Q-Learning

- An agent tries an action at a particular state, and evaluates its consequences in terms of the immediate reward or penalty it receives and its estimate of the value of the state to which it is taken.
- The paper shows that Q-learning converges to the optimum action-values with probability 1 so long as all actions are repeatedly sampled in all states and the action-values are represented discretely.

π(x) is the next state
recommended by policy π

Prob that y= π(x)

The the value of state x is

$$V^{\pi}(x) \equiv \Re_x(\pi(x)) + \gamma \sum_{y} P_{xy}[\pi(x)]\, V^{\pi}(y)$$

Instant reward of
going to the state which
policy π recommends

Learning
rate

Value of new state y

The goal of Q-learning is to find π*, such that for any given state *x*, π* recommends the action *a* that will maximize the value of current state.

$$V^{\pi^*}(x) = \max_a \left\{ \mathcal{R}_x(a) + \gamma \sum_y P_{xy}[a] V^{\pi^*}(y) \right\}$$

To get this optimal policy π^*, we build the matrix Q in incremental way, like in Dynamic Programming:

$$Q^{\pi}(x, a) = \Re_x(a) + \gamma \sum_y P_{xy}[\pi(x)] V^{\pi}(y).$$

Now, since we want $\pi(x)$ to be optimal, it will recommend $\max(V^{\pi}(y))$ with probability 1. So, equation becomes:

$$Q(x, a) = R_x(a) + \gamma * \max_{a'}(Q(x', a'))$$

Data Structures

- Matrix "R" is the reward matrix. *R[x][a]* denotes instant reward of performing action *a* at state *x*. Only the actions leading to goal state have positive reward.
- Matrix "Q" is the brain matrix. It represents the memory of what our agent has learned through experience. *Q[x][a]* denotes learned reward of performing action *a* at state *x.* Q can be initially zero.
- However, size of these matrices depends on the size of action and state space, which could be exponential. So, we generally use look-up tables instead.

References

- [1992] "Q-Learning". Christopher Watkins, Peter Dayan. Nature Publishing Group.