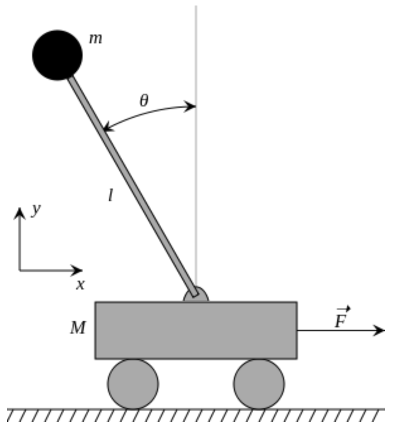
SMU tutorial 5

Petr Ryšavý

Monday 25th March, 2024

1 Problem 1 - regression models

Reinforcement learning is highly effective when used for standard control tasks. Consider the inverted pendulum problem as below.



https://commons.wikimedia.org/wiki/File:Cart-pendulum.svg#/media/File:Cart-pendulum.svg

Let the state space be represented by a tuple $s = (\theta, v)$, where $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ represents angle of the pendulum and $v \in \mathbb{R}$ is velocity of the cart. Let action be the force $F \in \{-2, -1, 0, 1, 2\}$ applied to the cart over a unit of time.

1. Explain why table-based Q-learning or SARSA is not directly applicable to this problem.

2. One of the ways to deal with this situation is to use a *regression model* where $\hat{U}(\mathbf{w}, s) = \mathbf{w}^T s$. Our goal is then to minimize the squared error

$$E_j(\mathbf{w}, s) = \frac{1}{2}\left(\hat{U}(\mathbf{w}, s) - u_j(s)\right)^2 \tag{1}$$

   using the following update rule

$$w_i \leftarrow w_i - \alpha \frac{\partial E_j(\mathbf{w}, s)}{\partial w_i}. \tag{2}$$

   Value $u_j(s)$ represents the sampled utility.

   Where does the update rule (2) come from? What is its meaning? (Providing an established name for this update rule is a sufficient answer.)

3. Plug the formula of the error (1) into the formula of the update (2) and arrange the result so that the derivative is applied only on the $\hat{U}$ function. Keep this formula for the problem in 5. Then propagate the derivative further and explain, how would you use this update rule.

4. Propose a different way to represent the state space so that table-based Q-learning or SARSA would be applicable.

5. In a more general case, we might have a set of n relevant features $\phi_i : S \to \mathbb{R}$. The estimated utility function is then $\hat{U}(\mathbf{w}, s) = \sum_{i=1}^{n} w_i \phi_i(s)$.

   For now, assume that the state space is the set $\{1, 2, 3, \ldots, |S|\}$. Suppose that we have $|S|$ features, such that $\phi_i(s) = 1$ iff $s = i$, i.e., each feature is nonzero for exactly one state. Suppose that the sampled state value is $u_j(s) = r(s) + \gamma \cdot \hat{U}(s')$. Use this model in the formula you derived in item 3 above. What is the name of the algorithm you obtained? Hint: think about what is the meaning of w_i and how it relates to $\hat{U}(s)$.

2 Problem 2 - Multiple Armed Bandids

Credit for this problem goes to Richard Sutton, University of Alberta.
http://www.incompleteideas.net/book/the-book.html, Exercise 2.2.

Consider a multi-arm bandit problem with $k = 5$ actions, denoted 1, 2, 3, 4, and 5. Consider applying to this problem a bandit algorithm using ε-greedy action selection, sample-average action-value estimates, and initial estimates of $Q_1(a) = 0$ for all a. Suppose the initial sequence of actions and rewards is

$$\begin{aligned} A_1 &= 1, & R_1 &= -2, \\ A_2 &= 2, & R_2 &= 3, \\ A_3 &= 3, & R_3 &= -1, \\ A_4 &= 2, & R_4 &= 2, \\ A_5 &= 3, & R_5 &= 0, \\ A_6 &= 4, & R_6 &= 5. \end{aligned}$$

On some of these time steps, the ε case may have occurred, causing an action to be selected at random. At which time steps did this definitely occur? On which time steps could this possibly have occurred?

3 Problem 3 - Adversarial Mutliarmed Bandit

Credit for this problem goes to Richard Sutton, University of Alberta.
http://www.incompleteideas.net/book/the-book.html, Exercise 2.10.

Suppose you face a 2-armed bandit task whose true action values change randomly from time step to time step. Specifically, suppose that, for any time step, the true values of actions 1 and 2 are respectively 10 and 20 with probability 0.5 (case A), and 90 and 80 with probability 0.5 (case B).

- If you are not able to tell which case you face at any step, what is the best expected reward you can achieve, and how should you behave to achieve it?
- Now suppose that on each step, you are told whether you are facing case A or case B (although you still don't know the true action values). This is an associative search task. What is the best expected reward you can achieve in this task, and how should you behave to achieve it?

If you are not able to tell which case you face at any step, what is the best expected reward you can achieve and how should you behave to achieve it? Now suppose that on each step you are told whether you are facing case A or case B (although you still don't know the true action values). This is an associative search task. What is the best expected reward you can achieve in this task, and how should you behave to achieve it?