CMPSCI 687 Course Project

Due December 10, 2019

Due Date: This course project is due on the last day of class, December 10. There will be no penalty for submitting late as long as you submit by midnight at the end of December 13.

Overview: This course project simulates the application of RL to a real problem, like a medical application or business application, where deploying a policy worse than the current one would be dangerous or costly. To simulate this, **1)** we will provide you with data, but not the underlying MDP that created the data and **2)** deploying a policy that is worse than the policy we used to generate the data will be costly for your grade—your grade will be the percent of policies that you provide us that are better than the policy we used to generate the data.

Tools and Collaboration: You can use *any* language and code libraries you find online. You can discuss the project with other students, including the approaches you plan to take. **However, you must write your code on your own.** You will also be submitting your code with compilation instructions, though your code will not influence your grade other than to confirm that it produces the policies you submit.

Details: We will provide you with one file, data.csv, and you will provide us with 100 policies (1.csv, 2.csv, etc.) and the code you used to generate these policies. Your policies must be different from the policy we used to generate the data (the details of this policy are included in data.csv), though they do not need to be different from each other—you could submit the same policy for each of the 100 policies you submit, though then your grade will be a zero if it is worse (but 100 if it is better). You can use *any* approach you want to compute the policies that you submit. We will use $\gamma = 1$, and you may assume that $R_t \in [-10, 10]$ always.

The file data.csv will contain the following values, where each item below indicates a row in the CSV file.

1. An integer m indicating the number of state features.
2. An integer $|\mathcal{A}|$ indicating the number of discrete actions. That is $\mathcal{A} = \{0, 1, 2, \dots, |\mathcal{A}| - 1\}$.
3. An integer k, indicating the Fourier basis order used by π_b, the policy that generated the data we are providing.
4. θ_b, the parameters of the policy π_b. The following describes precisely how θ_b parameterizes π_b: in short, it is a softmax policy using the Fourier basis of order k, and assuming that states are already normalized when they are provided (they will be). There will be $|\mathcal{A}|(k+1)^m$ real numbers on this row. Let θ_b^i denote the i^{th} block of $(k+1)^m$ numbers on this row, as a

column. Then, $\theta_b = [(\theta_b^1)^\intercal, (\theta_b^2)^\intercal, \dots, (\theta_b^{|\mathcal{A}|})^\intercal]^\intercal$. The policy π_b is defined as:

$$\pi_b(s,a) = \frac{e^{\phi(s)^\intercal \theta_b^a}}{\sum_{a' \in \mathcal{A}} e^{\phi(s)^\intercal \theta_b^{a'}}}, \tag{518}$$

where $\phi(s) \in \mathbb{R}^{(k+1)^m}$, the i^{th} element of which is: $\phi_i(s) = \cos(\pi c_i^\intercal s)$, where $c_i \in \mathbb{R}^{|\mathcal{S}|}$ is the number i written in base $k+1$ using little-endian (100 is one rather than 001), where each digit corresponds to one element of the vector c_i.

5. An integer, n, that indicates the number of episodes of data that will follow.

6. The next n rows each correspond to one episode of data. The row will contain $S_0, A_0, R_0, S_1, A_1, R_1, \dots, S_{T-1}, A_{T-1}, R_{T-1}$, where T is the length of the episode (which may vary across episodes). Here the states S_t will each be sequences of m real numbers, A_t will be a single integer, and R_t will be a real number. The total number of entries on this row is therefore $T(m+2)$. The states will really be observations—measurements about the world that the agent is interacting with—and so they may not be Markovian. Assume that $S_T = s_\infty$. The provided states will be normalized so that all values are in the range $[0,1]$.

7. Let T_1 be the length of the first episode of data above. This line contains T_1 real numbers, that correspond to $\pi_b(S_t, A_t)$ for $t \in \{0, 1, 2, \dots, T-1\}$. This row is provided to give you a way to test whether your policy representation matches ours.

What to submit: You will submit a .zip folder that contains files 1.csv, 2.csv, ..., 100.csv. If you want to submit a single policy, you still need to have all of these files, though they might be identical. These files each contain one row corresponding to new policy parameters to use in place of θ_b using the same policy parameterization described for π_b. There should also be a directory called "source" that contains source code for producing these policies. Inside of the "source" directory there should be a file called "readme.txt" that includes instructions for running your source code.

Hint: You probably do not want to write code, run it on our data first, and submit the policies it produces. If you do this, how confident are you that your code is correct and that the method you implemented will succeed? We recommend that you create your own MDPs, generate data, and see if your method succeeds on those tests. Running your code on our provided data should just be the final step right before submitting. We will ensure that particularly advanced methods that we did not cover in class are not necessary (e.g., using the high-confidence policy improvement algorithm from class with t-test and per-decision importance sampling can be enough to obtain a 90%).

Data availability: We will provide a link to the data here on December 3. To help you create your example MDPs (though we suggest starting tabular and building up!), the MDP that we use will likely have a very small value of m (likely 1), k will be small, and episode lengths will be short (likely < 10).