Using Policy Gradients to Account for Changes in Behaviour Policies under Off-policy Control

Lucas Lehnert LUCAS.LEHNERT@MAIL.MCGILL.CA
School of Computer Science
McGill University

Doina Precup DPRECUP@CS.MCGILL.CA
School of Computer Science
McGill University

Abstract

Off-policy learning refers to the problem of learning the value function of a behaviour, or policy, while selecting actions with a different policy. Gradient-based off-policy learning algorithms, such as GTD (Sutton et al., 2009b) and TDC/GQ (Sutton et al., 2009a), converge when selecting actions with a fixed policy even when using function approximation and incremental updates. In control problems, the behaviour policy is adapted over time. One key challenge in off-policy control is that adapting the policy results in changing the distribution of subsequent transitions the algorithm will see. We present the first off-policy gradient-based learning algorithm that accounts for how an adjustment of the policy at the current time step effects the distribution of future transition samples. We derive the algorithm in the style of policy gradients and show that our method performs favourably to existing approaches when used for off-policy control with linear function approximation.

Keywords: Reinforcement Learning, Off-policy Control, Linear Function approximation

For Reinforcement Learning (RL) Sutton and Barto (1998) provide algorithms that can learn the optimal control strategy in an unknown, stochastic environment, based on sampled transitions. Off-policy learning refers to the important case when these samples do not come from the behaviour of interest, but from some other sampling strategy. For example, Q-learning (Watkins and Dayan, 1992) aims to compute the optimal value achievable in an MDP, but actions are picked from an ε-greedy policy which is greedy with respect to the current value estimates. This is a typical practical case: the behaviour used to generate samples for learning control is not fixed, but depends on current value estimates.

In this paper, we tackle the problem of designing incremental off-policy control algorithms that can account for the interaction between the value estimates and the distribution of transitions the algorithm will sample. We build on gradient-based off-policy evaluation algorithms such as GQ(λ) (Maei and Sutton, 2010). However, GQ(λ) is designed for the prediction case and does not consider how variations in the policy effect the distribution from which transitions are sampled. Being able to account for the interaction between variations in parameter estimates and sampling distribution should stabilize learning. For example, SARSA suffers from oscillations in its value function estimates and is only guaranteed to converge to a sub-space of policies (Gordon, 2001, 1996).

We present a new gradient-based TD-learning algorithm similar to GQ that also incorporates policy gradients to correct for the drift in the sampling distribution. We leverage the policy gradient framework (Sutton et al., 2000) and directly analyze the interaction between the policy gradient and the sampling distribution. Conceptually, the idea is to

consider the sequence of Markov chains induced by the policy changes resulting from the change in values. This idea has been used to analyze approximate policy iteration (Perkins and Precup, 2003), but our algorithm is incremental, so the analysis needs to be more involved. The closest related approach is the off-policy actor-critic (Degris et al., 2012), in which gradient-based TD-learning methods are used to provide the critic and policy gradients are used to derive an update to the actor. We only estimate a value function, and the actor is derived from this value function.

1. Gradient Temporal Difference Methods

We consider an MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle$ where $\mathcal{S}$ is a (finite) state space, $\mathcal{A}$ a (finite) action space, $p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ a stochastic transition function, $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ a reward function, and $\gamma \in (0, 1)$ a discount factor. The behavior of an agent is described by a policy π which selects actions with probability $\pi(a|s)$ conditioned on s.

In TD-learning the prediction case assumes a fixed policy is used to generate infinite length trajectories. Q-learning tries to estimate the Q-function which predicts the cumulative reward of a trajectory that starts with a particular state and action:

$$Q^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, s_{t+1}) \middle| s_0 = s, a_0 = a \right].$$

In the control case the policy π is not held fixed and instead the agent tries to simultaneously improve its current policy estimate and estimate that policy's Q-function. We consider the case of approximating Q-functions with a linear model where we assume a *basis function* ϕ and a *parameter vector* $\theta \in \mathbb{R}^n$ and estimate θ so that $Q_\theta(s, a) = \phi_{s,a}^\top \theta \approx Q^\pi(s, a)$. Off-policy learning considers the problem of learning value estimates of a *target policy* π while sampling transitions using a different *behaviour policy* b.

Gradient-based TD-learning algorithms such as TDC and GQ are the first incremental off-policy TD-learning algorithms that remain stable if used with linear-function approximation. GQ performs two-time scale stochastic gradient descend (Borkar, 1997) on the Mean Squared Projected Bellman Error (MSPBE) objective defined as

$$\text{MSPBE}(\theta) = ||Q_\theta - \Pi T^\pi Q_\theta||_D^2. \tag{1}$$

where Q_θ is the vector listing all action-values $Q_\theta(s, a)$, and $||v||_D^2 = v^\top D v$ is a weighted norm where $D = \text{diag}\{d^{s,a}\}_{(s,a)\in\mathcal{S}\times\mathcal{A}}$ is a matrix. The Bellman operator is defined as $T^\pi v \stackrel{\text{def.}}{=} R + \gamma P^\pi v$, and the projection operator is defined as $\Pi \stackrel{\text{def.}}{=} \Phi(\Phi^\top D \Phi)^\top \Phi^\top D$. This projection operator projects the one-step lookahead $T^\pi Q_\theta$ back into the space of all representable value functions (Sutton et al., 2009a). Hence finding the parameter vector θ which minimizes MSPBE(θ) corresponds to finding the best approximation to the true value function Q^π.

One key innovation of the MSPBE objective is the use of the weight matrix D which applies a weight to every state-action pair (s, a). These weights are interpreted as the probability with which a state-action pair is visited and are called the *stationary distribution*. This distribution is invariant of the time step t and the start state of the trajectory. This use of the stationary distribution in conjunction with the projection operator stabilizes gradient

TD-learning under off-policy training when used with linear function approximation. In contrast, Precup et al. (2001) use importance sampling ratios to keep track of the probability of visiting a state, but this method suffers from high variance.

Further, the Bellman operator T^π as well as the stationary distribution both depend on the policy π. In the control case the action selection probability $\pi_\theta(a|s)$ is typically calculated using the action-values $Q_\theta(s,a)$, for example with a *Boltzmann policy* with action-selection probabilities $\pi_\theta(a|s) = \exp(\theta^\top \phi_{s,a}/\tau)/\sum_b \exp(\theta^\top \phi_{s,b}/\tau)$. As $\pi_\theta(a|s)$ depends on the current parameter vector θ, an update to θ changes the policy and with that the Bellman operator and the stationary distribution. The derivation of GQ(λ) does not consider this dependency and as a result the algorithm drifts on a non-stationary Markov chain.

2. Proposed approach

Our approach is to view the policy π as a differentiable operator applied to the Q-function.

Assumption 1 *The policy π_θ has action selection probabilities $\pi_\theta(a|s)$ that are differentiable with respect to $Q_\theta(s,a) = \phi_{s,a}^\top \theta$. Since Q_θ is linear in θ, Q_θ is differentiable and thus $\pi_\theta(a|s)$ is assumed to be continuously differentiable at all $\theta \in \mathbb{R}^n$.*

Using this assumption we parametrize the Bellman operator in θ and write

$$\text{MSPBE}(\theta) = ||Q_\theta - \Pi_\theta T_\theta Q_\theta||^2_{D_\theta}, \quad T_\theta v \overset{\text{def.}}{=} R + \gamma P_\theta v. \tag{2}$$

In the matrix P_θ the entry $P_\theta^{(s',a'),(s,a)} = p(s,a,s')\pi_\theta(a'|s')$. Since the policy depends on θ, the matrix P_θ and the Bellman operator T_θ depend on θ as well. Further we assume a steady stationary distribution $d(s)$ over the state space (similar to GQ) and define

$$d^{s,a} \overset{\text{def.}}{=} d(s)\pi_\theta(a|s), \; D = \text{diag}\{d^{s,a}\}_{(s,a)\in\mathcal{S}\times\mathcal{A}}. \tag{3}$$

Assuming a fixed stationary distribution $d(s)$ can be interpreted as averaging the probability of reaching a state in t time steps over all time steps. This assumption is also a limitation of PGQ as changing the control policy also effects $d(s)$. By considering $d(s)$ as steady, PGQ does not correct the distribution of the trajectory it has visited so far. At the time of writing we are not aware of any algorithm that has this ability. Under these assumptions, Lemma 1 states a new MSPBE gradient. The proof can be found in Appendix A.

Lemma 1 (PGQ Gradient Lemma) *For a finite state-action MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle$ and a basis function $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^n$, the gradient of the MSPBE objective with respect to the parameter vector θ is*

$$\begin{aligned} -\frac{1}{2}\nabla_\theta MSPBE(\theta) = {} & \mathbb{E}_{\pi_\theta}\left[\delta\phi_{s,a}\right] - \mathbb{E}_{\pi_\theta}\left[\gamma \mathbb{E}_{\pi_\theta}\left[\phi_{s',.}\right]\phi^\top\right] w - \mathbb{E}_{\pi_\theta}\left[\frac{\nabla_\theta \pi_\theta(a|s)}{\pi_\theta(a|s)}\delta\phi_{s,a}^\top\right] w \\ & + \frac{1}{2}\mathbb{E}_{\pi_\theta}\left[\frac{\nabla_\theta \pi_\theta(a|s)}{\pi_\theta(a|s)}(\phi_{s,a}^\top w)^2\right] - \gamma \mathbb{E}_{\pi_\theta}\left[\mathbb{E}_{\pi_\theta}\left[\frac{\nabla_\theta \pi_\theta(\cdot|s')}{\pi_\theta(\cdot|s')}\phi_{s',.}^\top \theta\right]\phi_{s,a}^\top\right] w, \end{aligned} \tag{4}$$

where $w = \mathbb{E}_{\pi_\theta}[\phi_{s,a}\phi_{s,a}^\top]^{-1}\mathbb{E}_{\pi_\theta}\left[\delta\phi_{s,a}\right]$ is the auxiliary weight vector,
$\delta = r(s,a,s') + \gamma \mathbb{E}_{\pi_\theta}[\phi_{s',.}^\top]\theta - \phi_{s,a}^\top\theta$ is the TD-error, $\mathbb{E}_{\pi_\theta}\left[\phi_{s',.}\right] = \sum_a \pi_\theta(a|s')\phi_{s',a}$, and
$\mathbb{E}_{\pi_\theta}\left[\frac{\nabla_\theta \pi_\theta(\cdot|s')}{\pi_\theta(\cdot|s')}\phi_{s',.}^\top\theta\right] = \sum_{s',a'} p(s,a,s')\pi_\theta(a'|s')\frac{\nabla_\theta \pi_\theta(a'|s')}{\pi_\theta(a'|s')}\phi_{s',a'}^\top\theta$.

2.1 Off-policy Conversion

The expectations stated in Lemma 1 are with respect to the target policy π_θ. However, in off-policy learning transition data is generated by using a different behaviour policy b. Hence we have to re-express (17) in terms of the behaviour policy. We correct for the difference in distribution between π and b using importance sampling ratios $\rho = \frac{\pi_\theta(a|s)}{b(a|s)}$, but we do not correct the stationary distribution over states. This is one key innovation of gradient based TD-learning algorithms (Sutton et al., 2009b): Sampling states form a stationary distribution with respect to b stabilizes learning with linear function approximation. Lemma 2 states the off-policy version of the MSPBE gradient, its proof can be found in Appendix B.

Lemma 2 (Off-policy PGQ Gradient Lemma) *For a finite state-action MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle$ and a basis function ϕ, the gradient of the MSPBE objective with respect to the parameter vector θ is*

$$
\begin{aligned}
-\frac{1}{2}\nabla_\theta MSPBE(\theta) = \mathbb{E}_b\left[\rho\delta\phi_{s,a}\right] - \mathbb{E}_b\left[\gamma\rho\mathbb{E}_{\pi_\theta}\left[\phi_{s',\cdot}\right]\phi^\top\right] w - \mathbb{E}_b\left[\rho\frac{\nabla_\theta\pi_\theta(a|s)}{\pi_\theta(a|s)}\delta\phi_{s,a}^\top\right] w \\
+ \frac{1}{2}\mathbb{E}_b\left[\rho\frac{\nabla_\theta\pi_\theta(a|s)}{\pi_\theta(a|s)}(\phi_{s,a}^\top w)^2\right] - \gamma\mathbb{E}_b\left[\rho\mathbb{E}_{\pi_\theta}\left[\frac{\nabla_\theta\pi_\theta(\cdot|s')}{\pi_\theta(\cdot|s')}\phi_{s',\cdot}^\top\theta\right]\phi_{s,a}^\top\right] w \qquad (5)
\end{aligned}
$$

where $w = \mathbb{E}_b[\rho\phi_{s,a}\phi_{s,a}^\top]^{-1}\mathbb{E}_b\left[\rho\delta\phi_{s,a}\right]$. The remaining terms are defined as in Lemma 1.

2.2 Sampling the PGQ Gradient

To obtain an incremental online learning algorithm we have to sample the gradient in Lemma 2. Similar to Sutton et al. (2009b) we update an auxiliary weight vector with

$$
w_{t+1} = w_t + \beta_t\rho_t(\delta_t - \phi_{s,a}^\top w_t)\phi_{s,a}. \qquad (6)
$$

Sampling (5) gives the update rule

$$
\begin{aligned}
\theta_{t+1} = \theta_t + \alpha_t\Big[\rho_t\delta_t\phi_{s,a} - \rho_t\gamma\mathbb{E}_{\pi_\theta}\left[\phi_{s',\cdot}\right]\phi_{s,a}^\top w_t\Big] \\
\underline{- \alpha_t\rho_t\frac{\nabla_\theta\pi_{\theta_t}(a|s)}{\pi_{\theta_t}(a|s)}\left[\delta_t\phi_{s,a}^\top w_t - \frac{1}{2}(w_t^\top\phi_{s,a})^2\right] - \alpha_t\rho_t\mathbb{E}_{\pi_\theta}\left[\frac{\nabla_\theta\pi_\theta(\cdot|s')}{\pi_\theta(\cdot|s')}\phi_{s',\cdot}^\top\theta\right]\phi_{s,a}^\top w_t}. \qquad (7)
\end{aligned}
$$

This rule is quite intuitive as it is the same as GQ(0) minus the underlined policy gradient correction term. Similar to Expected SARSA (Sutton and Barto, 1998, Exercise 6.10) we calculate the expectation across all actions at the next state s' analytically. The update rules (7) and (6) define the new algorithm, which we call *Policy-Gradient Q-learning (PGQ)*.

3. Experiments

In this section, we compare the PGQ algorithm to Q-learning and GQ on three standard off-policy control domains: Baird's counterexample, Mountain Car, and Acrobot. The behaviour and target policies are always Boltzmann policies.

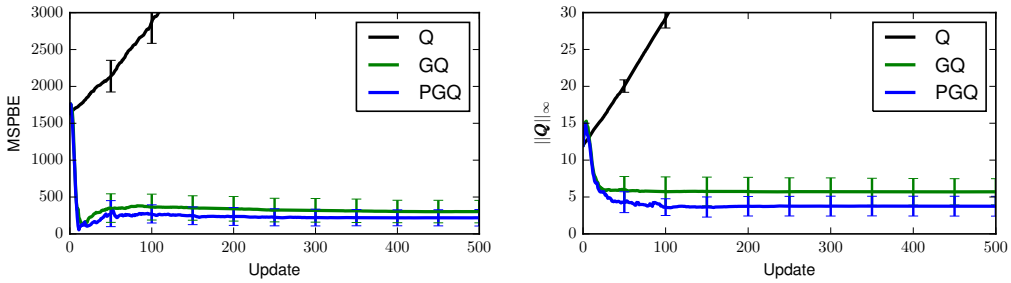

Figure 1: MSPBE and the highest Q-value (i.e. $||\boldsymbol{Q}||_\infty$) for off-policy training averaged over 20 runs. The control and target policy temperatures are $\tau = 10$ and $\tau = 0.2$ respectively. Q-learning uses $\alpha = 0.01$ while GQ and PGQ use $\alpha = 0.1$ and $\beta = 0.1$.

3.1 Baird's Counter Example

We tested the three algorithms on the 7 state "star" Baird counter example (Baird, 1995) for which divergence of Q-learning is monotonic. The parameter vector θ corresponding to the action that transitions to the 7th centre state is initialized with $(1, 1, 1, 1, 1, 1, 1, 10)$ and the remaining parameter entries are set to 1. In contrast to Maei et al. (2010) we do not fix the control or target policies but use Boltzmann action selection distributions computed directly from the action-value function. Updating was done with dynamic programming sweeps. Figure 1 shows that PGQ converges to a solution that has an error slightly lower than that of GQ suggesting that PGQ updates its estimates more efficiently.

3.2 Mountain Car

In the Mountain car task (Sutton and Barto, 1998) the agent has to drive an underpowered car out of a valley up a hill. The goal of the task is to reach the top of the hill by accelerating forward, backward, or coasting. The state space consists of the car's position and velocity which we tile into a 18×18 grid. The target temperature is set to 0.5 and the behavior temperature is set to 1.1. Figure 2 shows that the average episode length of GQ increases after approximately 50 episodes, i.e. GQ does not seem to converge to a good solution and instead finds a policy whose performance becomes worse over time. However, Q-learning and PGQ both find efficient policies with Q-learning performing significantly better than PGQ after approximately 150 episodes. Further the plot of the θ norms reveal that all three algorithms seem to converge to different solutions.

3.3 Acrobot

The Acrobot task (Sutton and Barto, 1998) consist of a two-link under-actuated robot arm mounted at the top end. Torque can be applied in two directions on the middle-joint or no torque can be applied by the agent. The state space contains four continuous variables, the angle and angular velocity of the top and middle joint respectively. We tile this space into 12 tiles along the angular positions and 14 tiles along the angular velocities. The goal of the agent is to learn how to apply torque to the middle joint to move the tip of the arm to

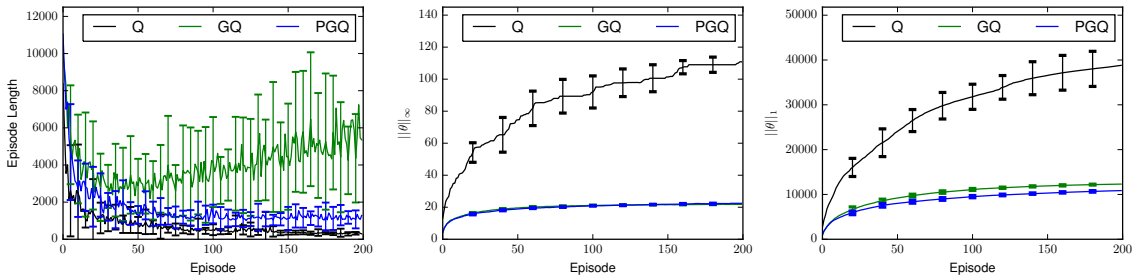

Figure 2: Episode length and the infinity and L1 norm of θ on the Mountain car task averaged over 20 repeats. Q-learning uses $\alpha = 0.5$, GQ and PGQ use $\alpha = 0.1$ and $\beta = 0.005$.

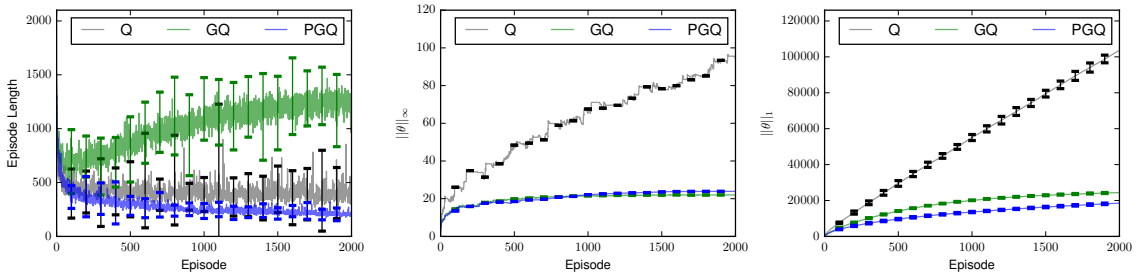

Figure 3: Episode length and the infinity and L1 norm of θ on the Acrobot task averaged over 20 repeats. All algorithms use $\alpha = 0.1$, GQ and PGQ use $\beta = 0.005$.

a certain height. For this task the control temperature is 1.1 and the target temperature is 0.5. Figure 3 shows that PGQ finds an efficient policy and outperforms Q-learning and GQ. Tthe episode lengths generated by GQ first decrease and after around 150 episodes start to increase indicating that GQ does not seem to converge to a good solution. Further, the linear growth of the action values found by Q-learning suggests that Q-learning does not converge on this control problem. These results highlight that under off-policy training on complex control problems the PGQ algorithm outperforms Q-learning and GQ.

4. Conclusion

We have presented a new gradient based TD-learning algorithm that incorporates policy gradients. The resulting algorithm is similar to GQ but also has a correction term in the direction of the gradient of the target policy. Our analysis accounts for the dependency of the Markov chain from which future data is sampled on the value function parameter vector θ. This dependency is analyzed by viewing the policy merely as an operator that is applied to the action-value function. With the exception of Perkins and Precup (2003) most previous work analyses the policy as a probability distribution which is represented separately in memory and is updated incrementally. However, Perkins and Precup use the idea of viewing the policy as an operator to primarily analyze convergence of a policy iteration algorithm. This paper presents the first approach to use this idea to derive a gradient-based algorithm which simultaneously evaluates and improves a policy in the context of off-policy control.

Appendix A. Proof of the PGQ Gradient Lemma 1

In this section we provide the proof of the PGQ Gradient Lemma 1. The ith entry in a vector ϕ is denoted with ϕ^i and the entry i, j in a matrix Φ is denoted with $\Phi^{i,j}$. The jth column in a matrix Φ is denoted with $\Phi^{:,j}$. Now we will proof Lemma 1.

Proof [Proof of Lemma 1] For the gradient derivation we first rewrite the MSPBE objective as

$$\begin{aligned}\text{MSPBE}(\theta) &= ||Q_\theta - \Pi_\theta T_\theta Q_\theta||^2_{D_\theta} \\ &= \Big(\Phi^\top D_\theta (T_\theta Q_\theta - Q_\theta)\Big)^\top (\Phi^\top D_\theta \Phi)^{-1} \Big(\Phi^\top D_\theta (T_\theta Q_\theta - Q_\theta)\Big).\end{aligned}$$

Since we want to compute gradients of an equation containing matrices as intermediate results, we first focus on partial derivatives with respect to θ^i:

$$\begin{aligned}&\frac{\partial}{\partial \theta^i}\text{MSPBE}(\theta) \\ &= \frac{\partial}{\partial \theta^i}\Big[\Big(\Phi^\top D_\theta (T_\theta Q_\theta - Q_\theta)\Big)^\top (\Phi^\top D_\theta \Phi)^{-1} \Big(\Phi^\top D_\theta (T_\theta Q_\theta - Q_\theta)\Big)\Big] \\ &= 2\frac{\partial}{\partial \theta^i}\Big[\Big(\Phi^\top D_\theta (T_\theta Q_\theta - Q_\theta)\Big)^\top\Big] (\Phi^\top D_\theta \Phi)^{-1} \Big(\Phi^\top D_\theta (T_\theta Q_\theta - Q_\theta)\Big) \\ &\quad + \Big(\Phi^\top D_\theta (T_\theta Q_\theta - Q_\theta)\Big)^\top \frac{\partial}{\partial \theta^i}\Big[(\Phi^\top D_\theta \Phi)^{-1}\Big] \Big(\Phi^\top D_\theta (T_\theta Q_\theta - Q_\theta)\Big).\end{aligned}$$

For the partial derivative on the Bellman error we have

$$\begin{aligned}\frac{\partial}{\partial \theta^i}[T_\theta Q_\theta - Q_\theta] &= \frac{\partial}{\partial \theta^i}[R + \gamma P_\theta \Phi\theta - \Phi\theta] \\ &= \gamma \frac{\partial P_\theta}{\partial \theta^i}\Phi\theta + \gamma P_\theta \Phi^{:,i} - \Phi:, i. \end{aligned} \tag{8}$$

For the derivative of the inverse feature covariance we have

$$\begin{aligned}\frac{\partial}{\partial \theta^i}\Big[(\Phi^\top D\Phi)^{-1}\Big] &= -(\Phi^\top D\Phi)^{-1}\frac{\partial}{\partial \theta^i}(\Phi^\top D\Phi)(\Phi^\top D\Phi)^{-1} \\ &= -(\Phi^\top D\Phi)^{-1}(\Phi^\top \frac{\partial D}{\partial \theta^i}\Phi)(\Phi^\top D\Phi)^{-1}. \end{aligned} \tag{9}$$

Similar to Sutton et al. (2009a) we define an auxiliary weight vector

$$\begin{aligned}w &= (\Phi^\top D_\theta \Phi)^{-1}\Big(\Phi^\top D_\theta (T_\theta Q_\theta - Q_\theta)\Big) \\ &= \mathbb{E}_{\pi_\theta}[\phi_{s,a}\phi_{s,a}^\top]^{-1}\mathbb{E}_{\pi_\theta}[\delta\phi_{s,a}], \end{aligned} \tag{10}$$

Plugging (8) and (9) back into the derivative of the MSPBE gives

$$\begin{aligned}\frac{\partial}{\partial \theta^i}\text{MSPBE}(\theta) &= 2\Big(\Phi^\top \frac{\partial D_\theta}{\partial \theta^i}(T_\theta Q_\theta - Q_\theta) \\ &\quad + \Phi^\top D_\theta(\gamma \frac{\partial P_\theta}{\partial \theta^i}\Phi\theta + \gamma P_\theta \Phi^{:,i} - \Phi^{:,i})\Big)^\top w - w(\Phi^\top \frac{\partial D_\theta}{\partial \theta^i}\Phi)w \\ &= 2\left(\Phi^\top \frac{\partial D_\theta}{\partial \theta^i}(T_\theta Q_\theta - Q_\theta)\right)^\top w - w(\Phi^\top \frac{\partial D_\theta}{\partial \theta^i}\Phi)w \\ &\quad + 2\left(\Phi^\top D_\theta(\gamma \frac{\partial P_\theta}{\partial \theta^i}\Phi\theta + \gamma P_\theta \Phi^{:,i} - \Phi^{:,i})\right)^\top w. \end{aligned} \tag{11}$$

Now we will further decompose the remaining derivative terms and rewrite them in terms of expectations. For the first term we have

$$\begin{aligned}\Phi^\top \frac{\partial D_\theta}{\partial \theta^i}(T_\theta Q_\theta - Q_\theta) &= \Phi^\top \operatorname{diag}\left\{ d_s \frac{\partial \pi_\theta(a|s)}{\partial \theta^i} \right\}_{(s,a)\in\mathcal{S}\times\mathcal{A}} (R + \gamma P_\theta Q_\theta - Q_\theta) \\ &= \sum_{s,a,s'} d_s \frac{\partial \pi_\theta(a|s)}{\partial \theta^i} \phi_{s,a}^\top \delta \\ &= \mathbb{E}_{\pi_\theta}\left[\frac{\partial \pi_\theta(a|s)/\partial \theta^i}{\pi_\theta(a|s)} \delta \phi_{s,a}^\top \right], \end{aligned} \tag{12}$$

where

$$\delta = r(s,a,s') + \gamma \mathbb{E}_{\pi_\theta}\left[\phi_{s',\cdot}^\top\right]\theta - \phi_{s,a}^\top \theta$$

is the TD-error. For the second term we obtain

$$\begin{aligned}\Phi^\top \frac{\partial D}{\partial \theta^i}\Phi &= \sum_{s,a} d(s) \frac{\partial \pi_\theta(a|s)}{\partial \theta^i} \phi_{s,a}\phi_{s,a}^\top \\ &= \mathbb{E}_{\pi_\theta}\left[\frac{\frac{\partial \pi_\theta(a|s)}{\partial \theta^i}}{\pi_\theta(a|s)} \phi_{s,a}\phi_{s,a}^\top \right]. \end{aligned} \tag{13}$$

For the third matrix term we have

$$\begin{aligned}&\Phi^\top D_\theta(\gamma \frac{\partial P_\theta}{\partial \theta^i}\Phi\theta + \gamma P_\theta \Phi^{:,i} - \Phi^{:,i}) \\ &= \Phi^\top D_\theta\left(\gamma \begin{bmatrix} \sum_{s',a'} t(s_1,a_1,s') \frac{\partial \pi_\theta(a'|s')}{\partial \theta^i} \theta^\top \phi_{s',a'} \\ \vdots \\ \sum_{s',a'} t(s_n,a_m,s') \frac{\partial \pi_\theta(a'|s')}{\partial \theta^i} \theta^\top \phi_{s',a'} \end{bmatrix} + \gamma \begin{bmatrix} \sum_{s',a'} t(s_1,a_1,s') \pi_\theta(a'|s') \phi_{s',a'}^i \\ \vdots \\ \sum_{s',a'} t(s_n,a_m,s') \pi_\theta(a'|s') \phi_{s',a'}^i \end{bmatrix} - \Phi^{:,i} \right) \\ &= \sum_{s,a} d(s,a)\phi^\top \left(\gamma \sum_{s',a'} t(s,a,s') \frac{\partial \pi_\theta(a'|s')}{\partial \theta^i} \theta^\top \phi_{s',a'}^i + \gamma \sum_{s',a'} t(s,a,s') \pi_\theta(a'|s') \phi_{s',a'}^i - \phi_{s,a}^i \right) \\ &= \mathbb{E}_{\pi_\theta}\left[\left(\gamma \mathbb{E}_{\pi_\theta}\left[\frac{\partial \pi_\theta(a'|s')/\partial \theta^i}{\pi_\theta(a'|s')} \theta^\top \phi_{s',a'} \right] + \gamma \mathbb{E}_{\pi_\theta}\left[\phi_{s',a'}^i \right] - \phi_{s,a}^i \right) \phi_{s,a}^\top \right]. \end{aligned} \tag{14}$$

Plugging (12), (13), and (14) into (11) gives

$$\begin{aligned}\frac{\partial}{\partial \theta^i}\mathrm{MSPBE}(\theta) = {} & 2\mathbb{E}_{\pi_\theta}\left[\frac{\frac{\partial \pi_\theta(a|s)}{\partial \theta^i}}{\pi_\theta(a|s)} \pi_\theta(a|s) \delta \phi_{s,a}^\top \right] w \\ & + 2\mathbb{E}_{\pi_\theta}\left[\left(\gamma \mathbb{E}_{\pi_\theta}\left[\frac{\partial \pi_\theta(a'|s')/\partial \theta^i}{\pi_\theta(a'|s')} \theta^\top \phi_{s',a'} \right] + \gamma \mathbb{E}_{\pi_\theta}\left[\phi_{s',a'}^i \right] - \phi_{s,a}^i \right) \phi_{s,a}^\top \right] w \\ & - w^\top \mathbb{E}_{\pi_\theta}\left[\frac{\frac{\partial \pi_\theta(a|s)}{\partial \theta^i}}{\pi_\theta(a|s)} \phi_{s,a}\phi_{s,a}^\top \right] w. \end{aligned} \tag{15}$$

Stacking the individual terms together into a gradient gives

$$\begin{aligned}
\nabla_\theta \text{MSPBE}(\theta) &= 2\mathbb{E}_{\pi_\theta}\left[\frac{\nabla_\theta \pi_\theta(a|s)}{\pi_\theta(a|s)}\pi_\theta(a|s)\delta\phi_{s,a}^\top\right] w \\
&\quad + 2\mathbb{E}_{\pi_\theta}\left[\left(\gamma\mathbb{E}_{\pi_\theta}\left[\frac{\nabla_\theta \pi_\theta(\cdot|s')}{\pi_\theta(\cdot|s')}\theta^\top\phi_{s',\cdot}\right] + \gamma\mathbb{E}_{\pi_\theta}\left[\phi_{s',\cdot}^\top\right] - \phi_{s,a}\right)\phi_{s,a}^\top\right] w \\
&\quad - w^\top\mathbb{E}_{\pi_\theta}\left[\frac{\nabla_\theta \pi_\theta(a|s)}{\pi_\theta(a|s)}\phi_{s,a}\phi_{s,a}^\top\right] w \\
&= 2\mathbb{E}_{\pi_\theta}\left[\frac{\nabla_\theta \pi_\theta(a|s)}{\pi_\theta(a|s)}\pi_\theta(a|s)\delta\phi_{s,a}^\top\right] w \\
&\quad + 2\mathbb{E}_{\pi_\theta}\left[\gamma\mathbb{E}_{\pi_\theta}\left[\frac{\nabla_\theta \pi_\theta(\cdot|s')}{\pi_\theta(\cdot|s')}\phi_{s',\cdot}^\top\theta\right]\right] w \\
&\quad + 2\mathbb{E}_{\pi_\theta}\left[\gamma\mathbb{E}_{\pi_\theta}\left[\phi_{s',\cdot}^\top\right]\right] w - 2\mathbb{E}_{\pi_\theta}\left[\phi_{s,a}\phi_{s,a}^\top\right]\underbrace{\mathbb{E}_{\pi_\theta}[\phi_{s,a}\phi_{s,a}^\top]^{-1}\mathbb{E}_{\pi_\theta}[\delta\phi_{s,a}]}_{=w} \\
&\quad - w^\top\mathbb{E}_{\pi_\theta}\left[\frac{\nabla_\theta \pi_\theta(a|s)}{\pi_\theta(a|s)}\phi_{s,a}\phi_{s,a}^\top\right] w \\
&= 2\mathbb{E}_{\pi_\theta}\left[\frac{\nabla_\theta \pi_\theta(a|s)}{\pi_\theta(a|s)}\pi_\theta(a|s)\delta\phi_{s,a}^\top\right] w \\
&\quad + 2\mathbb{E}_{\pi_\theta}\left[\gamma\mathbb{E}_{\pi_\theta}\left[\frac{\nabla_\theta \pi_\theta(\cdot|s')}{\pi_\theta(\cdot|s')}\phi_{s',\cdot}^\top\theta\right]\right] w \\
&\quad + 2\mathbb{E}_{\pi_\theta}\left[\gamma\mathbb{E}_{\pi_\theta}\left[\phi_{s',\cdot}^\top\right]\right] w - 2\mathbb{E}_{\pi_\theta}[\delta\phi_{s,a}] \\
&\quad - w^\top\mathbb{E}_{\pi_\theta}\left[\frac{\nabla_\theta \pi_\theta(a|s)}{\pi_\theta(a|s)}\phi_{s,a}\phi_{s,a}^\top\right] w. \qquad (16)
\end{aligned}$$

Multiplying both sides with $-1/2$ gives

$$\begin{aligned}
-\frac{1}{2}\nabla_\theta \text{MSPBE}(\theta) &= \mathbb{E}_{\pi_\theta}[\delta\phi_{s,a}] - \mathbb{E}_{\pi_\theta}\left[\gamma\mathbb{E}_{\pi_\theta}\left[\phi_{s',\cdot}\right]\phi^\top\right] w \\
&\quad - \mathbb{E}_{\pi_\theta}\left[\frac{\nabla_\theta \pi_\theta(a|s)}{\pi_\theta(a|s)}\delta\phi_{s,a}^\top\right] w + \frac{1}{2}\mathbb{E}_{\pi_\theta}\left[\frac{\nabla_\theta \pi_\theta(a|s)}{\pi_\theta(a|s)}(w^\top\phi_{s,a})^2\right] \\
&\quad - \gamma\mathbb{E}_{\pi_\theta}\left[\mathbb{E}_{\pi_\theta}\left[\frac{\nabla_\theta \pi_\theta(\cdot|s')}{\pi_\theta(\cdot|s')}\phi_{s',\cdot}^\top\theta\right]\phi_{s,a}^\top\right] w. \qquad (17)
\end{aligned}$$

■

Appendix B. Proof of the Off-policy PGQ Gradient Lemma 2

In this section we proof the off-policy conversion Lemma 2.

Proof [Proof of Lemma 2] We want to re-express the expectations stated in Lemma 1 in terms of the behavior policy b rather than the target policy π_θ. This is done using the common importance sampling correction defined as

$$\rho \overset{\text{def.}}{=} \frac{\pi_\theta(a|s)}{b(a|s)}. \qquad (18)$$

However, ρ only corrects the action selection probabilities and not the probability of sampling a transition starting at a state s. Similar to GTD (Sutton et al., 2009b), TDC (Sutton et al., 2009a), and GQ (Maei and Sutton, 2010), we do not correct the stationary distribution $d^b(s)$ from the behavior policy b to the target policy π_θ. Sutton et al. (2009b) have shown that doing this stabilizes off-policy training. Hence we can re-express the MSPBE gradient as

$$-\frac{1}{2}\nabla_\theta \text{MSPBE}(\theta) = \mathbb{E}_b\left[\rho\delta\phi_{s,a}\right] - \mathbb{E}_b\left[\gamma\rho\mathbb{E}_{\pi_\theta}\left[\phi_{s',\cdot}\right]\phi^\top\right]w - \mathbb{E}_b\left[\rho\frac{\nabla_\theta\pi_\theta(a|s)}{\pi_\theta(a|s)}\delta\phi_{s,a}^\top\right]w$$
$$+\frac{1}{2}\mathbb{E}_b\left[\rho\frac{\nabla_\theta\pi_\theta(a|s)}{\pi_\theta(a|s)}(\phi_{s,a}^\top w)^2\right] - \gamma\mathbb{E}_b\left[\rho\mathbb{E}_{\pi_\theta}\left[\frac{\nabla_\theta\pi_\theta(\cdot|s')}{\pi_\theta(\cdot|s')}\phi_{s',\cdot}^\top\theta\right]\phi_{s,a}^\top\right]w$$

and the auxiliary weight vector as

$$w = \mathbb{E}_b\left[\rho\phi_{s,a}\phi_{s,a}^\top\right]^{-1}\mathbb{E}_b\left[\rho\delta\phi_{s,a}\right].$$

■

References

Leemon Baird. Residual Algorithms: Reinforcement Learning with Function Approximation. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 30–37. Morgan Kaufmann, 1995.

V. S. Borkar and S.P. Meyn. The o.d.e. method for convergence of stochastic approximation and reinforcement learning. *SIAM J. Control Optim*, 38:447–469, 1999.

Vivek S. Borkar. Stochastic approximation with two time scales. *Systems & Control Letters*, 29(5):291 – 294, 1997.

Thomas Degris, Martha White, and Richard Sutton. Off-Policy Actor-Critic. In John Langford and Joelle Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, ICML '12, pages 457–464, New York, NY, USA, July 2012. Omnipress.

Geoffrey J. Gordon. Chattering in SARSA(λ) - A CMU Learning Lab Internal Report. Technical report, Carnegie Mellon University, 1996.

Geoffrey J. Gordon. Reinforcement Learning with Function Approximation Converges to a Region. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, pages 1040–1046. The MIT Press, 2001.

H. R. Maei and R. S. Sutton. GQ(λ): A general gradient algorithm for temporal-difference prediction learning with eligibility traces. In *Proceedings of the Third Conference on Artificial General Intelligence*, Advances in Intelligent Systems Research. Atlantis Press, March 2010.

Hamid Reza Maei, Csaba Szepesvari, Shalabh Bhatnagar, and Richard S. Sutton. Toward Off-Policy Learning Control with Function Approximation. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 719–726, Haifa, Israel, June 2010. Omnipress.

Theodore J. Perkins and Doina Precup. A convergent form of approximate policy iteration. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 1627–1634. MIT Press, 2003.

Doina Precup, Richard S. Sutton, and Sanjoy Dasgupta. Off-policy temporal difference learning with function approximation. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 417–424, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

Richard Sutton, Hamid Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvari, and Eric Wiewiora. Fast Gradient-Descent Methods for Temporal-Difference Learning with Linear Function Approximation. In Léon Bottou and Michael Littman, editors, *Proceedings of the 26th International Conference on Machine Learning*, pages 993–1000, Montreal, June 2009a. Omnipress.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book. MIT Press, Cambridge, MA, 1 edition, 1998.

Richard S Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In S. A. Solla, T. K. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 1057–1063. MIT Press, 2000.

Richard S Sutton, Hamid R. Maei, and Csaba Szepesvári. A Convergent $O(n)$ Temporal-difference Algorithm for Off-policy Learning with Linear Function Approximation. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1609–1616. Curran Associates, Inc., 2009b.

Christopher J.C.H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3):279–292, May 1992.