CMPUT 655
Assignment 8

October 20, 2024

Dikshant
1877098

---

1. **Policy Gradient**

   (a) What are the main differences between value-based methods and policy gradient (PG) methods?
   - *Value-based methods learn ... while PG methods learn ...*
   - *Value-based methods loss function is ... while PG methods optimize ...*

   (b) Why is the PG theorem so important?
   - *When we write the derivate of the policy performance, what distribution is there? What happens to this distribution at the end of the PG theorem?*

   (c) REINFORCE is a Monte Carlo (MC) method, i.e., it updates the policy based on MC estimates of the expected return.
   - *What are the pros/cons of being MC?*
   - *REINFORCE has high ... but low ...*

   (d) Actor-critic methods combine PG and value-based methods. How?
   - *They learn both a ... (critic) and a ... (actor). The role of the critic is to ... while the actor ...*
   - *Compared to REINFORCE, they have lower variance but higher bias. Why? Hint: think about how the critic is trained.*

Ans:

(a) Value-based methods learn value functions, while PG methods learn a parameterized policy directly.
Value-based methods loss function is based on minimizing value estimation errors, while PG methods optimize the expected return

(b) The Policy Gradient (PG) theorem is crucial because it provides a way to calculate the gradient of the expected return with respect to policy parameters without needing to differentiate through the state distribution.
- The policy gradient theorem introduces the state visitation distribution $d^{\pi(s)}$, which is the discounted probability of visiting each state under the current policy. It gets eliminated in the final expression.

(c) REINFORCE has high variance but low bias

(d) Actor-critic methods combine PG and value-based methods by learning both a value function (critic) and a policy (actor)
The critic estimates the value function, while the actor updates the policy based on the critic's feedback
Compared to REINFORCE, actor-critic methods have lower variance because the critic provides a more stable estimate of the value, leading to more consistent updates. However, they have higher bias because the critic's value function estimation can introduce approximation errors.

2. **Min-Variance Constant Baseline**

Derive the baseline minimizing the variance for REINFORCE (often called "optimal baseline"). First, write the gradient estimate of REINFORCE. Then, write its variance. Finally, compute its derivative to find the baseline

Ans:

Gradient Estimate of REINFORCE with baseline:

$$\nabla J(\theta) = \mathbb{E}_\pi \left[\nabla \log \pi_\theta(a|s) \, (G_t - b)\right]$$

Variance of the Gradient Estimate:

$$\text{Var}\left(\nabla J(\theta)\right) = \mathbb{E}_\pi \left[\left(\nabla \log \pi_\theta(a|s) \, (G_t - b)\right)^2\right] - \left(\mathbb{E}_\pi \left[\nabla \log \pi_\theta(a|s) \, (G_t - b)\right]\right)^2$$

Focusing on second term:

$$\mathbb{E}_\pi \left[\nabla \log \pi_\theta(a|s) \, (G_t - b)\right] = \mathbb{E}_\pi \left[\nabla \log \pi_\theta(a|s) \, G_t\right] - b \, \mathbb{E}_\pi \left[\nabla \log \pi_\theta(a|s)\right]$$

$$= \mathbb{E}_\pi \left[\nabla \log \pi_\theta(a|s) \, G_t\right]$$

Since b is a constant, we want to minimize the variance of the term inside the expectation:

$$\text{Var} = \mathbb{E}_\pi \left[\left(\nabla \log \pi_\theta(a|s)\right)^2 \, (G_t - b)^2\right]$$

For optimal baseline, we take derivative of Var w.r.t. b

$$\frac{\partial}{\partial b}\text{Var} = \frac{\partial}{\partial b}\mathbb{E}_\pi \left[\left(\nabla \log \pi_\theta(a|s)\right)^2 \, (G_t - b)^2\right] = 0$$

Computing the derivative gives:

$$\mathbb{E}_\pi \left[(\nabla \log \pi_\theta(a|s))^2 \; 2(G_t - b) \right] = 0$$

$$\mathbb{E}_\pi \left[(\nabla \log \pi_\theta(a|s))^2 \; (G_t - b) \right] = 0$$

$$b = \frac{\mathbb{E}_\pi \left[(\nabla \log \pi_\theta(a|s))^2 \; G_t \right]}{\mathbb{E}_\pi \left[(\nabla \log \pi_\theta(a|s))^2 \right]}$$

3. **Exercise 13.2**: *Generalize the box on page 199, the policy gradient theorem (13.5), the proof of the policy gradient theorem (page 325), and the steps leading to the REINFORCE update equation (13.8), so that (13.8) ends up with a factor of γ^t and thus aligns with the general algorithm given in the pseudocode.*

Ans: From page 199,

$$\eta_\gamma(s) = h(s) + \gamma \sum_{\bar{s}} \eta_\gamma(\bar{s}) \sum_a \pi(a|\bar{s}) p(s|\bar{s}, a)$$

$$\mu_\gamma(s) = \frac{\eta_\gamma(s)}{\sum_{s'} \eta_\gamma(s')}$$

we are assuming a single starting state s_0, so we can also write $\eta_\gamma(s)$ as:

$$\eta_\gamma(s) = \sum_{k=0}^{\infty} \gamma^k \Pr(s_0 \to s, k, \pi)$$

Policy gradient algorithm for episodic case:

$$
\begin{aligned}
\nabla v_\pi(s) &= \nabla\Big[\sum_a \pi(a|s) q_\pi(s,a)\Big], \quad \text{for all } s \in \mathcal{S} \\
&= \sum_a \Big[\nabla\pi(a|s) q_\pi(s,a) + \pi(a|s)\nabla q_\pi(s,a)\Big] \\
&= \sum_a \Big[\nabla\pi(a|s) q_\pi(s,a) + \pi(a|s)\nabla \sum_{s',r} p(s',r|s,a)(r + \gamma v_\pi(s'))\Big] \\
&= \sum_a \Big[\nabla\pi(a|s) q_\pi(s,a) + \gamma\pi(a|s) \sum_{s'} p(s'|s,a)\nabla v_\pi(s')\Big] \\
&= \sum_a \Big[\nabla\pi(a|s) q_\pi(s,a) + \gamma\pi(a|s) \sum_{s'} p(s'|s,a) \\
&\sum_{a'} \big[\nabla\pi(a'|s') q_\pi(s',a') + \gamma\pi(a'|s') \sum_{s''} p(s''|s',a')\nabla v_\pi(s'')\big]\Big] \\
&= \sum_{x\in\mathcal{S}} \sum_{k=0}^{\infty} \gamma^k Pr(s \to x, k, \pi) \sum_a \nabla\pi(a|x) q_\pi(x,a)
\end{aligned}
$$

Now, we have

$$
\begin{aligned}
\nabla_\theta J(\theta) &= \nabla_\theta v_\pi(s_0) \\
&= \sum_s \left(\sum_{k=0}^{\infty} \gamma^k \Pr(s_0 \to s, k, \pi)\right) \sum_a \nabla_\theta\pi(a|s) q_\pi(s,a) \\
&= \sum_s \eta_\gamma(s) \sum_a \nabla_\theta\pi(a|s) q_\pi(s,a) \\
&= \sum_{s'} \sum_s \frac{\eta_\gamma(s')}{\sum_{s'} \eta_\gamma(s')} \sum_a \nabla_\theta\pi(a|s) q_\pi(s,a) \\
&= \sum_{s'} \eta_\gamma(s') \sum_s \mu_\gamma(s) \sum_a \nabla_\theta\pi(a|s) q_\pi(s,a) \\
&\propto \sum_s \mu_\gamma(s) \sum_a \nabla_\theta\pi(a|s) q_\pi(s,a)
\end{aligned}
$$

Steps leading to the REINFORCE update equation (13.8)

$$
\begin{aligned}
\nabla_\theta J(\theta) &\propto \sum_s \mu_\gamma(s) \sum_a q_\pi(s,a,\theta)\nabla_\theta\pi(a|s) \\
&= \mathbb{E}_\pi\left[\gamma^t \sum_a \pi(a|s_t,\theta) q_\pi(s_t,a) \frac{\nabla_\theta\pi(a|s_t,\theta)}{\pi(a|s_t,\theta)}\right]
\end{aligned}
$$

$$= \mathbb{E}_\pi \left[\gamma^t q_\pi(s_t, a_t) \frac{\nabla_\theta \pi(a_t|s_t, \theta)}{\pi(a_t|s_t, \theta)} \right]$$
$$= \mathbb{E}_\pi \left[\gamma^t G_t \frac{\nabla_\theta \pi(a_t|s_t, \theta)}{\pi(a_t|s_t, \theta)} \right]$$

Discounted REINFORCE update

$$\theta_{t+1} \doteq \theta_t + \alpha \gamma^t G_t \frac{\nabla_\theta \pi(a_t|s_t, \theta)}{\pi(a_t|s_t, \theta)}$$
$$= \theta_t + \alpha \gamma^t G_t \nabla_\theta \ln \pi(a_t|s_t, \theta)$$

4. **REINFORCE**

(a) The algorithm implemented following the instructions does not explicitly learn how to control exploration. Why?
- *We do not learn ... in the softmax policy, and ... in the Gaussian policy.*
- *The softmax policy, however, implicitly learns exploration. Why?*
- *Write the softmax equation and denote with $h(s, a; w)$ the trained function.*
- *What happens to the exploration as we learn h? Does it become more/less greedy as the value of some state-action pairs grows?*

(b) The algorithm implemented is very simple. Name and and describe at least two techniques / methods to improve it. Write one sentence for each technique / method.
- *Actor-critic does not count.*
- *Hints: regularize gradients (how?), different optimizers (which ones?), different gradient estimators (which one did we talk about?)*

Ans:

(a)
- We do not learn the temperature parameter in the softmax policy, and the variance parameter in the Gaussian policy.
- It adjusts the relative probabilities of actions as it learns, indirectly controlling exploration.
- Softmax distribution:

$$\pi_w(a|s) = \frac{\exp(h(s, a; w))}{\sum_{a'} \exp(h(s, a'; w))}$$

- As some state-action pairs become more favorable, reduction in exploration, and the policy becomes more deterministic.

(b) 1. *Gradient Clipping*: Regularize gradients by clipping them to a maximum value, preventing instability in updates.
2. *Using Different Optimizers*: Replace standard gradient descent with adaptive optimizers like Adam for better convergence.
3. *Variance Reduction*: Use methods like GAE (Generalized Advantage Estimation) to reduce the variance of policy updates.

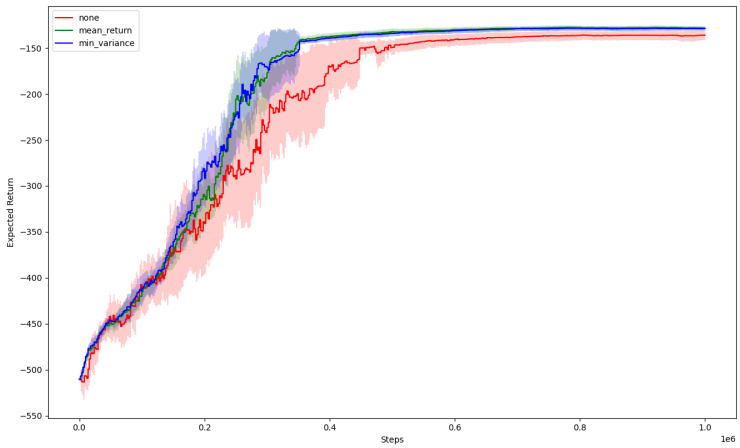

Figure 1: REINFORCE - Continuous actions

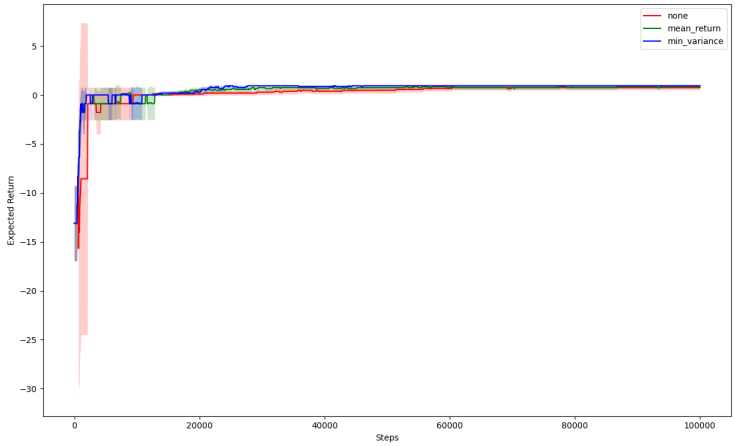

Figure 2: REINFORCE - Discrete actions

Code can be accessed on github from here: REINFORCE