Upside-Down Reinforcement Learning Can Diverge in Stochastic Environments With Episodic Resets

Miroslav Štrupl[1], Francesco Faccio[1], Dylan R. Ashley[1], Rupesh Kumar Srivastava[2], Jürgen Schmidhuber[1,2,3]

[1]The Swiss AI Lab IDSIA/USI/SUPSI, Lugano, Switzerland [2]NNAISENSE, Lugano, Switzerland [3]KAUST, Thuwal, Saudi Arabia

{struplm, francesco, dylan.ashley}@idsia.ch, rupesh@nnaisense.com, juergen@idsia.ch

Summary

- Upside-Down Reinforcement Learning (UDRL) is an approach for solving RL problems that does not require value functions and uses *only* supervised learning[2, 3].
- Ghosh et al. [1] proved that Goal-Conditional Supervised Learning (GCSL)---a simplified version of UDRL---optimizes a lower bound on goal-reaching performance.
- **Question: Does UDRL converge to the optimal policy in arbitrary environments?**
  Here we show that for a specific *episodic* UDRL algorithm (eUDRL, including GCSL), **this is not the case, and give the causes of this limitation.**
- **Assumptions:** finite (discrete) environments, no function approximation, unlimited number of samples.

Background

$\mathcal{M} = (\mathcal{S}, \mathcal{A}, p_T, \mu_0, r)$ an MDP where:

- $\mathcal{S}, \mathcal{A}$ - finite state and action spaces
- $p_T(s'|s,a)$ is a transition probability.
- $r(s', s, a)$ deterministic reward function.
- $\mu_0(s)$ initial state probability.
- return $G_t := \sum_{k=0}^{\infty} r(S_{t+1+k}, S_{t+k}, A_{t+1})$,
- policy $\pi(a|s)$
- state/action-value functions:
  $V^\pi(s) := \mathbb{E}_\pi[G_t | S_t = s; \pi]$,
  $Q^\pi(s,a) := \mathbb{E}_\pi[G_t | S_t = s, A_t = a; \pi]$.

In UDRL the agent takes (besides the state) an extra *command* input (h, g). We will fix the command interpretation: **``reach goal g in h number of steps".**
Objective: Become better at fulfilling commands

$$\pi(\overset{\text{action}}{a} \mid \underbrace{\overset{\mathcal{M}\text{-state}}{s}, \overset{\text{horizon}}{h}, \overset{\text{goal}}{g}}_{\bar{s}\ -\ \bar{\mathcal{M}}\ \text{state}})$$

Motivation: We extend the state space by the command to be able to view an eUDRL agent as an ordinary agent on a slightly bigger MDP $\bar{\mathcal{M}}$.

Command extension (CE) of an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p_T, r, \mu_0)$ is the MDP $\bar{\mathcal{M}} = (\bar{\mathcal{S}}, \mathcal{A}, \bar{p}_T, \bar{r}, \bar{\mu}_0, \rho)$, where:

(Items in this color has to be supplied in addition to $\mathcal{M}$)

- $\rho : \mathcal{S} \to \mathcal{G}$-goal map, $\mathcal{G}$-goal set
- $\bar{\mathcal{S}} := \mathcal{S} \times \{h \leq N\} \times \mathcal{G}$, N-max.hor., $\bar{\mathcal{S}}_A := \{(s,h,g) \in \bar{\mathcal{S}} | h = 0\}$-absorbing states
- $\bar{\mu}_0(s,h,g) := \mathbb{P}(H_0 = h, G_0 = g | S_0 = s)\mu_0(s)$

$$\begin{pmatrix} s_0 \\ h \\ g \end{pmatrix} \xrightarrow[r\ :\ 0]{\pi, p_T} \begin{pmatrix} s_1 \\ h-1 \\ g \end{pmatrix} \xrightarrow[0]{\pi, p_T} \cdots \begin{pmatrix} s_{h-1} \\ 1 \\ g \end{pmatrix} \xrightarrow[\substack{1\{\rho(s_h) = g\} \\ (1 \text{ iff hit})}]{\pi, p_T} \begin{pmatrix} s_h \\ 0 \\ g \end{pmatrix} \in \bar{\mathcal{S}}_A$$

$\bar{p}_T$: g-fixed, h-decreases by 1 til 0, s-evolves according to p_T for $h > 0$;
$\bar{r}$: non-zero just from $h = 1 \implies V^\pi(s,h,g) = \mathbb{P}(\rho(S_h) = g | \bar{S}_0 = (s,h,g); \pi)$.

Segment distribution $\Sigma \sim d_\Sigma^\pi$ - analogy to the state visitation distribution, **segment** - a continuous chunk of the trajectory

$$\Sigma = (\underset{\text{length}}{l(\Sigma)}, \underset{\text{the first state}}{S_0^\Sigma}, H_0^\Sigma, G_0^\Sigma, A_0^\Sigma, S_1^\Sigma, A_1^\Sigma, \ldots, \underset{\text{the last state}}{S_{l(\Sigma)}^\Sigma})$$

eUDRL learning algorithm

eUDRL [3] starts from an initial policy π_0 and generates a sequence of policies (π_n). Each iteration consists of two steps:

1. a batch of episodes is generated using the current policy π_n,
2. a new policy π_{n+1} is fitted to some action conditional of $d_\Sigma^{\pi_n}$

$$\tau = (s_0, a_0, \ldots, \overset{\downarrow t}{s_t}, a_t, \ldots, \overset{\downarrow t'}{s_{t+l(\sigma)}}, \ldots, s_N)$$
$$\sigma = (s_0^\sigma, a_0^\sigma, \ldots, s_{l(\sigma)}^\sigma)$$

σ evidences that a_0^σ might be good for reaching $\rho(s_{l(\sigma)}^\sigma)$ in $l(\sigma) (= t' - t)$ steps

$$\pi_{n+1} := \arg\max_\pi \mathbb{E}_\sigma \log\Big(\pi(a_0^\sigma \mid s_0^\sigma, \underbrace{l(\sigma)}_{h}, \underbrace{\rho(s_{l(\sigma)}^\sigma)}_{g})\Big).$$

* **lemma 4.1:**(eUDRL insensitivity to goal input at horizon 1) Let us have an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p_T, r, \mu_0)$ and its CE $\bar{\mathcal{M}} = (\bar{\mathcal{S}}, \mathcal{A}, \bar{p}_T, \bar{r}, \bar{\mu}_0, \rho)$, such that there exists a state $s \in \mathcal{S}$ and two goals $g_0 \neq g_1$, $g_0, g_1 \in \mathcal{G}$ such that $M_0 := \arg\max_{a \in \mathcal{A}} Q^*((s,1,g_0), a)$ and $M_1 := \arg\max_{a \in \mathcal{A}} Q^*((s,1,g_1), a)$ (optimal policy supports for g_0, g_1) have empty intersection $M_0 \cap M_1 = \emptyset$. Assume $Q_A^{\pi_n, g_i}(s,1,a) \geq q_i(1-\delta)$ where delta $\delta > 0$ and $q_i := \max_a Q_A^{\pi_n, g_i}(s,1,a)$. Then, when $\delta < 1$ (stochastic environment), the sequence (π_n) of policies produced by eUDRL recursion cannot tend to the optimal policy set.

eUDRL Non-Optimality in Stochastic Environments

eUDRL Recursion Rewrite

$$\pi_{n+1} := \arg\max_\pi \mathbb{E}_\sigma \log\Big(\pi(a_0^\sigma \mid s_0^\sigma, \underbrace{l(\sigma)}_{h}, \underbrace{\rho(s_{l(\sigma)}^\sigma)}_{g})\Big).$$

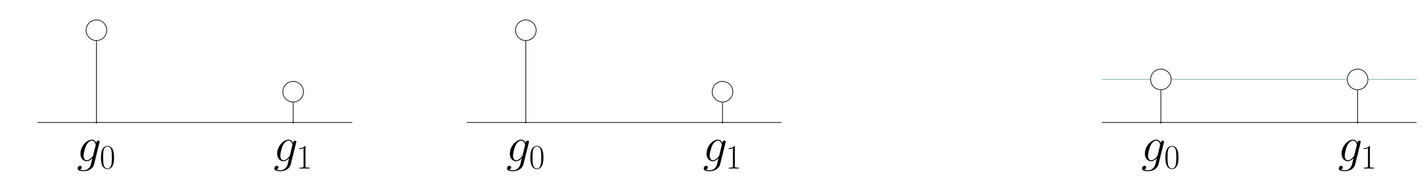
$$\pi_{n+1}(a|s,h,g) = \mathbb{P}(A_0^\Sigma = a | S_0^\Sigma = s, l(\Sigma) = h, \rho(S_{l(\Sigma)}^\Sigma) = g; \pi_n) \tag{3.1}$$
$$\propto \underbrace{\mathbb{P}(\rho(S_{l(\Sigma)}^\Sigma) = g | A_0^\Sigma = a, S_0^\Sigma = s, l(\Sigma) = h; \pi_n)}_{\substack{Q_A^{\pi_n, g}(s,h,a) \\ \text{average Q}}} \cdot \underbrace{\mathbb{P}(A_0^\Sigma = a | S_0^\Sigma = s, l(\Sigma) = h; \pi_n)}_{\substack{\pi_{A,n}(a|s,h) \\ \text{average policy}}}$$

where

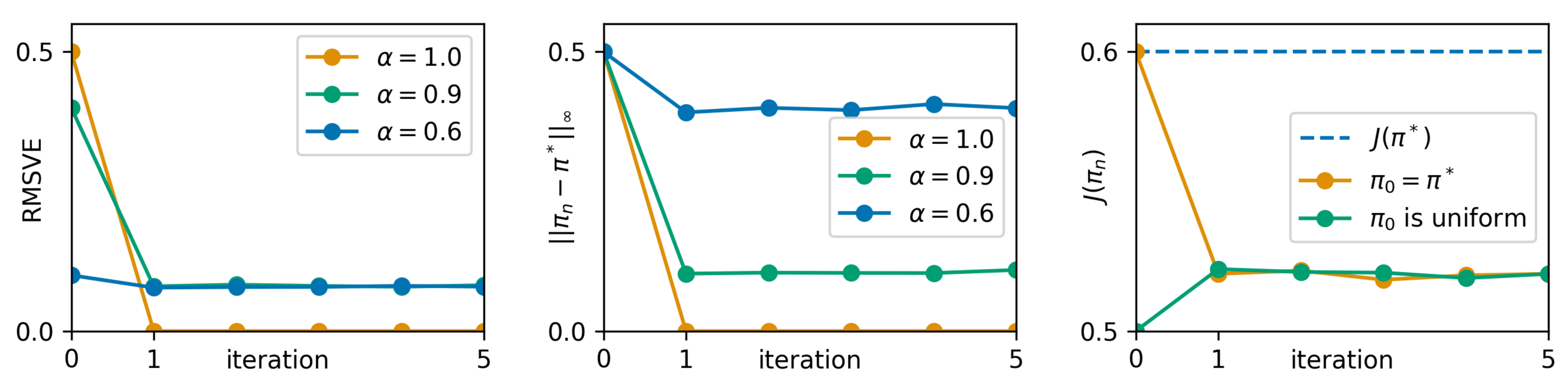
$$Q_A^{\pi_n, g}(s,h,a) = \mathbb{P}(\rho(S_h) = g | A_0 = a, S_0 = s; \pi_n)$$
$$\pi_{A,n}(a|s,h) = \sum_{h' \geq h, g' \in \mathcal{G}} \pi_n(a|h', g', s)\mathbb{P}(H_0^\Sigma = h', G_0^\Sigma = g' | S_0^\Sigma = s, l(\Sigma) = h; \pi_n)$$

Problem: "averaging" across goals g', and horizons h' is a problem.
E.g. $\pi_{A,n}$ is constant in g, everything has to be accounted in multiply by $Q_A^{\pi_n, g}$ step. (Formally see lemma 4.1 at the bottom*)

Ex: $(a_0 \in M_0)$: $\pi_{n+1}(a_0|s,1,g) \quad \propto \mathbb{P}(\rho(s_1) = g | A_0 = a_0, S_0 = s; \pi_n) \quad \times \pi_{A,n}(a_0 \mid s, 1)$

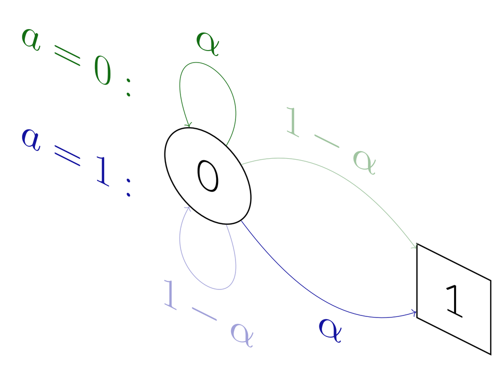


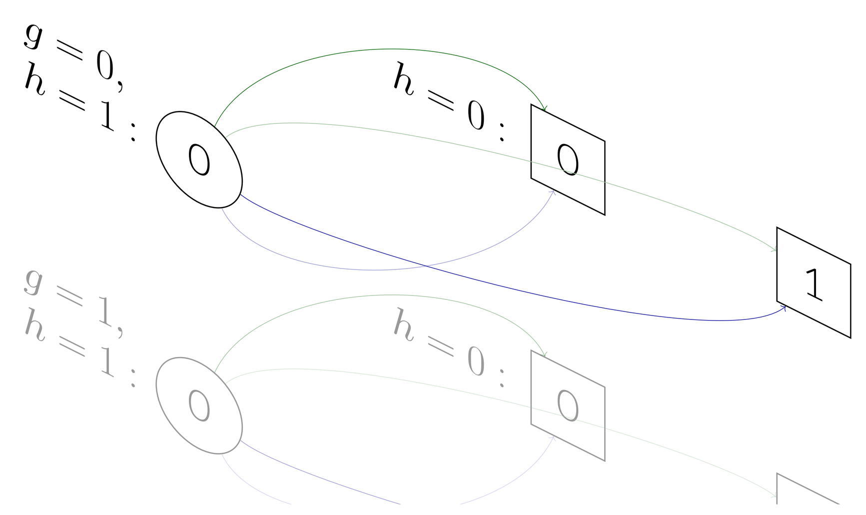
Demonstration



$\mathcal{M} : \mathcal{S} := \mathcal{A} := \{0, 1\}$,
$\mu_0 : S_0 := 0$

$\bar{\mathcal{M}} : N := 1, \mathcal{G} := \mathcal{S}, \rho := \mathrm{id}_\mathcal{S}$,
$\bar{\mu}_0 : H_0 := 1,\ G_0 | H_0, S_0 \sim \mathcal{U}(\mathcal{G})$





$\alpha \in [0.5, 1]$ -stochasticity,
$\alpha = 1$-deterministic p_T,

• Everything is constant for iteration > 0. • RMSVE and $\|\pi_n - \pi^*\|_\infty$ **do not approach 0** for stochastic case ($\alpha < 1$). Increasing the number of iterations or the sample size **does not help!** • There is **no monotony** in GCSL goal reaching objective $J(\pi_n) = \sum_{\bar{s} \in \bar{\mathcal{S}}} V^{\pi_n}(\bar{s})\bar{\mu}_0(\bar{s})$.

Conclusion

- Definitions **command extension** and **segment distribution** allowed for formal investigation of eUDRL/GCSL.
- The eUDRL **recursion rewrite** (3.1) helps to understand causes of eUDRL/GCSL non-optimality.
- We disproved eUDRL's convergence to the optimum for quite a large class of stochastic environments in Lemma 4.1.
- The example demonstrates that there is **no guarantee for monotonic improvement**.
- This result applies to certain existent implementations [3, 1] that nevertheless produce useful results in practice.

Acknowledgements & References

This work was supported by the ERC Advanced Grant (no: 742870) and by the Swiss National Supercomputing Centre (CSCS, project: s1090). We also thank NVIDIA Corporation for donating a DGX-1 as part of the Pioneers of AI Research Award and to IBM for donating a Minsky machine.

[1] D. Ghosh, A. Gupta, A. Reddy, J. Fu, C. Devin, B. Eysenbach, and S. Levine. Learning to reach goals via iterated supervised learning, 2019.

[2] J. Schmidhuber. Reinforcement learning upside down: Don't predict rewards -- just map them to actions, 2019.

[3] R. K. Srivastava, P. Shyam, F. Mutz, W. Jaśkowski, and J. Schmidhuber. Training agents using upside-down reinforcement learning, 2019.