Reinforcement Learning (E1 277)

- Assignment 04 -

1. Construct a global Lyapunov function for the limiting ODE of the TD(0) algorithm with linear function approximation. You must explicitly verify all the conditions in the definition of a global Lyapunov function.

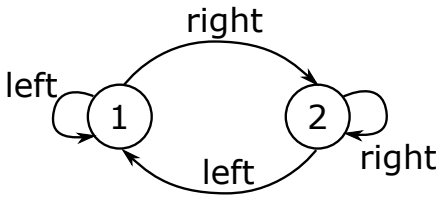

Figure 1: Markov Decision Process

2. Consider the two-state Markov decision process given in Fig. 1. Let $\phi(1) = 1$ and $\phi(2) = 2$ be the feature vectors associated with the two states. Suppose the behavior policy η satisfies $\eta(\text{right}|1) = \eta(\text{right}|2) = 0.5$, while the target policy μ satisfies $\mu(\text{right}|1) = 0.9$ and $\mu(\text{right}|2) = 0.9$. Let the discount factor $\gamma = 0.9$. Finally, let A, A', and A'' denote the limiting matrices corresponding to the on-policy TD(0), the naive off-policy TD(0), and the emphatic TD(0) algorithms, respectively; see (1), (8) and (11) in [Sutton et al 2016] (uploaded in files section).

   (a) Find the matrix P^μ and the stationary distribution d^μ associated with the Markov chain induced by the policy μ. Also, find A and determine if the limiting ODE associated with the on-policy TD(0) has a globally asymptotically stable equilibrium or not. (2)

   (b) Find the stationary distribution d^η and then compute A'. Again, determine if the limiting ODE associated with the naive off-policy algorithm has a globally asymptotically stable equilibrium or not. (2)

   (c) Compute the row vector f, whose s-th coordinate is given by

$$f(s) = d_\eta(s) \lim_{t\to\infty} \mathbb{E}_\eta[F_t|s_t = s], \tag{1}$$

   where $\{F_t\}$ is the follow on trace sequence defined in (10) in [Sutton et al, 2016]. Also, find A'' and verify if the limiting ODE associated with emphatic TD(0) has a globally asymptotically stable equilibrium or not. (2)

3. For a general emphatic TD(0) algorthm, let $f(s)$ be as defined in (1). Do you think it is possible to identify what $\sum_s f(s)$ should be? Please provide valid explanations.

4. Derive an update rule for Q-learning algorithm with linear function approximation. You should start from a suitably chosen optimization problem and then justify each and every step until you get an implementable algorithm. Please note that you don't have to discuss convergence of this algorithm, instead your justification for the update rule should come from the objective function you choose.