A Note on ACTION-DEPENDENT FACTORIZED BASELINES

Chiahan Yeh
Department of Computer Science
National Yang Ming Chiao Tung University
jerrym4a123.cs07@nycu.edu.tw

1 Introduction

In policy gradient method, we need to solve the high variance problem in some way. The high variance problem is severe especially in problems with long horizons or high dimensional action spaces. In the class, we learned three ways to reduce the variance. The first one is to set a reference level and use it as a baseline, which doesn't influence the bias. The second one is actor-critic algorithm. The third one is the advantage function. We use *V(s)* to be our baseline, which is a kind of state-dependent baseline.

In this paper, the researchers propose an action-dependent baseline. That is, if an action produced by the policy can be decomposed to multiple factors (like in the aspect of probability distribution), then we can utilized these information to construct our baseline. In other words, we can customize the baseline for every factor of a single action.

The main contributions of this paper is to provide a new perspective on additional information we can collect for designing the baseline. Using the structure in the policy parameterization or factorizing the policy probability has never been discussed before. In the prior work, Q-Prop method can also reduce the variance, but it is computationally expensive. The method proposed in this paper has less computational overhead. The Guided Policy Search method and efforts in multi-agent systems by Foerster et al. and Lowe et al. also apply the additional information to speed up the learning process, but their methods are still quite different from method proposed by this paper.

The view of this paper is new to me. I still have some questions about the implementation of the algorithm it proposed, but the algorithm is easy to understand. The derivation and the experiment result convince me that this method can really reduce the variance and accelerate the learning process, especially for high-dimensional tasks.

2 Problem Formulation

In this section, I will show the notation and preliminaries used in the following paragraph. After that, I will also mention the assumption we need and the optimization problem we will solve.

2.1 NOTATION

We define the markov decision process(MDP) by $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \rho_0, \gamma)$, $\mathcal{S} \subseteq \mathcal{R}^n$ is the n-dimensional state space. $\mathcal{A} \subseteq \mathcal{R}^m$ is the m-dimensional action space. $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathcal{R}_+$ is the transition probability function, $r : \mathcal{S} \times \mathcal{A} \to \mathcal{R}_+$ is the reward function, $\rho_0 : \mathcal{S} \to \mathcal{R}_+$is the initial state distribution function, $\gamma \in (0, 1]$ is the discount factor. The stochastic policy we want to optimize is $\pi_\theta : \mathcal{S} \times \mathcal{A} \to \mathcal{R}_+$,parameterized by θ. Let $\tau = (s_0, a_0, ...)$ denote a trajectory with $s_0 \sim \rho_0(s_0), a_t \sim \pi_\theta(a_t \mid s_t)$, and every $s_{t+1} \sim \mathcal{P}(s_{t+1} \mid s_t, a_t)$. Let $\eta(\pi_\theta)$ denote the expected return of all trajectories we take: $\eta(\pi_\theta) = \mathbb{E}_\tau[\Sigma_{t=0}^\infty \gamma^t r(s_t, a_t)] = \mathbb{E}_{\rho_0}\mathbb{E}_{\pi_\theta}[\Sigma_{t=0}^\infty \gamma^t r(s_t, a_t)]$. We use

$\hat{Q}(s_t, a_t) = \Sigma_{t'=t}^{\infty}\gamma^{t'-t}r_{t'}$ to denote the cumulative return (after time t), and use $Q(s_t, a_t)$ to be the function approximator of $\hat{Q}(s_t, a_t)$. For the action a_t, we use a_t^i to denote the i_{th} dimension of a_t.

2.2 THE SCORE FUNCTION ESTIMATOR

When we are estimating $\nabla_\theta \mathbb{E}_x[f(x)]$, where $x \sim p_\theta(x)$, and $p_\theta(x)$ is the distribution parameterized by θ. Suppose $\log p_\theta(x)$ is continuous in θ, then we can derive that

$$\begin{aligned}
\nabla_\theta \mathbb{E}_x[f(x)] &= \nabla_\theta \int p_\theta(x) f(x) dx \\
&= \int p_\theta(x) \frac{\nabla_\theta p_\theta(x)}{p_\theta(x)} f(x) dx \\
&= \int p_\theta(x) \nabla_\theta \log p_\theta(x) f(x) dx \quad \textbf{(by Chain Rule)} \\
&= \mathbb{E}_x[\nabla_\theta \log p_\theta(x) f(x)]
\end{aligned}$$

2.3 POLICY GRADIENT

Define the state visitation function $\rho_\pi(s) = \Sigma_{t=0}^{\infty}\gamma^t p(s_t = s)$, and $\hat{Q}(s_t, a_t) = \Sigma_{t=0}^{\infty}\gamma^{t'-t}r_{t'}$, then we can write the policy gradient as

$$\nabla_\theta \eta(\pi_\theta) = \mathbb{E}_{\rho_\pi, pi}[\nabla_\theta \log pi_\theta(a_t \mid s_t)\hat{Q}(s_t, a_t)]$$

If we use a state baseline, subtract a value from $\hat{Q}(s_t, a_t)$, then the equation would become

$$\nabla_\theta \eta(\pi_\theta) = \mathbb{E}_{\rho_\pi, pi}[\nabla_\theta \log pi_\theta(a_t \mid s_t)(\hat{Q}(s_t, a_t) - b(s_t))]$$

It can reduce the variance without introducing bias. With the score estimation estimator in 2.2, we can easily prove it like below:

$$\begin{aligned}
\mathbb{E}_{a_t}[\nabla_\theta \log \pi_\theta(a_t \mid s_t)(\hat{Q}(s_t, a_t))] - \mathbb{E}_{a_t}[\nabla_\theta \log \pi_\theta(a_t \mid s_t)(\hat{Q}(s_t, a_t) - b(s_t))] \\
= \mathbb{E}_{a_t}[\nabla_\theta \log \pi_\theta(a_t \mid s_t) b(s_t)] \\
= \nabla_\theta \mathbb{E}_{a_t}[b(s_t)] = 0
\end{aligned}$$

2.4 PROBLEM OF INTEREST AND ASSUMPTION

To be convenient, We assume that each dimension a_t^i of the action a_t is conditionally independent given the current state. In practice, there are two examples satisfy this assumption. The first one is to make $\pi_\theta(a_t \mid s_t)$ a multivariate Gaussian with diagonal variance. The diagonal variance means the covariance of two different dimension is 0, so each dimension of the multivariate Gaussian is independent to the other. The second example is that the action given by the policy has already been factorized to many parts, like a tuple, so each dimension just correspond to a part. Under the assumption, this paper provide an action-dependent baseline algorithm, and demonstrate the benefit gained from it. In the appendix of this paper, it also provide an algorithm which makes no assumption on the action dimensions, so we will talk about it in the last part.

3 Theoretical Analysis

3.1 BASELINES FOR POLICIES WITH CONDITIONALLY INDEPENDENT FACTORS

First it proposed the action-dependent baseline for policies with conditionally independent factors. Because the m-dimensional factors are conditionally independent, we have $\pi_\theta(a_t \mid s_t) = \prod_{i=1}^{m} \pi_\theta(a_t^i \mid s_t)$. By the logarithmic law, we can modify our policy gradient function like below:

$$\mathbb{E}_{\rho_\pi, \pi}[\nabla_\theta \log \pi_\theta(a_t \mid s_t)(\hat{Q}(s_t, a_t))] = \mathbb{E}_{\rho_\pi, \pi}[\Sigma_{i=1}^{m} \nabla_\theta \log \pi_\theta(a_t^i \mid s_t)\hat{Q}(s_t, a_t)] \tag{1}$$

And then, we set the baseline for the ith factor to depend on all the other actions in addition to the state. Let a_t^{-i} denote all dimensions other than i in a_t, and let $b_i(s_t, a_t^{-i})$ denote the ith baseline.

In this paper, it says "Due to conditional independence and the score function estimator,we have $\mathbb{E}_{a_t}[\nabla_\theta \log \pi_\theta(a_t^i \mid s_t) b_i(s_t, a_t^{-i})] = \mathbb{E}_{a_t^{-i}}[\nabla_\theta \mathbb{E}_{a_t^i}[b_i(s_t, a_t^{-i})]] = 0$", but i don't know how to prove it, so here we just assume it is true. After using this baseline, we can modify our policy gradient like this:

$$\begin{aligned} \nabla_\theta \eta(\pi_\theta) &= \mathbb{E}_{\rho_\pi,\pi}[\Sigma_{i=1}^m \nabla_\theta \log \pi_\theta(a_t^i \mid s_t)(\hat{Q}(s_t, a_t) - b_i(s_t, a_t^{-i}))] \\ &= \mathbb{E}_{\rho_\pi,\pi}[\Sigma_{i=1}^m \nabla_\theta \log \pi_\theta(a_t^i \mid s_t) \hat{A}_i(s_t, a_t)] \end{aligned} \tag{2}$$

where $\hat{A}_i(s_t, a_t) = \hat{Q}(s_t, a_t) - b_i(s_t, a_t^{-i})$.

3.2 OPTIMAL BASELINES

To evaluate the suboptimality of the algorithm we will mention, we need to establish the optimal baseline first. Define $z_i := \nabla_\theta \log \pi_\theta(a_t^i \mid s_t)$, and define the component policy gradient:

$$\nabla_\theta \eta(\pi_\theta) = \mathbb{E}_{\rho_\pi,\pi}[\nabla_\theta \log \pi_\theta(a_t^i \mid s_t)(\hat{Q}(s_t, a_t) - b_i(s_t, a_t^{-i}))] \tag{3}$$

In the original paper, There is no θ under the ∇ operator before η_i, that is $\nabla \eta(\pi_\theta)$. I think it is a little mistake. And we denote g_i to be the random variables:

$$g_i := \nabla_\theta \log \pi_\theta(a_t^i \mid s_t)(\hat{Q}(s_t, a_t) - b_i(s_t, a_t^{-i})), \quad a_t \sim \pi_\theta(a_t \mid s_t), s_t \sim \rho_\pi(s_t) \tag{4}$$

To make the derivation clean and simple, they make the assumption:

$$\nabla_\theta \log \pi_\theta(a_t^i \mid s_t)^T \nabla_\theta \log \pi_\theta(a_t^j \mid s_t) \equiv z_i^T z_j \approx 0, \quad \forall i \neq j \tag{5}$$

It means that different factors of the parameters will only influence itself. This assumption will make our derivation easier, but it is not necessary during the implementation. The baseline is still bias-free. Under these assumption, we can calculate the variance of the gradient:

$$\begin{aligned} Var(\Sigma_{i=1}^m g_i) &= \Sigma_i Var(g_i) + \Sigma_i \Sigma_{j \neq i} Cov(g_i, g_j) \quad \textbf{(split by whether i=j)} \\ &= \Sigma_i Var(g_i) + \Sigma_i \Sigma_{j \neq i} \mathbb{E}_{\rho_\pi,\pi}[g_i^T g_j] - \mathbb{E}_{\rho_\pi,\pi}[g_i]^T \mathbb{E}_{\rho_\pi,\pi}[g_j] \\ &= \Sigma_i Var(g_i) + 0 - \Sigma_i \Sigma_{j \neq i} \mathbb{E}_{\rho_\pi,\pi}[g_i]^T \mathbb{E}_{\rho_\pi,\pi}[g_j] \quad \textbf{(by assumption in Equation 4} \\ &= \Sigma_i Var(g_i) - \Sigma_i \Sigma_{j \neq i} M_{ij} \end{aligned} \tag{6}$$

We denote $M_{ij} := \mathbb{E}_{\rho_\pi,\pi}[z_i \hat{Q}(s_t, a_t)]^T \mathbb{E}_{\rho_\pi,\pi}[z_i \hat{Q}(s_t, a_t)]$, and $M = \Sigma_i \Sigma_j M_{ij}$. We can see that there is no $b_i(\bullet)$ in M, so M doesn't depend on $b_i(\bullet)$.

To minimize $Var(\Sigma_{i=1}^m g_i)$, all of $Var(g_i)$ need to be minimized. So we derive what $Var(g_i)$ is first.

$$\begin{aligned} Var(g_i) =& \mathbb{E}_{\rho_\pi,\pi}[z_i^T z_i (\hat{Q}(s_t, a_t) - b_i(s_t, a_t^{-i}))^2] \\ & - \mathbb{E}_{\rho_\pi,\pi}[z_i(\hat{Q}(s_t, a_t) - b_i(s_t, a_t^{-i}))]^T \mathbb{E}_{\rho_\pi,\pi}[z_i(\hat{Q}(s_t, a_t) - b_i(s_t, a_t^{-i}))] \\ =& \mathbb{E}_{\rho_\pi,\pi}[z_i^T z_i (\hat{Q}(s_t, a_t)^2 - 2 b_i(s_t, a_t^{-i}) \hat{Q}(s_t, a_t) + b_i(s_t, a_t^{-i})^2)] \\ & - \mathbb{E}_{\rho_\pi,\pi}[z_i(\hat{Q}(s_t, a_t))]^T \mathbb{E}_{\rho_\pi,\pi}[z_i(\hat{Q}(s_t, a_t))] \quad \textbf{(baseline is biasfree)} \\ =& \mathbb{E}_{\rho_\pi,\pi}[z_i^T z_i \hat{Q}(s_t, a_t)^2] \\ & + \mathbb{E}_{\rho_\pi, a_t^{-i}}[-2 b_i(s_t, a_t^{-i}) \mathbb{E}_{a_t^i}[z_i^T z_i \hat{Q}(s_t, a_t)] + b_i(s_t, a_t^{-i}))^2 \mathbb{E}_{a_t^i}[z_i^T z_i]] - M_{ii} \end{aligned} \tag{7}$$

In the original paper, it wrote the third line as $= \mathbb{E}_{\rho_\pi,\pi}[z_i^T z_i (\hat{Q}(s_t, a_t)^2 - 2 b_i(s_t, a_t^{-i}) \hat{Q}(s_t, a_t) + b_i(s_t, a_t^{-i}))^2]$, in which The place of the square of $b_i(s_t, a_t^{-i})$is wrong.

After finding the form of $Var(g_i)$ under some kind of action-dependent baseline, now we can minimize it by partial derivative.

$$\begin{aligned} & \frac{\partial}{\partial b_i}[Var(\Sigma_i g_i)] = \frac{\partial}{\partial b_i}[Var(g_i)] = 0 \\ \Longrightarrow & -2 \mathbb{E}_{a_t^i}[z_i^T z_i \hat{Q}(s_t, a_t)] + 2 b_i^*(s_t, a_t^{-i}) \mathbb{E}_{a_t^i}[z_i^T z_i] = 0 \\ \Longrightarrow & b_i^*(s_t, a_t^{-i}) = \frac{\mathbb{E}_{a_t^i}[z_i^T z_i \hat{Q}(s_t, a_t)]}{\mathbb{E}_{a_t^i}[z_i^T z_i]} \end{aligned} \tag{8}$$

Therefore, we get the optimal action-dependent baseline is:

$$b_i^*(s_t, a_t^{-i}) = \frac{\mathbb{E}_{a_t^i}[\nabla_\theta \log \pi_\theta(a_t^i \mid s_t)^T \nabla_\theta \log \pi_\theta(a_t^i \mid s_t) \hat{Q}(s_t, a_t)]}{\nabla_\theta \log \pi_\theta(a_t^i \mid s_t)^T \nabla_\theta \log \pi_\theta(a_t^i \mid s_t)} \tag{9}$$

3.3 SUBOPTIMALITY OF THE OPTIMAL STATE-DEPENDENT BASELINE

To evaluate the variance reduction, we first define the following notations:

$$\begin{aligned} Z_i &:= Z_i(s_t, a_t^{-i}) = \mathbb{E}_{a_t^i}[\nabla_\theta \log \pi_\theta(a_t^i \mid s_t)^T \nabla_\theta \log \pi_\theta(a_t^i \mid s_t)] \\ Y_i &:= Y_i(s_t, a_t^{-i}) = \mathbb{E}_{a_t^i}[\nabla_\theta \log \pi_\theta(a_t^i \mid s_t)^T \nabla_\theta \log \pi_\theta(a_t^i \mid s_t)\hat{Q}(s_t, a_t)] \end{aligned} \tag{10}$$

Then, we use I_b to denote the improvement of the baseline b compared to the optimal action-dependent baseline. The baseline is better if this value is smaller. We can calculate the variance reduction of optimal state-dependent baseline by simply calculating the difference of the variance of the optimal action-dependent and state-dependent baseline. By easy arrangement and derivation, we can get the variance reduction of the optimal state-dependent baseline $I_{b=b^*(s)}$:

$$I_{b=b^*(s)} = \Sigma_i \mathbb{E}_{\rho_\pi, a_t^{-i}}[\frac{1}{Z_i}(\frac{Z_i}{\Sigma_j Z_j}\Sigma_j Y_j - Y_i)^2] \tag{11}$$

The variance reduction must ≥ 0, which means the action-dependent is definitely better. How much is it better depends on the deviation of the per-component score-weighted marginalized Q(denoted Y_i) from the component weight of the overall aggregated marginalized Q values(denoted $\Sigma_j Y_j$). When the Q function is highly sensitive to the actions, the difference would be large.

3.4 GLOBAL ACTION-VALUE FUNCTION

Inspired by state-dependent baseline, we often used $b(s_t) = \mathbb{E}_{a_t}[\hat{(Q)}(s_t, a_t)] = V(s_t)$ as our baseline, in advantage function. Similarly, we can use $b(s_t, a_t^{-i}) = \mathbb{E}_{a_t^i}[\hat{(Q)}(s_t, a_t)]$ as the action-dependent baseline. Under the assumption that factors are conditionally independent, we can see that the variance reduction is close to the optimal one.(If the variables are independent, the expected value of the product can change to the product of two expected value, that is: $\mathbb{E}_{a_t^i}[z_i^T z_i \hat{Q}(s_t, a_t)] \approx \mathbb{E}_{a_t^i}[z_i^T z_i]\mathbb{E}_{a_t^i}[\hat{(Q)}(s_t, a_t)]$)

$$I_{b=\mathbb{E}_{a_t^i}[\hat{(Q)}(s_t,a_t)]} = \Sigma_i \mathbb{E}_{\rho_\pi, a_t^{-i}}[Z_i(\mathbb{E}_{a^i}[\hat{Q}(a_t, s_t)] - \frac{\mathbb{E}_{a_t^i}[z_i^T z_i \hat{Q}(s_t, a_t)]}{\mathbb{E}_{a_t^i}[z_i^T z_i]})^2] \approx 0 \tag{12}$$

3.5 MONTE CARLO/MEAN MARGINALIZED Q BASELINE

In the aspect of the implementation, we can use monte carlo to estimate the baseline:

$$b_i(s_t, a_t^{-i}) = \frac{1}{M}\Sigma_{j=0}^M Q_{\pi_\theta}(s_t, (a_t^{-i}, \alpha_j)) \tag{13}$$

$\alpha_j \sim \pi_\theta(a_t^i \mid s_t)$ are samples of the ith factor. We use $a_t^i \times a_t^i$ to be the samples of actions and evaluate the baseline by the samples. This paper also indicates that we can either use max or mean of the samples to compute the function.

It is know that Monte Carlo method is computationally expensive, especially when we use neural network to estimate the policy, the propagation can be even more computationally expensive. Therefore, we can replace the sample method with just using the mean value to denote a_t^i. So the baseline would be:

$$b_i(s_t, a_t^{-i}) = Q_{\pi_\theta}(s_t, (a_t^{-i}, \bar{a}_t^i)) \tag{14}$$

where $\bar{a}_t^i = \mathbb{E}_{\pi_\theta}[a_t^i]$ is the mean of a_t^i.

3.6 EXPERIMENT RESULTS

They compare their algorithm to state-only baseline, and their works is obviously better in all tasks. They also compare the monte carlo and mean marginalized Q baseline, and they found that the latter one performs slightly better towards the end of the learning process. The finite samples of monte carlo seems to inject errors and influence the quality of learning. Sub-sampling may reduce the bias, which is important when the action space is almost accurate. Furthermore, they found the benefit of the action-dependent baseline can be greater for very high dimensional problems. They do the experiment with m-DimTargetMatching task. And for action dimension are 12 and 100, the speed improvements are 0 percent and 9.3 percent. While actions dimensions are 400 and 2000, the improvements are 11.8 percent and 11.3 percent. In general, their work, the action-dependent baseline, is quite successful, especially for high-dimensional tasks.

4 Conclusion

To extend this paper, we can try to factorize the low dimensional action space. This research has got success in high dimensional tasks, because there is more information we can get in those tasks, and they are originally pretty hard to learn, so there is much space to improve. If we can find ways to factorize low dimensional action space and utilize the additional information, maybe we can accelerate the low-dimsional problems, too. We can learn an additional policy to factorize/analyze the action space, which will give out information for us to do policy gradient. There many tasks that the action distribution is not Gaussian distribution. Maybe we can approximate the distribution by many Gaussian distribution, and use the baseline proposed by this paper to accelerate them.