Assignment5 – Markov Decision Procedures

Given: May 30 Due: June 3

Problem 5.1 (HMMs in Python)

Implement filtering, *prediction* and *smoothing* for *HMMs* in Python by completing the *implementation* of `hmm.py` at `https://kwarc.info/teaching/AI/resources/AI2/hmm/`.

Hint: This problem uses `numpy`, which is a Python *library* for working with arrays/matrices. If you have never worked with `numpy` before, you can find many high-quality introductions online. Due to its popularity and frequent use for machine learning etc., it is definitely worth getting to know `numpy`. That being said, you only need very few and basic numpy functions for this assignment, which you should be able to find without problems (e.g. searching for *numpy matrix multiplication*).

Problem 5.2 (Markov Decision Processes)

1. Give an optimal policy π^* for the following MDP:
   - set of states: $S = \{0, 1, 2, 3, 4, 5\}$ with initial state 0
   - set of actions for $s \in S$: $A(s) = \{-1, 1\}$
   - transition model for $s, s' \in S$ and $a \in A(s)$: $P(s' \mid s, a)$ is such that
     - $s' = (s + a) \mod 6$ with probability $2/3$,
     - $s' = (s + 3) \mod 6$ with probability $1/3$.
   - reward function: $R(5) = 1$ and $R(s) = -0.1$ for $s \in S \setminus \{5\}$
2. State the Bellman equation.
3. Complete the following high-level description of the value iteration algorithm:
   - The algorithm keeps a table $U(s)$ for $s \in S$, that is initialized with
   - In each iteration, it uses the

     in order to
   - $U(s)$ will converge to the

Problem 5.3 (Bellman Equation)

State the *Bellman equation* and explain every symbol in the equation and what the equation is used for and how.

Problem 5.4 (MDP Example)

Consider the following world:

+50	-1	-1	-1	...	-1	-1	-1	-1
Start				...				
-50	+1	+1	+1	...	+1	+1	+1	+1

The world is 101 fields wide (i.e., 203 fields in total). In the $Start$ state an agent has two possible actions, Up and $Down$. It cannot return to $Start$ though and the cannot pass gray fields, so after the first move the only possible action is $Right$.

1. Model this world as a Markov Decision Process, i.e., give the components S, s_0, A, P, and R.
2. For what discount factors γ should the agent choose Up and for which $Down$? Compute the utility of each action (i.e., the utility of the successor state) as a function of γ.
3. What is the optimal policy if the upper path is better?