(Contd)

- Once we specify policy & fix actions for each state,
- state depends on $\underbrace{\pi(s)}_{\text{policy on state s}}$ & $\underbrace{\Pr(s_{t+1}=s' \mid s_t=s, a_t=a)}_{\text{transition probability}}$

Reward = total discounted reward

$R_t = r_t + \gamma r_{t+1} + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots$

↳ discounted sum of all rewards obtained from time t.

↓

prevents reward from going to infinity

Value Iteration VS Policy Iteration

- Constantly refines value function v (or Q)

Find $Q(s, a)$

$a = \arg\max Q(s, a)$

- Defines policy function that converges to most optimal policy (through policy gradient)

Find $\pi(s)$

Sample $a \sim \pi(s)$

} both uses MDP

Other RL algorithms

SARSA: (State-Action-Reward-State-Action) uses MDP to adjust value of Q-function based on next state (modified Q-learning that uses extra action & state

Monte Carlo Methods: Directly learns from experience & past a-s pairs without any prior knowledge of MDP probs. MC uses policy iteration.