CS885 project

Yingluo Xun, 20500462

1 Introduction

Reinforcement learning, or RL, has been a hot topic for years.

While action-value methods are common, they suffer from two important limitations (Degris et al., 2013). First, for continuous action spaces, it is impossible to achieve a global maximization at every step by greedy policy improvement. Second, a small change in the action-value function can cause large changes in the policy. Instead, a simple and computationally attractive alternative is to move the policy in the direction (locally? my understanding) of the gradient of Q, rather than globally maximizing Q (comment: I tend to view it as a tradeoff). And then the problem will be overcome by policy control methods. In fact, policy gradient methods are the benchmarks for most of the continuous RL problems today, for example, robotics control.

In this review I want to emphasis the benchmark methods for policy control problem, including policy gradient methods, and gradient-free policy search methods. Policy gradient, the mainstream, is a class of function approximation methods that learns a parameterized policy, which is commonly nonlinear, that can select actions without referring to a value function. The fundamental idea behind these algorithms is to adjust the parameter θ of the policy in the direction of the gradient of the performance, i.e. to maximize the expected performance. Although parameterization would cause bias, it provides compact and efficient representation of policy in continuous action space.

2 Notation & Background

Bellman Equation:

$$V_\pi(s) = (TV_\pi)(s) = \mathbb{E}_{a\sim\pi}[R(s,a) + \gamma \mathbb{E}_{s'|s,a}[V_\pi(s')]] \tag{1}$$

Optimal Bellman Equation:

$$V(s) = (TV)(s) = \max_{\pi(\cdot|s)\in P_{\mathbb{A}}} \mathbb{E}[R(s,a) + \gamma \mathbb{E}_{s'|s,a}[V(s')]] \tag{2}$$

Minimizing squared Bellman error:

$$\min_V \mathbb{E}_{s\sim\mu}\Big[\Big(\max_{\pi(\cdot|s)\in P_{\mathbb{A}}} \mathbb{E}[R(s,a) + \gamma \mathbb{E}_{s'|s,a}[V(s')]] - V(s)\Big)^2\Big] \tag{3}$$

Double-sample issue: For the problem f with similar type of the objective function (2), because of the inner conditional expectation $\mathbb{E}_{s'|s,a}[\cdot]$, it would require two independent sample of s' to get an unbiased estimate (from the same (s,a)) of gradient of f (Baird, 1995).

Stationary distribution:

$$\mu : \forall s', \sum_{s,a} d^\pi(s)\pi(a|s)p(s'|s,a) = d^\pi(s') \tag{4}$$

Ergodic & Average reward: a policy π is ergodic if has a well-defined stationary distribution ρ. Under the ergodic assumption, the average reward $\eta(\pi_\theta, \pi_b) = \sum_{s,a} \rho^{\pi_b}(s)\pi_\theta(a;s)Q_\theta(s,a)$, where π_b and π_θ might be the same.

Compatible function approximation: A method that approximation of the value function is linear in "features" of the stochastic policy, as mentioned in the Theorem 2 in Sutton et al. (1999), to preserve the true gradient of the policy (unbiased).

Long term expected discounted reward (Sutton et al, 1999)

$$\eta(\pi) = \lim_{n\to\infty} \mathbb{E}(\sum_{t=1}^{\infty} \gamma^{t-1} r_t | s_0, \pi) = \sum_s d^\pi(s) \sum_a \pi(s,a)Q(s,a) \tag{5}$$

where $d^\pi(s) = \sum_{t=0}^{\infty} \gamma^t Pr(s_t = s|s_0, \pi)$ is the stationary distribution of states under π, which we assume exists and is independent of s_0 for all policies.

Policy Gradient Theorem (Sutton et al, 1999):

$$\frac{\partial \eta}{\partial \theta} = \sum_s d^\pi(s) \sum_a \frac{\partial \pi(s,a)}{\partial \theta} Q^\pi(s,a) = \mathbb{E}_{s\sim\rho^\pi, a\sim\pi_\theta}[\nabla_\theta \log \pi_\theta(a|s) Q^\pi(s,a)] \tag{6}$$

It reduces computation to sampling. **Remark 1**: there is not term about $\frac{\partial\, d^{\pi_\theta}}{\partial\theta}$, the effect of policy changes on the distribution of states does not appear, which is convenient for sampling (Agrawal, 2018). **Remark 2**: The gradient $\nabla_\theta \pi(a|s)$ is the direction to increase the probability of action a according to function $\pi(\cdot|s)$ (Degris et al., 2013).
True Gradient: the expected value of the approximated gradient equal to the true gradient of the objective function, i.e. unbiased. Using function approximation will cause the bias. People often avoid bias by compatible function approximation.
Fisher Information & Fisher-Rao metric : $F = \langle \partial_i \log p(\xi;\theta) \partial_j \log p(\xi;\theta) \rangle_{p(\xi;\theta)}$, which could also be interpreted as second-order Taylor expansion of the KL-divergence [20].
Off-Policy Actor-Critic (Degris et al., 2012b):

$$\eta_\beta(\pi_\theta) = \int_S d^\beta(s) V^\pi(s) ds = \int_S \int_A d^\beta(s) \pi_\theta(a|s) Q^\pi(s,a) da ds \tag{7}$$

$$\nabla \eta_\beta(\pi_\theta) = \int_S \int_A d^\beta(s) \nabla \pi_\theta(a|s) Q^\pi(s,a) da ds = \mathbb{E}_{s\sim d^\beta, a\sim\beta(\cdot|s)}[\frac{\pi_\theta(a|s)}{\beta(a|s)} \nabla_\theta \log \pi_\theta(a|s) Q^\pi(s,a)] \tag{8}$$

3 Survey

I will first present two recent entropy-regularized methods, one provides the first **convergence guarantee** for general **nonlinear function approximation** (Dai et al, 2018), which is called *Smoothed Bellman Error Embedding (SBEED) algorithm*, and one called *Soft Actor-Critic*. Both of them claims to have beaten the benchmark methods. I will also introduce some benchmarks of KL-Divergence regularized methods such as Trust Region Policy Gradient (TRPO), Proximal Policy Gradient (PPO), Relative Entropy Policy Search (REPS), and EM-based algorithm Reward-Weighted Regression, and deterministic methods including Deterministic Policy Gradient (DPG) and Deep Deterministic Policy Gradient (DDPG).

There are three advantages of policy gradient methods (Sutton & Barto, 2018): policy representations can be chosen to adapt tasks by injecting knowledge about the desired form of the policy, incorporate domain knowledge so that the policy may be a simpler function to approximate. However, generally the sample efficiency is poor.

Policy gradient algorithms are mainly based on the policy gradient theorem (Sutton et al., 1999) and the extensions of it. DPG extended the theorem to a more specific, Deterministic Policy Gradient Theorem with compatible function approximation. Policy gradient algorithms always come with the Actor-Critic structure, which uses state-value functions in the policy improvement step, while incrementally updating value functions through Temporal-Difference methods. Although a critic (value function) is not required for action selection, it can help learn the policy parameters, for example, reducing variance by using approximated value as baselines.

Some algorithms introduced below, involving actor-critic structure, depend on semi-gradient approximation on TD-learning of value function. Why don't use Monte Carlo approximation, which is the true estimate of the value function? Because semi-gradient approximation enables faster convergence. However, TD might converge to a local optimum rather than global one (Sutton et al., 2018).

Comparative Discussion:Two types of algorithms currently dominates the continuous policy control problems: First, Trust-Region Policy Optimization (Schulman et al., 2015) and the its derivation Proximal Policy Optimization algorithms (Schulman et al., 2017). These policy-gradient algorithms are on-policy, batch learning, and updating with constraints. They are robust, applicable to high-dimensional problems, easy to implement, and require moderate parameter tuning. However, as **on-policy** algorithms, they suffer from poor sample efficiency (Haarnoja et al., 2018). The on-policy has worse sample efficiency than off-policy because off-policy allows experience-replay, or bootstrapping.

In contrast, off-policy policy-gradient algorithms such as the Deep Deterministic Policy Gradient (DDPG, Silver et al. 2014), and Smoothed Bellman Error Embedding (SBEED, Dai et al. 2018) learn by experience replay. They have much better data efficiency. However, DDPG might be difficult to tune, and are extreme brittle and hyperparameter sensitive (Haarnoja et al., 2018).

It's hard to say which method is theoretically better. However, empirically, PPO and DDPG used to be the state-of-the-art, and recently some better methods has been proposed and will be tested by time.

3.1 Entropy-Regularized Policy Search

As SBEED and Soft Actor-Critic (SAC) are new and both claimed to have beaten existing benchmark methods such DDPG, PPO, I put more emphasis on introducing them. They both base on maximum entropy framework, which augments the standard maximum reward reinforcement learning objective with an entropy maximization term, as Haarnoja et al (2018) pointed. While SBEED interprets entropy as shaping reward (Dai et al., 2018), SAC interprets maximizing entropy by encouraging exploration (Haarnoja et al., 2018).

3.1.1 SBEED

Dai et al. (2018) introduced a novel reinforcement learning optimization algorithm, SBEED, that can apply to any **differentiable** function class. It reformulated the Bellman equation into a novel primal-dual optimization problem, and

then designed a new algorithm based on a recent work on nonconvex stochastic programming (Ghamidi et al., 2013). SBEED learns the optimal value function and a stochastic policy in the primal, and the Bellman error in the dual.

They reformulated the Bellman equation into a novel primal-dual optimization problem using Nesterov's smoothing technique (Nesterov, 05), which turns the Bellman equation into

$$V_\lambda(s) = \max_{\pi(\cdot|s)\in P_{\mathbb{A}}} \Big(\mathbb{E}_{a\sim\pi(\cdot|s)}(R(s,a) + \gamma\mathbb{E}_{s'|s,a}[V_\lambda(s')]] + \lambda H(\pi, s)\Big) \tag{9}$$

where $H(\pi, s) := -\sum_{a\in A}\pi(a|s)\log\pi(a|s)$ is the enropy, and $\lambda \geq 0$ controls the degree of smoothing.

They also used the log-sum-exp (Boyd & Vandenberghe, 2004) smoothing approximation to the Bellman operator, which is a γ-contraction with bias under control and has temporal consistency :

$$V_\lambda(s) = (T_\lambda V_\lambda)(s) := \lambda\log\Big(\sum_{a\in A}\exp\Big(\frac{R(s,a) + \gamma\mathbb{E}_{s'|s,a}[V_\lambda(s')]}{\lambda}\Big)\Big) \tag{10}$$

Important Note: The reason why it is applicable to off-policy setting: The T_λ operator has a **unique fixed point** $(V^*_\lambda, \pi^*_\lambda)$ for equation (9), and they satisfies the **temporal consistency** proposition

$$V(s) = R(s,a) + \gamma\mathbb{E}_{s'|s,a}[V_\lambda(s')]] - \lambda\log\pi(a|s) \tag{11}$$

Then they proposed a new objective function, *mean squared consistency Bellman error* based on equation (10):

$$\min_{V,\pi} l(V,\pi) := \mathbb{E}_{s,a}\Big[\Big(R(s,a) + \gamma\mathbb{E}_{s'|s,a}[V(s')] - \lambda\log\pi(a|s) - V(s)\Big)^2\Big] \tag{12}$$

The double sample problem, as mentioned in section 2, is bypassed by using the conjugate of the square function $x^2 = max_\nu(2\nu x - \nu^2)$ (Boyd & Vandenberghe,2004), as well as the interchangeability principle, and also with substitution of dual function, which lead to the final primal-dual formulation:

$$\min_{V,\pi}\max_{\rho\in F_{S\times A}} L_\eta(V,\pi;\rho) := \mathbb{E}_{s,a,s'}\Big[(\delta(s,a,s') - V(s))^2\Big] - \eta\mathbb{E}_{s,a,s'}\Big[(\delta(s,a,s') - \rho(s,a))^2\Big] \tag{13}$$

where $\delta(s,a,s') := R(s,a) + \gamma V(s') - \lambda\log\pi(a,s)$. The extra variance is canceled by the second term, an auxiliary function. This is an important idea. And it turned $\mathbb{E}_{s'|s,a}[V(s')]$ from the objective function into the solution of ρ, the auxiliary function.

With the parameterization of (V,π,ρ), we have parameters $w = (w_V, w_\pi, w_\rho)$, the idea of the solution is as follows:1.solve the inner (dual) problem by standard least-squares regression with parameter w_ρ. 2.solve the outer (primal) problem by stochastic mirror descent (Nemirovski et al, 09).

Comparative Discussion: There are several advantages of SBEED (Dai et al., 2018): most important advantage, it provides **convergence guarantee** for off-policy training; the smoothing overcomes the non-smoothness caused by the max operator in the objective function; the double sample issue is bypassed, while TRPO can only bypass it in simulation environments; it works for off-policy training and hence yield high sample efficiency. However, the mirror gradient descent might be numerically heavy. SBEED is also relevant to TRPO and Natural Policy Gradient, as the updating of the w_π, the parameter of policy.

3.1.2 Soft Actor-Critic

Soft Actor-Critic is the first off-policy actor-critic method in the Maximum Entropy Reinforcement Learning framework proposed by Haarnoja et al. (2018).

Soft actor-critic, an off-policy actor-critic deep RL algorithm based on the maximum entropy reinforcement learning framework. Compared with DDPG, it has a **stochastic** actor. In this framework, the actor aims to maximize expected reward while maximizing entropy (Haarnoja et al., 2018). The objective function:

$$J(\pi) = \sum_{t=0}^{T-1}\mathbb{E}_{(s_t,a_t)\sim\rho_\pi}[r(s_t,a_t) + \alpha H(\pi(\cdot|s_t))]$$

The maximum entropy objective has a number of conceptual and practical advantages. First, the policy is encouraged to explore more, and avoid unpromising actions. Second, the policy can capture multiple modes of near-optimal behavior (Haarnoja et al., 2018).

In the policy improvement step, the policy is updated towards the exponential of the new Q-function. This choice of update can be guaranteed to result into an improved policy in terms of its soft value (Haarnoja et al., 2018). To restrict policy to a desired set, they use KL-divergence based projection.

Comparative Discussion: Empirically, the method outperforms the state-of-the-art deep reinforcement learning methods, including off-policy DDPG and on-policy PPO. It also has a far better sample efficiency than DDPG. (Haarnoja et al., 2018).

3.2 KL-Divergence Regularized Policy Search

Natural gradient was first applied to policy gradient methods by Kakade (2002), and subsequently shown in preliminary work to be the **true** natural policy gradient (Bagnell& Schneider, 2003). Intuitively, the natural gradient is the steepest direction that the policy (distribution) changes.

In this section, I will summarize the works inspired by Natural Policy Gradient (Kakade, 2002) and Approximately Optimal Approximate Reinforcement Learning (Kakade&Langford, 2002), also by Natural Actor Critic, which is proposed by Peters et al. (2005), which combines actor-critic with natural gradient. More specifically, the works on Natural Policy Gradient, Trust Region Policy Optimization, Proximal Policy Optimization, and Relative Entropy Policy Search algorithms.

It worth to notice that (Schulman et al., 15) unified some of these methods, showing that policy iteration update, standard policy gradient and natural policy gradient are special cases of TRPO.

Comparative Discussion: TRPO combines the natural policy gradient with trust region optimization, which involves second order optimization by calculating the Hessian matrix of the KL-Divergence (Schulman et al., 2015). PPO has improved over TRPO by using only first order optimization without calculating natural gradients (Schulman et al., 2017). Wu et al. (2017) also improved the computational efficiency on TRPO by an approximation to the Fisher matrix.

Comparative Discussion: REPS constrains the state-action marginals $p(s, a)$, while TRPO constrains the conditionals $p(a|s)$. Unlike REPS, TRPO does not require a costly nonlinear optimization in the inner loop (Schulman et al., 2015).

3.2.1 Natural Policy Gradient

This method calculates the steepest descent direction based on the underlying structure of the parameter space.

The main idea under Natural Policy Gradient (Kakade, 2002) is maximize the average reward, by properly measuring the closeness between the current policy and the updated policy based on the distribution of the paths.

The author followed the same **compatible function approximation** with Sutton et al (1999): $\nabla \log \pi(a; s, \theta) = \phi^{\pi}(s, a), Q(s, a; w) = w^t \phi^{\pi}(s, a)$. It worth to notice that w is the natural gradient we need in this specific setting.

The function approximation of the Q function is of linear structure, and it is compatible with the policy. It results in an **unbiased** gradient. This directly leads to the Fisher Information matrix in the policy gradient (Peters et al., 2005).
Exact gradient of the average reward: $\nabla \eta(\theta) = \sum_{s,a} \rho^{\pi}(s) \nabla \pi(a; s, \theta) Q^{\pi}(s, a) = F_{\theta} w$.

The method takes steepest descent direction of $\eta(\theta)$. Intuitively, $F(\theta)$ measures the distance on a probability manifold corresponding to state s and $F(\theta)$ is the average of such distance. The steepest descent direction this gives is:

$$\tilde{\nabla} \eta(\theta) := F(\theta)^{-1} \nabla \eta(\theta) \tag{14}$$

The Fisher information Matrix, in the natural policy gradient, is the metric tensor for the policy space. Peters et al. (2005) further proved that F_{θ} is indeed the true Fisher information matrix, rather than 'average Fisher information matrix'.

Peters et al. (2005) further combined Actor-Critic with Natural Policy Gradient. The actor updates are based on natural policy gradient, while the critic obtains both the natural policy gradient and additional parameters of a value function simultaneously by linear regression.

Comparative Discussion: Direct gradient descent on parameterized policies does not consider the underlying structure of the policy (distribution), or manifold, because distance in parameter space does not equivalent to distance in policy space, while Natural Gradient takes the underlying structure into consideration. More formally, the natural gradient solves the non-covariance issue that re-parameterization of the policy leads to a different gradient direction (Bagnell & Jeff Schneider, 2003), hence it is more stable and more reasonable. As the natural gradient analytically averages out the influence of the stochastic policy (including the baseline of the function approximator), it requires fewer data point for a good gradient estimate than 'vanilla gradients'(Peters et al., 2005). However, the inversion and formation of the Fisher matrix is computationally prohibitive.

Comparative Discussion: Both depending on KL-divergence and trying to develop a natural gradient, Natural Policy Gradient (Kakade, 2002) works on the space of **policies**, while Covariant Policy Search (Bagnell & Scheider, 2003) works on the **path-distribution** manifold. Though the starting point are different, the Covariant Policy Search agrees with the heuristic suggested by Kakade in the infinite horizon case (Bagnell & Scheider, 2003).

3.2.2 Trust Region Policy Gradient

Kakade and Langford (2002) proposed a policy updating scheme called conservative policy iteration, for which they could provide explicit lower bounds on the imporvement of η.

TRPO (Schulman et al., 2015) is an extension of the work (Kakade&Langford, 2002). Rather than mix the policies of form $\pi^{new} = (1 - \alpha)\pi + \alpha\pi'$ to update the policy, TRPO updates parameterized policy here. It is an iterative procedure for policy optimization, with guaranteed monotonic improvement. It is similar to natural policy gradient methods and it is effective for optimizing large nonlinear policies such as neural networks.

And there is an identity expresses the expected discounted return of another policy $\tilde{\pi}$ in terms of the advantage over π:

$$\eta(\tilde{\pi}) = \eta(\pi) + \sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a|s) A_\pi(s,a) = const + \sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a|s) A_\pi(s,a) \tag{15}$$

where $A_\pi(s,a) = Q_\pi(s,a) - V_\pi(s), \rho_{\tilde{\pi}}(s) = P(s_0 = s) + \gamma P(s_1 = s) + \gamma^2 P(s_2 = s)...$
The author introduced a local approximation to η, the surrogate objective that ignores the change in state distribution:

$$L_\pi(\tilde{\pi}) = \sum_s \rho_\pi(s) \sum_a \tilde{\pi}(a|s) A_\pi(a,s) = \mathbb{E}_{s\sim\pi, a\sim\tilde{\pi}}[A_\pi(a,s)] = \mathbb{E}_{s\sim\pi, a\sim\pi}[\frac{\tilde{\pi}(a|s)}{\pi(a|s)} A_\pi(s,a)] \tag{16}$$

Then the authors derived a lower bound of updated policy with respect to old policy based on KL divergence between them, which could **guarantee** the true objective η is **non-decreasing** with updating. The maximization of the lower bound involves penalty on the KL divergence. The authors approximated the maximization problem by using the constraint on average KL divergence rather than penalty to allow robust large updates, in a parameterized manner on policies $\pi(a|s) = \pi_\theta(a|s)$. The optimization problem, after some transformation, and replace the sum over the actions by an importance sampling estimator, we have the optimization problem:

$$\max_\theta \mathbb{E}_{s\sim\rho_{\theta_{old}}, a\sim\theta_{old}}\Big[\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)} A_{\theta_{old}}(s,a)\Big] \quad s.t.\ \mathbb{E}_{s\sim\rho_{\theta_{old}}}\Big[D_{KL}(\pi_{\theta_{old}}(\cdot|s), \pi_\theta(\cdot|s)) \leq \delta\Big] \tag{17}$$

Note: zeroth and first-order terms for $\bar{D}_{KL}$ are zero at θ_k, which is approximated by Fisher Information Matrix. In other words, Fisher Information matrix is the local quadratic approximation to the KL divergence. The advantage functions here are Q-value functions, which are approximated by unbiased Monte Carlo estimation.

The constrained problem can efficiently be approximately solved using the conjugate gradient algorithm, after making a linear approximation to the objective and a quadratic approximation to the constraint (Schulman et al., 2015).
Discussion: TRPO has robust performance on on-policy problems. However, there are several disadvantages: the vine methods to solve the double-sample problem can only work under simulation environments; no guarantee for convergence; it requires repeated computation of Fisher vector products, preventing it from scaling to the larger architectures typically used in experiments on learning from image observations in Atari and MuJoCo (Wu et al., 2017); it requires a large batch of rollouts in order to accurately estimate curvature (Wu et al., 2017).

3.2.3 Proximal Policy Gradient

PPO (Schulman et al., 2017) is just an extension of TRPO. Recall that TRPO maximizes a "surrogate" objective

$$L^{CPI}(\theta) = \hat{\mathbb{E}}_t\Big[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta old}(a|s)} \hat{A}_t\Big] = \hat{\mathbb{E}}_t\Big[r_t(\theta)\hat{A}_t\Big] \tag{18}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta old}(a|s)}$. The superscript refers to conservative policy iteration.
The paper proposed a new objective function,

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\Big[min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\Big] \tag{19}$$

where ϵ is a hyperparameter.
Intuitively, L^{CLIP} is a lower bound of L^{CPI}. With this scheme, the authors only ignored the change in probability ratio when it would make the objective improve, and they included it when it makes the objective worse. It worth to note that it uses SGD to optimize the surrogate loss.

An adaptive KL penalty coefficient method was also proposed, but the experiment results are worse than clipped method.
Comparative Discussion: PPO is a family that approximately enforce KL constraint without computing natural gradients, instead, it computes stochastic gradient descent (first order). It has the good properties of TRPO, and further improved over TRPO. It has robust performance, computation efficiency, applicable in joint architecture compared with TRPO, and is easy to implement (Schulman et al., 2017). However, it is still an on-policy method, which could suffer from data inefficiency.

3.2.4 Actor Critic using Kronecker-Factored Trust Region

Wu et al. (2017) proposed a method that combines a recent technique called Kronecker-factored approximate curvature uses a Kronecker-factored approximation to the Fisher matrix to perform efficient approximate natural gradient updates, which allows the Fisher matrix of the gradient to be inverted efficiently.

It is more tailored to the structure of neural networks, please refer to Section 2.2 of (Wu et al., 2017) for details. The approximation of the Fisher Information matrix can be interpreted as making the **assumption** that the second-order statistics of the activations and the backpropagated derivatives are uncorrelated (Wu et al., 2017).

With the good properties of TRPO, it is more scalable than TRPO as each update is comparable in cost to an SGD update, and it allows using small batches. It worth to notice that the efficiency improvement comes from the approximation of the Fisher matrix, i.e. sacrificing of the accuracy of the gradient. It is also empirically better than TRPO (Wu et al., 2017).

3.2.5 Relative Entropy Policy Search

The REPS (Peters et al., 2010) work is mainly built on Bagnell nad Schneider's (2003) classification that the constraint introduced in (Kakade 2002) can be treated as Taylor expansion of the **loss of information** or **relative entropy** between the path distributions generated by the original and updated policy. The optimal control problem is that maximizing discounted expectation of total rewards while bounding the loss in information by a maximal step size. It worth to notice that there is no parameterized policy here. It's applicable to the off-policy setting, but not much detail is presented.

The agent may converge to a stationary distribution as defined in equation (3). While maximizing the expected return based on all observed samples, the method also bounds the loss of information measured using relative entropy between the observed data distribution $q(s,a)$ and the data distribution $p^\pi(s,a) = d^\pi(s)\pi(a|s)$ generated by the new policy π, and ϵ is the maximal information loss.

The problem

$$\max_{\pi, d^\pi} \eta(\pi) = \sum_{s,a} d^\pi(s)\pi(a|s) r_{s,a}$$

$$s.t.\ \ D_{KL}(p^\pi \| q) = \sum_{s,a} d^\pi(s) \log \frac{d^\pi(s)\pi(a|s)}{q(s,a)} \leq \epsilon; \sum_{s,a,s'} d^\pi(s)\pi(a|s)p(s'|s,a)\phi_{s'} = \sum_{s'} d^\pi(s')\phi_{s'}; \sum_{s,a} d^\pi(s)\pi(a|s) = 1$$

where $\phi_{s'}$ is the feature vector, and the first constraint limits the information loss. It worth to notice that without the information loss bound constraint, the Bellman equation can be derived from Lagrangian. Then the policy is calculated by the combination of dual function of the function above and actor-critic framework.

The method is further developed by replacing $q(s,a)$ by sample average, to further reduce the method into a model-free setting. The π here could be parameterized, but not introduced in Peters et al. (2010)'s paper. The parameterized version was introduced by Deisenroth et al. (2013).

According to Duan et al. (2016)'s test, this method's empirical performance is not good on Walker test. I mention it here to broaden the horizon of the paper. The computation efficiency is higher than other methods, but the performance is worse.

3.3 Reward-Weighted Regression

This kind of methods mainly based on Expectation-Maximization, which was first discovered by Dayan and Hinton (1996). Peters and Schaal (2007) found that reinforcement learning can be reduced to a reward-weighted nonlinear regression problem.

Why it is called 'Reward-weighted regression'? It is because during the E-step of the EM, a re-weighting distribution based on rewards is proposed.

The objective function here is still the long term expected reward, which is approximated by the sampling. They established a lower bound for expected return:

$$\log J_u(\theta) = \log \sum_{i=1}^n \pi_\theta(a_i|s_i) u_\tau(r_i) = \log \sum_{i=1}^n q(i) \frac{\pi_\theta(a_i|s_i) u_\tau(r_i)}{q(i)} \geq \sum_{i=1}^n q(i) \log \frac{\pi_\theta(a_i|s_i) u_\tau(r_i)}{q(i)}$$

$$= \sum_{i=1}^n q(i)[\log \pi_\theta(a_i|s_i) + \log u_\tau(r_i) - \log q(i)]$$

Here, $\sum_{i=1}^n q(i) = 1$, a re-weighting distribution; the reward are transformed (better adaptive) by a monotonic non-negative transformation u; the sample is generated by the current policy θ. It worth to notice that the solution of Lagrange function gives the updating rule of EM algorithm. The algorithm is computationally efficient.

Through iteratively maximization of the lower bound, it's guaranteed that the policy update steps are improvements, and the algorithm will converge to a local optimal solution.

The method is not popular, but it provides a different view of the RL problem, i.e. the transformation of reinforcement learning problem into an efficient supervised learning problem. It is efficient, learns smoothly without dangerous jumps in solution space, and works well in application of complex robotic problems (Peters & Schaal, 2007).

3.4 Deterministic methods

The above methods assuming policies are stochastic. Rather than working on distribution of actions, this class of methods work on actions directly. It mainly based on actor-critic framework, TD-learning, the idea of Policy Gradient Theorem, and the compatible function approximation (Sutton et al., 1999). This class is also heavily based on the contribution of Degris et al. (2012), the first off-policy actor-critic framework.

The deterministic policy gradient is the expected gradient of the action-value function, which is the limiting case, as policy variance tends to zero, of the stochastic policy gradient (Silver et al., 2014). The computational complexity are no worse than prior methods. And the deterministic methods are especially helpful in many applications where a differentiable control policy is provided and stochastic policy gradient is inapplicable (Silver et al., 2014).

In the stochastic case, the policy gradient integrates over both state and action spaces, whereas in the deterministic case it only integrates over the state space. As a result, computing the stochastic policy gradient may require more samples, especially if the action space has many dimensions (Silver et al., 2014).

3.4.1 Deterministic Policy Gradient

DPG (Silver et al., 2014) is based on actor-critic framework and TD-learning. It introduces deterministic policy, with a new gradient updating rule. There are also some details about the local linear structure of the Q-value function (critic). The on-policy setting is relatively easy. The off-policy setting is based on Degris et al. (2012b), Off-Policy Actor Critic, the first actor-critic algorithm for off-policy reinforcement learning. It worth to notice that the objective is maximizing $\eta_\beta(\pi)$. Please refer to equation (7),(8).

For off-policy deterministic actor-critic, the calculation of δ_t is based on $\mu_\theta(s_{t+1})$, i.e. the next action a' does not follow the behavior policy in calculation. This ensures the update of critic toward V_*. In contrast, a' in the sample is selected by the behavior policy. There is no importance sampling in the gradients because Q-learning and deterministic behavior.

In this framework, the $Q(s,a)$ is parametereized by θ^Q and θ^μ. More precisely, $Q(s, \pi_{\theta^\mu(s)}|\theta^Q)$.
The policy gradient is calculated as

$$\nabla_{\theta^\mu}\eta_\beta \approx E_{s_t\sim d^\beta}[\nabla_a Q(s,a|\theta^Q)|_{s=s_t,a=\mu(s_t)}\nabla_{\theta_\mu}\mu(s|\theta^\mu)|_{s=s_t}]$$

and the critic gradient is calculated by minimization of Bellman error.
Linear function approximation is used for local critic. It represents the local advantage of deviating from the current policy. As a result, a linear function approximator is sufficient to select the **direction** in which the actor should adjust its policy parameters. Considering off-policy Q-learning may diverge when using linear function approximation, the author also mentioned the gradient temporal-difference learning by working on projected Bellman error rather than Bellman error. The natural gradient can also be extended to the deterministic case (Silver et al., 2014).

3.4.2 Deep Deterministic Policy Gradient

While DQN solves problems with high-dimensional state spaces, as it relies on greedy policy, it is difficult for DQN (Mnih et al., 2013) to handle large-space action space, let alone continuous action space. DDPG is the combination of DPG and DQN. DQN allows it to use neural network function approximators to learn in large state and action spaces online. DDPG uses slowly changing copy of value-function and policy networks to stabilize the training (my understanding is avoid semi-gradient issue). And the authors mentioned batch normalization to automatically scale features, and adding noise to the policy to encourage exploration.

It has the good properties of DPG. It is stable, relative sample efficient, empirically converges fast, and it allows sharing of variables for actor and critic networks. As a model free method, it still requires huge amount of data. Though using a Q-function estimator enables off-policy learning, the interplay between the deterministic actor network and the Q-function typically makes DDPG extremely difficult to stabilize and brittle to hyperparameter settings. As a consequence, it is difficult to extend DDPG to very complex, high-dimensional tasks, which makes DDPG more suitable for the on-policy gradient methods (Haarnoja et al., 2018).

4 Analysis

4.1 Open problem

Is it possible to have a unified framework for all reinforcement learning algorithms? There are some attempts, e.g. Nachum et al. (2017) attempted to bridge the gap between value and policy based reinforcement learning.

Why don't apply other kind of norm to RL? Current main stream is l_2-norm.

4.2 State-of-the-art

SBEED, as introduced in the Section 3.1, is data-efficiency, provably convergent even when **nonlinear function approximation** is used on off-policy samples. Empirical study shows the proposed algorithms achieves superior performance compared to state-of-the-art baselines on several MuJoCo control tasks (Dai et al. 2018).

Recently, Soft Actor-Critic has been proposed by Haarnoja et al. (2018), and as the experiments shows in the paper, the method does in fact exceed the performance of state-of-the-art deep reinforcement learning methods, including off-policy DDPG and on-policy PPO. It also has a far better sample efficiency than DDPG.

Recently, for the KL-Divergence Constraints for Policy Search line, Abdolmaleki et al. (2018) proposed Maximum a Posteriori Policy Optimisation, which benefits from the good properties of Trust Region class and Deterministic Policy Gradient class. It shows the scalability, robustness of on-policy algorithms, while offering the data-efficiency of off-policy, value-based methods (Abdolmaleki et al., 2018). The paper is not introduced in this review.

5 Conclusion

5.1 Learned

Despite the knowledge I have learned above, I have learned the authors' research methodologies, which I think will benefit my future career. My paper reading skills, and paper search skills have also been greatly improved.

First and most important, to do a good research, people should totally understand the previous work and form a big picture. We should understand the line, or network, of the development of a method. For example, only if we understand the methods' assumptions, can we relax or change them to lead to innovations. As I have summarized in the early part of the survey, most of the works are building on the previous people's framework or idea. Their proofs, intuitions, ideas are somewhat similar. For example, TRPO strictly follows the policy improvement theorem in (Kakade & Langford, 02), doubted the whether it is necessary to mix the policy, and adapted the method with natural gradient and trust region method. Another example, the root of DPG is based on (stochastic) policy gradient theorem and compatible function approximation, which is the work of Sutton (1999).

Obtain a qualitative view of a method is sometimes more important than understanding the details, as we don't need to fully understand a method when it is not the focus of our research. However, we should know how to use it, and its advantage, disadvantage, and assumptions.

Interpret a problem into an existing well-learned framework is important. For example, forming some reinforcement learning SGD problems into primal-dual problems has led to new developments of the optimization algorithms (Dai et al., 2018), and forming the reinforcement learning as probabilistic inference (Levine, 2018). Read more!

Due to limited time, I won't be able to give a thorough review of the topic, but I have formed a big picture of the history and the current stage of the area. I will further thoroughly read Sergey Levine and John schulman's recent work, and see if there is any exciting idea.

5.2 Future Research

I recommend **nonlinear function approximation** in RL as the future research direction. As Dai et al. (2018) has initiated the first nonlinear function approximation algorithm with convergence guarantee.

As Haarnoja et al. (2018) pointed out, several recent papers have noted the connection between Q-learning and policy gradient methods in the framework of maximum entropy learning, which could be an interesting area.

As the measure of probability distance is fundamental to RL framework, I believe incorporate other distance measure into existing methods might be interesting, such as Wasserstein distance. But it worth to notice that KL divergence has a very close relationship with log-likelihood and Fisher-Information, while Wasserstein distance does not have such a strong statistical property.

I also recommend the direction of research that casting Reinforcement Learning (RL) as an inference problem. Most of the state-of-the-art policy gradient algorithms falls into this category (Levine, 2018), for example, TRPO. I feel like most of the methods I have mentioned above, involving Entropy or KL-divergence, and EM-based algorithm, could be unified by the variational inference framework (Levine, 2018). And I would recommend this survey to the readers.

Haarnoja et al. (2018) suggest that entropy maximizing reinforcement learning algorithms can provide a 'promising avenue' for improved robustness and stability, and further exploration of maximum entropy methods, including methods that incorporate second order information (e.g., trust regions (Schulman et al., 2015)) or more expressive policy classes would be an interesting direction (Haarnoja et al, 2018).

Dai et al. (2018) mentioned that, although it is not the focus of the paper, for current linear function approximation in RL, several gradient-based algorithms can be interpreted as solving a primal-dual problem. This insight has led to novel, faster, and more robust algorithms by adopting sophisticated optimization techniques.

Wu et al. (2017) also mentioned that the designing of low-variance and low-bias gradient estimates is an active line of research.

References

[1] Richard S. Sutton, Andrew G. Barto, Reinforcement Learning: An Introduction, 2nd edition, 2018.

[2] Dai, B., Shaw, A., Li, L., Xiao, L., He, N., Chen, J., and Song, L. SBEED: Convergent Reinforcement Learning with Nonlinear Function Approximation, ICML 2018. .

[3] Nesterov, Yu. Smooth minimization of non-smooth functions. Mathematical programming, 103(1):127-152, 2005. .

[4] http://rail.eecs.berkeley.edu/deeprlcourse-fa17/f17docs/lecture_13_advanced_pg.pdf, .

[5] Kakade, Sham. A natural policy gradient. In NIPS, pp. 1531-1538, 2002.

[6] Kakade, Sham and Langford, John. Approximately optimal approximate reinforcement learning. In ICML, volume 2, pp.267–274, 2002.

[7] Peters, J., Vijayakumar, S., Schaal, S. (2005a). Natural actor-critic. In Proceedings of the European machine learning conference (pp. 280–291)

[8] Schulman, John, Levine, Sergey, Abbeel, Pieter, Jordan, Michael I, and Moritz, Philipp. Trust region policy optimization. In ICML, pp. 1889-1897, 2015.

[9] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017b. .

[10] Bagnell, J., and Schneider, J. 2003. Covariant policy search. In International Joint Conference on Artificial Intelligence.

[11] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller. Deterministic policy gradient algorithms. In Int. Conf on Machine Learning, 2014.

[12] R. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In Neural Information Processing Systems 12, 1999.

[13] Boyd, Stephen and Vandenberghe, Lieven. Convex Optimization. Cambridge University Press, Cambridge, England, 2004.

[14] Nemirovski, Arkadi, Juditsky, Anatoli, Lan, Guanghui, and Shapiro, Alexander. Robust stochastic approximation approach to stochastic programming. SIAM Journal on optimization, 19(4):1574-1609, 2009.

[15] Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. In Neural Information Processing Systems 12, pages 1057–1063.

[16] Shipra Agrawal. Lecture 5: Policy gradient methods. "https://ieor8100.github.io/rl/docs/Lecture%205%20-%20policy%20gradients.pdf"

[17] Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, Graves, Alex, Antonoglou, Ioannis, Wierstra, Daan, and Riedmiller, Martin. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602, 2013.

[18] Degris, T., White, M., and Sutton, R. S. (2012). Off-policy actor-critic. In 29th International Conference on Machine Learning.

[19] Yuhuai Wu*, Elman Mansimov*, Shun Liao, Roger Grosse, Jimmy Ba. Scalable trust-region method for deep reinforcement learning using Kronecker-factored approximation. NIPS, 2017. Spotlight.

[20] https://stats.stackexchange.com/questions/51185/connection-between-fisher-metric-and-the-relative-entropy

[21] Peters, J., Mulling, K., and Altun, Y. Relative entropy policy search. In AAAI, pp. 1607–1612, 2010.

[22] Abdolmaleki, A., Springenberg, J. T., Tassa, Y., Munos, R., Heess, N., and Riedmiller, M. (2018). Maximum a posteriori policy optimisation. In International Conference on Learning Representations (ICLR).

[23] J. Peters and S. Schaal. Reinforcement learning by reward-weighted regression for operational space control. In International Conference on Machine Learning (ICML), 2007. (Reward-Weighted Regression)

[24] Kober, J. and Peters, J. Policy search for motor primitives in robotics. In NIPS, pp. 849–856, 2009. (Reward-Weighted Regression)

[25] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, Sergey Levine. SOFT ACTOR-CRITIC: OFF-POLICY MAXIMUM ENTROPY DEEP REINFORCEMENT LEARNING WITH A STOCHASTIC ACTOR. arXiv:1801.01290, 2018.

[26] Sergey Levine. Reinforcement Learning and Control as Probabilistic Inference: Tutorial and Review. arXiv:1805.00909, 2018.

[27] Ghadimi, Saeed and Lan, Guanghui. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. SIAM Journal on Optimization, 23(4):23412368, 2013.

[28] Baird, Leemon. Residual algorithms: Reinforcement learning with function approximation. In ICML, pp. 30-37. Morgan Kaufmann, 1995.

[29] Deisenroth, M. P., Neumann, G., and Peters, J. A survey on policy search for robotics, foundations and trends in robotics. Found. Trends Robotics, 2(1-2):1–142, 2013.

6 Plan to read

Nachum, Ofir, Norouzi, Mohammad, Xu, Kelvin, and Schuurmans, Dale. Bridging the gap between value and policy based reinforcement learning. In NIPS, pp. 2772-2782, 2017.
Peters, J. and Schaal, S. Reinforcement learning of motor skills with policy gradients. Neural networks, 21(4):682–697, 2008. "A typical solution for policy gradient methods is to use the likelihood ratio gradient estimator"
Reinforcement Learning with Deep Energy-Based Policies.

7 Thanks

I am really thankful to this project. During the reading and search of the papers, my research skills have been greatly improved.

8 Random Notes for myself not counted for 8 pages

Avoiding global approximation is also a way to address the curse of dimensionality.

https://hips.seas.harvard.edu/blog/2013/01/25/the-natural-gradient/

Natural Policy Gradient is mainly based on optimization between policies (distributions). It defines a distance metric on the probability manifold through parameter space. The natural gradient is used to find the steepest director of optimization over distributions. Although distributions are parameterized, direct optimize the Euclidean distance between parameters does not make sense because of the geometry of the probability manifold. It requires special metric over one point in the probability manifold, for example, ϕ which is a parameter vector for distribution.

Intuitively, the Riemannian metric tensor describes how the geometry of a manifold affects a differential patch, $d\phi$, at the point ϕ.

Critic is used to learn the value function (TD-learning, MC), while actor is used to learn the policy (maximize the expected total reward).

Gradient TD solved projected Bellman error.

For deterministic off-policy actor-critic methods, when calculating the TD by sampling, the a' in $Q(s', a')$ is using $\pi_\theta(s')$ rather than the a' from behavior policy, because the deterministic policy gradient removes the integral over actions. (David et al., 2014)

For stochastic off-policy actor-critic methods, when calculating the TD, the importance sampling ratio is used for $Q(s', a')$.

I think the reason is the ratio $\pi_\theta(a')/\pi_b(a')$.

Why off-policy actor-critic works? I think this is because the critic updates as actor updates. The state visiting probability depend on behavior policy, but the value function depends on target policy.

Off policy control LSPI.

The objectives of value-function and policy are different. The first is always minimize the Bellman error, while the policy gradient methods always trying to maximize the discounted total rewards.

The core idea of algorithm design is turn expectation (sum over probability*return) into sampling (and removing the sum).

State-aggregation is interesting.

Kernel methods are generalized version of linear regression. The kernel functions assign weights to distances between two feature vectors. The feature vectors are mapped by radial basis functions from states.

TD is solely for value function approximation (prediction), as it is based on the Bellman error. Bellman error is a relationship of value function or action value function. It does not change policy.

Why don't use l1 Bellman error rather than l2 Bellman error?

Reduce variance by auxiliary function from SBEED.

Second order approximation of FIM:

$$D(p_{\theta'}(\tau)\|p_\theta(\tau)) = \frac{1}{2}(\theta' - \theta)^T F(\theta)(\theta' - \theta)$$

The logic is, policy iteration works, then TD works. Follows the SAC paper.

non-convex minimization, check Wu et al. (2018) section 2.2 for details.

Reward-Weighted Regression equation constraint also used in Lagrangian.

Why off-policy works? Because in $\delta_t, Q(s_{t+1}, a_{t+1})$, the a_{t+1} is selected by the target policy rather than behavior policy.

9 Question

For sbeed, why $E_{s,a,s'}$ and $E_{s,a}$ are different? It is directly relevant to why the double-sample issue is bypassed.

SBEED, temporal consistency, the v(s) same for all actions?