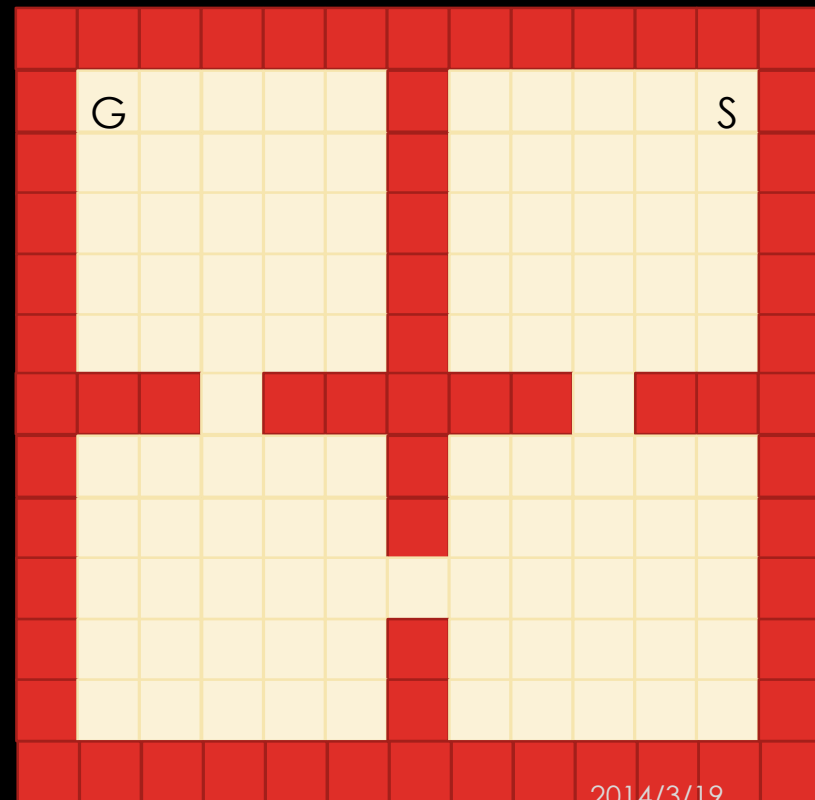


# 報酬はどこで作られるか？



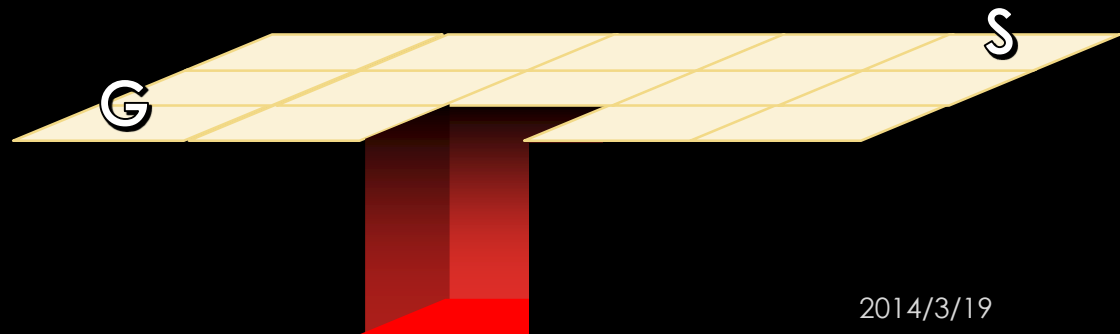
# 報酬設定による学習の高速化

- よくある設定: ゴールでは高い報酬、ゴール以外では一様の低い報酬  
→最適方策に収束するまで長い時間がかかる
- 適切なポテンシャルに応じた報酬設定があれば、大幅に高速化される  
(reward shaping)
- ヒトの場合、多くの報酬が後天的に獲得される



# 報酬だけでは学習できない事象

- “You can’t back up death” (Barto, 2009)
  - 穴に落ちると即死 → 強化学習できない
  - 落ちそうな場所に負の報酬があれば、穴に近づかないことが学習できる
  - その報酬は誰がどう設定するか？



# 複数の要求をどう報酬で表現するか

- 例: 車の運転を強化学習で獲得する場合
- 何を報酬にするか?
  - ともかく早く目的地に到着する
  - やさしい運転(加減速・車線変更の抑制)
  - 安全に運転する(十分な車間距離など)
  - 交通ルールの遵守(走行車線から追越禁止等)
- 一つ一つの項目も記述が難しいが、どう統合するかはもっと難しい

# 報酬設計に関する研究

- 探索を高速化するような報酬の生成方法の研究  
(Intrinsic motivation, Reward shaping, etc.)
- 多数の強化学習エージェントによるメタ学習  
(Genetic algorithm, 群強化学習 etc.)
- 徒弟学習: エキスパートの学習結果を利用する  
(Inverse Reinforcement Learning etc.)
- 今回は徒弟学習について少々

# エキスパートの行動を利用した タスク学習のアプローチ

- 教師つき学習によるタスク学習
  - 状態を入力、行動を出力とする教師つき学習にエキスパートの行動履歴を学習させる
  - 膨大なデータが必要になる
- 強化学習の学習データとして利用
  - すべての行動に関するデータが必要
  - エクスパートの報酬関数と違う場合、異なる行動が出力されてしまう
- 徒弟学習 (逆強化学習)
  - 未知の強化学習問題があるが、エキスパートはその問題の知識があり、最適解を実行していると仮定

類似度によって行動をコピー (理由は考えない)

## アプローチの比較

- ・ 教師つき学習
- ・ 強化学習
- ・ 徒弟学習



行動の模倣



試行錯誤による獲得



意図の模倣

環境を学習するために、  
すべてを試す

エキスパートの行動選択の  
背後にある知識を学習



真似してやらない  
(理由は知らない)

なんでしないのかな...  
よしやってみよう

## 人・モノ・場所の比較

- 教師つき学習
- 強化学習
- 徒弟学習



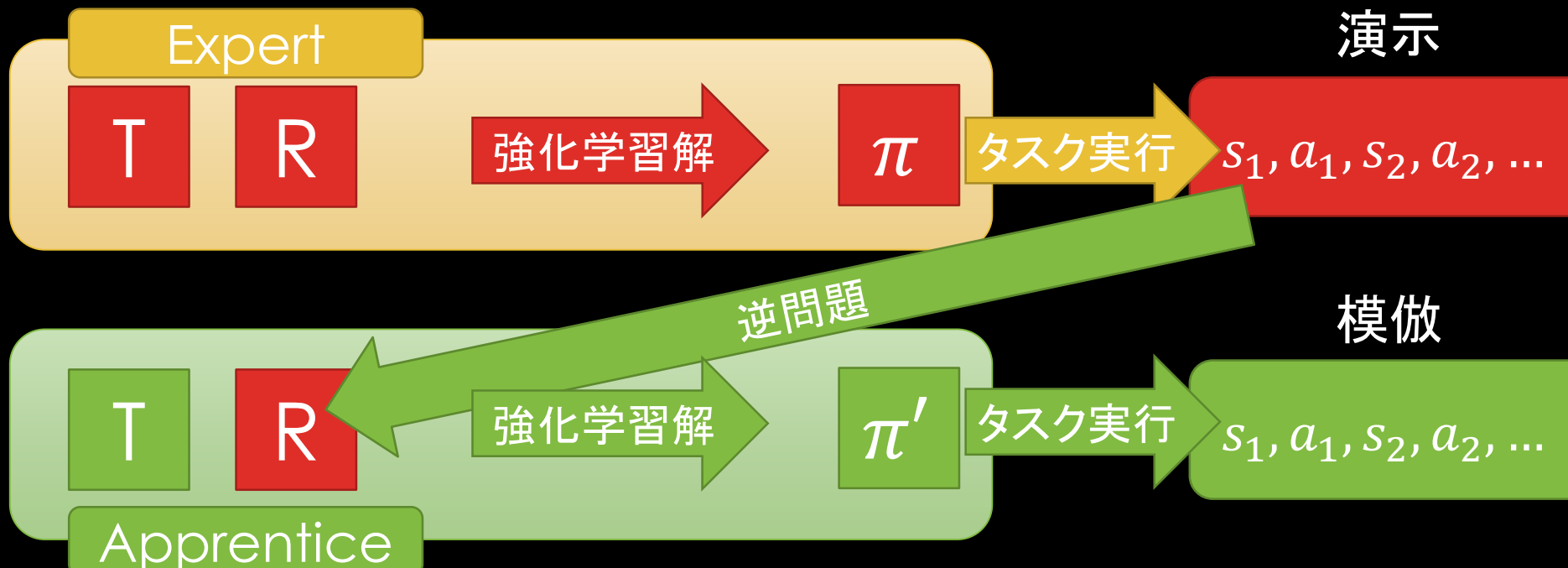
もしエキスパートが特定の場面で  
ある行動を避けていたら...

なにかやらない理由が  
あるのだろう、きっと

# Inverse Reinforcement Learning

(Abbeel&Ng, 2004)

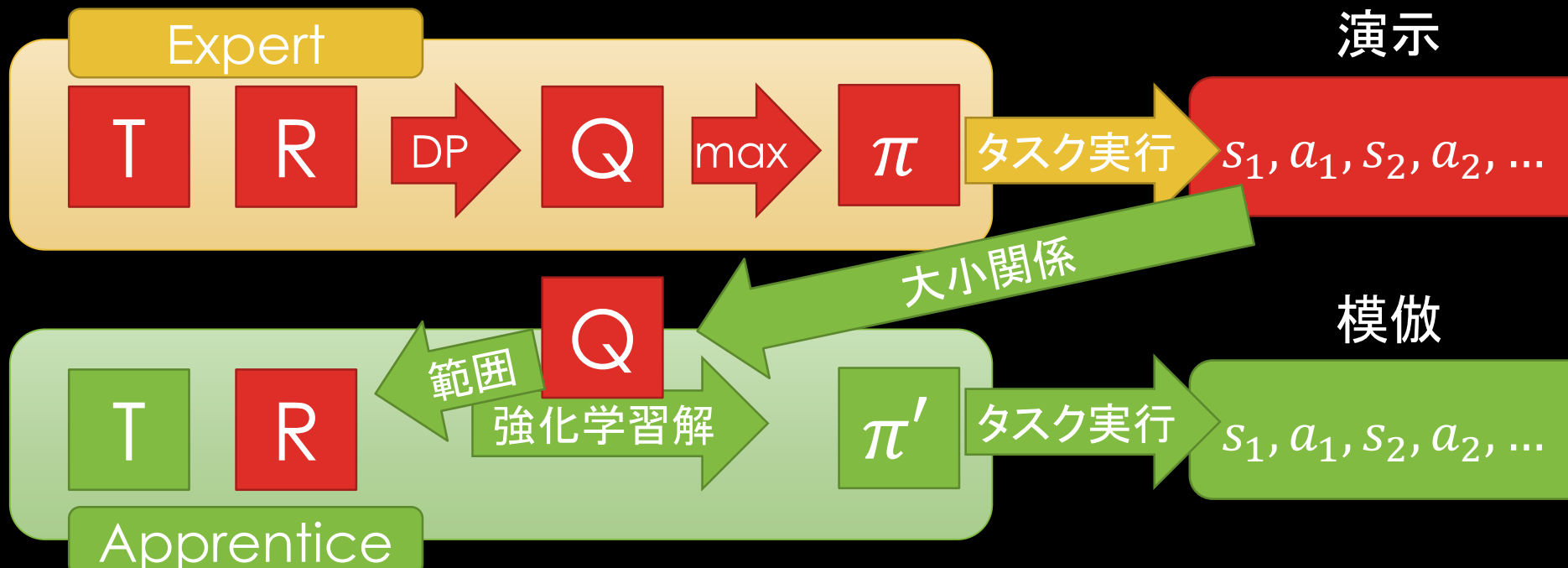
- 徒弟学習において、報酬関数を推定する問題設定
- Helicopter aerobatic airshow (Ng+, 2009)



# Inverse Reinforcement Learning

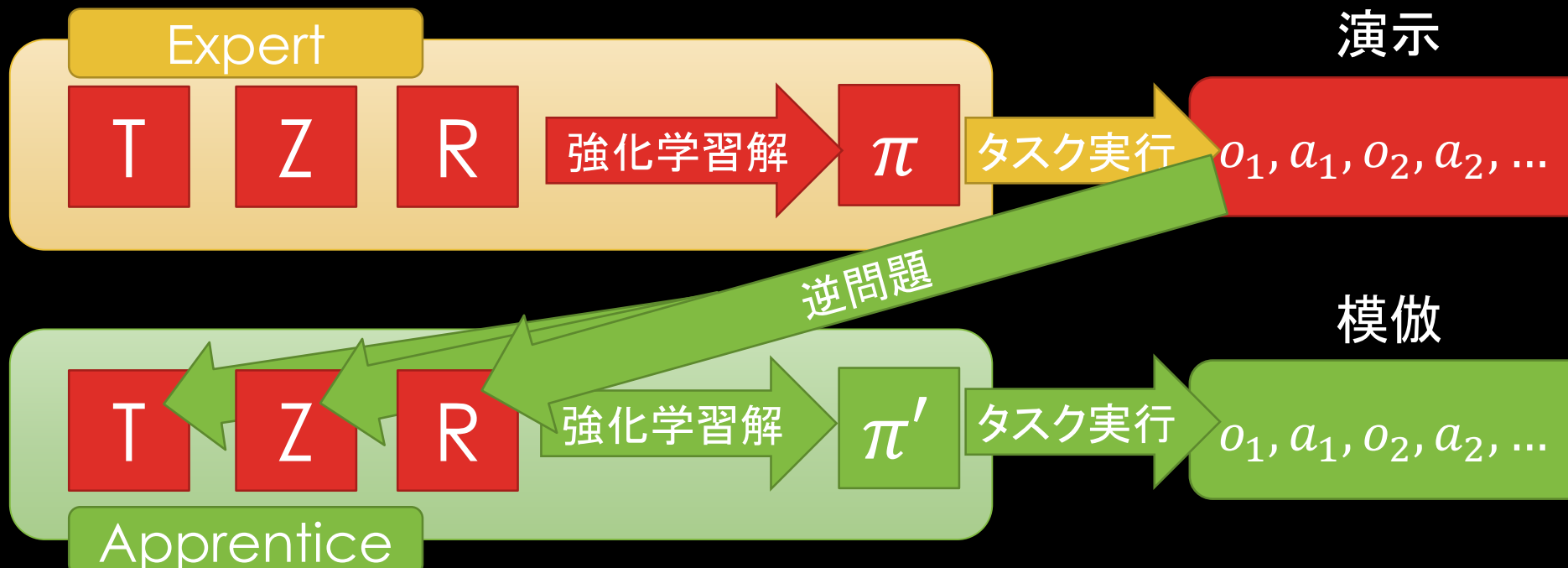
(Abbeel&Ng, 2004)

- 徒弟学習において、報酬関数を推定する問題設定
- Helicopter aerobatic airshow (Ng+, 2009)



# Apprenticeship Learning of Model Parameters (Makino&Takeuchi, 2012)

- 報酬関数だけではなく、部分観測環境の遷移関数・観測関数についても、エキスパートから学習したい



# モデルパラメータ推定の重要性

- 実用的な強化学習の問題は、状態が完全に観測できない (部分観測環境)
- 部分観測環境では、適切な環境モデルを記述することがそもそも困難である
  - 例: 対話システム...状態=ユーザーの心
- エキスパートの行動から、環境モデルの推定もできるのではないか?

# 環境モデルの徒弟学習(1)

- POMDPのパラメタライズ版  
 $\wp_{\theta} = \langle S, A, Z, T_{\theta}, O_{\theta}, b_{\theta}, R_{\theta}, \gamma \rangle$ 
  - $\theta$ : パラメータベクトル、事前分布  $p(\theta)$
  - 環境の真のパラメータ  $\theta^*$  は分からない
- 事後分布  $p(\theta|D)$  を演示  $D = (a_1 z_1 \cdots a_L z_L)$  より推定

$$\begin{array}{ccccc} p(\theta|D) & \propto & p(D|\theta) & p(\theta) \\ \text{posterior} & & \text{likelihood} & \text{prior} \end{array}$$

# 環境モデルの徒弟学習(2)

- 演示の尤度を分解する

$$\begin{aligned}
 & p(D|\boldsymbol{\theta}) \\
 &= p(a_1|\boldsymbol{\theta})p(z_1|\boldsymbol{\theta}, a_1)p(a_2|\boldsymbol{\theta}, a_1 z_1) \\
 &\quad \cdots p(z_L|\boldsymbol{\theta}, a_1 z_1 \cdots z_{L-1} a_L) \\
 &= p(a_1 \cdots a_L|\boldsymbol{\theta}, z_1 \cdots z_{L-1}) p(z_1 \cdots z_L|\boldsymbol{\theta}, a_1 \cdots a_L)
 \end{aligned}$$

行動選択の尤度

観測列の尤度



# 環境モデルの徒弟学習(3)

- **行動選択の尤度** はエキスパートの要素

$$p(a_1 \cdots a_L | \theta, z_1 \cdots z_{L-1}) = \prod_{i=1}^L \pi_{\theta}^*(b_{[\theta, a_1 \cdots z_{i-1}]}, a_i)$$

$\pi_{\theta}^*$  は、POMDP  $\rho_{\theta}$  に対する最適の方策

- **観測列の尤度** は環境からの反応として計算できる (IO-HMM と同様)

- これまでは、モデルパラメータ (報酬を除く、遷移・観測など) は後者のみから計算されていた

# アルゴリズム 1: MAP (事後分布最大) 推定

- 事後確率を最大にする  $\theta$  を探索する
$$\hat{\theta} = \operatorname{argmax}_{\theta} p(\theta|D)$$
- $p(\theta|D)$  の勾配の計算が難しい
  - 報酬のみが未知の場合は、局所勾配が計算できる
- 勾配を使わない最適化アルゴリズム COBYLA (Powell, 1998) を適用
  - 各々のパラメータ値の候補  $\theta$  に対し、ソルバーで POMDP  $\phi_{\theta}$  に対する最適ポリシー  $\pi_{\theta}^*$  を求めることで、事後確率を評価

# アルゴリズム2: MCMC (モンテカルロ法によるサンプリング)

- IO-HMM の MCMC アルゴリズムを元にする
  - 隠れ状態列  $s_1 \cdots s_L$  とパラメータ  $\theta$  を交互にサンプリングする
- メトロポリスアルゴリズム (Metropolis+, 1953) を導入
  - 提案サンプル  $\theta'$  を確率  $r$  で受理する
$$r = \min \left\{ 1, \frac{p(a_1 \cdots a_L | \theta')}{p(a_1 \cdots a_L | \theta)} \right\}$$
  - サンプル分布に行動選択の尤度が反映される

Loop

Sample  $s_1 \cdots s_L$

For each dimension  $i$

Sample  $\theta'_i$

Call POMDP solver

Accept  $\theta'_i$  with prob.  $r$

# 実験 (1)

- Tiger タスク (Kaelbling et al., 1998) のベイズ版
  - どちらかのドアの背後に虎がいる ( $p_i, 1 - p_i$ )
  - エージェントはどちらかのドアを選んで開けられる
    - 虎のいないドアなら良い (報酬 +10)
    - 虎に出会ってしまうと悪い (報酬  $r_t$ )
  - または、エージェントは「聞く」 (reward -1)
    - 虎の音が (左, 右) が聞こえる確率  
( $p_l, 1 - p_l$ ) 虎が左にいる場合  
( $1 - p_r, p_r$ ) 虎が右にいる場合



## 実験 (2)

- 100 個の演示を生成した
  - エキスパートはPOMDPソルバーの解に基づくソフトマックスポリシー ( $\beta=0.3$ ) で行動を選択
  - 各々の演示は100ステップの行動で構成されている (平均 22 エピソード)
- 各々の演示に対して、学習アルゴリズムを適用し、パラメータの事後分布と学習した行動を比較した
  - MCMC サンプラーの場合は、バーンイン100回の後の900サンプルを収集し、そのうちの90個を行動生成に利用

# 結果 (1)

- 推定されたパラメータ分布 (Makino&Takeuchi, 2012)

Table 1. Distribution of the estimated posterior parameters.

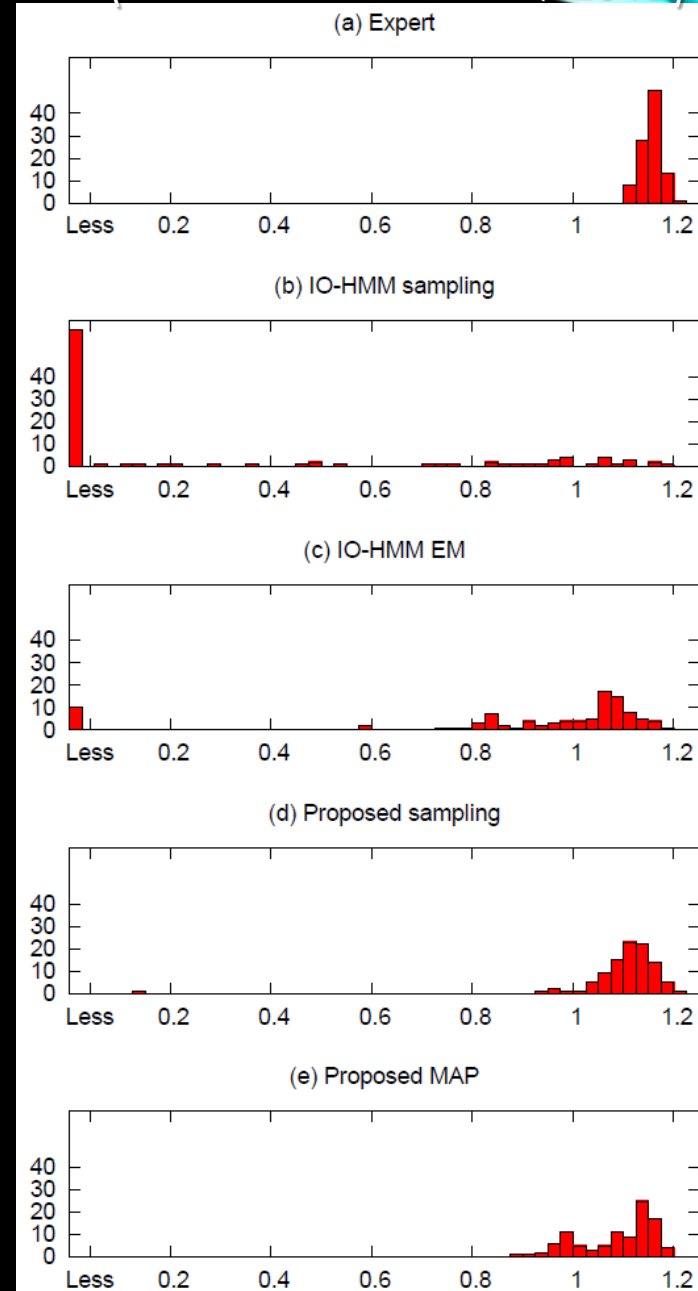
		Error of prior mean	IO-HMM		Proposed	
			Sampler	EM	Sampler	MAP
$p_i$ : prob. of tiger position	mean error	-0.1	-0.077	-0.05	-0.009	<b>-0.007</b>
	RMSE		0.200	0.09	<b>0.034</b>	0.066
	s.d. samples		0.186		<b>0.020</b>	
$p_l$ : prob. of hear left when the tiger is left	mean error	-0.225	-0.144	-0.066	-0.052	<b>-0.014</b>
	RMSE		0.202	0.144	0.178	<b>0.088</b>
	s.d. samples		0.031		<b>0.014</b>	
$p_r$ : prob. of hear right when the tiger is right	mean error	-0.225	-0.206	-0.104	-0.052	<b>-0.013</b>
	RMSE		0.263	0.170	0.179	<b>0.088</b>
	s.d. samples		0.032		<b>0.017</b>	
$r_t$ : reward of seeing the tiger	mean error	50.00	49.5	50.0	12.3	<b>10.3</b>
	RMSE		49.5	50.0	35.5	<b>19.2</b>
	s.d. samples		49.9		10.1	

RMSE: Root mean squared error of the estimate values.

s.d. samples: Average standard deviation of sampled values.

## 結果 (2)

- 学習したエージェントの行動を100,000 ステップシミュレーションし、ステップあたり報酬値の平均を評価
- 提案手法を利用することで、エキスパートに近い良さの行動を実現することができる





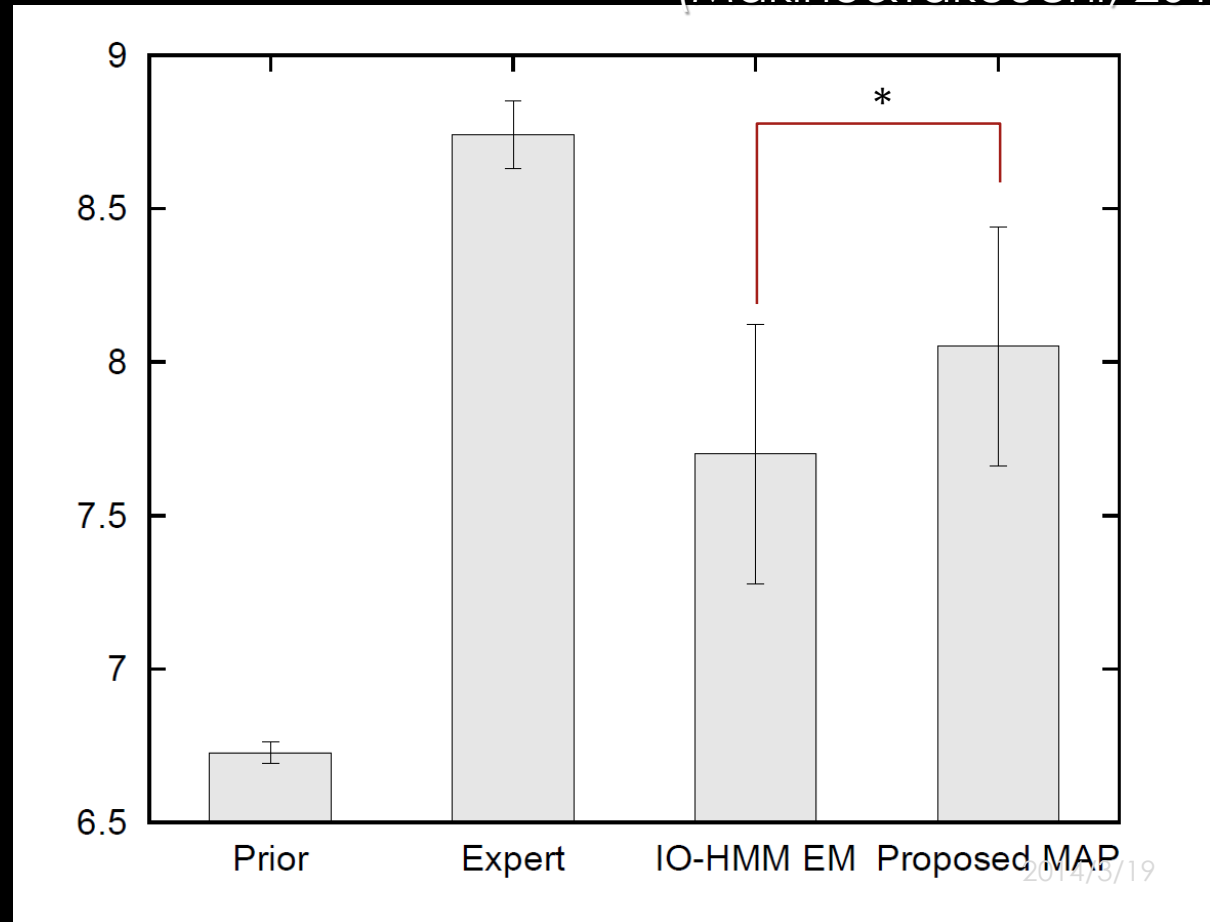


## 結果 (3)

- ステップあたり報酬の平均値 (12試行の平均)

\*:  $P < .05$

(Makino&Takeuchi, 2012)

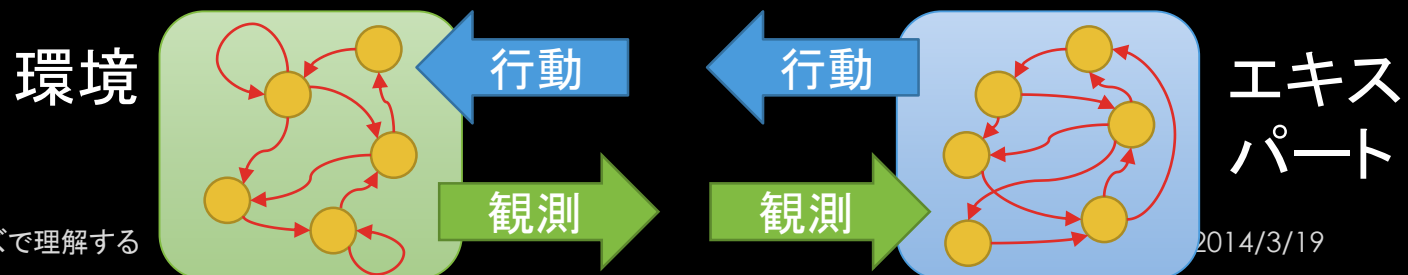


# 考察

- 提案手法は、非常に短い演示からでも適切に環境のモデルパラメータを推定できた
  - IO-HMMに基づく手法 (環境からの応答のみを利用) では、演示が短すぎる
  - エキスパートの行動が環境モデルに関する情報源として有用であることが示唆された
- ボトルネック: POMDP ソルバーの呼び出しが計算時間を消費する
  - 高速化するソルバーを現在研究中

# もっとベイズ的なアプローチも...

- Bayesian Nonparametric Policy Prior  
(Doshi-Velez+ 2010)
  - 環境モデルの事前分布に iPOMDP を仮定する
    - iPOMDP: HDP-HMM に入力を考慮したもの  
(Doshi-Velez+ 2009)
  - 方策の事前分布にも iPOMDP を適用する
    - 観測と行動を入れ替える
  - 環境と方策を交互にサンプリング



# 今回触れられなかった話題

- 連続状態、連続行動、連続時間の場合の強化学習
- 価値関数の関数近似 (ニューラルネット、カーネル法...)
  - Gradient TD (Sutton+, 2008), Generalized TD (Ueno+, 2011)
- 部分観測環境の表現学習
  - Predictive State Representation (Littman+ 2002), TD-Network (Sutton&Tanner 2005; Makino+, 2009)
- 階層的強化学習、時間的抽象化
  - Semi-Markov Decision Process (Sutton+, 1999)
- マルチエージェント環境
  - Interactive POMDP (Gmytrasiewicz&Doshi 2005)
- 転移学習 (マルチタスク学習), ...

# まとめ

- 強化学習は、未知の環境に対して効率的に探索し、利用する=データを作りつつ使うフレームワーク
- 「不確かな時は楽観的に」原理
- ベイズ統計の枠組みで不確かさを扱うことで、最適解が解析的に定義できる
  - 最適解はintractable→適切な近似法の研究
- 問題の枠組み自体も研究の余地がある
  - 報酬関数の獲得
  - エキスパートの演示からの学習 (徒弟学習)

# 参考文献

- Reinforcement Learning: An Introduction (Sutton & Barto, 1998)
- Reinforcement Learning: State-of-the-art (Wiering & van Otterlo, eds., 2012)
- 探索と利用のトレードオフとベイズ環境モデル (牧野 貴樹, 『計測と制御』 2013.2)
  - リレー解説「強化学習の最近の発展」  
『計測と制御』 2013.1～2013.12