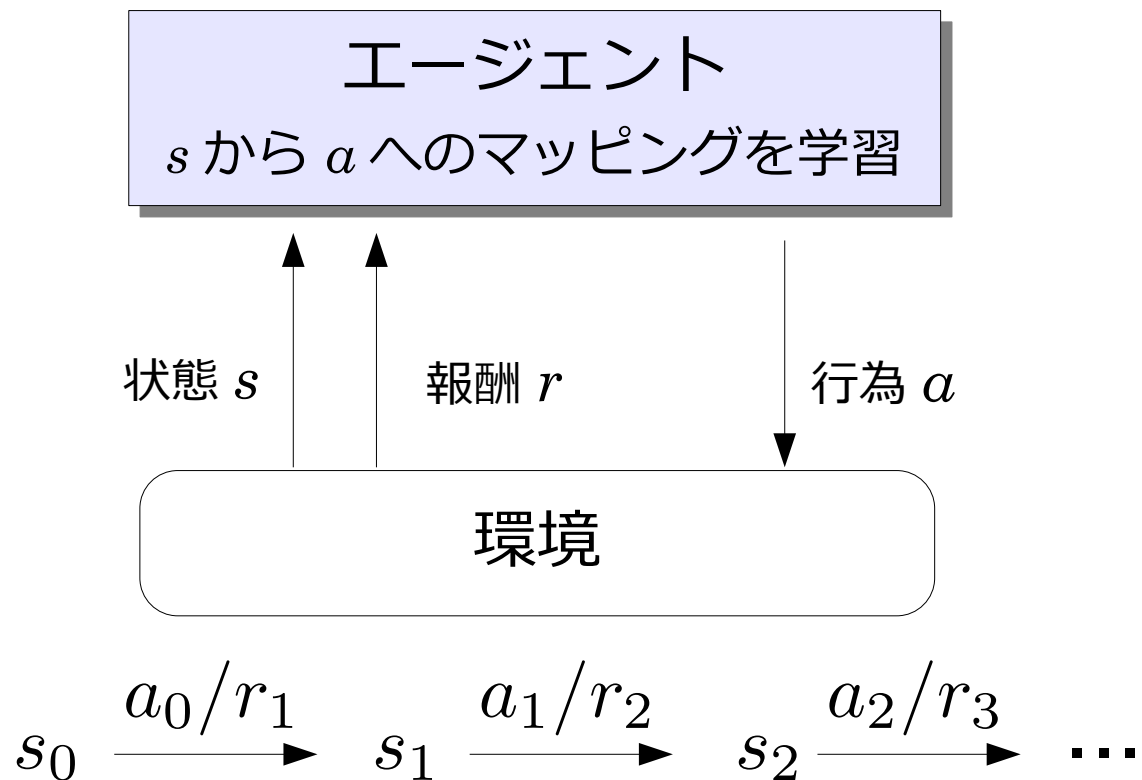


15. 強化学習

15.1 強化学習とは

- 強化学習の設定

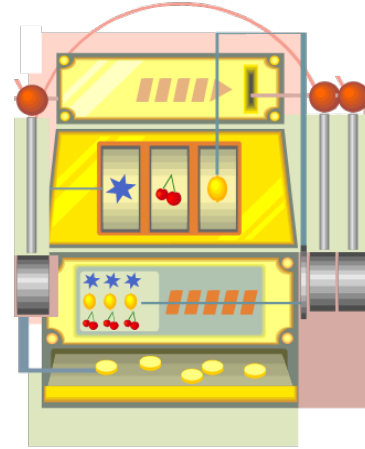
- 教師信号が間接的
- 報酬が遅れて与えられる
- 探索が可能
- 状態が非確定的な場合がある



15.2 1 状態問題の定式化 -K-armed bandit 問題-

- K-armed bandit の定義

- K 本の腕を持つスロットマシン
- i 番目の腕を引く行為 : a_i
- その行為の価値 : $Q(a_i)$



- 報酬 r が確定的な場合

- 全ての可能な a_i を試み、 $Q(a_i) = r(a_i)$ が最大となる a_i を探す

- 報酬 r_t が確率的な場合

$$Q_{t+1}(a_i) = Q_t(a_i) + \eta(r_{t+1}(a_i) - Q_t(a_i))$$

η は t の増加に伴って、減少させる

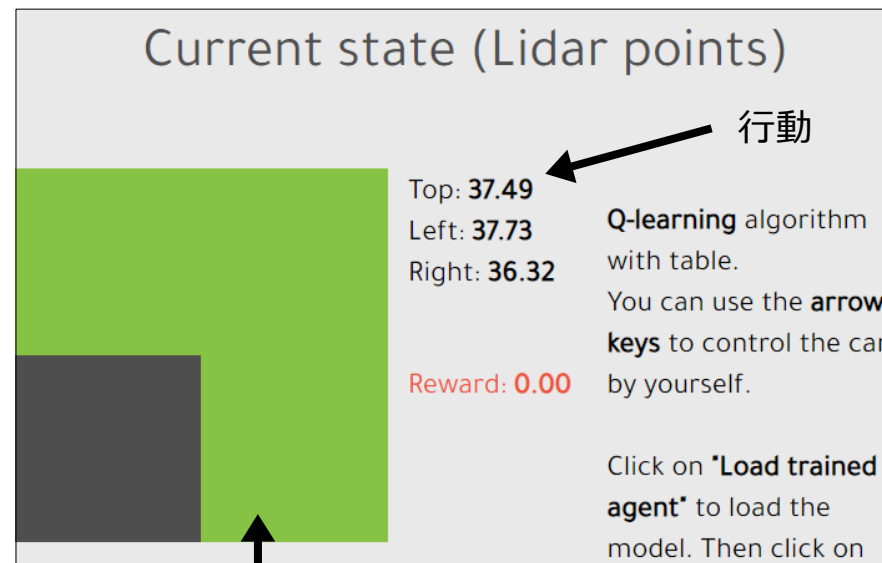
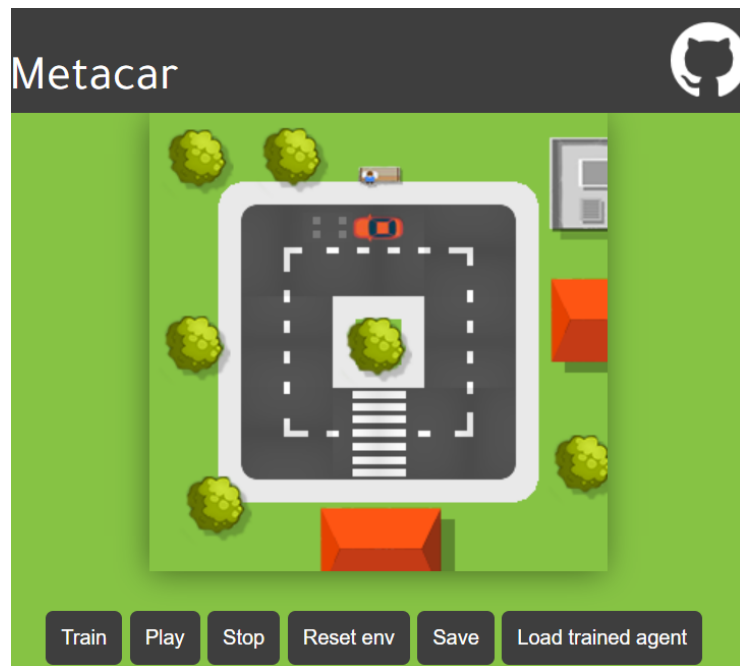
15.2 1 状態問題の定式化 -K-armed bandit 問題 -

- どのように a_i を選ぶか
 - 常に $Q_t(a_i)$ が最大のもものを選ぶ
 - もっと良い行為があるのに見逃してしまうかもしれない
 - いろいろな a_i を何度も試みる
 - 無駄な行為を何度も行ってしまうかもしれない
- ϵ -greedy 法
 - 確率 $1-\epsilon$ で最良の行為を選び、確率 ϵ でランダムに行為を選ぶ
- Boltzmann 分布を利用した方法
 - 温度 k を導入し、 k が下がるにつれて確率的振る舞いが少なくなるようにする

15.3 マルコフ決定過程による定式化

- 複数状態を持つ環境の例： Metacar

<https://www.metacar-project.com/qtable.html>



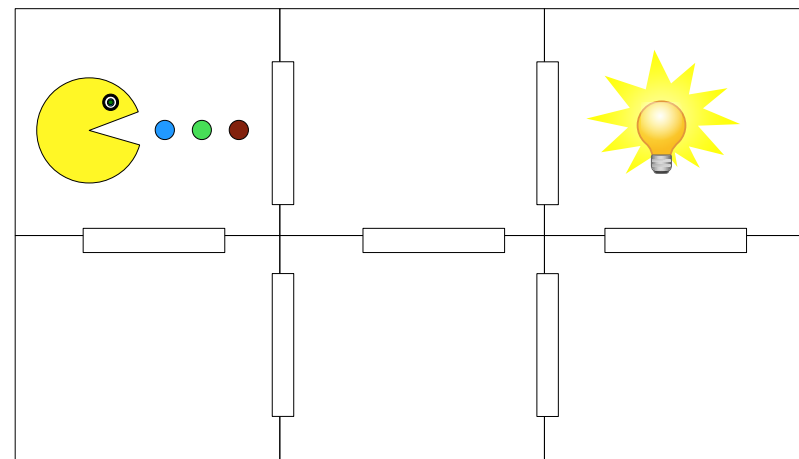
Top: 前進
Left: 左
Right: 右

車の前方の状態

1. Load trained agent でモデルをロード
2. Play で車をスタート
3. 左右矢印で走行を妨害

15.3 マルコフ決定過程による定式化

- マルコフ決定過程
 - 状態遷移を伴う問題の定式化
 - 時刻 t における状態 $s_t \in S$
 - 時刻 t における行為 $a_t \in A(s_t)$
 - 報酬 $r_{t+1} \in \mathbb{R}$
確率分布 $p(r_{t+1} | s_t, a_t)$
 - 次状態 $s_{t+1} \in S$
確率分布 $P(s_{t+1} | s_t, a_t)$



15.3 マルコフ決定過程による定式化

- 強化学習の学習目標
 - 最適政策 π^*
 - 状態から行為へのマッピング
 - 累積報酬の期待値が最大となる政策
 - 累積報酬の期待値

$$\begin{aligned} V^\pi(s_t) &= \mathbb{E}(r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots) \\ &= \mathbb{E}\left(\sum_{i=1}^{\infty} \gamma^{i-1} r_{t+i}\right) \end{aligned}$$

γ : 割引率 $0 \leq \gamma < 1$

15.3 マルコフ決定過程による定式化

- 最適政策に対する期待報酬

$$\begin{aligned} V^*(s_t) &= \max_{a_t} Q^*(s_t, a_t) \\ &= \max_{a_t} \mathbb{E} \left(\sum_{i=1}^{\infty} \gamma^{i-1} r_{t+i} \right) \\ &= \max_{a_t} \mathbb{E} \left(r_{t+1} + \gamma \sum_{i=1}^{\infty} \gamma^{i-1} r_{t+i+1} \right) \\ &= \max_{a_t} \mathbb{E} \left(r_{t+1} + \gamma V^*(s_{t+1}) \right) \end{aligned}$$

15.3 マルコフ決定過程による定式化

- 状態遷移確率を明示

$$V^*(s_t) = \max_{a_t} (\mathbb{E}(r_{t+1}) + \gamma \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t) V^*(s_{t+1}))$$

- Q 値による書き換え

$$Q^*(s_t, a_t) = \mathbb{E}(r_{t+1}) + \gamma \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t) \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1})$$

ベルマン方程式

15.4 モデルベースの手法

- 環境のモデル（状態遷移確率、報酬の確率分布）が与えられた場合の Q 値の求め方

Algorithm 15.1 Value iteration アルゴリズム

$V(s)$ を任意の値で初期化

repeat

 for all $s \in S$ do

 for all $a \in A$ do

$$Q(s, a) \leftarrow E(r|s, a) + \gamma \sum_{s' \in S} P(s'|s, a) V(s')$$

 end for

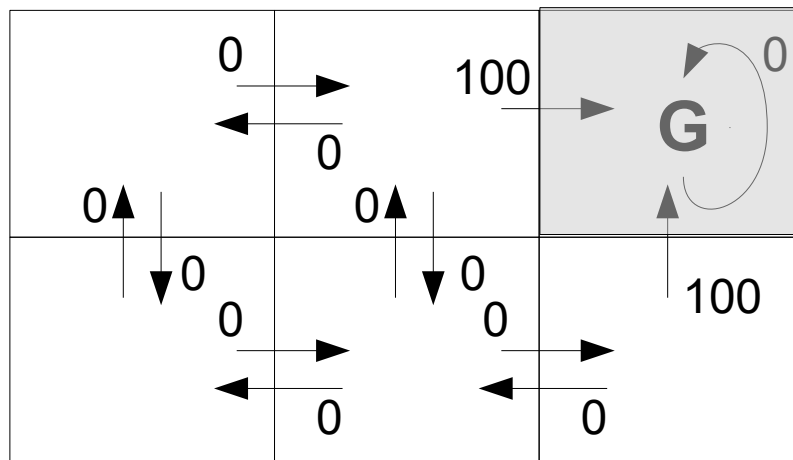
$$V(s) \leftarrow \max_a Q(s, a)$$

 end for

until $V(s)$ が収束

15.5 TD 学習

- 報酬と遷移が決定的な TD 学習



- ベルマン方程式

$$Q(s_t, a_t) = r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})$$

15.5 TD 学習

Algorithm 15.2 TD 学習 (報酬と遷移が決定的な場合)

$Q(s, a)$ を 0 に初期化

for all エピソード **do**

repeat

 探索基準に基づき行為 a を選択

 行為 a を実行し, 報酬 r と次状態 s' を観測

 以下の式で Q を更新

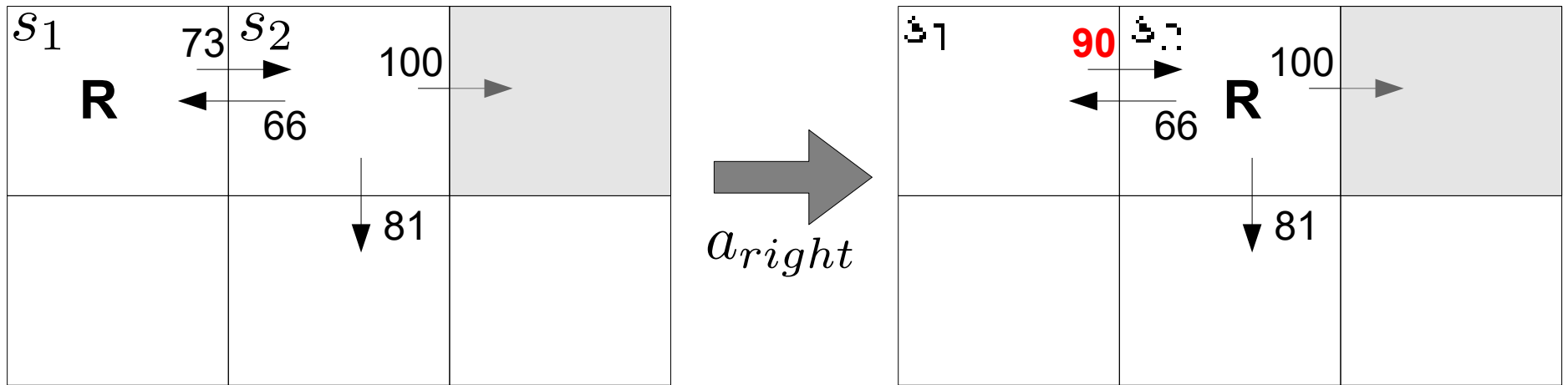
$$Q(s, a) \leftarrow r + \gamma \max_{a'} Q(s', a')$$

$s \leftarrow s'$

until s が終了状態

end for

15.5 TD 學習



$$\begin{aligned}
 Q(s_1, a_{right}) &\leftarrow r + \gamma \max_{a'} Q(s_2, a') \\
 &\leftarrow 0 + 0.9 \max\{66, 81, 100\} \\
 &\leftarrow 90
 \end{aligned}$$

15.5 TD 学習

- 報酬と遷移が確率的な TD (Temporal Difference) 学習

- ベルマン方程式

$$Q(s, a) \leftarrow Q(s, a) + \eta \left(\underbrace{r + \gamma \max_{a'} Q(s', a')}_{\text{TD 誤差}} - \underbrace{Q(s, a)}_{\text{TD 誤差}} \right)$$

- 理論的には、各状態に無限回訪問可能な場合に収束
- 実用的には無限回の訪問は不可能なので、状態推定関数等を用いて、複数の状態を同一とみなす等の工夫が必要

15.7 深層強化学習

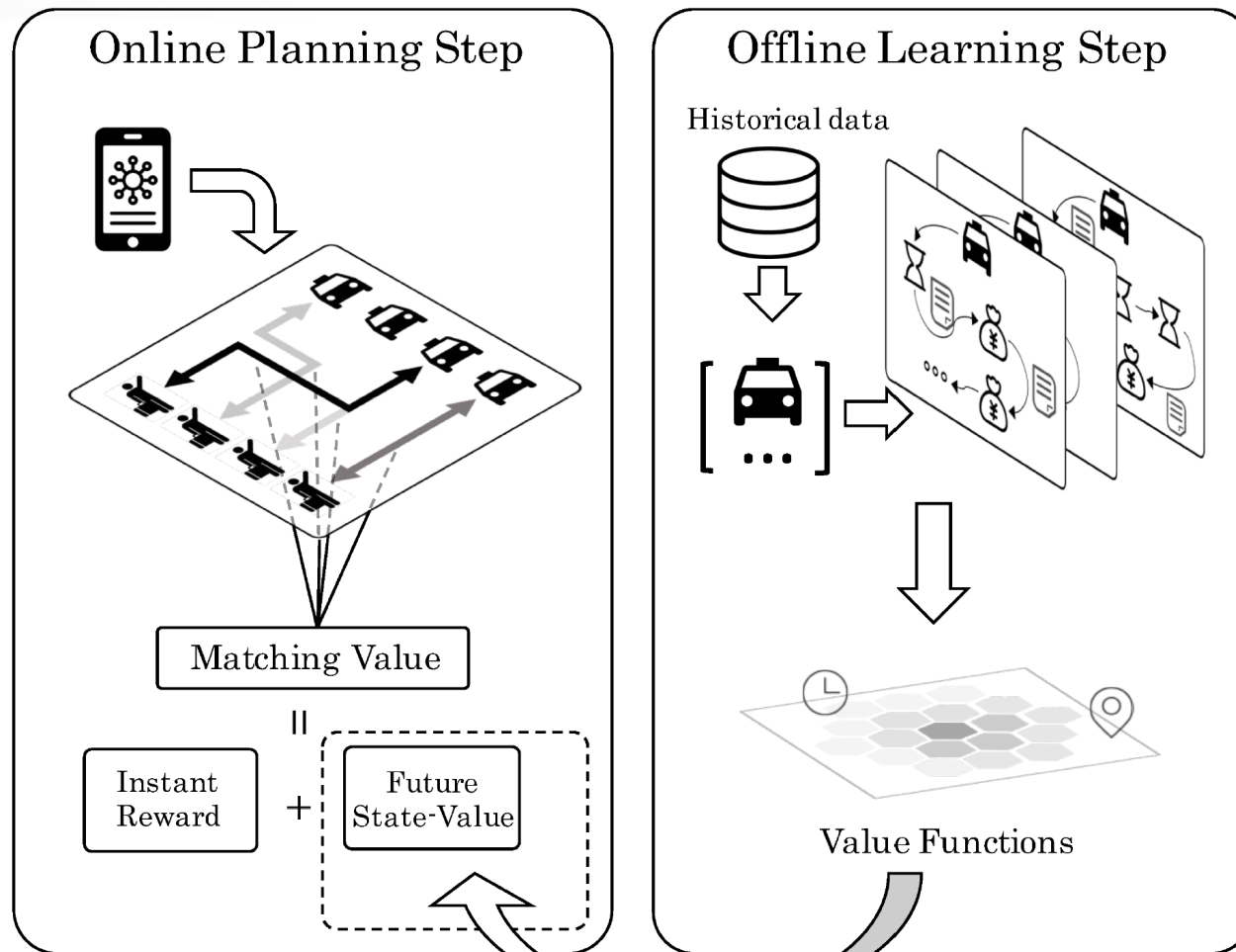
- $Q(s, a)$ の推定に DNN を用いる
 - ネットワークの誤差に TD 誤差を用いる
 - 一部の問題においては、状態を推定しなくとも局面そのものをネットワークの入力にできる
 - 例) ゲーム

事例紹介

- 滞納債務の取り立て（IBM）
 - 債務者に対するアプローチ（督促状、電話など）の手順を強化学習の枠組で学習
 - ニューヨーク州の徴税部門で 2009 年 12 月から稼働しており、3 年間で 100 万ドル以上の税収増の効果があると見積もられている
- Q 学習を適用した大車輪運動の獲得（横国大他）
 - <https://www.youtube.com/watch?v=3jsfuL9p2SQ>
- AlphaGo
 - Deep Q-Network を利用

事例紹介

- タクシー配車サービス Didi での配車最適化



過去のデータから
時間・地域の組合せ
を状態として価値
関数を学習

「即時報酬 + 将来の期待収益」の最大化

事例紹介

- 配車最適化モデルの転移学習
 - 一つの都市で学習したモデルを他の都市に適用
 - 価値関数を DNN で近似
 - 時間・地域に新たな特徴（需要、供給等）を加える
 - 参考資料
<https://speakerdeck.com/pacocat/reinforcement-learning-applications-in-taxi-dispatching-and-repositioning-domain>

参考図書

- 久保隆宏：Python で学ぶ強化学習 入門から実践まで，講談社 2019.

