World Value Functions: Knowledge Representation for Learning and Planning

Geraud Nangue Tasse, Benjamin Rosman, Steven James

School of Computer Science and Applied Mathematics
University of the Witwatersrand
Johannesburg, South Africa
geraud.nanguetasse1@students.wits.ac.za, {benjamin.rosman1, steven.james}@wits.ac.za

Abstract

We propose world value functions (WVFs), a type of goal-oriented general value function that represents how to solve not just a given task, but any other goal-reaching task in an agent's environment. This is achieved by equipping an agent with an internal goal space defined as all the world states where it experiences a terminal transition. The agent can then modify the standard task rewards to define its own reward function, which provably drives it to learn how to achieve all reachable internal goals, and the value of doing so in the current task. We demonstrate two key benefits of WVFs in the context of learning and planning. In particular, given a learned WVF, an agent can compute the optimal policy in a new task by simply estimating the task's reward function. Furthermore, we show that WVFs also implicitly encode the transition dynamics of the environment, and so can be used to perform planning. Experimental results show that WVFs can be learned faster than regular value functions, while their ability to infer the environment's dynamics can be used to integrate learning and planning methods to further improve sample efficiency.

Introduction

A grand challenge of artificial intelligence is to create general agents capable of solving a wide variety of tasks in the real world. To accomplish this, we require a general decision-making framework that models agents' interaction with the world, and a sufficiently general representation to capture the knowledge agents acquire. Reinforcement learning (RL) (Sutton, Barto et al. 1998) is one such framework and although it has made several major breakthroughs in recent years, ranging from robotics (Levine et al. 2016) to board games (Silver et al. 2017), these agents are typically narrowly designed to solve only a single task.

In RL, tasks are specified through a reward function from which the agent receives feedback. Most commonly, an agent represents its knowledge in the form of a value function, representing the sum of future rewards it expects to receive. However, since the value function is directly tied to one single reward function (and hence task), it is definitionally insufficient for constructing agents capable of solving a wide range of tasks.

In this work, we seek to overcome this limitation by proposing *world value functions* (WVFs), a goal-oriented knowledge representation that encodes how to solve not only the current task, but also any other goal-reaching task. In the literature, agents with such abilities are said to possess *mastery* (Veeriah, Oh, and Singh 2018), and we prove that WVFs do, in fact, possess this property in deterministic environments. Importantly, WVFs are a form of general value function (Sutton et al. 2011) that can be learned from a single stream of experience; no additional information or modifications to the standard RL framework are required.

WVFs have several desirable properties, which we formally prove in the deterministic setting. In particular, we show that (i) given a learned WVF, any new task can be solved by estimating its reward function, which reduces the problem to supervised learning; and (ii) WVFs implicitly encode the dynamics of the world and can be used for model-based RL. Experimental results in the Four Rooms domain (Sutton, Precup, and Singh 1999) validate our theoretical findings, while demonstrating that not only can WVFs be learned faster than regular value functions, they can also be leveraged to perform Dyna-style planning (Sutton 1990) to improve sample efficiency.

Preliminaries

We model an agent's environment as a Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, P, R)$, where (i) $\mathcal{S}$ is the state space, (ii) $\mathcal{A}$ is the action space, (iii) $P(s, a, s')$ are the transition dynamics of the world, and (iv) R is a reward function, bounded by $[R_{\text{MIN}}, R_{\text{MAX}}]$, representing the task the agent must solve. Note that in this work, we focus on environments with *deterministic* dynamics, but put no restrictions on their complexity.

The agent's aim is to compute a *policy* π from $\mathcal{S}$ to $\mathcal{A}$ that optimally solves a given task. This is often achieved by learning a value function that represents the expected return obtained under π starting from state s: $V^\pi(s) = \mathbb{E}^\pi\left[\sum_{t=0}^{\infty} r(s_t, a_t, s_{t+1})\right]$. Similarly, the action-value function $Q^\pi(s, a)$ represents the expected return obtained by executing a from s, and thereafter following π. The optimal action-value function is given by $Q^*(s, a) = \max_\pi Q^\pi(s, a)$ for all states s and actions a, and the optimal policy follows by acting greedily with respect to Q^* at each state.

World Value Functions

We now introduce *world value functions* (WVFs), which provably encode how to reach all achievable goals. We first define the internal goal space $\mathcal{G} \subseteq \mathcal{S}$ of the agent as all states where it experiences a terminal transition.

Different from other goal-oriented approaches where goals are specified by the environment, here the goal an agent wishes to achieve is chosen by itself. The agent's aim now is to simultaneously solve the current task, while also learning how to achieve its own internal goals. To do so, the agent can define its own goal-oriented reward function $\bar{R}$, which extends R to penalise itself for achieving goals it did not intend to:

$$\bar{R}(s,g,a,s') := \begin{cases} \bar{R}_{\text{MIN}} & \text{if } g \neq s \text{ and } s' \text{ is absorbing} \\ R(s,a,s') & \text{otherwise,} \end{cases}$$

where $\bar{R}_{\text{MIN}}$ is a large negative penalty that can be derived from the bounds of the reward function (Nangue Tasse, James, and Rosman 2020). Intuitively, the penalty $\bar{R}_{\text{MIN}}$ adds one bit of information to the agent's rewards, and we will later prove this is sufficient for the agent to learn how to achieve its internal goals in the current task.

The agent must now compute a *world policy* $\bar{\pi} : \mathcal{S} \times \mathcal{G} \to \Pr(\mathcal{A})$ that optimally reaches its internal goal states. Given a world policy $\bar{\pi}$, the corresponding WVF is defined as $\bar{Q}^{\bar{\pi}}(s,g,a) := \mathbb{E}^{\bar{\pi}}_{s'}\left[\bar{R}(s,g,a,s') + \bar{V}^{\bar{\pi}}(s',g)\right]$, where $\bar{V}^{\bar{\pi}}(s,g) := \mathbb{E}^{\bar{\pi}}\left[\sum_{t=0}^{\infty} \bar{R}(s_t,g,a_t,s_{t+1})\right]$.

Since the WVF satisfies the Bellman equations, $\bar{Q}^*(s,g,a)$ can be learned using any suitable RL algorithm, such as Q-learning (see Algorithm 1).

Properties of World Value Functions

While a learned WVF encodes the values of achieving all internal goals, it can still be used to solve the task in which it was learned. Theorem 1 below demonstrates that the current task's reward and value function can be recovered by simply maximising over goals:

Theorem 1. *Let $M = (\mathcal{S}, \mathcal{A}, P, R)$ be a deterministic task with optimal action-value function Q^* and optimal world action-value function $\bar{Q}^*$. Then for all (s,a,s') in $\mathcal{S} \times \mathcal{A} \times \mathcal{S}$, we have (i) $R(s,a,s') = \max_{g \in \mathcal{G}} \bar{R}(s,g,a,s')$, and (ii) $Q^*(s,a) = \max_{g \in \mathcal{G}} \bar{Q}^*(s,g,a)$.* □

As a result, the optimal policy for the current task can be obtained by computing $\pi^*(s) \in \arg\max_{a \in \mathcal{A}} \left(\max_{g \in \mathcal{G}} \bar{Q}^*(s,g,a)\right)$.

Having established WVFs as a type of task-specific general value function (GVF) (Sutton et al. 2011), we next prove in Theorem 2 that they do indeed have mastery—that is, they learn how to reach all achievable goal states in the world. We first formally define mastery as follows:

Definition 1. *Let $\bar{Q}^*$ be the optimal world action-value function for a task M. Then $\bar{Q}^*$ has mastery if for all $g \in \mathcal{G}$ reachable from $s \in \mathcal{S} \setminus \{g\}$, there exists an optimal world policy $\bar{\pi}^*(s,g) \in \arg\max_{a \in \mathcal{A}} \bar{Q}^*(s,g,a)$ such that $\bar{\pi}^* \in \arg\max_{\bar{\pi}} P^{\bar{\pi}}_s(s_T = g)$, where $P^{\bar{\pi}}_s(s_T = g)$ is the probability of reaching g from s under a policy $\bar{\pi}$.*

Theorem 2. *Let $\bar{Q}^*$ be the optimal world action-value function for a task M. Then $\bar{Q}^*$ has mastery.* □

Algorithm 1: Q-learning for WVFs

Initialise: WVF $\bar{Q}$, goal buffer $\mathcal{G}$, learning rate α
foreach *episode* **do**
  Observe initial state $s \in \mathcal{S}$ and sample $g \in \mathcal{G}$
  while *episode is not done* **do**
    $a \leftarrow \begin{cases} \arg\max_{a \in \mathcal{A}} \bar{Q}(s,g,a) & \text{w.p. } 1-\varepsilon \\ \text{a random action} & \text{w.p. } \varepsilon \end{cases}$
    Execute a, observe reward r and next state s'
    if s' *is absorbing* **then** $\mathcal{G} \leftarrow \mathcal{G} \cup \{s\}$
    for $g' \in \mathcal{G}$ **do**
      $\bar{r} \leftarrow \bar{R}_{\text{MIN}}$ **if** $g' \neq s$ and $s \in \mathcal{G}$ **else** r
      $\delta \leftarrow \left[\bar{r} + \max_{a'} \bar{Q}(s',g',a')\right] - \bar{Q}(s,g',a)$
      $\bar{Q}(s,g',a) \leftarrow \bar{Q}(s,g',a) + \alpha\delta$
    $s \leftarrow s'$

Finally, we note that while GVFs can also be used to construct goal-oriented value functions, questions remain open as to the origins of goals and how to define goal-specific rewards. WVFs are a subset of GVFs that answer these questions—goals are simply states with terminal transitions, while goal rewards are specified by $\bar{R}$. Answering these questions in this way confers several advantages, which we desribe below.

Planning with World Value Functions

If the agent's goal space coincides with the state space ($\mathcal{G} = \mathcal{S}$), then an optimal WVF will implicitly encode the dynamics of the world. We can then estimate the transition probabilities for each $s,a \in \mathcal{S} \times \mathcal{A}$ using only the reward function and optimal WVF. That is, $P(s,a,s')$ for all $s' \in \mathcal{S}$ can be obtained by simply solving the system of Bellman optimality equations given by each goal $g \in \mathcal{S}$: $\bar{Q}^*(s,g,a) = \sum_{s' \in \mathcal{S}} p(s,a,s')\left[\bar{R}(s,g,a,s') + \bar{V}^*(s',g)\right]$. In practice, if the transition probabilities are known to be non-zero only in a neighbourhood $\mathcal{N}(s)$ of state s (as is common in most domains), then we only require that the WVF be near-optimal for $s', g \in \mathcal{N}(s) \times \mathcal{N}(s)$.

Multitask Transfer with World Value Functions

We now show the advantage of WVFs under the assumption that an agent may be faced with solving several tasks within the same world. In other words, we assume that all tasks share the same state space, action space and dynamics, but differ in their reward functions. Formally, we define the world as a background MDP $M_0 = (\mathcal{S}_0, \mathcal{A}_0, P_0, R_0)$ with its own state space, action space, transition dynamics and background reward function. Any individual task M is defined by a reward function $R^{\tau}_M(s,a)$ that is non-zero only

for transitions entering terminal states. The reward function for the resulting MDP is then simply $R_M(s,a,s') \coloneqq R_0(s,a,s') + R^\tau_M(s,a)$. We denote the set of all such tasks as $\mathcal{M}$, and the corresponding set of optimal WVFs as $\bar{\mathcal{Q}}^*$.

One immediate result is that if tasks share the same background MDP, then their WVFs share the same world policy. That is, the agent has the same notion of goals and how to reach them, regardless of the current task. Similarly, if we require that the world policies be the same across tasks, then we have that the tasks must come from the same world. This is formalised by Theorem 3 below.

Theorem 3. *Let $\bar{\mathcal{Q}}^*$ be the set of optimal world $\bar{Q}$-value functions with mastery of tasks in $\mathcal{M}$. Then for all $s \neq g \in \mathcal{S} \times \mathcal{G}$,*

$$\bar{\pi}^*(s,g) \in \arg\max_{a \in \mathcal{A}} \bar{Q}^*_{M_1}(s,g,a)$$
$$\iff$$
$$\bar{\pi}^*(s,g) \in \arg\max_{a \in \mathcal{A}} \bar{Q}^*_{M_2}(s,g,a) \; \forall M_1, M_2 \in \mathcal{M}.$$

□

Since all tasks in $\mathcal{M}$ share the same dynamics (and consequently the same world policy), their corresponding WVFs can be written as $\bar{Q}^*_M(s,g,a) = G^*_{s,g,a} + \bar{R}^\tau_M(s',a')$ for some $s', a' \in \mathcal{S} \times \mathcal{A}$, where $G^*_{s,g,a}$ is a constant across tasks that represents the sum of rewards starting from s and taking action a up until g, but not including the terminal reward. Using this fact, Theorem 4 shows that the optimal value function and policy for any task can be obtained zero-shot from an arbitrary WVF given the task-specific rewards:

Theorem 4. *Let R^τ_M be the given task-specific reward function for a task $M \in \mathcal{M}$, and let $\bar{Q}^* \in \bar{\mathcal{Q}}^*$ be an arbitrary WVF. Let $\tilde{\bar{V}}_M(s,g)$ be the estimated WVF of M given by*

$$\max_{a \in \mathcal{A}} \bar{Q}^*(s,g,a) + \left(\max_{a \in \mathcal{A}} R^\tau_M(g,a) - \max_{a \in \mathcal{A}} \bar{Q}^*(g,g,a) \right).$$

Then, (i) for all $g \in \mathcal{G}$ reachable from $s \in \mathcal{S}$, $\bar{V}^*_M(s,g) = \tilde{\bar{V}}_M(s,g)$. (ii) $V^*_M(s) = \max_{g \in \mathcal{G}} \tilde{\bar{V}}(s,g)$, and $\pi^*_M(s) \in \arg\max_{a \in \mathcal{A}} \bar{Q}^*(s, \arg\max_{g \in \mathcal{G}} \tilde{\bar{V}}_M(s,g), a)$.

□

This has several important implications for transfer learning. Most importantly, an agent can learn an arbitrary WVF with unsupervised pretraining and then solve any new task by simply estimating the reward function (from experience or demonstrations).

Experiments

We empirically validate the properties of WVFs in the Four Rooms domain (Sutton, Precup, and Singh 1999), where an agent is required to reach various goal positions. The agent can move in any of the four cardinal directions at each timestep (with reward -0.1), but colliding with a wall leaves it in the same state. The agent also has a "done" action that can choose to terminate at any position (with reward 10 if it is the goal of the current task). For each of the experiments below, we consider the case where the agent's goals are the entire state space ($\mathcal{G} = \mathcal{S}$).

Learning World Value Functions

To verify that WVFs can be learned with standard model-free algorithms, we train an agent using Q-learning on a task where it must learn to navigate to either the middle of the top-left or bottom-right rooms. Figure 1a shows the learned WVF, which is generated by plotting the value functions for every goal position and displaying them at their respective xy positions. Note how the values with respect the "top-left" and "bottom-right" goals are high (red), reflecting the high rewards the agent receives for reaching the goals it intended to achieve. Figure 1b shows a close-up view of the learned WVF around the "top-left" goal. We can observe from the value gradient of the plots that the WVF does indeed learn how to reach all positions in the gridworld. We can then maximise over goals to obtain the regular value function and policy (Figure 1c).

Finally, we plot the returns obtained during the learning of both the WVF and regular value function, with results given by Figure 1d. Interestingly, this result indicates that it is more sample efficient to learn a WVF, despite the fact that it has an additional dimension that must be learned. We theorise this is due to the induced goal-directed exploration of Algorithm 1, which is far superior to ε-greedy exploration.

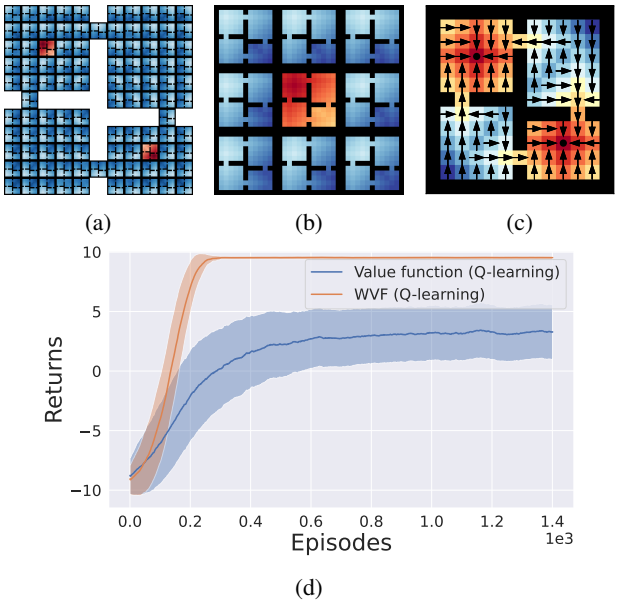
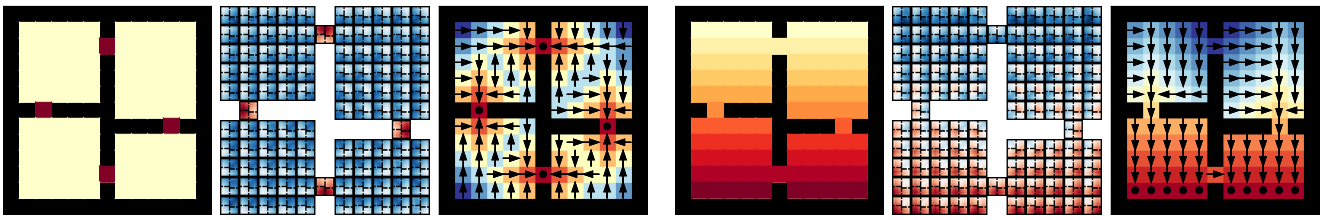


Figure 1: (a) Learned WVF. (b) Close-up view of the WVF for "top-left" goal. (c) Inferred values and policy for solving the current task. (d) Returns during training for both WVFs and regular value functions. Returns are calculated by greedy evaluation at the end of each episode. Mean and standard deviation over 25 random seeds are shown.

Multitask Transfer with World Value Functions

Having learned the WVF for the above task, we now show that it can be used to solve subsequent tasks by combining the WVF with the task-specific rewards as per Theorem 4. Critically, this means that any new task an agent might face



(a) Navigating to the hallways.

(b) Navigating to the bottom of the grid.

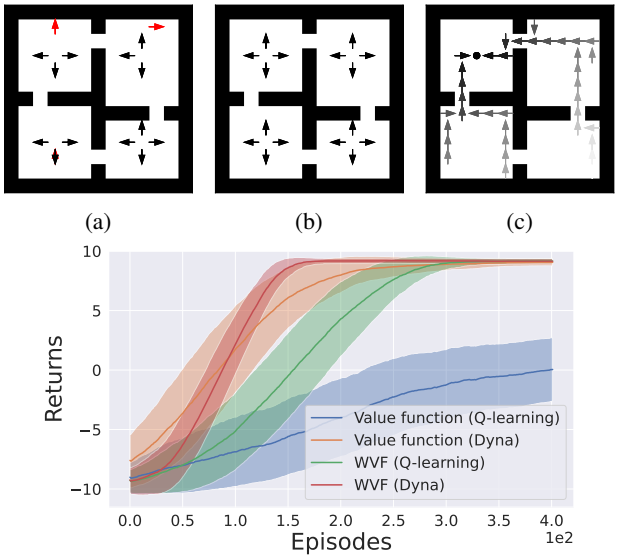
Figure 2: From left to right on each figure: The task specific rewards, the inferred WVF using Theorem 4, and the inferred values and policy from maximising over goals for (a) reaching any of the hallways, and (b) reaching the bottom of the grid.

can simply be solved by estimating its reward function, reducing the RL problem to a supervised learning one. We consider two new tasks: navigating to any of the hallways, and navigating to the bottom of the grid. Figures 2a and 2b illustrate the reward functions and subsequent WVFs and policies for these two tasks respectively. Importantly, given the reward functions (which can be estimated from data), the optimal policies can immediately be computed without further learning.

Planning with World Value Functions

Finally, we demonstrate that the transition probabilities can be inferred from the learned WVF. Figures 3 (left) and (middle) respectively show the transitions inferred by solving the Bellman equations with $s', g \in \mathcal{S} \times \mathcal{S}$ and $s', g \in \mathcal{N}(s) \times \mathcal{N}(s)$. For each, we infer the next state probabilities for taking each cardinal action at the center of each room, and place the corresponding arrow in the state with highest probability. The red arrows in Figure 3 (left) correspond to incorrectly inferred next states, which is a consequence of the learned WVF not being near optimal at all states for all goals. Figure 3 (middle) shows that in practice, if the WVF is not near-optimal, we can still infer dynamics by using $s', g \in \mathcal{N}(s) \times \mathcal{N}(s)$. Figure 3 (right) shows sample trajectories for following the optimal policy using the inferred transition probabilities. The gray-scale color of each arrow corresponds to the normalised value prediction for that state.

Finally, we also demonstrate that these inferred dynamics can be used to improve planning by integrating WVFs into a Dyna-style architecture (Sutton 1990). Our approach is illustrated by Algorithm 2 in the Appendix, where we combine both model-free and model-based updates to learn the WVF. Importantly, since the dynamics are inferred from the WVF, using them to plan (Dyna-style) at the start of training is detrimental, since the WVF will make incorrect predictions. We mitigate this by computing the mean-squared error of the Bellman equations using the inferred next state, $MSE = \frac{1}{|\mathcal{N}(s)|} \sum_{g \in \mathcal{N}(s)} \left(\bar{Q}(s,g,a) - \left[\bar{R}(s,g,a,s') + \bar{V}(s',g)\right]\right)$, and only use the WVF to plan when the error is less than a threshold ($MSE \leq 10^{-5}$). We compare our approach to Q-learning for both WVFs and regular value functions, as well as Dyna for regular value functions. The results in Figure 3d illustrate that sample efficiency can be greatly improved by integrating the planning capabilities of WVFs.



(d)

Figure 3: (a–b) Inferred one-step transitions. Red arrows indicate incorrect predictions. (c) Imagined rollouts using the learned WVF. (d) Returns during training for both WVFs and regular value functions, with and without planning. Mean and standard deviation over 25 random seeds are shown.

Conclusion

We introduced a new form of goal-oriented value function that encodes knowledge about how to solve all possible goal-reaching tasks in the world. This value function can be learned in a sample efficient manner, and can subsequently be used to infer the dynamics of the environment for model-based planning, or solve new tasks zero-shot given just their terminal rewards. An obvious path for future work is to extend these results to the stochastic high-dimensional setting. While prior work has demonstrated that WVFs can be learned with neural networks (Nangue Tasse, James, and Rosman 2020), planning in high-dimensional environments is still an open challenge; WVFs may provide a promising avenue for unifying both learning and planning in this space. Overall, our work is a step towards more general agents capable of solving any new task they may encounter.

References

Levine, S.; Finn, C.; Darrell, T.; and Abbeel, P. 2016. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1): 1334–1373.

Nangue Tasse, G.; James, S.; and Rosman, B. 2020. A Boolean Task Algebra for Reinforcement Learning. *Advances in Neural Information Processing Systems*, 33.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. 2017. Mastering the game of go without human knowledge. *Nature*, 550(7676): 354.

Sutton, R.; Barto, A.; et al. 1998. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge.

Sutton, R.; Modayil, J.; Delp, M.; Degris, T.; Pilarski, P.; White, A.; and Precup, D. 2011. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, 761–768.

Sutton, R.; Precup, D.; and Singh, S. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2): 181–211.

Sutton, R. S. 1990. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine learning proceedings 1990*, 216–224. Elsevier.

van Niekerk, B.; James, S.; Earle, A.; and Rosman, B. 2019. Composing Value Functions in Reinforcement Learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, 6401–6409. PMLR.

Veeriah, V.; Oh, J.; and Singh, S. 2018. Many-goals reinforcement learning. *arXiv preprint arXiv:1806.09605*.

Proofs of theoretical results

Theorem 1. *Let $M = (\mathcal{S}, \mathcal{A}, P, R)$ be a deterministic task with optimal action-value function Q^* and optimal world action-value function $\bar{Q}^*$. Then for all (s, a, s') in $\mathcal{S} \times \mathcal{A} \times \mathcal{S}$, we have (i) $R(s, a, s') = \max_{g \in \mathcal{G}} \bar{R}(s, g, a, s')$, and (ii) $Q^*(s, a) = \max_{g \in \mathcal{G}} \bar{Q}^*(s, g, a)$.*

Proof.

(i):

$$\begin{aligned}
&\max_{g \in \mathcal{G}} \bar{R}_M(s, g, a, s') \\
&= \begin{cases} \max\{\bar{R}_{\text{MIN}}, R_M(s, a, s')\}, & \text{if } s \in \mathcal{G} \\ \max_{g \in \mathcal{G}} R_M(s, a, s'), & \text{otherwise.} \end{cases} \\
&= R_M(s, a, s') \\
&(\bar{R}_{\text{MIN}} \leq R_{\text{MIN}} \leq R_M(s, a, s') \text{ by definition}).
\end{aligned}$$

(ii): Each g in $\mathcal{G}$ can be thought of as defining an MDP $M_g := (\mathcal{S}, \mathcal{A}, P, R_{M_g})$ with reward function $R_{M_g}(s, a, s') := \bar{R}_M(s, g, a, s')$ and optimal action-value function $Q^*_{M_g}(s, a) = \bar{Q}^*_M(s, g, a)$. Then using (i) we have $R_M(s, a, s') = \max_{g \in \mathcal{G}} R_{M_g}(s, a, s')$ and from van Niekerk et al. (2019, Corollary 1), we have that $Q^*_M(s, a) = \max_{g \in \mathcal{G}} Q^*_{M_g}(s, a) = \max_{g \in \mathcal{G}} \bar{Q}^*_M(s, g, a)$.

□

Theorem 2. *Let $\bar{Q}^*$ be the optimal world action-value function for a task M. Then $\bar{Q}^*$ has mastery.*

Proof. Let each g in $\mathcal{G}$ define an MDP M_g with reward function

$$R_{M_g} := \bar{R}_M(s, g, a, s')$$

for all (s, a, s') in $\mathcal{S} \times \mathcal{A} \times \mathcal{S}$. Define

$$\pi^*_g(s) \in \arg\max_{a \in \mathcal{A}} Q^*_{M,g}(s, a) \text{ for all } s \in \mathcal{S}.$$

If g *is* reachable from $s \in \mathcal{S} \setminus \{g\}$, then we show that following π^*_g must reach g. Since π^*_g is proper, it must reach a state $g' \in \mathcal{G}$ such that the transition $(g', \pi^*_g(g'), s')$ is terminal. Assume $g' \neq g$. Let π_g be a policy that produces the shortest trajectory to g. Let $G^{\pi^*_g}$ and G^{π_g} be the returns for the respective policies. Then,

$$\begin{aligned}
&G^{\pi^*_g} \geq G^{\pi_g} \\
&\implies G^{\pi^*_g}_{T-1} + R_{M_g}(g', \pi^*_g(g'), s') \geq G^{\pi_g}, \\
&\text{where } G^{\pi^*_g}_{T-1} = \sum_{t=0}^{T-1} R_{M_g}(s_t, \pi^*_g(s_t), s_{t+1}) \\
&\text{and } T \text{ is the time at which } g' \text{ is reached.} \\
&\implies G^{\pi^*_g}_{T-1} + \bar{R}_{\text{MIN}} \geq G^{\pi_g}, \text{ since } g \neq g' \in \mathcal{G} \\
&\implies \bar{R}_{\text{MIN}} \geq G^{\pi_g} - G^{\pi^*_g}_{T-1} \\
&\implies (R_{\text{MIN}} - R_{\text{MAX}})D \geq G^{\pi_g} - G^{\pi^*_g}_{T-1}, \\
&\quad \text{by definition of } \bar{R}_{\text{MIN}} \\
&\implies G^{\pi^*_g}_{T-1} - R_{\text{MAX}}D \geq G^{\pi_g} - R_{\text{MIN}}D, \\
&\quad \text{since } G^{\pi_g} \geq R_{\text{MIN}}D \\
&\implies G^{\pi^*_g}_{T-1} - R_{\text{MAX}}D \geq 0 \\
&\implies G^{\pi^*_g}_{T-1} \geq R_{\text{MAX}}D.
\end{aligned}$$

But this is a contradiction, since the result obtained by following an optimal trajectory up to a terminal state without the reward for entering the terminal state must be strictly less than receiving R_{MAX} for every step of the longest possible optimal trajectory. Hence we must have $g' = g$.

□

Theorem 3. *Let $\bar{\mathcal{Q}}^*$ be the set of optimal world $\bar{Q}$-value functions with mastery of tasks in $\mathcal{M}$. Then for all $s \neq g \in \mathcal{S} \times \mathcal{G}$,*

$$\bar{\pi}^*(s,g) \in \arg\max_{a\in\mathcal{A}} \bar{Q}^*_{M_1}(s,g,a)$$

$$\iff$$

$$\bar{\pi}^*(s,g) \in \arg\max_{a\in\mathcal{A}} \bar{Q}^*_{M_2}(s,g,a)\ \forall M_1, M_2 \in \mathcal{M}.$$

Proof. Let $g \in \mathcal{G}, s \in \mathcal{S} \setminus \{g\}$.

If g *is* reachable from s, then we are done since $\bar{Q}^*_{M_1}$ and $\bar{Q}^*_{M_2}$ have mastery (Theorem 2).

If g is unreachable from s, then for all (a, s') in $\mathcal{A} \times \mathcal{S}$ we have

$$\bar{R}_{M_1}(s,g,a,s') = \begin{cases} \bar{R}_{\text{MIN}}, & \text{if } s' \text{ is absorbing} \\ r_{s,a,s'}, & \text{otherwise} \end{cases}$$

where $r_{s,a,s'}$ is the reward for the non-terminal transition (s,a,s')

$$= \bar{R}_{M_2}(s,g,a,s')$$

$$\implies \bar{Q}^*_{M_1}(s,g,a) = \bar{Q}^*_{M_2}(s,g,a).$$

□

Theorem 4. *Let R^τ_M be the given task-specific reward function for a task $M \in \mathcal{M}$, and let $\bar{Q}^* \in \bar{\mathcal{Q}}^*$ be an arbitrary WVF. Let $\tilde{\bar{V}}_M(s,g)$ be the estimated WVF of M given by*

$$\max_{a\in\mathcal{A}} \bar{Q}^*(s,g,a) + \left(\max_{a\in\mathcal{A}} R^\tau_M(g,a) - \max_{a\in\mathcal{A}} \bar{Q}^*(g,g,a)\right).$$

Then,

(i) for all $g \in \mathcal{G}$ reachable from $s \in \mathcal{S}$, $\bar{V}^*_M(s,g) = \tilde{\bar{V}}_M(s,g)$.

(ii) $V^*_M(s) = \max_{g\in\mathcal{G}} \tilde{\bar{V}}(s,g)$, and $\pi^*_M(s) \in \arg\max_{a\in\mathcal{A}} \bar{Q}^*(s, \arg\max_{g\in\mathcal{G}} \tilde{\bar{V}}_M(s,g), a)$.

Proof.

(i): Let $g \in \mathcal{G}$ be a goal reachable from state $s \in \mathcal{S}$. If $g = s$, then

$$\max_{a\in\mathcal{A}} \bar{Q}^*(s,g,a) + \left(\max_{a\in\mathcal{A}} R^\tau_M(g,a) - \max_{a\in\mathcal{A}} \bar{Q}^*(g,g,a)\right)$$

$$= \max_{a\in\mathcal{A}} R^\tau(g,a) + \left(\max_{a\in\mathcal{A}} R^\tau_M(g,a) - \max_{a\in\mathcal{A}} R^\tau(g,a)\right)$$

$$= \max_{a\in\mathcal{A}} R^\tau_M(g,a) = \bar{V}^*_M(s,g)$$

If $g \neq s$, then

$$\max_{a\in\mathcal{A}} \bar{Q}^*(s,g,a) + \left(\max_{a\in\mathcal{A}} R^\tau_M(g,a) - \max_{a\in\mathcal{A}} \bar{Q}^*(g,g,a)\right)$$

$$= \max_{a\in\mathcal{A}} [G^*_{s,g,a} + R^\tau(g, a^{\max})] + \left(\max_{a\in\mathcal{A}} R^\tau_M(g,a) - \max_{a\in\mathcal{A}} R^\tau(g,a)\right),$$

follows from Theorem 2 and Theorem 3

$$= \max_{a\in\mathcal{A}} G^*_{s,g,a} + R^\tau(g,a^{\max}) + (R^\tau_M(g,a^{\max}_M) - R^\tau(g,a^{\max}))$$

$$= \max_{a\in\mathcal{A}} [G^*_{s,g,a} + \bar{R}^\tau_M(g,a^{\max}_M)] = \bar{V}^*_M(s,g)$$

(ii): Follows directly from (i) above and Theorem 3.

□

Algorithms

Algorithm 2: Dyna for WVFs using inferred transition functions

Initialise: WVF $\bar{Q}$, Reward function R, goal buffer $\mathcal{G}$, learning rate α

foreach *episode* **do**

  Observe initial state $s \in \mathcal{S}$ and sample $g \in \mathcal{G}$

  while *episode is not done* **do**

    $a \leftarrow \begin{cases} \arg\max_{a\in\mathcal{A}} \bar{Q}(s,g,a) & \text{w.p. } 1-\varepsilon \\ \text{a random action} & \text{w.p. } \varepsilon \end{cases}$

    Execute a, observe reward r and next state s'

    $R(s,a,.) \leftarrow r$

    if *s' is absorbing* **then** $\mathcal{G} \leftarrow \mathcal{G} \cup \{s\}$

    for $g' \in \mathcal{G}$ **do**

      $\bar{r} \leftarrow \bar{R}_{\text{MIN}}$ **if** $g' \neq s$ and $s \in \mathcal{G}$ **else** r

      $\delta \leftarrow \left[\bar{r} + \max_{a'} \bar{Q}(s',g',a')\right] - \bar{Q}(s,g',a)$

      $\bar{Q}(s,g',a) \leftarrow \bar{Q}(s,g',a) + \alpha\delta$

    repeat N **times**

      $s \leftarrow$ random previous state

      $a \leftarrow$ random previous action taken in s

      $r \leftarrow R(s,a,.)$

      $s' \leftarrow$ Solving $\mathcal{N}(s)$ Bellman equations

      $MSE \leftarrow \frac{1}{|\mathcal{N}(s)|} \sum_{g\in\mathcal{N}(s)} (\bar{Q}(s,g,a) - \left[\bar{R}(s,g,a,s') + \bar{V}(s',g)\right])$

      if $MSE \leq$ *threshold* **then**

        for $g' \in \mathcal{G}$ **do**

          $\bar{r} \leftarrow \bar{R}_{\text{MIN}}$ **if** $g' \neq s$ and $s \in \mathcal{G}$ **else** r

          $\delta \leftarrow \left[\bar{r} + \max_{a'} \bar{Q}(s',g',a')\right] - \bar{Q}(s,g',a)$

          $\bar{Q}(s,g',a) \leftarrow \bar{Q}(s,g',a) + \alpha\delta$

    $s \leftarrow s'$