1. Suppose a Q-learning agent, with fixed *alpha*, and discount gamma, was in state 34 did action 7, received reward 3 and ended up in state 65. What value(s) get updated? Give an expression for the new value. (You need to be as specific as possible)

Solution

q[34,7] = q[34,7] + alpha (3 + gamma** max_a *q[65,a] - q[34,7])*

2. In temporal difference learning (e.q. Q-learning), to get the average of a sequence of k values, we let $alpha_k = 1/k$. Explain why it may be advantageous to keep $alpha_k$ fixed in the context of reinforcement learning.

Solution

The initial values are not as good estimates as newer values, and so you may not want to weight them as much. It is simpler to ignore the counts (and so keep *alpha* fixed). With a fixed *alpha* it is able to adjust when the environment changes.

3. Explain what happens in reinforcement learning if the agent always chooses the action that maximizes the Q-value. Suggest two ways that can force the agent to explore.

Solution

It gets stuck in non-optimal policies because it does not explore enough to find the best action from each state. To explore, it can pick random actions occasionally. You could also set the initial values high, so that unexplored regions look good.

4. In MDPs and reinforcement learning explain why we often use discounting of future rewards.

Solution

With no discounting the sum of the rewards is often infinite. Discounting means that more recent rewards are more valuable than rewards far in the future.

5. What is the main difference between asynchronous value iteration and standard value iteration? Why does asynchronous value iteration often work better than standard value iteration?

Solution

In standard value iteration all of the values are updated from the previous values in a sweep through the values. In asynchronous value iteration, the values are updated from the current value and can be done in any order (you don't need to sweep through all of the values). It often works better because the latest values are always used and it can concentrate on updating values where they make the most difference (as it doesn't need to sweep through all of the values each time).

6. In feature-based reinforcement learning, what new data point does the experience <*s,a,r,s',a'*> give for the linear regression? (I.e., what is the new data point for what value?)

Solution

It gives a new estimate of *r+gammaQ(s',a')* for the value of *Q(s,a)*.

7. In learning under uncertainty, when the the EM algorithm used? What is the E-step? What is the M-step?

Solution

EM algorithm is used for learning probabilities when the value for some variable is not observed in the data. (E.g., the class variable may not be observed). In the E-step, the data is filled in based on the probabilistic model (we get the expected number in the data). In the M-step the probabilities are updated based on the augmented data (we get the maximum likelihood probabilities).

8. Solution

There are 15 possible states that could be entered, depending on which direction the robot actually went (up, left or right) and whether the treasure arrived, and where it arrived. Those that have a non-zero immediate reward and/or a future value give:

```
  Q[s*,a2]
          =
              0.8 * 0.8 * ( 0 + 0.9 * 2)      -- up, no treasure
            + 0.8 * 0.2 * 0.25 * ( 0 + 0.9 * 7)      -- up, treasure at top right
            + 0.1 * 0.2 * 0.25 (10 + 0.9*0)      -- right, treasure appears there
```

every other value is 0. Note that *0.2*0.25* is the probability that a treasure appears at the top right state.