強化確率ロボットモデルでの
逆強化学習について

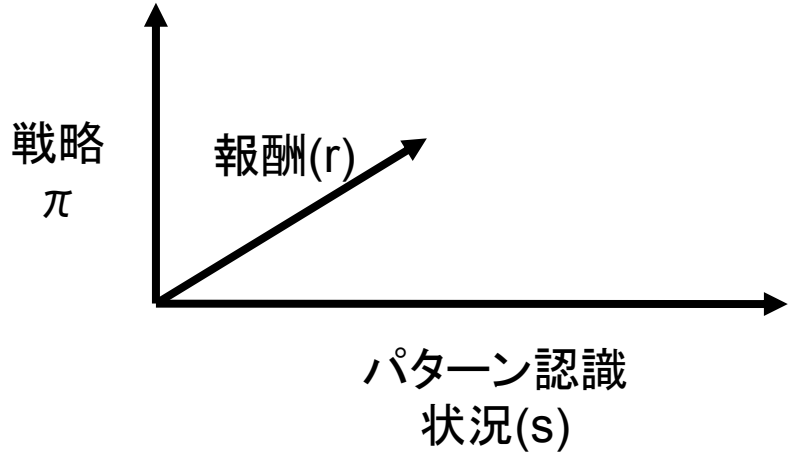
2016/01/31

@mabonakai0725

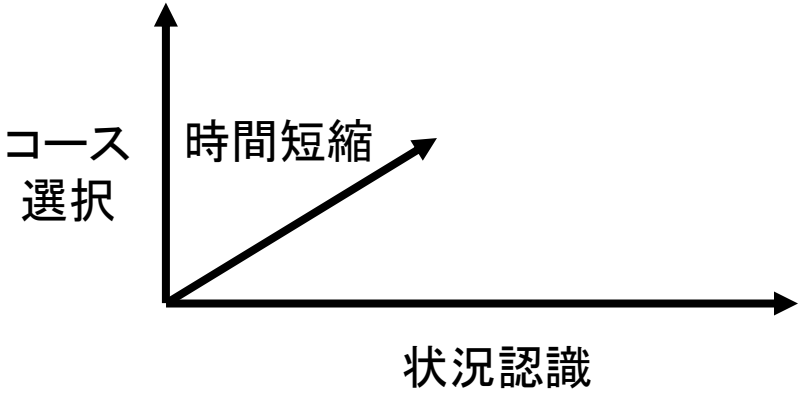
趣旨

1. 自立的ロボット（工作機械、自動運転）は PO-MDFのアルゴリズムで一般に記述できる
   POMDF:Partially Observable Markov Dicsion Process
2. MDFはML(パターン認識)に報酬と戦略の要素を入れた最適化技術である
3. POは部分的な観察により確信度を更新するベイズモデル
4. 最近は報酬も戦略も行動データから学習できる（報酬を学習する逆強化学習論文紹介）
   Googleの碁ロボット(AlphaGo)も同様の方法

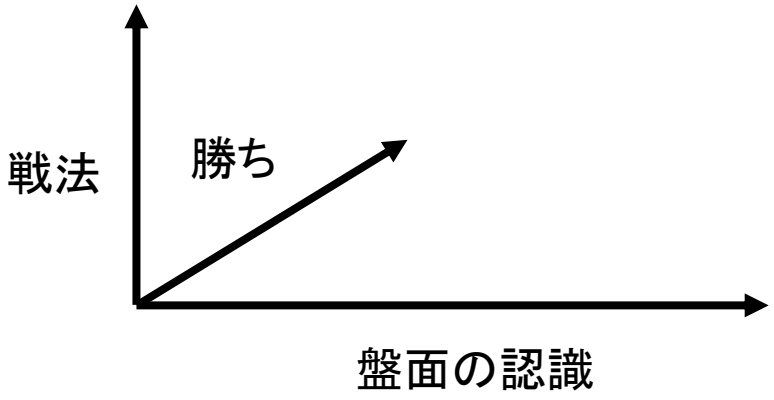
強化学習とパターン認識



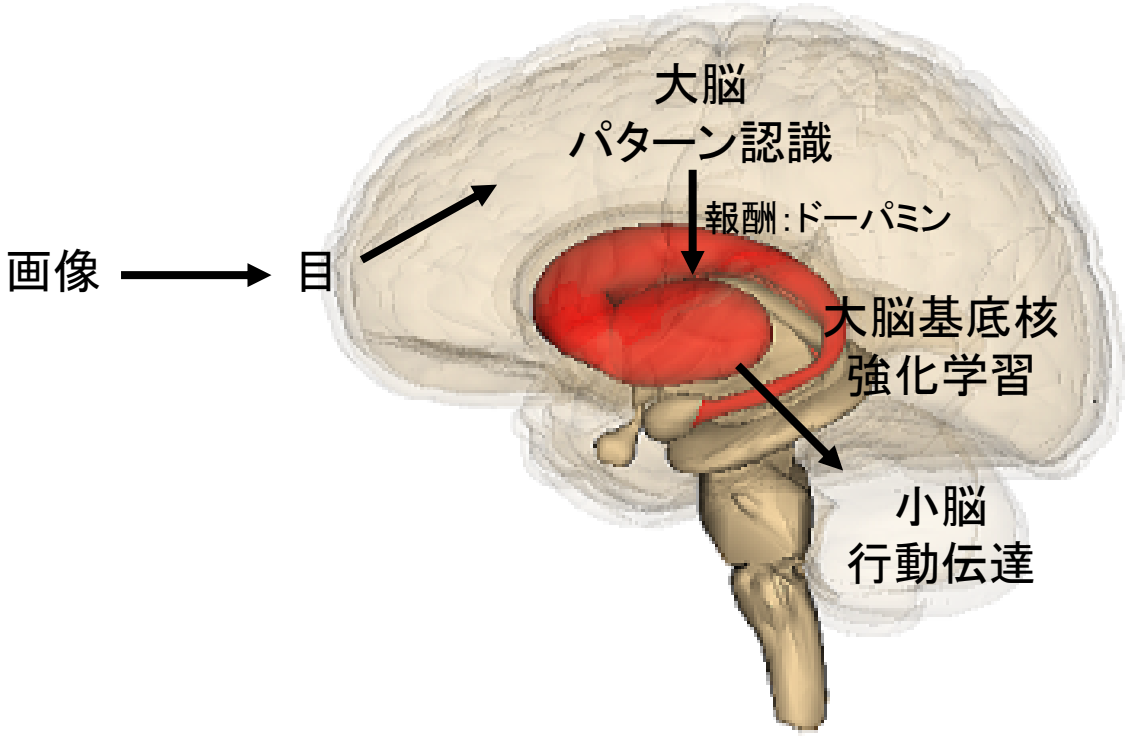
自動運転



囲碁ロボット



脳と強化学習の類似



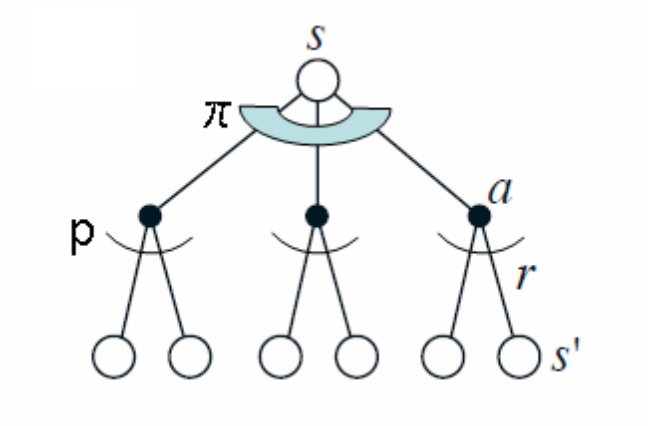
出典画像: Anatomography

大脳基底核の脳波と強化学習の価値関数のプロット図が同形を示す

銅谷賢治

強化学習

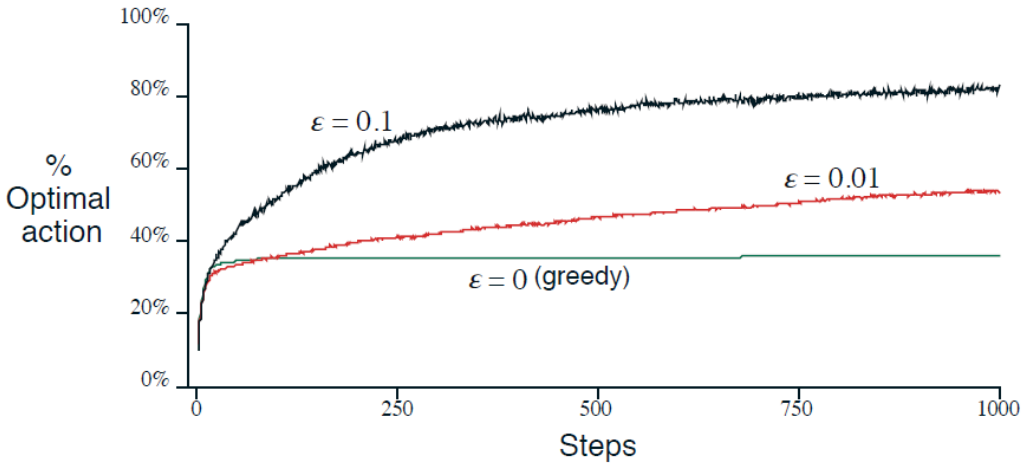
- 状況(s)で複数の行動(a)毎に報酬(r)があり、戦略(π)で行動を選択するモデル
- 行動(a)の選択は現在の状況(s)のみで決定される(マルコフ決定過程)
- 状況(s)に於いて将来の報酬(r)の合計が最大と予測する行動(a)を選択する
- 戦略(π)で選択した行動(a)と異なる行動を採る



- 強化学習は将来の報酬の合計である価値関数V(s)もしくは行動価値関数Q(s,a)の算出が目的

戦略と報酬

N腕バンディット問題　スロットマシンの当たり率は異なる
M回の試行でどの機械を選ぶと最大合計報酬を得れるか



ε 戦略：当たり率の良い機械に拘らない率
実験結果は過去の成功体験に拘らない方が沢山の報酬を得る
「秀才は駄目　冒険をしよう」という心強い教え

確率モデルで柔軟性を反映する

- 将来状況(s’)　行動(a’)の漸化式で表す
- 行動(a)は戦略πで確率的にずれる
- 将来状況(s')は遷移確率pで確率的にずれる

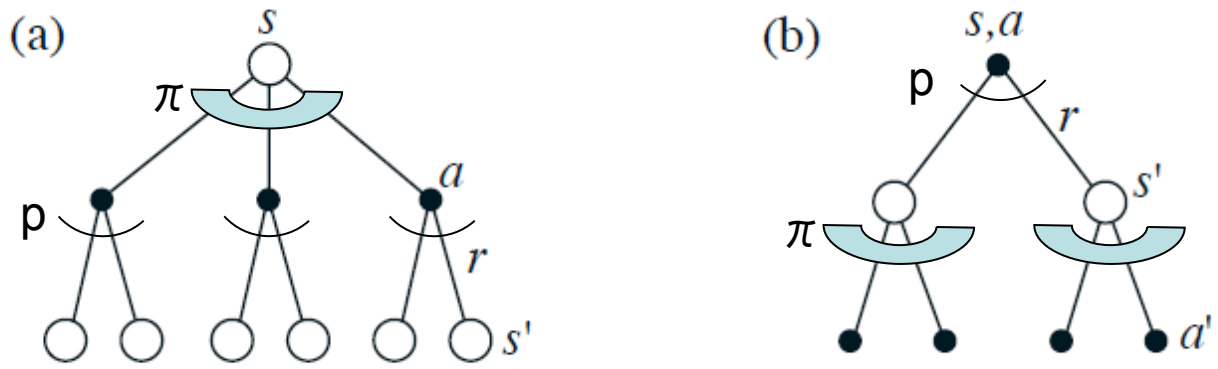
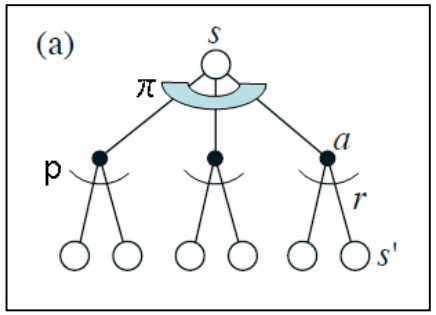


Figure 3.4: Backup diagrams for (a) v_π and (b) q_π.

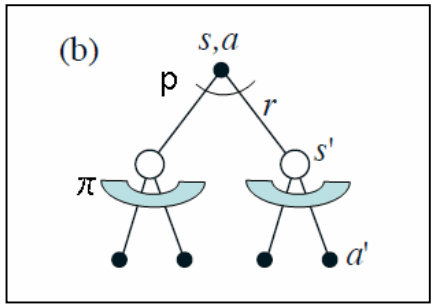
価値関数V　行動関数Q

- MDPでは次回の価値は将来の漸化式で計算

$$
\begin{aligned}
v(s) &= \mathbb{E}_\pi[R_{t+1} + \gamma v_k(S_{t+1}) \mid S_t = s] \\
&= \sum_a \pi(a|s) \sum_{s'} p(s'|s,a) \Big[r(s,a,s') + \gamma v(s') \Big],
\end{aligned}
$$



$$
\begin{aligned}
q_\pi(s,a) &= \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = a] \\
&= \sum_{s'} p(s'|s,a) \Big[r(s,a,s') + \gamma v_\pi(s') \Big].
\end{aligned}
$$



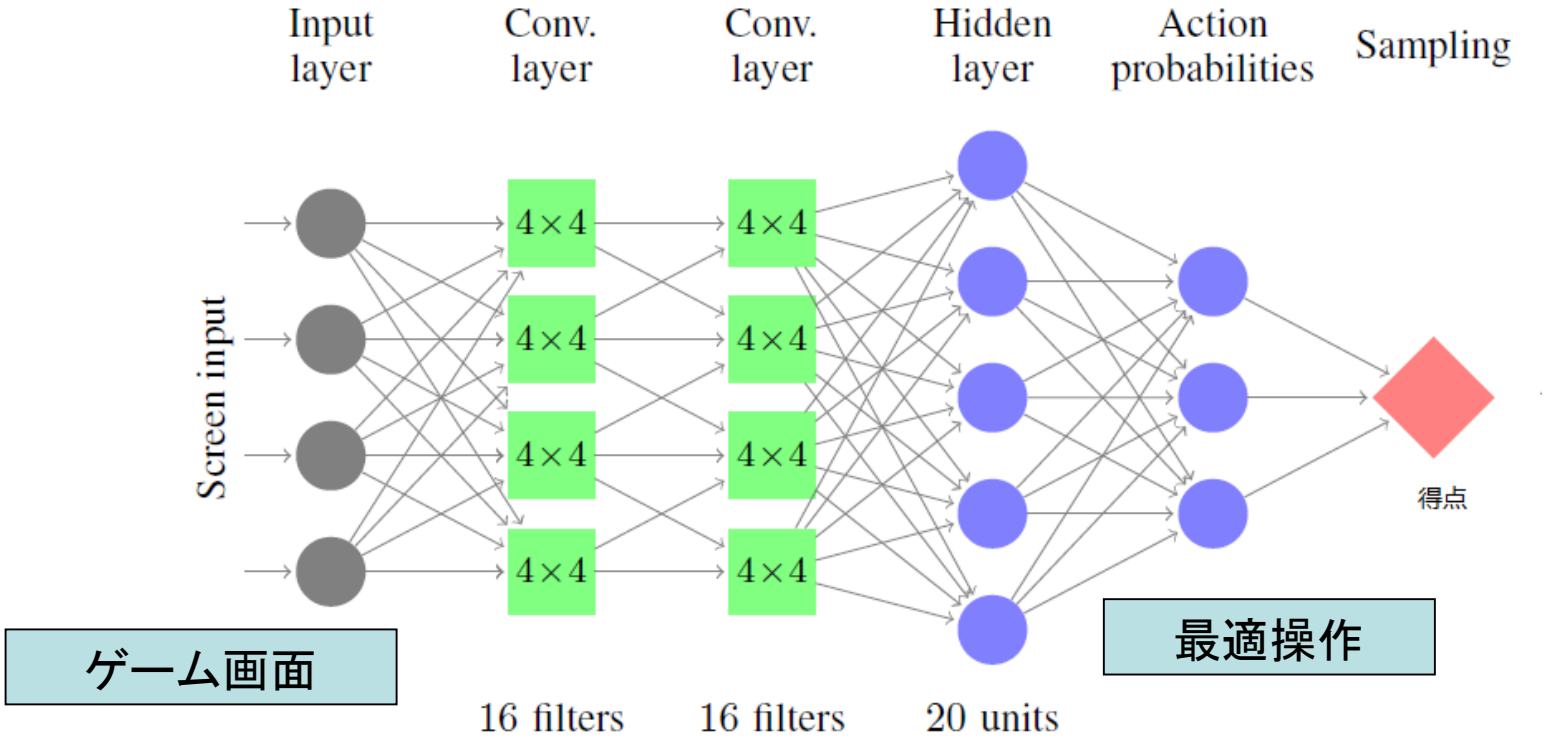
γ:価値の減少率　将来の価値は低く見積もる

価値関数V 行動関数Qの算出方法

- 将来の状況と行動の分岐を繰返し展開して末端からBackUpで関数を算出する（rollout)
- 将来価値にγ:減少率があるので無限に展開しなくて良い(n期先まで展開)。
  - 動的計画法 (遷移を定常になるまで繰返し)
  - ﾓﾝﾃｶﾙﾛ法 (ﾗﾝﾀﾞﾑに経路を辿り報酬確率を計算）
  - TD(n)法 (V関数のパラメータをSDGで計算)
  - Sarsa(n)法(Q関数のパラメータをSDGで計算)
  - ニューロ法　(Q関数をニューロで汎用化)

DQN(Deep Q-Learning Net)

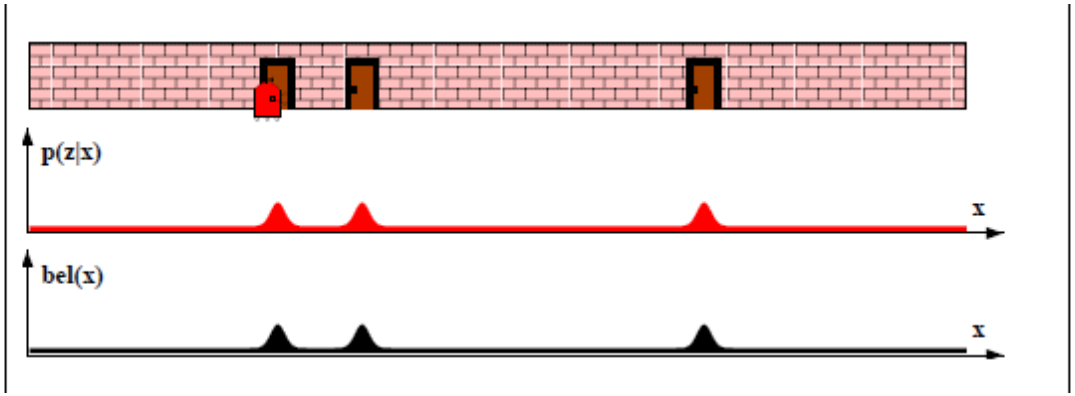
- 行動価値関数QをDeep CNNで訓練する



PO(部分観察)確率ロボットモデル

- 信念確率belを事前確率として観測結果pによる事後確率で信念確率を更新するモデル
- 事例で説明

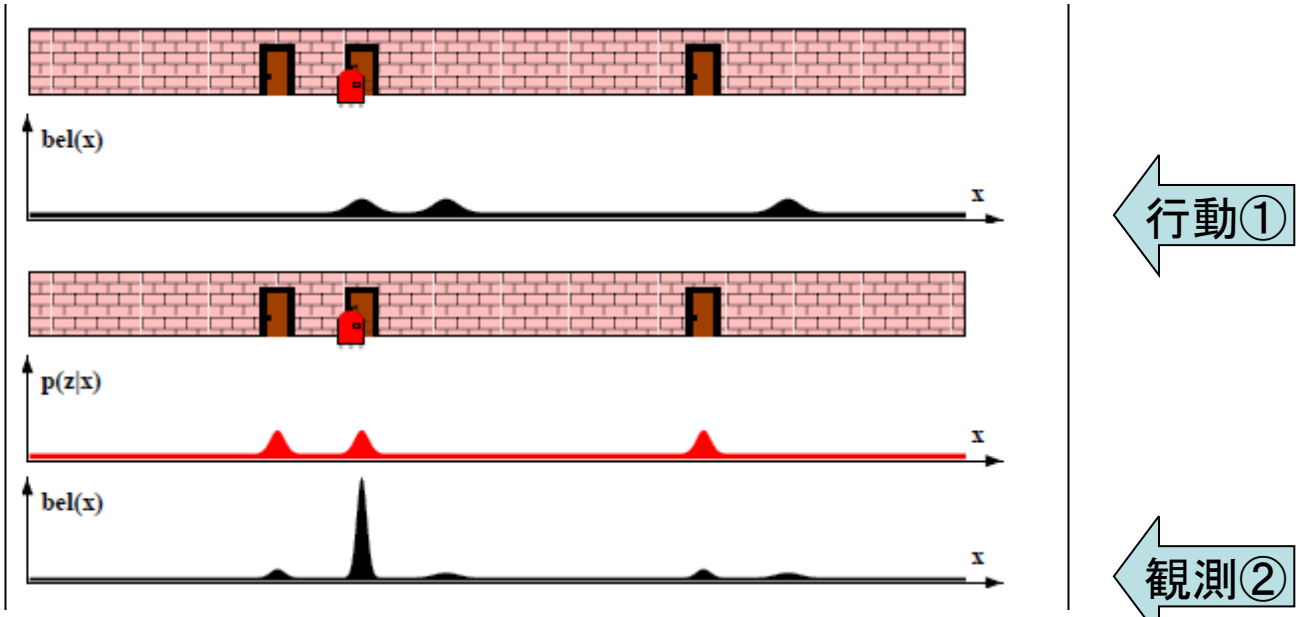
  ロボットにはドアの地図情報とドアを識別するセンサーがある



ドアの地図情報とドアの認識で自分の位置の信念確率

行動と観測による信念の更新

- 信念確率belを行動uと観測結果pで2回更新



自分の位置を観測結果で更新

ベイズ信念更新アルゴリズム

1: **Algorithm Bayes_filter**($bel(x_{t-1}), u_t, z_t$):
2: for all x_t do
3: $\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1})\ bel(x_{t-1})\ dx$
4: $bel(x_t) = \eta\ p(z_t \mid x_t)\ \overline{bel}(x_t)$
5: endfor
6: return $bel(x_t)$

Table 2.1 The general algorithm for Bayes fi ltering.

3行目	行動uによる状況xに対する信念の更新	矢印①
4行目	観測pによる状況xに対する信念の更新	矢印②

POMDP（確率ロボット+強化学習）

状況(s)での価値関数

状況(s)での戦略

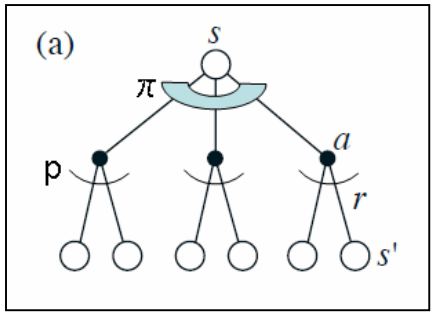
次の状況(s')での価値関数

$$v(s) = \sum_s bel(s) \sum_a \pi(a|s) \sum_{s'} p(s'|s,a) \left[r(s,a,s') + \gamma v(s') \right]$$

確信度項の追加

状況s→s'の遷移

行動(a)での報酬



逆強化学習(報酬の学習)

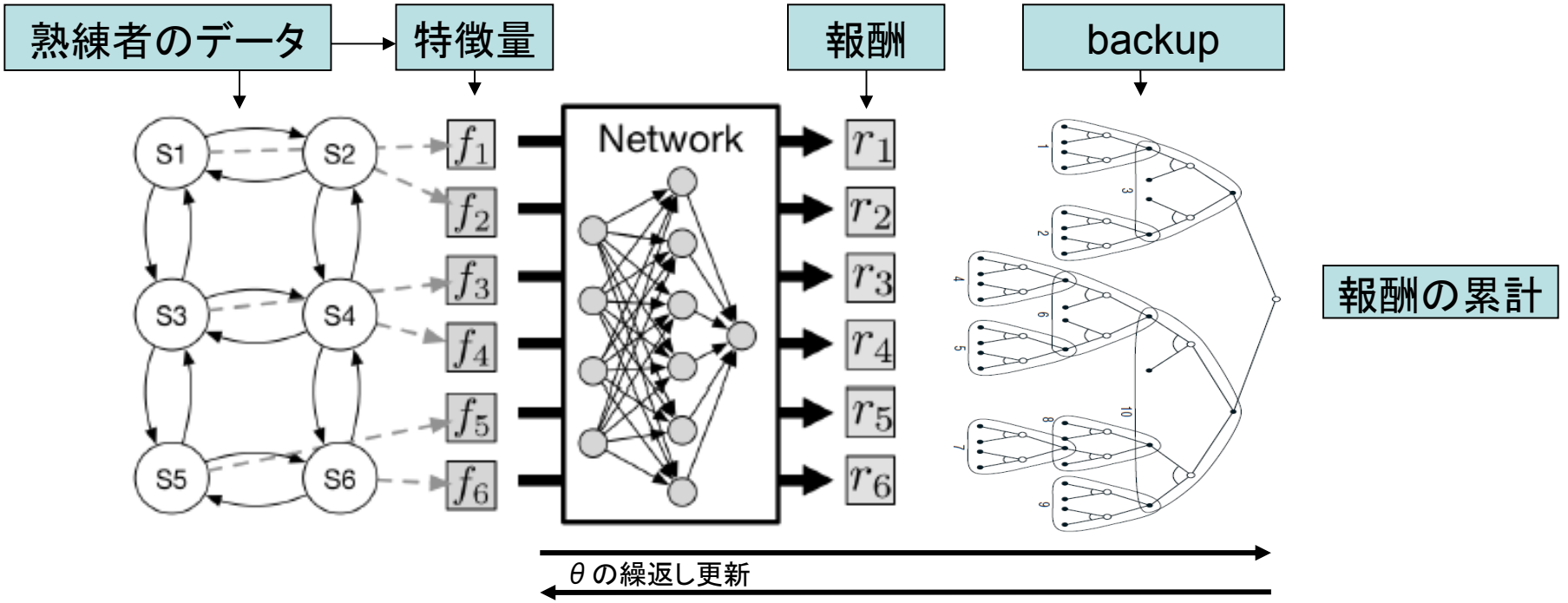
- ゲームでは報酬は得点として明らか
- 現実の世界では報酬は不明

↓

- 複数の専門家(熟練者)の行動データから報酬を学習するのが逆強化学習(IRL)
- 文献の紹介
  - Maximum Entropy Deep Inverse Reinforcement Learning (ICPR2015)
  - Inverse Reinforcement Learning with Locally Consistent Reward Functions (NIPS2015)

Maximum Entropy Deep Inverse Reinforcement Learning

報酬の累計が最大になる様に θ をDeep NNの重みとして解く



$$r = g(f, \theta).$$
$$g(f, \theta) = \theta^\top f.$$

← 報酬の関数

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial \mathcal{L}_\mathcal{D}}{\partial \theta} + \frac{\partial \mathcal{L}_\theta}{\partial \theta}.$$

← θ の微分値

$$\mathcal{L}(\theta) = \log P(\mathcal{D}, \theta | r) = \underbrace{\log P(\mathcal{D}|r)}_{\mathcal{L}_\mathcal{D}} + \underbrace{\log P(\theta)}_{\mathcal{L}_\theta}.$$

尤度の式　Dは熟練者のデータ

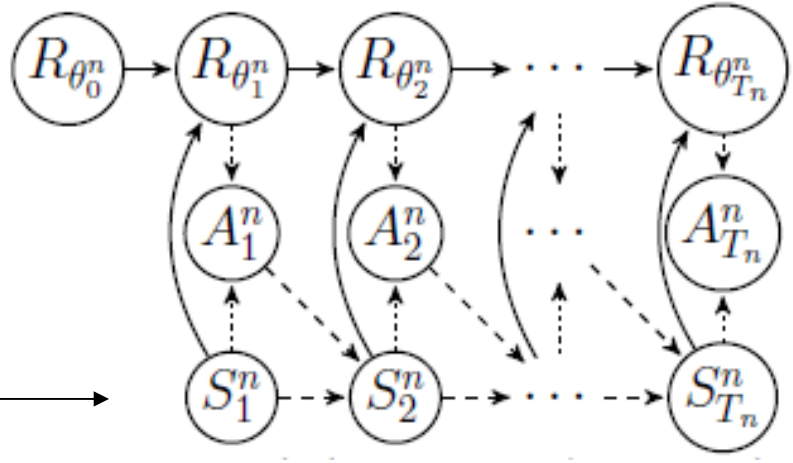
Inverse Reinforcement Learning with Locally Consistent Reward Functions

報酬の推移を隠れマルコフモデルで解く

報酬(潜在変数)

報酬関数のパラメータ
θ が潜在変数

熟練者のデータ



$r_\theta(s) \triangleq \theta^\top \phi_s$　　θ は報酬パラメータ　φ は特徴量

$$\tau_\omega(r_\theta, s, r_{\theta'}) \triangleq \begin{cases} \exp(\omega_{r_\theta r_{\theta'}}^\top \varphi_s)/(1+\sum_{r_{\bar{\theta}} \in \mathcal{R}\setminus\{r_{\bar{\theta}}\}} \exp(\omega_{r_\theta r_{\bar{\theta}}}^\top \varphi_s)) & \text{if } r_{\theta'} \neq r_{\tilde{\theta}}, \\ 1/(1+\sum_{r_{\bar{\theta}} \in \mathcal{R}\setminus\{r_{\bar{\theta}}\}} \exp(\omega_{r_\theta r_{\bar{\theta}}}^\top \varphi_s)) & \text{otherwise;} \end{cases}$$

ω はパラメータ　ψ は特徴量

報酬は2種類：下段は　rθ→rθ'の変動で切り替わる報酬

まとめ

- 限られたセンサデータから自立的に動くロボットのモデルはPOMDPで一般に記述できる
- POMDPでは、将来の価値の漸化式を解き累計価値の予測をする必要がある
- ゲームではDeepLearningを使うとゲーム画面と最高点が得た操作を対で学習させてモデル化できる
- 現実の問題では報酬(得点)が不明なので専門家（熟練者）のデータから報酬を推定できる逆強化学習の仕組みが有効である