6.4110/16.420
Representation, Inference and Reasoning in AI

Midterm Exam

Solutions

November 2, 2022

Answer the questions in the spaces provided on the question sheets. If you run out of room for an answer, continue on the back of the page.

You are permitted to use a single sheet of paper with notes on (both sides), and a calculator and a timer. If you use your phone for the calculator and timer, please restrict yourself to these functions.

Name: ______________________________

MIT email: ______________________________

Question	Points	Score
1	20	
2	9	
3	19	
4	12	
5	18	
6	22	
Total:	100	

6.4110/16.420
Midterm Exam, November 2, 2022

1 Search

1.1 Driving

1. (20 points) You are driving an electronic vehicle over a road network that connects up a set of small towns. You know the locations of the towns and the road map, which contains at most one road between each pair of distinct towns. The roads are all two-way.

   - The length of each road segment r is $length(r)$.
   - Your battery can hold m units of charge, maximum.
   - Every mile requires 1 unit of charge.
   - You drive 1 mile per minute.
   - Some towns have chargers that charge at the rate of 10 units per minute, and some have no charger at all.
   - If you run out of charge you can no longer continue.
   - Some roads have cell service and some do not. Your car is so modern it does not have a radio, so if there's no cell service you will be bored. When you are bored, your perceived time goes slower — you will perceive time twice as long as the actual time. (You won't be bored when charging the vechicle, since you can always walk around the town).
   - You would like to find a path (including charging stops) to a goal town that **minimizes your overall perceived time**.

   Formulate this as a min-cost-path problem, where the action space consists of $drive(r)$ where r is a road segment connecting two towns, and $charge(t)$ where $t \in \{10, 20, \ldots, 100\}$ is a duration of charging in minutes.

   (a) (4 points) What is a minimal space of states for this problem?

   > **Solution:** All possible values of (town, battery-charge)

   (b) (4 points) In what states s is the action $drive(r)$ executable?

   > **Solution:** When the town in s is equal to the start town of r and $length(r)$ is less than or equal to the battery charge in s.

   (c) (4 points) What is the cost function $C(s, a, s')$ (in units of minutes)?

   > **Solution:** If $a = drive(r)$ then $length(r)$ if r is not boring, else $2 \cdot length(r)$. Otherwise, for charge action with argument t the cost is t.

   (d) (4 points) Approximately what is the computational complexity of uniform-cost search in this problem, assuming n towns, m battery levels, every town has k roads connecting to it, and the optimal path has d actions?

   > **Solution:** UCS/Djikstra is $O(|E| \log |V|)$. In this problem $|V| = O(mn)$ (number of towns times charge levels) and $|E| = O(k + m)$ (k drive actions and 10 charge actions). Or, if we don't expand the whole space (and all the costs are 1) then the complexity is $O(b^d)$, where b is the average branching factor, $k + 10$ in this case. The actual answer is the minimum of these expressions: $O(\min(mnk \log(mn), (k + 10)^d))$ Note that the complexity of UCS in the general cost case (see AIMA) is actually $O(b^{1+C^*/\epsilon})$ and C^* is optimal path cost and ϵ is smallest action cost. But, we did not give these parameters in the problem, although we accepted that answer.

For each of the following heuristics, indicate whether it is admissible and whether it has a smaller search space than the original problem.

(e) (2 points) Minimum cost path if the battery level never decreases.
$\surd$ **admissible** $\surd$ **smaller search space than original**

(f) (2 points) Minimum cost path when the problem is the same as the original, but the penalty for boredom is twice as high.
◯ admissible ◯ smaller search space than original

1.2 Flying

2. (9 points) Now, let's assume that you are a drone, flying above the terrain, but staying at a fixed altitude.

- You can still go 1 mile per minute, your battery still holds m units, and the chargers and charging time remain the same (this is a *big* drone)!
- Drones never get bored.
- There are some high peaks that prevent you from moving within some regions of the space at your fixed altitude, essentially forming obstacles in a planar problem.
- This drone has no problem hovering and is insensitive to wind, so you can assume that forces and velocities are handled by the controller and you only need to worry about paths and charging.
- Drone motion is holonomic; it can move in any direction.
- The chargers are in the same place, and your start and goal locations will be towns, but the road network is irrelevant.

You can do long-distance induction charging (!). You are given a function $\textit{charge-delta}(l_i, l_j)$ that returns the aggregate change in charge (which may be positive or negative) due to flying from l_i to l_j in a straight line. Your battery has over-charge protection, so that it will stop accumulating charges when it is at max-capacity.

You decided to use trajectory optimization to approach this problem.

The overall cost of the trajectory is the sum of the costs of n individual linear segments.

$$J(((l_0, b_0), (l_1, b_1), \ldots, (l_n, b_n)) = \sum_i C((l_i, b_i), (l_{i+1}, b_{i+1}))$$

where $l_0 =$ `start`, b_0 is the initial battery charge, and $l_n =$ `goal`. The cost for each segment includes several terms, including the segment length, the degree to which the segment penetrates an obstacle, and the degree to which the charging dynamics are respected (α and β are constants that trade off the relative importance of the terms):

$$C((l_i, b_i), (l_j, b_j)) = |l_i - l_j|^2 + \alpha \cdot \textit{penetration}((l_i, l_j, \textit{obstacles})) + \beta \cdot \textit{charge}_{i,j}$$

(a) (2 points) Which of the following terms captures the degree to which charging dynamics are respected ($\textit{charge}_{i,j}$)?

◯ $|b_i - b_j|^2$
◯ $|b_i + b_j|^2$
◯ $|b_i + \textit{charge-delta}(l_i, l_j) + b_j|^2$
$\surd$ $|b_i + \textit{charge-delta}(l_i, l_j) - b_j|^2$

(b) (3 points) Explain your choice from above.

Solution: The change in charge between the endpoints has to match the charge-delta.

(c) (2 points) However, we're missing a term! What is it?

Solution: A penalty for c going negative.

(d) (2 points) Of these four terms (the three we listed, plus the one you provided), which ones, if any, need to be driven to 0 to obtain a legal trajectory?

◯ length √ **penetration** √ **charge-delta** √ **missing**

2 Symbolic representations

2.1 First-order Logic

3. (19 points) Let's think about a domain with 100 rooms and 100 keys. Here are some predicates:

- $\textit{is-room}(r)$
- $\textit{in}(k, r)$
- $\textit{connected}(r_1, r_2)$
- $\textit{opens}(k, r)$
- $\textit{unlocked}(r)$

(a) (5 points) Use the predicates above to write a first-order logic sentence encoding the constraint that if a key for a room is in that room, then the room is unlocked.

Solution:

$$\forall r.(\exists k.\textit{opens}(k, r) \wedge \textit{in}(k, r)) \rightarrow \textit{unlocked}(r)$$

(b) (5 points) Imagine that we are given the following facts, where R_a and R_b are constants naming rooms and K_a and K_b are constants naming keys:

- $\textit{connected}(R_a, R_b)$
- $\textit{in}(K_b, R_b) \wedge \forall x.\textit{in}(x, R_b) \rightarrow x = K_b$
- $\textit{opens}(K_a, R_a)$
- $\forall r.\forall k.\textit{opens}(k, r) \rightarrow (\textit{in}(k, r) \vee \exists r'.\textit{connected}(r, r') \wedge \textit{in}(k, r'))$
- $\forall r.\forall r'.\textit{connected}(r, r') \rightarrow \textit{connected}(r', r)$

Do these facts entail that K_a is in R_a? If so, outline the reasoning informally in English. If not, explain informally in English a set of additional fact(s) you could assume (not including $\textit{in}(K_a, R_a)$) so that the whole set would entail $\textit{in}(K_a, R_a)$.

Solution: We need to assume that there are no other rooms connected to R_a.

(c) (9 points) Now, let's assume that

- Every room has a key that opens it.
- The key to any room is either in that room or in a neighboring room.
- There are no keys in the rooms neighboring R_1.

From this, we would like to prove that there is a key to R_1 and is in R_1.

Below, we have written out a proof of this via resolution refutation, with several missing pieces. The first 5 lines contain the assumptions and negated conclusion, converted into clausal form (the functions κ and ν are Skolem functions handling the existential variables.)

Please fill in the rest of the proof! The third column should include the line numbers and the substitution needed to apply the resolution rule.

1	$opens(\kappa(x), x)$	assumption
2	$\neg opens(y, z) \vee in(y, z) \vee connected(z, \nu(z))$	assumption
3	$\neg opens(y, z) \vee in(y, z) \vee in(y, \nu(z))$	assumption
4	$\neg connected(R_1, a) \vee \neg in(b, a)$	assumption
5	$\neg opens(c, R_1) \vee \neg in(c, R_1)$	negated conclusion
6	$in(\kappa(z), z) \vee connected(z, \nu(z))$	**1, 2,** $\{x/z, y/\kappa(z)\}$
7	$in(\kappa(z), z) \vee in(\kappa(z), \nu(z))$	**1, 3,** $\{x/z, y/\kappa(z)\}$
8	$\neg in(\kappa(R_1), R_1)$	**1, 5,** $\{c/\kappa(R_1)\}$
9	$connected(R_1, \nu(R_1))$	6, 8, $\{z/R_1\}$
10	$in(\kappa(R_1), \nu(R_1))$	7, 8, $\{z/R_1\}$
11	$\neg in(b, \nu(R_1))$	4, 9 $\{a/\nu(R_1)\}$
12	**False**	10, 11, $\{b/\kappa(R_1)\}$

2.2 PDDL

4. (12 points) Let's continue the theme of keys and rooms, but this time writing operator descriptions. Use the following predicate definitions, with the obvious interpretations when `k` is a key and `r` is a room.

```
(opens ?k ?r)
(connected ?r1 ?r2)
(unlocked ?r)
(robot-in ?r)
(robot-holding ?k)
(in ?k ?r)
```

(a) (4 points) Here is an operator describing how a robot can move from room to room. In order to move between rooms, the rooms must be connected and the destination unlocked.

```
(:action move
    :parameters (?r1 ?r2)
    :precondition (and (connected ?r1 ?r2) (unlocked ?r2) (robot-in ?r1))
    :effect (and (robot-in ?r2) (not (robot-in ?r1))))
```

Complete the operator description for unlocking a room. In order to unlock a room, the robot must be in a connected room and holding a key to the room it is unlocking.

```
(:action unlock
    :parameters (?r1 ?r2 ?k)
    :precondition


    :effect


)
```

Solution:

```
(:action unlock
    :parameters (?r1 ?r2 ?k)
    :precondition (and (connected ?r1 ?r2) (robot-in ?r1)
                       (opens ?k ?r2) (robot-holding ?k))
    :effect (and (unlocked ?r2)))
```

(b) (8 points) Assume there is one more operator

```
(:action pick-up
    :parameters (?r ?k)
    :precondition (and (robot-in ?r)
                       (in ?k ?r))
    :effect (and (robot-holding ?k) (not (in ?k ?r))))
```

Here is an initial state:

```
(connected R0 R1) (connected R1 R2) (connected R2 R3)
(connected R1 R0) (connected R2 R1) (connected R3 R2)
(unlocked R1) (unlocked R2) (unlocked R3)
(robot-in R1) (opens K0 R0) (in K0 R3)
```

And goal: `(and (robot-in R0) (unlocked R0) (robot-holding K0))`

We want to base a heuristic on a relaxed plan graph for this problem.

i. First, what is the shortest solution for this problem? Please list the actions.

> **Solution:**
> `(move R1 R2) (move R2 R3)`
> `(pick-up K0) (move R3 R2) (move R2 R1)`
> `(unlock R1 R0 K0) (move R1 R0)`

ii. At what level in the RPG is the fluent `(robot-holding K0)`? **3**

iii. At what level in the RPG is the fluent `(unlocked R0)`? **4**

iv. What is the value of H_{add} for this goal? **12**

v. What is the value of H_{max} for this goal? **5**

3 Probabilistic inference

Let's think some more about rooms and keys!

3.1 Factor graphs

5. (18 points) Let us again imagine a domain, now with 100 rooms and 100 keys, one for each room. Now, assume we're completely uncertain about the locations of the keys—it's equally likely that each key is in each room, multiple keys can all be in the same room, but no key can be in multiple rooms. Beyond these constraints the key locations are independent.

   (a) (3 points) We could formulate this problem in two different ways:

   - Have one random variable for each key, which can take on values representing the possible rooms it might be in.
   - Have one random variable for each room, which can take on values representing the possible sets of keys that might be in it.

   Both of these formulations can be used but the second one is significantly more difficult to work with. Explain why.

   Solution: The domains of the random variables in the second formulation are very large and we have to do extra work to encode that each key is in exactly one room, but that is automatically encoded in the first formulation.

   (b) (3 points) We will use the first formulation, but for **only three rooms and three keys**. Our initial factor graph has three independent random variables, K_1, K_2, and K_3, with a factor attached to each one. These factors share the same table (with domain $\{r_1, r_2, r_3\}$). Provide that table.

   Solution: $\{r_1 : 1/3, r_2 : 1/3, r_3 : 1/3\}$

(c) (6 points) We look in room 1 and see (for sure) that it contains a single key, but we can't tell which one. We can model this observation by adding a new factor to the graph, which is connected to all three random variables.

The entries in this new factor can all be either 0 or 1. Fill in the rows in that factor's table that have value 1 (we can assume all other rows have value 0) in this factor.

K_1	K_2	K_3	value
r_1	r_2	r_2	1
r_1	r_2	r_3	1
r_1	r_3	r_2	1
r_1	r_3	r_3	1
r_2	r_1	r_2	1
r_2	r_1	r_3	1
r_2	r_2	r_1	1
r_2	r_3	r_1	1
r_3	r_1	r_2	1
r_3	r_1	r_3	1
r_3	r_2	r_1	1
r_3	r_3	r_1	1

(d) (2 points) What is the posterior distribution on the location of key 1?

> **Solution:** $\{r_1 : 1/3, r_2 : 1/3, r_3 : 1/3\}$

(e) (2 points) Now, assume we obtain evidence that key 2 is in r_2. This changes the factor on node K_2 to be $\{r_1 : 0, r_2 : 1, r_3 : 0\}$. What are the non-zero rows in the table resulting from multiplying this table with the one from the previous step?

K_1	K_2	K_3	value
r_1	r_2	r_2	1
r_1	r_2	r_3	1
r_2	r_2	r_1	1
r_3	r_2	r_1	1

(f) (2 points) What is the posterior on the location of key 1?

> **Solution:** $(.5, .25, .25)$

3.2 HMMs: Are my leftovers edible?

6. (22 points) I use an HMM to model the state of my leftover food over time. The state space is $\{tasty, smelly, furry\}$. Initially, I'm sure they're tasty: $P(S_0 = tasty) = 1$. The state transition model is:

		S_{t+1} tasty	S_{t+1} smelly	S_{t+1} furry
	tasty	$\frac{2}{3}$	$\frac{1}{3}$	0
S_t	smelly	$\frac{1}{6}$	$\frac{1}{2}$	$\frac{1}{3}$
	furry	0	0	1

(a) (6 points) What is the state distribution $P(S_2 \mid S_0 = tasty)$? Show your work. It's fine to leave numerical quantities in terms of expressions (e.g. `7 * 5 / 3`).

Solution: $P(S_1) = (\frac{2}{3}, \frac{1}{3}, 0)$ obtained directly from the transition model. Now, we can compute the joint on S_1, S_2 by multiplying the this factor by the table, getting

		S_2 tasty	S_2 smelly	S_2 furry
	tasty	$\frac{2}{3} \times \frac{2}{3}$	$\frac{1}{3} \times \frac{2}{3}$	$0 \times \frac{2}{3}$
S_1	smelly	$\frac{1}{6} \times \frac{1}{3}$	$\frac{1}{2} \times \frac{1}{3}$	$\frac{1}{3} \times \frac{1}{3}$
	furry	0×0	0×0	1×0

Now, we marginalize out S_1 to get a factor on S_2:

$$\left(\frac{2}{3} \times \frac{2}{3} + \frac{1}{6} \times \frac{1}{3}, \frac{1}{3} \times \frac{2}{3} + \frac{1}{2} \times \frac{1}{3}, \frac{1}{3} \times \frac{1}{3}\right)$$

$$\left(\frac{1}{2}, \frac{7}{18}, \frac{1}{9}\right)$$

(b) (2 points) You observe your leftovers on day 4 and they're definitely furry ($S_4 = furry$). You decide to formulate this inference problem as a factor graph and use message passing to find a distribution on the state of your leftovers on day 2. What is the table in the factor connecting nodes S_2 and S_3?

Solution: The transition table, with S_2 in the role of S_t and S_3 as S_{t+1}.

(c) (5 points) Let ϕ_{ij} be the factor between nodes S_i and S_j. Provide the values of the following messages (we calculated a couple of them for you). You can provide a vector or matrix of numbers, with the assumption that the possible values are, in order, (tasty, smelly, furry). We're asking for the messages in the order we would want them to compute $P(S_2)$.

i. $S_0 \to \phi_{01} = (1, 0, 0)$

ii. $\phi_{01} \to S_1$

Solution: $\left(\frac{2}{3}, \frac{1}{3}, 0\right)$

iii. $S_1 \to \phi_{12}$

Solution: $\left(\frac{2}{3}, \frac{1}{3}, 0\right)$

iv. $\phi_{12} \to S_2$

Solution: $\left(\frac{1}{2}, \frac{7}{18}, \frac{1}{9}\right)$

v. $S_4 \to \phi_{34} = (0, 0, 1)$

vi. $\phi_{34} \to S_3$

Solution: $\left(0, \frac{1}{3}, 1\right)$

vii. $S_3 \to \phi_{23}$

Solution: $\left(0, \frac{1}{3}, 1\right)$

viii. $\phi_{23} \to S_2 = \left(\frac{1}{9}, \frac{1}{2}, 1\right)$

(d) (3 points) What is $P(S_2 \mid S_0 = \textit{tasty}, S_4 = \textit{furry})$? Just explain how to compute this using the messages you generated above (without numbers).

> **Solution:** Multiply the messages coming in from ϕ_{12} and from ϕ_{23}, then normalize.

(e) (3 points) You decide to estimate $P(S_2)$ by forward (prior) sampling. You generate 100 sample trajectories of length 3 for your leftovers. Given those trajectories how can you construct an estimate of $P(S_2)$?

> **Solution:** (number of trajectories ending in tasty / 100, number of trajectories ending in smelly / 100, number of trajectories ending in furry / 100)

(f) (3 points) You decide that three discrete values is not sufficient precision for describing the state of your leftovers! You would like to score them on a continuum from 0.0 (completely fresh and tasty) to 1.0 (totally decomposed). Would a Kalman filter be a good choice of solution for predicting the current state of your leftovers given some previous observations? Explain why or why not.

> **Solution:** No, because we need the distributions to be Gaussian and this one is over a fixed interval.