Unifying Task Specification in Reinforcement Learning

Martha White [1]

Abstract

Reinforcement learning tasks are typically specified as Markov decision processes. This formalism has been highly successful, though specifications often couple the dynamics of the environment and the learning objective. This lack of modularity can complicate generalization of the task specification, as well as obfuscate connections between different task settings, such as episodic and continuing. In this work, we introduce the RL task formalism, that provides a unification through simple constructs including a generalization to transition-based discounting. Through a series of examples, we demonstrate the generality and utility of this formalism. Finally, we extend standard learning constructs, including Bellman operators, and extend some seminal theoretical results, including approximation errors bounds. Overall, we provide a well-understood and sound formalism on which to build theoretical results and simplify algorithm use and development.

1. Introduction

Reinforcement learning is a formalism for trial-and-error interaction between an agent and an unknown environment. This interaction is typically specified by a Markov decision process (MDP), which contains a transition model, reward model, and potentially discount parameters γ specifying a discount on the sum of future values in the return. Domains are typically separated into two cases: episodic problems (finite horizon) and continuing problems (infinite horizon). In episodic problems, the agent reaches some terminal state, and is teleported back to a start state. In continuing problems, the agent interaction is continual, with a discount to ensure a finite total reward (e.g., constant $\gamma < 1$).

This formalism has a long and successful tradition, but is limited in the problems that can be specified. Progressively there have been additions to specify a broader range of objectives, including options (Sutton et al., 1999), state-based discounting (Sutton, 1995; Sutton et al., 2011) and interest functions (Reza and Sutton, 2010; Sutton et al., 2016). These generalizations have particularly been driven by off-policy learning and the introduction of general value functions for Horde (Sutton et al., 2011; White, 2015), where predictive knowledge can be encoded as more complex prediction and control tasks. Generalizations to problem specifications provide exciting learning opportunities, but can also reduce clarity and complicate algorithm development and theory. For example, options and general value functions have significant overlap, but because of different terminology and formalization, the connections are not transparent. Another example is the classic divide between episodic and continuing problems, which typically require different convergence proofs (Bertsekas and Tsitsiklis, 1996; Tsitsiklis and Van Roy, 1997; Sutton et al., 2009) and different algorithm specifications.

In this work, we propose a formalism for reinforcement learning task specification that unifies many of these generalizations. The focus of the formalism is to separate the specification of the dynamics of the environment and the specification of the objective within that environment. Though natural, this represents a significant change in the way tasks are currently specified in reinforcement learning and has important ramifications for simplifying implementation, algorithm development and theory. The paper consists of two main contributions. First, we demonstrate the utility of this formalism by showing unification of previous tasks specified in reinforcement learning, including options, general value functions and episodic and continuing, and further providing case studies of utility. We demonstrate how to specify episodic and continuing tasks with only modifications to the discount function, without the addition of states and modifications to the underlying Markov decision process. This enables a unification that significantly simplifies implementation and easily generalizes theory to cover both settings. Second, we prove novel contraction bounds on the Bellman operator for these generalized RL tasks, and show that previous bounds for both episodic and continuing tasks are subsumed by this more general result. Overall, our goal is to provide an RL task formalism that requires minimal modifications to previous task specification, with significant gains in simplicity and unification across common settings.

[1]Department of Computer Science, Indiana University. Correspondence to: Martha White <martha@indiana.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

2. Generalized problem formulation

We assume the agent interacts with an environment formalized by a Markov decision process (MDP): $(\mathcal{S}, \mathcal{A}, \Pr)$ where $\mathcal{S}$ is the set of states, $n = |\mathcal{S}|$; $\mathcal{A}$ is the set of actions; and $\Pr : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition probability function where $\Pr(s, a, s')$ is the probability of transitioning from state s into state s' when taking action a. A **reinforcement learning task** (RL task) is specified on top of these transition dynamics, as the tuple $(\mathcal{P}, r, \gamma, \mathbf{i})$ where

1. $\mathcal{P}$ is a set of policies $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$;
2. the reward function $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ specifies reward received from (s, a, s');
3. $\gamma : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is a **transition-based discount function**[1];
4. $\mathbf{i} : \mathcal{S} \to [0, \infty)$ is an interest function that specifies the user defined interest in a state.

Each task could have different reward functions within the same environment. For example, in a navigation task within an office, one agent could have the goal to navigate to the kitchen and the other the conference room. For a reinforcement learning task, whether prediction or control, a set or class of policies is typically considered. For prediction (policy evaluation), we often select one policy and evaluate its long-term discounted reward. For control, where a policy is learned, the set of policies may consist of all policies parameterized by weights that specify the action-value from states, with the goal to find the weights that yield the optimal policy. For either prediction or control in an RL task, we often evaluate the return of a policy: the cumulative discounted reward obtained from following that policy

$$G_t = \sum_{i=0}^{\infty} \left(\prod_{j=0}^{i-1} \gamma(s_{t+j}, a_{t+j}, s_{t+1+j}) \right) R_{t+1+i}$$

where $\prod_{j=0}^{-1} \gamma(s_{t+j}, a_{t+j}, s_{t+1+j}) := 1$. Note that this subsumes the setting with a constant discount $\gamma_c \in [0, 1)$, by using $\gamma(s, a, s') = \gamma_c$ for every s, a, s' and so giving $\prod_{j=0}^{i-1} \gamma(s_{t+j}, a_{t+j}, s_{t+1+j}) = \gamma_c^i$ for $i > 0$ and $\gamma_c^0 = 1$ for $i = 0$. As another example, the end of the episode, $\gamma(s, a, s') = 0$, making the product of these discounts zero and so terminating the recursion. We further explain how transition-based discount enables specification of episodic tasks and discuss the utility of the generalization to transition-based discounting throughout this paper. Finally, the interest function $\mathbf{i}$ specifies the degree of importance of each state for the task. For example, if an agent is only interested in learning an optimal policy for a subset of the environment, the interest function could be set to one for those states and to zero otherwise.

[1] We describe a further probabilistic generalization in Appendix A; much of the treatment remains the same, but the notation becomes cumbersome and the utility obfuscated.

We first explain the specification and use of such tasks, and then define a generalized Bellman operator and resulting algorithmic extensions and approximation bounds.

2.1. Unifying episodic and continuing specification

The RL task specification enables episodic and continuing problems to be easily encoded with only modification to the transition-based discount. Previous approaches, including the absorbing state formulation (Sutton and Barto, 1998b) and state-based discounting (Sutton, 1995; Reza and Sutton, 2010; Sutton et al., 2011)(van Hasselt, 2011, Section 2.1.1), require special cases or modifications to the set of states and underlying MDP, coupling task specification and the dynamics of the environment.

We demonstrate how transition-based discounting seamlessly enables episodic or continuing tasks to be specified in an MDP via a simple chain world. Consider the chain world with three states s_1, s_2 and s_3 in Figure 1. The start state is s_1 and the two actions are *right* and *left*. The reward is -1 per step, with termination occurring when taking action *right* from state s_3, which causes a transition back to state s_1. The discount is 1 for each step, unless specified otherwise. The interest is set to 1 in all states, which is the typical case, meaning performance from each state is equally important.

Figure 1a depicts the classical approach to specifying episodic problems using an absorbing state, drawn as a square. The agent reaches the goal—transitioning right from state s_3—then forever stays in the absorbing state, receiving a reward of zero. This encapsulates the definition of the return, but does not allow the agent to start another episode. In practice, when this absorbing state is reached, the agent is "teleported" to a start state to start another episode. This episodic interaction can instead be represented the same way as a continuing problem, by specifying a transition-based discount $\gamma(s_3, \textit{right}, s_1) = 0$. This defines the same return, but now the agent simply transitions normally to a start state, and no hypothetical states are added.

To further understand the equivalence, consider the updates made by TD (see equation (3)). Assume linear function approximation with feature function $\mathbf{x} : \mathcal{S} \to \mathbb{R}^d$, with weights $\mathbf{w} \in \mathbb{R}^d$. When the agent takes action *right* from s_3, the agent transitions from s_3 to s_1 with probability one and so $\gamma_{t+1} = \gamma(s_3, \textit{right}, s_1) = 0$. This correctly gives

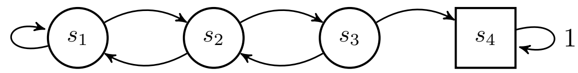
$$\delta_t = r_{t+1} + \gamma_{t+1}\mathbf{x}(s_1)^\top \mathbf{w} - \mathbf{x}(s_3)^\top \mathbf{w} = r_{t+1} - \mathbf{x}(s_3)^\top \mathbf{w}$$

and correctly clears the eligibility trace for the next step

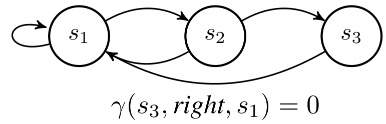
$$\mathbf{e}_{t+1} = \lambda_{t+1}\gamma_{t+1}\mathbf{e}_t + \mathbf{x}(s_1) = \mathbf{x}(s_1).$$

The stationary distribution is also clearly equal to the original episodic task, since the absorbing state is not used in the computation of the stationary distribution.

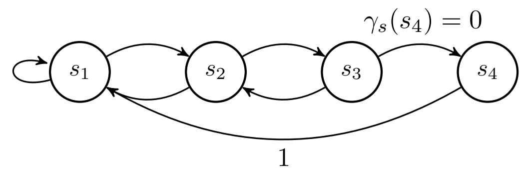
Another strategy is to still introduce hypothetical states, but use state-based γ, as discussed in Figure 1c. Unlike absorbing states, the agent does not stay indefinitely in the hypothetical state. When the agent goes *right* from s_3, it transitions to hypothetical state s_4, and then transition deterministically to the start state s_1, with $\gamma_s(s_4) = 0$. As before, we get the correct update, because $\gamma_{t+1} = \gamma_s(s_4) = 0$. Because the stationary distribution has some non-zero probability in the hypothetical state s_4, we must set $\mathbf{x}(s_4) = \mathbf{x}(s_1)$ (or $\mathbf{x}(s_4) = \mathbf{0}$). Otherwise, the value of the hypothetical state will be learned, wasting function approximation resources and potentially modifying the approximation quality of the value in other states. We could have tried state-based discounting without adding an additional state s_4. However, this leads to incorrect value estimates, as depicted in Figure 1d; the relationship between transition-based and state-based is further discussed in Appendix B.1. Overall, to keep the specification of the RL task and the MDP separate, transition-based discounting is necessary to enable the unified specification of episodic and continuing tasks.



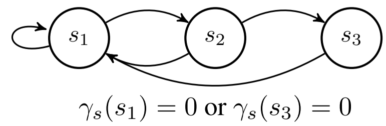
(a) Absorbing state formulation.



(b) Transition-based termination, $\gamma(s_3, \textit{right}, s_1) = 0$.



(c) State-based termination with $\gamma_s(s_4) = 0$.



(d) Incorrect state-based termination.

Figure 1: Three different ways to represent episodic problems as continuing problems. For (c), the state-based discount cannot represent the episodic chain problem without adding states. To see why, consider the two cases for representing termination: $\gamma_s(s_1) = 0$ or $\gamma_s(s_3) = 0$. For simplicity, assume that $\pi(s, \textit{right}) = 0.75$ for all states $s \in \{s_1, s_2, s_3\}$ and transitions are deterministic. If $\gamma_s(s_3) = 0$, then the value for taking action *right* from s_2 is $r(s_2, \textit{right}, s_3) + \gamma_s(s_3)\mathbf{v}_\pi(s_3) = -1$ and the value for taking action *right* from s_3 is $r(s_3, \textit{right}, s_1) + \gamma_s(s_1)\mathbf{v}_\pi(s_1) \neq -1$, which are both incorrect. If $\gamma_s(s_1) = 0$, then the value of taking action *right* from s_3 is $-1 + \gamma_s(s_1)\mathbf{v}_\pi(s_1) = -1$, which is correct. However, the value of taking action *left* from s_2 is $-1 + \gamma_s(s_1)\mathbf{v}_\pi(s_1) = -1$, which is incorrect.

2.2. Options as RL tasks

The options framework (Sutton et al., 1999) generically covers a wide range of settings, with discussion about macro-actions, option models, interrupting options and intra-option value learning. These concepts at the time merited their own language, but with recent generalizations can be more conveniently cast as RL subtasks.

Proposition 1. *An option, defined as the tuple (Sutton et al., 1999, Section 2) $(\pi, \beta, \mathcal{I})$ with policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$, termination function $\beta : \mathcal{S} \to [0, 1]$ and an initiation set $\mathcal{I} \subset \mathcal{S}$ from which the option can be run, can be equivalently cast as an RL task.*

This proof is mainly definitional, but we state it as an explicit proposition for clarity. The discount function $\gamma(s, a, s') = 1 - \beta(s')$ for all s, a, s' specifies termination. The interest function, $\mathbf{i}(s) = 1$ if $s \in \mathcal{I}$ and $\mathbf{i}(s) = 0$ otherwise, focuses learning resources on the states of interest. If a value function for the policy is queried, it would only make sense to query it from these states of interest. If the policy for this option is optimized for this interest function, the policy should only be run starting from $s \in \mathcal{I}$, as elsewhere will be poorly learned. The rewards for the RL task correspond to the rewards associated with the MDP.

RL tasks generalize options, by generalizing termination conditions to transition-based discounting and by providing degrees of interest rather than binary interest. Further, the policies associated with RL subtasks can be used as macro-actions, to specify a semi-Markov decision process (Sutton et al., 1999, Theorem 1).

2.3. General value functions

In a similar spirit of abstraction as options, general value functions were introduced for single predictive or goal-oriented questions about the world (Sutton et al., 2011). The idea is to encode predictive knowledge in the form of value function predictions: with a collection or horde of prediction demons, this constitutes knowledge (Sutton et al., 2011; Modayil et al., 2014; White, 2015). The work on Horde (Sutton et al., 2011) and nexting (Modayil et al., 2014) provide numerous examples of the utility of the types of questions that can be specified by general value functions, and so by RL tasks, because general value functions can naturally can be specified as an RL task.

The generalization to RL tasks provide additional benefits for predictive knowledge. The separation into underlying MDP dynamics and task specification is particularly useful in off-policy learning, with the Horde formalism, where many demons (value functions) are learned off-policy. These demons share the underlying dynamics, and even feature representation, but have separate prediction and control tasks; keeping these separate from the MDP is key for avoiding complications (see Appendix B.2). Transition-based discounts, over state-based discounts, additionally enable the prediction of a *change*, caused by transitioning between states. Consider the taxi domain, described more fully in Section 3, where the agent's goal is to pick up and drop off passengers in a grid world with walls. The taxi agent may wish to predict the probability of hitting a wall, when following a given policy. This can be encoded by setting $\gamma(s, a, s) = 0$ if a movement action causes the agent to remain in the same state, which occurs when trying to move through a wall. In addition to episodic problems and hard termination, transition-based questions also enable soft termination for transitions. Hard termination uses $\gamma(s, a, s') = 0$ and soft termination $\gamma(s, a, s') = \epsilon$ for some small positive value ϵ. Soft terminations can be useful for incorporating some of the value of a policy right after the soft termination. If two policies are equivalent up to a transition, but have very different returns after the transition, a soft termination will reflect that difference. We empirically demonstrate the utility of soft termination in the next section.

3. Demonstration in the taxi domain

To better ground this generalized formalism and provide some intuition, we provide a demonstration of RL task specification. We explore different transition-based discounts in the taxi domain (Dietterich, 2000; Diuk et al., 2008). The goal of the agent is to take a passenger from a source platform to a destination platform, depicted in Figure 2. The agent receives a reward of -1 on each step, except for successful pickup and drop-off, giving reward 0. We modify the domain to include the orientation of the taxi, with additional cost for not continuing in the current orientation. This encodes that turning right, left or going backwards are more costly than going forwards, with additional negative rewards of -0.05, -0.1 and -0.2 respectively. This additional cost is further multiplied by a factor of 2 when there is a passenger in the vehicle. For grid size g and the number of pickup/dropoff locations l, the full state information is a 5-tuple: (x position of taxi $\in \{1, \ldots, g\}$, y position of taxi $\in \{1, \ldots, g\}$, location of passenger $\in \{1, \ldots, l+1\}$, location of destination $\in \{1, \ldots, l\}$, orientation of car $\in \{N, E, S, W\}$). The location of the passenger can be in one of the pickup/drop-off locations, or in the taxi. Optimal policies and value functions are computed iteratively, with an extensive number of iterations.

Figure 2 illustrates three policies for one part of the taxi domain, obtained with three different discount functions. The optimal policy is learned using a soft-termination, which takes into consideration the importance of approaching the passenger location with the right orientation to minimize turns after picking up the passenger. A suboptimal policy is in fact learned with hard termination, as the policy prefers to greedily minimize turns to get to the passenger. For further details, refer to the caption in Figure 2.

We also compare to a constant γ, which corresponds to an average reward goal, as demonstrated in Equation (8). The table in Figure 2(e) summarizes the results. Though in theory it should in fact recognize the relative values of orientation before and after picking up a passenger, and obtain the same solution as the soft-termination policy, in practice we find that numerical imprecision actually causes a suboptimal policy to be learned. Because most of the rewards are negative per step, small differences in orientation can be more difficult to distinguish amongst for an infinite discounted sum. This result actually suggests that having multiple subgoals, as one might have with RL subtasks, could enable better chaining of decisions and local evaluation of the optimal action. The utility of learning with a smaller γ_c has been previously described (Jiang et al., 2015), however, here we further advocate that enabling γ that provides subtasks is another strategy to improve learning.

4. Objectives and algorithms

With an intuition for the specification of problems as RL tasks, we now turn to generalizing some key algorithmic concepts to enable learning for RL tasks. We first generalize the definition of the Bellman operator for the value function. For a policy $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$, define $\mathbf{P}_\pi, \mathbf{P}_{\pi,\gamma} \in \mathbb{R}^{n \times n}$ and $\mathbf{r}_\pi, \mathbf{v}_\pi \in \mathbb{R}^n$, indexed by states $s, s' \in \mathcal{S}$,

$$\mathbf{P}_\pi(s, s') := \sum_{a \in \mathcal{A}} \pi(s, a) \mathrm{Pr}(s, a, s')$$

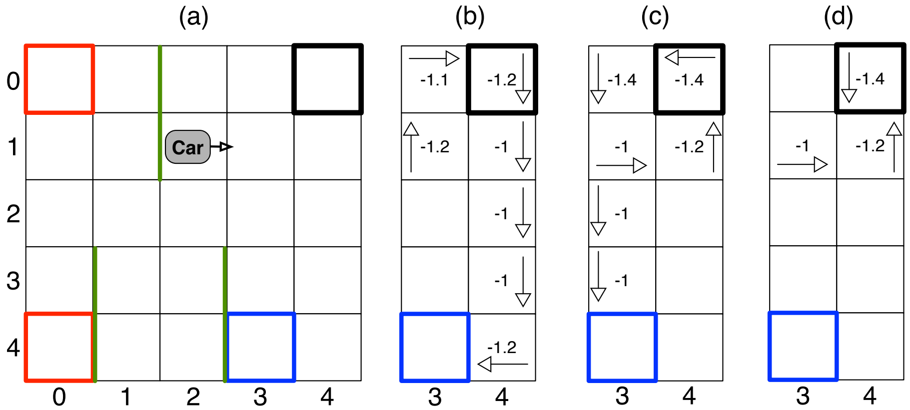
$$\mathbf{P}_{\pi,\gamma}(s, s') := \sum_{a \in \mathcal{A}} \pi(s, a) \mathrm{Pr}(s, a, s') \gamma(s, a, s')$$

$$\mathbf{r}_\pi(s) := \sum_{a \in \mathcal{A}} \pi(s, a) \sum_{s' \in \mathcal{S}} \mathrm{Pr}(s, a, s') r(s, a, s')$$

$$\mathbf{v}_\pi(s) := \mathbf{r}_\pi(s) + \sum_{s' \in \mathcal{S}} \mathbf{P}_{\pi,\gamma}(s, s') \mathbf{v}_\pi(s').$$

where $\mathbf{v}_\pi(s)$ is the expected return, starting from a state $s \in \mathcal{S}$. To compute a value function that satisfies this recursion, we define a Bellman operator. The Bellman operator has been generalized to include state-based discounting and a state-based trace parameter[2] (Sutton et al., 2016, Eq. 29).

[2] A generalization to state-based trace parameters has been considered (Sutton, 1995; Sutton and Barto, 1998b; Reza and



(e)

	Total Pickup and Dropoff	Added Cost for Turns
Trans-Soft	7.74 ± 0.03	5.54 ± 0.01
Trans-Hard	7.73 ± 0.03	5.83 ± 0.01
State-based	0.00 ± 0.00	18.8 ± 0.02
$\gamma_c = 0.1$	0.00 ± 0.00	2.48 ± 0.01
$\gamma_c = 0.3$	0.02 ± 0.01	2.49 ± 0.01
$\gamma_c = 0.5$	0.04 ± 0.01	2.51 ± 0.01
$\gamma_c = 0.6$	0.03 ± 0.01	2.49 ± 0.01
$\gamma_c = 0.7$	7.12 ± 0.03	4.52 ± 0.01
$\gamma_c = 0.8$	7.34 ± 0.03	4.62 ± 0.01
$\gamma_c = 0.9$	3.52 ± 0.06	4.57 ± 0.02
$\gamma_c = 0.99$	0.01 ± 0.01	2.45 ± 0.01

Figure 2: **(a)** The taxi domain, where the pickup/drop-off platforms are at (0,0), (0,4), (3,0) and (4,4). The Passenger P is at the source platform (4,4), outlined in black. The Car starts in (2,3), with orientation E as indicated the arrow, needs to bring the passenger to destination D platform at (3,0), outlined in blue. In **(b) - (d)**, there are simulated trajectories for policies learned using hard and soft termination.
(b) The optimal strategy, with γ(Car in source, Pickup, P in Car) = 0.1 and a discount 0.99 elsewhere. The sequence of taxi locations are $(3,3)$, $(3,4)$, $(4,4)$, $(4,4)$ with Pickup action, $(4,3)$, $(4,2)$, $(4,1)$, $(4,0)$, $(3,0)$. Successful pickup and drop-off with total reward -7.7.
(c) For γ(Car in source, Pickup, P in Car) = 0, the agent does not learn the optimal strategy. The agent minimizes orientation cost to the subgoal, not accounting for orientation after picking up the passenger. Consequently, it takes more left turns after pickup, resulting in more total negative reward. The sequence of locations are $(3,3)$, $(4,3)$, $(4,4)$, $(4,4)$ with Pickup action, $(3,4)$, $(3,3)$, $(3,2)$, $(3,1)$, $(3,0)$. Successful pickup and drop-off with total reward -8.
(d) For state-based γ(Car in source and P in Car) = 0, the agent remains around the source and does not complete a successful drop-off. The sequence of locations are $(3,3)$, $(4,3)$, $(4,4)$, $(4,4)$ with Pickup action, $(4,3)$, $(4,4)$, $(4,3)$.... The agent enters the source and pickups up the passenger. When it leaves to location (4,3), its value function indicates better value going to (4,4) because the negative return will again be cutoff by γ(Car in source and P in Car) = 0, even without actually performing a pickup. Since the cost to get to the destination is higher than the -2.6 return received from going back to $(4,4)$, the agent stays around $(4,4)$ indefinitely.
(e) Number of successful passenger pickup and dropoff, as well as additional cost incurred from turns, over 100 steps, with 5000 runs, reported for a range of constant γ_c and the policies in Figure 2. Due to numerical imprecision, several constant discounts do not get close enough to the passenger to pickup or drop-off. The state-based approach, that does not add additional states for termination, oscillates after picking up the passenger, and so constantly gets negative reward.

We further generalize the definition to the transition-based setting. The trace parameter $\lambda : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1]$ influences the fixed point and provides a modified (biased) return, called the λ-return; this parameter is typically motivated as a bias-variance trade-off parameter (Kearns and Singh, 2000). Because the focus of this work is on generalizing the discount, we opt for a simple constant λ_c in the main body of the text; we provide generalizations to transition-based trace parameters in the appendix.

The generalized Bellman operator $\mathrm{T}^{(\lambda)} : \mathbb{R}^n \to \mathbb{R}^n$ is

$$\mathrm{T}^{(\lambda)}\mathbf{v} := \mathbf{r}_\pi^\lambda + \mathbf{P}_\pi^\lambda \mathbf{v}, \quad \forall \mathbf{v} \in \mathbb{R}^n \tag{1}$$

$$\text{where} \quad \mathbf{P}_\pi^\lambda := (\mathbf{I} - \lambda_c \mathbf{P}_{\pi,\gamma})^{-1} \mathbf{P}_{\pi,\gamma}(1-\lambda_c) \tag{2}$$

$$\mathbf{r}_\pi^\lambda := (\mathbf{I} - \lambda_c \mathbf{P}_{\pi,\gamma})^{-1} \mathbf{r}_\pi$$

To see why this is the definition of the Bellman operator, we define the expected λ-return, $\mathbf{v}_{\pi,\lambda} \in \mathbb{R}^n$ for a given approximate value function, given by a vector $\mathbf{v} \in \mathbb{R}^n$.

$$\begin{aligned}\mathbf{v}_{\pi,\lambda}(s) &:= \mathbf{r}_\pi(s) + \sum_{s' \in \mathcal{S}} \mathbf{P}_{\pi,\gamma}(s,s')\left[(1-\lambda_c)\mathbf{v}(s') + \lambda_c \mathbf{v}_{\pi,\lambda}(s')\right] \\ &= \mathbf{r}_\pi(s) + (1-\lambda_c)\mathbf{P}_{\pi,\gamma}(s,:)\mathbf{v} + \lambda_c \mathbf{P}_{\pi,\gamma}(s,:)\mathbf{v}_{\pi,\lambda}.\end{aligned}$$

Continuing the recursion, we obtain[3]

$$\begin{aligned}\mathbf{v}_{\pi,\lambda} &= \left[\sum_{i=0}^{\infty} (\lambda_c \mathbf{P}_{\pi,\gamma})^i\right](\mathbf{r}_\pi + (1-\lambda_c)\mathbf{P}_{\pi,\gamma}\mathbf{v}) \\ &= (\mathbf{I} - \lambda_c \mathbf{P}_{\pi,\gamma})^{-1}(\mathbf{r}_\pi + (1-\lambda_c)\mathbf{P}_{\pi,\gamma}\mathbf{v}) = \mathrm{T}^{(\lambda)}\mathbf{v}\end{aligned}$$

The fixed point for this formula satisfies $\mathrm{T}^{(\lambda)}\mathbf{v} = \mathbf{v}$ for the Bellman operator defined in Equation (1).

To see how this generalized Bellman operator modifies the algorithms, we consider the extension to temporal difference algorithms. Many algorithms can be easily generalized by replacing γ_c or $\gamma_s(s_{t+1})$ with transition-based $\gamma(s_t, a_t, s_{t+1})$. For example, the TD algorithm is generalized by setting the discount on each step to $\gamma_{t+1} = \gamma(s_t, a_t, s_{t+1})$,

$$\begin{aligned}\mathbf{w}_{t+1} &= \mathbf{w}_t + \alpha_t \delta_t \mathbf{e}_t \qquad \triangleright \text{ for some step-size } \alpha_t \\ \delta_t &:= r_{t+1} + \gamma_{t+1}\mathbf{x}(s_{t+1})^\top \mathbf{w} - \mathbf{x}(s_t)^\top \mathbf{w} \qquad (3) \\ \mathbf{e}_t &= \gamma_t \lambda_c \mathbf{e}_{t-1} + \mathbf{x}(s_t).\end{aligned}$$

Sutton, 2010; Sutton et al., 2014; Yu, 2012).

[3]For a matrix $\mathbf{M}$ with maximum eigenvalue less than 1, $\sum_{i=0}^{\infty} \mathbf{M}^i = (\mathbf{I} - \mathbf{M})^{-1}$. We show in Lemma 3 that $\mathbf{P}_{\pi,\gamma}$ satisfies this condition, implying $\lambda_c \mathbf{P}_{\pi,\gamma}$ satisfies this condition and so this infinite sum is well-defined.

The generalized TD fixed-point, under linear function approximation, can be expressed as a linear system $\mathbf{A}\mathbf{w} = \mathbf{b}$

$$\mathbf{A} = \mathbf{X}^\top \mathbf{D}(\mathbf{I} - \lambda_c \mathbf{P}_{\pi,\gamma})^{-1}(\mathbf{I} - \mathbf{P}_{\pi,\gamma})\mathbf{X}$$
$$\mathbf{b} = \mathbf{X}^\top \mathbf{D}(\mathbf{I} - \lambda_c \mathbf{P}_{\pi,\gamma})^{-1}\mathbf{r}_\pi$$

where each row in $\mathbf{X} \in \mathbb{R}^{n\times d}$ corresponds to features for a state, and $\mathbf{D} \in \mathbb{R}^{n\times n}$ is a diagonal weighting matrix. Typically, $\mathbf{D} = \mathrm{diag}(\mathbf{d}_\mu)$, where $\mathbf{d}_\mu \in \mathbb{R}^n$ is the stationary distribution for the behavior policy $\mu : \mathcal{S} \times \mathcal{A} \to [0, 1]$ generating the stream of interaction. In on-policy learning, $\mathbf{d}_\mu = \mathbf{d}_\pi$. With the addition of the interest function, this weighting changes to $\mathbf{D} = \mathrm{diag}(\mathbf{d}_\mu \circ \mathbf{i})$, where $\circ$ denotes element-wise product (Hadamard product). More recently, a new algorithm, emphatic TD (ETD) (Mahmood et al., 2015; Sutton et al., 2016) specified yet another weighting, $\mathbf{D} = \mathbf{M}$ where $\mathbf{M} = \mathrm{diag}(\mathbf{m})$ with $\mathbf{m} = (\mathbf{I} - \mathbf{P}^\lambda_\pi)^{-1}(\mathbf{d}_\mu \circ \mathbf{i})$. Importantly, even for off-policy sampling, with this weighting, $\mathbf{A}$ is guaranteed to be positive definite. We show in the next section that the generalized Bellman operator for both the on-policy and emphasis weighting is a contraction, and further in the appendix that the emphasis weighting with a transition-based trace function is also a contraction.

5. Generalized theoretical properties

In this section, we provide a general approach to incorporating transition-based discounting into approximation bounds. Most previous bounds have assumed a constant discount. For example, ETD was introduced with state-based γ_s; however, (Hallak et al., 2015) analyzed approximation error bounds of ETD using a constant discount γ_c. By using matrix norms on $\mathbf{P}_{\pi,\gamma}$, we generalize previous approximation bounds to both the episodic and continuing case.

Define the set of bounded vectors for the general space of value functions $\mathcal{V} = \{\mathbf{v} \in \mathbb{R}^n : \|\mathbf{v}\|_{D_\mu} < \infty\}$. Let $\mathcal{F}_v \subset \mathcal{V}$ be a subspace of possible solutions, e.g., $\mathcal{F}_v = \{\mathbf{X}\mathbf{w} | \mathbf{w} \in \mathbb{R}^d, \|\mathbf{w}\|_2 < \infty\}$.

A1. The action space $\mathcal{A}$ and state space $\mathcal{S}$ are finite.

A2. For polices $\mu, \pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$, there exist unique invariant distributions $\mathbf{d}_\mu, \mathbf{d}_\pi$ such that $\mathbf{d}_\pi \mathbf{P}_\pi = \mathbf{d}_\pi$ and $\mathbf{d}_\mu \mathbf{P}_\mu = \mathbf{d}_\mu$. This assumption is typically satisfied by assuming an ergodic Markov chain for the policy.

A3. There exist transition s, a, s' such that $\gamma(s, a, s') < 1$ and $\pi(s, a)P(s, a, s') > 0$. This assumptions states that the policy reaches some part of the space where the discount is less than 1.

A4. Assume for any $\mathbf{v} \in \mathcal{F}_v$, if $\mathbf{v}(s) = 0$ for all $s \in \mathcal{S}$ where $\mathbf{i}(s) > 0$, then $\mathbf{v}(s) = 0$ for all $s \in \mathcal{S}$ s.t. $\mathbf{i}(s) = 0$. For linear function approximation, this requires $\mathcal{F} = \mathrm{span}\{\mathbf{x}(s) : s \in \mathcal{S}, \mathbf{i}(s) \neq 0\}$.

For weighted norm $\|\mathbf{v}\|_\mathbf{D} = \sqrt{\mathbf{v}^\top \mathbf{D}\mathbf{v}}$, if we can take the square root of $\mathbf{D}$, the induced matrix norm is $\|\mathbf{P}^\lambda_\pi\|_\mathbf{D} = \left\|\mathbf{D}^{1/2}\mathbf{P}^\lambda_\pi \mathbf{D}^{1/2}\right\|_{sp}$, where the spectral norm $\|\cdot\|_{sp}$ is the largest singular value of the matrix. For simplicity of notation below, define $s_\mathbf{D} := \|\mathbf{P}^\lambda_\pi\|_\mathbf{D}$. For any diagonalizable, nonnegative matrix $\mathbf{D}$, the projection $\Pi_\mathbf{D} : \mathcal{V} \to \mathcal{F}_v$ onto $\mathcal{F}_v$ exists and is defined $\Pi_\mathbf{D}\mathbf{z} = \mathrm{argmin}_{\mathbf{v}\in\mathcal{F}_v} \|\mathbf{z} - \mathbf{v}\|_\mathbf{D}$.

5.1. Approximation bound

We first prove that the generalized Bellman operator in Equation 1 is a contraction. We extend the bound from (Tsitsiklis and Van Roy, 1997; Hallak et al., 2015) for constant discount and constant trace parameter, to the general transition-based setting. The normed difference to the true value function could be defined by multiple weightings. A well-known result is that for $\mathbf{D} = \mathbf{D}_\pi$ the Bellman operator is a contraction for constant γ_c and λ_c (Tsitsiklis and Van Roy, 1997); recently, this has been generalized for a variant of ETD to $\mathbf{M}$, still with constant parameters (Hallak et al., 2015). We extend this result for transition-based γ for both $\mathbf{D}_\pi$ and the transition-based emphasis matrix $\mathbf{M}$.

Lemma 1. *For $\mathbf{D} = \mathbf{D}_\pi$ or $\mathbf{D} = \mathbf{M}$,*

$$s_\mathbf{D} = \|\mathbf{P}^\lambda_\pi\|_\mathbf{D} < 1.$$

Proof: For $\mathbf{D} = \mathbf{M}$: let $\boldsymbol{\xi} \in \mathbb{R}^n$ be the vector of row sums for $\mathbf{P}^\lambda_\pi$: $\mathbf{P}^\lambda_\pi \mathbf{1} = \boldsymbol{\xi}$. Then for any $\mathbf{v} \in \mathcal{V}$, with $\mathbf{v} \neq \mathbf{0}$,

$$\begin{aligned}
\|\mathbf{P}^\lambda_\pi \mathbf{v}\|^2_\mathbf{M} &= \sum_{s\in\mathcal{S}} \mathbf{m}(s) \left(\sum_{s'\in\mathcal{S}} \mathbf{P}^\lambda_\pi(s, s')\mathbf{v}(s')\right)^2 \\
&= \sum_{s\in\mathcal{S}} \mathbf{m}(s)\boldsymbol{\xi}(s)^2 \left(\sum_{s'\in\mathcal{S}} \frac{\mathbf{P}^\lambda_\pi(s, s')}{\boldsymbol{\xi}(s)}\mathbf{v}(s')\right)^2 \\
&\le \sum_{s\in\mathcal{S}} \mathbf{m}(s)\boldsymbol{\xi}(s)^2 \sum_{s'\in\mathcal{S}} \frac{\mathbf{P}^\lambda_\pi(s, s')}{\boldsymbol{\xi}(s)}\mathbf{v}(s')^2 \\
&= \sum_{s'\in\mathcal{S}} \mathbf{v}(s')^2 \sum_{s\in\mathcal{S}} \mathbf{m}(s)\boldsymbol{\xi}(s)\mathbf{P}^\lambda_\pi(s, s') \\
&= \mathbf{v}^\top \mathrm{diag}\left((\mathbf{m} \circ \boldsymbol{\xi})^\top \mathbf{P}^\lambda_\pi\right)\mathbf{v}
\end{aligned}$$

where the first inequality follows from Jensen's inequality, because $\mathbf{P}^\lambda_\pi(s, :)$ is normalized. Now because $\boldsymbol{\xi}$ has entries that are less than 1, because the row sums of $\mathbf{P}^\lambda_\pi$ are less than 1 as shown in Lemma 4, and because each of the values in the above product are nonnegative,

$$\begin{aligned}
&\mathbf{v}^\top \mathrm{diag}\left((\mathbf{m} \circ \boldsymbol{\xi})^\top \mathbf{P}^\lambda_\pi\right)\mathbf{v} \\
&\quad\le \mathbf{v}^\top \mathrm{diag}\left(\mathbf{m}^\top \mathbf{P}^\lambda_\pi\right)\mathbf{v} \\
&\quad= \mathbf{v}^\top \mathrm{diag}\left(\mathbf{m}^\top(\mathbf{P}^\lambda_\pi - \mathbf{I}) + \mathbf{m}^\top\right)\mathbf{v} \\
&\quad= \mathbf{v}^\top \mathrm{diag}\left(-(\mathbf{d}_\pi \circ \mathbf{i})^\top + \mathbf{m}^\top\right)\mathbf{v} \\
&\quad= \mathbf{v}^\top \mathrm{diag}\left(\mathbf{m}^\top\right)\mathbf{v} - \mathbf{v}^\top \mathrm{diag}\left((\mathbf{d}_\pi \circ \mathbf{i})^\top\right)\mathbf{v} \\
&\quad< \|\mathbf{v}\|^2_\mathbf{M}
\end{aligned}$$

The last inequality is a strict inequality because $\mathbf{d}_\pi \circ \mathbf{i}$ has at least one positive entry where $\mathbf{v}$ has a positive entry. Otherwise, if $\mathbf{v}(s) = 0$ everywhere with $\mathbf{i}(s) > 0$, then $\mathbf{v} = \mathbf{0}$, which we assumed was not the case.

Therefore, $\|\mathbf{P}_\pi^\lambda \mathbf{v}\|_\mathbf{M} < \|\mathbf{v}\|_\mathbf{M}$ for any $\mathbf{v} \neq \mathbf{0}$, giving $\|\mathbf{P}_\pi^\lambda\|_\mathbf{M} := \max_{\mathbf{v}\in\mathbb{R}^n, \mathbf{v}\neq 0} \frac{\|\mathbf{P}_\pi^\lambda \mathbf{v}\|_\mathbf{M}}{\|\mathbf{v}\|_\mathbf{M}} < 1$. This exact same proof follows through verbatim for the generalization of $\mathbf{P}_\pi^\lambda$ to transition-based trace λ.

For $\mathbf{D} = \mathbf{D}_\pi$: Again, we use Jensen's inequality, but now rely on the property $\mathbf{d}_\pi \mathbf{P}_\pi = \mathbf{d}_\pi$. Because of Assumption A3, for some $s < 1$, for any non-negative $\mathbf{v}_+$,

$$\begin{aligned}
\mathbf{d}_\pi \mathbf{P}_{\pi,\gamma}\mathbf{v}_+ &= \sum_s \sum_a \mathbf{d}_\pi(s)\mathrm{Pr}(s,a,:) \circ \gamma(s,a,:)\mathbf{v}_+ \\
&\leq s \sum_s \sum_a \mathbf{d}_\pi(s)\mathrm{Pr}(s,a,:)\mathbf{v}_+ = s\mathbf{d}_\pi \mathbf{v}.
\end{aligned}$$

Therefore, because vectors $\mathbf{P}_{\pi,\gamma}\mathbf{v}_+$ are also non-negative,

$$\begin{aligned}
\mathbf{d}_\pi \mathbf{P}_\pi^\lambda \mathbf{v}_+ &= \mathbf{d}_\pi \left(\sum_{k=0}^\infty (\mathbf{P}_{\pi,\gamma}\lambda_c)^k \mathbf{P}_{\pi,\gamma}(1-\lambda_c) \right) \mathbf{v}_+ \\
&\leq (1-\lambda_c) \sum_{k=0}^\infty (s\lambda_c)^k \mathbf{d}_\pi \mathbf{P}_{\pi,\gamma}\mathbf{v}_+ \\
&\leq (1-\lambda_c)(1-s\lambda_c)^{-1} s\mathbf{d}_\pi \mathbf{v}_+
\end{aligned}$$

and so

$$\begin{aligned}
\|\mathbf{P}_\pi^\lambda \mathbf{v}\|_{\mathbf{D}_\pi}^2 &\leq \sum_{s\in\mathcal{S}} \mathbf{d}_\pi(s)\boldsymbol{\xi}(s)^2 \sum_{s'\in\mathcal{S}} \frac{\mathbf{P}_\pi^\lambda(s,s')}{\boldsymbol{\xi}(s)} \mathbf{v}(s')^2 \\
&= \sum_{s'\in\mathcal{S}} \mathbf{v}(s')^2 \sum_{s\in\mathcal{S}} \mathbf{d}_\pi(s)\boldsymbol{\xi}(s)\mathbf{P}_\pi^\lambda(s,s') \\
&\leq \sum_{s'\in\mathcal{S}} \mathbf{v}(s')^2 \sum_{s\in\mathcal{S}} \mathbf{d}_\pi(s)\mathbf{P}_\pi^\lambda(s,s') \\
&\leq \tfrac{s(1-\lambda_c)}{1-\lambda_c s} \sum_{s'\in\mathcal{S}} \mathbf{d}(s')\mathbf{v}(s')^2 \\
&\leq \tfrac{s - s\lambda_c}{1-\lambda_c s} \|\mathbf{v}\|_{\mathbf{D}_\pi}^2
\end{aligned}$$

where $\frac{s-s\lambda_c}{1-\lambda_c s} < 1$ since $s < 1$. ■

Lemma 2. *Under assumptions A1-A3, the Bellman operator* $\mathrm{T}^{(\lambda)}$ *in Equation* (1) *is a contraction under a norm weighted by* $\mathbf{D} = \mathbf{D}_\pi$ *or* $\mathbf{D} = \mathbf{M}_\pi$*, i.e., for* $\mathbf{v} \in \mathcal{V}$

$$\|\mathrm{T}^{(\lambda)}\mathbf{v}\|_\mathbf{D} < \|\mathbf{v}\|_\mathbf{D}.$$

Further, because the projection $\Pi_\mathbf{D}$ *is a contraction,* $\Pi_\mathbf{D}\mathrm{T}^{(\lambda)}$ *is also a contraction and has a unique fixed point* $\Pi_\mathbf{D}\mathrm{T}^{(\lambda)}\mathbf{v} = \mathbf{v}$ *for* $\mathbf{v} \in \mathcal{F}_v$.

Proof: Because any vector $\mathbf{v}$ can be written $\mathbf{v} = \mathbf{v}_1 - \mathbf{v}_2$,

$$\begin{aligned}
\|\mathrm{T}^{(\lambda)}\mathbf{v}\|_\mathbf{D} = \|\mathrm{T}^{(\lambda)}(\mathbf{v}_1 - \mathbf{v}_2)\|_\mathbf{D} &= \|\mathbf{P}_\pi^\lambda(\mathbf{v}_1 - \mathbf{v}_2)\|_\mathbf{D} \\
&\leq \|\mathbf{P}_\pi^\lambda\|_\mathbf{D}\|\mathbf{v}\|_\mathbf{D} \\
&< \|\mathbf{v}\|_\mathbf{D}
\end{aligned}$$

where the last inequality follows from Lemma 1. By the Banach Fixed Point theorem, because the Bellman operator is a contraction under $\mathbf{D}$, it has a unique fixed point. ■

Theorem 1. *If* $\mathbf{D}$ *satisfies* $s_\mathbf{D} < 1$*, then there exists* $\mathbf{v} \in \mathcal{F}_v$ *such that* $\Pi_\mathbf{D}\mathrm{T}^{(\lambda)}\mathbf{v} = \mathbf{v}$*, and the error to the true value function is bounded as*

$$\|\mathbf{v} - \mathbf{v}^*\|_\mathbf{D} \leq (1 - s_\mathbf{D})^{-1}\|\Pi_\mathbf{D}\mathbf{v}^* - \mathbf{v}^*\|_\mathbf{D}. \tag{4}$$

For constant discount $\gamma_c \in [0,1)$ *and constant trace parameter* $\lambda_c \in [0,1]$*, this bound reduces to the original bound (Tsitsiklis and Van Roy, 1997, Lemma 6):*

$$(1 - s_\mathbf{D})^{-1} \leq \frac{1-\gamma_c\lambda_c}{1-\gamma_c}.$$

Proof: Let $\mathbf{v}$ be the unique fixed point of $\Pi_\mathbf{D}\mathrm{T}^{(\lambda)}$, which exists by Lemma 2.

$$\begin{aligned}
\|\mathbf{v} - \mathbf{v}^*\|_\mathbf{D} &\leq \|\mathbf{v} - \Pi_\mathbf{D}\mathbf{v}^*\|_\mathbf{D} + \|\Pi_\mathbf{D}\mathbf{v}^* - \mathbf{v}^*\|_\mathbf{D} \\
&= \|\Pi\mathrm{T}^{(\lambda)}\mathbf{v} - \Pi_\mathbf{D}\mathbf{v}^*\|_\mathbf{D} + \|\Pi_\mathbf{D}\mathbf{v}^* - \mathbf{v}^*\|_\mathbf{D} \\
&\leq \|\mathrm{T}^{(\lambda)}\mathbf{v} - \mathbf{v}^*\|_\mathbf{D} + \|\Pi_\mathbf{D}\mathbf{v}^* - \mathbf{v}^*\|_\mathbf{D} \\
&= \|\mathrm{T}^{(\lambda)}(\mathbf{v} - \mathbf{v}^*)\|_\mathbf{D} + \|\Pi_\mathbf{D}\mathbf{v}^* - \mathbf{v}^*\|_\mathbf{D} \\
&= \|\mathbf{P}_\pi^\lambda(\mathbf{v} - \mathbf{v}^*)\|_\mathbf{D} + \|\Pi_\mathbf{D}\mathbf{v}^* - \mathbf{v}^*\|_\mathbf{D} \\
&= \|\mathbf{P}_\pi^\lambda\|_\mathbf{D}\|\mathbf{v} - \mathbf{v}^*\|_\mathbf{D} + \|\Pi_\mathbf{D}\mathbf{v}^* - \mathbf{v}^*\|_\mathbf{D} \\
&= s_\mathbf{D}\|\mathbf{v} - \mathbf{v}^*\|_\mathbf{D} + \|\Pi_\mathbf{D}\mathbf{v}^* - \mathbf{v}^*\|_\mathbf{D}
\end{aligned}$$

where the second inequality is due to $\|\Pi\mathrm{T}^{(\lambda)}\mathbf{v}\|_\mathbf{D} \leq \|\mathrm{T}^{(\lambda)}\mathbf{v}\|_\mathbf{D}$, the second equality due to $\mathrm{T}^{(\lambda)}\mathbf{v}^* = \mathbf{v}^*$ and the third equality due to $\mathrm{T}^{(\lambda)}\mathbf{v} - \mathrm{T}^{(\lambda)}\mathbf{v}^* = \mathbf{P}_\pi^\lambda(\mathbf{v} - \mathbf{v}^*)$ because the $\mathbf{r}_\pi$ cancels. By rearranging terms, we get

$$(1 - s_\mathbf{D})\|\mathbf{v} - \mathbf{v}^*\|_{\mathbf{D}_\pi} \leq \|\Pi\mathbf{v}^* - \mathbf{v}^*\|_{\mathbf{D}_\pi}$$

and since $s_\mathbf{D} < 1$, we get the final result.

For constant $\gamma_c < 1$ and λ_c, because $\mathbf{P}_{\pi,\gamma} = \gamma\mathbf{P}_\pi$

$$\begin{aligned}
s_\mathbf{D} &= \|\mathbf{P}_\pi^\lambda\|_\mathbf{D} \\
&= \|\mathbf{D}^{1/2}\left(\sum_{i=0}^\infty \gamma_c^i\lambda_c^i\mathbf{P}_\pi^i\right)\gamma_c(1-\lambda_c)\mathbf{P}_\pi\mathbf{D}^{1/2}\|_2 \\
&\leq \gamma_c(1-\lambda_c)\sum_{i=0}^\infty \gamma_c^i\lambda_c^i\|\mathbf{D}^{1/2}\mathbf{P}_\pi^{i+1}\mathbf{D}^{1/2}\|_2 \\
&= \gamma_c(1-\lambda_c)\sum_{i=0}^\infty \gamma_c^i\lambda_c^i\|\mathbf{P}_\pi^{i+1}\|_\mathbf{D} \\
&\leq \gamma_c(1-\lambda_c)\sum_{i=0}^\infty \gamma_c^i\lambda_c^i \\
&= \frac{\gamma_c(1-\lambda_c)}{1-\gamma_c\lambda_c}
\end{aligned}$$

■

We provide generalizations to transition-based trace parameters in the appendix, for the emphasis weighting, and also discuss issues with generalizing to state-based termination for a standard weighting with $\mathbf{d}_\pi$. We show that for any transition-based discounting function $\lambda : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \rightarrow [0, 1]$, the above contraction results hold under emphasis weighting. We then provide a general form for an upper bound on $\|\mathbf{P}_\pi^\lambda\|_{\mathbf{D}_\pi}$ for transition-based discounting, based on the contraction properties of two matrices within $\mathbf{P}_\pi^\lambda$. We further provide an example where the Bellman operator is not a contraction even under the simpler generalization to state-based discounting, and discuss the requirements for the transition-based generalizations to ensure a contraction with weighting $\mathbf{d}_\pi$. This further motivates the emphasis weighting as a more flexible scheme for convergence under general setting—both off-policy and transition-based generalization.

5.2. Properties of TD algorithms

Using this characterization of $\mathbf{P}_\pi^\lambda$, we can re-examine previous results for temporal difference algorithms that either used state-based or constant discounts.

Convergence of Emphatic TD for RL tasks. We can extend previous convergence results for ETD, for learning value functions and action-value functions, for the RL task formalism. For policy evaluation, ETD and ELSTD, the least-squares version of ETD that uses the above defined $\mathbf{A}$ and $\mathbf{b}$ with $\mathbf{D} = \mathbf{M}$, have both been shown to converge with probability one (Yu, 2015). As an important component of this proof is convergence in expectation, which relies on $\mathbf{A}$ being positive definite. In particular, for appropriate step-sizes α_t (see (Yu, 2015)), if $\mathbf{A}$ is positive definite, the iterative update is convergent $\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha_t(\mathbf{b} - \mathbf{A}\mathbf{w}_t)$. For the generalization to transition-based discounting, convergence in expectation extends for the emphatic algorithms. We provide these details in the appendix for completeness, with theorem statement and proof in Appendix F and pseudocode in Appendix D.

Convergence rate of LSTD(λ). Tagorti and Scherrer (2015) recently provided convergence rates for LSTD(λ) for continuing tasks, for some $\gamma_c < 1$. These results can be extended to the episodic setting with the generic treatment of $\mathbf{P}_\pi^\lambda$. For example, in (Tagorti and Scherrer, 2015, Lemma 1), which describes the sensitivity of LSTD, the proof extends by replacing the matrix $(1-\lambda_c)\gamma_c\mathbf{P}_\pi(\mathbf{I}-\lambda_c\gamma_c\mathbf{P}_\pi)^{-1}$ (which they call M in their proof) with the generalization $\mathbf{P}_\pi^\lambda$, resulting instead in the constant $\frac{1}{1-s_\mathbf{D}}$ in the bound rather than $\frac{1-\lambda_c\gamma_c}{1-\gamma_c}$. Further, this generalizes convergence rate results to emphatic LSTD, since $\mathbf{M}$ satisfies the required convergence properties, with rates dictated by $s_\mathbf{M}$ rather than $s_{\boldsymbol{D}_\mu}$ for standard LSTD.

Insights into $s_\mathbf{D}$. Though the generalized form enables unified episodic and continuing results, the resulting bound parameter $s_\mathbf{D}$ is more difficult to interpret than for constant γ_c, λ_c. With λ_c increasing to one, the constant $\frac{1-\gamma_c\lambda_c}{1-\gamma_c}$ in the upper bound decreased to one. For γ_c decreasing to zero, the bound also decreases to one. These trends are intuitive, as the problem should be simpler when γ_c is small, and bias should be less when λ_c is close to one. More generally, however, the discount can be small or large for different transitions, making it more difficult to intuit the trend.

To gain some intuition for $s_\mathbf{D}$, consider a random policy in the taxi domain, with $s_\mathbf{D}$ summarized in Table 1. As λ_c goes to one, $s_\mathbf{D}$ goes to zero and so $(1 - s_\mathbf{D})^{-1}$ goes to one. Some outcomes of note are that 1) hard or soft termination for the pickup results in the exact same $s_\mathbf{D}$; 2) for a constant gamma of $\gamma_c = 0.99$, the episodic discount had a slightly smaller $s_\mathbf{D}$; and 3) increasing λ_c has a much stronger effect than including more terminations. Whereas, when we added random terminations, so that from 1% and 10% of the states, termination occurred on at least one path within 5 steps or even more aggressively on every path within 5 steps, the values of $s_\mathbf{D}$ were similar.

λ_c	0.0	0.5	0.9	0.99	0.999
Episodic taxi	0.989	0.979	0.903	0.483	0.086
$\gamma_c = 0.99$	0.990	0.980	0.908	0.497	0.090
1% single path	0.989	0.978	0.898	0.467	0.086
10% single path	0.987	0.975	0.887	0.439	0.086
1% all paths	0.978	0.956	0.813	0.304	0.042
10% all paths	0.898	0.815	0.468	0.081	0.009

Table 1: The $s_\mathbf{D}$ values for increasing λ_c, with discount settings described in the text.

6. Discussion and conclusion

The goal of this paper is to provide intuition and examples of how to use the *RL task formalism*. Consequently, to avoid jarring the explanation, technical contributions were not emphasized, and in some cases included only in the appendix. For this reason, we would like to highlight and summarize the technical contributions, which include 1) the introduction of the RL task formalism, and of transition-based discounts; 2) an explicit characterization of the relationship between state-based and transition-based discounting; and 3) generalized approximation bounds, applying to both episodic and continuing tasks; and 4) insights into—and issues with—extending contraction results for both state-based and transition-based discounting. Through intuition from simple examples and fundamental theoretical extensions, this work provides a relatively complete characterization of the RL task formalism, as a foundation for use in practice and theory.

Acknowledgements

Thanks to Hado van Hasselt for helpful discussions about transition-based discounting, and probabilistic discounts.

References

Dimitri P Bertsekas and John N Tsitsiklis. *Neuro-dynamic programming*. Athena Scientific Press, 1996.

Thomas G Dietterich. Hierarchical Reinforcement Learning with the MAXQ Value Function Decomposition. *Journal of Artificial Intelligence Research*, 2000.

Carlos Diuk, Andre Cohen, and Michael L Littman. An object-oriented representation for efficient reinforcement learning. In *International Conference on Machine Learning*, 2008.

Assaf Hallak, Aviv Tamar, Rémi Munos, and Shie Mannor. Generalized Emphatic Temporal Difference Learning: Bias-Variance Analysis. *CoRR abs/1509.05172*, 2015.

Nan Jiang, Alex Kulesza, Satinder P Singh, and Richard L Lewis. The Dependence of Effective Planning Horizon on Model Accuracy. *International Conference on Autonomous Agents and Multiagent Systems*, 2015.

Michael J Kearns and Satinder P Singh. Bias-Variance error bounds for temporal difference updates. In *Annual Conference on Learning Theory*, 2000.

Ashique Rupam Mahmood, Huizhen Yu, Martha White, and Richard S Sutton. Emphatic temporal-difference learning. In *European Workshop on Reinforcement Learning*, 2015.

Joseph Modayil, Adam White, and Richard S Sutton. Multi-timescale nexting in a reinforcement learning robot. *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems*, 2014.

Hamid Maei Reza and Richard S Sutton. GQ (λ): A general gradient algorithm for temporal-difference prediction learning with eligibility traces. In *AGI*, 2010.

Richard S Sutton. TD models: Modeling the world at a mixture of time scales. In *International Conference on Machine Learning*, 1995.

Richard S Sutton and A G Barto. *Introduction to reinforcement learning*. MIT Press, 1998a.

Richard S Sutton and A G Barto. *Reinforcement Learning: An Introduction*. MIT press, 1998b.

Richard S Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 1999.

Richard S Sutton, Hamid Maei Reza, Doina Precup, and Shalab Bhatnagar. Fast gradient-descent methods for temporal-difference learning with linear function approximation. *International Conference on Machine Learning*, 2009.

Richard S Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M Pilarski, Adam White, and Doina Precup. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *International Conference on Autonomous Agents and Multiagent Systems*, 2011.

Richard S Sutton, Ashique Rupam Mahmood, Doina Precup, and Hado van Hasselt. A new Q(lambda) with interim forward view and Monte Carlo equivalence. In *International Conference on Machine Learning*, 2014.

Richard S Sutton, Ashique Rupam Mahmood, and Martha White. An emphatic approach to the problem of off-policy temporal-difference learning. *The Journal of Machine Learning Research*, 2016.

Manel Tagorti and Bruno Scherrer. On the Rate of Convergence and Error Bounds for LSTD(λ). In *International Conference on Machine Learning*, 2015.

Johnathan N Tsitsiklis and Benjamin Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 1997.

Hado Philip van Hasselt. *Insights in Reinforcement Learning*. PhD thesis, Hado van Hasselt, 2011.

Harm van Seijen and Rich Sutton. True online TD(lambda). In *International Conference on Machine Learning*, 2014.

Adam White. *Developing a predictive approach to knowledge*. PhD thesis, University of Alberta, 2015.

Huizhen Yu. Least Squares Temporal Difference Methods: An Analysis under General Conditions. *SIAM Journal on Control and Optimization*, 2012.

Huizhen Yu. On convergence of emphatic temporal-difference learning. In *Annual Conference on Learning Theory*, 2015.

A. More general formulation with probabilistic discounts

In the introduction of transition-based discounting, we could have instead assumed that we had a more general probability model: $\Pr(r, \gamma | s, a, s')$. Now, both the reward and discount are not just functions of states and action, but also are stochastic. This generalization in fact, does not much alter the treatment in this paper. This is because, when taking the expectations for value function, the Bellman operator and the $\mathbf{A}$ matrix, we are left again with $\gamma(s, a, s')$. To see why,

$$\begin{aligned}\mathbf{v}_\pi(s) &= \sum_{a,s'} \pi(s,a)\Pr(s,a,s')\mathrm{E}[r + \gamma \mathbf{v}_\pi(s') | s, a, s'] \\ &= \sum_{a,s'} \pi(s,a)\Pr(s,a,s')\mathrm{E}[r | s, a, s'] \\ &\quad + \sum_{a,s'} \pi(s,a)\Pr(s,a,s')\mathrm{E}[\gamma | s, a, s']\mathbf{v}_\pi(s') \\ &= \mathbf{r}_\pi(s) + \sum_{s'} \mathbf{P}_{\pi,\gamma}(s,s')\mathbf{v}_\pi(s')\end{aligned}$$

for $\gamma(s, a, s') = \mathrm{E}[\gamma | s, a, s']$.

B. Relationship between state-based and transition-based discounting

In this section, we show that for any MDP with transition-based discounting, we can construct an equivalent MDP with state-based discounting. The MDPs are equivalent in the sense that learned policies and value functions learned in either MDP would have equal values when evaluated on the states in the original transition-based MDP. This equality ignores practicality of learning in the larger induced state-based MDP, and at the end of this section, we discuss advantages of the more compact transition-based MDP.

B.1. Equivalence result

The equivalence is obtained by introducing hypothetical states for each transition. The key is then to prove that the stationary distribution for the state-based MDP, with additional hypothetical states, provides the same solution even with function approximation. For each triplet s, a, s', add a new hypothetical state $f_{sas'}$, with set $\mathcal{F}$ comprised of these additional states. Each transition now goes through a hypothetical state, $f_{sas'}$, and allows the discount in the hypothetical state to be set to $\gamma(s, a, s')$. The induced state-based MDP has state set $\bar{\mathcal{S}} = \mathcal{S} \cup \mathcal{F}$ with $|\bar{\mathcal{S}}| = |\mathcal{A}|n^2 + n$. We define the other models in the proof in Appendix B.3.

The choice of action in the hypothetical states is irrelevant. To extend the policy π, we arbitrarily choose that the policy uniformly selects actions when in the hypothetical states and define $\bar{\pi}(s, a) = \pi(s, a)$ for $s \in \mathcal{S}$ and $\bar{\pi}(s, a) = 1/|\mathcal{A}|$ otherwise. For linear function approximation, we also need to assume $\mathbf{x}(f_{sas'}) = \mathbf{x}(s')$ for $f_{sas'} \in \mathcal{F}$.

Theorem 2. *For a given transition-based MDP $(\Pr, r, \mathcal{S}, \mathcal{A}, \gamma)$ and policy π, assume that the stationary distribution $\mathbf{d}_\pi$ exists. Define state-based MDP $(\bar{Pr}, \bar{r}, \bar{\mathcal{S}}, \mathcal{A}, \bar{\gamma}_s)$ with extended $\bar{\pi}$, all as above. Then the stationary distribution $\bar{\mathbf{d}}_\pi$ for $\bar{\pi}$ exists and satisfies*

$$\frac{\bar{\mathbf{d}}_\pi(s)}{\sum_{i \in \mathcal{S}} \bar{\mathbf{d}}_\pi(i)} = \mathbf{d}_\pi(s). \tag{5}$$

$\forall s \in \mathcal{S}$, $\bar{\mathbf{v}}_\pi(s) = \mathbf{v}_\pi(s)$ and $\bar{\pi}(s, a) = \pi(s, a)$ for all $s \in \mathcal{S}, a \in \mathcal{A}$ with $\pi = \operatorname{argmin}_\pi \sum_{s \in \mathcal{S}} \mathbf{d}_\pi(s)\mathbf{v}_\pi(s)$; $\bar{\pi} = \operatorname{argmin}_\pi \sum_{i \in \bar{\mathcal{S}}} \bar{\mathbf{d}}_\pi(s)\bar{\mathbf{v}}_\pi(s)$

B.2. Advantages of transition-based discounting over state-based discounting

Though the two have equal representational abilities, there are several disadvantages of state-based discounting that compound to make the more general transition-based discount strictly more desirable. The disadvantages of using an induced state-based MDP, rather than the original transition-based MDP, arises from the addition of states and include the following.

Compactness. In the worst-case, for a transition-based MDP with n states, the induced state-based MDP can have $|\mathcal{A}|n^2 + n$ states.

Problem definition changes for different discounts. For the same underlying MDP, multiple learners with different discount functions would have different induced state-based MDPs. This complicates code and reduces opportunities for sharing variables and computation.

Overhead. Additional states must be stored, with additional algorithmic updates in those non-states, or cases to avoid these updates, and the need to carefully set features for hypothetical states. This overhead is both computational as well as conceptual, as it complicates the code.

Stationary distribution. This distribution superfluously includes hypothetical states and requires renormalization to obtain the stationary distribution for the original transition-based MDP.

Off-policy learning. In off-policy learning, one goal is to learn many value functions with different discounts (White, 2015). As mentioned above, these learners may have different induced state-based MDPs, which complicates implementation and even theory. To theoretically characterize a set of off-policy learners, it would be necessary to consider different induced state-based MDPs. Further, sharing information, such as the features, is again complicated by using induced state-based MDP rather than a single transition-based MDP, with varying discount functions.

Specification of algorithms. Often algorithms are introduced either for the episodic case (e.g., true-online TD (van

Seijen and Sutton, 2014)) or the continuing case (e.g., the lower-variance version of ETD (Hallak et al., 2015)). When kept separately, with explicit loops over episodes, the algorithm itself is different (e.g., Sarsa (Sutton and Barto, 1998a, Figure 8.8)); or, if a state-based approach is used, fake states and fake transitions would have to be explicitly added to make the update the same for continuing or episodic. For the generalized formulation, the only difference is the γ_{t+1} that is passed to the algorithm; the algorithm itself remains exactly the same in the two settings. As a minor example, for episodic problems, there is typically an explicit (error-prone) step to clear traces; with generalized discounting, the traces are automatically cleared at the end of an episode by γ_{t+1}.

Experimental design. When presenting results for episodic and continuing problem, often the former uses number of episodes and the later number of steps. In reality both simply consist of a trajectory of information, with the former having $\gamma_{t+1} = 0$ for some steps. A unified view with number of steps enables more consistent presentation of results across domains. Related to this difference, a common but rarely discussed decision when implementing episodic tasks is the cut-off for the maximum number of steps in an episode. If set too small, an algorithm that takes longer to reach the goal in the first few episodes, but then learns more quickly afterwards, could be unfairly penalized. Instead learning could be limited to some maximum number of steps to constrain learning time similarly for both continuing and episodic problems, where multiple episodes could occur within that maximum number of steps.

B.3. Proof of Theorem 2

This prove illustrates the representability relationship between transition-based discounting and state-based discounting. This equivalence could be obtained more compactly if $\gamma(s, a, s')$ is not different for every s'; however, the proof becomes much more involved. Since our main goal is to simply show a representability result, we opt for interpretability. Note that, in addition, the result in Theorem 2 fills a gap in the previous theory, which indicated that state-based discounting could be used to represent episodic problems, but did not explicitly demonstrate that the stationary distribution would be equivalent (see (Yu, 2015)).

Define transition probabilities $\bar{\Pr} : \bar{\mathcal{S}} \times \bar{\mathcal{S}} \to [0, 1]$

$$\bar{\Pr}(i, a, j) = \begin{cases} \Pr(i, a, s') & i \in \mathcal{S}, j = f_{ias'} \\ 1 & j \in \mathcal{S}, i = f_{saj} \\ 0 & \text{otherwise} \end{cases}$$

rewards

$$\bar{r}(i, a, j) = \begin{cases} r(i, a, s') & i \in \mathcal{S}, j = f_{ias'} \\ 0 & \text{otherwise} \end{cases}$$

and state-based discount function $\bar{\gamma}_s : \bar{\mathcal{S}} \to [0, 1]$

$$\bar{\gamma}_s(i) = \begin{cases} \gamma(s, a, s') & i = f_{sas'} \\ 1 & \text{otherwise} \end{cases}$$

Theorem 2 For a given transition-based MDP $(\Pr, r, \mathcal{S}, \mathcal{A}, \gamma)$ and policy π, assume that the stationary distribution $\mathbf{d}_\pi$ exists. Define state-based MDP $(\bar{\Pr}, \bar{r}, \bar{\mathcal{S}}, \mathcal{A}, \bar{\gamma}_s)$ with extended $\bar{\pi}$, all as above. Then the stationary distribution $\bar{\mathbf{d}}_\pi$ for $\bar{\pi}$ exists and satisfies

$$\frac{\bar{\mathbf{d}}_\pi(s)}{\sum_{i \in \mathcal{S}} \bar{\mathbf{d}}_\pi(i)} = \mathbf{d}_\pi(s) \tag{6}$$

and for all $s \in \mathcal{S}$, $\bar{\mathbf{v}}_\pi(s) = \mathbf{v}_\pi(s)$.

Proof: Define matrix $\bar{\mathbf{P}}_\pi \in \mathbb{R}^{(n+|\mathcal{A}|n^2) \times (n+|\mathcal{A}|n^2)}$ where $\bar{\mathbf{P}}_\pi(i, j) = \sum_{a \in \mathcal{A}} \bar{\pi}(i, a) \bar{\Pr}(i, a, j)$, giving

$$\bar{\mathbf{P}}_\pi(i, j) = \begin{cases} \pi(i, a) \Pr(i, a, s') & i \in \mathcal{S}, j = f_{ias'} \\ 1 & i = f_{saj}, a \in \mathcal{A}, j \in \mathcal{S} \\ 0 & \text{otherwise} \end{cases}$$

Define

$$\bar{\mathbf{d}}_\pi(i) := \frac{1}{c} \begin{cases} \mathbf{d}_\pi(i) & i \in \mathcal{S} \\ \mathbf{d}_\pi(s) \pi(s, a) \Pr(s, a, s') & i = f_{sas'} \end{cases}$$

where $c > 0$ is a normalizer to ensure that $\mathbf{1}^\top \bar{\mathbf{d}}_\pi = 1$. Now we need to show that $\bar{\mathbf{d}}_\pi \bar{\mathbf{P}}_\pi = \bar{\mathbf{d}}_\pi$. For any $j \in \mathcal{S}$,

$$\bar{\mathbf{d}}_\pi \bar{\mathbf{P}}_\pi(:, j) = \frac{1}{c} \Big(\sum_{s \in \mathcal{S}} \mathbf{d}_\pi(s) \bar{\mathbf{P}}_\pi(s, j) + \sum_{f \in \mathcal{F}} \bar{\mathbf{d}}_\pi(f) \bar{\mathbf{P}}_\pi(f, j) \Big)$$

Case 1: $j \in \mathcal{S}$
For the first component, because $\bar{\mathbf{P}}_\pi(s, j) = 0$ by definition of $\bar{\Pr}$, we get

$$\sum_{s \in \mathcal{S}} \mathbf{d}_\pi(s) \bar{\mathbf{P}}_\pi(s, j) = 0$$

For the second component, because $\bar{\mathbf{P}}_\pi(f_{saj}, j) = 1$,

$$\begin{aligned} &\sum_{f_{sas'} \in \mathcal{F}} \bar{\mathbf{d}}_\pi(f_{sas'}) \bar{\mathbf{P}}_\pi(f_{sas'}, j) \\ &= \sum_{f_{saj} \in \mathcal{F}} \bar{\mathbf{d}}_\pi(f_{saj}) \bar{\mathbf{P}}_\pi(f_{saj}, j) \\ &= \sum_{f_{saj} \in \mathcal{F}} \bar{\mathbf{d}}_\pi(f_{saj}) \\ &= \sum_{s \in \mathcal{S}} \mathbf{d}_\pi(s) \sum_{a \in \mathcal{A}} \pi(s, a) \Pr(s, a, j) \\ &= \sum_{s \in \mathcal{S}} \mathbf{d}_\pi(s) \mathbf{P}_\pi(s, j) \\ &= \mathbf{d}_\pi(j) \end{aligned}$$

where the last line follows from the definition of the stationary distribution. Therefore, for $j \in \mathcal{S}$

$$\bar{\mathbf{d}}_\pi \bar{\mathbf{P}}_\pi(:, j) = \frac{1}{c}\mathbf{d}_\pi(j) = \bar{\mathbf{d}}_\pi(j)$$

Case 2: $j = f_{sas'} \in \mathcal{F}$
For the first component, because $\bar{\mathbf{P}}_\pi(i, f_{sas'}) = 0$ for all $i \neq s$ and because $\bar{\mathbf{P}}_\pi(s, f_{sas'}) = \pi(s,a)\mathrm{Pr}(s,a,s')$ by construction,

$$\begin{aligned}\sum_{i\in\mathcal{S}} \mathbf{d}_\pi(i)\bar{\mathbf{P}}_\pi(i, f_{sas'}) &= \mathbf{d}_\pi(s)\bar{\mathbf{P}}_\pi(s, f_{sas'}) \\ &= \mathbf{d}_\pi(s)\pi(s,a)\mathrm{Pr}(s,a,s') \\ &= c\ \bar{\mathbf{d}}_\pi(f_{sas'}).\end{aligned}$$

For the second component, because $\bar{\mathbf{P}}_\pi(f, j) = 0$ for all $f, j \in \mathcal{F}$, we get

$$\sum_{f\in\mathcal{F}} \bar{\mathbf{d}}_\pi(f)\bar{\mathbf{P}}_\pi(f, j) = 0.$$

Therefore, for $j = f_{ss'} \in \mathcal{F}$, $\bar{\mathbf{d}}_\pi \bar{\mathbf{P}}_\pi(:, j) = \bar{\mathbf{d}}_\pi(j)$.

Finally, clearly by normalizing the first component of $\bar{\mathbf{d}}_\pi$ over $s \in \mathcal{S}$, we get the same proportion across states as in $\mathbf{d}_\pi$, satisfying (6).

To see why $\bar{\mathbf{v}}_\pi(s) = \mathbf{v}_\pi(s)$ for all $s \in \mathcal{S}$, first notice that

$$\bar{\mathbf{r}}_\pi(i) = \begin{cases} \mathbf{r}_\pi(i) & i \in \mathcal{S} \\ 0 & \text{otherwise} \end{cases}$$

and for any $f_{sas'} \in \mathcal{F}$

$$\begin{aligned}\bar{\mathbf{v}}_\pi(f_{sas'}) &= 0 + \sum_{j\in\bar{\mathcal{S}}} \bar{\mathbf{P}}_\pi(f_{sas'}, j)\bar{\gamma}_s(j)\bar{\mathbf{v}}_\pi(j) \\ &= \bar{\mathbf{v}}_\pi(s').\end{aligned}$$

Now for any $s \in \mathcal{S}$,

$$\begin{aligned}\bar{\mathbf{v}}_\pi(s) &= \bar{\mathbf{r}}_\pi(s) + \sum_{j\in\bar{\mathcal{S}}} \bar{\mathbf{P}}_\pi(s, j)\bar{\gamma}_s(j)\bar{\mathbf{v}}_\pi(j) \\ &= \mathbf{r}_\pi(s) + \sum_{f_{sas'}\in\mathcal{F}} \bar{\mathbf{P}}_\pi(s, f_{sas'})\bar{\gamma}_s(f_{sas'})\bar{\mathbf{v}}_\pi(f_{sas'}) \\ &= \mathbf{r}_\pi(s) + \sum_{s'\in\mathcal{S}}\sum_{a\in\mathcal{A}} \mathrm{Pr}(s,a,s')\gamma(s,a,s')\bar{\mathbf{v}}_\pi(s')\end{aligned}$$

Therefore, because it satisfies the same fixed point equation, $\bar{\mathbf{v}}_\pi(s) = \mathbf{v}_\pi(s)$ for all $s \in \mathcal{S}$.

With this equivalence, it is clear that

$$\begin{aligned}&\sum_{i\in\bar{\mathcal{S}}} \bar{\mathbf{d}}_\pi(i)\bar{\mathbf{v}}_\pi(i) \\ &= \frac{1}{c}\sum_{s\in\mathcal{S}} \mathbf{d}_\pi(s)\mathbf{v}_\pi(s) + \frac{1}{c}\sum_{f_{ss'}\in\mathcal{F}} \mathbf{d}_\pi(s)\mathbf{P}_\pi(s,s')\mathbf{v}_\pi(s') \\ &= \frac{1}{c}\sum_{s\in\mathcal{S}} \mathbf{d}_\pi(s)\mathbf{v}_\pi(s) + \frac{1}{c}\sum_{s\in\mathcal{S}}\sum_{s'\in\mathcal{S}} \mathbf{d}_\pi(s)\mathbf{P}_\pi(s,s')\mathbf{v}_\pi(s') \\ &= \frac{1}{c}\sum_{s\in\mathcal{S}} \mathbf{d}_\pi(s)\mathbf{v}_\pi(s) + \frac{1}{c}\sum_{s'\in\mathcal{S}} \mathbf{d}_\pi(s')\mathbf{v}_\pi(s') \\ &= \frac{2}{c}\sum_{s\in\mathcal{S}} \mathbf{d}_\pi(s)\mathbf{v}_\pi(s)\end{aligned}$$

Therefore, optimizing either results in the same policy. ■

C. Discounting and average reward for control

The common wisdom is that discounting is useful for asking predictive questions, but for control, the end goal is average reward. One of the main reasons for this view is that it has been previously shown that, for a constant discount, optimizing the expected return is equivalent to optimizing average reward. This can be easily seen by expanding the expected return weighting according to the stationary distribution for a policy, given constant discount $\gamma_c < 1$,

$$\begin{aligned}\mathbf{d}_\pi \mathbf{v}_\pi &= \mathbf{d}_\pi(\mathbf{r}_\pi + \mathbf{P}_{\pi,\gamma}\mathbf{v}_\pi) &(7)\\ &= \mathbf{d}_\pi\mathbf{r}_\pi + \gamma_c\mathbf{d}_\pi\mathbf{P}_\pi\mathbf{v}_\pi \\ &= \mathbf{d}_\pi\mathbf{r}_\pi + \gamma_c\mathbf{d}_\pi\mathbf{v}_\pi \\ \implies \mathbf{d}_\pi\mathbf{v}_\pi &= \frac{1}{1-\gamma_c}\mathbf{d}_\pi\mathbf{r}_\pi. &(8)\end{aligned}$$

Therefore, the constant $\gamma_c < 1$ simply scales the average reward objective, so optimizing either provides the same policy. This argument, however, does not extend to transition-based discounting, because $\gamma(s,a,s')$ can significantly change weighting in returns in a non-uniform way, affecting the choice of the optimal policy. We demonstrate this in the case study for the taxi domain in Section 3.

D. Algorithms

We show how to write generalized pseudo-code for two algorithms: true-online TD (λ) and ELSTDQ(λ). We choose these two algorithms because they generally demonstrate how one would extend to transition-based γ, and further previously had a few unclear points in their implementation. For TO-TD, the pseudo-code has been given for episodic tasks (van Seijen and Sutton, 2014), rather than more generally, and has treated v_{old} carefully at the beginning of episodes, which is not necessary. LSTDQ has typically only

Algorithm 1 True-online TD(λ)

$\mathbf{w} \leftarrow \mathbf{0}, \quad \mathbf{e} \leftarrow \mathbf{0}, \quad v_{\text{old}} \leftarrow 0$
Obtain initial $\mathbf{x}_0$
while agent interacting with environment, $t = 0, 1, \ldots$ **do**
  Obtain next feature vector $\mathbf{x}_{t+1}$, reward r_{t+1} and discount γ_{t+1}
  $\hat{v} = \mathbf{w}^\top \mathbf{x}_t$
  $\hat{v}' = \mathbf{w}^\top \mathbf{x}_{t+1}$
  $\delta \leftarrow r_{t+1} + \gamma_{t+1}\hat{v}' - \hat{v}$
  $\mathbf{e} \leftarrow \mathbf{e} + \mathbf{x}_t$
  $\mathbf{w} \leftarrow \mathbf{w} + \alpha(\delta + \hat{v} - v_{\text{old}})\mathbf{e} - \alpha(\hat{v} - v_{\text{old}})\mathbf{x}_t$
  $v_{\text{old}} \leftarrow \hat{v}'$
  $\mathbf{e} \leftarrow \gamma_{t+1}\lambda_{t+1}\mathbf{e} - \alpha\gamma_{t+1}\lambda_{t+1}(\mathbf{e}^\top \mathbf{x}_{t+1})\mathbf{x}_{t+1}$
return $\mathbf{w}$

been written for a batch of data, without importance sampling; we provide an ELSTDQ variant here with importance sampling, where LSTDQ is a special case using $M = 1$.

There are a few other implementation details that merit clarification. We use the notation γ_{t+1} for $\gamma(s_t, a_t, s_{t+1})$, and λ_{t+1} for $\lambda(s_t, a_t, s_{t+1})$. Further, unlike previous pseudocode, we do not reinitialize v_{old} specially at the start of an episode (i.e., when $\gamma_{t+1} = 0$). This is because the value of v_{old} is not relevant for the next step after $\gamma_{t+1} = 0$. The eligibility trace is zeroed, and so $\alpha(\delta + \hat{v} - v_{\text{old}})\mathbf{e} - \alpha(\hat{v} - v_{\text{old}})\mathbf{x} = \alpha\delta\mathbf{x}$. Finally, for both algorithms, we stage the updates to the traces. This is to avoid saving both γ_t, λ_t and $\gamma_{t+1}, \lambda_{t+1}$ across timesteps.

Algorithm 2 ELSTDQ(λ)

$\mathbf{A} \leftarrow \mathbf{0}, \quad \mathbf{b} \leftarrow \mathbf{0}, \quad \mathbf{e} \leftarrow \mathbf{0}$
$F \leftarrow 0, \quad M \leftarrow 0$
Obtain initial action-value feature vector $\mathbf{x}_0$ (implicitly $\mathbf{x}(s_0, a_0)$) and action a_0
while agent interacting with environment, $t = 0, 1, \ldots$ **do**
  Obtain next action-value feature vector $\mathbf{x}_{t+1}$, action a_{t+1} reward r_{t+1} and discount γ_{t+1}
  $\rho_{t+1} \leftarrow \frac{\pi(\mathbf{x}_{t+1}, a_{t+1})}{\mu(\mathbf{x}_{t+1}, a_{t+1})}$
  $F \leftarrow \gamma_{t+1} F + i(\mathbf{x}_t)$
  $M \leftarrow \lambda_{t+1} i(\mathbf{x}_t) + (1 - \lambda_{t+1})F$
  $\mathbf{e} \leftarrow \mathbf{e} + M\mathbf{x}_t$
  $\mathbf{A} \leftarrow \mathbf{A} + \mathbf{e}\left(\mathbf{x}_t - \rho_{t+1}\gamma_{t+1}\mathbf{x}_{t+1}\right)^\top$
  $\mathbf{b} \leftarrow \mathbf{b} + \mathbf{e} r_{t+1}$
  $\mathbf{e} \leftarrow \gamma_{t+1}\lambda_{t+1}\rho_{t+1}\mathbf{e}$
  $F \leftarrow \rho_{t+1} F$
return $\mathbf{A}^{-1}\mathbf{b}$ // The solution $\mathbf{w}$ to the linear system

E. Lemmas

To bound the maximum eigenvalues of the discount-weighted transition matrix, we first provide the following lemma. This lemma is independently interesting, in that it explicitly verifies the previous Assumption 2.1 (Yu, 2015).

Lemma 3. *Under Assumption A3, the maximum eigenvalue of $\mathbf{P}_{\pi,\gamma}$ is less than 1:*

$$r(\mathbf{P}_{\pi,\gamma}) < 1.$$

Proof:

Part 1: We first show that $r(\tilde{\mathbf{M}}) < 1$ for

$$\tilde{\mathbf{M}} = \begin{cases} \mathbf{M}_{kl} - \delta & \text{if } i = k, j = l \\ \mathbf{M}_{kl} & \text{otherwise.} \end{cases}$$

for any i, j, and any $0 < \delta < \mathbf{M}_{ij}$.

For $\mathbf{M} = \mathbf{P}_\pi$, we know that $r(\mathbf{M}) = 1$ and that, by assumption, $\mathbf{P}_\pi$ is irreducible. We know that $\tilde{\mathbf{M}}$ is still irreducible, because the connectivity is not changed (since no additional entries are zeroed). By the Perron-Frobenius theorem, we know that the eigenvector $\mathbf{x}$ that corresponds to the maximum eigenvalue $r(\tilde{\mathbf{M}})$ has strictly positive entries and so $\delta\mathbf{x}_i\mathbf{x}_j > 0$. Therefore,

$$\begin{aligned} \mathbf{x}^\top \tilde{\mathbf{M}} \mathbf{x} &= \sum_{k,l:(k,l)\neq(i,j)} \mathbf{x}_k \mathbf{M}_{kl} \mathbf{x}_l + \mathbf{x}_i(\mathbf{M}_{ij} - \delta)\mathbf{x}_j \\ &= \mathbf{x}^\top \mathbf{M}\mathbf{x} - \delta\mathbf{x}_i\mathbf{x}_j \\ &< \mathbf{x}^\top \mathbf{M}\mathbf{x}. \end{aligned}$$

We know that $\mathbf{x}^\top \mathbf{M}\mathbf{x} \leq 1$, because $r(\mathbf{M}) = 1$ and $\mathbf{x}$ is a unit vector. Therefore, using the fact that $r(\tilde{\mathbf{M}}) = \mathbf{x}^\top \tilde{\mathbf{M}}\mathbf{x}$ by Courant-Fischer-Weyl[4], we get

$$r(\tilde{\mathbf{M}}) = \mathbf{x}^\top \tilde{\mathbf{M}}\mathbf{x} < 1.$$

Part 2: Next we show that further reducing entries, even to zero values, will not increase the maximum eigenvalue. This follows simply from the fact that non-negative matrices are guaranteed to have a non-negative eigenvector $\mathbf{x}$ that corresponds to the maximum eigenvalue.

To see why, for notational convenience, we now let $\mathbf{M}$ be the matrix where entry i, j in $\mathbf{P}_\pi$ was reduced by δ. Let $\tilde{\mathbf{M}}$ further reduce an entry by δ, now potentially to a minimum value of 0, so that $\tilde{\mathbf{M}}$ is guaranteed to be non-negative (rather than strictly positive). Using the same argument as above,

[4]The Courant-Fischer-Weyl min-max principle states that the maximum eigenvalue $r(\mathbf{M})$ of a matrix $\mathbf{M}$ corresponds to $\text{argmax}_{\mathbf{x}:\mathbf{x}\neq\mathbf{0}} \mathbf{x}^\top \mathbf{M}\mathbf{x}/(\mathbf{x}^\top\mathbf{x})$, where the corresponding $\mathbf{x}$ that gives the maximum is an eigenvector of $\mathbf{M}$. Therefore, for this $\mathbf{x}$, $\mathbf{x}^\top \mathbf{M}\mathbf{x} = r(\mathbf{M})$ and $\|\mathbf{x}\|_2^2 = \mathbf{x}^\top\mathbf{x} = 1$.

we obtain that for the non-negative eigenvector of $\tilde{\mathbf{M}}$

$$\begin{aligned}\mathbf{x}^\top \tilde{\mathbf{M}}\mathbf{x} &= \mathbf{x}^\top \mathbf{M}\mathbf{x} - \delta \mathbf{x}_i \mathbf{x}_j \\ &\leq r(\mathbf{M})\end{aligned}$$

because $\delta \mathbf{x}_i \mathbf{x}_j \geq 0$. Therefore, with further reduction, $r(\tilde{\mathbf{M}})$ cannot increase and so $r(\tilde{\mathbf{M}}) \leq r(\mathbf{M}) < 1$.

Part 3: Finally, we can see that for any γ and $\mathbf{P}_\pi$ as given under Assumptions 2 and 3, $\mathbf{M} = \mathbf{P}_{\pi,\gamma}$ satisfies the above construction. ■

Now we additionally provide definitions for the extension to transition-based discounts. To do so, we will need to define

$$\begin{aligned}\mathbf{P}_{\pi,\gamma,\lambda}(s,s') &:= \sum_{a\in\mathcal{A}} \pi(s,a)\Pr(s,a,s')\gamma(s,a,s')\lambda(s,a,s') \\ \mathbf{P}_{\pi,\gamma,1\text{–}\lambda}(s,s') &:= \sum_{a\in\mathcal{A}} \pi(s,a)\Pr(s,a,s')\gamma(s,a,s')(1-\lambda(s,a,s')) \\ &= \mathbf{P}_{\pi,\gamma} - \mathbf{P}_{\pi,\gamma,\lambda}\end{aligned}$$

Then we obtain the following generalized definition of $\mathbf{P}^\lambda_\pi$ and necessary properties for convergence within ETD.

Lemma 4. *Under Assumption A3, $\mathbf{I} - \mathbf{P}_{\pi,\gamma,\lambda}$ is non-singular and the matrix*

$$\mathbf{P}^\lambda_\pi = (\mathbf{I} - \mathbf{P}_{\pi,\gamma,\lambda})^{-1}\, \mathbf{P}_{\pi,\gamma,1\text{–}\lambda}$$

is non-negative and has rows that sum to no greater than 1,

$$0 \leq \mathbf{P}^\lambda_\pi \leq 1 \qquad \textit{and} \qquad \mathbf{P}^\lambda_\pi \mathbf{1} \leq \mathbf{1}.$$

Proof:

By Lemma 3, we know $r(\mathbf{P}_{\pi,\gamma}) < 1$. Because $0 \leq \lambda(s,a,s') \leq 1$ for all (s,a,s'), this means that $r(\mathbf{P}_{\pi,\gamma,1\text{–}\lambda}) < 1$. Therefore $\mathbf{I} - \mathbf{P}_{\pi,\gamma,\lambda}$ is non-singular and so $\mathbf{P}^\lambda_\pi$ is well-defined.

Notice that $\mathbf{I} - \mathbf{P}_{\pi,\gamma,\lambda}$ is a non-singular *M-matrix*, since the maximum eigenvalue of $\mathbf{P}_{\pi,\gamma,\lambda}$ is less than one and entry-wise $\mathbf{P}_{\pi,\gamma,\lambda} \geq 0$. Therefore, the inverse of $\mathbf{I} - \mathbf{P}_{\pi,\gamma,\lambda}$ is positive, making $(\mathbf{I} - \mathbf{P}_{\pi,\gamma,\lambda})^{-1}\, \mathbf{P}_{\pi,\gamma,1\text{–}\lambda}$ a positive matrix. The fact that the matrix has entries that are less than or equal to 1 follows from showing $\mathbf{P}^\lambda_\pi \mathbf{1} \leq \mathbf{1}$ below.

To show that the matrix rows always sum to less than 1, we use a simple inductive argument. Since

$$\mathbf{P}^\lambda_\pi = \sum_{k=0}^{\infty} (\mathbf{P}_{\pi,\gamma,\lambda})^k \mathbf{P}_{\pi,\gamma,1\text{–}\lambda}$$

we simply need to show that for every t, $\sum_{k=0}^{t} (\mathbf{P}_{\pi,\gamma,\lambda})^k \mathbf{P}_{\pi,\gamma,1\text{–}\lambda}\mathbf{1} \leq \mathbf{1}$.

For the base case, $\mathbf{t} = \mathbf{0}$: clearly

$$\mathbf{P}_{\pi,\gamma,1\text{–}\lambda}\mathbf{1} \leq \mathbf{1}$$

For $\mathbf{t} > \mathbf{0}$: Assume that

$$\sum_{k=0}^{t} (\mathbf{P}_{\pi,\gamma,\lambda})^k \mathbf{P}_{\pi,\gamma,1\text{–}\lambda}\mathbf{1} \leq \mathbf{1}.$$

Then

$$\begin{aligned}&\sum_{k=0}^{t+1} (\mathbf{P}_{\pi,\gamma,\lambda})^k \mathbf{P}_{\pi,\gamma,1\text{–}\lambda}\mathbf{1} \\ &= \mathbf{P}_{\pi,\gamma,1\text{–}\lambda}\mathbf{1} + \left[\sum_{k=1}^{t+1} (\mathbf{P}_{\pi,\gamma,\lambda})^k \mathbf{P}_{\pi,\gamma,1\text{–}\lambda}\mathbf{1}\right] \\ &= \mathbf{P}_{\pi,\gamma,1\text{–}\lambda}\mathbf{1} + \mathbf{P}_{\pi,\gamma,\lambda}\left[\sum_{k=0}^{t} (\mathbf{P}_{\pi,\gamma,\lambda})^k \mathbf{P}_{\pi,\gamma,1\text{–}\lambda}\mathbf{1}\right] \\ &\leq \mathbf{P}_{\pi,\gamma,1\text{–}\lambda}\mathbf{1} + \mathbf{P}_{\pi,\gamma,\lambda}\mathbf{1} \\ &= \mathbf{P}_{\pi,\gamma}\mathbf{1} \\ &\leq \mathbf{1}\end{aligned}$$

completing the proof. ■

F. Convergence of emphatic algorithms for the RL task formalism

We start with convergence in expectation of ETD for transition-based discounts. The results for state-based MDPs should automatically extend to transition-based MDPs, due to the equivalence proved in Section B.1. However, in an effort to similarly generalize the writing of the theoretical analysis to the more general transition-based MDP setting, as we did for algorithms and implementation, we explicitly extend the proof for transition-based MDPs.

Theorem 3. *Assume the value function is approximated using linear function approximation: $\mathbf{v}(s) = \mathbf{x}(s)^\top \mathbf{w}$. For $\mathbf{X}$ with linearly independent columns (i.e. linearly independent features), with an interest function $\mathbf{i} : \mathcal{S} \to (0,\infty)$ and $\mathbf{M} = \mathrm{diag}(\mathbf{m})$ for $\mathbf{m} = (\mathbf{I} - \mathbf{P}^\lambda_\pi)^{-1}(\mathbf{d} \circ \mathbf{i})$, the matrix $\mathbf{A}$ is positive definite.*

$$\begin{aligned}\mathbf{A} &:= \mathbf{X}^\top \mathbf{M}(\mathbf{I} - \mathbf{P}_{\pi,\gamma,\lambda})^{-1}(\mathbf{I} - \mathbf{P}_{\pi,\gamma})\mathbf{X} \\ &= \mathbf{X}^\top \mathbf{M}(\mathbf{I} - \mathbf{P}^\lambda_\pi)\mathbf{X}\end{aligned}$$

is positive definite.

Proof:

First, we write an equivalent definition for $\mathbf{P}^\lambda_\pi$,

$$\begin{aligned}\mathbf{P}^\lambda_\pi &= (\mathbf{I} - \mathbf{P}_{\pi,\gamma,\lambda})^{-1}\, \mathbf{P}_{\pi,\gamma,1\text{–}\lambda} \\ &= (\mathbf{I} - \mathbf{P}_{\pi,\gamma,\lambda})^{-1}\, (\mathbf{P}_{\pi,\gamma} - \mathbf{P}_{\pi,\gamma,\lambda}) \\ &= (\mathbf{I} - \mathbf{P}_{\pi,\gamma,\lambda})^{-1}\, (\mathbf{P}_{\pi,\gamma} - \mathbf{I} + \mathbf{I} - \mathbf{P}_{\pi,\gamma,\lambda}) \\ &= \mathbf{I} - (\mathbf{I} - \mathbf{P}_{\pi,\gamma,\lambda})^{-1}(\mathbf{I} - \mathbf{P}_{\pi,\gamma}).\end{aligned}$$

Since $\mathbf{X}$ is a full rank matrix, to prove that

$$\mathbf{A} = \mathbf{X}^\top \mathbf{M}(\mathbf{I} - \mathbf{P}_\pi^\lambda)\mathbf{X}$$

is positive definite, we need to prove that $\mathbf{M}(\mathbf{I} - \mathbf{P}_\pi^\lambda)$ is positive definite.

As in (Sutton et al., 2016, Theorem 1), (Yu, 2015, Proposition C.1), we need to show that for $\mathbf{M}(\mathbf{I} - \mathbf{P}_\pi^\lambda)$, (a) the diagonal entries are nonnegative, (b) the off-diagonal entries are nonpositive (c) its row sums are nonnegative and (d) the columns sums are are positive. The requirements (a) - (c) follow from Lemma 4, because $\mathbf{M}$ is a non-negative diagonal weighting matrix. To show (d), first if $\mathbf{i}(s) > 0$ for all $s \in \mathcal{S}$, the vector of columns sums is

$$\mathbf{1}^\top \mathbf{M}(\mathbf{I} - \mathbf{P}_\pi^\lambda) = \mathbf{m}^\top (\mathbf{I} - \mathbf{P}_\pi^\lambda) = (\mathbf{d}_\pi \circ \mathbf{i})^\top$$

which always has positive entries.

Otherwise, if $\mathbf{i}(s) = 0$ for some $s \in \mathcal{S}$, we can prove that $\mathbf{M}(\mathbf{I} - \mathbf{P}_\pi^\lambda)$ is positive definite using the same argument as in (Yu, 2015, Corollary C.1). The proof nicely encapsulates $\mathbf{P}_\pi^\lambda$ generically as a matrix $\mathbf{Q}$. We simply have to ensure that the inverse of $\mathbf{I} - \mathbf{P}_\pi^\lambda$ exists and that $\mathbf{P}_\pi^\lambda$ has entries less than or equal to 1, both of which were showed in Lemma 4. The first condition is to have well-defined matrices, and the second to ensure that $\mathbf{Q}$ has a block-diagonal structure. Therefore, under Assumption 4, we can follow the same proof as (Yu, 2015, Corollary C.1) to ensure that $\mathbf{A}$ is positive definite. ∎

For the proofs for ELSTDQ, the main difference is in using action-value functions. We construct the augmented space, with states $\bar{\mathcal{S}} = \mathcal{S} \times \mathcal{A}$ and

$$\mathbf{P}_{\pi,\gamma,q}((s,a),(s,a')) := P(s,a,s')\gamma(s,a,s')\pi(s',a')$$
$$\mathbf{P}_{\pi,\gamma,\lambda,q}((s,a),(s,a')) := P(s,a,s')\gamma(s,a,s')\lambda(s,a,s')\pi(s',a')$$

giving

$$\begin{aligned}
\mathbf{i}_q((s,a)) &:= \mathbf{i}(s)\\
\mathbf{d}_{\mu,q}((s,a)) &:= \mathbf{d}_\mu(s)\mu(s,a)\\
\mathbf{r}_q((s,a)) &:= \sum_{s'\in\mathcal{S}} \Pr(s,a,s')r(s,a,s')\\
\mathbf{P}_\pi^{\lambda,q} &:= (\mathbf{I} - \mathbf{P}_{\pi,\gamma,q})^{-1}(\mathbf{P}_{\pi,\gamma,q} - \mathbf{P}_{\pi,\gamma,\lambda,q})\\
\mathbf{M}_q &:= \operatorname{diag}\left(\mathbf{d}_{\mu,q} \circ \mathbf{i}_q (\mathbf{I} - \mathbf{P}_\pi^{\lambda,q})^{-1}\right).
\end{aligned}$$

Then

$$\begin{aligned}
\mathbf{A}_q &:= \mathbf{X}_q^\top \mathbf{M}_q (\mathbf{I} - \mathbf{P}_\pi^{\lambda,q})\mathbf{X}\\
\mathbf{b}_q &:= \mathbf{X}_q^\top \mathbf{M}_q (\mathbf{I} - \mathbf{P}_{\pi,\gamma,q})\mathbf{r}_q.
\end{aligned}$$

The projected Bellman operator is defined as $\Pi_{\mathbf{M}_q} T^{(\lambda,q)}\mathbf{q} = \mathbf{r}_q + \mathbf{P}_\pi^{\lambda,q}\mathbf{q}$ where ELSTD(λ) converges to the projected Bellman operator fixed point $\Pi_{\mathbf{M}_q} T^{(\lambda,q)}\mathbf{q} = \mathbf{q}$.

Further, this MDP with the assumptions on the subspace produced by the state-action features satisfies the conditions of Theorem 3, and so $\mathbf{A}_q$ is also positive definite.

Similarly, the other properties of the Bellman operator and the weighted norm on $\mathbf{P}_\pi^{\lambda,q}$ extend, giving a unique fixed point for the action-value Bellman operator $\mathbf{P}_\pi^{\lambda,q}$ and $\|\mathbf{P}_\pi^{\lambda,q}\|_{\mathbf{M}_q} < 1$.

Corollary 1. *Assume the action-value function is approximated using linear function approximation: $\mathbf{x}(s,a)^\top\mathbf{w}$. For $\mathbf{X}$ with linearly independent columns (i.e. linearly independent features), $\mathbf{A}_q$ is positive definite.*

G. Issues with transition-based trace without emphatic weighting

A natural goal is to similarly generalize the contraction properties of $\mathbf{P}_\pi^\lambda$ under the weighting $\mathbf{d}_\pi$, from constant λ_c to transition-based trace. To do so, unlike under emphatic weighting, we need to restrict the set of possible trace functions. Notice that, because of Assumption A3, for some $s_\lambda < 1$ and $s_{1-\lambda} < 1$, for any non-negative $\mathbf{v}_+$,

$$\begin{aligned}
&\mathbf{d}_\pi \mathbf{P}_{\pi,\gamma,\lambda}\mathbf{v}_+\\
&= \sum_s \sum_a \mathbf{d}_\pi(s)\Pr(s,a,:) \circ \gamma(s,a,:) \circ \lambda(s,a,:)\mathbf{v}_+\\
&\le s_\lambda \sum_s \sum_a \mathbf{d}_\pi(s)\Pr(s,a,:)\mathbf{v}_+ = s_\lambda \mathbf{d}_\pi \mathbf{v}_+
\end{aligned}$$

and similarly

$$\mathbf{d}_\pi \mathbf{P}_{\pi,\gamma,1-\lambda}\mathbf{v}_+ \le s_{1-\lambda}\mathbf{d}_\pi \mathbf{v}_+.$$

The generalized bound on the $\mathbf{d}_\pi$ weighted norm is given in the following lemma.

Lemma 5.

$$\|\mathbf{P}_\pi^\lambda\|_{\mathbf{D}_\pi} \le \frac{s_{1-\lambda}}{1 - s_\lambda}.$$

Now, the norm is only a contraction if $s_{1-\lambda} < 1 - s_\lambda$. As we have seen, for constant trace, this inequality holds, since $s_{1-\lambda} = s(1-\lambda)$ and $s_\lambda = s\lambda$ for some $s < 1$. In general, however, there are instances where this is not true. We provide such an example below.[5]

Consider a 2-state MDP, with uniform probabilities of transitioning and uniform policy, and so $\mathbf{d}_\pi = [0.5, 0.5]$. Let $\gamma_c = 0.99$ and set λ to be 0.9 when entering state s_1 and 0

[5] Thanks to an anonymous reviewer for pointing out this example.

when entering state s_2. Then for any $\mathbf{v}_+$,

$$\begin{aligned}
\mathbf{d}_\pi \mathbf{P}_{\pi,\gamma,\lambda} \mathbf{v}_+ &= \gamma_c \mathbf{d}_\pi \mathbf{P}_\pi \begin{bmatrix} 0.9 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{v}_+ \\
&= \gamma_c \mathbf{d}_\pi \begin{bmatrix} 0.9 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{v}_+ \\
&= \gamma_c 0.9 \mathbf{d}_\pi \begin{bmatrix} 1.0 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{v}_+ \\
&\leq \gamma_c 0.9 \mathbf{d}_\pi \mathbf{v}_+
\end{aligned}$$

where for $\mathbf{v}_+ = [v \;\; 0]^\top$ for any $v \geq 0$, this bound is tight. Similarly,

$$\begin{aligned}
\mathbf{d}_\pi \mathbf{P}_{\pi,\gamma,1-\lambda} \mathbf{v}_+ &= \gamma_c \mathbf{d}_\pi \mathbf{P}_\pi \begin{bmatrix} 0.1 & 0 \\ 0 & 1.0 \end{bmatrix} \mathbf{v}_+ \\
&\leq \gamma_c \mathbf{d}_\pi \mathbf{v}_+.
\end{aligned}$$

where for $\mathbf{v}_+ = [0 \;\; v]^\top$ for any $v \geq 0$, this bound is tight.

Therefore, $s_\lambda = 0.9\gamma_c$ and $s_{1-\lambda} = \gamma_c$, and so we get $1 - s_\lambda = 0.9\gamma_c < s_{1-\lambda} = \gamma_c$, which makes the upper bound in the above lemma $1.\bar{1}$. Computing $\mathbf{P}_\pi^\lambda$,

$$\mathbf{P}_\pi^\lambda = \begin{bmatrix} 0.0893 & 0.8927 \\ 0.0893 & 0.8927 \end{bmatrix}.$$

we can see that this is not a contraction.