GQ(λ): A general gradient algorithm for temporal-difference prediction learning with eligibility traces

Hamid Reza Maei and **Richard S. Sutton**
Reinforcement Learning and Artificial Intelligence Laboratory, University of Alberta, Edmonton, Canada

Abstract

A new family of gradient temporal-difference learning algorithms have recently been introduced by Sutton, Maei and others in which function approximation is much more straightforward. In this paper, we introduce the GQ(λ) algorithm which can be seen as extension of that work to a more general setting including eligibility traces and off-policy learning of temporally abstract predictions. These extensions bring us closer to the ultimate goal of this work—the development of a universal prediction learning algorithm suitable for learning experientially grounded knowledge of the world. Eligibility traces are essential to this goal because they bridge the temporal gaps in cause and effect when experience is processed at a temporally fine resolution. Temporally abstract predictions are also essential as the means for representing abstract, higher-level knowledge about courses of action, or options. GQ(λ) can be thought of as an extension of Q-learning. We extend existing convergence results for policy evaluation to this setting and carry out a forward-view/backward-view analysis to derive and prove the validity of the new algorithm.

Introduction

One of the main challenges in artificial intelligence (AI) is to connect the low-level experience to high-level representations (grounded world knowledge). Low-level experience refers to rich signals received back and forth between the agent and the world. Recent theoretical developments in temporal-difference learning combined with mathematical ideas developed for temporally abstract options, known as intra-option learning, can be used to address this challenge (Sutton, 2009).

Intra-option learning (Sutton, Precup, and Singh, 1998) is seen as a potential method for temporal-abstraction in reinforcement learning. Intra-option learning is a type of off-policy learning. Off-policy learning refers to learning about a *target policy* while following another policy, known as *behavior policy*. Off-policy learning arises in Q-learning where the target policy is a greedy optimal policy while the behavior policy is exploratory. It is also needed for intra-option learning. Intra-option methods look inside options and allow AI agent to learn about multiple different options simultaneously from a single stream of received data. *Option* refers to a temporally course of actions with a termination condition. Options are ubiquitous in our everyday life. For example, to go for hiking, we need to consider and evaluate multiple options such as transportation options to the hiking trail. Each option includes a course of primitive actions and only is excited in particular states. The main feature of intra-option learning is its ability to predict the consequences of each option policy without executing it while data is received from a different policy.

Temporal difference (TD) methods in reinforcement learning are considered as powerful techniques for prediction problems. In this paper, we consider predictions always in the form of answers to the questions. Questions are like "If of follow this trail, would I see a creek?" The answers to such questions are in the form of a single scalar (value function) that tells us about the expected future consequences given the current state. In general, due to the large number of states, it is not feasible to compute the exact value of each state entry. One of the key features of TD methods is their ability to generalize predictions to states that may not have visited; this is known as function approximation.

Recently, Sutton et al. (2009b) and Maei et al. (2009) introduced a new family of gradient TD methods in which function approximation is much more straightforward than conventional methods. Prior to their work, the existing classical TD algorithms (e.g.; TD(λ) and Q-learning) with function approximation could become unstable and diverge (Baird, 1995; Tsitsiklis and Van Roy, 1997).

In this paper, we extend their work to a more general setting that includes off-policy learning of temporally abstract predictions and eligibility traces. Temporally abstract predictions are essential for representing higher-level knowledge about the course of actions, or options (Sutton et al., 1998). Eligibility traces bridge between the temporal gaps when experience is processes at a temporally fine resolution.

In this paper, we introduce the GQ(λ) algorithm that can be thought of as an extension to Q-learning (Watkins and Dayan, 1989); one of the most popular off-policy learning algorithms in reinforcement learning.

Our algorithm incorporates gradient-descent ideas originally developed by Sutton et al. (2009a,b), for option conditional predictions with varying eligibility traces. We extend existing convergence results for policy evaluation to this setting and carry forward-view/backward-view analysis and prove the validity of the new algorithm.

The organization of the paper is as follows: First, we describe the problem setting and define our notations. Then we introduce the GQ(λ) algorithm and describe how to use it. In the next sections we provide derivation of the algorithm and carry out analytical analysis on the equivalence of TD forward-view/backward-view. We finish the paper with convergence proof and conclusion section.

Notation and background

We consider the problem of policy evaluation in finite state-action Markov Decision Process (MDP). Under standard conditions, however, our results can be extended to MDPs with infinite state–action pairs. We use a standard reinforcement learning (RL) framework. In this setting, data is obtained from a continually evolving MDP with states $s_t \in \mathcal{S}$, actions $a_t \in \mathcal{A}$, and rewards $r_t \in \Re$, for $t = 1, 2, \ldots$, with each state and reward as a function of the preceding state and action. Actions are chosen according to the behavior policy b, which is assumed fixed and exciting, $b(s, a) > 0, \forall s, a$. We consider the transition probabilities between state–action pairs, and for simplicity we assume there is a finite number N of state–action pairs.

Suppose the agent find itself at time t in a state–action pair s_t, a_t. The agent likes its answer at that time to tell something about the future sequence $s_{t+1}, a_{t+1}, \ldots, s_{t+k}$ if actions from $t+1$ on were taken according to the option until it terminated at time $t+k$. The option policy is denoted $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ and whose termination condition is denoted $\beta : \mathcal{S} \to [0, 1]$.

The answer is always in the form of a single number, and of course we have to be more specific about what we are trying to predict. There are two common cases: 1) we are trying to predict the outcome of the option; we want to know about the expected value of some function of the state at the time the option terminates. We call this function the *outcome target function*, and denote it $z : \mathcal{S} \to \Re$, 2) we are trying to predict the transient; that is, what happens during the option rather than its end. The most common thing to predict about the transient is the total or discounted reward during the option. We denote the reward function $r : \mathcal{S} \times \mathcal{A} \to \Re$. Finally, the answer could conceivably be a mixture of both a transient *and* an outcome. Here we will present the algorithm for answering questions with both an outcome part z and a transient part r, with the two added together. In the common place where one wants only one of the two, the other is set to zero.

Now we can start to state the goal of learning more precisely. In particular, we would like our answer to be equal to the expected value of the outcome target function at termination plus the cumulative sum of the transient reward function along the way:

$$Q^\pi(s_t, a_t) \equiv \mathbb{E}\big[r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{k-1} r_{t+k} + z_{t+k} \mid \pi, \beta\big], \tag{1}$$

where $\gamma \in (0, 1]$ is discount factor and $Q^\pi(s, a)$ denotes action value function that evaluates policy π given state-action pair s, a. To simplify the notation, from now on, we drop the superscript π on action values.

In many problems the number of state-action pairs is large and therefore it is not feasible to compute the action values for each state-action entry. Therefore, we need to approximate the action values through generalization techniques. Here, we use linear function approximation; that is, the answer to a question is always formed linearly as $Q_\theta(s, a) = \theta^\top \phi(s, a) \approx Q(s, a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$, where $\theta \in \Re^n$ is a learned weight vector and $\phi(s, a) \in \Re^n$ indicates a state–action feature vector. The goal is to learn parameter vector θ through a learning method such as TD learning.

The above (1) describes the target in a Monte Carlo sense, but of course we want to include the possibility of temporal-difference learning; one of the widely used techniques in reinforcement learning. To do this, we provide an eligibility-trace function $\lambda : \mathcal{S} \to [0, 1]$ as described in Sutton and Barto (1998). We let eligibility-trace function, λ, to vary over different states.

In the next section, first we introduce GQ(λ); a general temporal-difference learning algorithm that is stable under off-policy training, and show how to use it. Then in later sections we provide the derivation of the algorithm and convergence proof.

The GQ(λ) algorithm

In this section we introduce the GQ(λ) algorithm for off-policy learning about the outcomes and transients of options, in other words, intra-option GQ(λ) for learning the answer to a question chosen from a wide (possibly universal) class of option-conditional predictive questions.

To specify the question one provides four functions: π and β, for the option, and z and r, for the target functions. To specify how the answers will be formed one provides their functional form (here in linear form), the feature vectors $\phi(s, a)$ for all state–action pairs, and the eligibility-trace function λ. The discount factor γ can be taken to be 1, and thus ignored; the same effect as discounting can be achieved through the choice of β.

Now, we specify the GQ(λ) algorithm as follows: The weight vector $\theta \in \Re^n$ is initialized arbitrarily. The secondary weight vector $w \in \Re^n$ is initialized to zero. An auxiliary memory vector known as the eligibility trace $e \in \Re^n$ is also initialized to zero. Their update rules are

$$\theta_{t+1} = \theta_t + \alpha_{\theta,t}\Big[\delta_t e_t - \kappa_{t+1}(w_t^\top e_t)\bar{\phi}_{t+1}\Big], \tag{2}$$

$$w_{t+1} = w_t + \alpha_{w,t}\left[\delta_t e_t - (w_t^\top \phi_t)\phi_t\right], \tag{3}$$

and

$$e_t = \phi_t + (1-\beta_t)\lambda_t\rho_t e_{t-1}, \tag{4}$$

where,

$$\delta_t = r_{t+1} + \beta_{t+1}z_{t+1} + (1-\beta_{t+1})\theta_t^\top\bar{\phi}_{t+1} - \theta_t^\top\phi_t, \tag{5}$$

$$\bar{\phi}_t = \sum_a \pi(s_t,a)\phi(s_t,a),$$

$$\rho_t = \frac{\pi(s_t,a_t)}{b(s_t,a_t)}, \quad \kappa_t = (1-\beta_t)(1-\lambda_t),$$

ϕ_t is an alternate notation for $\phi(s_t,a_t)$, and $\alpha_{\theta,t} > 0$, $\alpha_{w,t} > 0$, are constant or decreasing step-size parameters for θ and w weights respectively. Here, δ_t, can be seen as one-step TD error.

In the next section we introduce a Bellman error objective function and later show that the GQ(λ) algorithm follows its gradient-descent direction and eventually converges to what that can be described as the fixed-point of TD(λ) under off-policy training.

Objective function

The key element in this paper is to extend the mean-square *projected* Bellman error objective function (MSPBE), first introduced by Sutton et al. (2009b), to the case where it incorporates eligibility traces and option-conditional probabilities. We start with an off-policy, λ-weighted version of the projected-Bellman-error objective function:

$$J(\theta) \quad = \quad \| \, Q_\theta - \Pi T_\pi^{\lambda\beta} Q_\theta \, \|_D^2 \tag{6}$$

where $Q_\theta = \Phi\theta \in \Re^N$ is the vector of approximate action values for each state–action pair, Φ is an $N \times n$ matrix whose rows are the state–action feature vectors $\phi(s,a)$, $\Pi = \Phi(\Phi^\top D\Phi)^{-1}\Phi^\top D$ is a projection matrix that projects any point in the action value space into the linear space of approximate action values, D is an $N \times N$ diagonal matrix whose diagonal entries correspond to the frequency with which each state–action pair is visited under the behavior policy, $T_\pi^{\lambda\beta}$ is a λ-weighted state–action version of the affine $N \times N$ Bellman operator for the target policy π with termination probability β, and finally the norm, $\| \, v \, \|_D^2$, is defined as $v^\top Dv$. The operator $T_\pi^{\lambda\beta}$ takes as input an arbitrary vector $Q \in \Re^N$ and returns a vector giving for each state–action pair the expected corrected λ-return if the Markov decision process was started in that state-action pair, actions were taken according to π, and Q was used to correct the return truncations. When Q_θ is used for the corrections we can write

$$T_\pi^{\lambda\beta} Q_\theta(s,a) = \mathbb{E}_\pi\Big[g_t^{\lambda\beta} \mid s_t = s, a_t = a \Big], \tag{7}$$

where $g_t^{\lambda\beta}$ is the λ-return (while following behavior policy) starting from state-action pair s_t, a_t :

$$\begin{aligned} g_t^{\lambda\beta} \quad &= \quad r_{t+1} + \beta_{t+1}z_{t+1} \\ &\quad +(1-\beta_{t+1})\left[(1-\lambda_{t+1})\theta^\top\phi_{t+1} + \lambda_{t+1}g_{t+1}^{\lambda}\right], \end{aligned} \tag{8}$$

where ϕ_t is an alternate notation for $\phi(s_t,a_t)$.

It would be easier to work with this objective function if we write it in terms of statistical expectations. To do this, first, let's consider the following identities:

$$\mathbb{E}_\pi\Big[\delta_t^{\lambda\beta} \mid s_t = s, a_t = a\Big] = T_\pi^{\lambda\beta}Q_\theta(s,a) - Q_\theta(s,a),$$

where

$$\delta_t^{\lambda\beta} \equiv g_t^{\lambda\beta} - \theta^\top\phi_t, \tag{9}$$

$$\begin{aligned} &\mathbb{E}_\pi\Big[\delta_t^{\lambda\beta}\phi_t\Big] \\ &= \quad \sum_{s,a} D_{sa,sa}\phi(s,a)\mathbb{E}_\pi\Big[\delta_t^{\lambda\beta} \mid s_t = s, a_t = a\Big] \\ &= \quad \Phi^\top D(T_\pi^{\lambda\beta}Q_\theta - Q_\theta), \end{aligned}$$

and

$$\mathbb{E}_b\big[\phi\phi^\top\big] = \sum_{s,a} D_{sa,sa}\phi(s,a)\phi^\top(s,a) = \Phi^\top D\Phi.$$

Note that $D_{sa,sa}$ indicates the diagonal entry of matrix D and corresponds to the frequency with which state-action pair s,a, is visited under the behavior policy b. Here, $\mathbb{E}_\pi[.] = \sum_{s,a} D_{sa,sa}\mathbb{E}_\pi[. \mid s,a]$ because the data has been generated and observed according to the behavior policy.

Given identities above, one can follow similar steps as in Sutton et al. (2009); as follows, to show that the objective function can be written in terms of statistical expectations:

$$\begin{aligned} &J(\theta) \\ &= \quad \| \, Q_\theta - \Pi T_\pi^{\lambda\beta} Q_\theta \, \|_D^2 \\ &= \quad \| \, \Pi(T_\pi^{\lambda\beta} Q_\theta - Q_\theta) \, \|_D^2 \\ &= \quad (\Pi(T_\pi^{\lambda\beta} Q_\theta - Q_\theta))^\top D(\Pi(T_\pi^{\lambda\beta} Q_\theta - Q_\theta)) \\ &= \quad (T_\pi^{\lambda\beta} Q_\theta - Q_\theta)^\top \Pi^\top D\Pi(T_\pi^{\lambda\beta} Q_\theta - Q_\theta) \\ &= \quad (T_\pi^{\lambda\beta} Q_\theta - Q_\theta)^\top D^\top\Phi(\Phi^\top D\Phi)^{-1}\Phi^\top D(T_\pi^{\lambda\beta} Q_\theta - Q_\theta) \\ &= \quad (\Phi^\top D(T_\pi^{\lambda\beta} Q_\theta - Q_\theta))^\top(\Phi^\top D\Phi)^{-1}\Phi^\top D(T_\pi^{\lambda\beta} Q_\theta - Q_\theta) \\ &= \quad \mathbb{E}_\pi\big[\delta^{\lambda\beta}\phi\big]^\top \mathbb{E}_b\big[\phi\phi^\top\big]^{-1}\mathbb{E}_\pi\big[\delta^{\lambda\beta}\phi\big], \end{aligned} \tag{10}$$

where we have used the identity $\Pi^\top D\Pi = D^\top\Phi(\Phi^\top D\Phi)^{-1}\Phi^\top D$.

In our off-policy setting, however, we cannot work with expectations conditional on π; we need to convert them to expectations conditional on b (the behavior policy) which we can then directly sample from. To do this, we introduce an off-policy version of the multi-step TD error, $\delta_t^{\lambda\beta\rho}$,

$$\delta_t^{\lambda\beta\rho} \quad \equiv \quad g_t^{\lambda\beta\rho} - \theta_t^\top\phi_t, \tag{11}$$

where

$$\begin{aligned} g_t^{\lambda\beta\rho} \quad &\equiv \quad r_{t+1} + \beta_{t+1}z_{t+1} \\ &\quad +(1-\beta_{t+1})[(1-\lambda_{t+1})\theta^\top\bar{\phi}_{t+1} + \lambda_{t+1}\rho_{t+1}g_{t+1}^{\lambda\beta\rho}], \end{aligned} \tag{12}$$

is off-policy λ-return and

$$\bar{\phi}_t = \sum_a \pi(s_t, a)\phi(s_t, a) \quad \text{and} \quad \rho_t = \frac{\pi(s_t, a_t)}{b(s_t, a_t)}.$$

The next theorem makes this transformation very simple.

Theorem 1. *Transforming conditional expectations. Let b and π denote the behavior and target policies respectively, and $\delta^{\lambda\beta}$, $\delta^{\lambda\beta\rho}$ are defined in equations (9 ,11), then*

$$\mathbb{E}_\pi\left[\delta_t^{\lambda\beta}\phi_t\right] = \mathbb{E}_b\left[\delta_t^{\lambda\beta\rho}\phi_t\right]. \tag{13}$$

Proof. First, we show $\mathbb{E}_b\left[g_t^{\lambda\beta\rho} \mid s_t, a_t\right] = \mathbb{E}_\pi\left[g_t^{\lambda\beta} \mid s_t, a_t\right]$. To do this, let's write $g_t^{\lambda\beta\rho}$ (12) in the following compact form:

$$g_t^{\lambda\beta\rho} = \zeta_{t+1} + \kappa_{t+1}\theta^\top\bar{\phi}_{t+1} + (1-\beta_{t+1})\lambda_{t+1}\rho_{t+1}g_{t+1}^{\lambda\beta\rho},$$

where $\zeta_t \equiv r_t + \beta_t z_t$, and $\kappa_t \equiv (1-\beta_t)(1-\lambda_t)$. Now consider the identity $\mathbb{E}_b[\bar{\phi}_t \mid s_t, a_t] = \mathbb{E}_\pi[\phi_t \mid s_t, a_t]$ as we expand the term $\mathbb{E}_b\left[g_t^{\lambda\beta\rho} \mid s_t, a_t\right]$, thus we have

$$\begin{aligned}
&\mathbb{E}_b\left[g_t^{\lambda\beta\rho} \mid s_t, a_t\right] \\
&= \mathbb{E}_b\left[\zeta_{t+1} + \kappa_{t+1}\theta^\top\bar{\phi}_{t+1} \mid s_t, a_t\right] \\
&\quad + \mathbb{E}_b\left[(1-\beta_{t+1})\lambda_{t+1}\rho_{t+1}g_{t+1}^{\lambda\beta\rho} \mid s_t, a_t\right] \\
&= \mathbb{E}_\pi\left[\zeta_{t+1} + \kappa_{t+1}\theta^\top\phi_{t+1} | s_t, a_t\right] \\
&\quad + \mathbb{E}_b\left[(1-\beta_{t+1})\lambda_{t+1}\rho_{t+1}g_{t+1}^{\lambda\beta\rho} \mid s_t, a_t\right] \\
&= \mathbb{E}_\pi\left[\zeta_{t+1} + \kappa_{t+1}\theta^\top\phi_{t+1} | s_t, a_t\right] \\
&\quad + \sum_{s_{t+1}} \mathbb{P}(s_{t+1}|s_t, a_t) \sum_{a_{t+1}} b(s_{t+1}, a_{t+1}) \frac{\pi(s_{t+1}, a_{t+1})}{b(s_{t+1}, a_{t+1})} \\
&\quad \times (1-\beta_{t+1})\lambda_{t+1}\mathbb{E}_b\left[g_{t+1}^{\lambda\beta\rho} | s_{t+1}, a_{t+1}\right] \\
&= \mathbb{E}_\pi\left[\zeta_{t+1} + \kappa_{t+1}\theta^\top\phi_{t+1} | s_t, a_t\right] \\
&\quad + \sum_{s_{t+1}} \mathbb{P}(s_{t+1}|s_t, a_t) \sum_{a_{t+1}} \pi(s_{t+1}, a_{t+1}) \\
&\quad \times (1-\beta_{t+1})\lambda_{t+1}\mathbb{E}_b\left[g_{t+1}^{\lambda\beta\rho} | s_{t+1}, a_{t+1}\right] \\
&= \mathbb{E}_\pi\left[\zeta_{t+1} + \kappa_{t+1}\theta^\top\phi_{t+1} | s_t, a_t\right] \\
&\quad + \mathbb{E}_\pi\left[(1-\beta_{t+1})\lambda_{t+1}\mathbb{E}_b\left[g_{t+1}^{\lambda\beta\rho} | s_{t+1}, a_{t+1}\right] \mid s_t, a_t\right],
\end{aligned}$$

which as it continues to roll out, and as a result, gives us $\mathbb{E}_b\left[g_t^{\lambda\beta\rho} \mid s_t, a_t\right] = \mathbb{E}_\pi\left[g_t^{\lambda\beta} \mid s_t, a_t\right]$. From definitions of $\delta_t^{\lambda\beta}$ and $\delta_t^{\lambda\beta\rho}$, it is immediate that $\mathbb{E}_\pi\left[\delta_t^{\lambda\beta}\phi_t\right] = \mathbb{E}_b\left[\delta_t^{\lambda\beta\rho}\phi_t\right]$. □

Thus, the objective function $J(\theta)$ in Equation (10) can be written in the following form

$$J(\theta) = \mathbb{E}_b\left[\delta^{\lambda\beta\rho}\phi\right]^\top \mathbb{E}_b\left[\phi\phi^\top\right]^{-1} \mathbb{E}_b\left[\delta^{\lambda\beta\rho}\phi\right], \tag{14}$$

in which the expectations are conditioned over behavioral policy.

Derivation of GQ(λ) algorithm: forward-view/backward-view analysis

We derive GQ(λ) algorithm based on gradient-descent in the $J(\theta)$ objective function (14). Thus, we update the modifiable parameter θ proportional to $-\frac{1}{2}\nabla J(\theta)$. Note that all the gradients in this paper are with respect to the main weight vector θ, and so are denoted simply by ∇, thus we have

$$\begin{aligned}
&-\frac{1}{2}\nabla J(\theta) \\
&= -\frac{1}{2}\nabla\left(\mathbb{E}_b\left[\delta^{\lambda\beta\rho}\phi\right]^\top \mathbb{E}_b\left[\phi\phi^\top\right]^{-1} \mathbb{E}_b\left[\delta^{\lambda\beta\rho}\phi\right]\right) \\
&= -\left(\nabla\mathbb{E}_b\left[\delta^{\lambda\beta\rho}\phi\right]\right)^\top \mathbb{E}_b\left[\phi\phi^\top\right]^{-1} \mathbb{E}_b\left[\delta^{\lambda\beta\rho}\phi\right] \\
&= -\nabla\mathbb{E}_b\left[\delta^{\lambda\beta\rho}\phi^\top\right] \mathbb{E}_b\left[\phi\phi^\top\right]^{-1} \mathbb{E}_b\left[\delta^{\lambda\beta\rho}\phi\right] \\
&= -\mathbb{E}_b\left[(\nabla\delta^{\lambda\beta\rho})\phi^\top\right] \mathbb{E}_b\left[\phi\phi^\top\right]^{-1} \mathbb{E}_b\left[\delta^{\lambda\beta\rho}\phi\right] \\
&= -\mathbb{E}_b\left[\left(\nabla g^{\lambda\beta\rho} - \phi\right)\phi^\top\right] \mathbb{E}_b\left[\phi\phi^\top\right]^{-1} \mathbb{E}_b\left[\delta^{\lambda\beta\rho}\phi\right] \\
&= \left(\mathbb{E}_b\left[\phi\phi^\top\right] - \mathbb{E}_b\left[\nabla g^{\lambda\beta\rho}\phi^\top\right]\right) \mathbb{E}_b\left[\phi\phi^\top\right]^{-1} \mathbb{E}_b\left[\delta^{\lambda\beta\rho}\phi\right] \\
&= \mathbb{E}_b\left[\delta^{\lambda\beta\rho}\phi\right] - \mathbb{E}_b\left[\nabla g^{\lambda\beta\rho}\phi^\top\right] \mathbb{E}_b\left[\phi\phi^\top\right]^{-1} \mathbb{E}_b\left[\delta^{\lambda\beta\rho}\phi\right] \\
&\approx \mathbb{E}_b\left[\delta^{\lambda\beta\rho}\phi\right] - \mathbb{E}_b\left[\nabla g^{\lambda\beta\rho}\phi^\top\right] w,
\end{aligned} \tag{15}$$

where, in the final expression, we assume that we have a quasi-stationary estimate $w \in \Re^n$ such that

$$w \approx \mathbb{E}_b\left[\phi\phi^\top\right]^{-1} \mathbb{E}_b\left[\delta^{\lambda\beta\rho}\phi\right]. \tag{16}$$

Because the expectations in the final expression (15) are not known, to update the modifiable parameter θ, we use stochastic gradient-descent approach; that is, we sample from the final expression (15) and update θ along this sample direction, where it yields the following forward-view algorithm:

$$\theta_{t+1} = \theta_t + \alpha_{\theta,t}\left(\delta_t^{\lambda\beta\rho}\phi_t - \nabla g_t^{\lambda\beta\rho}\phi_t^\top w_t\right), \tag{17}$$

where $\alpha_{\theta,t}$ is a sequence of positive step-size parameters. The desired approximation for w (16), is the solution to a least-squares problem, which can be found incrementally with linear complexity by the LMS algorithm that uses $\delta_t^{\lambda\rho}$ as its target. The standard algorithm for doing this is the following forward-view algorithm

$$w_{t+1} = w_t + \alpha_{w,t}\left(\delta_t^{\lambda\beta\rho} - w_t^\top\phi_t\right)\phi_t, \tag{18}$$

where $\alpha_{w,t}$ is another sequence of positive step-size parameters. Note that w fixed-point in the above expression is $\mathbb{E}_b\left[\phi\phi^\top\right]^{-1} \mathbb{E}_b\left[\delta^{\lambda\beta\rho}\phi\right]$.

We now turn to converting these forward-view algorithms to backward-view forms that are more convenient for low-memory mechanistic implementation. For the first term in equation (17); that is, $\delta_t^{\lambda\beta\rho}\phi_t$, which is called forward-view version of TD update, we can substitute $\delta_t e_t$ (backward-view TD update), just as in conventional TD(λ) algorithm (Sutton and Barto ,1998). This has been shown in the following theorem:

Theorem 2. *Equivalence of TD forward-view and backward-view. The forward-view description of TD update is equivalent to the mechanistic backward-view; that is,*

$$\mathbb{E}_b\Big[\delta_t^{\lambda\beta\rho}\phi_t\Big] = \mathbb{E}_b[\delta_t e_t]\,, \tag{19}$$

where $\delta_t^{\lambda\beta\rho}$ is multi-step TD error, δ_t is one-step TD error and e_t denotes eligibility trace defined in equations (11, 4,5) respectively.

Proof. We start by finding a recursive way of writing the multi-step off-policy TD error. Let $\zeta_t = r_t + \beta_t z_t$, then

$$\begin{aligned}
&\delta_t^{\lambda\beta\rho}\\
&= g_t^{\lambda\beta\rho} - \theta_t^\top\phi_t\\
&= \zeta_{t+1} + (1-\beta_{t+1})\Big[(1-\lambda_{t+1})\theta_t^\top\bar{\phi}_{t+1}\\
&\quad +\lambda_{t+1}\rho_{t+1}g_{t+1}^{\lambda\rho}\Big] - \theta_t^\top\phi_t\\
&= \zeta_{t+1} + (1-\beta_{t+1})\theta_t^\top\bar{\phi}_{t+1} - \theta_t^\top\phi_t\\
&\quad -(1-\beta_{t+1})\lambda_{t+1}\theta_t^\top\bar{\phi}_{t+1} + (1-\beta_{t+1})\lambda_{t+1}\rho_{t+1}g_{t+1}^{\lambda\beta\rho}\\
&= \delta_t\\
&\quad -(1-\beta_{t+1})\lambda_{t+1}\theta_t^\top\bar{\phi}_{t+1} + (1-\beta_{t+1})\lambda_{t+1}\rho_{t+1}g_{t+1}^{\lambda\beta\rho}\\
&\quad +(1-\beta_{t+1})\lambda_{t+1}\left(-\rho_{t+1}\theta_t^\top\phi_{t+1} + \rho_{t+1}\theta_t^\top\phi_{t+1}\right)\\
&= \delta_t + (1-\beta_{t+1})\lambda_{t+1}\rho_{t+1}\left(g_{t+1}^{\lambda\beta\rho} - \theta_t^\top\phi_{t+1}\right)\\
&\quad +(1-\beta_{t+1})\lambda_{t+1}\theta_t^\top\left(\rho_{t+1}\phi_{t+1} - \bar{\phi}_{t+1}\right)\\
&= \delta_t + (1-\beta_{t+1})\lambda_{t+1}\rho_{t+1}\delta_{t+1}^{\lambda\beta\rho}\\
&\quad +(1-\beta_{t+1})\lambda_{t+1}\theta_t^\top\left(\rho_{t+1}\phi_{t+1} - \bar{\phi}_{t+1}\right).
\end{aligned}$$

Note that the last part of the above equation has expected value of vector zero under the behavior policy because

$$\begin{aligned}
\mathbb{E}_b[\rho_t\phi_t \mid s_t] &= \sum_a b(s_t,a)\frac{\pi(s_t,a)}{b(s_t,a)}\phi(s_t,a)\\
&= \sum_a \pi(s_t,a)\phi(s_t,a) \equiv \bar{\phi}_t.
\end{aligned}$$

Putting all these together, we can write the TD update (in expectation) in a simple way in terms of eligibility traces which leads to backward-view:

$$\begin{aligned}
&\mathbb{E}_b\Big[\delta_t^{\lambda\beta\rho}\phi_t\Big]\\
&= \mathbb{E}_b\Big[\Big(\delta_t + (1-\beta_{t+1})\lambda_{t+1}\rho_{t+1}\delta_{t+1}^{\lambda\beta\rho}\Big)\phi_t\Big]\\
&\quad +\mathbb{E}_b\big[(1-\beta_{t+1})\lambda_{t+1}\theta^\top\left(\rho_{t+1}\phi_{t+1} - \bar{\phi}_{t+1}\right)\phi_t\big]\\
&= \mathbb{E}_b[\delta_t\phi_t] + \mathbb{E}_b\Big[(1-\beta_{t+1})\lambda_{t+1}\rho_{t+1}\delta_{t+1}^{\lambda\beta\rho}\phi_t\Big] + 0\\
&= \mathbb{E}_b[\delta_t\phi_t] + \mathbb{E}_b\Big[(1-\beta_t)\lambda_t\rho_t\delta_t^{\lambda\beta\rho}\phi_{t-1}\Big]\\
&= \mathbb{E}_b[\delta_t\phi_t] + \mathbb{E}_b[(1-\beta_t)\lambda_t\rho_t\Big(\delta_t + (1-\beta_{t+1})\lambda_{t+1}\rho_{t+1}\\
&\quad \times\delta_{t+1}^{\lambda\beta\rho} + (1-\beta_{t+1})\lambda_{t+1}\theta^\top\left(\rho_{t+1}\phi_{t+1} - \bar{\phi}_{t+1}\right)\Big)\phi_{t-1}]\\
&= \mathbb{E}_b[\delta_t\phi_t] + \mathbb{E}_b[(1-\beta_t)\lambda_t\rho_t\delta_t\phi_{t-1}]\\
&\quad +\mathbb{E}_b\Big[(1-\beta_t)\lambda_t\rho_t(1-\beta_{t+1})\lambda_{t+1}\rho_{t+1}\delta_{t+1}^{\lambda\beta\rho}\phi_{t-1}\Big] + 0\\
&= \mathbb{E}_b[\delta_t\left(\phi_t + (1-\beta_t)\lambda_t\rho_t\phi_{t-1}\right)]\\
&\quad +\mathbb{E}_b\Big[(1-\beta_{t-1})\lambda_{t-1}\rho_{t-1}(1-\beta_t)\lambda_t\rho_t\delta_t^{\lambda\beta\rho}\phi_{t-2}\Big]\\
&\vdots\\
&= \mathbb{E}_b\Big[\delta_t\Big(\phi_t + (1-\beta_t)\lambda_t\rho_t\phi_{t-1}\\
&\quad +(1-\beta_t)\lambda_t\rho_t(1-\beta_{t-1})\lambda_{t-1}\rho_{t-1}\phi_{t-2} + \cdots\Big)\Big]\\
&= \mathbb{E}_b[\delta_t e_t]\,,
\end{aligned} \tag{20}$$

where $e_t = \phi_t + (1-\beta_t)\lambda_t\rho_t e_{t-1}$, which gives us a backward view algorithm for the TD(λ) update. □

For the second term of the gradient update (15), we can use the following trick: we take the gradient of the forward-backward relationship just established in theorem 2; that is, $\nabla\mathbb{E}_b\Big[\delta_t^{\lambda\beta\rho}\phi_t\Big] = \nabla\mathbb{E}_b[\delta_t e_t]$, then $\mathbb{E}_b\Big[\nabla\delta_t^{\lambda\beta\rho}\phi_t^\top\Big] = \mathbb{E}_b\big[\nabla\delta_t e_t^\top\big]$, and consequently we get, $\mathbb{E}_b\Big[\nabla g_t^{\lambda\beta\rho}\phi_t^\top\Big] - \mathbb{E}_b\big[\phi_t\phi_t^\top\big] = \mathbb{E}_b\big[\big((1-\beta_{t+1})\bar{\phi}_{t+1} - \phi_t\big)e_t^\top\big]$. By arranging the terms and using Equation (4), and $\mathbb{E}_b[\rho_t\phi_t \mid s_t = s] = \bar{\phi}_t$, we get

$$\begin{aligned}
&\mathbb{E}_b\Big[\nabla g_t^{\lambda\beta\rho}\phi_t^\top\Big]\\
&= \mathbb{E}_b\big[\phi_t\phi_t^\top\big] + \mathbb{E}_b\big[(1-\beta_{t+1})\bar{\phi}_{t+1}e_t^\top\big] - \mathbb{E}_b\big[\phi_t e_t^\top\big]\\
&= \mathbb{E}_b\big[\phi_t\phi_t^\top\big] + \mathbb{E}_b\big[(1-\beta_{t+1})\bar{\phi}_{t+1}e_t^\top\big]\\
&\quad -\mathbb{E}_b\Big[\phi_t\left(\phi_t + (1-\beta_t)\lambda_t\rho_t e_{t-1}\right)^\top\Big]\\
&= \mathbb{E}_b\big[(1-\beta_{t+1})\bar{\phi}_{t+1}e_t^\top\big] - \mathbb{E}_b\big[(1-\beta_t)\lambda_t\rho_t\phi_t e_{t-1}^\top\big]\\
&= \mathbb{E}_b\big[(1-\beta_{t+1})\bar{\phi}_{t+1}e_t^\top\big] - \mathbb{E}_b\big[(1-\beta_t)\lambda_t\bar{\phi}_t e_{t-1}^\top\big]\\
&= \mathbb{E}_b\big[(1-\beta_{t+1})\bar{\phi}_{t+1}e_t^\top\big] - \mathbb{E}_b\big[(1-\beta_{t+1})\lambda_{t+1}\bar{\phi}_{t+1}e_t^\top\big]\\
&= \mathbb{E}_b\big[(1-\beta_{t+1})(1-\lambda_{t+1})\bar{\phi}_{t+1}e_t^\top\big]\,.
\end{aligned} \tag{21}$$

Returning now to the forward-view equation for updating θ (17), it should be clear that for the first term

we can substitute $\delta_t e_t$, based on (19), just as in conventional TD(λ), and for the second term we can substitute based on (21), thus the backward-view update is as follows:

$$\theta_{t+1} = \theta_t + \alpha_{\theta,t}\Big[\delta_t e_t - \kappa_{t+1}(e_t^\top w_t)\bar{\phi}_{t+1}\Big], \qquad (22)$$

where $\kappa_t = (1-\beta_t)(1-\lambda_t)$. The forward-view algorithm for w, (18), is particularly simple to convert to a backward-view form. The first term is again the same as the conventional linear TD(λ) update, and the second term is already in a suitable mechanistic form. The simplest backward-view update is

$$w_{t+1} = w_t + \alpha_{w,t}\Big[\delta_t e_t - (w_t^\top \phi_t)\phi_t\Big]. \qquad (23)$$

Convergence of GQ(λ)

In this section, we show that GQ(λ) converges with probability one to the TD(λ) fixed-point under standard assumptions. The TD(λ) fixed-point, θ^*, is a point which satisfies in

$$0 = \mathbb{E}_b[\delta_t e_t] = -A\theta^* + b, \qquad (24)$$

where

$$A = \mathbb{E}_b\Big[e_t\left(\phi_t - (1-\beta_{t+1})\bar{\phi}_{t+1}\right)^\top\Big], \qquad (25)$$

$$b = \mathbb{E}_b[(r_{t+1} + \beta_{t+1} z_{t+1})\, e_t]. \qquad (26)$$

Theorem 3. *Convergence of GQ(λ). Consider the GQ(λ) iterations (2,3,4) with step-size sequences $\alpha_{\theta,t}$ and $\alpha_{w,t}$ satisfing $\alpha_{\theta,t}, \alpha_{w,t} > 0$, $\sum_{t=0}^{\infty}\alpha_{\theta,t} = \sum_{t=0}^{\infty}\alpha_{w,t} = \infty$, $\sum_{t=0}^{\infty}\alpha_{\theta,t}^2$, $\sum_{t=0}^{\infty}\alpha_{w,t}^2 < \infty$ and that $\frac{\alpha_{\theta,t}}{\alpha_{w,t}} \to 0$ as $t \to \infty$. Further assume that ϕ_t is a Markov process with a unique invariant distribution and that the ϕ_t, e_t, z_t, and r_t sequences have uniformly bounded second moments. Assume that A (25) and $C = \mathbb{E}_b\big[\phi_t\phi_t^\top\big]$ are non-singular matrices. Then the parameter vector θ_t converges with probability one to the TD(λ) fixed-point θ^* (24).*

Proof. We use Lemma 6.7 (Bertsekas and Tsitsiklis 1996) that can be applied here and follow the proof of convergence for the TDC algorithm in Sutton et al. (2009b). For the brevity, we have omitted the proof. □

Conclusion

The GQ(λ) algorithm, which has been introduced in this paper, incorporates varying eligibility traces and option-conditional probabilities for policy evaluation. To derive GQ(λ), we carried out a forward-view/backward-view analysis. We extended the existing convergence results to show that GQ(λ) is guaranteed to converge to the TD(λ) fixed-point. GQ(λ) is a general gradient TD method for off-policy learning and as such can be seen as extension of Q-learning. GQ(λ) is able to learn about temporally abstract predictions, which makes it suitable to use for learning experientially grounded knowledge. In addition, GQ(λ) is online, incremental and its computational complexity scales only linearly with the size of features. Thus, it is suitable for large-scale applications. Our work, however, is limited to policy evaluation. Interesting future works is to extend GQ(λ) for control problems and gather extensive empirical data on large-scale real-world applications.

References

Baird, L. C. (1995). Residual algorithms: Reinforcement learning with function approximation. In *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 30–37. Morgan Kaufmann.

Bertsekas, D. P., Tsitsiklis, J. (1996). *Neuro-Dynamic Programming.* Athena Scientific, 1996.

Maei, H. R., Szepesvári, Cs, Bhatnagar, S., Precup, D., Silver D., Sutton, R. S. (2009). Convergent Temporal-Difference Learning with Arbitrary Smooth Function Approximation. In *Accepted in Advances in Neural Information Processing Systems 22.* MIT Press.

Sutton, R. S. (1988). Learning to predict by the method of temporal differences. *Machine Learning 3*:9–44.

Sutton, R. S., Barto, A. G. (1998). *Reinforcement Learning: An Introduction.* MIT Press.

Sutton, R. S., Precup, D., Singh, S. (1998). Intra-option learning about temporally abstract actions. *Proceedings of the 15th International Conference on Machine Learning*, pp. 556564.

Sutton, R. S., Szepesvári, Cs., Maei, H. R. (2009a). A convergent O(n) algorithm for off-policy temporal-difference learning with linear function approximation. In *Advances in Neural Information Processing Systems 21.* MIT Press.

Sutton, R. S., Maei, H. R, Precup, D., Bhatnagar, S., Silver, D., Szepesvári, Cs., Wiewiora, E. (2009b). Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th International Conference on Machine Learning, Montreal, Canada.*

Sutton, R. S. (2009). The grand challenge of predictive empirical abstract knowledge. *Working Notes of the IJCAI-09 Workshop on Grand Challenges for Reasoning from Experiences.*

Tsitsiklis, J. N., and Van Roy, B. (1997). An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control 42*:674–690.

Watkins, C. J. C. H., and Dayan, P. (1992). Q-learning. *Machine Learning*, 8:279–292.