A Note on Stein Variational Policy Gradient

Tsung-Han Liu
Department of Computer Science
National Chiao Tung University
bh2142.cs08@nctu.edu.tw

1 Introduction

Deep neural networks trained with policy gradient methods performs impressively in some simulation environment. However these algorithms are not yet applicable for hard real world tasks because of high variance, slow convergence, and insufficient exploration. The performance is also very sensitive to initializations due to the non-convexity of neural networks.

Stein Variational Policy Gradient (SVPG) allow simultaneous exploration and exploitation of multiple policies. Instead of learning one optimal policy, SVPG attempts to learn a distribution of policies parameters which balance good expected rewards and the diversity of policies drawn from the distribution. The policies sampled from such distribution will be good policies. The SVPG paper modeled such distribution as an optimization problem and find a closed-form solution of it. Therefore, SVPG can leverage Stein Variational Gradient Descent (SVGD) method to deterministically and efficiently approximate the target distribution with a set of policy particles.

Unlike off-policy method which attempts to retrieve global optimal policy via exploring more trjectories with behavior policy, SVPG directly learning a set of good policies different from each other in the parameter space. Benefit from deterministic updation utilized the gradient, SVGD method used in SVPG to approximate a given target distribution converges faster then the Markov chain Monte Carlo (MCMC) method. Also, the utilization of the gradient of the expected rewards J makes it easy to implement SVPG on the top of existing policy gradient methods, such as A2C, REINFORCE and DPG.

2 Problem Formulation

2.1 Reinforcement Learning

A *Markov decision process* is a tuple $(S, A, R, p, p_0, \gamma)$ where S is a set of states, A is a set of actions, $R\left(s, a\right)$ is a reward function, $p\left(s'|a, s\right)$ is a transition kernel, p_0 is an initial state distribution, and $\gamma \in [0, 1)$ A policy $\pi\left(a|s\right)$ is a distribution over actions given a state. A trajectory $\tau^{\pi} = (s_0, a_0, r_0, s_1, a_1, r_1, \ldots)$ where $s_o \sim p_0, a_t \sim \pi\left(\cdot|s_t\right)$. In the model free setting, we don't know what the transition kernel $p\left(s_{t+1}|s_t, a_t\right)$ exactly is.

The goal of reinforcement learning is to find a policy π for choosing actions to maximize the expected return.

$$J\left(\pi\right) = \mathop{\mathbb{E}}_{\tau^{\pi}}\left[\sum_{t=0}^{\infty}\gamma^{t}r(s_t, a_t)\right] \tag{1}$$

In policy gradient method, policy $\pi_\theta\left(a|s\right)$ is parameterized by θ. The policy is imporved by calculating the gradient of the expected return J with respect to the policy parameters θ and applying gradient descent $\theta' \leftarrow \theta + \epsilon\nabla_\theta J(\theta)$. Since the gradient of the expected reward is hard to calculate,

Preprint. Under review.

we may leverages two use full tools. The state value function is

$$V^{\pi}(s) = \mathop{\mathbb{E}}_{\tau}\left[G_t | s_t = s; \pi\right] \tag{2}$$

where $G_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i}$ The V function is the expected reward following the policy π from state s. The state action value function

$$Q^{\pi}(s,a) = \mathop{\mathbb{E}}_{\tau}\left[G_t | s_t = s, a_t = a; \pi\right] \tag{3}$$

repersents the expected total rewards received from state s and taking action a. There are a variety of ways to estimate the gradient of the expected rewrads. Such as the well known REINFORCE

$$\nabla_{\theta} J(\theta) \approx \sum_{t=0}^{\infty} G_t \nabla_{\theta} \log \pi_{\theta}\left(a_t | s_t\right) \tag{4}$$

where $G_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i}$ is the discounted and accumulated reward from step t. A2C which has smaller variance than REINFORCE

$$\nabla_{\theta} J(\theta) \approx \sum_{t=0}^{\infty} \left[Q^{\pi}(s_t, a_t) - V^{\pi}(s_t)\right] \nabla_{\theta} \log \pi_{\theta}\left(a_t | s_t\right) \tag{5}$$

and Determinstic Policy Gradient

$$\nabla_{\theta} J(\theta) \approx \sum_{t=0}^{\infty} \nabla_{\theta} \mu_{\theta}(s_t) \nabla_a Q^{\mu}(s_t, a = \mu) \tag{6}$$

where the $\mu_{\theta}(s)$ is the action determinstically generated by the policy.Silver and Lever [2014]

2.2 Maximum Entropy Policy Optimization

In order to draw different policies for better exploration, SVPG seeks a distribution q of policy parameter θ rather than one optimal policy that maximize the expected reward. The optimization problem is formulate as the following.

$$\max_{q} \left\{ \mathop{\mathbb{E}}_{q(\theta)} [J(\theta)] - \alpha \mathrm{D}_{\mathrm{KL}}\left(q \| q_0\right) \right\} \tag{7}$$

where q maximizes the expected reward regularized by KL divergence $\mathrm{D_{KL}(q \| q_0)} = \mathbb{E}_{q(\theta)}\left[\log q(\theta) - \log q_0(\theta)\right]$ with a prior distribution q_0

When we use an uninformative prior distribution $q_0(\theta) = const$ the regularization term in the above optimzation problem can be simplified to the entropy $\mathrm{H(q)} = \mathbb{E}_{q(\theta)}\left[-\log q(\theta)\right]$ since $q_0(\theta)$ being a constant does not affect the solution of the optimization problem.

$$\max_{q} \left\{ \mathop{\mathbb{E}}_{q(\theta)} [J(\theta)] + \alpha \mathrm{H}\left(q\right) \right\} \tag{8}$$

The entropy term above perserved the diversity of policies drawn from the optimal distribution which benifit exploring in the θ parameter space. The temperature coefficient α balances the exploration and the explotiation.

3 Theoretical Analysis

Theorem 1. *The optimization problem repersenting in (8) has a closed form solution.*

$$q\left(\theta\right) \propto \exp\left(\frac{1}{\alpha} J(\theta)\right) \tag{9}$$

Proof. This proof is not provided by the paper, it is my own new proof. Apply the method of the Largrange Multipliers of to the (8) under constraint $\int_\theta q(\theta)d\theta = 1$

Assume the Largrange Multiplier λ satisfies the following equation.

$$\nabla\left\{ \mathop{\mathbb{E}}_{q(\theta)}[J(\theta)] + \alpha \mathrm{H}(q) \right\} = \lambda \nabla \left\{ \int_\theta q(\theta)d\theta - 1 \right\} \tag{10}$$

The solution of the optimization problem can be transform into finding the maxima of the Largrangian

$$\mathcal{L}(q,\lambda) = \mathop{\mathbb{E}}_{q(\theta)}[J(\theta)] + \alpha \mathrm{H}(q) - \lambda \int_\theta q(\theta)d\theta + 1 \tag{11}$$

Take derivative of the Largrangian and set it to zero.

$$\begin{aligned}
\frac{d}{dq}\left\{ \mathop{\mathbb{E}}_{q(\theta)}[J(\theta)] + \alpha \mathrm{H}(q) - \lambda \int_\theta q(\theta)d\theta + 1 \right\} &= 0 \\
\frac{d}{dq}\left\{ \int_\theta q(\theta)J(\theta)d\theta - \alpha \int_\theta q(\theta)\log q(\theta)d\theta - \lambda \int_\theta q(\theta)d\theta \right\} &= 0 \\
\frac{d}{dq}\left\{ \int_\theta q(\theta)\frac{1}{\alpha}J(\theta)d\theta - \int_\theta q(\theta)\log q(\theta)d\theta - \int_\theta q(\theta)\frac{\lambda}{\alpha}d\theta \right\} &= 0 \\
\frac{d}{dq}\int_\theta q(\theta)\left[\frac{1}{\alpha}J(\theta) - \frac{\lambda}{\alpha}\right]d\theta &= \frac{d}{dq}\int_\theta q(\theta)\log q(\theta)d\theta \\
C_1 + \int_\theta q(\theta)\left[\frac{1}{\alpha}J(\theta) - \frac{\lambda}{\alpha}\right]d\theta &= C_2 + \int_\theta q(\theta)\log q(\theta)d\theta \\
q(\theta)\left[\frac{1}{\alpha}J(\theta) - \frac{\lambda}{\alpha}\right] &= q(\theta)\log q(\theta) \\
\frac{1}{\alpha}J(\theta) - \frac{\lambda}{\alpha} &= \log q(\theta) \\
\frac{1}{\exp\left(\frac{\lambda}{\alpha}\right)}\exp\left(\frac{1}{\alpha}J(\theta)\right) &= q(\theta)
\end{aligned}$$

□

The coefficient α controls the strength of exploitation in the parameter space. When $\alpha \to 0$, $q(\theta)$ becomes a Dirac measure that samples drawn from $q(\theta)$ will be around the global optima of the expected return $J(\theta)$

Stein Variational Gradient Descent (SVGD) is a nonparameteric variational inference algorithm that leverages efficient deterministic dynamics to transport a set of particles $\{\theta_i\}_{i=1}^n$to approximate given target distribution $q(\theta)$. SVGD iteratively updates the particles $\{\theta_i\}$ via

$$\theta_i \leftarrow \theta_i + \epsilon\phi^*(\theta_i) \tag{12}$$

where ϵ is the step size and $\phi(\cdot)$ is a function in the unit ball of a reporducing kernel Hilbert space (RKHS) $\mathcal{H} = \mathcal{H}_0 \times \cdots \times \mathcal{H}_0$ of $d \times 1$ vector-valued functions, chosen to maximumly decrease the KL divergence between the particles and the target distribution.

$$\phi^* \leftarrow \max_{\phi\in\mathcal{H}, \|\phi\|_{\mathcal{H}}\leq 1}\left\{ -\frac{d}{d\epsilon}\mathrm{D}_{\mathrm{KL}}\left(\rho_{[\epsilon\phi]}\|q\right) \right\} \tag{13}$$

where $\rho_{[\epsilon\phi]}$ denotes the distribution updated distribution $\theta' = \theta + \epsilon\phi(\theta)$

Lemma 1. *The above optimization problem (13) has a closed form solution.*

$$\phi^*(\theta) = \mathop{\mathbb{E}}_{\vartheta\sim\rho}[k(\vartheta,\theta)\nabla_\vartheta \log q(\vartheta) + \nabla_\vartheta k(\vartheta,\theta)]$$

where $k(\vartheta,\theta)$ is the positive definite kernel associated with the RKHS $\mathcal{H}_0$ and the distribution of θ is ρ

Proof. This has been proved by Liu and Wang [2016]. □

With suffciently small $\{\epsilon_t\}$, $\phi^*_\infty = 0$ so that $\{\rho_t\}$ weakly converges to q as $t \to \infty$
Lemma 2. *For bounded testing functions h*

$$\frac{1}{n}\sum_{i=1}^{n} h(\theta_i) - \mathop{\mathbb{E}}_{\rho}[h(\theta)] = \mathcal{O}(\frac{1}{\sqrt{n}})$$

which implies when n getting large, the approximation of ϕ^* by a set of particles converges to its target value.

Such theoretical results was established in the mean field theory of interacting particle systems (eg. Moral [2013]).

Lemma 1 suggests the direction of steepest descent that maximizes the negative gradient of the KL divergence between the next-iter distribution ρ_{t+1} and the target distribution q. Since we use particles to approximate the distribution, with Lemma 2 we can derive the Stein Variational Gradient by replacing the expectation in Lemma 1 with the current particles.

$$\hat{\phi}(\theta_i) = \frac{1}{n}\sum_{j=1}^{n}\left[k(\theta_j,\theta_i)\nabla_{\theta_j}\log q(\theta_j) + \nabla_{\theta_j}k(\theta_j,\theta_i)\right] \tag{14}$$

Suspiciously, I don't think the convergence claim mention in Lemma 1 is sufficiently persuasive, so I leverages the latest research on SVGD (Korba et al. [2020]). The Remark 3 of that paper suggested that the weak convergence can be proven using a semi-convexity result on the KL and the Proposition 3 guarantee exponentially fast convergence if the target distribution satisfies the Stein log-Sobolev inequality.

By plugging in the solution suggested by the Theorem1 as the target distribution, we get the Stein Variational Policy Gradient algorithm.

Algorithm 1: Stein Variational Policy Gradient

```
Input: Learning rate $\epsilon$, kernel $k(x, x')$, temperature coefficient, initial policy particles $\{\theta_i\}_{i=1}^n$
for iteration $t = 1, 2, \dots, T$ do
    for particle $i = 1, 2, \dots, n$ do
        $\Delta\theta_i \leftarrow \frac{1}{n}\sum_{j=1}^{n}\left[k(\theta_j,\theta_i)\nabla_{\theta_j}\left(\frac{1}{\alpha}J(\theta_j)\right) + \nabla_{\theta_j}k(\theta_j,\theta_i)\right]$
        $\theta_i \leftarrow \theta_i + \epsilon\Delta_{\theta_i}$
    end
end
```

The gradient of the the expected rewards $J(\theta)$ is calculated by (4)~(6) When we only use single particle i.e. $n = 1$ and choose kernel $\nabla_x k(x, x') = 0$, the SVGD is reduced to a simple gradient ascent method.

In practice, we can use the Gaussian RBF kernel $k(x, x') = \exp\left(-\|x - x'\|_2^2/h\right)$ with bandwidth $h = \frac{m^2}{\log(n+1)}$ where m denotes the median of pairwise distances between the particles $\{\theta_i\}$. This heuristic of choosing the bandwidth is based on the intution $\sum_j(\theta_j, \theta_i) \approx n\exp\left(-\frac{m^2}{h}\right) = 1$ so that the situation of only few particles is interacting with each other is prevented.
The temperature coefficient α controls the balance between exploration and exploitation.

$$\sum_{j=1}^{n}\left[\underbrace{k(\theta_j,\theta_i)\nabla_{\theta_j}\left(\frac{1}{\alpha}J(\theta_j)\right)}_{\text{exploitation}} + \underbrace{\nabla_{\theta_j}k(\theta_j,\theta_i)}_{\text{exploration}}\right] \tag{15}$$

When α is large, exploration dominates exploitation, policy particles are diversified too much, so that we cannot attain good policies. When $\alpha \to 0$, exploitation dominates exploration, the algorithm is then reduced to running n independent policy gradents.

4 Conclusion

SVPG provides a framework to simultaneously train a group of different policies base on the existing policy gradient algorithms. In the other word, if there is another strong policy gradient method proposed in the future, SVPG can easily be extended. Implementing in distributed systems may benifit from the parallel nature of SVPG.

Currently the temperature coefficient α balancing exploration and exploitation still need to be choose by human according to the different base policy gradient algorithm and the environment. There is potential to designing a annealing process for α which may leads to even high expected rewards via "exploration then exploitation". The impact of the kerenl choice is still unknown.

References

David Silver and Guy Lever. Deterministic policy gradient algorithms, 2014.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm, 2016.

P. Del Moral. Mean field simulation for monte carlo integration, 2013.

Anna Korba, Adil Salim, Michael Arbel, Giulia Luise, and Arthur Gretton. A non-asymptotic analysis for stein variational gradient descent, 2020.

Kamil Ciosek and Shimon Whiteson. Expected policy gradients, 2017.