Gradient Analysis and Approximations for Off-Policy Optimization

Ramki Gummadi[*]
Google Research.

Dale Schuurmans[†]
Google Research.

Abstract

In this paper, we first study the on-policy gradients for a range of fundamental policy optimization approaches from $Q-$learning, Policy Gradient to Imitation Learning within a common framework and derive a novel characterization of their relative differences under some natural assumptions. This analysis is also extended to the offline RL scenario, which highlights the role of a simple two-dimensional scaling function that depends on the prediction error and the log density ratio of the importance weights. Our characterization suggests several novel types of approximate gradient updates, which show promise in an empirical analysis of synthetic experiments.

1 Introduction

Policy optimization in the context of reinforcement learning typically refers to learning a parameterized policy to maximize some notion of expected return, either via direct interactions with the environment (on-policy) or via learning from a dataset (off-policy). A general approach to policy optimization is based on the policy gradient theorem (Sutton et al., 1999), and learns a parameterized policy using unbiased Monte Carlo estimates of the gradient of the expected reward). Alternately, value-based methods attempt to solve a proxy task of fitting a parameterized $Q-$function, which seeks to explicitly model the returns associated with arbitrary state, action pairs. Even though the objective formulations are very different, it is not surprising that they are closely related. Indeed, the equivalence has led to methods that bridge the gap between the two approaches (Nachum et al., 2017; O'Donoghue et al., 2017). More explicitly, it has also been observed in prior work (Schulman et al., 2018) that policy gradients and $Q-$learning gradients have an interesting relation under certain assumptions. As one of our contributions, we offer a novel and simple perspective on the relation between policy gradients and $Q-$learning gradients under some basic assumptions in the on-policy case about the policy parameterization and an equivalence between the target return estimates.

Policy gradient algorithms can take a variety of forms corresponding to how the stochastic gradient sample estimator is defined. These variations, called baselines (Weaver & Tao, 2001; Greensmith et al., 2001), can result in different distributional properties for the stochastic gradient (e.g. its variance), while having the same mean. Traditionally, baselines are considered only a state dependent function, but recent literature has considered different forms of state-action baselines and their benefits (Wu et al., 2018; Gu et al., 2017; Tucker et al., 2018). In order to derive the relation between policy gradient algorithms and $Q-$learning, we analyze a particular form of policy gradient with a novel state-action baseline equal to the parameterized policy logits themselves. Our analysis shows that the $Q-$learning gradients can be considered a biased version of policy gradient that results from an ablation of two reward-free terms in the Policy Gradient with Policy Baseline (PGPB) expression. In addition, we also show that dropping only one of the two terms leads to the gradient of a novel objective that minimizes the variance of the prediction error.

Another approach to policy optimization is based on formulating policy search as an instance of probabilistic inference (Levine, 2018), where a target policy whose likelihood is proportional to exponentiated returns is considered. In this paper, we also derive a novel expression for the gradient estimate of the maximum likelihood imitation learning objective with respect to the exponentiated

[*] `gsrk@google.com`

[†] `schuurmans@google.com`

returns target policy. This highlights a key difference between the imitation learning objective and the squared error objective in $Q-$learning gradient as being an exponential, as opposed to linear scaling applied to the return prediction error.

Prior work has also shown that it is important to consider distribution shifts from the behavior policy (Espeholt et al., 2018; Wu et al., 2019) in practice, even with nominally off-policy value learning algorithms. Taking these importance corrections into account, we characteirze a two-dimensional scaling function that is present in all the gradient expressions derived, whose inputs comprise of the prediction error as well as the log-density ratio of the importance weights. An important property of this scaling function is that the behavior probability and the target return do not show up anywhere else in the gradient expressions except as inputs into this function. We identify four key properties of this scaling function that are common to all the cases considered, and use them to guide the design of a more practical approximation alternative to the exponential scaling that was derived in the case of the imitation learning gradient. We also find clear evidence of the benefits of such approximations in our numerical evaluations.

Several prior works have made the case for surrogate objectives in policy optimization, even with a well defined reward to be maximized (Chen et al., 2019; Oh et al., 2018; Liang et al., 2018; Schulman et al., 2017; Kumar et al., 2020). Recent work (Kumar et al., 2019) has also studied a very general class of gradient updates for policy optimization. By comparison, in this paper, the range of possible gradient updates being considered is much more narrow and inspired by the specific objectives of policy gradient, Q-learning and imitation learning (maximum likelihood). In particular, the scaling function we refer to in this paper can not represent arbitrary state action dependent functions, and the gradient approximations we consider capture any dependence of the final gradient on the reward and the behavior probability weights within the scaling function.

In this work, we make certain key assumptions to enable the comparison between different objectives. This includes an implicit on-policy assumption for the squared error objective in $Q-$learning. Typically, RL methods also differ widely in how target return estimates are constructed[1]. We ignore these variations in the theoretical analysis by using the exact return is used in place of empirical target estimates.

2 Setting

Consider the standard model-free RL setting of learning a parameterized policy using data sampled from an unknown, infinite horizon MDP with a fixed discount factor. For simpler exposition, we assume a discrete action space, but the core ideas also apply to continuous action settings. We assume that the model can be represented with a combined set of learnable parameters denoted by θ, even though many practical RL algorithms actually require use of a critic network with parameters different from a separate actor network. More precisely, for a given state-action pair, (s, a), let $\pi_\theta(a|s)$ denote the conditional likelihood for action a at state s. Given the single model, in order to derive our comparisons of various learning objectives and their gradients, we deliberately use the notation $q_\theta(s, a)$, that is suggestive of a parametric value function, to instead denote the policy logits.

Assumption 1 *Assume a softmax final layer for the policy network and denote the logits of the policy parameterization by $q_\theta(s, a)$:*

$$\pi_\theta(a|s) \triangleq \exp\left(q_\theta(s, a)\right) / \sum_u \exp(q_\theta(s, u))$$

In order to quantitatively relate the different objective gradients, it is necessary to also define a specific sampling distribution on the data from which the learning objective is constructed. In the case of policy gradient, this distribution is unambiguously defined to be the marginals corresponding to the on-policy trajectory distribution. However, for value based methods like $Q-$learning, it is typically not explicitly specified even though it has been observed that the learning is more challenging when the data is significantly off-policy (Espeholt et al., 2018). We therefore assume that the standard squared error loss for $Q-$learning comes with an on-policy assumption. More formally, assume

[1] unless it is a bandit setup.

some initial state distribution denoted by $\mu(s)$, and let $d^{\pi}_{\mu}(s)$ denote the discounted state visitation distribution on the trajectories induced by the policy π starting from an initial state distribution μ as

$$d^{\pi}_{\mu}(s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^{t}\mathop{\mathbb{E}}_{s_0\sim\mu}\left[Pr^{\pi}(s_t = s|s_0)\right]$$

With this notation, we can write the on-policy assumption as being an expectation with respect to:

$$D_{\pi} \triangleq d^{\pi}_{\mu}(s)\pi(a|s) \tag{1}$$

Additionally, let $Q^{\pi}(s,a)$ denote the true value obtained in the MDP starting from a state-action pair, (s,a), and subsequently following the trajectory generated using π. In practice, $Q^{\pi}(s,a)$ is not directly available during policy optimization, so a target estimate, $\hat{T}(s,a)$, which is possibly biased with respect to $Q^{\pi}(s,a)$, is typically considered. For policy gradient algorithms, a Monte-Carlo estimate of the infinte horizon discounted reward is used. For value based methods like $Q-$learning, a bootstrapped target estimate on top of the instantaneous reward sample is considered instead. Other possibilities for the target estimates that trade-off bias for variance like TD-λ are also possible (Sutton & Barto, 1998). Of special interest is the bandit scenario, for which the several variants to define a target estimate all collapse into a single version, which is equal to the reward sample observed. In order to derive a theoretical relation between the gradients of different possible approaches to policy optimization, we also ignore these practical differences in constructing the return target estimates and consider a version of the respective objectives' gradients where the particular $\hat{T}(s,a)$ is replaced directly with $Q^{\pi}(s,a)$.

3 Gradient Analysis of Various Objectives

In this section, we derive sample estimates for the gradients of four different objectives grounded under the set of common assumptions from Section 2. For simplicity, we only consider the on-policy expressions in this Section, which will be extended to the offline scenario in Section 4.

3.1 Q-learning Gradient

We first consider the squared error minimization objective for $Q-$learning, which fits a parameterized $q_{\theta}(s,a)$ to the target return, $\hat{T}(s,a)$ on a dataset $\mathcal{D}$. Note that both $\mathcal{D}$ as well as the target sample, $\hat{T}(s,a)$ may implicitly even depend on the policy π, but these dependencies are never considered for the gradient calculation while minimizing the squared prediction error in $Q-$learning. Under the assumptions from Section 2, we can write the gradient of the negative MSE objective as:

$$G_{QL} = -\nabla_{\theta}\frac{1}{2}\mathop{\mathbb{E}}_{(s,a)\sim\mathcal{D}}\left[(\hat{T}(s,a) - q_{\theta}(s,a))^2\right] \tag{2}$$

$$= \mathop{\mathbb{E}}_{(s,a)\sim\mathcal{D}}(\hat{T}(s,a) - q_{\theta}(s,a))\nabla_{\theta}q_{\theta}(s,a) \tag{3}$$

$$= \mathop{\mathbb{E}}_{(s,a)\sim D_{\pi}}\left[(Q^{\pi}(s,a) - q_{\theta}(s,a))\nabla_{\theta}q_{\theta}(s,a)\right] \text{ using Equation 1 and assuming } \hat{T}\equiv Q^{\pi} \tag{4}$$

Written as a single sample stochastic gradient estimate of the maximization objective, we have:

$$G_{QL} = \mathop{\mathbb{E}}_{(s,a)\sim D_{\pi}}\hat{G}_{QL}(s,a) \tag{5}$$

$$\text{where,}\quad \hat{G}_{QL}(s,a,\theta) \triangleq \left(Q^{\pi}(s,a) - q_{\theta}(s,a)\right)\nabla_{\theta}q_{\theta}(s,a) \tag{6}$$

3.2 Policy Gradient with Policy Baseline (PGPB)

From the standard policy gradient theorem (Sutton et al., 1999):

$$G_{PG}(\theta) = \sum_{s} d^{\pi}_{\mu}(s)\sum_{a} Q^{\pi}(s,a)\nabla_{\theta}\pi_{\theta}(a|s) \tag{7}$$

$$= \sum_{s} d^{\pi}_{\mu}(s)\sum_{a}\pi_{\theta}(a|s)Q^{\pi}(s,a)\nabla_{\theta}\log\pi_{\theta}(a|s) \tag{8}$$

$$= \mathop{\mathbb{E}}_{(s,a)\sim D_{\pi}}\left[Q^{\pi}(s,a)\nabla_{\theta}\log\pi_{\theta}(a|s)\right] \tag{9}$$

To derive an unbiased sample estimate for Equation 9, there are several versions corresponding to particular choices of control variates, also commonly referred to as baselines. In order to relate to the $Q-$learning gradients, consider a state-action baseline equal to the policy logits themselves, i.e. $q_\theta(s, a)$. Note that Assumption 1 implies the following identity:

$$\nabla_\theta \log \pi_\theta(a|s) = \nabla_\theta q_\theta(s,a) - \sum_u \pi_\theta(u|s) \nabla_\theta q_\theta(s,u) \tag{10}$$

Using this, we can write[2]:

$$G_{PG}(\theta) = \mathop{\mathbb{E}}_{(s,a)\sim D_\pi} \left[Q^\pi(s,a) \nabla_\theta \log \pi_\theta(a|s) \right] \tag{11}$$

$$= \mathop{\mathbb{E}}_{s\sim d^\pi_\mu} \left[\mathop{\mathbb{E}}_{a|s\sim\pi} \left[(Q^\pi(s,a) - q_\theta(s,a)) \nabla_\theta \log \pi_\theta(a|s) \right] + \mathop{\mathbb{E}}_{u|s\sim\pi} \left[q_\theta(s,u) \nabla_\theta \log \pi_\theta(u|s) \right] \right] \tag{12}$$

$$= \mathop{\mathbb{E}}_{s\sim d^\pi_\mu} \left[\mathop{\mathbb{E}}_{a|s\sim\pi} \left[(Q^\pi(s,a) - q_\theta(s,a)) \nabla_\theta \log \pi_\theta(a|s) \right] + \nabla_\theta \mathop{\mathbb{E}}_{u|s\sim\pi_\theta} \left[\hat{q}_\theta(s,u) \right] \right] \tag{13}$$

$$= \mathop{\mathbb{E}}_{s\sim d^\pi_\mu} \left[\mathop{\mathbb{E}}_{a|s\sim\pi} \left[\Big(Q^\pi(s,a) - q_\theta(s,a) \Big) \Big(\nabla_\theta q_\theta(s,a) - \mathop{\mathbb{E}}_{u|s\sim\pi} \left[\nabla_\theta q_\theta(s,u) \right] \Big) \right] + \nabla_\theta \mathop{\mathbb{E}}_{u|s\sim\pi_\theta} \left[\hat{q}_\theta(s,u) \right] \right] \tag{14}$$

In the last term in Equation 14, we use $\hat{q}_\theta$ rather than q_θ to denote a partial derivative that does not consider $\hat{q}_\theta$ for the gradient with respect to θ (i.e., a 'stop gradient'). We can therefore define a sample gradient estimate for $G_{PG}(\theta) = \mathbb{E}_{(s,a)\sim D_\pi} \hat{G}_{PGPB}(s, a, \theta)$ as:

$$\hat{G}_{PGPB}(s,a,\theta) \triangleq \Big(Q^\pi(s,a) - q_\theta(s,a) \Big) \Big(\nabla_\theta q_\theta(s,a) - \mathop{\mathbb{E}}_{u|s\sim\pi} \left[\nabla_\theta q_\theta(s,u) \right] \Big) + \nabla_\theta \mathop{\mathbb{E}}_{u|s\sim\pi_\theta} \left[\hat{q}_\theta(s,u) \right] \tag{15}$$

The gradient estimate $\hat{G}_{PGPB}(s, a, \theta)$ is unbiased for the expected reward objective. Notice that there are two terms, $\mathbb{E}_{u|s\sim\pi} \left[\nabla_\theta q_\theta(s,u) \right]$ and $\nabla_\theta \mathbb{E}_{u|s\sim\pi_\theta} \left[\hat{q}_\theta(s,u) \right]$ in Equation 15 which do not depend on the sampled action, a or the reward estimate, $Q^\pi(s, a)$. Therefore, both these terms can be computed easily (with zero variance in case of discrete actions, or low variance with the reparameterization trick in case of continuous actions for suitable parameterization; and without querying the environment or the dataset). Comparing $\hat{G}_{PGPB}(s, a, \theta)$ with $\hat{G}_{QL}(s, a, \theta)$ from Equation 6, we can interpret the $Q-$learning gradient as a biased version of policy gradient obtained by dropping the two aforementioned terms in Equation 15.

3.3 Variance Minimization Gradient

By contrast, consider an alternate gradient estimate that results from dropping only one of the two additional terms in $\hat{G}_{PGPB}$ compared to $\hat{G}_Q$, $\nabla_\theta \mathbb{E}_{u|s\sim\pi_\theta} \left[\hat{q}_\theta(s,u) \right]$. Denote this by $\hat{G}_V$:

$$\hat{G}_V(s,a,\theta) \triangleq \Big(Q^\pi(s,a) - q_\theta(s,a) \Big) \Big(\nabla_\theta q_\theta(s,a) - \mathop{\mathbb{E}}_{u|s\sim\pi} \left[\nabla_\theta q_\theta(s,u) \right] \Big) \tag{16}$$

As shown below, this is equivalent to the gradient of an objective which minimizes the variance, instead of the expectation of the squared error across actions.

$$\mathbb{E}\left[\hat{G}_V(s,a,\theta) \right] = \mathop{\mathbb{E}}_{(s,a)\sim D_\pi} \left[\Big(Q^\pi(s,a) - q_\theta(s,a) \Big) \Big(\nabla_\theta q_\theta(s,a) - \mathop{\mathbb{E}}_{u|s\sim\pi} \left[\nabla_\theta q_\theta(s,u) \right] \Big) \right] \tag{17}$$

$$= \mathop{\mathbb{E}}_{s\sim d^\pi_\mu} \left[\mathop{\mathbb{E}}_{a|s\sim\pi} \left[\Big(Q^\pi(s,a) - q_\theta(s,a) \Big) \nabla_\theta q_\theta(s,a) \right] - \mathop{\mathbb{E}}_{a|s\sim\pi} \Big[Q^\pi(s,a) - q_\theta(s,a) \Big] \mathop{\mathbb{E}}_{u|s\sim\pi} \Big[\nabla_\theta q_\theta(s,u) \Big] \right] \tag{18}$$

$$= -\frac{1}{2} \mathop{\mathbb{E}}_{s\sim d^\pi_\mu} \left[\nabla_\theta \mathop{\mathbb{E}}_{a|s\sim\hat{\pi}} \left[(Q^\pi(s,a) - q_\theta(s,a))^2 \right] - \nabla_\theta \left(\mathop{\mathbb{E}}_{a|s\sim\hat{\pi}} \left[Q^\pi(s,a) - q_\theta(s,a) \right] \right)^2 \right] \tag{19}$$

$$= -\frac{1}{2} \mathop{\mathbb{E}}_{s\sim d^\pi_\mu} \nabla_\theta \mathop{\mathbb{V}}_{a|s\sim\hat{\pi}} \Big[Q^\pi(s,a) - q_\theta(s,a) \Big] \text{ where } \mathbb{V} \text{ denotes variance.} \tag{20}$$

In the last two equalities above, we use $\hat{\pi}$ rather than π to indicate that the gradient operator can pass through the expectations without considering the implicit dependency of π on θ.

[2] We suppress π's dependence on θ when there are no gradients through that term.

3.4 Imitation Learning

Next, consider the log-likelihood objective with respect to a target distribution defined by $\hat{p}_T(a|s) \triangleq \exp\Big(Q^\pi(s,a) - F(Q^\pi)(s)\Big)$, where $F(q)(s) \triangleq \log \sum_a \exp q(s,a)$ is used to denote the LogSum-Exp reduction over actions for an arbitrary state-action value function $q(s,a)$. Below, we derive an expression for the gradient of a Maximum Likelihood objective, which highlights it's differences with respect to the squared error objective of $Q-$learning.

$$G_{IL} \triangleq \mathop{\mathbb{E}}_{s\sim d^\pi_\mu}\left[\mathop{\mathbb{E}}_{a|s\sim\hat{p}_T} \nabla_\theta \log \pi_\theta(a|s)\right] \tag{21}$$

$$= \mathop{\mathbb{E}}_{s\sim d^\pi_\mu}\left[\mathop{\mathbb{E}}_{a|s\sim\hat{p}_T}\left[\nabla_\theta q_\theta(s,a) - \sum_u \pi_\theta(u|s)\nabla_\theta q_\theta(s,u)\right]\right] \tag{22}$$

$$= \mathop{\mathbb{E}}_{s\sim d^\pi_\mu}\left[\mathop{\mathbb{E}}_{a|s\sim\hat{p}_T}\Big[\nabla_\theta q_\theta(s,a)\Big] - \sum_u \pi_\theta(u|s)\nabla_\theta q_\theta(s,u)\right] \tag{23}$$

$$= \mathop{\mathbb{E}}_{s\sim d^\pi_\mu}\left[\mathop{\mathbb{E}}_{a|s\sim\pi}\left[\frac{\hat{p}_T(a|s)}{\pi_\theta(a|s)}\nabla_\theta q_\theta(s,a)\right] - \mathop{\mathbb{E}}_{u|s\sim\pi}\Big[\nabla_\theta q_\theta(s,u)\Big]\right] \tag{24}$$

$$= \mathop{\mathbb{E}}_{s\sim d^\pi_\mu}\left[\mathop{\mathbb{E}}_{a|s\sim\pi}\left[\left(\frac{\hat{p}_T(a|s)}{\pi_\theta(a|s)} - 1\right)\nabla_\theta q_\theta(s,a)\right]\right] \tag{25}$$

$$= \mathop{\mathbb{E}}_{s\sim d^\pi_\mu}\left[\mathop{\mathbb{E}}_{a|s\sim\pi}\left[\Big(\exp\Big(Q^\pi(s,a) - q_\theta(s,a) + F(q_\theta)(s) - F(Q^\pi)(s)\Big) - 1\Big)\nabla_\theta q_\theta(s,a)\right]\right] \tag{26}$$

Now consider a biased gradient estimate which drops the log-normalizer difference term, $F(Q^\pi)(s) - F(q_\theta)(s)$, as $\hat{G}_{IL}$.

$$\hat{G}_{IL}(s,a,\theta) \triangleq \Big(\exp\big(Q^\pi(s,a) - q_\theta(s,a)\big) - 1\Big)\nabla_\theta q_\theta(s,a) \tag{27}$$

In Oh et al. (2018), the authors propose a novel objective derived from modifying $Q-$learning by clipping the prediction error below at 0, which is referred to as self imitation learning. We write the gradient expression corresponding to the self imitation learning objective as $\hat{G}_{SIL}$ next.

$$\hat{G}_{SIL}(s,a,\theta) = \max(Q^\pi(s,a) - q_\theta(s,a), 0)\nabla_\theta q_\theta(s,a) \tag{28}$$

This suggests that an alternate way to interpret the SIL gradient is by considering $\max(x,0)$ as an approximation to $e^x - 1$.

4 Gradient Scaling Functions with Off-Policy Correction

Until now, we made an assumption that the sampling distribution for states and actions matches the policy π_θ. In this section, that assumption is relaxed. Let π_b denote a behavior policy that is different from the current policy. Note that π_b may not be explicitly available, but can be estimated using conditional maximum-likelihood as:

$$b = \arg\max_\theta \sum_{(s,a)\sim\mathcal{D}} \log \pi_\theta(a|s) \tag{29}$$

Consider off-policy importance weights for action conditionals[3] between the behavior policy and the current policy, $\frac{\pi_\theta(a|s)}{\pi_b(a|s)}$. To simplify the expressions, denote the off-policy log density ratio and the prediction error as the following two sample dependent scalar 'learning signals':

$$\Delta_O \triangleq \log \frac{\pi_\theta(a|s)}{\pi_b(a|s)} \tag{30}$$

[3] We consider exploration issues outside the scope of this work and ignore any mismatch in the state-marginals while computing the importance weights.

$$\Delta_R \triangleq Q^\pi(s,a) - q_\theta(s,a) \tag{31}$$

Δ_O is the logarithm of the off-policy importance weight ratio, and can also be viewed as a single sample unbiased estimate of the KL divergence between the behavior and current policies. When the dataset is on-policy, this implies $\Delta_O = 0$ for all samples, (s,a). By contrast, $\Delta_R = 0$ when the value prediction error is zero at the current sample, (s,a). With this notation, the five gradient expressions in Equations 6, 15, 16, 27, 28 can be extended to the off-policy case where the dependence on Δ_O and Δ_R can be captured as a function $f : \mathbb{R}^2 \mapsto \mathbb{R}$.

$$\hat{G}_{QL}(s,a,\theta) = f_{SQ}(\Delta_O, \Delta_R)\Big(\nabla_\theta q_\theta(s,a)\Big) \tag{32}$$

$$\hat{G}_{IL}(s,a,\theta) = f_{IL}(\Delta_O, \Delta_R)\Big(\nabla_\theta q_\theta(s,a)\Big) \tag{33}$$

$$\hat{G}_{SIL}(s,a,\theta) = f_{SIL}(\Delta_O, \Delta_R)\Big(\nabla_\theta q_\theta(s,a)\Big) \tag{34}$$

$$\hat{G}_{PGPB}(s,a,\theta) = f_{SQ}(\Delta_O, \Delta_R)\Big(\nabla_\theta q_\theta(s,a) - \mathop{\mathbb{E}}_{u|s\sim\pi}[\nabla_\theta q_\theta(s,u)]\Big) + \nabla_\theta \mathop{\mathbb{E}}_{u|s\sim\pi_\theta}[\hat{q}_\theta(s,u)] \tag{35}$$

$$\hat{G}_{V}(s,a,\theta) = f_{SQ}(\Delta_O, \Delta_R)\Big(\nabla_\theta q_\theta(s,a) - \mathop{\mathbb{E}}_{u|s\sim\pi}[\nabla_\theta q_\theta(s,u)]\Big) \tag{36}$$

The scaling functions, f_{SQ}, f_{IL}, f_{SIL} used above are:

$$f_{SQ}(\Delta_O, \Delta_R) = e^{\Delta_O}\Delta_R \tag{37}$$

$$f_{IL}(\Delta_O, \Delta_R) = e^{\Delta_O}(e^{\Delta_R} - 1) \tag{38}$$

$$f_{SIL}(\Delta_O, \Delta_R) = e^{\Delta_O}\max(\Delta_R, 0) \tag{39}$$

In all cases above, it is easy to verify that $f \in \{f_{SQ}, f_{IL}, f_{SIL}\}$ satisfies the properties listed below. While we do not currently have a formal theoretical justification for why these properties might be important, they seem like reasonable constraints to use in constructing closely related, but slightly more general and improved alternative update rules.

Assumption 2 *Let $f : \mathbb{R}^2 \mapsto \mathbb{R}$ denote a function that captures the dependence of a gradient update on Δ_O, Δ_R. Then, we require that f satisfies the following constraints:*

1. *$f(\Delta_O, 0) = 0, \ \forall \ \Delta_O \in \mathbb{R}$.*
2. *$f(-\infty, \Delta_R) = 0, \ \forall \ \Delta_R \in \mathbb{R}$.*
3. *$f(\Delta_O, \Delta_R)$ is non-decreasing and convex in Δ_R for any fixed Δ_O.*
4. *$|f(\Delta_O, \Delta_R)|$ is non-decreasing and convex in Δ_O for any fixed Δ_R. ($|.|$ denotes absolute value.)*

4.1 A PRACTICAL APPROXIMATION TO THE IMITATION LEARNING GRADIENT

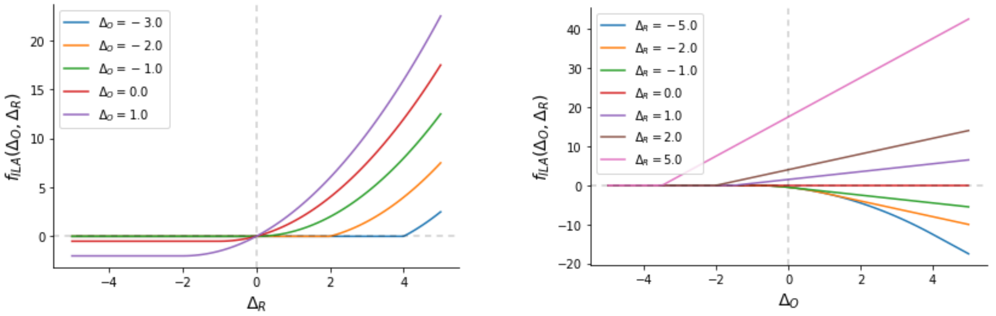
We now construct a piece-wise polynomial second order approximation[4], f_{ILA}, to $f_{IL}(\Delta_O, \Delta_R) = e^{\Delta_O}(e^{\Delta_R} - 1)$ used in $\hat{G}_{IL}$ from Equation 33 that is consistent with the 4 constraints listed in Assumption 2.

$$f_{ILA}(\Delta_O, \Delta_R) = \begin{cases} -\frac{1}{2}(1+\Delta_O)^2, & \text{if } 1 + \Delta_O + \Delta_R \leq 0 < 1 + \Delta_O \\ \Delta_R \max\left(1 + \Delta_O + \frac{\Delta_R}{2}, 0\right), & \text{otherwise} \end{cases} \tag{40}$$

Using this, we define the approximate gradient estimate $\hat{G}_{ILA}$ as:

$$\hat{G}_{ILA}(s,a,\theta) = f_{ILA}(\Delta_O, \Delta_R)\Big(\nabla_\theta q_\theta(s,a)\Big) \tag{41}$$

A visualization for the projection of f_{ILA} along various slices of Δ_O and Δ_R is shown in Figures 1a and 1b. When highly off-policy ($\Delta_O \ll 0$), note that $f_{ILA}(\Delta_O, \Delta_R) = 0$ for Δ_R less than a positive threshold. This threshold increases as the sample becomes more off-policy (ie. as $\Delta_O \to -\infty$). For



(a) Projection of f_{ILA} onto Δ_R for varying Δ_O

(b) Projection of f_{ILA} onto Δ_O for varying Δ_R

Figure 1: One-dimensional projections for $f_{ILA}(\Delta_O, \Delta_R)$ from Equation 40.

large values of Δ_R, the update strength is quadratic in Δ_R. Conversely, it saturates to a small non-positive constant as $\Delta_R \to -\infty$.

The full range of gradients characterized are summarized in Table 1, which mixes the three types of updates (squared error minimization, variance minimzation and PGPB), with various scaling functions across the rows. In addition to the six types of updates derived in Equations 32, 33, 34, 35, 36, 41, the table also lists six natural additional variations for a total of twelve update rules being considered in the numerical evaluations in Section 5.

Table 1: A summary of the various gradients derived and their relationships.

$$\mathcal{U}_Q(f) \equiv f(\Delta_O, \Delta_R)\Big(\nabla_\theta q_\theta(s,a)\Big)$$
$$\mathcal{U}_V(f) \equiv f(\Delta_O, \Delta_R)\Big(\nabla_\theta q_\theta(s,a) - \mathbb{E}_{u|s\sim\pi}\left[\nabla_\theta q_\theta(s,u)\right]\Big)$$
$$\mathcal{U}_P(f) \equiv f(\Delta_O, \Delta_R)\Big(\nabla_\theta q_\theta(s,a) - \mathbb{E}_{u|s\sim\pi}\left[\nabla_\theta q_\theta(s,u)\right]\Big) + \nabla_\theta\, \mathbb{E}_{u|s\sim\pi_\theta}\left[\hat{q}_\theta(s,u)\right]$$

f	$\mathcal{U}_Q(f)$	$\mathcal{U}_V(f)$	$\mathcal{U}_P(f)$
$f_{SQ}(x,y) \triangleq e^x y$	$\hat{G}_Q$	$\hat{G}_V$	$\hat{G}_{PGPB}$
$f_{IL}(x,y) \triangleq e^x(e^y - 1)$	$\hat{G}_{IL}$	$\hat{G}_{IL,V}$	$\hat{G}_{IL,PGPB}$
$f_{SIL}(x,y) \triangleq e^x \max(y,0)$	$\hat{G}_{SIL}$	$\hat{G}_{SIL,V}$	$\hat{G}_{SIL,PGPB}$
$f_{ILA}(x,y) \triangleq \begin{cases} -\frac{1}{2}(1+x)^2, & \text{if } 1+x+y \le 0 < 1+x \\ y\max\left(1+x+\frac{y}{2}, 0\right), & \text{otherwise} \end{cases}$	$\hat{G}_{ILA}$	$\hat{G}_{ILA,V}$	$\hat{G}_{ILA,PGPB}$

5 Empirical Analysis

In this section, we consider a synthetic example that allows for clear numerical comparison among the tweleve update rules summarized in Table 1, without additional confounding issues. To avoid ambiguities with the target sample definition, we consider a (state-dependent) bandit problem under the batch policy optimization framework. The dataset consists of (s, a, r) tuples, where the state space is two-dimensional, i.e. $s = (x_0, x_1) \in \mathbb{R}^2$ and the action space is discrete with cardinality 8, i.e. $a \in \{0, \dots, 7\}$. The reward is defined as $r(s,a) = \sigma(\langle s, \Psi(a)\rangle)$, where $\Psi(a) \triangleq (\cos(2\pi a/8), \sin(2\pi a/8))$ denotes a 2D embedding of the actions onto the unit circle; $\langle a, b\rangle$ denotes the dot product of a, b; and $\sigma(t) = e^t/(1+e^t)$ is the sigmoid function. To generate the dataset, the states are sampled from a standard Gaussian and actions are sampled uniformly. We consider a linear function approximator for the parametric model, $q_\theta(s,a)$, which consists of only

[4] ILA is used as an abbreviation for 'Imitation Learning Approximation'.

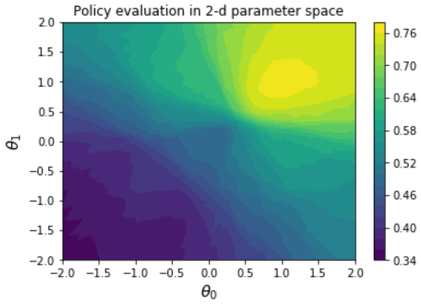


Figure 2: Visualization of policy returns in the parameter space

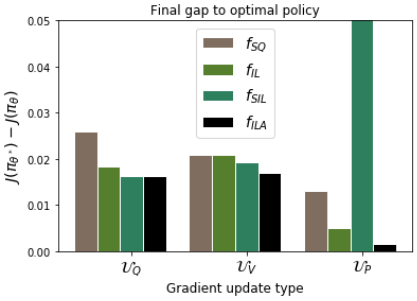


Figure 3: Final reward gap to optimal policy for the twelve gradient update rules in Table 1.

two learnable parameters[5], θ_0, θ_1 as: $q_\theta(s,a) = \langle (\theta_0(1+x_0) - 1, \theta_1(1+x_1) - 1), \Psi(a) \rangle$. Note that the true reward can not be fit exactly using a linear function approximator due to the sigmoid non-linearity. However, we can verify from the problem structure that the optimal parameters to maximize the return are $(\theta_0, \theta_1) = (1, 1)$. Figure 2 gives a numerical visualization of the problem structure in terms of expected policy return as a function of the learnable parameters across the two-dimensional search space traversed by the various gradient update rules. To compare the various update rules, trajectories of θ starting from an intialization of $(0, 0)$ are generated for each of the twelve gradient updates listed in Table 1 using vanilla SGD with an initial learning rate of 0.01 and decreasing linearly as $O(1/t)$. In Figure 3, we plot the 'reward gap', i.e., $J(\pi_{\theta^*}) - J(\pi_\theta)$ where[6] $J(\pi_\theta)$ denotes the expected reward of policy π_θ at the end of 10000 SGD iterations for each of the twelve update rules listed in Table 1. More detailed learning curves are provided in the Appendix in Figure 4. The Euclidean distance of the parameter values to the optimal θ^* at the end is also shown in Figure 5 in the Appendix. We observe from Figure 3 that the best performing update is the combination of the scaling function f_{ILA} derived from an approximation to Imitation Learning in Equation 40 with the PGPB style update from Equation 35:

$$\hat{G}_{ILA,PGPB}(s,a,\theta) = f_{ILA}(\Delta_O, \Delta_R)\Big(\nabla_\theta q_\theta(s,a) - \mathop{\mathbb{E}}_{u|s\sim\pi}\left[\nabla_\theta q_\theta(s,u)\right]\Big) + \nabla_\theta \mathop{\mathbb{E}}_{u|s\sim\pi_\theta}\left[\hat{q}_\theta(s,u)\right] \tag{42}$$

While the exponential scaling function f_{IL} performs worse than f_{ILA}, it still does competitively in these experiments. However, it is not practical in more complex settings where the reward and the importance weights have a much bigger range. Within each style of the gradient update, we find that the imitation learning scaling functions perform better than others in terms of both the speed of convergence as well as the final objective value. By contrast, the vanilla squared error objective, saturates at a suboptimal solution even as it gets there typically much faster compared to policy gradient. By contrast, policy gradient converges to a good final solution in this problem, but is consistently slow. Our generalizations show the potential to get the best of both worlds, which suggests an intriguing possibility for further theoretical study about their convergence guarantees.

6 Conclusion

In this paper, we studied gradient expressions motivated by a range of common objectives in RL under a unified framework, which provides novel insights about their relation. For the offline RL scenario, the sample gradients depend on a simple scaling function comprised of two learning signals, the prediction error and the log density ratio of the importance weights. A comparison of several examples inspires the design for novel approximate updates, which show promise over baselines in preliminary experiments. Applying these ideas to actor-critic settings with two separate models, and theoretical questions about convergence guarantees are some natural questions for further research.

[5] The simpler alternative of $q_\theta(s,a) = \langle (\theta_0 x_0, \theta_1 x_1), \Psi(a) \rangle$ results in a policy parameterization that is scale invariant wrt θ, and hence without a proper solution.

[6] $\theta^* = (1, 1)$.

7 Acknowledgements

We would like to thank Eugene Brevdo, Jincheng Mei and Junfeng Wen for feedback on an initial draft.

References

Minmin Chen, Ramki Gummadi, Chris Harris, and Dale Schuurmans. Surrogate objectives for batch policy optimization in one-step decision making. In *Advances in Neural Information Processing Systems*, 2019. URL.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Volodymir Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. *ArXiv:1802.01561*, 2018. URL.

Evan Greensmith, Peter L. Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. In *Advances in Neural Information Processing Systems*, 2001. URL.

Shixiang Gu, Timothy Lillicrap, Zoubin Ghahramani, Richard E. Turner, and Sergey Levine. Q-prop: Sample-efficient policy gradient with an off-policy critic. In *ICLR*, 2017. URL.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *ArXiv:2006.04779*, 2020. URL.

Saurabh Kumar, Robert Dadashi, Zafarali Ahmed, Dale Schuurmans, and Marc G. Bellemare. Generalized policy updates for policy optimization. In *NeurIPS 2019 Optimization Foundations for Reinforcement Learning Workshop*, 2019. URL.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *ArXiv:1805.00909*, 2018. URL.

Chen Liang, Mohammad Norouzi, Jonathan Berant, Quoc V Le, and Ni Lao. Memory augmented policy optimization for program synthesis and semantic parsing. In *Advances in Neural Information Processing Systems*, 2018. URL.

Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Bridging the gap between value and policy based reinforcement learning. In *Advances in Neural Information Processing Systems*, 2017. URL.

Brendan O'Donoghue, Remi Munos, Koray Kavukcuoglu, and Volodymyr Mnih. Combining policy gradient and q-learning. In *ICLR*, 2017. URL.

Junhyuk Oh, Yijie Guo, Satinder Singh, and Honglak Lee. Self-imitation learning. In *ICML*, 2018. URL.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *ArXiv:1707.06347*, 2017. URL.

John Schulman, Xi Chen, and Pieter Abbeel. Equivalence between policy gradients and soft q-learning. *ArXiv:1704.06440*, 2018. URL.

Richard Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, 1999. URL.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA, 1998.

George Tucker, Surya Bhupatiraju, Shixiang Gu, Richard E. Turner, Zoubin Ghahramani, and Sergey Levine. The mirage of action-dependent baselines in reinforcement learning. In *ICML*, 2018. URL.

Lex Weaver and Nigel Tao. The optimal reward baseline for gradient-based reinforcement learning. In *UAI*, 2001. URL.

Cathy Wu, Aravind Rajeswaran, Yan Duan, Vikash Kumar, Alexandre M. Bayen, Sham M. Kakade, Igor Mordatch, and Pieter Abbeel. Variance reduction for policy gradient with action-dependent factorized baselines. In *ICLR*, 2018. URL.

Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *ArXiv:1911.11361*, 2019. URL.

A Appendix

A.1 Derivation for the scaling function approximation

Consider the following lower bound to $e^{\Delta_O}(e^{\Delta_R}-1)$ which is also exact upto second order around $(\Delta_O, \Delta_R) = (0,0)$. Let $X \sim U[\Delta_O, \Delta_O + \Delta_R]$ be a continuous RV with uniform density. Then:

$$\mathbb{E}[e^X] = \int_{\Delta_O}^{\Delta_O+\Delta_R} \frac{e^x}{\Delta_R} dx = \frac{e^{\Delta_O+\Delta_R} - e^{\Delta_O}}{\Delta_R} \text{ and } e^{\mathbb{E}[X]} = e^{\Delta_O + \frac{\Delta_R}{2}}$$

Jensen's inequality implies that

$$0 < e^{\Delta_O+\frac{\Delta_R}{2}} \leq \frac{e^{\Delta_O+\Delta_R} - e^{\Delta_O}}{\Delta_R} \ \ \forall \ \Delta_O, \Delta_R \in \mathbb{R}$$

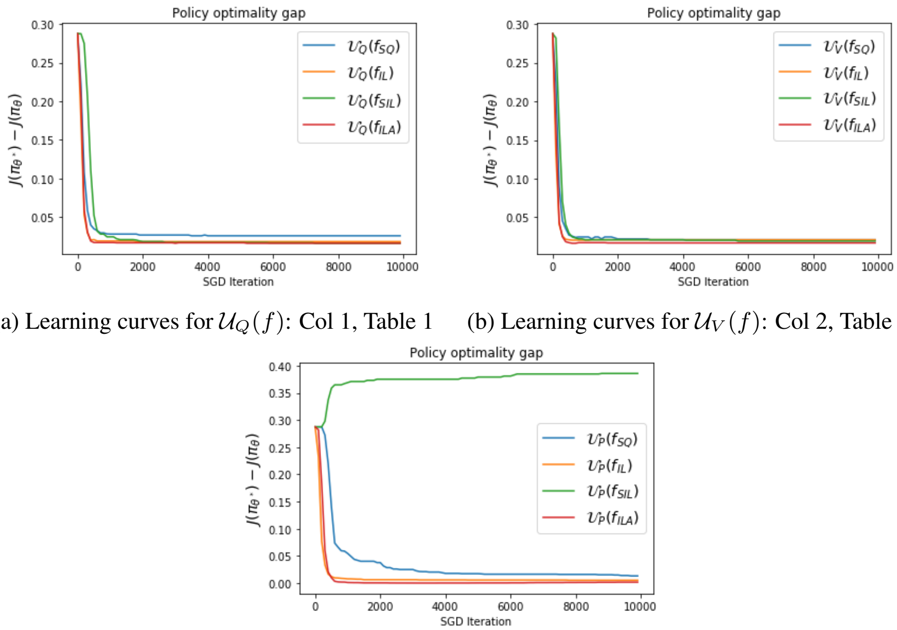
Consider the following approximation inspired by the above inequality:

$$e^{\Delta_O+\frac{\Delta_R}{2}} \approx \frac{e^{\Delta_O+\Delta_R} - e^{\Delta_O}}{\Delta_R} \tag{43}$$

We then consider the following sequence of approximations:

$$\begin{aligned} e^{\Delta_O}(e^{\Delta_R}-1) &= e^{\Delta_O+\Delta_R} - e^{\Delta_O} && (44) \\ &\approx \Delta_R e^{\Delta_O+\frac{\Delta_R}{2}} \quad \text{Using Eq 43} && (45) \\ &\geq \Delta_R \max\left(1 + \Delta_O + \frac{\Delta_R}{2}, 0\right) \quad \text{since } e^x \geq \max(1+x, 0) \ \forall x \in \mathbb{R} && (46) \end{aligned}$$

Unfortunately, $f(\Delta_O, \Delta_R) \triangleq \Delta_R \max\left(1 + \Delta_O + \frac{\Delta_R}{2}, 0\right)$ satisfies the constraints 1, 2 and 4, but not constraint 3, in Assumption 2. To check this, we can verify that it is non-decreasing and convex in Δ_R if and only if $1 + \Delta_O \leq 0$. Equation 40 makes a slight fix to ensure that all 4 constraints hold while also remaining a good second order approximation when Δ_O, Δ_R are small.



(a) Learning curves for $\mathcal{U}_Q(f)$: Col 1, Table 1

(b) Learning curves for $\mathcal{U}_V(f)$: Col 2, Table 1

(c) Learning curves for $\mathcal{U}_P(f)$: Col 3, Table 1

Figure 4: Gap to the optimal policy reward for the twelve updates listed in Table 1 grouped by the update type in each column, and compared across scaling functions $f \in \{f_{SQ}, f_{IL}, f_{SIL}, f_{ILA}\}$

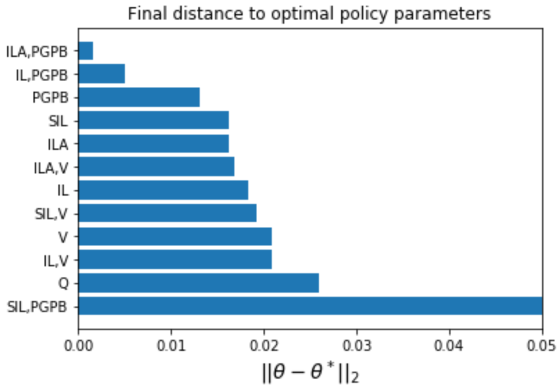


Figure 5: Final euclidean distance to the optimal parameters, $\theta^* = (1, 1)$ for the tweleve update rules considered.