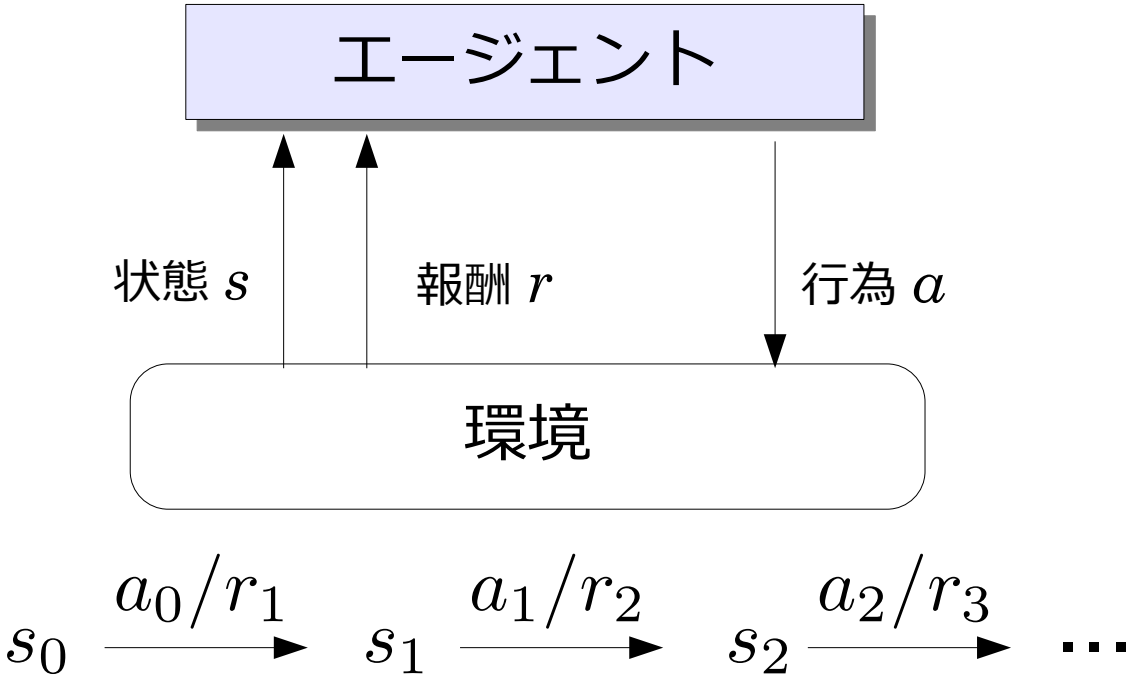
15. 強化学習

15.1 強化学習とは

- 強化学習の設定
  - 環境内に置かれたエージェントが、状態から行為への最適なマッピングを学習する



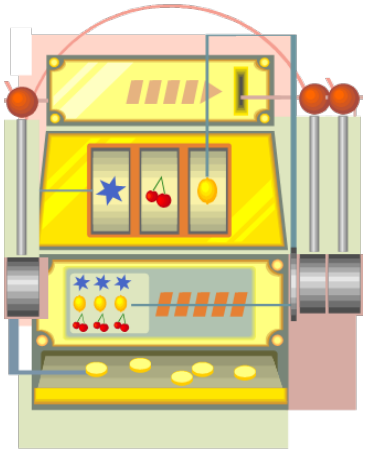
15.1 強化学習とは

- 強化学習の位置づけ：中間的学習
  - 正解は逐一与えられず、時間遅れを伴った報酬として、出力へのフィードバックが与えられる
- 強化学習の定義
  - 報酬を得るために，環境に対して何らかの行為を行う意思決定エージェントの学習
  - 時刻は離散的に進む
- 報酬仮説
  - すべての目標は累積期待報酬最大化で記述できる

ある時刻 t でエージェントがいかにうまく振る舞ったかを示すスカラー値

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 $(i=1,...,K)$
  - （即時）報酬： $r(a_i)$
  - 行為の価値 :$Q(a_i)$
    - 報酬が確定的な場合
      - すべての a_i を 1 度試み、 $Q(a_i) = r(a_i)$ が最大となる a_i を求める
    - 報酬が確率的な場合
      - すべての a_i を何度か試み、 $Q(a_i)$ = （ $r(a_i)$ の平均値 ） が最大となる a_i を求める



15.2 1 状態問題の定式化 -K-armed bandit 問題 -

- 時刻 t での報酬の平均値 $Q_t(a_i)$ の計算

$$
\begin{aligned}
Q_t(a_i) &= \frac{1}{t}\sum_{j=1}^{t} r_j(a_i) \\
&= \frac{1}{t}\left(r_t(a_i) + \sum_{j=1}^{t-1} r_j(a_i)\right) \\
&= \frac{1}{t}\left(r_t(a_i) + (t-1)Q_{t-1}(a_i)\right) \\
&= Q_{t-1}(a_i) + \frac{1}{t}(r_t(a_i) - Q_{t-1}(a_i))
\end{aligned}
$$

- Q 値のインクリメンタルな更新式

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

学習率 η は t の増加に伴って減少させるべきだが、
t が大きいとき、定数として扱える

15.2 1 状態問題の定式化 -K-armed bandit 問題 -

- どのように行為 a_i を選ぶか　探索と活用のトレードオフ
  - 常に $Q_t(a_i)$ が最大のものを選ぶ
    - 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 ε でランダムに行為を選び、残りの確率 $1\text{-}\varepsilon$ でもっとも欲張りな行為を選ぶ

15.3 マルコフ決定過程による定式化

- 状態が複数あるときの強化学習
  - 環境にマルコフ性を仮定
    - 遷移先の状態：直前の状態とそこでの行為のみに依存
    - 報酬：直前の状態と遷移先のみに依存
  - 目標：累積期待報酬を最大とする、状態から行為へのマッピング関数の獲得

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　$p(r_{t+1} \mid s_t, a_t)$
  - 次状態　$s_{t+1} \in S$
    確率分布　$P(s_{t+1} \mid s_t, a_t)$

15.3 マルコフ決定過程による定式化

- 問題の具体例： FrozenLake-v0

  https://gym.openai.com/envs/FrozenLake-v0/

  - エージェントは 4×4 のタイル上で初期状態 S からゴール G を目指して移動する
  - F (Frozen) の状態は歩行可能（ただし滑る）
  - H (Hole) の状態では、穴に落ちてエピソードは終了する
  - 報酬の例
    - G: 1
    - H: -1

S	F	F	F
F	H	F	H
F	F	F	H
H	F	F	G

15.3 マルコフ決定過程による定式化

- 学習目標
  - 最適政策 π^* の獲得
    - 政策 π ：状態から行為へのマッピング
    - 累積報酬の期待値（＝将来の平均）が最大となる政策が最適政策
- 状態価値関数
  - 時刻 t で状態 s_t にいて、その後、政策 π に従って行動したときに得られる累積報酬の期待値

$$V^{\pi}(s_t) = \mathbb{E}(r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots)$$

$$= \mathbb{E}(\sum_{i=1}^{\infty} \gamma^{i-1} r_{t+i})$$

γ : 割引率　$0 \leqq \gamma < 1$

15.3 マルコフ決定過程による定式化

- 1 状態先の状態価値関数を用いた定義

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

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

$$\mathrm{V}^*(s_t) = \max_{a_t} Q^*(s_t, a_t)$$

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

ベルマン方程式

15.4 モデルベースの手法

- 強化学習の目標： Q 値の推定
- 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：モデルベースの方法
  - 基本的には動的計画法
- 環境のモデルが与えられない場合：モデルフリーの方法
  - 得られた報酬に基づき、順次 Q 値を更新

15.4 モデルベースの手法

- モデルベースの Q 値の求め方

Algorithm 15.1 Value iteration アルゴリズム

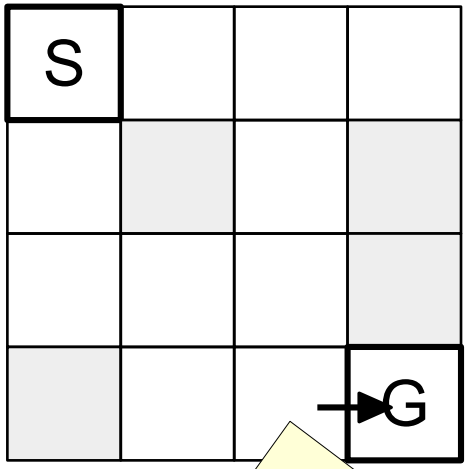
$V(s)$ を任意の値で初期化
repeat
  for all $s \in S$ **do**
    for all $a \in A$ **do**
      $Q(s,a) \leftarrow E(r|s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V(s')$
    end for
    $V(s) \leftarrow \max_a Q(s,a)$
  end for
until $V(s)$ が収束

15.5 TD 学習

- モデルフリー学習
  - エージェントが探索しながら、得られる報酬に基づいて Q 値を更新
  - 初期状態から終了状態に至る過程をエピソードとよぶ
- Q 値を更新するタイミングに基づく分類
  - エピソードが終了してから更新：モンテカルロ法
  - 一定範囲先の報酬を用いて更新： TD 学習
    - TD: Temporal Difference

15.5 TD 学習

- 報酬と遷移が決定的な TD 学習



この状態で行為 a_{right} を行えば、必ず右のタイルに移動し、報酬 1 が得られる

- 報酬と遷移が決定的な場合のベルマン方程式

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

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 学習

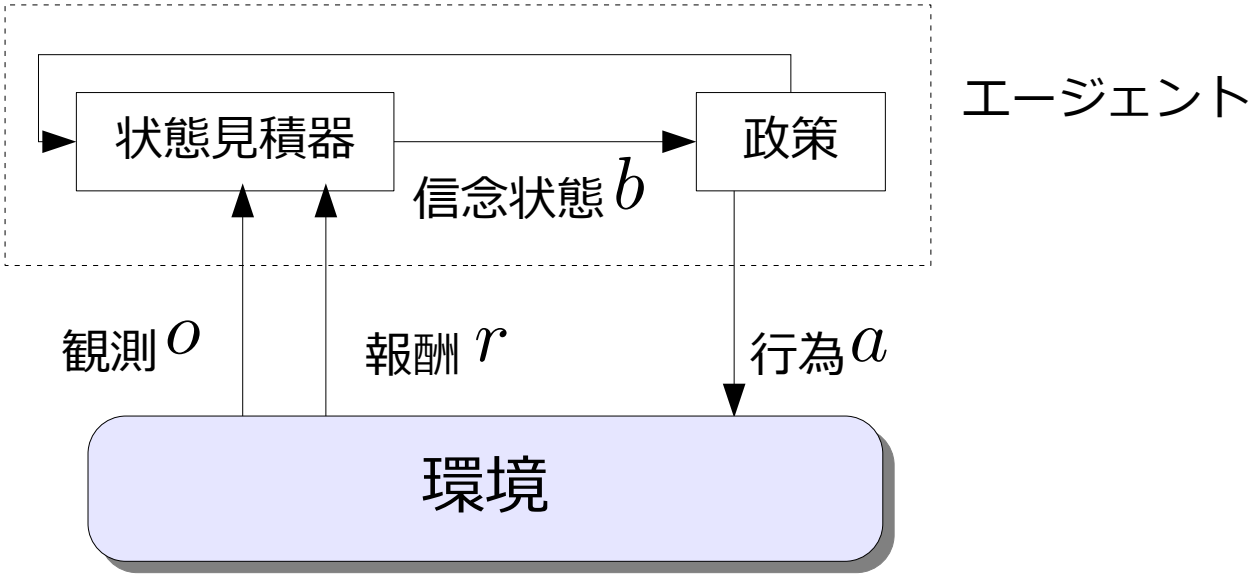
- 報酬と遷移が確率的な TD 学習
  - ベルマン方程式

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

TD 誤差

    - 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

15.6 部分観測マルコフ決定過程による定式化



- 状態 s_t で行為 a_t を行うと観測 o_{t+1} が確率的に得られる
- エージェントは状態の確率分布を信念状態 b_t として持つ
- エージェントは、信念状態 b_t 、行為 a_t 、観測 o_{t+1} から次の信念状態 b_{t+1}を推定する状態見積器 (state estimator) を内部に持つ

15.7 深層強化学習

- 政策関数による状態価値関数の表現

$$V^{\pi}(s_t) = \sum_{a} \pi(a \mid s) \times Q(s_t, a_t)$$

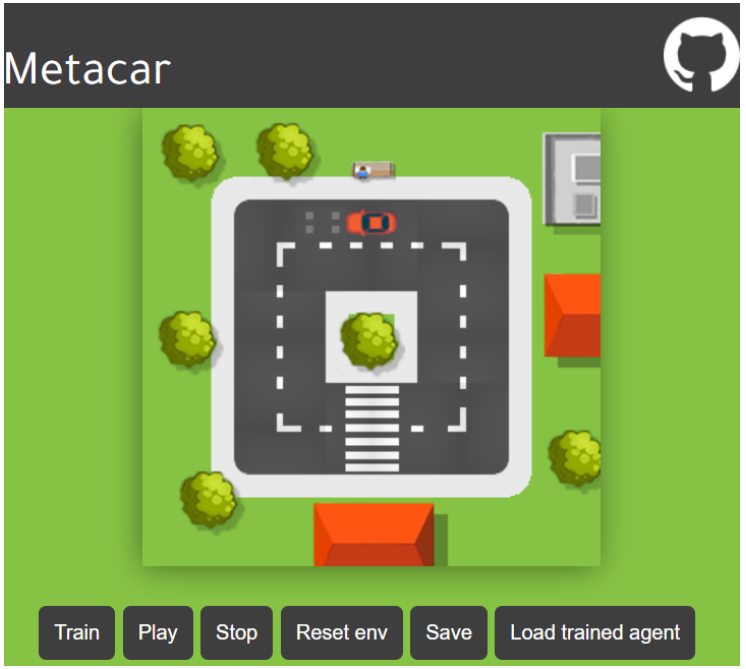
- 価値関数勾配法
  - $Q(s, a)$ の推定に DNN を用いる
  - DNN の学習のための誤差に TD 誤差を用いる
- 方策関数勾配法
  - $\pi(a \mid s)$ の推定に DNN を用いる
  - V を最大とするようにパラメータを修正する

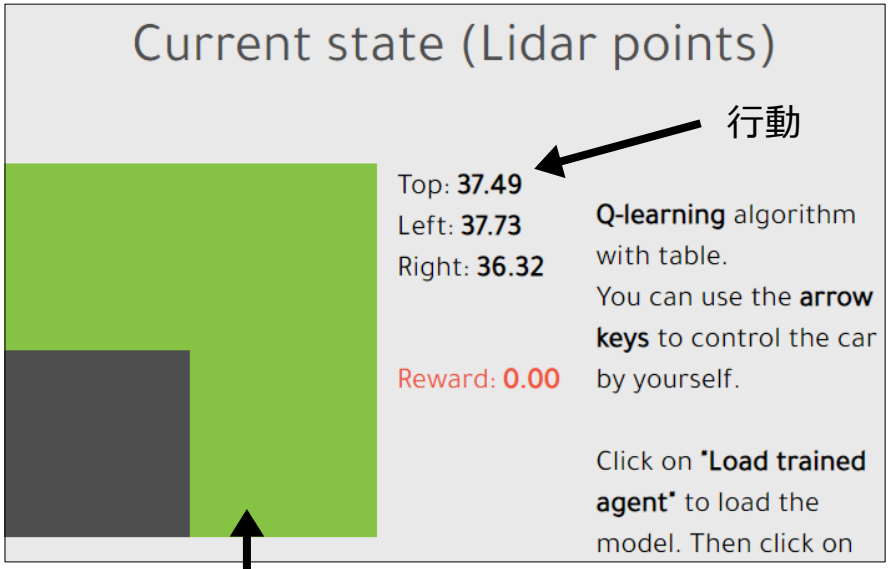
補足

デモ

- 複数状態を持つ環境の例： Metacar

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





行動

Top: 前進
Left: 左
Right: 右

車の前方の状態

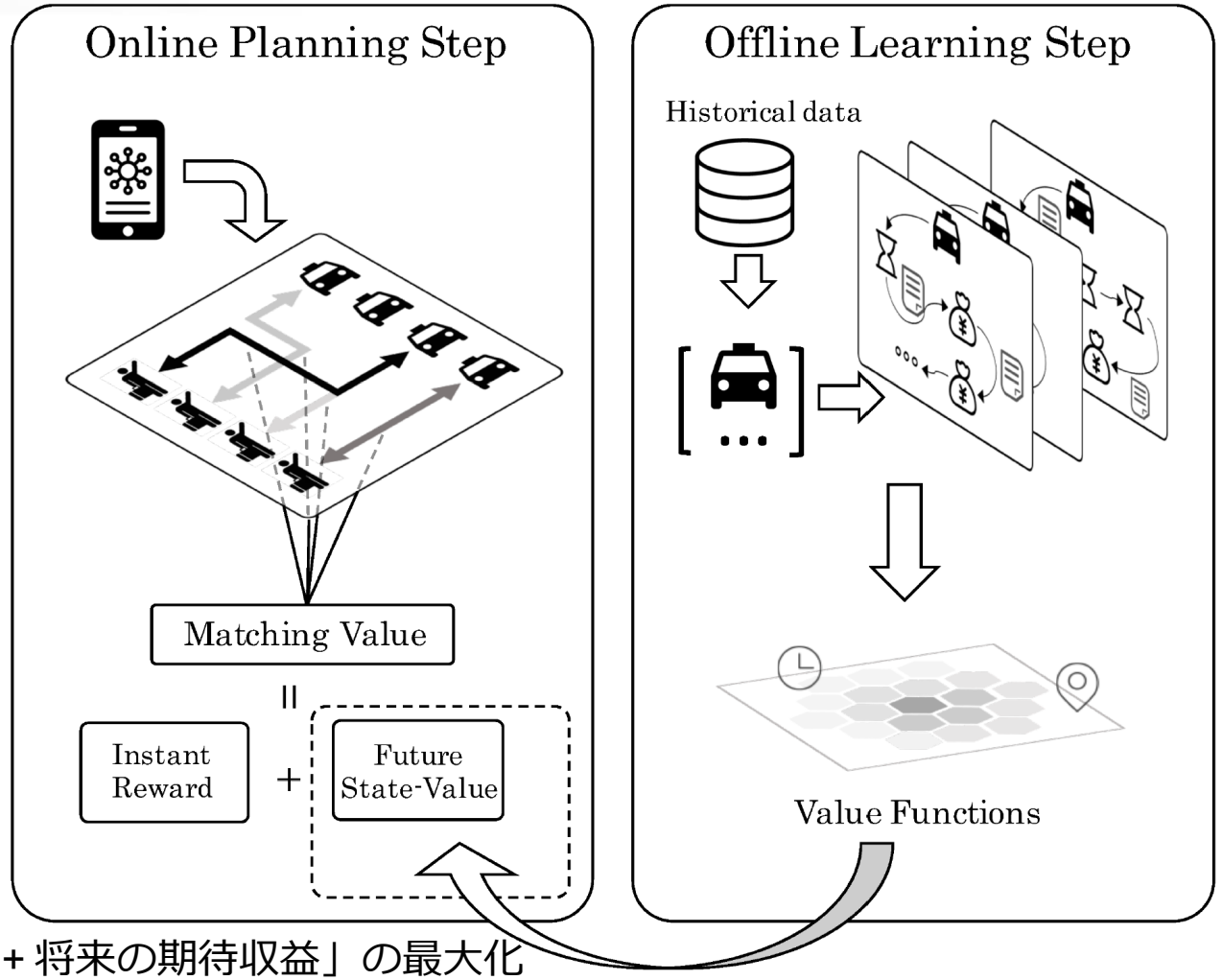
1. Load trained agent でモデルをロード
2. Play で車をスタート
3. 左右矢印で走行を妨害

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