Q 1.1 Consider an MDP with 2 states \( \{A, B\} \) and 2 actions: “stay” at current state and “move” to other state. Let \( r \) be the reward function such that \( r(A) = 1, r(B) = 0 \). Let \( \gamma \) be the discounting factor. What is the optimal policy \( \pi^*(A) \) and \( \pi^*(B) \)? What are \( V^*(A), V^*(B) \)?

- A. Stay, Stay, \( 1/(1-\gamma), 1 \)
- B. Stay, Move, \( 1/(1-\gamma), 1/(1-\gamma) \)
- C. Move, Move, \( 1/(1-\gamma), 1 \)
- D. Stay, Move, \( 1/(1-\gamma), \gamma/(1-\gamma) \)
Q 1.1 Consider an MDP with 2 states \{A, B\} and 2 actions: “stay” at current state and “move” to other state. Let \( r \) be the reward function such that \( r(A) = 1, r(B) = 0 \). Let \( \gamma \) be the discounting factor. What is the optimal policy \( \pi^*(A) \) and \( \pi^*(B) \)? What are \( V^*(A), V^*(B) \)?

- A. Stay, Stay, \( 1/(1-\gamma) \), 1
- B. Stay, Move, \( 1/(1-\gamma) \), \( 1/(1-\gamma) \)
- C. Move, Move, \( 1/(1-\gamma) \), 1
- D. Stay, Move, \( 1/(1-\gamma) \), \( \gamma/(1-\gamma) \)
Q 1.1 Consider an MDP with 2 states \{A, B\} and 2 actions: "stay" at current state and "move" to other state. Let \( r \) be the reward function such that \( r(A) = 1, r(B) = 0 \). Let \( \gamma \) be the discounting factor. What is the optimal policy \( \pi^*(A) \) and \( \pi^*(B) \)? What are \( V^*(A), V^*(B) \)?

• A. Stay, Stay, \( 1/(1-\gamma) \), 1
• B. Stay, Move, \( 1/(1-\gamma) \), \( 1/(1-\gamma) \)
• C. Move, Move, \( 1/(1-\gamma) \), 1
• D. Stay, Move, \( 1/(1-\gamma) \), \( \gamma/(1-\gamma) \) Note: want to stay at A, if at B, move to A. Starting at A, sequence A,A,A,... rewards 1, \( \gamma \), \( \gamma^2 \),.... Start at B, sequence B,A,A,... rewards 0, \( \gamma \), \( \gamma^2 \),.... Sums to \( 1/(1-\gamma) \), \( \gamma/(1-\gamma) \).
Q 2.1 For Q learning to converge to the true Q function, we must

- A. Visit every state and try every action
- B. Perform at least 20,000 iterations.
- C. Re-start with different random initial table values.
- D. Prioritize exploitation over exploration.
Q 2.1 For Q learning to converge to the true Q function, we must

• A. Visit every state and try every action
• B. Perform at least 20,000 iterations.
• C. Re-start with different random initial table values.
• D. Prioritize exploitation over exploration.
Break & Quiz

Q 2.1 For Q learning to converge to the true Q function, we must

• A. Visit every state and try every action
• B. Perform at least 20,000 iterations. (No: this is dependent on the particular problem, not a general constant).
• C. Re-start with different random initial table values. (No: this is not necessary in general).
• D. Prioritize exploitation over exploration. (No: insufficient exploration means potentially unupdated state action pairs).