Markov Decision Processes

Instructions

Answer all the mandatory questions and **choose one** of the proposed track. You can work in individually or in teams of 2. If you use Colab, export your final notebook and submit the .ipynb in your Github repo. Include a link to the live Colab corresponding to your snapshot.

Deadline : Monday February 5th.

Assignment Submission : On Github Classroom https://classroom.github.com/g/LogpXqte

Question/Comments: Just comment on this document.

TAs: Matthew Smith <matthew.smith5@mail.mcgill.ca>, Maziar Gomrokchi <maziar.gomrokchi@mail.mcgill.ca>

Mandatory Questions

Bellman Optimality Equations

Using a contraction argument, show that there exists a solution to the Bellman **optimality** equations. That is : show that the Bellman **optimality** operator is a contraction mapping. (Doina covered the linear case in class ; here you need to go through the same steps but in the nonlinear case).

Policy Iteration

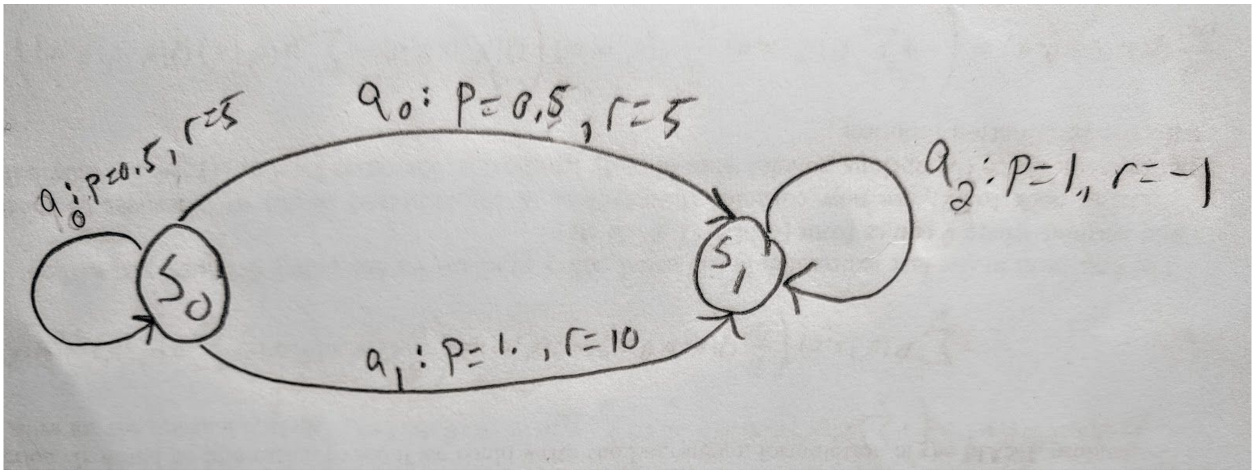
Show that the values of two successive policies generated by policy iteration are nondecreasing. Assume a finite MDP and conclude (explain why) that policy iteration must terminate under a finite number of steps. Finally, show that upon termination, policy iteration must have found an optimal policy (ie. one which satisfies the optimality equations).

Track 1

Implement and compare empirically the performance of value iteration, policy iteration and modified policy iteration. Modified policy iteration is a simple variant of policy iteration in which the evaluation step is only partial. You can consult the Puterman (1994) textbook for more information. You should first implement the algorithms in a 2-states MDP specified as follows:

Transition Probabilities: P(s_0 | s_0, a_0) = 0.5, P(s_1 | s_0, a_0) = 0.5, P(s_0 | s_0, a_1) = 0, P(s_1 | s_0, a_1) = 1, P(s_1 | s_0, a_2) = 0, P(s_1 | s_1, a_2) = 1

(State s_0 has access to actions a_0 and a_1, while in state s_1 the agent can only choose a_2. As Preeti Vyas commented, the probabilities are simply zero for the actions that cannot be taken.)



Rewards: r(s_0, a_0) = 5, r(s_0, a_1) = 10, r(s_1, a_2) = -1
Discount factor : 0.95

Advice: Implement these algorithms in matrix form. Your code will be much more succinct.

Also run your experiments in a **second MDP of your choice** (chain MDP, grid world, etc.). Explain and explore how the convergence rate of these algorithms is affected by the discount factor.

Track 2

Equivalence of Policy Iteration with Newton’s Method

In “On the Convergence of Policy Iteration in Stationary Dynamic Programming” by Puterman and Brumelle (1979), the authors show that policy iteration can be seen as a Newton-Kantorovich iteration approach. Re-derive the main result and explain the geometric intuition behind this result.

Track 3

Matrix Splittings

Both value iteration and policy iteration can be accelerated through *matrix splitting* methods. Standard acceleration methods such as Gauss-Seidel or Jacobi iterations can in fact be shown to correspond to certain matrix splittings. Explain what a matrix splitting is and how it can be combined with value iteration and policy iteration. Show the convergence of value iteration with matrix splitting methods. You can consult Puterman (1994) section 6.3 for more information.