CS 486 — Lecture 21: Reinforcement Learning

1 Passive and Active RL

- A passive agent has a fixed policy π and wants to learn $V^\pi(s)$, how good the policy is.
- An active agent must decide on what policy it should follow.

2 Active ADP Agent

- ADP — adaptive dynamic programming.
- Recall the passive ADP agent:
  - Learns the reward function $R(s)$ through the observed rewards.
  - Learns the transition probabilities P for the policy π.
  - Solves $V^\pi(s)$ using the simplified Bellman equation:

$$V^\pi(s) = R(s) + \gamma \sum_{s'} P(s'|s, \pi(s))V^\pi(s')$$

- An active ADP agent should also:
  - Learn the transition probabilities P for all (s, a).
  - Learn the values of $V^*(s)$, the expected utilities of the optimal policy for all the states.
- We see that if an agent follows the optimal policy of the learned model, it does not learn accurate utility values and the true optimal policy.
- For example, the agent might stick to a less safe route in our box word despite following the optimal policy (ie: same length as the safer route, but more dangerous)!
- So, we should also *explore* — we should take actions to improve the current learned model, and perhaps find new, better routes! This way, we might learn the true model.
- Note we can't just do pure exploration or pure exploitation — the latter may get stuck, the former will never improve and result in never applying what was learned by the agent.
- The optimal exploration policy we discuss now is known as the GLIE scheme.
  - GLIE: Greedy in the Limit of Infinite Exploration.
  - The agent must try each action in each state an unbounded number of times.
  - So, the agent eventually learns the true model and must eventually act in a greedy way.
  - We use the following update rule for value iteration:

$$V^+(s) = R(s) + \gamma \max_a f(\sum_{s'} P(s'|s, a)V^+(s), N(s, a))$$

  - $V^+(s)$ is the optimistic estimate of the utility of the state, s.
  - $N(s, a)$ is the number of times action a has been tried for the state.
  - $f(u, n)$ is the exploration function, trading off preference for high values of u and preference for low values of n.

- We prefer (s, a) that the agent hasn't tried very often, and actions that are of high utility.
- An example of the exploration function is:

$$f(u, n) = \begin{cases} R^+ & \text{if } n < N_e \\ u & \text{otherwise} \end{cases}$$

where R^+ is an optimistic estimate of the best possible reward obtainable in any state, and N_e is a fixed parameter.
- The agent will try each state-action pair at least N_e times.

3 Active TD Agent

- TD — temporal difference.
- Recall the passive TD agent:
  - When a transition occurs from s to s', update $V^\pi(s)$ as follows:

$$V^\pi(s) = V^\pi(s) + \alpha(R(s) + \gamma V^\pi(s') - V^\pi(s))$$

  - α is the learning rate, and it should decrease as the number of times a state has been visited increases.
  - $R(s) + \gamma V^\pi(s')$ is the target value of $V\pi(s)$ based on the transition.
- We can define an active TD agent as such:
  - Learn the utility values $V(s)$ via:

$$V^*(s) = V^*(s) + \alpha(R(s) + \gamma V^*(s') - V^*(s))$$

  - Learn the transition probabilities for all state-action pairs:

$$P(s'|s, a)$$

  - Determine the optimal policy using the utility values and the transition probabilities:

$$\pi^*(s) = \arg\max_a \sum_{s'} P(s'|s, a)V^*(s')$$

- Another way we can define an active TD agent is as follows:
  - We define $Q(s, a)$ as the expected total discount reward starting from the next state.
  - Meanwhile, we define $Q'(s, a)$, which obtains $R(s)$ and the expected total discounted reward starting from the next state.
  - Instead of using Q, we use Q' for the action-utility representation.
  - Let us define the equilibrium value for Q':

$$Q'(s, a) = R(s) + \gamma \sum_{s'} P(s'|s, a) \max_{a'} Q'(s', a')$$

  - And when we transition from s to s' by taking a, $Q'(s, a)$ should change to $R(s) + \gamma \max_{a'} Q(s', a')$.
  - So, the temporal difference equation is:

$$Q'(s, a) = Q'(s, a) + \alpha[R(s + \gamma \max_{a'} Q'(s'a') - Q'(s, a)]$$