Temporal Difference Learning

Temporal Difference

Idéa behind Temporal Difference:

Use that there are two estimates of the value of a state: *before* and *after*

- Estimate before the action

$$V^{\pi}(s_t)$$

- Estimate after the action

$$r_{t+1} + \gamma \cdot V^{\pi}(s_{t+1})$$

Important observation:

The second estimate is better!

Update the value estimate using the *difference*

$$V^{\pi}(s_t) \leftarrow V^{\pi}(s_t) + \eta\Delta$$

$$\Delta = [r_{t+1} + \gamma \cdot V^{\pi}(s_{t+1})] - V^{\pi}(s_t)$$

Δ serves as a measure of the surprise / disappointment

Learns *considerably faster* that the Monte-Carlo method

Q-learning

Problem:
An estimate of V is not sufficient for computing π since the agent does not have δ and r!

Trick:
Estimate $Q(s, a)$ instead of $V(s)$

$Q(s, a)$: Expected total reward when doing a from s.

$$\pi(s) = \arg\max_a Q(s, a)$$

$$V^{\star}(s) = \max_a Q^{\star}(s, a)$$

How can we learn Q?

The Q-function can also be learned using Temporal-Difference

$$Q(s, a) \leftarrow Q(s, a) + \eta \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right]$$

s' is the next state.

Off-policy learning

Q-learning finds the values for an optimal policy without any need to follow that policy

SARSA-learning

Almost the same as Q-learning, but one uses the current policy to select a':

$$Q(s, a) \leftarrow Q(s, a) + \eta \left[r + \gamma Q(s', a') - Q(s, a) \right]$$

The name comes from the experience-tuples structure:

$$< s, a, r, s', a' >$$

On-policy learning

SARSA finds the value of the policy used

What do we do when...

- The environment is not fully observable
- There are way too many states
- The states are not discrete
- The agent is acting in continuous time

The Exploration–Exploitation dilemma

If an agent strictly follows a greedy policy based on the current estimate of Q, learning is not guaranteed to converge to $Q^\star$

Simple solution:
Use a policy which has a certain probability of "making mistakes"

- ϵ-greedy
  Sometimes (with probability ϵ) make a random action instead of the one that seems best (greedy)
- Softmax
  Assign a probability to choose each action depending on how good they seem

Accelerated learning

Eligibility Trace

Idéa: TD updates can be used to improve not only the last state's value, but also states we have visited earlier.

$$\forall s, a : Q(s,a) \leftarrow Q(s,a) + \eta \left[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\right] \cdot e$$

e is a remaining trace (eligibility trace) encoding how long ago we were in s doing a.

Often denoted TD(λ) where λ is the time constant of the trace e