Reinforcement Learning David Silver: Lecture 2 Notes

Name: Eli Andrew

- Reward: $G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$
  This is a single sample of G_t and therefore contains R rather than $E[R]$.
- Discount factor is useful as a way of dealing with uncertainty in your model. Current rewards not just because they are in the present but because future rewards are more uncertain due to the constraints on our model.
- Discounting also avoids infinte rewards.
- Value function: $v(s)$ gives long-term value of state s. $v(s) = E[G_t|S_t = s]$
- Bellman Equation for MRPs: value function can be decomposed into two parts
  - Immediate reward: R_{t+1}
  - Discounted future reward: $\gamma v(S_{t+1})$
  - Calculation:
    $v(s) = E[G_t|S_t = s]$
    $= E[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots |S_t = s]$
    $= E[R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + \dots)|S_t = s]$
    $= E[R_{t+1} + \gamma G_{t+1}|S_t = s]$
    $= E[R_{t+1} + \gamma v(S_{t+1})|S_t = s]$
- Bellman Equation expressed using matrices: $v = R + \gamma P v$ where v is a column vector with one entry per state.
- Solved directly the Bellman equation solution is $v = (I - \gamma P)^{-1} R$
- Iterative methods for large MDPs: (1) dynamic programming, (2) Monte-carlo simulation, (3) temporal-difference learning
- Markov Decision Process is a Markov Reward Process but with Actions. In other words the reward process is (S, P, R, γ) and the decision process is (S, A, P, R, γ)
- Policy definition: a distribution over actions given states $\pi(a|s) = P[A_t = a|S = s]$
- Because of the Markov property, we do not need to consider R in the policy because s fully characterizes the evolution from this point onwards.
- The policy itself, with the states that it picks, defines a Markov process (S, P^π), and the state and rewards the policy draws defines a Markov reward process, where:
  $P^\pi_{(s,s')} = \sum_{a \in A} \pi(a|s) P^a_{s,s'}$
  $R^\pi_{(s,s')} = \sum_{a \in A} \pi(a|s) R^a_s$

- State value function $v_\pi(s)$ of an MDP is the expected reward starting at state s and then following policy π: $v_\pi(s) = E_\pi[G_t|S_t = s]$
- Action value function $q_\pi(s, a)$ is the expected return starting from state s, taking action a, and then following policy π. $q_\pi(s, a) = E_\pi[G_t|S_t = s, A_t = a]$
- Decomposed state-value function: $v_\pi(s) = E_\pi[R_{t+1} + \gamma v_\pi(S_{t+1})|S_t = s]$
- Decomposed action-value function: $q_\pi(s, a) = E_\pi[R_{t+1}+\gamma q_\pi(S_{t+1}, A_{t+1})|S_t = s, A_t = a]$
- Bellman equation for V^π: $v_\pi(s) = \sum_{a\in A} \pi(a|s)q_\pi(s, a)$ where the policy is giving us the probability of taking the action a given we're in state s and the action-value function is giving us the value of the the action.
- Bellman equation for Q^π: $q_\pi(s, a) = R^a_s + \gamma \sum_{s'\in S} P^a_{ss'} v_\pi(s')$ where we are getting our immediate reward R^a_s for the current state and then the discounted reward over all possible states $s' \in S$ we could end up in when taking action a from state s.