Problem-set 3 — solutions

What is reinforcement learning

Problem 1. *Which of the following are suitable for reinforcement learning, and which are more suitable to other methods? If it is suitable for RL, suggest a state space, action space, and rewards.*

1. *A six-legged robot. The controller can produce the actions: lift leg, move leg forward, move leg backward, lower leg, independently for each leg. The robot contains an accurate position sensor, which gives x, y and height of the center of the body. Goal is to get it to walk forward.*
2. *Fingerprint reader on a phone. It needs to recognize fingerprints of the user.*
3. *Self-driving car controller. The controller should autonomously drive the car to a given location, without harm to living things or physical objects.*
4. *Portfolio of investments. There are seven possible investments. You will build a portfolio of three, so as to get the maximal return. You divide the money you invest evenly between the three investments.*
5. *Get a wheeled robot to navigate to my office in the Kilburn building from anywhere in the Kilburn or IT buildings. The robot can detect where it is from the numbers on doors, but has neither a map nor a planner.*

Answer:

- RL could be used. State could be the position of the center of the robot and the position of the legs. Perhaps you also need to know the position of the legs in previous time steps. Actions are movements of the legs. Reward could be the distance travelled forward.
- This is a classic supervised learning problem. Not an RL problem. Train it on known fingerprints.
- Some aspect may employ RL. For example, steering control to stay in the middle of the lane. Some aspects would require a human teacher.
- This could be an RL problem. The actions are which of $\binom{7}{3}$ to use. There might be no states, or the states might be economic properties, bull or bear, volatility, etc. Reward is immediate return or cumulative return.

- An RL problem. State space is current location, actions are drive straight, turn left or right (by some angle), reward is 1 when my office is reached and zero otherwise. Alternatively, reward could be 1 when my office is reached and a small negative amount otherwise, to reward shorter over longer paths. Although discounting does that as well.

Note: I might ask a question like this on an exam. Hopefully, with better examples. If so, I would not necessarily expect you to give the same answers as mine. I would be looking an awareness of what could be an RL problem, and the fact that you need a state space, and action space, and a reward structure.

End of answer!

Problem 2. *One way to train an agent to play a game, is to get a champion player to produce a database of board positions with the best move from them. Use that to train a machine learning model to take the board position as input and produce the best move as output, trained with a standard supervised learning (learning from labelled examples) approach. Briefly describe what you think are the advantages and disadvantages to this approach relative to a reinforcement learning approach.*

Answer:

Advantages: If it could learn the data, it would perform correctly on board positions it has seen, whereas an RL learner might never encounter board positions found in human games. If it could generalize to board positions it has not been trained on, and to a very wide range of board positions, it could be a very good player. Although there may be an astronomical number of board positions, it might be the case that the number of board positions which actually occur in an intelligently played game is vastly smaller, which would make generalization more plausible.

Disadvangages: Due to the number of nodes in a game tree of an interesting game, it might not be possible to humans to label enough board positions to achieve good results. This approach is not able to produce a system which achieves better performance than humans.

End of answer!

Learning equations

Problem 3. *I have claimed in the lectures that the following update equation*

$$V(a) \leftarrow V(a) + \frac{1}{t_a}\left[r_t - V(a)\right]$$

results in mean rewards when action a is taken. Here t_a is the number of times that action a was taken, and r_t is the reward at a time when action a is taken. I am now going to get you to show that this is true for a single sequence.

Let there be a sequence of real numbers $x_1, \ldots, x_n, \ldots$. Let V_n obey,

$$V_0 = 0; \tag{1}$$

$$V_n = V_{n-1} + \frac{1}{n}(x_n - V_{n-1}). \tag{2}$$

Prove that $V(n) = \frac{1}{n}\sum_{i=1}^{n} x_i$ i.e. the mean of xs up to the current time.

Hint 1: *it might be useful to write equation (2) as*

$$V_n = V_{n-1}\frac{n-1}{n} + \frac{x_n}{n}.$$

Hint2: *Use induction*

Answer:

Base case: $V(1) = x_1$ which is the mean of one data point.

True for $n-1$ implies true for n: Assume that it is true for $V(n-1)$, so that

$$V_{n-1} = \frac{1}{n-1}\sum_{i=1}^{n-1} x_i.$$

It follows that

$$V_n = \frac{n-1}{n}V_{n-1} + \frac{x_n}{n} \tag{3}$$

$$= \frac{n-1}{n}\frac{1}{n-1}\sum_{i=1}^{n-1} x_i + \frac{x_n}{n} \tag{4}$$

$$= \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right). \tag{5}$$

QED

Note: I would not a question like this on a 2 hour exam. I might on an open-book, week long exam, like we had last semester.

End of answer!

Problem 4. *Now we study the learning equation with a constant learning rate α. The learning equation is*

$$V(a) \leftarrow V(a) + \alpha\left[r_t - V(a)\right].$$

Assume that the xs are all a constant value T. In other words, $x_i = T$ for all $i \in [1, \infty)$.

1. *If the learning equations from Problem 3, equations (1) and (2) are used, show that $V_n = T$ for all $n > 0$. (This should be pretty easy.)*

2. *Show that if the learning equation with a constant learning rate is used, equation (4, then V_n decays exponentially in n to T. (This might be pretty hard, depending on how much you have learned about solving recursion relations.)*

   To do this, write it as a recursion relation,

$$V_n = V_{n-1}(1-\alpha) + \alpha x_n; \tag{6}$$

$$V_n = V_{n-1}(1-\alpha) + \alpha T, \text{ since } x_n = T \text{ for all } n. \tag{7}$$

   If you know how to solve a recursion relation like equation (7), then solve it. If you don't know how to solve a recursion relation like this, see the appendix, where I walk you through it.

Answer:

Following the steps in the appendix, and assuming $V_n = 0$, we find,

$n = 1$**:** $V_1 = \alpha T$;

$n = 2$**:** $V_2 = \alpha T(1-\alpha) + \alpha T$

$n - 3$**:** $V_3 = \alpha T(1-\alpha)^2 + \alpha T(1-\alpha) + \alpha T$

General pattern: Now I am ready to guess the general pattern, which is

$$V_n = \alpha T \sum_{i=0}^{n-1} (1-\alpha)^i.$$

Using Lemma 1 with $\lambda = 1 - \alpha$ yields,

$$V_n = \alpha T \frac{1-(1-\alpha)^n}{1-(1-\alpha)}; \tag{8}$$

$$= T - T(1-\alpha)^n. \tag{9}$$

Since $1 - \alpha$ is less than 1, this is converging exponentially to the correct value T, from below. If α is small, it will be a slow exponent.

Note: I would never, never, ¡insert lots of nevers¿ ask this on an exam of any kind. Too hard, I think. I am just trying to give you a feel for why these learning equations make sense.

End of answer!

Q-Learning

Problem 5. *Figure 1 shows the game tree for NIM-N with $N = 4$. This is the game that starts with N matches, each player in turn can take $1, 2$ or 3 matches, and whoever takes the last match loses.*

Suppose, both players use tabular Q-learning to learn in self play which runs long enough for the Q-values to be very close to the converged values. The state would the number of matches remaining, and the actions are the number of matches taken.

Write out the Q-values for s=4 and a=1,2 and 3. Note: only count the decision nodes of the given player. Don't count the nodes of the opponent.

Answer:

$Q(4,1) = -\gamma$. $Q(4,2) = -\gamma$. $Q(4,3) = 1$.

Note: I can imagine asking a question like this on an exam.

End of answer!

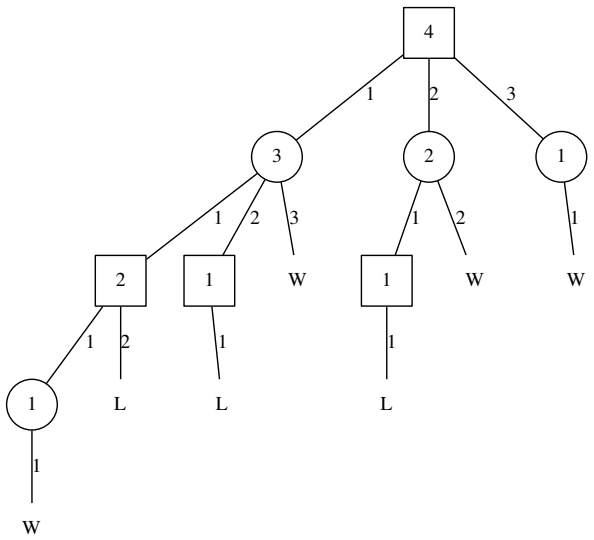

Figure 1: The game tree for NimN with N=4. Squares are Player 1; circles are Player 2, numbers in the shapes are the number of matches remaining, and numbers on the links are the numbers of matches taken in move. This is related to Problem

Problem 6. *Figure 2 shows a chain of nodes. Each node is a state, and from each node there are two actions: move left (L) and move right (R). The picture shows the Q-table values after Q-learning has trained for a very long time (forever). For example, it shows that* $Q(c,R)$ *is* $\gamma^2 10$ *and* $Q(c,L)$ *is* $\gamma^4 10$.

Show that if the Q-tables have these values, the Q-table values will not change under the learning dynamics. I.e. these values are fixed points for the learning dynamics. Do this for node c and node e.

The learning dynamics uses the following update rule,

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha\left(r_t + \gamma \max_{a'} Q(s_{t+1},a') - Q(s_t,a)\right).$$

What I am asking you to show is when you update, for example $Q(c,L)$*, its value is exactly the same after updating as it is before updating. Don't do this for every node, but just nodes c and e.*

Note: *The fact that these values are fixed points does not show that the dynamics converges to these values, only that if it ever reaches them, they will not*

change. To show convergence, we would need to show that this is an attracting *fixed point. This is harder, and I have no intention in asking you to do it.*

Answer:

Consider first $Q(c, R)$. Then $s_t = c, s_{t+1} = d$ and $r_t = 0$. $Q(c, R) = 10\gamma^2$ by assumption. Also, since by assumption $Q(d, R) = 10\gamma$ and $Q(d, L) = 10\gamma^3$, it follows that the best action from node d is right, and $\max_{a'} Q(d, a') = 10\gamma$. And the reward is zero. So the update equation is

$$Q(c, R) \leftarrow Q(c, R) + \alpha(r + \gamma \max_{a'} Q(d, a') - Q(c, R), \tag{10}$$

$$10\gamma^2 \leftarrow 10\gamma^2 + \alpha(0 + \gamma 10\gamma - 10\gamma^2, \tag{11}$$

$$10\gamma^2 \leftarrow 10\gamma^2. \tag{12}$$

So, no change. Likewise, with $Q(c, L) = 10\gamma^4$, and $\max_{a'} Q(b, a') = 10\gamma^3$, so $(\gamma^2 \max_{a'} Q(b, a') - Q(c, L))$ is $(\gamma^4 - \gamma^4)$ which is 0 so again no change during the update. For node e, $Q(e, R)$ is an immediate reward update, so $(r - Q(e, R) = (10 - 10) = 0$. Finally, for $Q(e, L)$ again we have $(\gamma \max_{a'} Q(d, a') - Q(e, L)) = 10\gamma^2 - 10\gamma^2 = 0$.

Note: I might ask on an exam to produce the final state of Q-learning on a graph. This question is a little too annoying to answer.

End of answer!

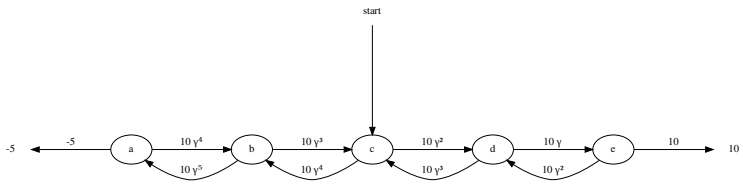

Figure 2: A chain of states illustrating a Q-table after learning using Q-learning.

Appendix: Solving a simple recursion relation

We want to solve the recursion relation,

$$V_n = V_{n-1}(1-\alpha) + \alpha T.$$

This recursion relation is very simple and can be solved. (Most cannot be solved.) There are formulaic steps to solve equations like this, but here are the steps I suggest you take. These often work.

Step 1: Working out the recursion, compute small values of n: $0, 1, 2$, etc. until you see the pattern. You may need to simplify the equations, or maybe go further, $4, 5$, until you see the pattern.

Step 2: Once you see the pattern, write down the general formula for V_n. It probably involves a sum of n terms.

Step 3: Prove the general formula is correct using induction. Or don't. I often don't, unless I am doing it for a research paper or money or something. At least check it for higher values of n than you computed in Step 1.

Step 4: Simplify the sum using Lemma 1. You will get a formula which exponentially decays to the constant value T.

Lemma 1. *Let S be a sum of exponents in a positive real number λ,*

$$S = 1 + \lambda + \lambda^2 + \ldots + \lambda^{n-1}; \tag{13}$$

$$S = \sum_{i=0}^{n-1} \lambda^i. \tag{14}$$

Then S is given by the following formula,

$$S = \frac{1-\lambda^n}{1-\lambda}.$$

The sum is exponentially decreasing if λ is less than 1, and exponentially increasing if λ is greater than 1.

Proof. This involves a standard trick, which is to turn it into what is called a telescoping sum. Multiply both sides of equation (13) by $1-\lambda$. This yields,

$$(1-\lambda)S = \left(1 + \lambda + \lambda^2 + \ldots + \lambda^{n-1}\right)(1-\lambda), \tag{15}$$

$$= 1 - \lambda + \lambda - \lambda^2 + \lambda^2 - \ldots - \lambda^{n-1} + \lambda^{n-1} - \lambda^n \tag{16}$$

$$= 1 - \lambda^n. \tag{17}$$

(The middle equation is the "telescoping sum".) This proves the formula. □