Q Learning

Quality Function

* V(s, a) = Value of state/action pair
* Q(s, a) = Quality of state/action pair (not just the value)
* Q(s, a) = $\mathbb{E}(R(s', s, a) + \gamma V(s'))$ -> expected future value given my current state **s**, taking action **a** and ending up in state **s'** - the immediate reward **R** for taking **a** + discounted value of the **s'**
* $Q(s, a) = \sum_{s'} P(s' \mid s, a)(R(s', s, a) + \gamma V(s'))$ same thing written as summation of probabilities of ending in state s'

Value/Policy Iteration vs Reinforcement Learning

* in value iteration/policy iteration we were given both the **reward function** and the **transition function**

* $V(s) = \max_a Q(s, a)$ - **state value** is simply value of taking action that yielding maximum value for the given state

* $\pi(s, a) = argmax_a Q(s, a)$ - **optimum policy** that takes action yielding max value in each state

* in RL we need to discover the reward and transition functions through exploration

Bellman's Equation

* $V(s) = \max_{\pi} \mathbb{E}(r_0 + \gamma V(s'))$

Temporal Difference Learning

* $V(s_k) = \mathbb{E}(r_k + \gamma V(s_{k+1}))$ -> expected value for each state (Bellman optimality condition)
* to iteratively update the state value we do:

$$V^{new}(s_k) = V^{old}(s_k) + \overset{\text{weight}}{\alpha}\Big(\overbrace{\underbrace{\underset{\text{new info from the current step}}{r_k} + \gamma V^{old}(s_{k+1})}_{\text{TD Target Estimate}} - V^{old}(s_k)}^{\text{TD Error}}\Big)$$

This is TD(0) - just going 1 step into the future, but it could also be n-steps TD(N)

Q-Learning

* Q-Learning is just TD learning on a Q function!!!

$$Q^{new}(s_k, a_k) = Q^{old}(s_k, a_k) + \alpha(\overbrace{\underbrace{r_k + \gamma \max_a Q(s_{k+1}, a)}_{\text{TD Target Estimate}} - Q^{old}(s_k, a_k))}^{\text{TD Error}}$$

* What happens to the Q^{new} if I experience higher/lower reward than expected by Q^{old}?

Q-Learning Target Estimate

$$r_k + \gamma \max_a Q(s_{k+1}, a)$$

* r_k comes from the current step BUT not necessarily by following optimal policy => **exploration vs exploitation**

* $Q(s_{k+1}, a)$ - we are maximizing over action - i.e. using the action yielding max value for s_{k+1} - i.e. following current optimal policy

* Off policy - because it is not using current policy to take steps - **allows to learn by imitation or from experience replay**

SARSA
State-Action-Reward-State-Action

$$Q^{new}(s_k, a_k) = Q^{old}(s_k, a_k) + \alpha(r_k + \gamma Q^{old}(s_{k+1}, a_{k+1}) - Q^{old}(s_k, a_k))$$

* r_k - is coming from the current policy => On Policy algo
* always doing what you think is the best thing