Reinforcement Learning – Tutorial

西村　礼恩

August 9, 2018

Contents

第 1 章

Reinforcement Learning and Credit Assignment

1.1 有限マルコフ決定過程

有限マルコフ決定過程 (Markov decision process)$\mathcal{P}$ を仮定する。これは 6-タプル $\mathcal{P} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, p, \pi, \gamma)$:

- $\mathcal{S}$ は状態の有限集合; S_t 値 $s, s' \in \mathcal{S}$ を持つ時刻 t の状態の確率変数であり、離散確率分布を持つ。
- $\mathcal{A}$ は行動の有限集合 (しばしば状態依存する $\mathcal{A}(s)$); A_t は値 $s, a' \in \mathcal{A}$ を持つ時刻 t における行動の確率変数であり、離散確率分布を持つ。
- $p(S_{t+1} = s', R_{t+1} = r|S_t = s, A_t = a)$ は状態-行動の条件付き状態-報酬に対する遷移報酬分布。
- $\pi(A_{t+1} = a'|S_{t+1} = s')$ は方針であり、状態が与えられた行動に対する分布。
- $\gamma \in [0, 1]$ は割引率。

MDP は時系列

$$(S_0, A_0, R_1, S_1, A_1, R_2, \cdots,), \tag{1.1.1}$$

を生成する。周辺分布は

$$p(s', r\,|s, a) = \Pr[S_{t+1} = s', R_{r+1} = r|S_t = s, A_t = a], \tag{1.1.2}$$

$$p(r\,|s, a) = \Pr[R_{r+1} = r|S_t = s, A_t = a] = \sum_{s'} p(s', r|s, a), \tag{1.1.3}$$

$$p(s'\,|s, a) = \Pr[S_{t+1} = s'|S_t = s, A_t = a] = \sum_{r} p(s', r'|s, a). \tag{1.1.4}$$

期待値についても言及する。

- $\mathbb{E}_\pi$ は方針 π によって生成された状態、行動と報酬の MDP 系列を確率変数とする期待値。
- $\mathbb{E}_{s',s,a,r,a'}$ は確率変数 S_{t+1} が値 $s' \in \mathcal{S}$ で、S_t が値 $s \in \mathcal{S}$ で、A_t が値 $a \in \mathcal{A}$ で、A_{t+1} が値 $a' \in \mathcal{A}$ で、R_{t+1} が値 $r \in \mathcal{R}$ のときの期待値。

収益 (return)G_t は時刻 $t + 1$ から開始とする累積報酬であり:

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}. \tag{1.1.5}$$

State-Value and Action-Value Function. 方針 π と状態 s についての状態価値関数 $v^\pi(s)$ は

$$v^\pi(s) = \mathbb{E}_\pi[G_t|S_t = s] = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s], \tag{1.1.6}$$

として定義される。方針 π についての行動価値関数 $q^\pi(s,a)$ は $S_t = s$ から開始した時に行動 $A_t = a$ を取った収益の期待値である。

$$q^\pi(s,a) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a] = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a], \tag{1.1.7}$$

$$v^\pi(s) = \sum_{a \in \mathcal{A}} \pi(a \mid s)\, q^\pi(s,a) \tag{1.1.8}$$

最適行動価値関数 q_* は最適状態価値関数 v_* を通して

$$v_*(s) = \max_\pi v^\pi(s), \tag{1.1.9}$$

$$\pi_* = \arg\max_\pi v^\pi(s) \ \text{ for all } s, \tag{1.1.10}$$

$$q_*(s,a) = \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) | S_t = s, A_t = a], \tag{1.1.11}$$

と表すことができる。同様に最適状態価値関数は最適行動価値関数を用いて

$$v_*(s) = \max_a q^{\pi_*}(s,a) = \max_a \mathbb{E}_{\pi_*}[G_t | S_t = s, A_t = a] = \tag{1.1.12}$$

$$\max_a \mathbb{E}[R_{t+1} + \gamma G_{t+1} | S_t = s, A_t = a] = \tag{1.1.13}$$

$$\max_a \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) | S_t = s, A_t = a]. \tag{1.1.14}$$

Finite time horizon and no discount. – 時刻 $T+1$ のとき端末時に報酬 R_{T+1} を受け取る長さ T のエピソードのみを考慮する有限時間期間 (finite time horizon) について考える。有限時間期間 MDP は有限系列を生成し、将来報酬に割引しない。方針 π についての行動価値関数 q は

$$q^\pi(s,a) \ = \ \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a] = \mathbb{E}_\pi[\sum_{k=0}^{T-t} R_{t+k+1} \mid S_t = s, A_t = a] \tag{1.1.15}$$

$$= \ \mathbb{E}_\pi[R_{t+1} + G_{t+1} \mid S_t = s, A_t = a] \tag{1.1.16}$$

$$= \ \sum_{s',r} p(s', r \mid s, a) \left[r + \sum_{a'} \pi(a' \mid s')\, q^\pi(s', a') \right] \tag{1.1.17}$$

$$\overset{1.1.8}{=} \sum_{s',r} p(s', r \mid s, a)\, [r + v^\pi(s')] \tag{1.1.18}$$

$$= \ \mathcal{T}^\pi[s](s,a). \tag{1.1.19}$$

となる。$\mathcal{T}^\pi[s](s,a)$ を方針 π についてのベルマン作用素 (Bellman operator) という。しばしばこの作用素を単に

$$\mathcal{T}^\pi[s](s,a) = r(s,a) + \mathbb{E}_{s',a'}[q(s',a')], \tag{1.1.20}$$

とし、

$$r(s,a) = \sum_r r\, p(r \mid s, a),$$

$$\mathbb{E}_{s',a'}[q(s',a')] = \sum_{s'} p(s' \mid s, a) \sum_{a'} \pi(a' \mid s') q(s', a').$$

ベルマン方程式1.1.20から

$$q^\pi(s,a) \overset{1.1.17}{=} \sum_{s',r} r\, p(s', r \mid s, a) + \sum_{s',r} p(s', r \mid s, a) \sum_{a'} \pi(a' \mid s') q(s', a')$$

$$= \ \sum_r r\, p(r \mid s, a) + \mathbb{E}_{s',a'}[q(s',a') | s, a]$$

$$= \ r(s,a) + \mathbb{E}_{s',a'}[q(s',a') | s, a].$$

が得られる。方針 π について時刻 $t=0$ での期待収益は

$$v_0^\pi = \mathbb{E}_\pi[G_0] = \mathbb{E}_\pi[\sum_{t=0}^{T} R_{t+1}],$$
$$\pi^* = \arg\max_\pi v_0^\pi.$$

Monte Cartlo(MC) — $q^\pi(s,a)$ を推定するために、MC はデータにあるすべての観測された $(G_t \mid S_t = s, A_t = a)$ の算出平均 (arithmetic mean) を計算する。

$$(q^\pi)^{i+1}(s_t,a_t) = (q^\pi)^i(s_t,a_t) + \alpha\left(\sum_{t'=t}^{T} r_{t'} - (q^\pi)^i(s_t,a_t)\right). \tag{1.1.21}$$

この更新は *constant-α MC* と呼ばれる。

Temporal Difference(TD) methods — TD はベルマン方程式を元に更新する。

$$(\hat{q}^\pi)^{\text{new}}(s,a) = r(s,a) + \mathbb{E}_{s',a'}[q(s',a')|s,a]. \tag{1.1.22}$$

ベルマン作用素は収束するので、この新しい推定値 $(\hat{q}^\pi)^{\text{new}}$ はベルマン作用素の不動点 q^π に近づく。
TD 法はベルマン残差 (the Bellman residual)$B(s,a)$ を最小化しようとする:

$$B(s,a) = (\hat{q}^\pi)^{\text{new}}(s,a) - r(s,a) - \mathbb{E}_{s',a'}[q(s',a')|s,a]. \tag{1.1.23}$$

TD 法は $B(s,a)$ の推定 $\hat{B}(s,a)$ と学習率 α を用いて更新する。

$$(\hat{q}^\pi)^{\text{new}}(s,a) \leftarrow (\hat{q}^\pi)(s,a) + \alpha\hat{B}(s,a). \tag{1.1.24}$$

TD 残差 $B(s,a)$ は R_{t+1} と S_{t+1} によって推定されるが、$\hat{q}^\pi(s,a)$ は現在のサンプルにより更新されない、すなわち推定値のため固定される。しかし、そのサンプルはどの (s',a') を選択するか決定する。TD 法はどのように a' を選択するかで異なる。**SARSA** は方針からのサンプリング a' を選択する:

$$\mathbb{E}_{s',a'}[\hat{q}^\pi(s',a')] \approx \hat{q}^\pi(S_{t+1},A_{t+1}). \tag{1.1.25}$$

そして、**期待 SARSA(exptected SARSA)** は行動に対しての平均を取り、

$$\mathbb{E}_{s',a'}[\hat{q}^\pi(s',a')] \approx \sum_a \pi(a \mid S_{t+1})\, \hat{q}^\pi(S_{t+1},a). \tag{1.1.26}$$

Variance Reduction — 算術平均のような不偏最小分散推定器 (unbiased minimal variance estimator) により報酬推定を $r(s,a) = \hat{R}_t(s) = [\frac{1}{N}\sum_i r_i \mid s_i = s]$ とする。これは与えられた N 個の状態 s で観測された報酬サンプルの平均。これにより、分散が減る。

Q-learning は off-policy　TD アルゴリズムであり、収束されることは証明されている。Q-learning は次の式を用いる。

$$\mathbb{E}_{s',a'}[\hat{q}^\pi(s',a')] \approx \max_a \hat{q}(S_{t+1},a). \tag{1.1.27}$$

Q-learning によって学習された行動価値関数 q は従う方針と独立して q_* に近似する。より正確には、q は Q-learning により確率 1 で最適 q_* に近似する。しかしながら、方針は学習中に遭遇した状態-行動対を決定するので、その収束はすべての行動-状態対を訪問し、何度も更新することを必要とする。

1.2 Policy Gradient

Policy Gradient における目的は次の目的関数を最大化することにある。

$$J(\pi_\theta) \stackrel{\text{def}}{=} \sum_{s\in\mathcal{S}} d^\pi(s) v^\pi(s) \stackrel{(1.1.8)}{=} \sum_{s\in\mathcal{S}} d^\pi(s) \sum_{a\in\mathcal{A}} \pi_\theta(a \mid s)\, q^\pi(s,a). \tag{1.2.1}$$

ここで、$d^\pi(s) = \lim_{t\to\infty} \Pr(s_t = s \mid s_0, \pi_\theta)$ とする。これは、timestep t 間において方針 π_θ に従い、状態 s_0 から開始したときに $s_t = s$ になる確率である。PageRank アルゴリズムに関係ある。

Theorem 1.2.1. (Policy Gradient Theorem). equation 1.2.1の勾配は状態分布 $d^\pi(\cdot)$ の勾配を含まない。

$$\begin{aligned}\nabla_\theta J(\pi_\theta) &= \nabla_\theta \sum_{s\in\mathcal{S}} d^\pi(s) \sum_{a\in\mathcal{A}} \pi_\theta(a \mid s)\, q^\pi(s,a) \\ &\propto \sum_{s\in\mathcal{S}} d^\pi(s) \sum_{a\in\mathcal{A}} q^\pi(s,a)\, \nabla_\theta \pi_\theta(a \mid s).\end{aligned} \tag{1.2.2}$$

Proof. (Sutton and Barto, 2017) に従って証明していく。

$$\begin{aligned}\nabla_\theta v^\pi(s) &\stackrel{1.1.8}{=} \nabla_\theta \left(\pi_\theta(a \mid s)\, q^\pi(s,a)\right) \\ &= \sum_{a\in\mathcal{A}} \left(\nabla_\theta \pi_\theta(a \mid s)\, q^\pi(s,a) + \pi_\theta(a \mid s)\, \nabla_\theta q^\pi(s,a)\right) \\ &\stackrel{1.1.18}{=} \sum_{a\in\mathcal{A}} \left(\nabla_\theta \pi_\theta(a \mid s)\, q^\pi(s,a) + \pi_\theta(a \mid s)\, \nabla_\theta \left(\sum_{s',r} p(s', r \mid s, a)\left[r + v^\pi(s')\right]\right)\right) \\ &= \sum_{a\in\mathcal{A}} \left(\nabla_\theta \pi_\theta(a \mid s)\, q^\pi(s,a) + \pi_\theta(a \mid s)\, \nabla_\theta \left(\sum_{s'} p(s' \mid s, a) v^\pi(s')\right)\right) \\ &= \sum_{a\in\mathcal{A}} \left(\nabla_\theta \pi_\theta(a \mid s)\, q^\pi(s,a) + \pi_\theta(a \mid s) \sum_{s'} p(s' \mid s, a) \nabla_\theta v^\pi(s')\right).\end{aligned} \tag{1.2.3}$$

ここから equation 1.2.3が状態価値関数の再帰的な形になっていることがわかる。次に、方針 π_θ に従って状態 s から状態 x へステップ k で遷移する確率を $\rho^\pi(s \to x, k)$ とする。

$$s \xrightarrow{a\sim\pi_\theta(.|s)} s' \xrightarrow{a\sim\pi_\theta(.|s')} s'' \xrightarrow{a\sim\pi_\theta(.|s'')} \dots \tag{1.2.4}$$

- $k = 0$ の時、$\rho^\pi(s \to s, k = 0) = 1$。
- $k = 1$ の時、

$$\rho^\pi(s \to s', k = 1) = \sum_a \pi_\theta(a \mid s)\, P(s' \mid s, a). \tag{1.2.5}$$

- 状態 s から中間地点 s' に k ステップで移動した後、最後のステップで最終状態 x に移動すると考えると、訪問確率 (visitation probability) を再帰的に更新することができるので:

$$\rho^\pi(s \to x, k+1) = \sum_{s'} \rho^\pi(s \to s', k)\, \rho^\pi(s' \to x, 1). \tag{1.2.6}$$

次に、上記した考え方の元 equation 1.2.3の再帰的表現を展開する。ここで計算しやすいように $\kappa(s) = \sum_{a\in\mathcal{A}} \nabla_\theta \pi_\theta(a \mid s) q^\pi(s,a)$ とする。

$$
\begin{aligned}
\nabla_\theta v^\pi(s) &\overset{1.2.3}{=} \kappa(s) + \sum_a \pi_\theta(a|s) \sum_{s'} p(s'|s,a) \nabla_\theta v^\pi(s') \\
&= \kappa(s) + \sum_{s'} \sum_a \pi_\theta(a|s) p(s'|s,a) \nabla_\theta v^\pi(s') \\
&\overset{1.2.5}{=} \kappa(s) + \sum_{s'} \rho^\pi(s \to s', 1) \nabla_\theta v^\pi(s') \\
&= \kappa(s) + \sum_{s'} \rho^\pi(s \to s', 1) \left[\kappa(s') + \sum_{s''} \rho^\pi(s' \to s'', 1) \nabla_\theta v^\pi(s'') \right] \\
&\quad s' \text{を } s \to s'' \text{の中間点 (middle point) として考えると} \\
&= \kappa(s) + \sum_{s'} \rho^\pi(s \to s', 1) \kappa(s') + \sum_{s''} \rho^\pi(s \to s'', 2) \nabla_\theta v^\pi(s'') \\
&= \kappa(s) + \sum_{s'} \rho^\pi(s \to s', 1) \kappa(s') + \sum_{s''} \rho^\pi(s \to s'', 2) \kappa(s'') + \sum_{s'''} \rho^\pi(s \to s''', 3) \nabla_\theta v^\pi(s''') \\
&= \ldots \quad ;\nabla_\theta v^\pi(.) \text{ の部分的な展開をしていく} \\
&= \sum_{x \in \mathcal{S}} \sum_{k=0}^{\infty} \rho^\pi(s \to x, k) \kappa(x) \qquad (1.2.7)
\end{aligned}
$$

ここで行動価値関数の微分 $\nabla_\theta q^\pi(s,a)$ の式変形ができたので、目的関数 $J(\pi_\theta)$ も同様に適応させる。

$$
\begin{aligned}
\nabla_\theta J(\pi_\theta) &= \nabla_\theta v^\pi(s_0) \\
&= \sum_s \sum_{k=0}^{\infty} \rho^\pi(s_0 \to s, k) \kappa(s) \\
\text{ここで、} \eta(s) &= \sum_{k=0}^{\infty} \rho^\pi(s_0 \to s, k) \text{ とおくと、} \\
\nabla_\theta J(\pi_\theta) &= \sum_s \eta(s) \kappa(s) \\
&\overset{\text{normalize}}{=} \sum_s \eta(s) \sum_s \frac{\eta(s)}{\sum_s \eta(s)} \kappa(s) \\
&\propto \sum_s \frac{\eta(s)}{\sum_s \eta(s)} \kappa(s) \\
&= \sum_s d^\pi(s) \kappa(s) \qquad (1.2.8) \\
&= \sum_s d^\pi(s) \sum_{a \in \mathcal{A}} \nabla_\theta \pi_\theta(a \mid s) q^\pi(s,a) = (1.2.2).
\end{aligned}
$$

equation 1.2.8は、$\frac{\eta(s)}{\sum_s \eta(s)}$ が定常分布であるから $d^\pi(s)$ となる。(よくわかってない)

□

次に equation 1.2.2を再び期待値に式変形をする。

$$
\nabla_\theta \ln \pi_\theta(a \mid s) = \frac{\partial \ln \pi_\theta}{\partial \pi_\theta} \frac{\partial \pi_\theta}{\partial \theta} = \frac{1}{\pi_\theta} \nabla_\theta \pi_\theta(a \mid s) \qquad (1.2.9)
$$

$$
\begin{aligned}
\nabla_\theta J(\pi_\theta) &\overset{(1.2.2)}{=} \sum_s d^\pi(s) \sum_{a \in \mathcal{A}} \frac{1}{\pi_\theta(a \mid s)} \nabla_\theta \pi_\theta(a \mid s) [\pi_\theta(a \mid s)\, q^\pi(s,a)] \\
&\overset{(1.2.9)}{=} \sum_s d^\pi(s) \sum_{a \in \mathcal{A}} \nabla_\theta \ln \pi_\theta(a \mid s)\, \pi_\theta(a \mid s)\, q^\pi(s,a) \\
&= \mathbb{E}_{\pi_\theta} \left[\nabla_\theta \ln \pi_\theta(a \mid s)\, q^\pi(s,a) \right]. \qquad (1.2.10)
\end{aligned}
$$

1.3 Count-based Exploration with the Successor Representation

現在のアルゴリズムは問題特化したモデルやハンドクラフトの特徴を使っている。より一般化が求められる。この新しいアルゴリズムは traditional successor representation と違い substochastic successor representation を用いている。後者は前者同様に観測してきたそれぞれの状態や特徴の回数を暗黙的に数えることができる。この拡張には２つの独立した研究で繋がっている。traditional tabular domain として `RiverSim`, `SixArms` がある。時系列意思決定問題は即時と遅延報酬のトレードオフについて取り組む必要がある。RUDDER は delayed-reward に対するアプローチをしたのに対し、このアルゴリズムは探索について研究している。もっとも簡単な探索は vanilla DQN のような uniform action を行うようなものだが、sparse reward に対して探索は失敗する。model-based アプローチは力学の学習を行い将来行動の計画 (planning) を行う。すべての状態が列挙され一意に識別されれば、サンプルの複雑さは ϵ-最適方針に収束する前にエージェントの選択する準最適な行動の最大数を制限することを証明したアルゴリズムはある。[?] 関数近似を使う場合、状態訪問の考え方は役に立たず、モデル自身の学習を困難化させる。報酬ボーナスは探索を促進させるがこの分野では広く用いられない。Successor representation は TD を通して学習でき、環境の力学の遷移を暗黙的に推定することも可能である。substochastic successor representation は暗黙的に状態訪問を数えることができ、探索促進させる。

1.3.1 Preliminaries

一般的に強化学習での値はスカラーやベクトルで表現されるが、ここでは行列形式として表されるので準備をする。行列形式では $\boldsymbol{v}, \boldsymbol{r} \in \mathbb{R}^{\|\mathcal{S}\|}$、$P_\pi \in \mathbb{R}^{\|\mathcal{S}\| \times \|\mathcal{S}\|}$ とする。

$$
\begin{aligned}
v^\pi(s) &\overset{1.1.6}{=} \sum_a \pi(a \mid s) \left[\sum_{s',r} p(s', r \mid s, a) \left[r + \gamma v(s') \right] \right] \\
&= \sum_a \pi(a \mid s) [r + \gamma \sum_{s'} p(s' \mid s, a) v_\pi(s')] && (1.3.1) \\
\boldsymbol{v}_\pi &= \boldsymbol{r} + \gamma P_\pi \boldsymbol{v}_\pi && (1.3.2) \\
&= (\boldsymbol{I} - \gamma P_\pi)^{-1} \boldsymbol{r}. && (1.3.3)
\end{aligned}
$$

ここで P_π を π に従ったある状態からある状態へ遷移する確率関数とし、すなわち、$P_\pi(s, s') = \sum_a \pi(a \mid s)\, p(s' \mid s, a)$ である。

典型的な model-based アルゴリズムは行列 P_π とベクトル $\boldsymbol{r}$ とそれらをつかった v_π の推定を学習する。equation 1.3.3を解くために $P_\pi, \boldsymbol{r}$ の経験的推測 $\hat{P}_\pi, \hat{\boldsymbol{r}}$ を使う。

$$
\hat{P}_\pi(s' \mid s) = \frac{n(s, s')}{n(s)}, \qquad \hat{\boldsymbol{r}}(s) = \frac{C(s, s')}{n(s)}. \tag{1.3.4}
$$

ここで $\hat{\boldsymbol{r}}(s)$ はベクトルの i-th を表し、$n(s, s')$ は観測された遷移 $s \to s'$ の数を表し、$n(s) = \sum_{s \in \mathcal{S}} n(s, s')$、そして $C(s, s')$ は $n(s, s')$ に関連のある報酬の総和である。これを用いたアプローチは巨大な状態空間をもった問題に対し解けない。なぜなら、全遷移の保持をすることは困難であり、学習されたモデルは正確にいくつものタイムステップで推測できないので複数の状態に渡って一般化するアプローチは失敗する。

この論文に紹介するアルゴリズムは successor representation をベースにしており、successor state の類似性による状態間の一般化を定義した表現学習 (representation learning) として提案された手法である。方針 π についての SR は

$$
\Psi_\pi(s, s') = \mathbb{E}_{\pi,p}[\sum_{t=0}^{\infty} \gamma^t \, \mathbb{I}\{S_t = s'\} \mid S_0 = s], \tag{1.3.5}
$$

と定義され、ここで総和は $\mathbb{I}$ により収束すると仮定する。Dayan(1993) は期待値が TD 学習を通してサンプルからの推定によって示した。これもまた γP_π のノイマン級数 (Neumann series) に対応し、

$$\Psi_\pi = \sum_{t=0}^{\infty} \gamma^t (P_\pi)^t = (I - \gamma P_\pi)^{-1}. \tag{1.3.6}$$

SR は価値関数 ($\boldsymbol{v}_\pi = \Psi_\pi \boldsymbol{r}$) を計算する一部分であることに注意する。

SR の定義は特徴に拡張することができる。Successor features は SR を関数近似問題に一般化する。SF も TD 学習で学習可能である。

Definition 1.3.1. (Successor Features). $\boldsymbol{\psi}_\pi(s) \in \mathbb{R}^d$ を方針 π に従うとき、状態 s の SF を示すとする。特徴表現 (feature representation)$\boldsymbol{\phi} \in \mathbb{R}^d$ が与えられたとき、

$$\boldsymbol{\psi}_\pi(s) = \mathbb{E}_{\pi,p}\left[\sum_{t=0}^{\infty} \gamma^t \boldsymbol{\phi}(S_t) \mid S_0 = s\right]. \tag{1.3.7}$$

行列表現では、$\Psi_\pi = \sum_{t=0}^{\infty} \gamma^t (P_\pi)^t \Phi = (I - \gamma P_\pi)^{-1} \Phi$ である。

1.3.2 The Substochastic Successor Representation

SSR は SR にそれぞれの状態からの"phantom"遷移を追加したようなものである。

Definition 1.3.2. (Substochastic Successor Representation). $\tilde{P}_\pi$ は環境の力学や方針によって誘発された substochastic matrix であり、$\tilde{P}_\pi(s'|s) = \frac{n(s,s')}{n(s)+1}$. substochastic successor representation を $\tilde{\Psi}_\pi$ とすると、

$$\tilde{\Psi}_\pi = \sum_{t=0}^{\infty} \gamma^t \tilde{P}_\pi^t = (I - \gamma \tilde{P}_\pi)^{-1}. \tag{1.3.8}$$

Theorem 1.3.1. $\hat{P}_\pi$ を successor representation $\hat{\Psi}_\pi$ と関連のある経験的遷移確率行列 (*emprical transition probability matrix*) とする。$\boldsymbol{e}$ は全一ベクトルを表し、$N \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ は対角行列、すなわち $N(s,s) = \frac{1}{n(s)+1}$ とする。$0 \leq \gamma < 1$ が与えられたとき、

$$\hat{\Psi}_\pi \boldsymbol{e} \approx \frac{\boldsymbol{e} - \gamma \hat{\Psi}_\pi N \boldsymbol{e}}{1-\gamma}, \tag{1.3.9}$$

と定義する。

この定理により、SSR は $\hat{\Psi}_\pi N \boldsymbol{e}$ によって近似される。そのベクトルの s-entry は

$$(\hat{\Psi}_\pi N \boldsymbol{e})(s) = \sum_{s'} \sum_{t \geq 0} \gamma^t \hat{P}_\pi^t(s' \mid s) \frac{1}{n(s')+1}, \tag{1.3.10}$$

である。ここで、$\hat{P}_\pi^t(s' \mid s)$ は $s \to s'$ の t-step 経験的遷移確率である。SSR は SR のわずかな変化がわかった後、将来の"逆訪問回数 (future inverse visit counts)"の合計を回復するために用いることができる。(よくわからん)
ここから、agent の報酬を $r(s,a) - \beta \hat{\Psi}_\pi \boldsymbol{e}$ となる。

1.4 アルゴリズムの実装

1.4.1 DQN と DDPG の違い (?)

DQN は毎 step ごとに actor によって `sgd_update_frequency` 個分の transition を生成している。それに対して、DDPG は毎 step ごとに 1 transition しか生成していない。この違いは何だろうか。DQN の

`self.total_steps / sgd_update_frequency % network_update_freq == 0` としているところから、DDPG に近い更新に見える。

両方共、experience replay を使う off-policy なのは理解できる。あまり、深入りするような話でもない？