Trajectory-wise Control Variates for Variance Reduction in Policy Gradient Methods

Xinyan Yan[*][*]
Georgia Tech
xinyan.yan@cc.gatech.edu

Ching-An Cheng[*]
Georgia Tech
cacheng@gatech.edu

Byron Boots
Georgia Tech
bboots@cc.gatech.edu

Abstract

Policy gradient methods have demonstrated success in reinforcement learning, but they are also notoriously sample-inefficient. This can be attributed, at least in part, to the high variance in gradient estimates based on Monte Carlo methods. Previous research has endeavored to contend with this problem by control variates (CVs), including the early baselines, state dependent CVs, and the more recent state-action dependent CVs. In this work, we analyze the properties and drawbacks of previous CV techniques. Surprisingly, we find that these works have overlooked an important fact that Monte Carlo gradient estimates are generated by *trajectories* of states and actions. We show that ignoring the correlation across the trajectories can result in suboptimal variance reduction, and we propose a simple fix: a class of *trajectory-wise* CVs, that can further drive down the variance. We show that trajectory-wise CVs can be computed recursively and requires only learning Q-functions like the previous CVs for policy gradient. We further prove that the proposed approach are optimal for variance reduction under reasonable assumptions.

1 Introduction

Policy gradient methods [1–6] are a popular class of model-free reinforcement learning (RL) algorithms. They have many advantages, including simple update rules and convergence guarantees [2, 7–9]. However, basic policy gradient methods, like REINFORCE [1], are also notoriously sample inefficiency. This can be attributed, at least in part, to the high variance in Monte Carlo gradient estimates, which stems from both policy stochasticity necessary for exploration as well as stochastic environmental dynamics. The high variance is further exacerbated as the horizon becomes longer and the dimension becomes higher. If the variance of gradient estimates can be reduced, then the learning speed of policy gradient methods can be accelerated [10, 6].

Variance reduction has been studied since early work of policy gradient methods. For example, function approximators (critics) have been adopted to (partially) replace the Monte Carlo estimates, which reduces variance but at the expense of bias in the search direction [2, 11–15]. This bias-variance tradeoff can work well in practice, but can also diverge when not tuned carefully [14, 16, 9].

Another line of research uses the control variate (CV) method from statistics, designed for reducing variance in Monte Carlo methods without introducing bias [2, 17–24]. For policy gradient especially, the state dependent CVs (also known as baselines or reward reshaping [17, 19]) have been thoroughly investigated [18]. Common state dependent CVs are constructed as approximators of the policy's value function, which admits update rules based on standard policy evaluation techniques. Overall, state dependent CVs are simple to implement and have been found to be quite effective, but the resulting policy gradients can still posses detrimentally high variance, especially in problems that has a long horizon. This has motivated the recent development of state-action dependent CVs [20–23, 25], which can further reduce the variance due to randomness in the *actions*.

[*] Equal contribution.

Optimization Foundations for Reinforcement Learning Workshop at NeurIPS 2019, Vancouver, Canada.

Considering the decades-long development of CV methods, one might wonder if there is a need for new policy gradient CV techniques. In this paper, we argue that the past development of CVs for policy gradients has overlooked an important fact that the Monte Carlo gradient estimates are generated by rolling out a policy and collecting statistics along a *trajectory* of states and actions. Instead the focus has been on sampling *pairs* of states and actions, ignoring the correlation between states and actions *across* time steps. Recently [23] empirically analyzed the variance of instantaneous state-action pairs and compared this to the variance correlations across time steps in multiple simulated robot locomotion tasks. They found that the variance due to long-term trajectories is often larger than the variance due to instantaneous state-action pairs, implying a potential room for improvement.

In this paper, we theoretically analyze the properties of previous CVs, and show that indeed the variance due to long-term trajectories has non-negligible effects. Motivated by this observation, we propose a family of trajectory-wise CVs, called TrajCV, which augment existing CVs with extra terms to *additionally* cancel this long-term variance. We show that TrajCV is particularly effective when the transition dynamics, despite unknown, is close to deterministic. Like existing CVs, TrajCV requires only knowledge of state-action value function (i.e. Q-function) approximates, and can be computed recursively. Furthermore, we prove that TrajCV is optimal for variance reduction under reasonable assumptions. These theoretical insights are validated in simulation (Appendix A).

Upon finishing this work we discovered a recent technical report [24] that is motivated similarly and details exactly the equation (11) that TrajCV uses. Their empirical results on simulated LQG tasks are encouraging too: TrajCV demonstrated superior performance compared with previous state and state-action dependent CVs. By contrast, we derive TrajCV following a completely different route, which brings extra insight into the previous deficiency and suggests natural ways for improvement. In addition, we analyze other potential trajectory-wise CVs and prove the proposed idea is optimal.

2 Problem Setup and Background

We consider *episodic* policy optimization in a finite-horizon Markov Decision Process (MDP) [26, 27] with horizon h, state space $\mathcal{S}$, action space $\mathcal{A}$, instantaneous cost function $c : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, initial state distribution p_{init}, and dynamics $\mathcal{P}$.[2] Given a parameterized *stochastic* policy class Π the goal is to search for a policy in Π that achieves low accumulated costs averaged over trajectories

$$J(\pi) := \mathbf{E}[\textstyle\sum_{t=1}^{h} C_t], \quad \text{where } C_t := c(S_t, A_t), \quad S_1 \sim p_{\text{init}}, \quad A_t \sim \pi_{S_t}, \quad S_{t+1} \sim \mathcal{P}_{S_t, A_t} \quad (1)$$

where $\mathcal{P}_{s,a}$ denotes the distribution of the next state after applying action $a \in \mathcal{A}$ at state $s \in \mathcal{S}$, and π_s denotes the distribution of action at state $s \in \mathcal{S}$. Note that S_t and A_t are the sampled state and action at step t. For simplicity of writing, we embed the time information into the definition of state, e.g., $c(S_t, A_t)$ can represent non-stationary functions. The randomness in (1) consists of the randomness in the start state, policy, and dynamics. In this work, we focus on the case where the dynamics $\mathcal{P}$ and the start state distribution p_{init} are unknown, but the instantaneous cost c is known.

We use uppercases to denote random variables, such as S_t and A_t, with the exception of J. We will use subscript ${}_{i..j}$ to denote the set of random variables, i.e. $X_{1..5} = \{X_1, \ldots, X_5\}$, and ${}_{i:j}$ to denote summation (i.e. $C_{1:h} = \sum_{t=1}^{h} C_t$). As we will be frequently manipulating conditional distributions, we adopt the subscript notation below to write conditional expectation and variance. For $\mathbf{E}_{X|Y}[f(X,Y)]$ of some function f, X denotes the random variable where the expectation is defined and Y denotes the conditioned one. Furthermore, for $f(X_{1..N}, Y)$, we use $\mathbf{E}_{|Y}[f(X_{1..N}, Y)]$ as a shorthand to denote taking the expectation over all other random variables (i.e. $X_{1..N}$) conditioned on Y. This subscript notation also applies to variance, which is denoted as $\mathbf{Var}$.

2.1 Policy Gradient Methods: Pros and Cons

The goal of this paper is to improve policy gradient methods [1, 7, 2–4, 13, 5, 6]. These algorithms treat minimizing (1) as a first-order stochastic non-convex optimization problem, where noisy, unbiased gradient estimates of J in (1) are used to inform policy search. The basic idea is to apply the likelihood-ratio method to derive the gradient of (1). Let us define $N_t := \nabla \log \pi_{S_t}(A_t)$, where ∇ is the derivative with respect to the policy parameters, and define q^π as the Q-function of π; that is, $q^\pi(S_t, A_t) = \mathbf{E}[C_{t:h}]$ where the expectation is generated by taking A_t at S_t and then π afterwards. Define $G := G_{1:h}$ and $G_t := N_t C_{t:h}$. Then it follows [1]

$$\nabla J(\pi) = \mathbf{E}[\textstyle\sum_{t=1}^{h} N_t q^\pi(S_t, A_t)] = \mathbf{E}\left[G\right], \quad (2)$$

[2] We use one-based indexing throughout the manuscript.

where the second equality is due to $q^\pi(S_t, A_t) := \mathbf{E}[C_{t:h}]$. Eq. (2) is an expectation over trajectories generated by running π, so we can treat the random vector G as an unbiased estimate of $\nabla J(\pi)$, which can be computed by executing the policy π starting from initial distribution and then recording the statistics G_t, for $t \in \{1, \dots, h\}$. This technique is known as the Monte Carlo estimate.

The policy gradient methods optimize policies based on gradient estimates constructed using the idea above. While they have numerous advantages as we discussed, simply using the Monte Carlo estimate G in policy optimization (i.e. the vanilla implementation of REINFORCE) can result in poor parameter updates due to excessive variance [13, 14], making learning sample-inefficient.

2.2 Variance Reduction and Control Variate

A powerful technique for reducing the variance in the Monte Carlo estimates is the CV method [28, 29]. Leveraging correlation between random estimates, the CV method has formed the backbone of many state-of-the-art stochastic optimization algorithms [30–32], in particular, practical policy gradient methods for RL [18] because of the high-variance issue of G discussed in the previous section. Below, we review the basics of the CV method as well as previous CV techniques designed for reducing the variance of G. Without loss of generality, we suppose only one trajectory is sampled from the MDP to construct the estimate of (2) and study the variance of different single-sample estimates. We remind that the variance can be always further reduced, when more i.i.d. trajectories are sampled (i.e. using mini batches).

2.3 The Control Variate Method

Consider the problem of estimating the expectation $\mathbf{E}[X]$, where X is a (possibly multivariate) random variable. The CV method [28, 29] is a technique for synthesizing unbiased estimates of $\mathbf{E}[X]$ that potentially have lower variance than the naive sample estimate X. It works as follows: Assume that we have access to another random variable Y, called the *CV*, whose expectation $\mathbf{E}[Y]$ is cheaper to estimate than $\mathbf{E}[X]$. Then we can devise this new estimate by a linear combination,

$$X - \Omega^\top (Y - \mathbf{E}[Y]), \tag{3}$$

where Ω is a properly-shaped matrix. Due to the linearity of expectation, the estimate in (3) is unbiased. While theoretically one can compute the optimal Ω, it requires many data. Therefore, in practice Ω is often set as the identity matrix, which often works well when Y is positively correlated with X. The resulting estimate $X - (Y - \mathbf{E}[Y])$ is known as the *difference estimator* [29] and has variance $\mathbf{Var}[X - Y]$, meaning that if Y is close to X then the variance becomes smaller.

2.4 Common Control Variates for Policy Gradient Methods

The art to various CV methods lies in the design of the correlated random variable Y. The choice is often domain-dependent, based on how X is generated. When estimating the policy gradient in (2), many structures (e.g. the Markov property) can be leveraged to design CVs, as we shall discuss. Following previous works (e.g. [18, 23]) here we focus on the policy gradient component G_t of G given in (2) for simplicity of exposition.[3] The most commonly used CVs for policy gradient [1, 17, 18] are state-dependent functions $\widehat{v} : \mathcal{S} \to \mathbb{R}$, which leads to the difference estimator

$$\tilde{G}_t^{\text{s}} := G_t - \left(N_t \hat{V}_t - \mathbf{E}_{A_t|S_t}[N_t \hat{V}_t]\right) = G_t - N_t \hat{V}_t, \quad \text{where } \hat{V}_t := \hat{v}(S_t), \tag{4}$$

and the expectation vanishes as $\mathbf{E}_{A_t|S_t}[N_t \hat{V}_t] = \hat{V}_t \nabla \mathbf{E}_{A_t|S_t}[1] = 0$.[4] Recently, *state-action CVs* $\hat{q} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ have also been proposed [20–23, 25, 33], in an attempt to reduce more variance through CVs that better correlate with G_t. The state-action CVs yields the difference estimator

$$\tilde{G}_t^{\text{sa}} := G_t - \left(N_t \hat{Q}_t - \mathbf{E}_{A_t|S_t}[N_t \hat{Q}_t]\right), \quad \text{where } \hat{Q}_t := \hat{q}(S_t, A_t). \tag{5}$$

Usually $\hat{v}$ and $\hat{q}$ are constructed as function approximators of the value function v^π and the Q-function q^π of the current policy π, respectively, and learned by policy evaluation, e.g., variants of TD(λ) [34], during policy optimization. Therefore, these methods can also be viewed as unbiased actor-critic approaches. In practice, it has been observed that these CVs indeed accelerate policy optimization, especially in simulated robot control tasks [20, 21, 23, 25, 33].

[3] The variance of G can be bounded by the variance of G_t (Appendix B.3). Tighter bounds can be derived when assumptions on the MDP is made, e.g., faster mixing rate [18].

[4] State dependent functions naturally include non-stationary constant baselines in our notation.

3 Why We Need New Control Variates

Given the decades-long development of CVs for policy gradient reviewed above, one might wonder if there is a need for new policy gradient CV techniques. If so, what is the additional gain we can potentially have? To answer, let us analyze the variance of policy gradient component G_t and how the CVs above reduce it. By the law of total variance, $\mathbf{Var}[G_t]$ can be decomposed into *three terms*

$$\mathbf{Var}_{S_t}\mathbf{E}_{|S_t}\left[N_t C_{t:h}\right] + \mathbf{E}_{S_t}\mathbf{Var}_{A_t|S_t}\left[N_t \mathbf{E}_{|S_t,A_t}\left[C_{t:h}\right]\right] + \mathbf{E}_{S_t,A_t}\mathbf{Var}_{|S_t,A_t}\left[N_t C_{t:h}\right], \quad (6)$$

where the first term is due to the randomness of policy and dynamics before getting to S_t, the second term is due to policy randomness alone at step t, i.e. selecting A_t, and the third term is due to again both the policy and the dynamics randomness in the future trajectories, i.e. after S_t and A_t.[5] Let us measure the size of these three terms by their trace and define

$$\begin{aligned}\mathbb{V}_{S_t} \coloneqq \mathbf{Tr}\left(\mathbf{Var}_{S_t}\mathbf{E}_{|S_t}\left[N_t C_{t:h}\right]\right), \quad \mathbb{V}_{A_t|S_t} \coloneqq \mathbf{Tr}\left(\mathbf{E}_{S_t}\mathbf{Var}_{A_t|S_t}\left[N_t \mathbf{E}_{|S_t,A_t}\left[C_{t:h}\right]\right]\right), \\ \mathbb{V}_{|S_t,A_t} \coloneqq \mathbf{Tr}\left(\mathbf{E}_{S_t,A_t}\mathbf{Var}_{|S_t,A_t}\left[N_t C_{t:h}\right]\right).\end{aligned} \quad (7)$$

Hence, $\mathbf{Tr}\left(\mathbf{Var}[G_t]\right) = \mathbb{V}_{S_t} + \mathbb{V}_{A_t|S_t} + \mathbb{V}_{|S_t,A_t}$. The following theorem shows the size of each term when the policy is Gaussian, which is commonly the case for problems with continuous actions.

Theorem 3.1. Suppose the policy π is Gaussian such that $\pi_{S_t}(A_t) = \mathcal{N}(A_t|\mu_\theta(S_t), \sigma I)$, where μ_θ is the mean function, and θ and $\sigma > 0$ are learnable parameters. Assume the cost function c is bounded and the Q-function $q^\pi(s,a)$ is analytic in a. Then for small enough σ, it holds that $\mathbb{V}_{S_t} = O(h^2)$, $\mathbb{V}_{A_t|S_t} = O(h^2/\sigma^4)$, and $\mathbb{V}_{|S_t,A_t} = O(h^2/\sigma^4)$.

Here we focus on the effects due to the problem horizon h and the policy variance σ. Theorem 3.1 (proved in Appendix B.2) shows that, when the stochasticity in policy decreases (e.g. when it passes the initial exploration phase) the terms $\mathbb{V}_{A_t|S_t}$ and $\mathbb{V}_{|S_t,A_t}$ will dominate variance in policy gradients. An intuitive explanation to this effect is that, as the policy becomes more deterministic, it becomes harder to compute the derivative through zero-order feedback (i.e. accumulated costs). In particular, one can expect that $\mathbb{V}_{|S_t,A_t}$ is likely to be larger than $\mathbb{V}_{A_t|S_t}$ when the variation of $C_{t:h}$ is larger than the variation of $q^\pi(S_t,A_t) \coloneqq \mathbf{E}_{|S_t,A_t}[C_{t:h}]$.[6] With this insight, let us analyze $\mathbf{Var}[\tilde{G}_t^{\text{sa}}]$ to see why using Q-function estimates as CVs (in Section 2.4) can reduce the variance.[7] Akin to the derivation of (6), one can show that $\mathbf{Var}[\tilde{G}_t^{\text{sa}}]$ can be written as

$$\mathbf{Var}_{S_t}\mathbf{E}_{|S_t}[N_t C_{t:h}] + \mathbf{E}_{S_t}\mathbf{Var}_{A_t|S_t}[N_t(\mathbf{E}_{|S_t,A_t}[C_{t:h}] - \hat{Q}_t)] + \mathbf{E}_{S_t,A_t}\mathbf{Var}_{|S_t,A_t}[N_t C_{t:h}]. \quad (8)$$

Comparing (6) and (8), we can see that the CVs in the literature have been focusing on reducing *the second term* $\mathbf{Var}_{A_t|S_t}$. Apparently, from the decomposition (8), the optimal choice of the state-action CV $\hat{q}$ is the Q-function of the current policy q^π, because $q^\pi(S_t,A_t) \coloneqq \mathbf{E}_{|S_t,A_t}[C_{t:h}]$, which explains why $\hat{q}$ can be constructed by policy evaluation. When $\hat{q} = q^\pi$, the effect of $\mathbf{Var}_{A_t|S_t}$ can be completely removed. In practice, $\hat{q}$ is never perfect (let alone the state-dependent version); nonetheless, improvement in learning speed has been consistently reported.

However, Theorem 3.1 suggests that $\mathbf{Var}_{|S_t,A_t}$ is a similar same size as $\mathbf{Var}_{A_t|S_t}$, implying that even when we completely remove the second term $\mathbf{Var}_{A_t|S_t}$, the variance of the gradient estimate can still be significant. Indeed, recently [23] empirically analyzed the three variance components in (8) in LQG and simulated robot locomotion tasks. They found that the third term $\mathbf{Var}_{|S_t,A_t}$ is sometimes close to the second term $\mathbf{Var}_{A_t|S_t}$, and both of them are several orders of magnitude larger than the first term $\mathbf{Var}_{S_t}$. Our Theorem 3.1 supports their finding and implies that there is a potential for improvement by reducing $\mathbf{Var}_{|S_t,A_t}$. We discuss exactly how to do this next.

4 Trajectory-wise Control Variates

We propose a new family of trajectory-wise CVs, called TrajCV, that improves upon existing state or state-action CV techniques by tackling *additionally* $\mathbf{Var}_{|S_t,A_t}$, the variance due to randomness in trajectory after step t. While this idea sounds intuitively pleasing, a technical challenge immediately

[5] The law of total variance: $\mathbf{Var}[f(X,Y)] = \mathbf{E}_X\mathbf{Var}_{Y|X}[f(X,Y)] + \mathbf{Var}_X\mathbf{E}_{Y|X}[f(X,Y)]$ [35].

[6] The empirical size of the three terms on the simulated CartPole task in different policy optimization stages can be found in Appendix A.

[7] Discussion on $\mathbf{Var}[\tilde{G}_t^{\text{s}}]$ is omitted in that $\tilde{G}_t^{\text{s}}$ is subsumed by $\tilde{G}_t^{\text{sa}}$.

arises. Recall in designing CVs, we need to know the expectation of the proposed CV function over the randomness that we wish to reduce (see (3)). In this case, suppose we propose a CV $g(S_{t..h}, A_{t..h})$, we would need to know its conditional expectation $\mathbf{E}_{|S_t,A_t}[g(S_{t..h}, A_{t..h})]$. This need makes reducing $\mathbf{Var}_{|S_t,A_t}$ fundamentally different from reducing $\mathbf{Var}_{A_t|S_t}$, the latter of which has been the main focus in the literature: Because the dynamics $\mathcal{P}$ is unknown, we do not have access to the distribution of trajectories after step t and therefore cannot compute $\mathbf{E}_{|S_t,A_t}$; by contrast, reducing $\mathbf{Var}_{A_t|S_t}$ only requires knowing the policy π.

At first glance this seems like an impossible quest. But we will show that by a clever divide-and-conquer trick, an unbiased CV can actually be devised to reduce the variance $\mathbf{Var}_{|S_t,A_t}$. The main idea is to *1)* decompose $\mathbf{Var}_{|S_t,A_t}$ through repeatedly invoking the law of total variance and then *2)* attack the terms that are *amenable* to reduction using CVs. As expected, the future variance cannot be completely reduced, because of the unknown dynamics. But we should be able to reduce the randomness due to known distributions, namely, the future uses of policy π.

4.1 A Divide-and-Conquer Strategy

Before giving the details, let us first elucidate our idea using a toy problem. Consider estimating $\mathbf{E}\left[f(X_{1..5})\right]$, the expectation of a function f of 5 random variables. We can apply the law of total variance repeatedly, in the order indicated by the subscript, and decompose the variance into

$$\mathbf{Var}\left[f(X_{1..5})\right] = \textstyle\sum_{k=1}^{5} \mathbf{E}_{X_{1..k-1}} \mathbf{Var}_{X_k|X_{1..k-1}} \mathbf{E}_{X_{k+1..n}|X_{1..k}}[f(X_{1..5})] \tag{9}$$

For example, suppose we wish to reduce $\mathbf{Var}_{X_3|X_{1..2}}$ we simply need to consider a CV in the form $g\left(X_{1..3}\right)$, which does not depends on random variables with larger indices. With the difference estimator $f(X_{1..5}) - g\left(X_{1..3}\right) + \mathbf{E}_{X_3|X_{1..2}}[g\left(X_{1..3}\right)]$, the variance $\mathbf{Var}_{X_3|X_{1..2}}$ changes into $\mathbf{E}_{X_{1..2}}\mathbf{Var}_{X_3|X_{1..2}}[\mathbf{E}_{X_{4..5}|X_{1..3}}[f(X_{1..5})] - g(X_{1..3})]$. Apparently when g is optimally chosen as $g^\star(X_{1..3}) := \mathbf{E}_{X_{4..5}|X_{1..3}}[f(X_{1..5})]$, this term vanishes.

Fact 1 A key property of designing CVs by the recursive decomposition above is that the inclusion of the extra term, e.g. $g\left(X_{1..3}\right) - \mathbf{E}_{X_3|X_{1..2}}\left[g(X_{1..3})\right]$, in the difference estimator only affects a single component $\mathbf{Var}_{X_3|X_{1..2}}$ in the total variance, *without influencing the other terms*. This separation property hence allows for a divide-and-conquer strategy: we can design CVs for each term separately and then combine them; the reduction on each term will add up and reduce the total variance.

Fact 2 There is still one missing piece before we can adopt the above idea to design CVs for estimating policy gradients: the ordering of random variables. In the example above, we need to know $\mathbf{E}_{X_3|X_{1..2}}\left[g(X_{1..3})\right]$ to compute the difference estimator. Namely, it implicitly assumes the knowledge about $p(X_3|X_{1..2})$, which may or may not be accessible. Suppose $p(X_3|X_{1..2})$ is not available but $p(X_3|X_{4..5})$ is. We can consider instead using the law of total variance in a different order, e.g. $X_4 \to X_5 \to X_3 \to X_1 \to X_2$, and utilize the information $p(X_3|X_{4..5})$ to construct a difference estimator to reduce $\mathbf{Var}_{X_3|X_{4..5}}$. Therefore, the design of CVs hinges also on the information available. Recall that we only know about the policy but not the dynamics in RL.

4.2 Design of TrajCV

We present TrajCVs for policy gradient below. For transparency, again let us focus on the component G_t, which is a function of $S_{t..h}$ and $A_{t..h}$. Given the information of known random variables (i.e. the policy) and the Markovian structure, a natural ordering for applying law of total variance is

$$S_t \to A_t \to S_{t+1} \to A_{t+1} \to \cdots \to S_h \to A_h. \tag{10}$$

Suppose now we want to reduce $\mathbf{Var}_{A_k|S_{t..k},A_{t..k-1}}$ for some $k > t$. Based on Section 4.1, we may consider a CV in the form $g_k(S_{t..k}, A_{t..k})$, whose the optimal choice is $g_k^\star(S_{t..k}, A_{t..k}) = \mathbf{E}_{|S_{t..k},A_{t..k}}\left[N_t C_{t:h}\right] = N_t\left(C_{t:k-1} + \mathbf{E}_{|S_{t..k},A_{t..k}}[C_{k:h}]\right) = N_t\left(C_{t:k-1} + q^\pi(S_k, A_k)\right)$, where the last equality is due to the Markovian structure and the definition of q^π. This suggests practically we can use $g_k(S_{t..k}, A_{t..k}) := N_t(C_{t:k-1} + \hat{Q}_k)$, where $\hat{Q}_k := \hat{q}(S_k, A_k)$ and $\hat{q} \approx q^\pi$ as was in (5). In other words, we showed that finding the optimal CV for reducing variance in policy gradient can be reduced to learning the Q-function; this enables us to take advantage of existing policy evaluation algorithms. Now we combine $\{g_k(S_{t..k}, A_{t..k})\}_{k=t+1}^{h}$ to build the CV for G_t. Because from Section 4.1 these terms do not interfere with each other, we can simply add them together into $\sum_{k=t+1}^{h} g_k(S_{t..k}, A_{t..k})$. To construct TrajCV, we further add them onto state-action

CV. Equivalently, we have derived a difference estimator:

$$\begin{aligned}\tilde{G}_t^{\text{traj}} &:= \tilde{G}_t^{\text{sa}} - N_t \textstyle\sum_{k=t+1}^{h} \left(g_k(S_{t..k}, A_{t..k}) - \mathbf{E}_{A_k|S_k}[g_k(S_{k..h}, A_{k..h})]\right) \\ &= \tilde{G}_t^{\text{sa}} - N_t \textstyle\sum_{k=t+1}^{h} (\hat{Q}_k - \mathbf{E}_{A_k|S_k}[\hat{Q}_k]) \qquad (11)\end{aligned}$$

Comparing TrajCV in (11) and state-action CV in (4), we see that the TrajCV has additional terms $N_t(\hat{Q}_k - \mathbf{E}_{A_k|S_k}[\hat{Q}_k])$, for $t < k \leq h$, which can be viewed as multiplying N_t with estimates of future advantage functions. [8] Appealing to law of total variance, $\mathbf{Var}[\tilde{G}_t^{\text{traj}}]$ can be decomposed into

$$\begin{aligned}&\mathbf{Var}_{S_t}\mathbf{E}_{|S_t}\left[N_t C_{t:h}\right] + \mathbf{E}_{S_t}\mathbf{Var}_{A_t|S_t}[N_t(\mathbf{E}_{|S_t,A_t}\left[C_{t:h}\right] - \hat{Q}_t)] + \qquad (12) \\ &\sum_{k=t}^{h} \mathbf{E}_{S_k,A_k}\mathbf{Var}_{S_{k+1}|S_k,A_k}\left[N_t \mathbf{E}_{|S_{k+1}}[C_{k:h}]\right] + \sum_{k=t}^{h} \mathbf{E}_{S_{k+1}}\mathbf{Var}_{A_{k+1}|S_{k+1}}[N_t(\mathbf{E}_{|S_{k+1},A_{k+1}}[C_{k:h}] - \hat{Q}_{k+1})]\end{aligned}$$

where we further decompose the effect of $\mathbf{Var}_{|S_t,A_t}$ in the second line into the randomness in dynamics and actions, respectively. Therefore, suppose the underlying dynamics is deterministic (i.e. $\mathbf{Var}_{S_{k+1}|S_k,A_k}$ vanishes), and $\hat{q} = q^\pi$, then using TrajCV (11) would completely remove $\mathbf{Var}_{A_t|S_t}$ and $\mathbf{Var}_{|S_t,A_t}$, the latter of which previous CVs (4) and (5) cannot affect. Note that in implementation of TrajCV for $G_{1:h}$, we only need to compute quantities $\hat{Q}_t$, $\mathbf{E}_{A_t|S_t}[\hat{Q}_t]$ and $\nabla\mathbf{E}_{A_t|S_t}[\hat{Q}_t]$ along a trajectory (done in $O(h)$ time) and they can be used to compute $\{\tilde{G}_t^{\text{traj}}\}_{t=1}^{h}$ (11).[9]

4.3 The Natural Ordering in (10) is Optimal

Recall in Section 4.1 we mentioned that the admissible ordering of random variables used in invoking the law of total variance depends on the information available. Here we show that the chosen ordering (10) is indeed the best ordering to adopt, as we only know the policy, not the dynamics.

We compare (10) against potential orderings constructed by reparameterizing the policy such that its randomness in action becomes *independent of the input state*. We suppose the policy $\pi \in \Pi$ can be reparameterized by a function $\omega : \mathcal{S} \times \mathcal{R} \to \mathcal{A}$ and a distribution p_R, so that for all $s \in \mathcal{S}$, $\omega(s, R)$ and π_s are equal. Reparameterization makes designing a larger family of TrajCVs possible. When applying the law of total variance, the ordering the random variables now can have many possibilities. One might ask, given all possible orderings of random variables, which ordering we should pick to design the CV. Interestingly, to this question, the most natural one and the optimal one coincide. The proof is deferred to Appendix B.

Theorem 4.1. Suppose that policy specified by ω and p_R is known, but the dynamics d is unknown. Assume the optimal CV of a given ordering of random variables $S_{t..h}$ and $R_{t..h}$ can be obtained. The the optimal ordering that minimizes the residue variance is the natural ordering (10) .

Theorem 4.1 tells us that if the optimal CVs are attainable (i.e. we can estimate the Q-function exactly), then the natural ordering is optimal. However, in practice, the CVs are almost always suboptimal due to error in estimation. If the dynamics is relatively accurate and the computing resources for simulation are abundant, then although the residue is higher, the ordering that orders all actions before states except S_t could actually be superior. Therefore, the ordering of random variables based on the relative accuracy of different estimates is an interesting practical question to pursue in future work.

5 Conclusion

We provide theoretical insights into the importance of considering long-term effects in designing CVs for estimating policy gradient, especially for problems with a long horizon. The fix turns out to be quite simple: just padding additional terms (cf. (11)) onto the existing CVs, which can be done using Q-function approximators used in existing CVs without requiring new information. Interestingly we prove this simple idea is actually optimal. Preliminary experimental results supporting the findings are deferred to Appendix A. Important future work includes considering the different bias and variance trade-off discussed in Section 4.3, and learning the linear combination weights of the CVs for policy gradient components $\{G_t\}_{t=1}^{h}$.

Acknowledgments

This work was partially supported by NSF CAREER award 1750483.

[8] For brevity, we may use CV to mean the difference estimator of that CV when there is no confusion.

[9] Ways of approximately evaluating the expectation over actions are deferred to Appendix A.

References

[1] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[2] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, pages 1057–1063, 2000.

[3] Sham M Kakade. A natural policy gradient. In *Advances in Neural Information Processing Systems*, pages 1531–1538, 2002.

[4] Jan Peters and Stefan Schaal. Natural actor-critic. *Neurocomputing*, 71(7-9):1180–1190, 2008.

[5] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897, 2015.

[6] Ching-An Cheng, Xinyan Yan, Nathan Ratliff, and Byron Boots. Predictor-corrector policy optimization. In *International Conference on Machine Learning*, 2019.

[7] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in Neural Information Processing Systems*, pages 1008–1014, 2000.

[8] Ching-An Cheng, Xinyan Yan, Nolan Wagener, and Byron Boots. Fast policy learning through imitation and reinforcement. In *Conference on Uncertainty in Artificial Intelligence*, 2018.

[9] Long Yang and Yu Zhang. Policy optimization with stochastic mirror descent. *arXiv preprint arXiv:1906.10462*, 2019.

[10] Saeed Ghadimi, Guanghui Lan, and Hongchao Zhang. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, 155(1-2):267–305, 2016.

[11] Hajime Kimura, Shigenobu Kobayashi, et al. An analysis of actor-critic algorithms using eligibility traces: reinforcement learning with imperfect value functions. *Journal of Japanese Society for Artificial Intelligence*, 15(2):267–275, 2000.

[12] Philip Thomas. Bias in natural actor-critic algorithms. In *International Conference on Machine Learning*, pages 441–448, 2014.

[13] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International Conference on Machine Learning*, 2014.

[14] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016.

[15] Wen Sun, J Andrew Bagnell, and Byron Boots. Truncated horizon policy search: Combining reinforcement learning & imitation learning. In *International Conference on Learning Representations*, 2018.

[16] Yonathan Efroni, Gal Dalal, Bruno Scherrer, and Shie Mannor. Beyond the one step greedy approach in reinforcement learning. In *International Conference on Machine Learning*, 2019.

[17] Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *International Conference on Machine Learning*, volume 99, pages 278–287, 1999.

[18] Evan Greensmith, Peter L Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(Nov):1471–1530, 2004.

[19] Tang Jie and Pieter Abbeel. On a connection between importance sampling and the likelihood ratio policy gradient. In *Advances in Neural Information Processing Systems*, pages 1000–1008, 2010.

[20] Shixiang Gu, Timothy Lillicrap, Zoubin Ghahramani, Richard E Turner, and Sergey Levine. Q-prop: Sample-efficient policy gradient with an off-policy critic. In *International Conference on Learning Representations*, 2017.

[21] Hao Liu, Yihao Feng, Yi Mao, Dengyong Zhou, Jian Peng, and Qiang Liu. Action-depedent control variates for policy optimization via stein's identity. In *International Conference on Learning Representations*, 2018.

[22] Will Grathwohl, Dami Choi, Yuhuai Wu, Geoffrey Roeder, and David Duvenaud. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. In *International Conference on Learning Representations*, 2018.

[23] George Tucker, Surya Bhupatiraju, Shixiang Gu, Richard E Turner, Zoubin Ghahramani, and Sergey Levine. The mirage of action-dependent baselines in reinforcement learning. *arXiv preprint arXiv:1802.10031*, 2018.

[24] Sergey Pankov. Reward-estimation variance elimination in sequential decision processes. *arXiv preprint arXiv:1811.06225*, 2018.

[25] Cathy Wu, Aravind Rajeswaran, Yan Duan, Vikash Kumar, Alexandre M Bayen, Sham Kakade, Igor Mordatch, and Pieter Abbeel. Variance reduction for policy gradient with action-dependent factorized baselines. In *International Conference on Learning Representation*, 2018.

[26] Richard Bellman. A Markovian decision process. *Journal of Mathematics and Mechanics*, pages 679–684, 1957.

[27] Dimitri P Bertsekas, Dimitri P Bertsekas, Dimitri P Bertsekas, and Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena scientific Belmont, MA, 1995.

[28] Sheldon M Ross. *A course in simulation*. Prentice Hall PTR, 1990.

[29] Art B. Owen. *Monte Carlo theory, methods and examples*. 2013.

[30] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.

[31] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.

[32] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2014.

[33] Kamil Ciosek and Shimon Whiteson. Expected policy gradients for reinforcement learning. *arXiv preprint arXiv:1801.03326*, 2018.

[34] Satinder P Singh and Richard S Sutton. Reinforcement learning with replacing eligibility traces. *Machine learning*, 22(1-3):123–158, 1996.

[35] Kai Lai Chung. *A course in probability theory*. Academic press, 2001.

[36] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *arXiv preprint arXiv:1606.01540*, 2016.

[37] Jeongseok Lee, Michael X. Grey, Sehoon Ha, Tobias Kunz, Sumit Jain, Yuting Ye, Siddhartha S. Srinivasa, Mike Stilman, and C. Karen Liu. DART: Dynamic animation and robotics toolkit. *The Journal of Open Source Software*, 3(22):500, feb 2018.

[38] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.