Notes

January 16, 2019

A collection of notes from the textbook, *Reinforcement Learning* by Richard Sutton and Andrew Barto. Available at `http://incompleteideas.net/book/the-book.html`

1 Action selection

Most RL methods require some form of policy or action-value based action selection alogrithm.

- **Greedy Selection**: Choosing the best action.

$$A = \text{argmax}_a Q(a)$$

- **ε-greedy Selection**: Simple exploration with ε-probability.

$$A \leftarrow \begin{cases} \text{argmax}_a Q(a) & \text{with probability 1 - } \varepsilon \text{ (breaking ties randomly)} \\ \text{a random action} & \text{with probability } \varepsilon \end{cases}$$

- **Upper Confidence Bound (UCB)**: Takes into account the proximity of the estimate to being maximal and the uncertanty in the estimates. Does not perform well on large state spaces.

$$A_t = \underset{a}{\text{argmax}} \left[Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}} \right]$$

  Where:

  - $c > 0$ is the degree of exploration
  - $N_t(a)$ is the number of times that action a has been selected prior to time t. If $N_t(a) = 0$, then a is considered to be a maximizing action.

2 Performance Measures

- **Optimal Action %**: Requires knowledge of the workings of the environment and whether the action was optimal. Plot % over steps.

- **Average reward**: Simply plot the average reward over steps. Good for comparing specific implementation of agent in specific implementation of environment.

- **Average reward w.r.t. parameter**: Plot the average reward over first n=1000 steps against input parameter(s) (ε, α, c, Q_0 etc) on a logarithmic scale. Good for comparing learning algorithms' general effectiveness and finding the best parameter value.

3 Algorithms

- **Dynamic Programming / Value Iteration**: Updating state values by sweeping through all states. Computationally expensive, especially on large state spaces.

```
Parameters:
  a small threshold $\theta$ > 0
  Initialize $V(s)$ $\forall s \in S^+$ arbitrarily, except that
  $V(terminal) = 0$

Loop:
  $\Delta \leftarrow 0$
  Loop for each $s \in S^+$:
    $v \leftarrow V(s)$
    $V(s) \leftarrow \texttt{max}_a \sum_{s',r} p(s',r|s,a)\,[r + \gamma V(s')]$
    $\Delta \leftarrow \texttt{max}(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Output a deterministic policy $\pi$, such that
  $\pi(s) = \texttt{argmax}_a \sum_{s',r} p(s',r|s,a)\,[r + \gamma V(s')]$
```