Exercise Set 5 - Reinforcement Learning
Chapter 9,10 - Advanced policy-based methods

Instructions

This is the fifth exercise booklet for Reinforcement Learning. It covers both ungraded exercises to practice at home or during the tutorial sessions as well as graded homework exercises and graded coding assignments. The graded assignments are clearly marked.

- Make sure you deliver answers in a clear and structured format. LaTeXhas our preference. Messy handwritten answers will not be graded.
- Pre-pend the name of your TA to the file name you hand in and remember to put your name and student ID on the submission;
- The deadline for this assignment is **October 14th 2022 at 13:00** and will cover the material of chapter 7-8. All questions marked 'Homework' in this booklet need to be handed in on Canvas. The coding assignments need to be handed in separately through the codegra.de platform integrated on canvas.

Contents

9 Policy gradient methods **2**
9.1 *Exam Question: Actor-critic algorithm (partial) . . . 2
9.2 *Exam Question (partial): Policy gradient methods . . . 3
9.3 *Exam Question: Deterministic policy gradients . . . 3
9.4 Homework: Limits of policy gradients . . . 3
9.5 Homework: Coding Assignment - Policy Gradients . . . 4

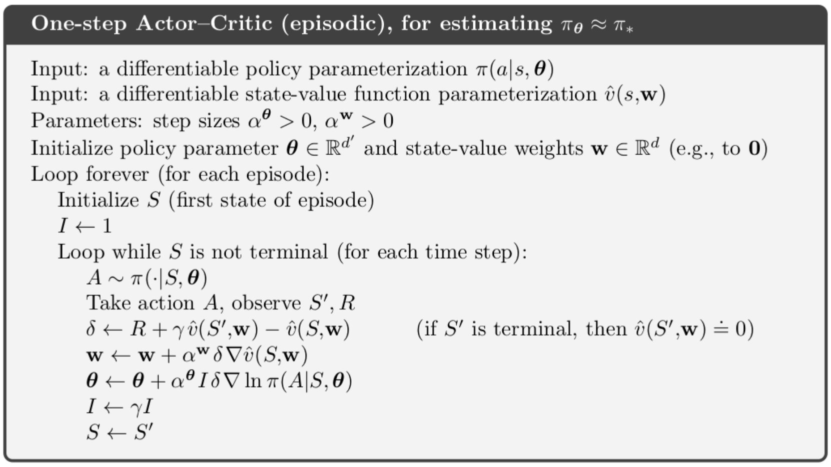
10 Advanced policy-search **5**
10.1 Natural policy gradient . . . 5
10.2 Trust Region Policy Optimization . . . 6
10.3 Homework: Update Directions . . . 6

Lecture 9: Policy gradient methods

9.1 *Exam Question: Actor-critic algorithm (partial)

Consider the actor-critic algorithm shown below:

Figure 1: Algorithm pseudo-code.



One-step Actor–Critic (episodic), for estimating $\pi_{\boldsymbol{\theta}} \approx \pi_*$

```
Input: a differentiable policy parameterization $\pi(a|s,\boldsymbol{\theta})$
Input: a differentiable state-value function parameterization $\hat{v}(s,\mathbf{w})$
Parameters: step sizes $\alpha^{\boldsymbol{\theta}} > 0$, $\alpha^{\mathbf{w}} > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ and state-value weights $\mathbf{w} \in \mathbb{R}^{d}$ (e.g., to $\mathbf{0}$)
Loop forever (for each episode):
    Initialize $S$ (first state of episode)
    $I \leftarrow 1$
    Loop while $S$ is not terminal (for each time step):
        $A \sim \pi(\cdot|S,\boldsymbol{\theta})$
        Take action $A$, observe $S', R$
        $\delta \leftarrow R + \gamma \hat{v}(S',\mathbf{w}) - \hat{v}(S,\mathbf{w})$        (if $S'$ is terminal, then $\hat{v}(S',\mathbf{w}) \doteq 0$)
        $\mathbf{w} \leftarrow \mathbf{w} + \alpha^{\mathbf{w}} \delta \nabla \hat{v}(S,\mathbf{w})$
        $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha^{\boldsymbol{\theta}} I \delta \nabla \ln \pi(A|S,\boldsymbol{\theta})$
        $I \leftarrow \gamma I$
        $S \leftarrow S'$
```

1. In the lecture, we have used a slightly different actor-critic update, namely:

$$\theta_{t+1} = \theta_t + \alpha\ \hat{q}_{\mathbf{w}}(s_t, a_t) \nabla_\theta \log \pi_\theta(a_t|s_t)$$

   There are two main differences between that update and the update in the algorithm in Figure 1 (other than the discount factor). Describe the two differences, and for each difference, say why the authors might have chosen to perform the update this way.

2. The algorithm in Figure 1 uses a discount factor γ, which we have not considered in the lecture. There is a special case where we can ignore the factor I in the policy update and get the same result (even if $\gamma < 1$). What is this case? *Hint: For a given state, the inclusion of the factor I does not change the direction of the expected gradient* $\mathbb{E}_{a\sim\pi}[I\nabla \log \pi(A|S,\theta)]$.

9.2 *Exam Question (partial): Policy gradient methods

For each of the following equations, determine whether they are unbiased estimators of the gradient $\nabla_\theta \mathbb{E}_\tau [G(\tau)]$ of undiscounted returns $G(\tau)$. Briefly explain your answer.

1. $$\mathbb{E}_\tau \left[\sum_{t'=1}^{T} r_{t'} \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \right]$$

2. $$\mathbb{E}_\tau \left[\sum_{t'=1}^{T} r_{t'} \sum_{t=t'}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \right]$$

3. $$\mathbb{E}_\tau \left[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \left(G_t(\tau) - \hat{V}_w(s_t) \right) \right]$$

4. $$\mathbb{E}_\tau \left[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) q_\pi(a_t, s_t) \right]$$

5. $$\mathbb{E}_\tau \left[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \left(G_t(\tau) - \hat{q}_w(s_t, a_t) \right) \right]$$

6. $$\mathbb{E}_\tau \left[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \hat{q}_w(s_t, a_t) \right]$$

9.3 *Exam Question: Deterministic policy gradients

1. Why can deterministic policy gradients only be used with continuous actions?

9.4 Homework: Limits of policy gradients

In this section we parameterize our policy with a univariate Gaussian probability density $\mathcal{N}(\mu(\theta_\mu), \sigma(\theta_\sigma))$ over real-valued actions. We consider a scenario with a single state where you can assume that an episode solely consists of one action and one reward. Mean and variance are learned:

$$\pi(a|s, \theta) = \frac{1}{\sigma(\theta_\sigma)\sqrt{2\pi}} \exp\left(-\frac{(a - \mu(\theta_\mu))^2}{2\sigma(\theta_\sigma)^2} \right) \tag{1}$$

1. (2 pts.) Calculate $\nabla \log \pi(a|\theta)$ w.r.t. parameters θ_σ and θ_μ for two different parametrizations. (When using normal distribution as a policy, it is common to parametrize standard deviation as an exponential, other parametrization is used for illustration purposes):

   (a) $\mu(\theta_\mu) = \theta_\mu$, $\sigma(\theta_\sigma) = \exp(\theta_\sigma)$

   (b) $\mu(\theta_\mu) = \theta_\mu$, $\sigma(\theta_\sigma) = \theta_\sigma^2$

2. (2 pts.) Suppose our current parameters are $\mu = 0$ and $\sigma = 4$. We now observe an episode with $a = 3$ and reward $r = 3$. Perform a gradient update on θ_μ and θ_σ with learning rate $\alpha = 0.1$ using the policy gradient for both parametrizations. What are the new parameter values? What are the new policies $\mathcal{N}(\mu, \sigma)$?

3. (1 pts.) Use the results you obtained in the previous sub-question (updated policies) to explain a drawback of a simple policy gradient.

9.5 Homework: Coding Assignment - Policy Gradients

1. (1 pt.) We have spent a lot of time working on value based methods. We will now switch to policy based methods, i.e. learn a policy directly rather than learn a value function from which the policy follows. Mention two advantages of using a policy based method.

2. Download the notebook *RLLab5_PG.zip* from canvas assignments and follow the instructions.

Lecture 10: Advanced policy-search

10.1 Natural policy gradient

To explore the behavior of the gradients with different policy parametrizations we will use a stateless continuous bandit environment. In this environment, the agent performs a single action and receives a single reward before the episode is terminated. Furthermore, we assume that the reward function is known and the policy is represented by a normal distribution $\mathcal{N}(\mu(\theta_\mu), \sigma(\theta_\sigma))$.

It is common to parametrize standard deviation as an exponential of a parameter but in this exercise, we will consider a different parametrization (that serves better for illustration purposes) with hyperparameter k (we treat k as a design choice and thus its value cannot be changed during optimization). We will also ignore the fact that with this parametrization, the standard deviation could technically become negative:

$$r = a - a^2 \tag{2}$$

$$\pi(a|\theta) = \frac{1}{\sigma(\theta_\sigma)\sqrt{2\pi}} \exp\left(- \frac{(a - \mu(\theta_\mu))^2}{2\sigma(\theta_\sigma)^2} \right) \tag{3}$$

$$\mu(\theta_\mu) = \theta_\mu \tag{4}$$

$$\sigma(\theta_\sigma) = k\theta_\sigma \tag{5}$$

1. For this problem we can calculate policy gradient analytically. Calculate the gradient of expected reward $\mathbb{E}_a[r]$ with respect to parameters θ_σ and θ_μ

2. In natural policy gradients, we take the update direction as $u = F^{-1}\nabla J(\theta)$. The fisher information matrix F and is then given by:

   $$F_\theta = \mathbb{E}_a\Big[\nabla_\theta \log \pi(a|\theta) \nabla_\theta \log \pi(a|\theta)^T\Big] \tag{6}$$

   The derivatives of the log probability wrt. parameters are:

   $$\nabla_{\theta_\mu} \log \pi(a|\theta) = \frac{(a - \theta_\mu)}{(k\theta_\sigma)^2} \tag{7}$$

   $$\nabla_{\theta_\sigma} \log \pi(a|\theta) = \frac{(a - \theta_\mu)^2}{k^2{\theta_\sigma}^3} - \frac{1}{\theta_\sigma} \tag{8}$$

   Calculate the Fisher information matrix F_θ for our Gaussian policy.

   Hint: You can use results for central moments of a normal distribution:

   $$\mathbb{E}_a\Big[\big(a - \theta_\mu\big)\Big] = 0, \quad \mathbb{E}_a\Big[\big(a - \theta_\mu\big)^2\Big] = \sigma(\theta_\sigma)^2$$
   $$\mathbb{E}_a\Big[\big(a - \theta_\mu\big)^3\Big] = 0, \quad \mathbb{E}_a\Big[\big(a - \theta_\mu\big)^4\Big] = 3\sigma(\theta_\sigma)^4$$

3. Consider two different parameterizations that represent the same policy $\mathcal{N}(0, 0.1)$:

   (a) $\theta_\mu = 0$, $\theta_\sigma = 1$, $k = 0.1$

   (b) $\theta_\mu = 0$, $\theta_\sigma = 0.01$, $k = 10$

   Perform a single gradient update with step size $\alpha = 1$ for both of them using natural policy gradient. Compare both policies. How is the result that you obtained related to the constraint of the natural policy gradient update?

4. Plot the gradient directions with different $k = \{0.1, 1, 10\}$ for both policy gradient and natural policy gradient using the notebook we provided. Would you expect both algorithms to work well if we use same step size α to update both σ and μ (for all k)? Would using separate step sizes α_μ and α_σ improve the performance of both algorithms?

10.2 Trust Region Policy Optimization

Policy gradient methods that use SGD to optimize a policy π_θ with parameters θ, depend implicitly on a linear approximation of the expected return $J(\theta)$. If we make gradient steps that are too large, this approximation fails, and we might not be able to find good policies.

Trust Region Policy Optimization (TRPO) is an algorithm that tries to make sure that we stay in a safe region around the current parameters, such that the approximation holds. In practice, it looks a lot like the natural gradient method. In this exercise, we assume a two-armed bandit setting with a Bernoulli policy, directly parameterised by θ:

$$\begin{aligned} \pi_\theta(a) &= \theta^a (1-\theta)^{1-a}, \ a \in 0, 1 \\ r(a) &= a \end{aligned}$$

1. Let's start with the Natural Policy Gradient (NPG), which tries to compute a good update direction for the parameters. Compute this direction, given by $u = F^{-1}\nabla J(\theta)$, where F is the Fischer information matrix (here it's a 1D matrix), given by $F = -\mathbb{E}_a\left[\nabla^2_\theta \log \pi_\theta\right]$, and $J(\theta)$ is the expected return, given by $\mathbb{E}_a\left[r(a)\right]$.

2. Evaluate $u = F^{-1}\nabla J(\theta)$ for two settings of $\theta \in \{0.1, 0.5\}$. Then compute the KL divergence between the policy before and after the update suggested by the NPG (with a learning rate of 1). The KL divergence is given by $D_{KL}(\pi_{\theta_0} || \pi_\theta) = \mathbb{E}_{a \sim \pi_{\theta_0}}\left[\log \frac{\pi_{\theta_0}(a)}{\pi_\theta(a)}\right]$, where θ_0 are the old parameter values, and θ the updated values.

3. The primary practical difference between TRPO and NPG, is that TRPO sets the maximum allowed step size of an update in addition to the direction (which is the same as the NPG direction). It does this by requiring that the norm of the NPG update is equal to some hyperparameter value, which itself may be interpreted as a desired D_{KL} between the policy before and after the parameter update. The TRPO update is $\theta = \theta_0 + \beta u$, where $\beta = \sqrt{2D_{KL}\,(u^T F u)^{-1}}$. For the initial value $\theta = 0.5$, what should the value of β be, if we want the update for $\theta = 0.5$ to have (approximately) the same D_{KL} as the update for $\theta = 0.1$ in step 2. of this exercise?

4. Compute D_{KL} for this update and compare to the update you found for $\theta = 0.1$ in question 2 of this exercise.

10.3 Homework: Update Directions

Consider a game of rock, paper, scissors. The policy is parametrized by a Categorical distribution with parameters $\boldsymbol{\theta} = (\theta_{\text{rock}}, \theta_{\text{paper}}, \theta_{\text{scissors}})$ where each parameter corresponds to probability of selecting the corresponding action:

$$p(a = \text{rock}|\boldsymbol{\theta}) = \theta_{\text{rock}} \tag{9}$$

$$p(a = \text{paper}|\boldsymbol{\theta}) = \theta_{\text{paper}} \tag{10}$$

$$p(a = \text{scissors}|\boldsymbol{\theta}) = \theta_{\text{scissors}}. \tag{11}$$

Furthermore, the fisher information matrix for the categorical distribution is given as:

$$\boldsymbol{F} = \begin{bmatrix} \frac{1}{\theta_{\text{rock}}} & 0 & 0 \\ 0 & \frac{1}{\theta_{\text{paper}}} & 0 \\ 0 & 0 & \frac{1}{\theta_{\text{scissors}}} \end{bmatrix} \tag{12}$$

Alice starts with a *uniform policy.* She samples several games and performs an update using Natural Policy Gradient (NPG) after which she notices that $\theta_{\text{scissors}} > \theta_{\text{rock}} > \theta_{\text{paper}}$.

You can assume that learning rates and target KL are > 0. As part of your answer, give the update equation for each learning algorithm and use them to reason about your answer.

1. (1.5 pts.) Alice claims that she would always obtain the same ordering ($\theta_{\text{scissors}} > \theta_{\text{rock}} > \theta_{\text{paper}}$) if she had used TRPO for the update instead (using the same data). Bob claims that the ordering could be different depending on the sampled data (different sample but used for both NPG and TRPO), learning rate of NPG and target KL used for TRPO. Who is right? Explain your answer.

2. (1.5 pts.) Alice also claims that she would always obtain the same ordering ($\theta_{\text{scissors}} > \theta_{\text{rock}} > \theta_{\text{paper}}$) if she had used Policy Gradient (PG) for the update instead (using the same data). Bob claims that the ordering could be different depending on the sampled data (different sample but used for both NPG and PG) and learning rates used for PG and NPG. Who is right? Explain your answer.

3. (1 pt.) In general, which updates (from PG, NPG, TRPO) update parameters in the same direction in the parameter space?