Reinforcement learning tutorial

Wouter Caarls

Koroibot Summer School, September 25th, 2014

1 Introduction

In this tutorial you will get some hands-on experience with reinforcement learning for a simple dynamical system. Although the learning algorithm is provided, you can play with all its parameters to see the effects.

Before you get started, get the Matlab toolbox from `http://wouter.caarls.org/files/kss2014_tutorial.tgz`, and start `pendgui` from the toolbox directory.

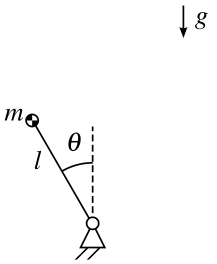
The system is the pendulum from Figure 1a, and the goal is to swing it up from the stable to the unstable equilibrium. The GUI will show you the learning curve (average return vs episode number) as well as the value function (expected return over the state space), see Figure 1b.

The parameters are:

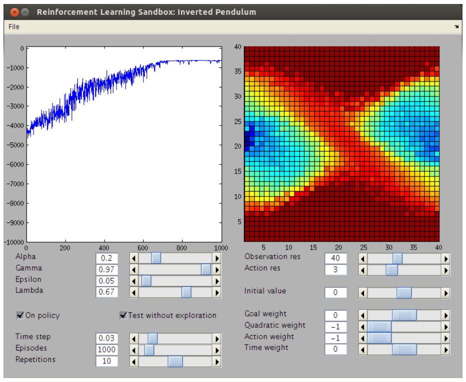
α Learning rate

γ Discount rate

ε Exploration rate



(a) Inverted pendulum system



Alpha 0.2
Gamma 0.97
Epsilon 0.05
Lambda 0.67

☑ On policy ☑ Test without exploration

Time step 0.03
Episodes 1000
Repetitions 10

Observation res 40
Action res 3

Initial value 0

Goal weight 0
Quadratic weight −1
Action weight −1
Time weight 0

(b) GUI

λ Trace decay rate

On-policy Selects between off-policy (Q-learning) and on-policy (SARSA)

Test without exploration Draw learning curve from special test runs in which exploration is turned off

Time step Time between control steps

Episodes Number of episodes to learn for

Repetitions Number of runs to average to produce a nice learning curve

Observation res Number of discrete states per dimension

Action res Number of discrete actions that can be chosen by the control policy

Initial value Initial value of the value function

Goal weight Reward for reaching the goal state

Quadratic weight Weight factor for the quadratic position and velocity errors

Action weight Weight factor for the square of the applied torque

Time weight Reward per time step

The reward function is:

$$\begin{aligned} r = & w_{\text{goal}} && \text{goal\_reached}(s_{\text{pos}}, s_{\text{vel}}) + \\ & w_{\text{quadratic}} && \left(5s_{\text{pos}}^2 + 0.1s_{\text{vel}}^2\right) + \\ & w_{\text{action}} && a^2 + \\ & w_{\text{time}} && \end{aligned}$$

2 Assignments

Make sure to reset all settings in between exercises! (File→Reset)

2.1 Learning rate α

Compare the learning curves at $\alpha = 0.2$ and $\alpha = 0.7$. What happens? Why do you think that is?
Hint: α acts as a noise filter, but the system is deterministic. What could cause the noise that must be filtered?

2.2 On-policy vs off-policy learning

Set α to 0.7.
Compare on-policy and off-policy learning. Are the results what you would expect? Why? Hint: also compare on-policy and off-policy learning with ε set to 0.01.

2.3 State space resolution

Set α to 0.7 and the observation resolution to 20.
Compare off-policy and on-policy learning. Can you explain the result?
Hint: both work much better at $\alpha = 0.1$, so there must be another source of noise that affects both methods. What is it?

2.4 Discount rate γ

Compare learning with $\gamma = 0.97$ and $\gamma = 0.87$. What happens to the value function? Why?
Hint: look at the way it behaves on the path towards the goal state.

2.5 Discount rate γ (2)

Now compare to $\gamma = 0.57$. Can you explain the effect?
Hint: how does the reward function of the swing-up motion change over the course of the optimal path?

2.6 Reward function

Set γ to 0.57
Compare between "path" rewards (weights set to <0, -1, -1, 0>) and "goal" rewards (weights set to <1, 0, 0, 0>). Why does this reward function cause such a different behavior?
Note: by increasing and decreasing gamma you can now more clearly see the effect of question 4.

2.7 On-policy vs off-policy learning (2)

Set γ to 0.57, use "goal" reward weights <1, 0, 0, 0>, and set ε to 0.45.
Compare on-policy and off-policy learning. What do you see in the value function? Explain.
Hint: What does such a high exploration rate do to the probability of reaching the goal?

2.8 Initial value

Set the number of episodes to 2000.
Compare the learning curves at initial value -500 to initial value 0. What happens?
Hint: Which state will be selected the next time a state is reached once it has been updated with a lower than expected reward (initial value 0) versus a higher than expected reward (initial initial value -500)?