Problem Set #1: Markov Decision Processes, Value Iteration, Linear Programming

---

Deliverable: pdf write-up by Wednesday September 9th, 23:59pm. Your solution should be a pdf consisting of 3 pages, one page dedicated to each question (in order). Your PDF is to be submitted into www.gradescope.com (if you are registered or on the waitlist, you should have an account with your email address on file with the Registrar's Office; if not, email the instructors to get yourself added). Note that some of the starter code will generate plots. Don't include all of them, include whatever is sufficient to show that you correctly solved the problem. Whenever including a figure, make sure it is accompanied by a discussion of that figure.

Refer to the class webpage for the homework policy.

Various starter files are provided on the course website.

You are free to use any linear programming solver of your choice for this assignment. If you don't have in-depth experience with a particular solver already, I recommend you try out CVX. CVX can solve a broad class of convex optimization problems, including linear programs, but also quadratic programs (which you will need for future assignments!). The file cvx_intro.m provides download/installation instructions, as well as a few examples of problems being solved with CVX.

1. Shortest Path in a Graph as an MDP

Consider the problem of finding the shortest path in a graph (V, E) (with V the set of vertices, and E the set of directed edges), and where v_G is the destination vertex. Every edge in the graph takes equally long to traverse. Describe how this problem can be encoded into a Markov decision process, (S, A, T, γ, R), such that the optimal solution in the Markov decision process is the shortest path in the graph. Your MDP is required to have rewards that are positive (zero included) for all transitions.

2. Value Iteration

(a) **Implement value iteration into value_iteration.m.** Run value iteration for the provided gridworld and report the values obtained for each state. (You can check your results against the ones provided in the starter-code to ensure correctness!) Look at main.m to get started.

(b) **Queuing.** Consider a server that has to serve three queues denoted by a, b, c. Each queue can be of length zero through five. At the beginning of each time-step, the server gets to choose between each of the three queues. If the server stays with its current queue and the current queue is non-empty, then it clears a request from that queue. If the server moves to another queue, this takes up an entire time-step and no request is cleared in that time-step. Clearing a request results in a reward of $+1$, moving between queues results in a reward of 0. In each time-step, after the server has chosen its action and this action has been executed, a request is added to each queue with size strictly smaller than five with probability p_a, p_b, p_c respectively. This happens independently for each of the queues. The discount factor $\gamma = 0.99$. (i) Implement the queue MDP, (S, A, T, γ, R), that models this setting. (ii) For $p_a = 0.1, p_b = 0.2, p_c = 0.5, \gamma = 0.9$, using value iteration (which you already implemented for part (a)) compute the values and optimal actions for all states. Report the values and the optimal actions for the following states: $(0, 0, 0, a), (2, 0, 4, b), (5, 2, 1, c)$.

3. Solving MDPs with Linear Programming

(a) **Implementation.** Implement the (primal) linear programming based solution to MDPs into linear_programming.m. Solve again the two MDPs you solved in the previous question, and verify that you obtain the same results. Report that you indeed get the same results, or, if not, report the results you obtained with LP formulations the same way you did for value iteration.

(b) **Theory.** In class we covered the linear programming formulation (both primal and dual) that solves infinite horizon MDPs with discounting, which you have used for the previous questions. Write down both the primal and the dual LP formulations for the finite horizon setting, with horizon H.