Tópicos especiais em Inteligência Computacional II:
Aprendizado por Reforço
Exercício 3: LSPI

Wouter Caarls
wouter@ele.puc-rio.br

April 25, 2019

Least Squares Policy Iteration (LSPI)[1] is a batch reinforcement learning method for linear approximators that uses a policy iteration scheme where the policy evaluation part is solved using linear least squares. In this exercise you will write an implementation of it and see how it performs on the pendulum swing-up problem.

1 Initialization

In order to run LSPI, you will need a batch of experience and a way of calculating the feature vector for arbitrary state-action pairs. For this exercise, the experience will be generated randomly across the state space, and the features will be radial basis functions.

Exercise 1.1 Choose feature centers uniformly distributed over the state-action space with 11 centers over the position dimension $\in [-\pi, \pi]$, 11 over the velocity dimension $\in [-12\pi, 12\pi]$, and 3 over the voltage dimension $\in [-3, 3]$ (use `linspace`). Use `meshgrid` to generate the cartesian product of the positions, velocities and voltages.

Exercise 1.2 Choose a batch size $N = 10000$ and generate N transitions starting from uniformly distributed random states s and taking random actions a. Use `pendulum` to calculate the next state. Record the starting state, action, and next state s' for each transition.

Exercise 1.3 Calculate the reward for each transition using the following reward function

$$r = -5{s'}^2_{\text{pos}} - 0.1{s'}^2_{\text{vel}} - a^2$$

[1] Lagoudakis and Parr, 2003. `http://www.jmlr.org/papers/volume4/lagoudakis03a/lagoudakis03a.pdf`

Such a smooth reward function will be easier to approximate using few parameters than a hard 'box'.

Exercise 1.4 Initialize a random policy with actions a': the actions to take in the next state s'. This is the policy that will be improved during policy iteration.

2 Features

We need a function to calculate the feature values of a particular state-action combination (s, a). Since these will be radial basis function features, they depend only on the distance between (s, a) and the particular center.

Exercise 2.1 First calculate the distance of (s, a) to all centers. In order to have comparable distances in each dimension, divide the velocity component by 12.

Exercise 2.2 Then determine the feature activation using the distance, for example using the `normpdf` function with $\sigma = 1$. Make sure this function is vectorized, otherwise it will run very slowly!

3 Policy iteration

Now we can finally program the policy iteration loop that will do the actual optimization.

Exercise 3.1 Create a loop for policy evaluation and policy improvement to occur. Use a fixed number $n = 10$ iterations.

Exercise 3.2 At the beginning of every iteration, do policy evaluation by calculating $\phi(s, a)$ and $\phi(s', a')$ for the current policy, creating the matrix

$$A_{i*} = \phi(s_i, a_i) - \gamma\phi(s'_i, a'_i)$$

and solving $A\theta = r$. Choose $\gamma = 0.9$

Exercise 3.3 After policy evaluation, improve the current policy by calculating the value $\phi(s', *)^\top\theta$ for all actions $a' \in [-3, 0, 3]$ and choosing the action with the highest value. You should now have a working algorithm.

4 Questions

With all the coding done we can start analyzing the performance of the algorithm. These questions will need to be answered in your report.

Exercise 4.1 To see the algorithm working, scatter-plot the states using the actions of the current policy as colors at the start of each iteration. Copy the final plot to your report and compare the result to the various examples of linear SARSA given in the slides. How does it look?

Exercise 4.2 Reduce the batch size to $N = 1000$ to speed up the algorithm, and increase the number of iterations to $n = 50$. What is the convergence behavior?

Exercise 4.3 Compare the convergence speed for $\gamma = 0.9$ and $\gamma = 0.99$ for the original settings and explain the result.

Exercise 4.4 Reduce the number of basis functions per position/velocity dimension to 5, and also try increasing them to 21. What are the results in terms of computation time, convergence rate and quality of the solution? Explain.

Exercise 4.5 Try increasing and decreasing the support of the basis functions σ. What happens? Explain.

Exercise 4.6 Why does LSPI not require a learning rate α?