Successor Uncertainties: Exploration and Uncertainty in Temporal Difference Learning

David Janz*, Jiri Hron*, Przemysław Mazur, Katja Hofmann, José Miguel Hernández-Lobato, Sebastian Tschiatschek

University of Cambridge · Microsoft Research · Wayve · The Alan Turing Institute

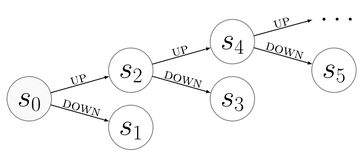
tl;dr: *Randomised Value Functions* with *Successor Features* yield SOTA on exploration benchmarks and double DQN's Atari scores

Climbing up a tree

Set-up: tree MDP with states $\mathcal{S} = \{s_1, s_2, \dots, s_{2L}\}$, actions $\mathcal{A} = \{\text{UP}, \text{DOWN}\}$ and transition kernel $\mathcal{T}$ mapping $\mathcal{S} \times \mathcal{A}$ to rewards and next states. DOWN ends game, only reward when s_{2L} reached.

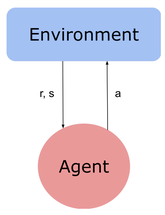
Agent tries to find a policy $\pi\colon \mathcal{S} \to \mathcal{A}$ maximising return $\mathbb{E}_\pi[\Sigma_{t\geq 1}\gamma^{t-1}r_t]$. $\mathcal{T}$ unknown to the agent, must explore the environment to learn.

Challenge: $\mathcal{O}(2^L)$ policies with same outcomes in most states, only one with positive return.



A reinforcement learning approach

Q function: Q function gives the expected return after taking $a \in \mathcal{A}$ in $s \in \mathcal{S}$, and then following π



$$Q^\pi(s,a) = \mathbb{E}_\pi[\Sigma_{t\geq 1}\gamma^{t-1}r_t \mid s_0 = s, a_0 = a]$$
$$= \mathbb{E}_\pi[r_1 + \gamma Q^\pi(s_1, \pi(s_1)) | s_0 = s, a_0 = a]$$

Q^π can be used to obtain a better policy

$$\hat{\pi}\colon s \mapsto \text{argmax}_{a\in\mathcal{A}} Q^\pi(s,a)$$

Temporal Difference Learning: Evaluate Q^π for the new policy by repeated application of the Bellman operator

$$(B^\pi Q)(s,a) = \mathbb{E}_{r,s'\sim\mathcal{T}(s,a)}[r + \gamma Q(s', \pi(s'))]$$

Exploration via Posterior Sampling

PSRL (Strens, 2000; Osband et al., 2013): Model the uncertainty about $\mathcal{T}$ by $P_\mathcal{T}$. To solve the MDP, iteratively (i) act optimally w.r.t. a sampled $\mathcal{T} \sim P_\mathcal{T}$, (ii) use the collected data update $P_\mathcal{T}$.

Unfortunately, PSRL is often impractical:

- Repeatedly solving the *sampled* MDP $\mathcal{T} \sim P_\mathcal{T}$ can be very expensive.
- PSRL cannot straightforwardly generalise from seen to unseen states.

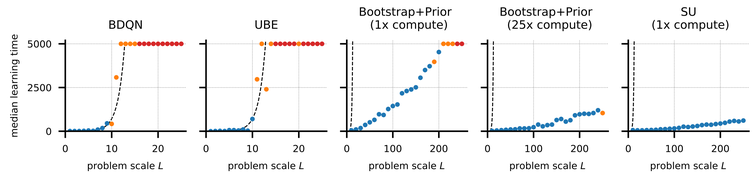
RVF (Osband et al., 2016): Model and act greedily w.r.t. $Q \sim P_Q$ to avoid solving the *sampled* $\mathcal{T} \sim P_\mathcal{T}$. Use a parametric model (e.g., $Q(s,a) = \langle\phi(s,a), w\rangle + \varepsilon$) to generalise to unseen states.

Pathologies of Randomised Value Functions

Prop: *If $P_Q = \prod_{s,a} P_{Q(s,a)}$, all $P_{Q(s,a)}$ symmetric around the same point, then greediness w.r.t. $Q \sim P_Q$ is equiv. to acting uniformly at random.*

Even P_Q which **propagate uncertainty** (O'Donoghue et al., 2018), i.e., satisfy $\text{Var}(Q(s,a)) = \text{Var}(r + \gamma Q(s', \pi(s')) \,|\, s, a)$, suffer from the pathology. Propagation of uncertainty is also not necessary.

Prop: *Under posterior sampling, any distribution over exploration policies can be induced by a P_Q which does not propagate uncertainty.*



Median # episodes to solve the tree MDP (5 seeds). Blue = all (5), orange = some (1-4), red = none of the 5 runs finished within 5000 episodes. Dashed line for uniform policy. Note the varying x-axis scale!

Successor Uncertainties

Idea: Model $\mathcal{T} \sim P_\mathcal{T}$ as in PSRL but remain computationally tractable!

Let $\phi\colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ be a state-action embedding and assume $\exists\, w \in \mathbb{R}^d$ such that $\mathbb{E}[r_{t+1} | s_t = s, a_t = a] = \langle\phi(s,a), w\rangle$ for all s, a. Then

$$Q^\pi(s,a) = \langle \Sigma_{t\geq 1}\gamma^{t-1}\mathbb{E}[\phi(s_t,a_t)|s_0 = s, a_0 = a], w\rangle = \langle\psi(s,a), w\rangle\,,$$

where the ψ is known as **successor features** (Dayan, 1993).

SU model of $\mathcal{T}$: Bayesian inference for $w \sim \mathcal{N}(0, \theta I_d)$, $r \mid w, s, a \sim \mathcal{N}(\langle\phi(s,a), w\rangle, \beta)$, empirical frequencies for the transition probabilities.

$\Rightarrow$ With posterior $w \sim \mathcal{N}(\mu_w, \Sigma_w)$, a **non-diagonal P_Q model:**

$$Q \sim \mathcal{N}(\langle\Psi, \mu_w\rangle, \Psi\Sigma_w\Psi^\top)\,, \qquad \Psi = [\psi(s,a)]_{s,a\in\mathcal{S}\times\mathcal{A}}$$

Adding SU to your favourite NN architecture is simple!
Just add a head (one for ϕ, one for ψ); with $\phi_t := (s_t, a_t)$, $\psi_t := \psi(s_t, a_t)$

$$\underbrace{|\langle\phi_t, w\rangle - r_{t+1} - \langle\psi_{t+1}, w\rangle^-|^2}_{=:\ell_t^Q} + \underbrace{\|\psi_t - \phi_t - \gamma\psi_{t+1}^-\|^2}_{=:\ell_t^r} + \underbrace{|\langle\phi_t, w\rangle - r_{t+1}|^2}_{=:\ell_t^{\text{SF}}},$$

jointly learn ϕ and ψ by minimising $\ell \propto \Sigma_t[\ell_t^Q + \ell_t^r + \ell_t^{\text{SF}}]$.

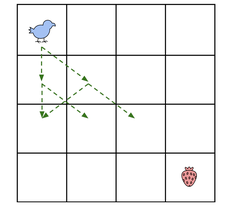
Notice that $\Sigma_t \ell_t^Q$ is the **usual Q value loss** and the equal weighting of all 3 losses (no additional hyperparameter)!

Deep Sea MDP

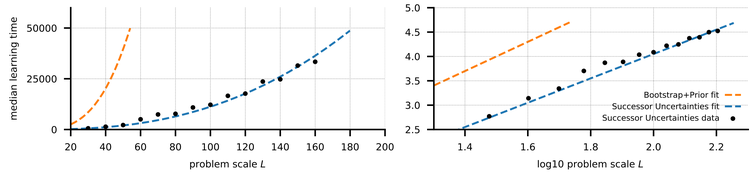
A very hard exploration task introduced together with the Bootstrap+Prior (Osband et al., 2018).

Set-up: Two actions: down left for zero and down right for $-0.01/L$ reward, except for executing L down right for $L(-0.01)/L + 1 = 0.99$ total return.

Results: BDQN, UBE exceed comp. limit. SU scales better than Bootstrap+Prior when $L \gg 0$.



Inspired by Figure 1 in (Osband et al., 2019).

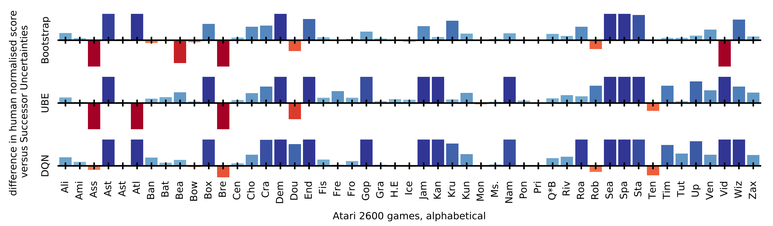


Median # episodes to solve the chain MDP (5 seeds). SU curve $\log_{10}(T) = 2.5\log_{10}(L) - 0.95$. Boostrap+Prior curve taken from Figure 8 in (Osband et al., 2018).

Atari 2600

Set-up: 49 Atari games as in (Mnih et al., 2015), NN architecture as in (van Hasselt et al., 2016)

Algorithm	Median	Superhuman %
SU (ours)	**2.09**	**77.55%**
BootDQN	1.60	67.35%
UBE	1.07	51.02%
DQN	1.00	48.98%



Bars show the difference in human normalised score between SU and BootDQN (top), UBE (middle) and DQN (bottom) for each of the 49 Atari 2600 games. Blue indicates SU performed better, red worse. SU outperforms the baselines on $36/49$, $43/49$ and $42/49$ games respectively. Y-axis clipped to $[-2.5, 2.5]$.

Citations:

1. Dayan, P. Improving generalization for temporal difference learning: the successor representation. Neural Computation, 1993.
2. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. Nature, 2015.
3. O'Donoghue, B., Osband, I., Munos, R., and Mnih, V. The uncertainty Bellman equation and exploration. ICML, 2018.
4. Osband, I., Russo, D., and Van Roy, B. (More) efficient reinforcement learning via posterior sampling. NeurIPS, 2013.
5. Osband, I., Van Roy, B., and Wen, Z. Generalization and exploration via randomized value functions. ICML, 2016.
6. Osband, I., Aslanides, J., and Cassirer, A. Randomized prior functions for deep reinforcement learning. NeurIPS, 2018.
7. Strens, M. A Bayesian framework for reinforcement learning. ICML, 2000.
8. van Hasselt, H., Guez, A., and Silver, D. Deep reinforcement learning with double Q-learning. AAAI, 2016.