機械学習(3)：
強化学習

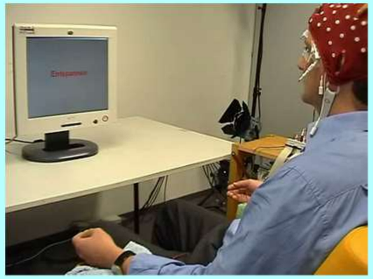
杉山 将
sugi@k.u-tokyo.ac.jp
http://www.ms.k.u-tokyo.ac.jp

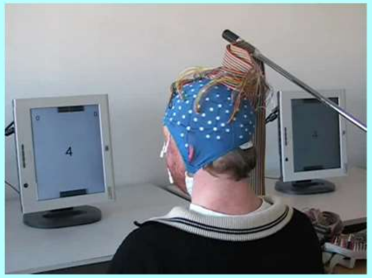
機械学習

目標:コンピュータにヒトのような学習能力を身につけさせる

教師付き学習:人間が教師となり，コンピュータを学習させる

- 回帰，分類など





脳波によるコンピュータの操作
（独フラウンホーファーとの共同研究）

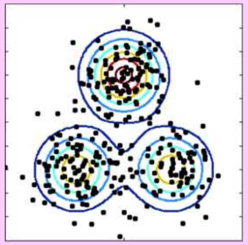
強化学習:エージェントが試行錯誤を通じて学習する

- ロボット制御，アートなど

ヒューマノイドの運動制御
（NICT･ATRとの共同研究）

教師なし学習:コンピュータが人間の手を介さずに学習する

- 異常検知，クラスタリングなど

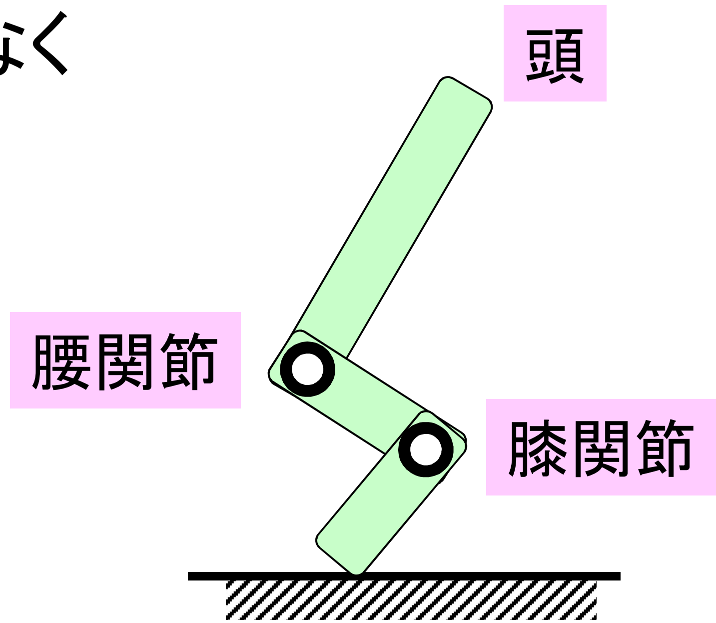


強化学習(reinforcement learning)

- エージェントに最適な行動政策を獲得させるための枠組み
  - 試行錯誤を繰り返し，うまく行動できるよう制御パターンを更新していく
- ユーザーが与えるもの：報酬
  - 行動が成功 ➡ 正の報酬を与える（“褒める”）
  - 行動が失敗 ➡ 負の報酬を与える（“叱る”）
- エージェントは，得られる報酬が最大になるように行動規則を更新する．

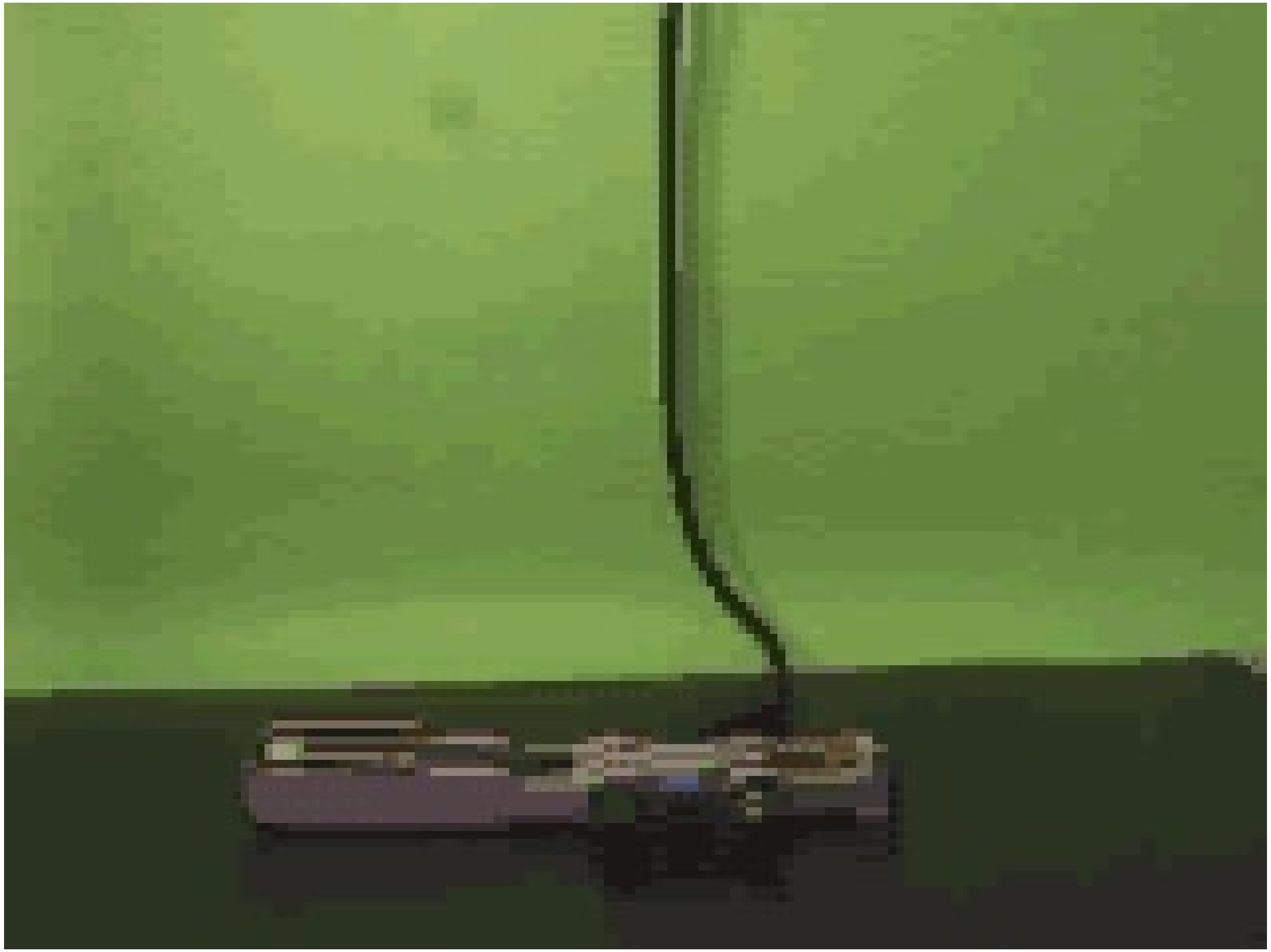

例：ロボットの起き上がり動作の獲得

- ロボット：
  - 関節が二つ
  - 関節の角度を操作できる
- 目標：
  - 床に横たわっている状態から，明示的に動きを教わることなく起き上がり動作を獲得する
- 報酬：
  - 頭の高さ



起き上がり動作の獲得：初期状態

■学習前



From http://www.kawato.jst.go.jp/xmorimo/stup_movie.html

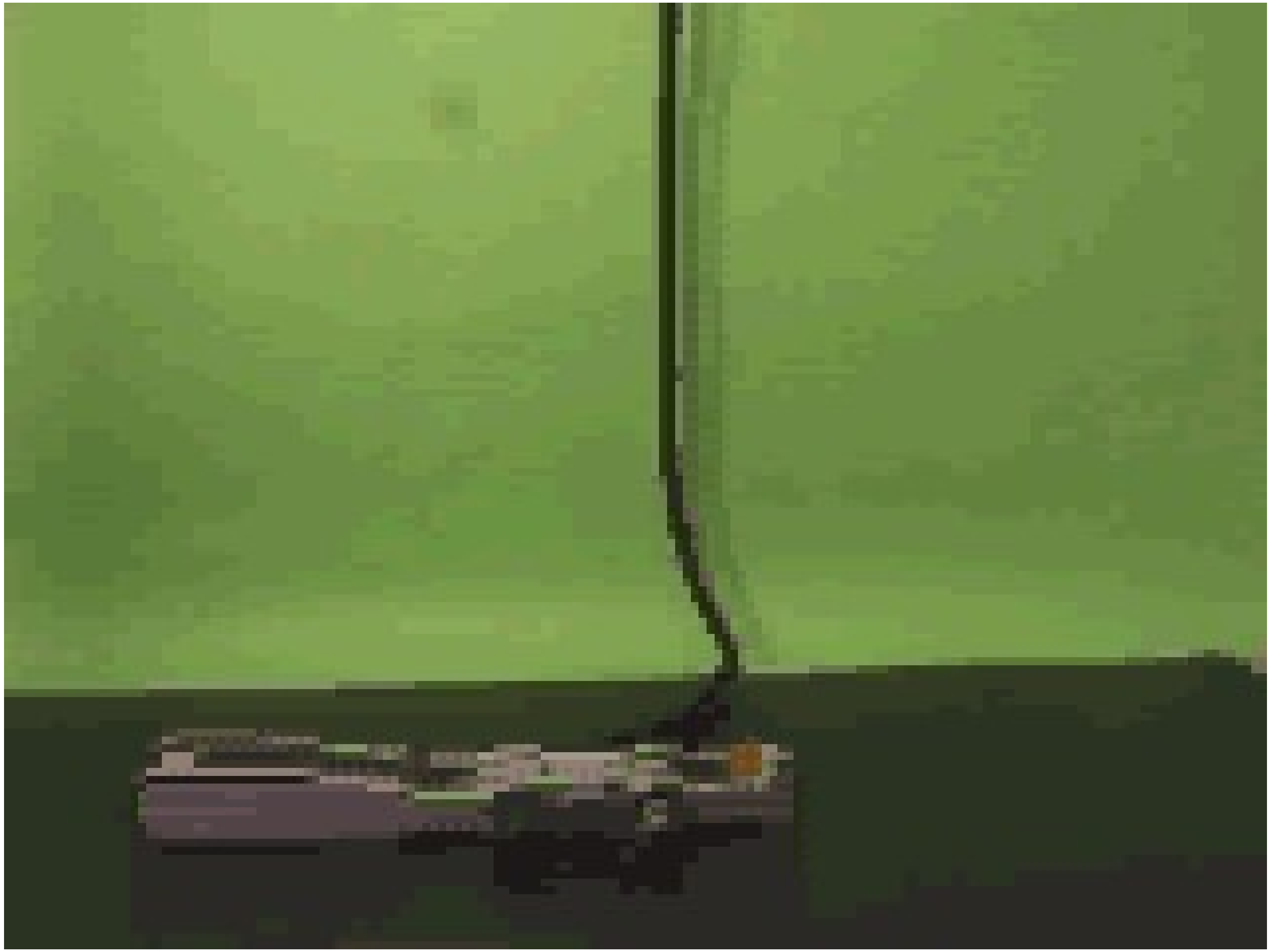
起き上がり動作の獲得：学習途中

- 750回の学習後

From http://www.kawato.jst.go.jp/xmorimo/stup_movie.html

起き上がり動作の獲得:学習後

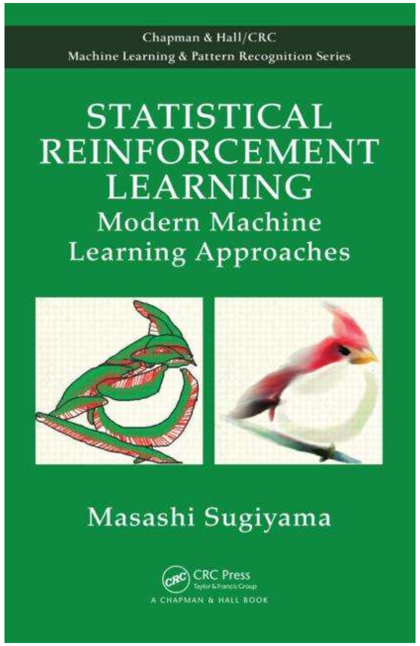
- 920回の学習後

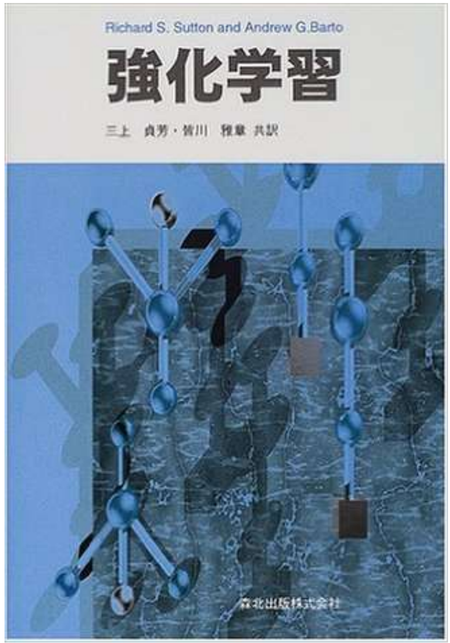


From http://www.kawato.jst.go.jp/xmorimo/stup_movie.html

参考書

- Sugiyama, M. Statistical Reinforcement Learning, Chapman and Hall/CRC, 2015
- Sutton, R. S. & Barto, A. G.（著）, 三上 貞芳 皆川 雅章（訳）, 強化学習, 森北出版, 2000



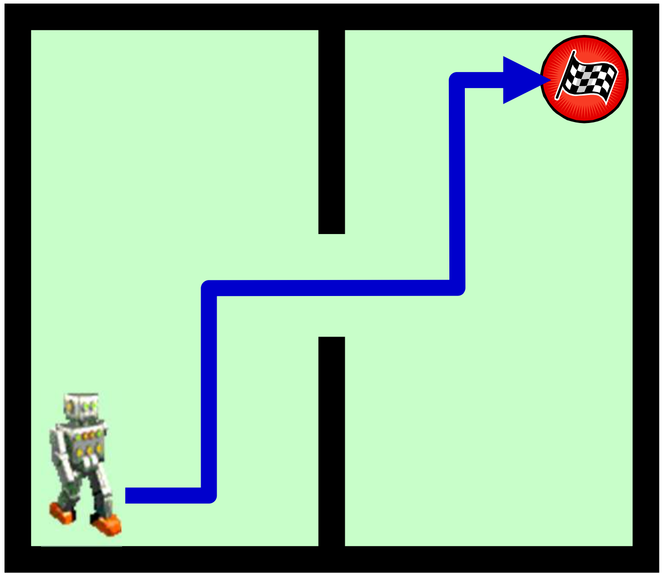


講義の流れ

1. 強化学習の定式化
2. 動的計画法
3. 政策反復法
4. 政策探索法
5. モデルに基づく強化学習
6. 逆強化学習

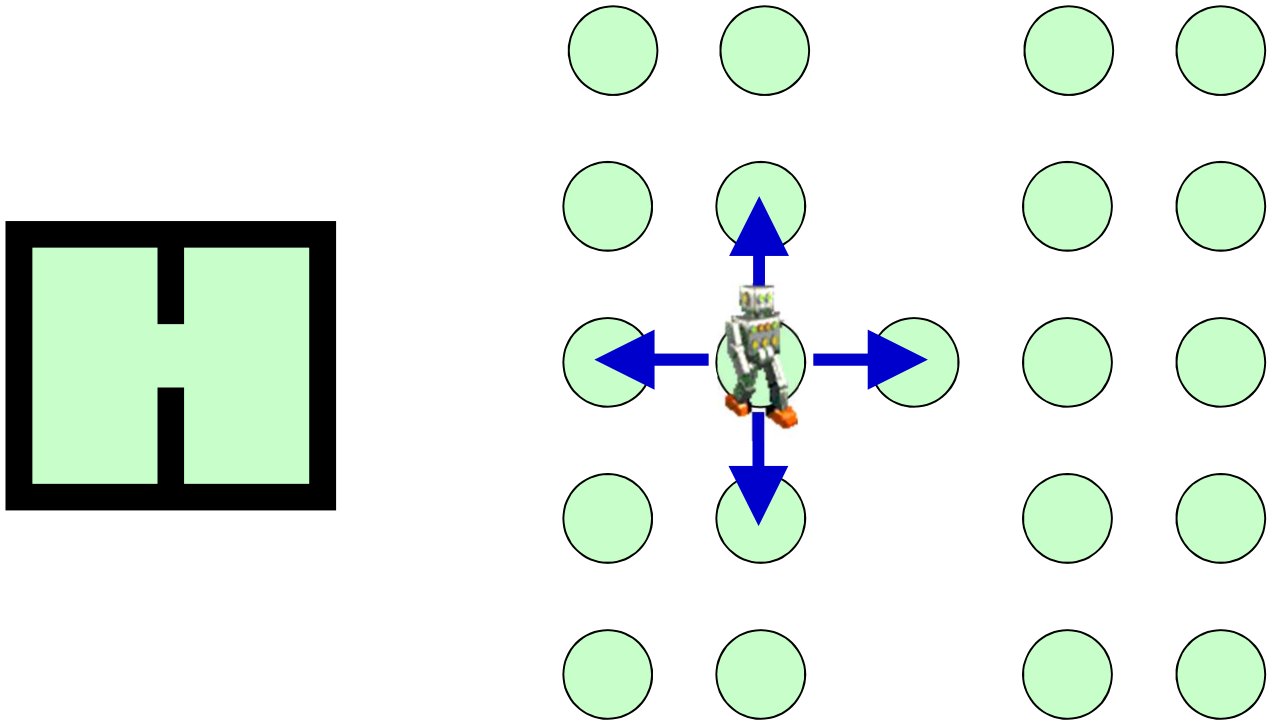
強化学習の定式化

- 迷路の中のロボットをゴールに誘導するナビゲーション問題を例に，強化学習の枠組みを説明する．



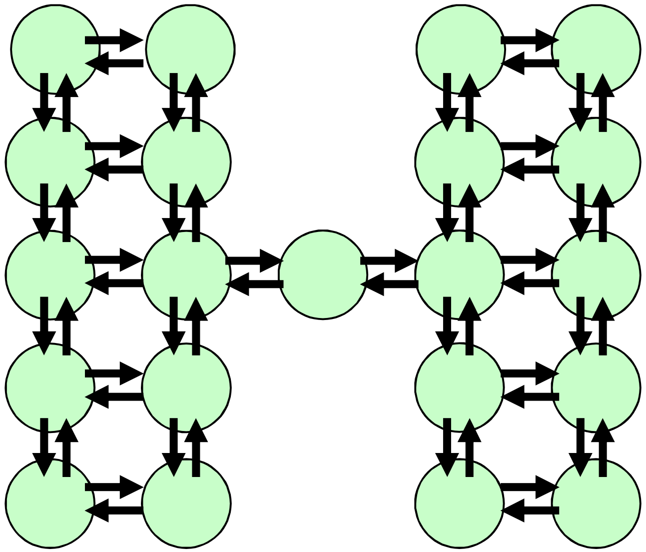
状態と行動

- 状態 s : ロボットの位置
- 行動 a : 上, 下, 左, 右に移動



状態遷移

- **状態遷移関数** $s' = T(s, a)$: 状態sで行動 a をとった時の行き先 s' を出力



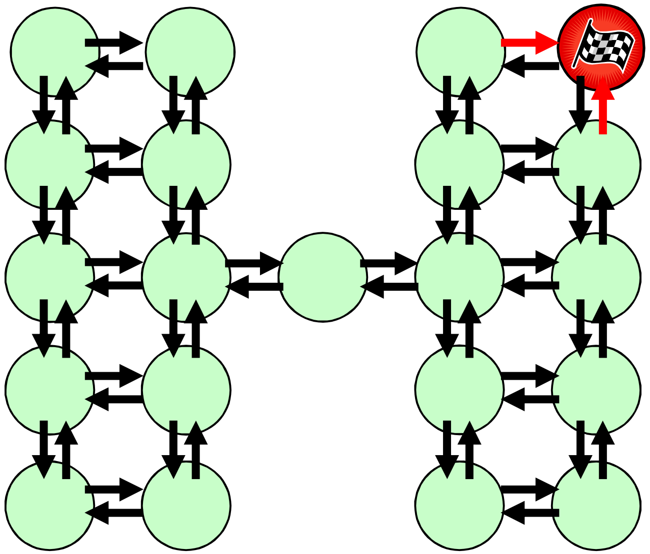
$s \xrightarrow{a} s'$ $s' = T(s, a)$

状態遷移がわかる
＝迷路の地図がわかる

- ノイズなどにより状態遷移が確率的な場合は, **状態遷移確率** $p(s'|s, a)$ を考える.
  - 例 : 右に行ったつもりが, 10%の確率で左に行く

即時報酬

■ **即時報酬関数** $R(s,a,s')$: 状態 s で行動 a をとって s' に移動したときに得られる報酬

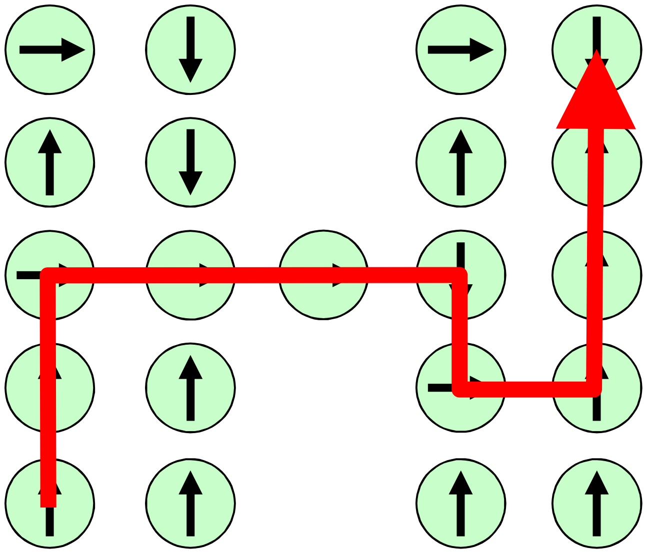


$$R(s,a,s') = \begin{cases} 1 & \text{if } s' \text{ is goal} \\ 0 & \text{otherwise} \end{cases}$$

即時報酬がわかる
＝ゴールの位置がわかる

政策

- 政策関数 $a = \pi(s)$: 状態 s において
  とるべき行動 a を記した関数



$$s_0 \xrightarrow{\pi(s_0)} s_1 \xrightarrow{\pi(s_1)} s_2 \xrightarrow{\pi(s_2)} s_3 \cdots$$

政策を決める
＝軌跡が得られる

- 行動の選択が確率的な場合は,
  行動選択確率 $\pi(a|s)$ を考える.

強化学習の目的

- 政策 π に従って行動し続けたときに得られる割引報酬和の期待値を最大にする政策を求める

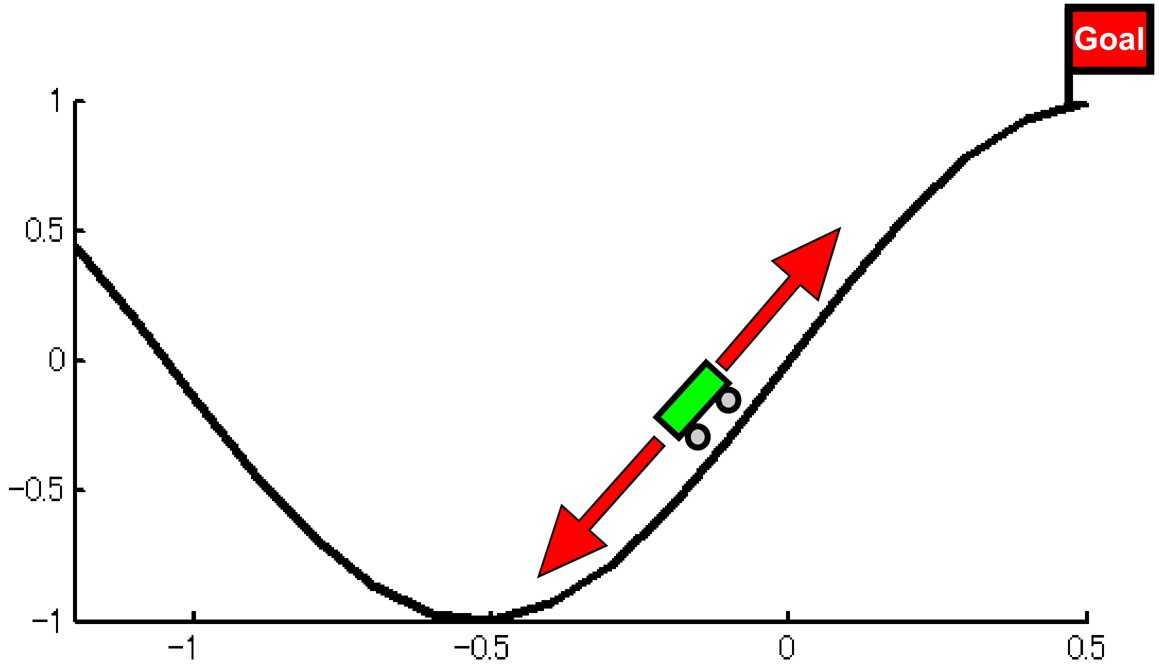
$$\max_{\pi}\left[\mathbb{E}\left(\sum_{t=0}^{\infty}\gamma^{t}R(s_t,a_t,s_{t+1})\right)\right]$$

$$s_0 \xrightarrow{\pi(s_0)} s_1 \xrightarrow{\pi(s_1)} s_2 \xrightarrow{\pi(s_2)} s_3 \cdots$$

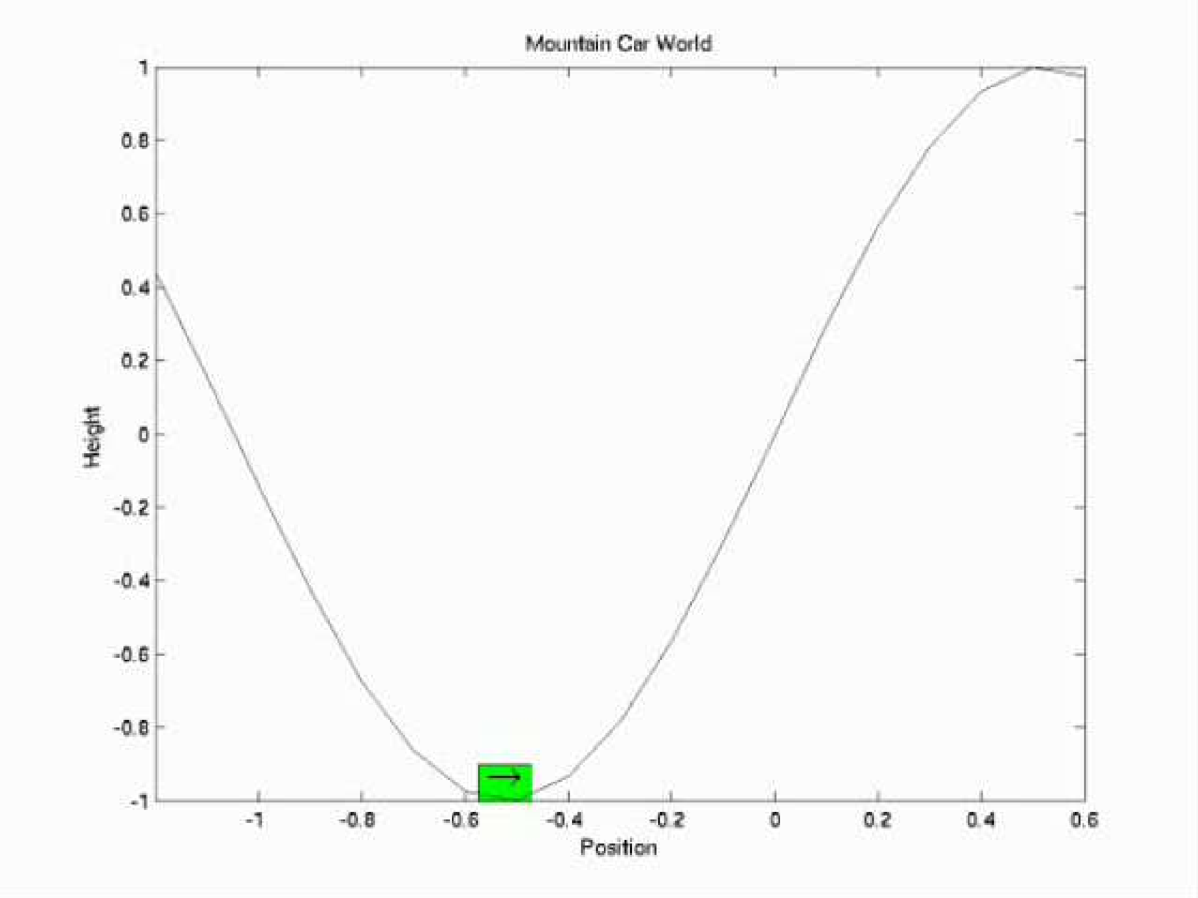
- 即時報酬ではなく長期的な報酬を考慮する
  - “損して得をとれ”
- $0 < \gamma < 1$ ：割引率(将来得る報酬は割引いて評価)
  - 報酬は将来もらうより今もらうほうがうれしい
  - 早くゴールに到達したほうが良い

例：山登り問題

- 車を山の頂上のゴールに誘導する.
- 報酬はゴールまでの距離に反比例：
  - ゴールに近いほど多くの報酬がもらえる
- 車のパワーは山を登れるほど強くない.

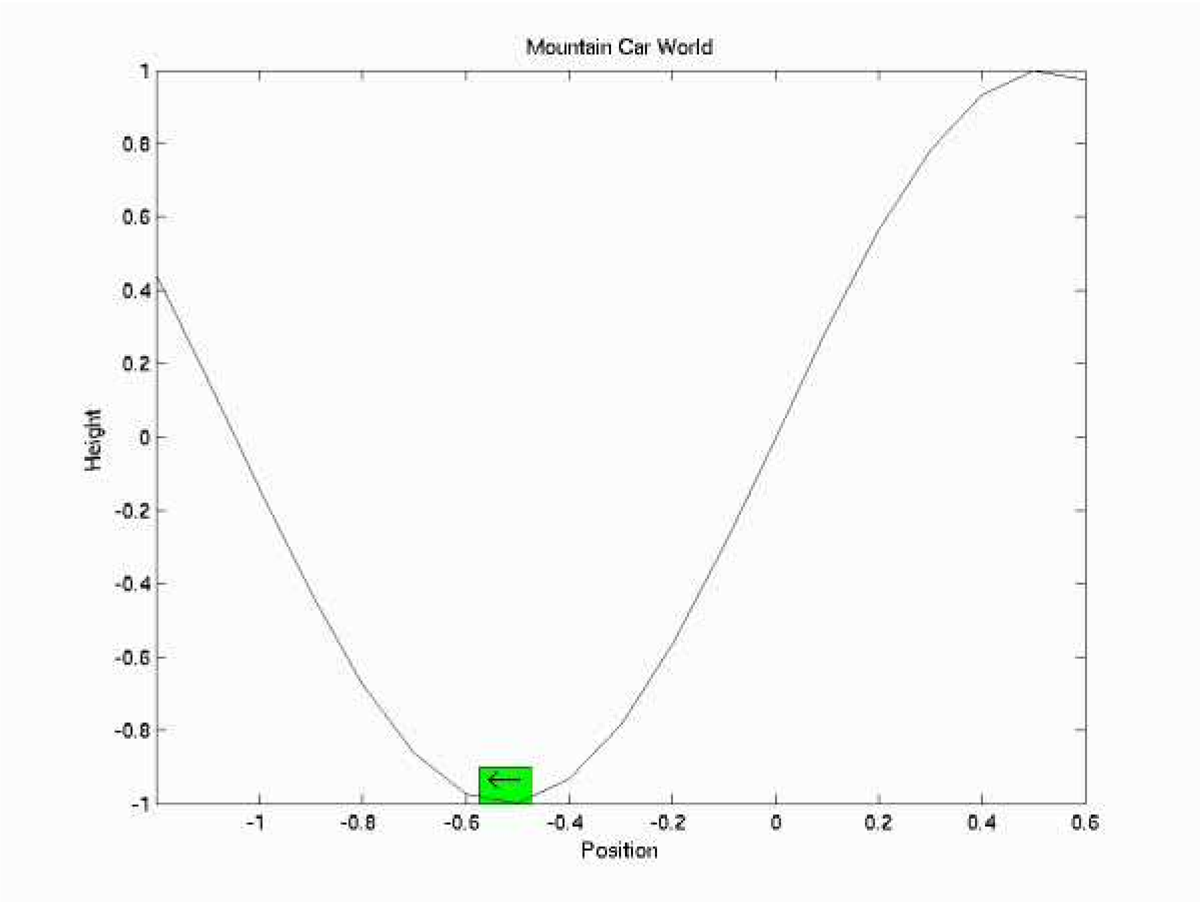


短期的な報酬を最大化した場合



- すぐにもらえる報酬を最大化するために，右に進もうとする．
- しかしパワー不足で山を登りきれない．

長期な報酬を最大化した場合



- まずは，損をしてでも左の山に登る.
- そして，加速をつけて右の山を一気に登る.

強化学習の定式化のまとめ

■ マルコフ決定問題:

- 状態集合 $\mathcal{S} = \{s\}$
- 行動集合 $\mathcal{A} = \{a\}$
- 状態遷移 $p(s'|s, a)$
- 即時報酬 $R(s, a, s')$
- 割引率 γ

状態遷移は
マルコフ性を持つと仮定
（次の状態s'への遷移
確率は，現在の状態sと
行動aのみに依存する）

■ ゴール:

- 期待割引報酬和を最大にする政策 $\pi(a|s)$ を獲得

$$\max_{\pi} \left[\mathbb{E} \left(\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1}) \right) \right]$$

講義の流れ

1. 強化学習の定式化
2. 動的計画法
3. 政策反復法
4. 政策探索法
5. モデルに基づく強化学習
6. 逆強化学習

政策のよさの尺度：状態価値

- 状態価値関数 $V^{\pi}(s)$：状態 s から政策 π に従って行動し続けたときに得られる割引報酬和の期待値

$$s_0 \xrightarrow{\pi(s_0)} s_1 \xrightarrow{\pi(s_1)} s_2 \xrightarrow{\pi(s_2)} s_3 \cdots$$

$$V^{\pi}(s_0) = \mathbb{E}\left(\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})\right)$$

$$0 < \gamma < 1$$

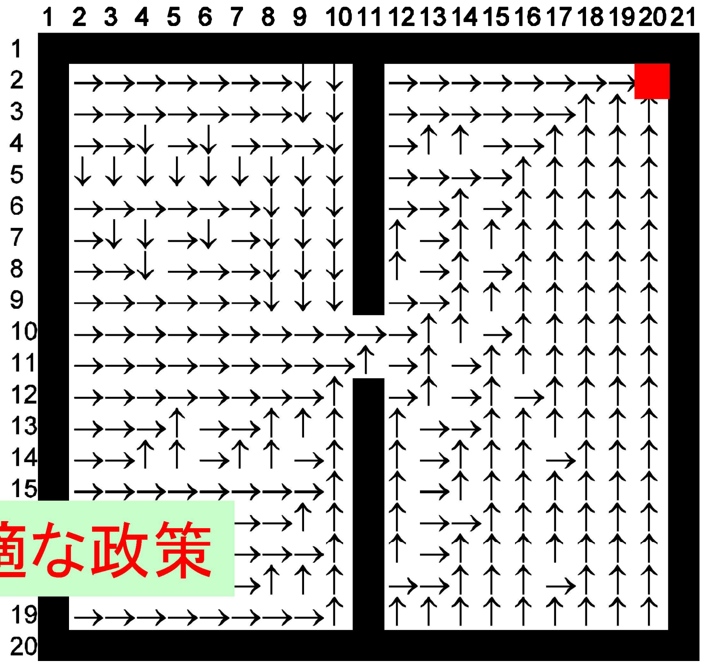
最適な状態価値関数

- 最適な状態価値関数:

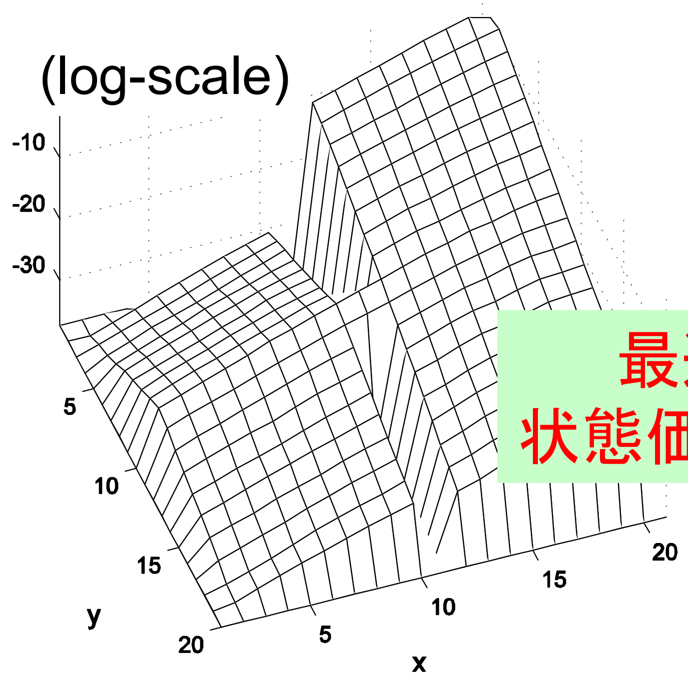
$$V^*(s) = \max_{\pi} V^{\pi}(s) \qquad V^{\pi}(s_0) = \mathbb{E}\left(\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})\right)$$

- 状態 s において, $V^*(s')$ が最大となる行動 a をとるのが最適

$$s \xrightarrow{a} s'$$



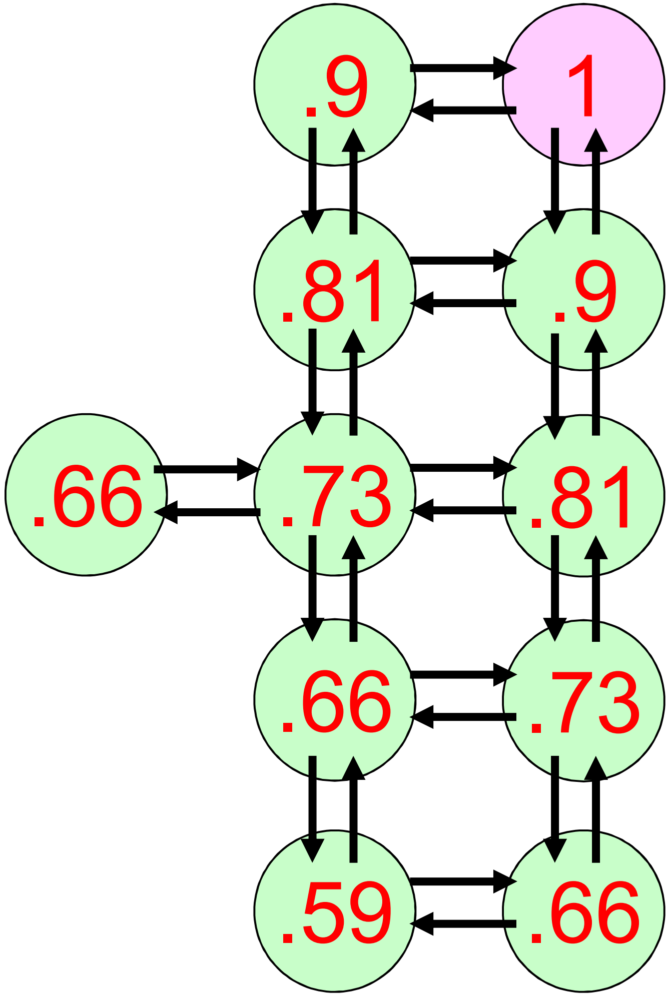
最適な政策



最適な状態価値関数

動的計画法

■ゴールから順に状態価値$V^*(s)$を伝播させる

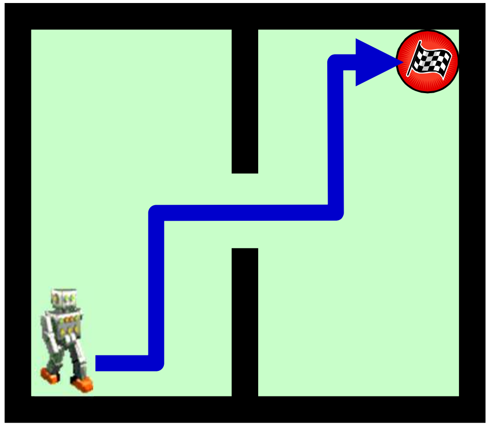


$$V^*(s) = \max_{\pi} V^{\pi}(s)$$

$$\gamma = 0.9$$

$$V^{\pi}(s_0) = \mathbb{E}\left(\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})\right)$$

$$R(s, a, s') = \begin{cases} 1 & \text{if } s' \text{ is goal} \\ 0 & \text{otherwise} \end{cases}$$



動的計画法（続き）

- 動的計画法:ゴールから順に状態価値を伝播させる
  - 単純でわかりやすい
  - 実装が比較的簡単
  - 計算時間がかかる
  - 状態遷移が確率的な場合，状態遷移確率を推定する必要がある

発表の流れ

1. 強化学習の定式化
2. 動的計画法
3. 政策反復法
4. 政策探索法
5. モデルに基づく強化学習
6. 逆強化学習

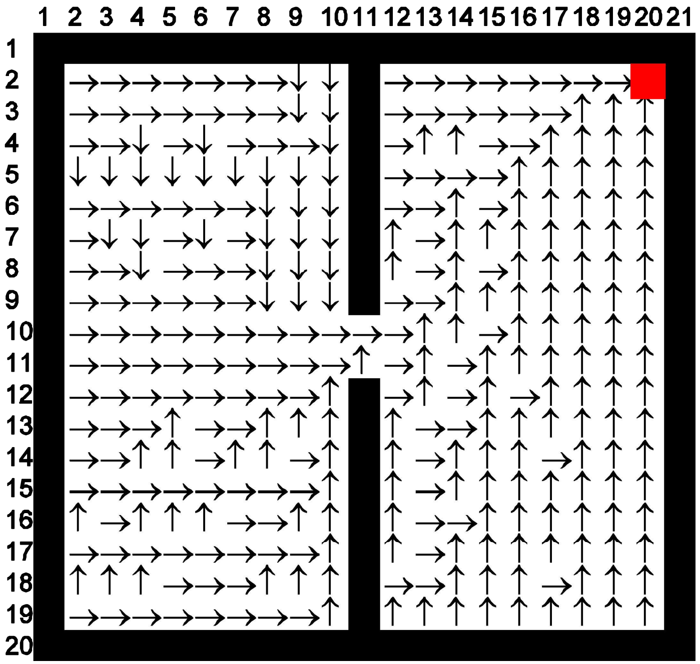
最適な政策

- 最適な政策に対する状態価値関数:

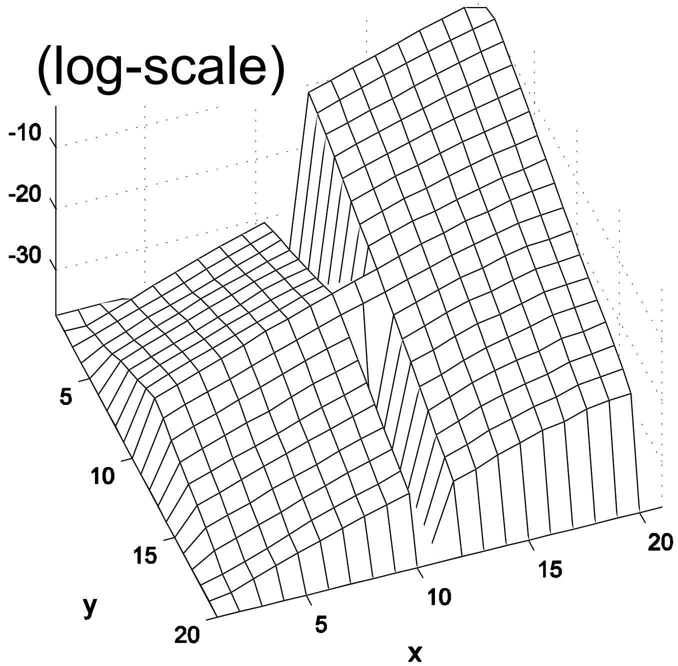
$$V^*(s) = \max_{\pi} V^{\pi}(s)$$

- 最適な政策を明示的に表現できないか？

最適な政策



最適な状態価値関数



状態行動価値

- 状態行動価値関数 $Q^\pi(s,a)$: 状態 s で行動 a をとり，その後政策 π に従って行動し続けたときに得られる割引報酬和の期待値

$$Q^\pi(s,a) = \mathbb{E}\left(\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1}) \middle| s_0 = s, a_0 = a\right)$$

- 最適な政策 π^* は

$$\pi^*(s) = \underset{a'}{\mathrm{argmax}}\, Q^*(s,a')$$

$$Q^*(s,a) = \max_\pi Q^\pi(s,a)$$

- π^* も Q^* も未知なので，交互に推定する

政策反復法

現在の政策に従って標本を収集 → 現在の政策の価値関数を推定 → 政策を更新 → 現在の政策に従って標本を収集

$$\pi(s) \longleftarrow \operatorname*{argmax}_{a'} Q^{\pi}(s, a')$$

- 政策反復法は最適政策 π^* に収束する
- 無限和で表される状態行動価値関数をどうやって計算機で計算するか？

$$Q^{\pi}(s,a) = \mathbb{E}\left(\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1}) \middle| s_0 = s, a_0 = a \right)$$

ベルマン方程式

- **ベルマン方程式**(再帰表現):

$$s \xrightarrow{a} s' \xrightarrow{a'} \cdots$$

$$Q^\pi(s,a) = \mathbb{E}\Big[\ \underbrace{R(s,a,s')}_{\text{次に得られる報酬}} + \underbrace{\gamma Q^\pi(s',a')}_{\text{その後に得られる報酬和}}\ \Big]$$

- $Q^\pi(s,a)$ はベルマン方程式(連立一次方程式)を解くことにより, **厳密に計算**できる.
- しかし, 現実には $R(s,a,s')$ と $p(s'|s,a)$ が未知.
- これらを**標本**から推定する:

$$\{(s_t, a_t, r_t)\}_t$$

- ベルマン方程式が近似的に解ける!

政策反復法の欠点

$$Q^{\pi}(s,a) = \mathbb{E}\left[R(s,a,s') + \gamma Q^{\pi}(s',a')\right]$$

- ベルマン方程式の変数の数が膨大

$$|\mathcal{S}| \times |\mathcal{A}|$$

  - 連続状態・連続行動の場合は無限大！
- そのため，
  - 計算コストがかかる（連続状態・連続行動の場合はそもそも計算不可能）
  - ノイズを含む標本に過適合しやすい

価値関数の近似

- 線形モデルで関数を近似:

$$\widehat{Q}^{\pi}(s,a)=\sum_{i=1}^{k} w_i \phi_i(s,a)$$

w_i :パラメータ

$\phi_i(s,a)$:基底関数

- パラメータは，ベルマン方程式が近似的に満たされるように決定.

$$\widehat{Q}^{\pi}(s,a)\approx \mathbb{E}\left[R(s,a,s')+\gamma \widehat{Q}^{\pi}(s',a')\right]$$

- パラメータ数を適当な数に抑えれば，計算コストが大幅に低減できる.

ベルマン二乗残差の最小化

$$\min_{\{w_i\}_{i=1}^{k}} \Big[\overbrace{\mathbb{E}\Big(\underbrace{R(s,a,s')}_{\text{即時報酬}} + \gamma \widehat{Q}^{\pi}(s',a') \Big)}^{\text{累積報酬モデルの再帰的表現}} \underbrace{- \overbrace{\widehat{Q}^{\pi}(s,a)}^{\text{累積報酬のモデル}} \Big]^2}_{\text{即時報酬のモデル}}$$

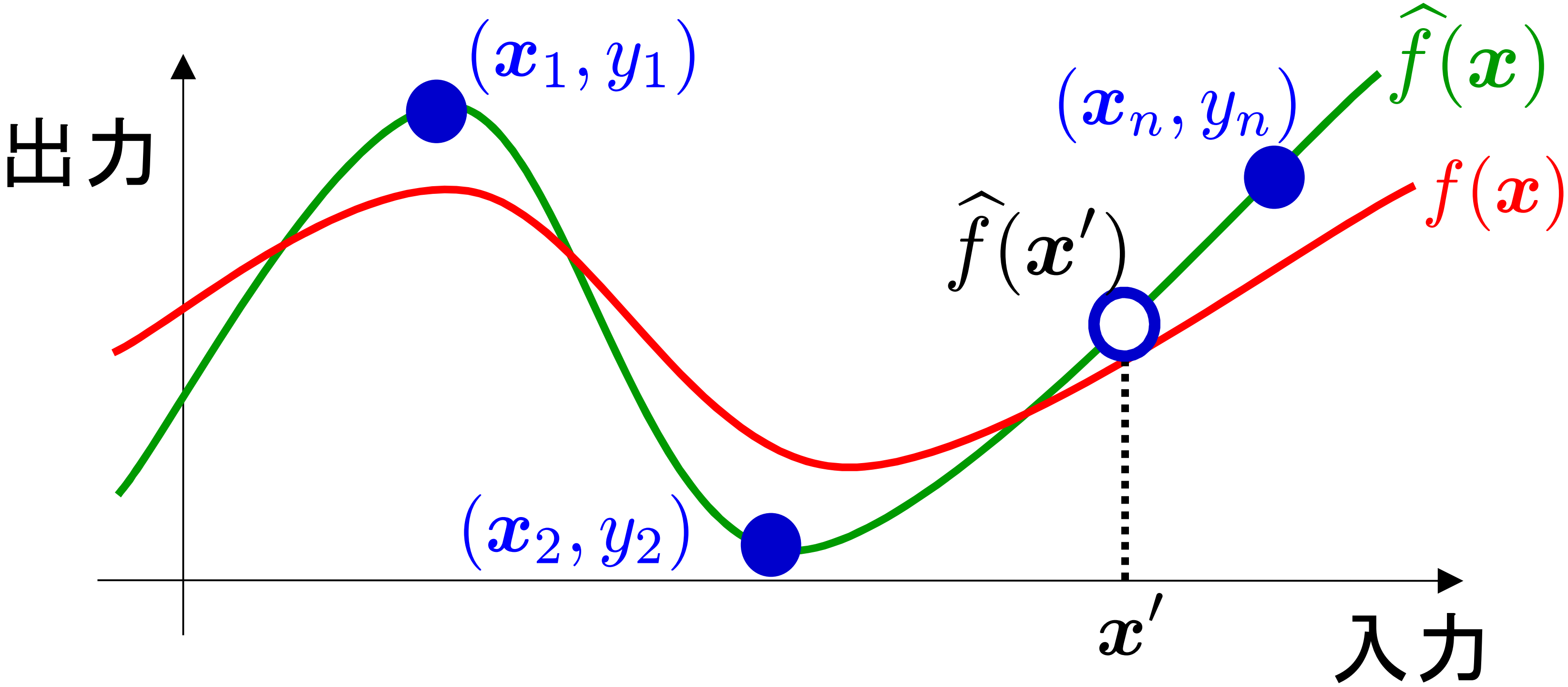
$$\widehat{Q}^{\pi}(s,a) = \sum_{i=1}^{k} w_i \phi_i(s,a)$$

$$\widehat{R}(s,a) = \sum_{i=1}^{k} w_i \left(\phi_i(s,a) - \gamma \mathbb{E}\phi_i(s',a')\right)$$

- 即時報酬関数の回帰と等価
- 結局，普通の回帰問題を解けばよい！

回帰問題

- ゴール: 訓練標本 $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ から関数 $f(\boldsymbol{x})$を学習する
- 未知のテスト入力 $\boldsymbol{x}'$での出力 $f(\boldsymbol{x}')$が推定できるようになる



訓練データの独立性

- マルコフ性のため，訓練データは独立でない.

$$\{(s_t, a_t, r_t)\}_t$$

$$p(s_{t+1} | s_t, a_t)$$

- 対処法：軌跡間の独立性を利用
  - 初期状態 s_0 を独立に生成

$$s_0 \overset{\text{i.i.d.}}{\sim} p(s)$$

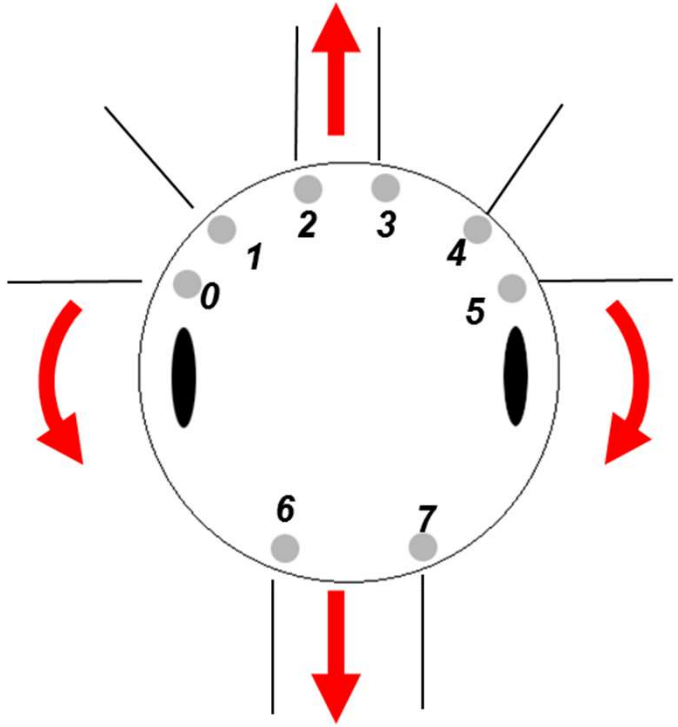
$$\{(s_t^{(1)}, a_t^{(1)}, r_t^{(1)})\}_t$$

$$\{(s_t^{(2)}, a_t^{(2)}, r_t^{(2)})\}_t$$

$$\vdots$$

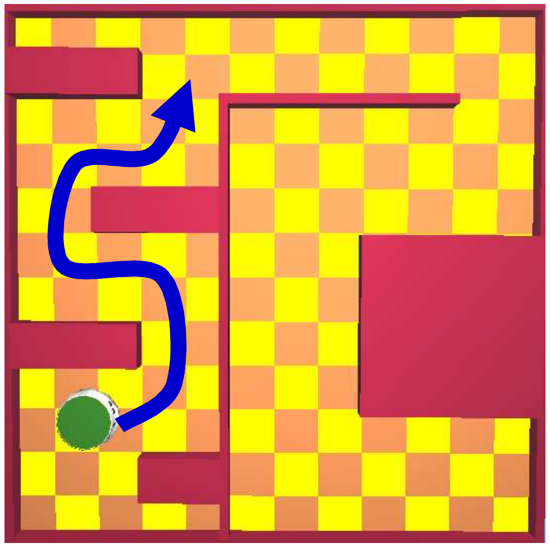
ロボットの障害物回避（1）

- Kheperaロボット
  - ２つの車輪：前進，後進，左回転，右回転
  - ８個の赤外線センサー：壁との距離を計測

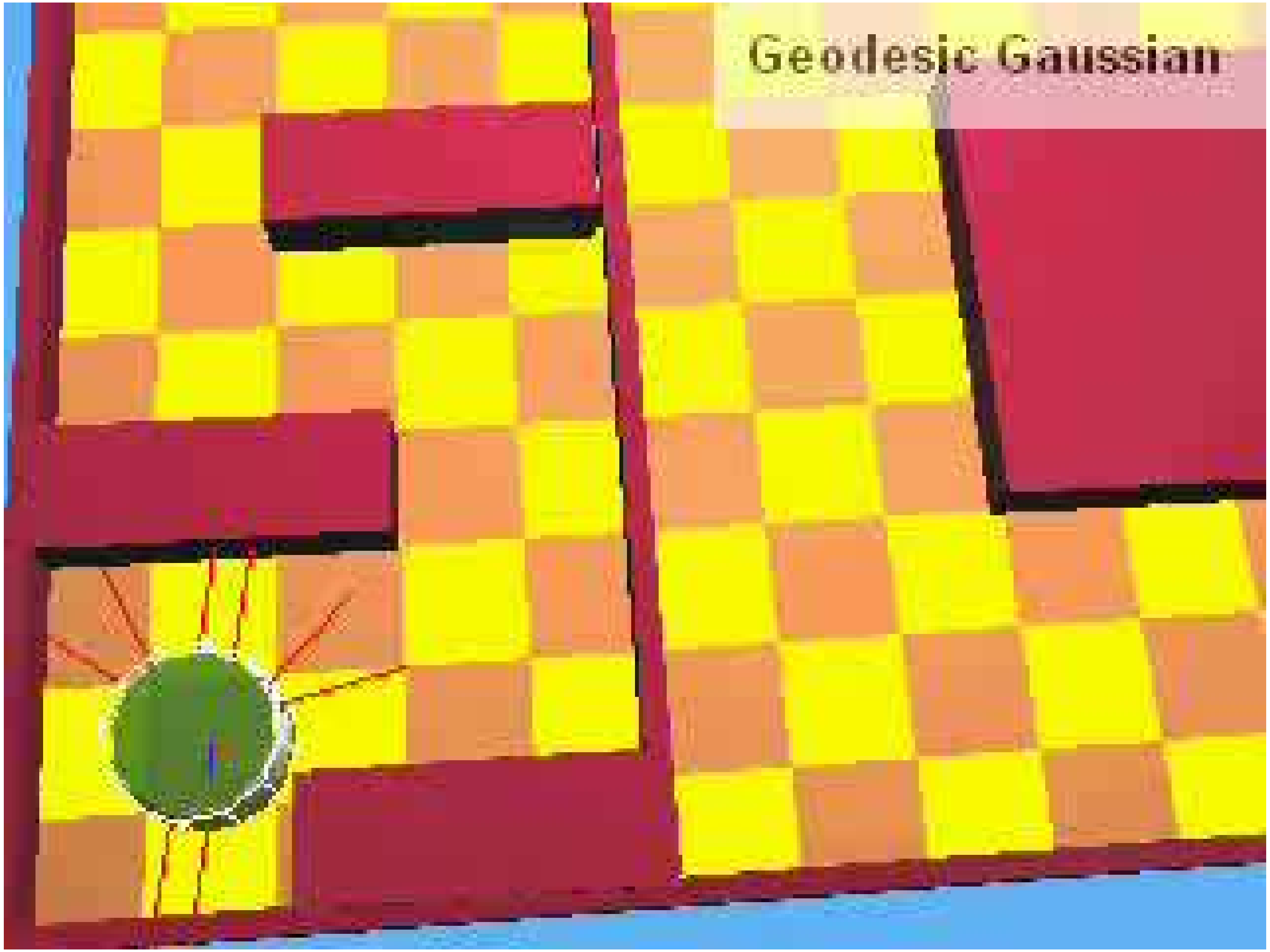


ロボットの障害物回避(2)

- ゴール:障害物を避けて進む
- 報酬:
  - 前に進んだらプラスの報酬
  - 壁にぶつかったらマイナスの報酬
- どうやって障害物を避けるかは教えない!

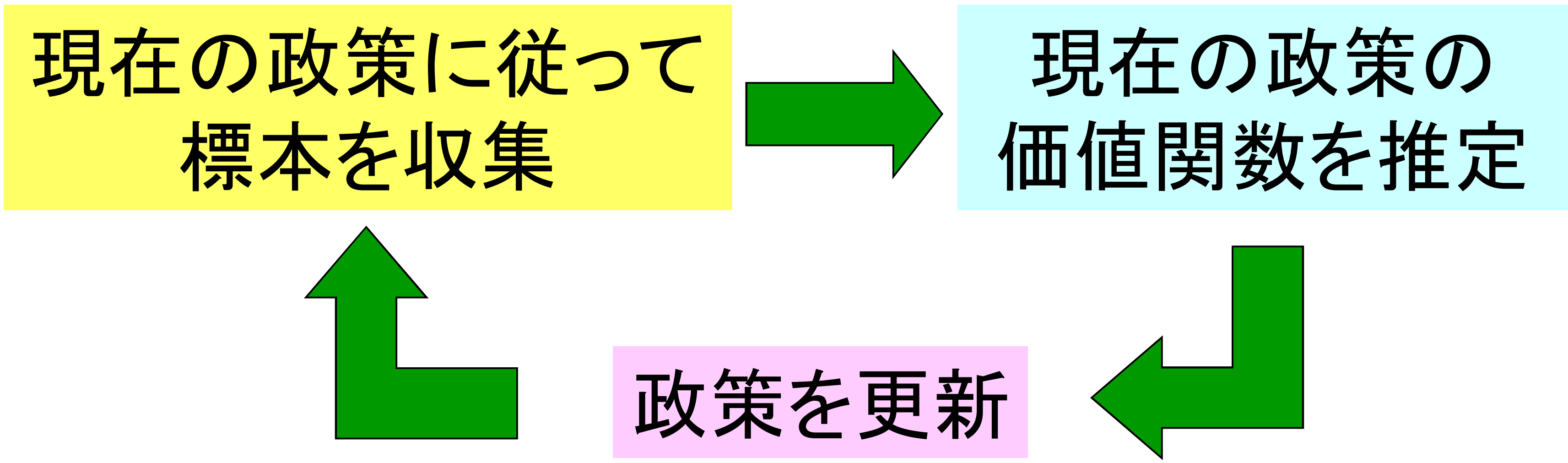


ロボットの障害物回避(3)



ここまでのまとめ

- 最小二乗政策反復法:



- 価値関数をデータから推定＝即時報酬の回帰
- 状態遷移データはマルコフ性を持つため,
  独立でなく扱いにくい
- 軌跡間の独立性を利用する

講義の流れ

1. 強化学習の定式化
2. 動的計画法
3. 政策反復法
4. 政策探索法
   - A) 政策勾配法
   - B) 事前政策探索法
5. モデルに基づく強化学習
6. 逆強化学習

政策反復法の弱点

現在の政策πに対して
状態行動価値関数
$Q^\pi(s,a)$ を計算

$Q^\pi(s,a)$ に基づいて
政策を更新

$$\pi(s) \longleftarrow \operatorname*{argmax}_a Q^\pi(s,a)$$

- 行動が連続のとき $\operatorname{argmax}_a Q^\pi(s,a)$ の計算が困難
- 報酬和(収益)を最大にする政策を直接学習する:

$$\mathbb{E}\left(\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})\right)$$

政策モデル

- 政策関数を直接モデル化 : $\pi(a|s;\boldsymbol{\theta})$
  - 例 : ガウスモデル

$$\pi(a|\boldsymbol{s};\boldsymbol{\eta},\sigma^2)=\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(a-\boldsymbol{s}^\top\boldsymbol{\eta})^2}{2\sigma^2}\right)$$

- 収益を最大にするようにパラメータを決定 :

$$\underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, J(\boldsymbol{\theta})$$

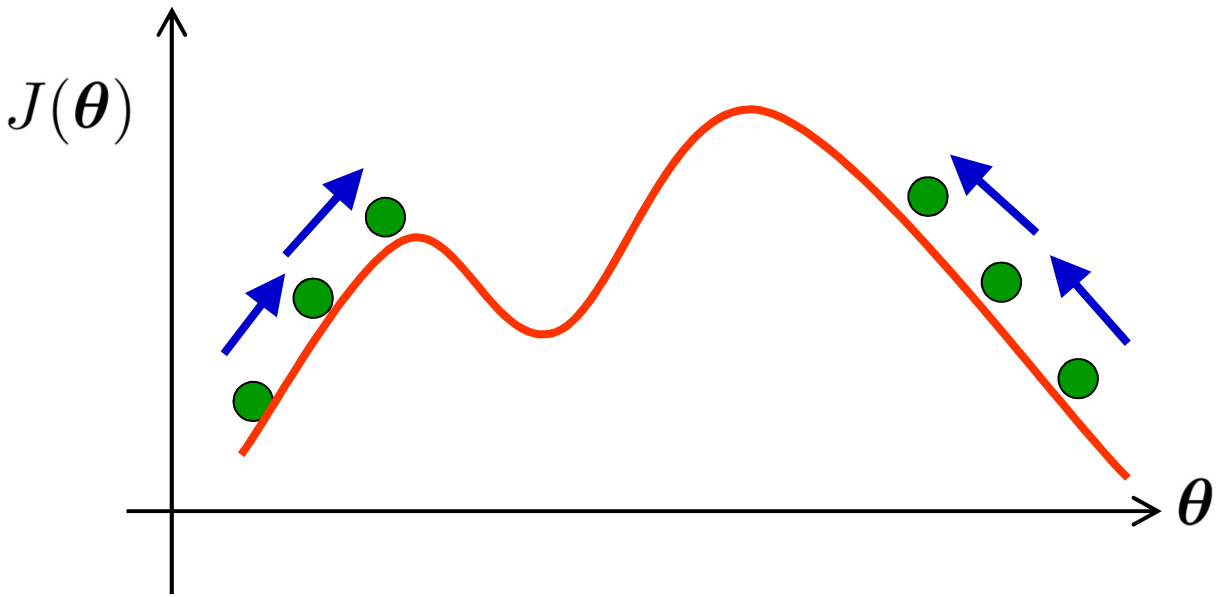
$$J(\boldsymbol{\theta})=\mathbb{E}\left(\sum_{t=0}^{\infty}\gamma^t R(s_t,a_t,s_{t+1})\right)$$

政策勾配法

- 収益の勾配を上昇するように
政策パラメータ $\boldsymbol{\theta}$ を学習

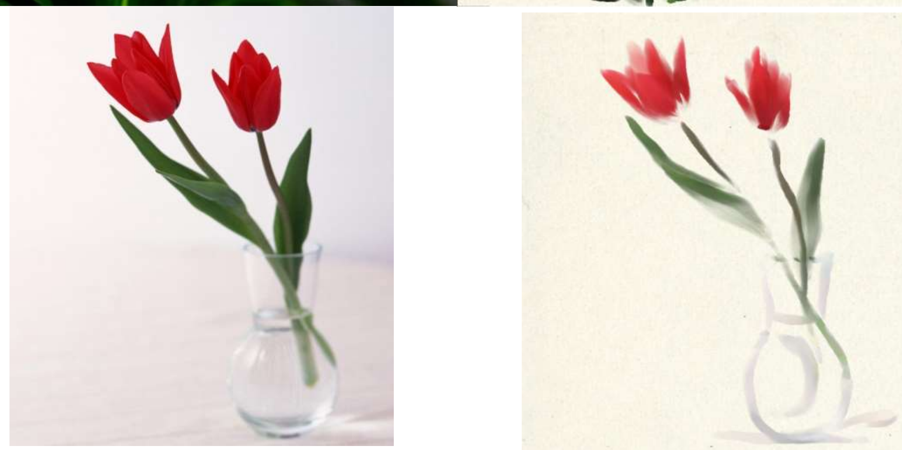
$$\boldsymbol{\theta} \longleftarrow \boldsymbol{\theta} + \varepsilon \nabla_{\boldsymbol{\theta}} \mathbb{E}\left(\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1}) \right)$$

$\varepsilon > 0$



コンピュータ・アートへの応用

強化学習による「筆ロボット」の学習

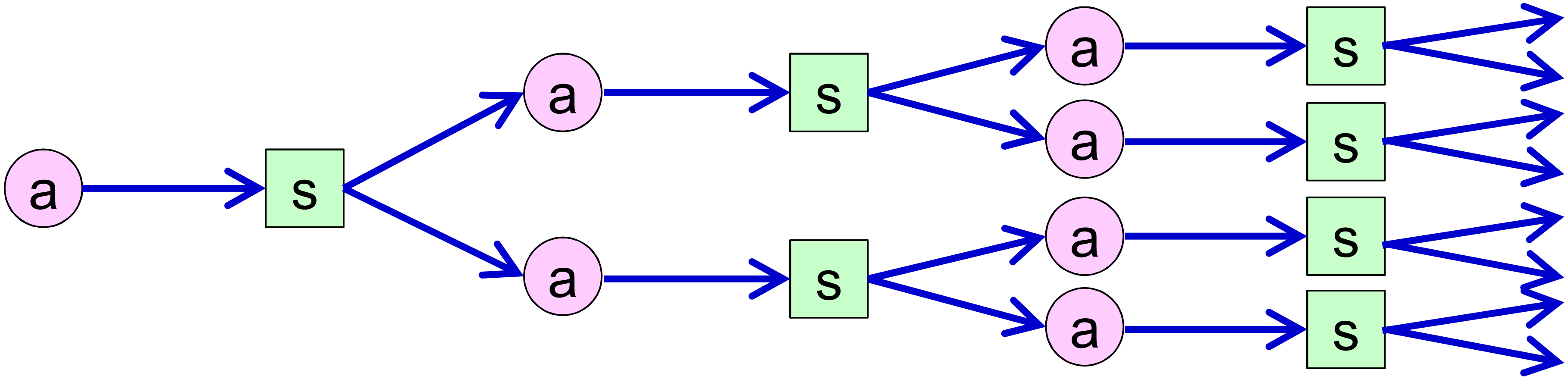


講義の流れ

1. 強化学習の定式化
2. 動的計画法
3. 政策反復法
4. 政策探索法
   A) 政策勾配法
   B) 事前政策探索法
5. モデルに基づく強化学習
6. 逆強化学習

直接政策探索の弱点

- 確率的な政策 $\pi(a|s)$ を用いることにより，状態空間を広く探索する



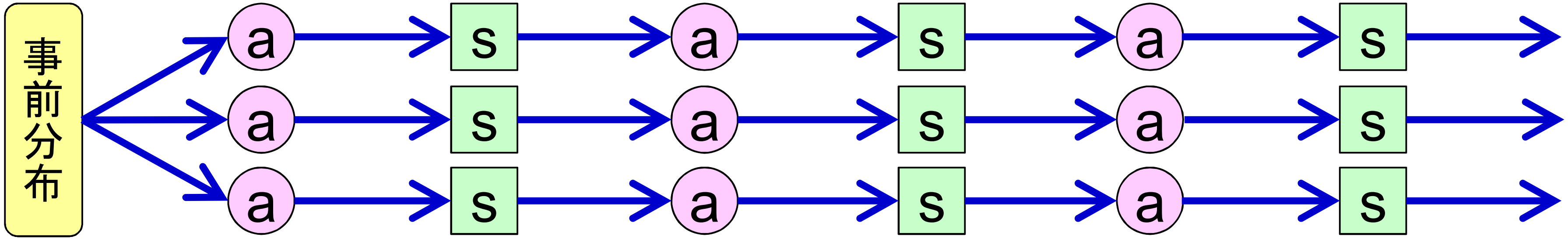
この図では確定的な状態遷移を仮定

- 軌跡が長くなると勾配推定の分散が拡大する

$$\boldsymbol{\theta} \longleftarrow \boldsymbol{\theta} + \varepsilon \nabla_{\boldsymbol{\theta}} \mathbb{E}\left(\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1}) \right)$$

事前政策探索法

- 決定的な政策 $a = \pi(s)$ を事前分布からサンプリング



この図では確定的な状態遷移を仮定

- 事前政策探索法による政策勾配は，直接政策探索法による政策勾配よりも分散が小さい

- 標本の再利用も可能

ヒューマノイドロボット制御

講義の流れ

1. 強化学習の定式化
2. 動的計画法
3. 政策反復法
4. 政策探索法
5. モデルに基づく強化学習
6. 逆強化学習

期待値の標本近似

$$\pi \longleftarrow \pi + \varepsilon \nabla \mathbb{E}\left(\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})\right)$$

- これまで紹介した全ての強化学習法では，状態遷移確率 $p(s'|s,a)$ に関する期待値を標本 $\{(s_t, a_t, r_t)\}_t$ の平均で近似している

$$\int f(\boldsymbol{x}) q(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \approx \frac{1}{n} \sum_{i=1}^{n} f(\boldsymbol{x}_i) \qquad \boldsymbol{x}_i \overset{\text{i.i.d.}}{\sim} q(\boldsymbol{x})$$

- 大数の法則により，一致性が保証
- 標本が少ないとき，近似精度が良くない

状態遷移推定

$$\pi \longleftarrow \pi + \varepsilon \nabla \mathbb{E}\left(\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})\right)$$

- 状態遷移確率　$p(s'|s,a)$ を標本 $\{(s_t, a_t, r_t)\}_t$ から明示的に推定し，推定した $\widehat{p}(s'|s,a)$ を用いて期待値を計算する
- 状態遷移モデル $\widehat{p}(s'|s,a)$ からは無限に標本を生成できる
- どうやって状態遷移確率を推定するか？

最小二乗条件付き確率密度推定

$$\min_{\boldsymbol{\alpha}} \iiint \left(q_{\boldsymbol{\alpha}}(\boldsymbol{s}'|\boldsymbol{s},a) - p(\boldsymbol{s}'|\boldsymbol{s},a) \right)^2 p(\boldsymbol{s},a)\mathrm{d}\boldsymbol{s}\mathrm{d}a\mathrm{d}\boldsymbol{s}'$$

- 条件付き確率密度のモデル $q_{\boldsymbol{\alpha}}(\boldsymbol{s}'|\boldsymbol{s},a)$ を真の条件付き確率 $p(\boldsymbol{s}'|\boldsymbol{s},a)$に最小二乗法で直接適合させる
- うまくモデルを選べば，解析的かつ効率よく解が計算できる

講義の流れ

1. 強化学習の定式化
2. 動的計画法
3. 政策反復法
4. 政策探索法
5. モデルに基づく強化学習
6. 逆強化学習

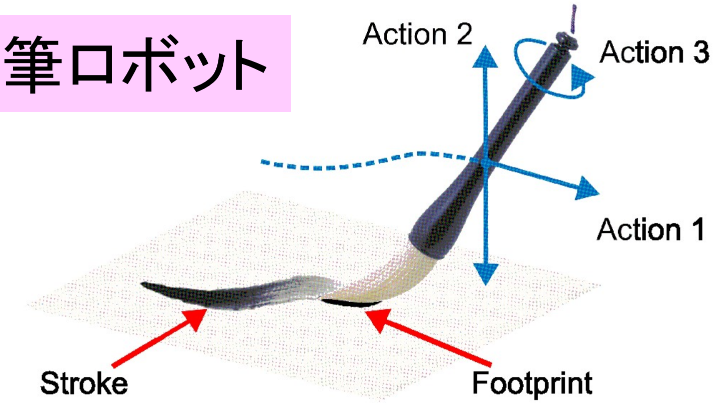
報酬関数の設計

- 強化学習では期待報酬和を最大にするように政策を学習
- 実問題では報酬の設計が難しい
- 逆強化学習：報酬関数をデータから学習する
  - 報酬関数をモデル化： $R_{\beta}(s_t, a_t, s_{t+1})$
  - “良い”行動をいくつか教え，良い行動に対する報酬が大きくなるようにパラメータ β を学習

コンピュータ・アートへの応用

- ユーザの描画データから“画風”を学習

筆ロボット



講義の流れ

1. 強化学習の定式化
2. 動的計画法
3. 政策反復法
4. 政策探索法
5. モデルに基づく強化学習
6. 逆強化学習

まとめ

- 強化学習では，報酬の情報を元にエージェントが自動的に行動規則を最適化する
- ユーザは報酬を設計するだけでよい
  - 報酬の設計が難しい場合は逆強化学習を使う
- データからの政策学習に様々な機械学習技術が活用できる