Reinforcement Learning – exercise 07

Hung Ngo & Vien Ngo
Machine Learning & Robotics lab, University of Stuttgart
Universittsstrae 38, 70569 Stuttgart, Germany

June 6, 2016

1 Tabular Dyna Mazes

Reproduce results in examples 9.1, 9.2, and 9.3 of Sutton & Barto's book. Use the same parameter setting: initial action values are *one* (terminal states must have *zero* values for all actions!), zero reward everywhere except reward of 1 at goal state, planning steps $N = \{0, 5, 50\}$, step-size parameter $\alpha = 0.1$, discount factor $\gamma = 0.95$, and exploration parameter $\epsilon = 0.1$. For Dyna-Q+, choose several *reasonable* values of κ, e.g., $\kappa \in \{0.0002, \ldots, 0.002\}$, for preliminary evaluations, then select the best one for comparison with other learning methods.

2 Linear Dyna for Mountain Car

Linear Dyna is described in Sutton et al.'s UAI'08 paper. A linear model is used to predict next state feature vector $\phi' = \phi(s') = F_a\phi$ and next reward $r = b_a^\top \phi$ given the current state feature vector $\phi(s)$ and action a. A linear regressor is used to approximate the *state* value function $V(\phi) \approx \theta^\top \phi$, and the greedy action is chosen using one-step look-ahead $a \leftarrow \arg\max_{a'} \left[b_{a'}^\top \phi + \gamma \theta^\top F_{a'} \phi\right]$. A linear Dyna algorithm with TD(0) update & prioritized sweeping for control is presented in the lecture slides. The imaginary experience is generated using the feature vectors ϕ_{e_j} of the centroids j: unit basis vector $\phi_{e_j} = (0, \ldots, 0, 1, 0, \ldots, 0)$ with all zero components except the jth, which is 1.

Implement this algorithm to solve the mountain car problem in HW6, with exactly the same parameter setting. Furthermore: initialize $F_a = \beta\mathbb{I}$ with $\mathbb{I}$ the identity matrix, $\beta = 0.1$ for all actions a; set planning parameter p to some values e.g. $\{10, 50, 100\}$ for preliminary evaluations, then select the best one for comparison with other learning methods.