1 Policy Gradient Methods

1.1 Policy Approximation and its Advantages

1.2 The Policy Gradient Theorem

1.3 REINFORCE: Monte Carlo Policy Gradient

1.4 REINFORCE with Baseline

1.5 Actor-Critic Methods

1.6 Policy Gradient for Continuing Problems

1.7 Policy Parameterization for Continuous Actions

1.8 Summary

1.9 Learning Objectives (UA RL MOOC)

Lesson 1: Learning Parameterized Policies

1. Understand how to define policies as parameterized functions

Parameterized policy: $\pi(a|s, \boldsymbol{\theta})$. Parameterized value function: $\hat{q}(s, a, \boldsymbol{w})$

Constraints on the policy parameterization

- $\pi(a|s, \boldsymbol{\theta}) \geq 0 \quad \forall a \in \mathcal{A}, s \in \S$
- $\sum_{a \in \mathcal{A}} \pi(a|s, \boldsymbol{\theta}) = 1 \quad \forall s \in \S$

2. Define one class of parameterized policies based on the softmax function

$$\pi(a|s, \boldsymbol{\theta}) \doteq \frac{e^{h(s,a,\boldsymbol{\theta})}}{\sum_{b \in \mathcal{A}} e^{h(s,a,\boldsymbol{\theta})}}$$

h function is action preference (it can be parameterized in any way), the exponential function guarantees the probability is positive for each action. The denominator normalizes the output of each action s.t. the sum over actions is one. Note action preferences != action values; only the difference between preferences is important.

3. Understand the advantages of using parameterized policies over action-value based methods

- Parameterized policies can autonomously decrease exploration over time. Specifically, the policy can start off stochastic to guarantee exploration; as learning progresses, the policy can naturally converge towards a deterministic greedy policy
- They can avoid failures due to deterministic policies with limited function approximation
- Sometimes the policy is less complicated than the value function

Lesson 2: Policy Gradient for Continuing Tasks

4. Describe the objective for policy gradient algorithms

Our objective:

$$r(\pi) = \sum_s \mu(s) \sum_a \pi(a|s, \boldsymbol{\theta}) \sum_{s',r} p(s', r|s, a) r$$

- $\sum_{s',r} p(s', r|s, a) r$ is $\mathbb{E}[R_t|S_t = s, A_t = a]$ expected reward if we start in state s and take action a
- $\sum_a \pi(a|s, \boldsymbol{\theta}) \sum_{s',r} p(s', r|s, a) r$ is $\mathbb{E}[R_t|S_t = s]$ expected reward of state s
- $r(\pi)$ is $\mathbb{E}_\pi[R_t]$ overall average reward by considering the fraction of time we spend in state s under policy π.

Optimizing the average reward objective (policy gradient method):

$$\nabla r(\pi) = \nabla \sum_s \mu(s) \sum_a \pi(a|s, \boldsymbol{\theta}) \sum_{s',r} p(s', r|s, a) r$$

The main challenge is modifying policy changes the distribution $\mu(s)$; in

contrast, recall in $\overline{VE}$ objective, the distribution is fixed. We do gradient ascent for PG, while gradient descent in optimizing the mean squared value error.

5. Describe the results of the policy gradient theorem

Note that chain rule of gradient requires to estimate $\nabla\mu(s)$, which is difficult to estimate since it depends on a long-term interaction between the policy and the environment. The PG theorem proves we don't need that, and following is our result:

$$\nabla r(\pi) = \sum_s \mu(s) \sum_a \nabla\pi(a|s, \boldsymbol{\theta}) q_\pi(s, a)$$

$\sum_a \nabla\pi(a|s, \boldsymbol{\theta}) q_\pi(s, a)$. This is a sum over the gradients of each action probability, weighted by the value of the associated action. $r(\pi)$ takes the above expression and sum that over each state. This gives the direction to move the policy parameters to most rapidly increase the overall average reward.

6. Understand the importance of the policy gradient theorem

Lesson 3: Actor-Critic for Continuing Tasks

7. Derive a sample-based estimate for the gradient of the average reward objective

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha \frac{\nabla\pi(A_t|S_t, \boldsymbol{\theta_t})}{\pi(A_t|S_t, \boldsymbol{\theta_t})} q_\pi(S_t, A_t)$$

8. Describe the actor-critic algorithm for control with function approximation, for continuing tasks

Full algorithm see chap 13.6

- actor: parameterized policy
- critic: value functions, evaluating the actions selected by the actor

In the update rule, we don't have access to q_π, so we have to approximate it. For example, one-step bootstrap return TD method:

$$q_\pi(s, a) = R_{t+1} - \overline{R} + \hat{v}(S_{t+1}, \boldsymbol{w})$$

. To further improve, we subtract a baseline to reduces the update variance

(results in faster learning):

$$q_\pi(s,a) = R_{t+1} - \overline{R} + \hat{v}(S_{t+1}, \boldsymbol{w}) - \hat{v}(S_t, \boldsymbol{w})$$

. Note this is equals the TD error δ.

Lesson 4: Policy Parameterizations

9. Derive the actor-critic update for a softmax policy with linear action preferences

$$\boldsymbol{w} = \boldsymbol{w} + \alpha^{\boldsymbol{w}} \delta \underbrace{\nabla \hat{v}(S, \boldsymbol{w})}_{\boldsymbol{x}(s)} \quad \text{just like semi-gradient TD}$$

$$\boldsymbol{\theta} = \boldsymbol{\theta} + \alpha^{\boldsymbol{\theta}} \delta \quad \underbrace{\nabla ln \pi(A|S, \boldsymbol{\theta})}_{\boldsymbol{x}_h(s,a) - \sum_b \pi(b|s,\boldsymbol{\theta}) x_h(s,b)}$$

10. Implement this algorithm

11. Design concrete function approximators for an average reward actor-critic algorithm

12. Analyze the performance of an average reward agent

13. Derive the actor-critic update for a gaussian policy

Probability density means that for a given range, the probability of x lying in that range will be the area under the probability density curve.

Gaussian Distribution

$$p.d.f = p(x) \doteq \frac{1}{\sigma\sqrt{2\pi}} exp(-\frac{(x-\mu)^2}{2\sigma^2})$$

μ = mean of distribution, σ = standard deviation = $\sqrt{variance}$. Gaussian Policy

$$p.d.f = \pi(a|s, \boldsymbol{\theta}) \doteq \frac{1}{\sigma(s,\boldsymbol{\theta})\sqrt{2\pi}} exp(-\frac{(a-\mu(s,\boldsymbol{\theta}))^2}{2\sigma(s,\boldsymbol{\theta})^2})$$

$\mu(s, \boldsymbol{\theta})$ can be any parameterized function. σ controls the degree of exploration; usually initialized to be large s.t a wide range of actions are tried. As learning process, we expect the variance to shrink and the policy to concentrate around

the best action in each state. The agent can reduce the amount of exploration through learning.

For θ,

$$\nabla ln\pi(a|s, \boldsymbol{\theta}_\mu) = \frac{1}{\sigma(s, \boldsymbol{\theta})^2}(a - \mu(s, \boldsymbol{\theta}))\boldsymbol{x}(s)$$

For σ,

$$\nabla ln\pi(a|s, \boldsymbol{\theta}_\sigma) = (\frac{(a - \mu(s, \boldsymbol{\theta}))^2}{\sigma(s, \boldsymbol{\theta})^2} - 1)\boldsymbol{x}(s)$$

14. Apply average reward actor-critic with a gaussian policy to a particular task with continuous actions