מבוא לבינה מלאכותית - תשפ"ד - תרגיל 5

אחמד דלאשה 324059856 | עבד נסאר 207063108

9 באוגוסט 2024

Theoretical questions :

Question number 8 :

The computational complexity of the Value Iteration (VI) algorithm depends on several factors, including the number of states $|S|$, the number of actions $|A|$, the number of iterations k, and the cost of computing the expected value for each state-action pair. Let's break down the complexity step by step.

Factors Influencing Complexity

1. Number of States ($|S|$): The total number of states in the Markov Decision Process (MDP).
2. Number of Actions ($|A|$): The total number of actions available in each state.
3. Number of Iterations (k): The number of iterations the algorithm runs until convergence.
4. Transition and Reward Computation: For each state-action pair, we need to compute the expected value, which involves summing over the possible next states.

Step-by-Step Complexity Analysis

1. Initialization:
   - Initializing the value function V for all states takes $O\left(|S|\right)$
2. Value Update in Each Iteration:
   - For each state s, for each action a, we need to compute the expected value:

   $$V\left(s\right) \leftarrow max_a \sum_{s'} P\left(s' \mid s, a\right) \left[R\left(s, a, s'\right) + \gamma V\left(s'\right)\right]$$

   - This involves:
     - Transition Probability Summation: Summing over all possible next states s' to compute the expected value. If we assume the number of next states is proportional to $|S|$, this summation takes $O\left(|S|\right)$.
     - Maximization: Maximizing over all actions for each state, which takes $(O\left(A\right))$.
3. Total Cost per Iteration:
   - For each state s, we consider all actions a, and for each action, we sum over all next states s'. Therefore, the complexity per state-action pair is $O\left(|S|\right)$.

- The total complexity per iteration is $O(|S| \times |A| \times |S|)$

4. Number of Iterations (S):

   - The algorithm runs for k iterations until convergence. In the worst case, k could be large, but for practical purposes, it's often a fixed number of iterations or until the value function changes by less than a threshold $\emptyset$.

Overall Complexity

Combining these factors, the overall computational complexity of the Value Iteration algorithm is:

$$O\left(k \times |S|^2 \times |A|\right)$$

Question number 9 :

Discount Factor (γ)

- **High ($\gamma \approx 1$)**: Values future rewards highly. Encourages long-term planning and risk-taking for higher rewards (e.g., distant exit with +10).
- **Low ($\gamma \approx 0$)**: Values immediate rewards more. Encourages safer, short-term gains (e.g., close exit with +1).

Noise

- **High**: Increases action uncertainty. Encourages risk-averse behavior to avoid unintended negative outcomes (e.g., avoiding the cliff).
- **Low**: Decreases action uncertainty. Encourages taking direct and risky paths for higher rewards (e.g., crossing the bridge).

Living Reward

- **High (positive)**: Rewards staying alive. Encourages prolonged episodes and exploration, avoiding exits and risks.
- **Low (negative or zero)**: Penalizes each step. Encourages quick resolution to minimize penalties, even if it means taking risks (e.g., quickly reaching exits).

Question number 10 :

Boltzmann Exploration (Softmax)

Description: In Boltzmann exploration, actions are selected based on a probability distribution derived from the Q-values of the actions. The probability of selecting an action a in state s is given by:

$$P(a|s) = \frac{e^{Q(s,a)/\tau}}{\sum_{a'} e^{Q(s,a')/\tau}}$$

where τ is the temperature parameter controlling exploration.

Effects on Learning Process:

- **Number of Times an Action is Selected:**
  - **High Temperature (τ)**: Actions are selected more uniformly, leading to extensive exploration.
  - **Low Temperature (τ)**: Actions with higher Q-values are selected more often, reducing exploration.
- **Variability of the Estimated Q-values:**
  - **High Temperature**: Results in varied Q-value estimates due to thorough exploration.
  - **Low Temperature**: Focuses on higher Q-value actions, leading to quicker but potentially suboptimal convergence.

Comparison to ϵ-greedy:

- **Exploration:**
  - ϵ-greedy: Selects a random action with probability ϵ, otherwise selects the best action.
  - Boltzmann: Selects actions probabilistically based on Q-values, offering refined exploration.
- **Action Selection:**
  - ϵ-greedy: Can choose completely random actions during exploration.
  - Boltzmann: More likely to select higher-valued actions, even during exploration.
- **Learning Variability:**
  - ϵ-greedy: High variability due to potentially poor action selection during exploration.
  - Boltzmann: More stable Q-value estimates due to focused exploration.