Angela Sy
AA 228 / CS 238: Decision Making Under Uncertainty
Project 2: Reinforcement Learning

For this project, we were given 3 data sets representing sample data from different Markov Decision Processes. Based on the sample states, actions, next states, and rewards, we were tasked to find the policy action for each state that maximizes the total expected reward.

I used two methods to learn the expected utility of each state – Value Iteration and Q-learning. I chose to use Value Iteration for the small and medium data sets because we had a known model for the MDP and there was a relatively high ratio of data points to existing states. On the other hand, Q-Learning was better for the large data set as the model was unknown and we had to perform some exploration. Knowing the expected utility of actions for each state, I was able to extract the policy for each MDP.

Model-Based Reinforcement Learning: Value Iteration

For the first two data sets, I used Maximum Likelihood Estimation to learn the Transition and Reward Functions, and then implemented Value Iteration to find the expected value of each state. From this, I extracted the best policy by taking the action that corresponded to the highest expected utility for each state.

Table 1: Results for Value Iteration on Small and Medium Datasets

	# States	# Actions	Score	Time (in seconds)	% of states with optimal policy found
Small	100	5	52971.6335	6.389	100
Medium	251,001	2	6339.6766	59.069	50.16

1. Small Dataset – Grid World

Value Iteration worked well for the small data set as we had a comprehensive data set where all states had been visited in the sample data. Hence, the Transition and Reward functions were quite accurate and allowed us to come up with relatively accurate values for the different states.

2. Medium Dataset – Inverted Cart Pendulum

The second data set had many more states than the first. As can be seen in Table 1, only about half of all the data sets were present in the sample. Hence, our Transition and Reward functions were less accurate and could not predict the exact utility of each state. States where no best action was found based on the calculated utilities were given a random action.

Overall, this method was effective for finding the policy for states where we had almost complete information about the rewards in each state, but lacking when we had sparse data and no known model of the world.

Model-Free Reinforcement Learning: Q-Learning

For the large data set, Q-learning seemed to be the most appropriate method since the model of the MDP was unknown and many of the states were not present in the sample data. Q-learning is a learning algorithm where we search through the state space and apply incremental updates to the corresponding q-value (or expected utility) of each state as we observe rewards in every state. After searching through the space for a certain period of time, we use the approximated q-values of each state to determine the best policy for each state.

Table 2: Results for Q-Learning on Large Dataset

	# States	# Actions	Score	Time (in seconds)	% of states with optimal policy found
Large	531,441	157	-366.0502	122.089	0.0129

3. Large Dataset – Unknown Model

As can be seen in the table, the majority of states were not given an optimal action policy. This can partially be attributed to constraints in the amount of data provided. The exploration that was implemented in the algorithm was limited to the (state, action, successor state) groups that had already been observed in the sample data and so did not represent true exploration of the state space. To improve this algorithm further, I would allow exploration of the states by choosing a random start state and then proceeding through successor states, instead of following the data points given in the sample data. Another option is to use eligibility traces to allow states with an observed reward to propagate these utilities back to neighbor states, so that rewards of states observed later will be reflected in states already visited.