Intelligent Agents
2024
Quiz
7. November 2024

- place your student ID card (carte de legitimation) on the desk in front of you.
- this is an open-book examination: all non-electronic documents allowed.
- when choosing the right answer, consider that the given explanation also has to be correct.
- mark the number of your copy on the top of each page to make sure we identify all pages of your exam.
- for questions with a single answer, the correct answer gives you 3 points.
- one question has multiple answers and each correct answer will give you 2 points.
- for each incorrect answer, you lose one point.

Seat No:

1. In Q-learning, how is the Q-table computed, assuming that the rewards r are between 0 and 1 and that we want to ensure that all actions are eventually explored?

   a) Start with all 1 and form a weighted average $\alpha r + (1 - \alpha)q(s)$
   b) Start with all 0 and form a weighted average $\alpha r + (1 - \alpha)q(s)$
   c) Start with all 1 and replace by observed discounted reward $r + \gamma q(s')$
   d) Start with all 0 and replace by observed discounted reward $r + \gamma q(s')$

   Your answer: *a*

   b,d: starting from 0 means actions with unknown rewards are never tried.
   c,d: the formula is the definition of the q-value, not the update used in learning.

2. What is the difficulty with using UCB in multi-agent learning (multiple answers)?

   a) It requires observation of the rewards obtained by other agents.
   b) An agent has no influence on actions of other agents.
   c) When other agents change their strategies, the confidence bounds are no longer valid.
   d) If other agents also choose their actions according to an exploration strategy, the observed rewards are not stable and the bounds are not accurate.

   Your answer: *c,d*

   a: UCB could not even use this information.
   b: Forcing actions of other agents would help with exploration, but is not foreseen in UCB.

3. What problem of UCB does Thompson sampling solve?

   a) Need to know rewards of actions that were not actually taken.
   b) Robustness against an adversary who designs an unfavorable payoff function.
   c) Slow convergence to the correct bounds.
   d) Inability to make use of correlations among the rewards for different actions.

   Your answer: *d*

   a: UCB also does not need to know these counterfactual rewards.
   b: UCB and Thompson sampling both are not designed for adversarial settings.
   c: In theory, Thompson sampling converges slower than UCB.

4. What is the underlying algorithm in depth-limited search?

   a) Heuristic search (A*)
   b) Breadth-first search
   c) Depth-first search
   d) Monte-Carlo tree search

   Your answer: *c*

   a: A has a notion of cost, not depth.*
   b: breadth-first search already searches in the order of depth, so depth limits make no sense.
   d: the random sampling in Monte-Carlo search is not related to depth limits.

5. What is the main difference between minimax search and Monte-Carlo tree search?

   a) In minimax search, the moves are selected to minimize the adversary's gain, while in Monte-Carlo search, they are selected randomly.
   b) in minimax search, horizon states are evaluated using a deterministic evaluation function, while in Monte-Carlo search, their value is estimated as the average of a set of random roll-outs.
   c) Minimax search applies to deterministic scenarios, while Monte-Carlo search applies to scenarios with chance (like games with dice rolls).
   d) Minimax search is guaranteed to find the optimal move, while Monte-Carlo search only does so with some probability.

   Your answer: *b*

   a: the rollout strategies in Monte-Carlo tree search can also select moves to minimize the adversary's gain.
   c: the Monte-Carlo refers to the fact that horizon states are evaluated by random sampling, not to the uncertainty in the scenario.
   d: because of the need to approximate the values of horizon states, minimax search in general does not guarantee the optimal move either.

6. In the context of the exercises, which of the following statements are true (multiple answers):

   a) Reactive agents are well suited for situations where you have only probabilistic knowledge of the environment.
   b) Deliberative agents will always find the optimal solution when using A*.
   c) Deliberative agents will always find the optimal solution when using A* with heuristic h = 0.
   d) The first solution of BFS is always the same as the first solution of A*.

   Your answer: *a,c*

   b: It depends on the heuristic.
   d: If each action has a different cost, then the least number of actions is not equivalent to least cost.

7. Why are factored representations particularly important for planning with multiple agents?

   a) To avoid the combinatorial complexity when selecting actions for many agents at the same time.
   b) To avoid the combinatorial complexity when combining multiple individual state spaces.
   c) Because they allow to combine the objectives of multiple agents into a single objective.
   d) Because they make each agent's planning problem easier to solve.

   Your answer: *c, (b)*

   a: This complexity still exists and is not solvable by factoring.
   b: It does help, but this is not the main reason. Since many students answered this, we accepted this answer as well.
   d: true but it's not more important in multi-agent settings.

8. What is the role of mutual exclusion (mutex) constraints in planning?

   a) Ensure that among alternative actions for the same effect, only one is used in the plan.
   b) Model the fact that an agent can only carry out one action at a time.
   c) Ensure that no action invalidates a precondition of another action that can be taken at the same time.
   d) Model the inconsistencies that exist among values of state variables.

   Your answer: *c*

   a: only one action is considered anyway.
   b: parallel actions are still allowed.
   d: they avoid inconsistencies, not model them.

9. In reinforcement learning, what does importance sampling do?

   a) Sample states with a frequency proportional to their value function.
   b) Train only on the most frequently occurring states and infer the policy for other states by extrapolation.
   c) Train on frequently occurring states less often and upweight their importance in the loss function.
   d) Sample states with a frequency proportional to their observed rewards.

   Your answer: *c*

   a: it could be an interesting idea as the policy should drive to the states with highest values, but it's not importance sampling.
   b: we exactly want to gather more data on the infrequent states to avoid errors caused by extrapolation.
   d: this would focus too much on rewards rather than how to get to the states with the rewards.