CMPT 419/983: Theoretical Foundations of Reinforcement Learning
Assignment 4

Total marks: 250

Due: December 10, 11:59 pm

Submission Instructions

- Assignments typed in Latex (the Latex source is provided) are preferred, but can be handwritten.
- You can write the code in any language, but are not allowed to make use of automatic differentiation (using Numpy + Python is preferred). The code and plots is to be submitted as a separate zip file. All code files and plots should be stored in a directory named `a4` and then zip the directory for submission.
- Assignment (PDF + separate zip file for code and plots) is to be submitted online via Coursys.

(1) [100 marks] Proving the remaining results from class

- Prove that the Jacobian of $h : \mathbb{R}^A \to \mathbb{R}^A$ is given by:

$$H(\pi_\theta) \in \mathbb{R}^{A \times A} = \text{diag}(\pi_\theta) - \pi_\theta \, \pi_\theta^T ,$$

where $\text{diag}(\pi_\theta) \in \mathbb{R}^{A \times A}$ is a diagonal matrix s.t. for any s, $[\text{diag}(\pi_\theta)]_{a,a} = \pi_\theta(a|s)$ and $\pi_\theta(\cdot|s) \in \mathbb{R}^A$ s.t. $\pi_\theta(a|s) = \frac{\exp(\theta(s,a))}{\sum_{a'} \exp(\theta(s,a'))}$. Using the above calculation, show that for any s', a',

$$\frac{\partial \log(\pi_\theta(a'|s'))}{\partial \theta(s,a)} = \mathcal{I}\left\{s' = s\right\} \left[\mathcal{I}\left\{a' = a\right\} - \pi_\theta(a|s)\right] \text{ [20 marks]}$$

- For arbitrary policies $\pi, \tilde{\pi}$: prove that

$$v^\pi(\rho) - v^{\tilde{\pi}}(\rho) = \frac{1}{1-\gamma} \, \mathbb{E}_{s \sim d^\pi} \mathbb{E}_{a \sim \pi(\cdot|s)}[\mathfrak{a}^{\tilde{\pi}}(s,a)] \text{ [25 marks]}$$

- Consider the modified TRPO update: for $\theta \in \mathbb{R}^d$,

$$\theta_{t+1} = \arg\max_\theta \left[v^{\pi_t}(\rho) + \underbrace{\frac{\mathbb{E}_{s \sim d^{\pi_t}} \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \mathfrak{a}^{\pi_t}(s,a)}{1-\gamma}}_{\text{Term (i)}} - \beta \underbrace{\sum_s d_t^\pi(s) \, \text{KL}(\pi_t(\cdot|s) || \pi_\theta(\cdot|s))}_{\text{Term (ii)}} \right]$$

Prove that using a linear approximation of Term (i) and a quadratic approximation of Term (ii) results in the following update:

$$\theta_{t+1} = \arg\max_{\theta} \left[v^{\pi_t}(\rho) + \langle \nabla_\theta \text{Term (i)}|_{\theta=\theta_t}, \theta \rangle - \frac{\beta}{2} (\theta - \theta_t)^T \left[\nabla^2_\theta \text{Term (ii)}|_{\theta=\theta_t}\right] (\theta - \theta_t) \right]$$

and recovers the update for natural gradient descent, i.e. $\theta_{t+1} = \theta_t + \frac{1}{\beta} F^{\dagger}_{\theta_t} \nabla J(\theta_t)$. [30 marks]

- For the finite-horizon problem in Lecture 12, prove the following result that bounds the difference in the performance of the same deterministic policy π on two different MDPs: $M = (\mathcal{S}, \mathcal{A}, \{\mathcal{P}_h\}_{h=0}^{H-1}, \{r_h\}_{h=0}^{H-1})$ and $\tilde{M} = (\mathcal{S}, \mathcal{A}, \{\tilde{\mathcal{P}}_h\}_{h=0}^{H-1}, \{\tilde{r}_h\}_{h=0}^{H-1})$. If $v_0^{\pi,M}$ is the value function of policy π on MDP M from $h = 0$, assuming $v_H^{\pi,M} = v_H^{\pi,\tilde{M}} = 0$, prove that for a starting state $s_0 \in \mathcal{S}$,

$$\begin{aligned} &v_0^{\pi,M}(s_0) - v_0^{\pi,\tilde{M}}(s_0) \\ &= \mathbb{E}_{\substack{s_{h+1} \sim \tilde{\mathcal{P}}_h(\cdot|s_h a_h), \\ a_h = \pi_h(s_h)}} \sum_{h=0}^{H-1} \left[[r_h(s_h, a_h) - \tilde{r}_h(s_h, a_h)] + \langle \mathcal{P}_h(\cdot|s_h, a_h) - \tilde{\mathcal{P}}_h(\cdot|s_h, a_h), v_{h+1}^{\pi,M} \rangle \right] \end{aligned}$$ [25 marks]

(2) [70 marks] A new PG method Consider the tabular softmax PG update, i.e. for $\theta \in \mathbb{R}^{SA}$, $\pi_\theta(a|s) = \frac{\exp(\theta(s,a))}{\sum_{a'} \exp(\theta(s,a'))}$. The update $\theta_{t+1} = \theta_t + \eta \nabla_\theta J(\theta_t)$ can be alternatively written as:

$$\theta_{t+1} = \arg\max_{\theta} \left[\langle \theta - \theta_t, \nabla_\theta J(\theta_t) \rangle - \frac{1}{\eta} \|\theta - \theta_t\|_2^2 \right]$$

As in Lecture 8, the update can be generalized to measure the distance between θ and θ_t using a Bregman divergence, resulting in the following update:

$$\theta_{t+1} = \arg\max_{\theta} \left[\langle \theta - \theta_t, \nabla_\theta J(\theta_t) \rangle - \frac{1}{\eta} D_\Psi(\theta, \theta_t) \right], \tag{1}$$

where ψ is the mirror map.

- For $\theta(s, \cdot) \in \mathbb{R}^A$, prove that choosing $\psi(\theta(s, \cdot)) = \log\left(\sum_a \exp(\theta(s,a))\right)$ results in $D_\psi(\theta(s,\cdot), \theta'(s,\cdot)) = \text{KL}(\pi'(\cdot|s)||\pi(\cdot|s))$, where $\pi_\theta(a|s) = \frac{\exp(\theta(s,a))}{\sum_{a'} \exp(\theta(s,a'))}$ [25 marks]

- For the update in Eq. (1), we will use the following Bregman divergence: $D_\Psi(\theta, \theta') = \sum_s d^{\pi'_\theta}(s)\, D_\psi(\theta(s,\cdot), \theta'(s,\cdot))$. If $\pi_t := \pi_{\theta_t}$, prove that this results in the following update rule:

$$\theta_{t+1} = \arg\max_{\theta} \left[\mathbb{E}_{s \sim d^{\pi_t}} \mathbb{E}_{a \sim \pi_t(\cdot|s)} \left[\left(\frac{\mathfrak{a}^{\pi_t}(s,a)}{1-\gamma} + \frac{1}{\eta} \right) \log\left(\frac{\pi_\theta(a|s)}{\pi_t(a|s)} \right) \right] \right]$$ [30 marks]

- For the tabular policy parameterization, prove that the above problem is equivalent to the following problem:

$$\pi_{t+1}(\cdot|s) = \arg\max_{\pi(\cdot|s) \in \Delta_A} \left[\mathbb{E}_{s \sim d^{\pi_t}} \left[\mathbb{E}_{a \sim \pi_t(\cdot|s)} \left(\frac{\mathfrak{a}^{\pi_t}(s,a)}{1-\gamma} \log\left(\frac{\pi(a|s)}{\pi_t(a|s)} \right) \right) - \frac{1}{\eta} \text{KL}\left(\pi_t(\cdot|s)||\pi(\cdot|s)\right) \right] \right]$$ [10 marks]

- Compare the above expression to the regularized TRPO update (**v1**) for the tabular parameterization where Π_θ consists of distributions $\pi(\cdot|s) \in \mathbb{R}^A$ for each state s. What are the advantages/disadvantages of using one over the other? [5 marks]

(3) [50 marks] PG with Log-linear policies When $\mathcal{S}$ and $\mathcal{A}$ is large, function approximation may be needed to reduce the dimension. For simplicity, let us consider log-linear policy in the bandit setting (with deterministic rewards). A log-linear policy is such that for all $a \in [K] := \{1, 2, \ldots, K\}$,

log-linear policy 中 \theta 的维度是d=2

$$\pi_\theta(a) = \frac{\exp([\Phi\theta](a))}{\sum_{a' \in [K]} \exp([\Phi\theta](a'))}$$

where $\Phi \in \mathbb{R}^{K \times d}$ is the feature matrix with full column rank $d \leq K$. When $d = K$ and the features are one-hot vectors, log-linear policies reduce to the tabular setting. The objective is:

$$J(\theta) := \mathbb{E}_{a \sim \pi_\theta}[r(a)] = \langle \pi_\theta, r \rangle$$

- Derive the softmax policy gradient $\nabla_\theta J(\theta)$ for the above setting. [10 marks]

Consider the bandit setting with $K = 4$ with the following linear features and reward vector:

$$\Phi = \begin{bmatrix} 0 & 2 \\ 0.4 & 0 \\ -2 & 0 \\ 0 & -0.4 \end{bmatrix} \quad r = \begin{bmatrix} 1 \\ 0.9 \\ 0.8 \\ 0.7 \end{bmatrix}$$

bandit with softmax tabular setting: L9

Compare log-linear polices with the above linear features to tabular features in both the deterministic and stochastic settings. For the stochastic setting, use the importance-weighted reward estimate to construct the gradient. In each setting initialize $\theta_0(a) = 0$ for all $a \in [K]$, grid-search over the range of $\eta \in \{10^{-3}, 10^{-2}, 10^{-1}, 1\}$ to select the step-size. L11

- In the deterministic setting, with $T = 1000$, plot the suboptimality gap: $\langle \pi^*, r \rangle - \langle \pi_{\theta_t}, r \rangle$ for both linear and tabular features (with the corresponding best step-size) on the same plot. Use a log-scale (if required) to better visualize the results. Save the final plot as `pg.png`. [20 marks]
- In the stochastic setting, with $T = 10^4$, run the experiment for both linear and tabular features (and different step-sizes) 20 times, and plot the average suboptimality gap and standard deviation on the same plot. Use a log-scale (if required) to better visualize the results. Save the final plot as `spg.png`. [20 marks]

(4) [30 marks] Empirical Evaluation In `mdp.py`, the following functions will generate an instance of an MDP and the corresponding v^*.

- `generate_cliffworld()` will return the following:
  - $\mathcal{S} := \{0, 1, \ldots, 20\}$
  - $\mathcal{A} := \{0, 1, \ldots, 3\}$
  - $r \in \mathbb{R}^{21 \times 4}$
  - $\mathcal{P} \in \mathbb{R}^{21 \times 4 \times 21}$ where `P[s][a][next_s]` $= \mathcal{P}[s'|s, a]$
  - $\rho \in \mathbb{R}^{21}$

tabular 的theta: tabular softmax PG 和 NPG
1. 更新theta得到新的pi；
2. 通过pi计算v^{\pi_theta}；
迭代

For the given MDP, let $\gamma := 0.9$ and implement PG and NPG. For both PG and NPG, use $\eta = \frac{1}{L} = \frac{(1-\gamma)^3}{8}$ and let $\theta_0(\cdot|s) = 0$ for all $s \in \mathcal{S}$. Run each algorithm for $T = 10^4$ iterations and plot the log suboptimality gap: $\log\left(v^{\pi^*}(\rho) - v^{\pi_{\theta_t}}(\rho)\right)$. Save the final plot as `pg_vs_npg.png`.