Sample questions for COMP-424 final exam

Doina Precup

These are examples of questions from past exams. They are provided without solutions. However, Doina and the TAs would be happy to answer questions about the solutions, if you try them. Note that the exam also has questions similar to those on the homeworks.

1. **Search algorithms**

   (a) Suppose you have an admissible heuristic h. Is h^2 admissible? Is $\sqrt{h}$ admissible? Would using any of these alternatives be better or worse than using h in the A^* algorithm?

   Answer: h^2 may not be admissible, because $h^2 \geq h$ when $h \geq 1$, so it may exceed the optimal distance to goal. $\sqrt{h} \leq h$ for $h \geq 1$, so it is admissible (assuming integer values for the heuristic, which is typical). It will likely work worse than h, though, because its estimate is farther from the optimal value (so it is a worse estimate of the cost-to-go).

   (b) Suppose you were using a genetic algorithm and you have the following two individual, represented as strings of integers:

   1324421 and 2751421

   Show the result of performing crossover between the 3rd and 4th digit.

   Answer: 1321421 and 2754421

   (c) Which of the following statements, contrasting genetic algorithms and simulated annealing, are true?

      i. Genetic algorithms are used for minimization problems while simulated annealing is used for maximization problems
      Answer: False

      ii. Genetic algorithms maintain several possible solutions, whereas simulated annealing works with one solution.
      Answer: True

      iii. Genetic algorithms maintain one solution, whereas simulated annealing maintains several possible solutions.
      Answer: False

      iv. Simulated annealing is guaranteed to produce the best solution, while genetic algorithms do not have such a guarantee.
      Answer: True

2. **Machine learning**

   (a) Which of the following can learn the OR function (circle all that apply):

   i. linear perceptron
   Answer: yes

   ii. a single sigmoid neuron
   Answer: yes

   iii. a network of sigmoid neurons with one hidden layer
   Answer: yes

   iv. none of the above

   (b) Which of the following can learn the XOR function (circle all that apply):

   i. linear perceptron
   Answer: no

   ii. a single sigmoid neuron
   Answer: no

   iii. a network of sigmoid neurons with one hidden layer
   Answer: yes

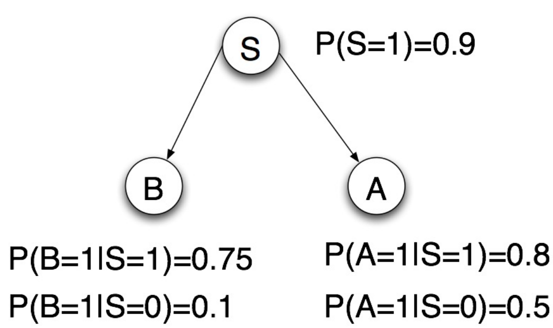
   iv. none of the above

3. **Bayes nets**

   The Starfleet academy has decided to create a class of android students. 90% of these androids study hard for their exams. Out of the androids who study hard for an exam, 80% get an A. Out of the androids who do not study, only half get an A. Androids who study hard have a 75% probability of depleting their battery in less that a day. Androids who do not study hard have a longer battery life: only 10% of them deplete their batteries within the next day.

   (a) Draw a Bayes net describing the problem statement above.

   Answer: Let A be the random variable denoting whether the Android gets an A in the exam, S denote whether he studies, and B denote whether the battery gets depleted in less than a day. The Bayes net is as follows:



(b) You notice that your android has depleted its battery in less than a day. What is the probability that it will get an A on the exam it had yesterday?

Answer: The evidence is $B = 1$ and you want to compute $P(A = 1|B = 1)$. We can do this as follows:

$$
\begin{aligned}
P(A=1|B=1) &= \frac{P(A=1,B=1)}{P(B=1)} \\
P(A=1,B=1) &= P(A=1,B=1,S=1) + P(A=1,B=1,S=0) \\
&= P(S=1)P(A=1|S=1)P(B=1|S=1) + P(S=0)P(A=1|S=0)P(B=1|S=0 \\
&= 0.9*0.75*0.8 + 0.1*0.1*0.5 \\
P(A=0,B=1) &= P(A=0,B=1,S=1) + P(A=0,B=1,S=0) \\
&= P(S=1)P(A=0|S=1)P(B=1|S=1) + P(S=0)P(A=0|S=0)P(B=1|S=0 \\
&= 0.9*0.75*0.2 + 0.1*0.1*0.5 \\
P(B=1) &= P(A=1,B=1) + P(A=0,B=1)
\end{aligned}
$$

You can now finalize the calculations.

(c) Your friend does not believe that androids are much good at studying. He says he is willing to pay you \$10 if your android gets an A in the class. Recharging the battery costs \$5 Suppose that you could program your android to study or not to study at will (this is not very ethical, but it is technically feasible). What is the best course of action for you?

Answer: If you program the android to study, it will have probability $P(A = 1|S = 1)$ to get an A and probability $P(B = 1|S = 1)$ to get its battery depleted. Your total utility is: $10 * (A = 1|S = 1) + (-5) * P(B = 1|S = 1) = 10 * 0.8 - 5 * 0.75 = 4.25$. If the android does not study, the utility is $10 * P(A = 1|S = 0) + (-5) * P(B = 1|S = 0) = 10 * 0.5 - 5 * 0.1 = 4.5$. So the optimal course of action (which maximizes utility) is to program the android to not study.

4. **Logic**

(a) Translate the following sentences in first-order logic:

i. All citizens of Fredonia speak the same language.

ii. The Fredonese language has two dialects

iii. Each citizen of Fredonia speaks exactly one of the two dialects

(b) Translate the knowledge base above into conjunctive normal form (using Skolem constants and functions as appropriate)

Answer: The first statement translates to:

$$\forall x Citizen(x, Fredonia) \rightarrow Speaks(x, Fredonese)$$

The second statement translates to:

$$\exists x \exists y Dialect(x, Fredonese) \wedge Dialect(y, Fredonese)$$

which can be skolemized as:

$$Dialect(X0, Fredonese) \land Dialect(X1, Fredonese)$$

For the third statement, using the skolemization above, we have:

$$\forall x Citizen(x, Fredonia) \rightarrow ((Speaks(x, X0) \land \neg Speaks(x, X1)) \lor (\neg Speaks(x, X0) \land Speaks(x, X1)))$$

5. **Naive Bayes**

Data the android is about to play in a concert on the Enterprise and he wants to use a naive Bayes classifier to predict whether he will impress Captain Picard. He believes that the outcome depends on whether Picard has been reading Shakespeare or not for the three days before the concert. For the previous five concerts, Data has observed Picard and noted on which days he read Shakespeare. His observations look like this:

D1	D2	D3	LC
1	1	0	yes
0	0	1	no
1	1	1	yes
1	0	1	no
0	0	0	no

(a) Show the Naive Bayes model that Data obtains, using maximum likelihood, from these instances.

Answer: LC is the top node, with arrows going to D1, D2 and D3. From the data, the max. likelihood probabilities are obtained by counting:

$$\begin{aligned}
P(LC = 1) &= 2/5 \\
P(D1 = 1 LC = 1) &= 1 \\
P(D1 = 1 LC = 0) &= 1/3 \\
P(D2 = 1 LC = 1) &= 1 \\
P(D2 = 1 LC = 0) &= 0 \\
P(D3 = 1 LC = 1) &= 1/2 \\
P(D3 = 1 LC = 0) &= 2/3
\end{aligned}$$

(b) If Picard reads Shakespeare only on day 1, how likely is he to enjoy Data's concert? What if he reads Shakespeare on days 1 and 3?

Answer: We need to compute $P(LC = 1|D = 1)$:

$$\begin{aligned}
P(LC = 1|D1 = 1) &= \frac{P(LC = 1, D1 = 1)}{P(D1 = 1)} \\
&= \frac{P(LC = 1)P(D1 = 1|LC = 1)}{P(LC = 1)P(D1 = 1|LC = 1) + P(LC = 0)P(D1 = 1|LC = 0)} \\
&= \frac{2/5 * 1}{2/5 * 1 + 3/5 * 1/3} = \frac{2}{3}
\end{aligned}$$

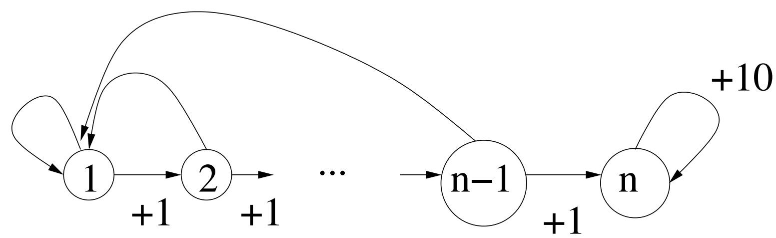
For the second part, we need $P(LC=1|D1=1,D3=1)$, which we can similarly compute as:

$$\begin{aligned} P(LC=1|D1=1,D3=1) &= \frac{P(LC=1,D1=1,D3=1)}{P(D1=1,D3=1)} \\ &= \frac{P(LC=1)P(D1=1|LC=1)P(D3=1|LC=1)}{P(D1=1,D3=1,LC=1)+P(D1=1,D3=1,LC=0)} \end{aligned}$$

You can finalize the calculation similarly to above

6. **Markov Decision Processes**

Consider the n-state MDP in the figure below. In state n there is just one action that collects a reward of $+10$, and terminates the episode. In all the other states there are two actions: float, which moves deterministically one step to the right, and reset, which deterministically goes back to state 1. There is a reward of $+1$ for a float and 0 for reset. The discount factor is $\gamma = \frac{1}{2}$.



(a) What is the optimal policy?

Answer: Always go right

(b) What is the optimal value of state n, $V^*(n)$?

Answer: $10 + 10 * \frac{1}{2} + 10 * \left(\frac{1}{2}\right)^2 + \ldots = 10 * \frac{1}{1-1/2} = 20$

(c) Compute the optimal value function, $V^*(k)$ for all $k = 1, \ldots n-1$.

Answer:

$$V^*(n-1) = 1 + \frac{1}{2}V^*(n)$$

$$V^*(n-2) = 1 + \frac{1}{2}V^*(n-1) = 1 * \left(1 + \frac{1}{2}\right) + \left(\frac{1}{2}\right)^2 V^*(n) = \frac{1-(1/2)^2}{1-1/2} + \left(\frac{1}{2}\right)^2 V^*(n)$$

$$\ldots$$

$$V^*(n-k) = 1 + \frac{1}{2}V^*(n-k+1) = 1 + \frac{1}{2} + \ldots \left(\frac{1}{2}\right)^{k-1} + \left(\frac{1}{2}\right)^k V^*(n) = \frac{1-(1/2)^k}{1-1/2} + \left(\frac{1}{2}\right)^k V^*(n)$$

(d) Suppose you are doing value iteration to figure out these values. You start with all value estimates equal to 0. Show all the non-zero values after 1 and 2 iterations respectively.

Answer: After one iteration, $V_1(n) = 10$ and $V_1(n-k) = 1 \forall k > 0$. After two iterations:

$$\begin{aligned} V_2(n) &= 10 + 10 * 1/2 = 15 \\ V_2(n-1) &= 1 + 1/2 * V_1(n) = 6 \\ V_2(n-k) &= 1 + 1/2 * V_1(n-k+1) = 1 + 1/2 = 3/2 \forall k > 1 \end{aligned}$$

(e) Suppose that instead of knowing the model of the MDP, you decided to always float and observed the following trajectories:

- n-1, float, +1, n, float, +10, n.
- n-2, float, +1, n-1, float, +1, n, float, +10, n.

What would be the Monte Carlo estimate for all states, based on this data? What would be the certainty equivalence estimate?

Answer: The Monte Carlo estimate is $V(n) = 10$, $V(n-1) = 6$,$V(n-2) = 4.5$. The certainty equivalent estimate will fit the model based on these transitions: $P(s_{t+1} = n|s_t = n) = 1$, $P(s_{t+1} = n|s_t = n-1) = 1$, $P(s_{t+1} = n-1|s_t = n-2) = 1$. So it will correctly estimate the values for all these states, as is item (c).

7. **Applying AI methods**

You have been hired by a large retail company who is having trouble managing its inventory. The company provides 100 kinds of products, and for each product there are two or three suppliers. For each supplier, the company keeps track of their price, of whether they deliver on time or if not, how late they are on each order, and on how many broken products they get in each order. The company also knows at what prices it is able to sell each product, and how fast they can sell it. The company has past data about all the orders it placed or got in the last 5 years. The company also has limited space to store its stock of products. You are required to develop a software package which would decide as well as possible what kinds of products to keep in stock, and what orders to place in order to get them.

Choose **one AI technique** to solve this task. Explain in detail how it would be implemented, what are the pros and the cons.

Answer: There are several valid ways of solving this problem, but perhaps the most straightforward is to use a Markov Decision Process in which states consist of the content of the warehouse (what products are there and how many), the actions consist of what to order, the rewards are obtained as the amount of money received for selling products minus the amount paid for orders, and the model of the MDP captures the information above about the prices, transitions etc.

8. **Problem formulation**

You have been hired by a scientific computing center to help them manage the requests they get. Presently, m scientists have each submitted a list of n computations that they would like to complete. The center has 3 machines: a parallel supercomputer, a quantum computer and a cluster of regular computers. Each scientist submits, with each job, a list of the computers that would be suitable to execute it. There are $m \leq k < mn$ days on which to perform computations before the center closes for Christmas. To be fair, at least one computation from each scientist's list has to be performed. Each computation takes exactly one day. Each computer can only do one computation on any given day.

(a) Describe this as a constraint satisfaction problem. Clearly specify what are the variables, the domains of the variables and the constraints.

Answer: Let $C_{ij}, i = 1, \ldots m, j = 1 \ldots n$ be the list of computations, where the index i indicates the scientist who submitted the computation and j is the number of the computation. These are the variables of the problem. The domain of each variable is $(\{0\} \cup D_{ij})$, where 0 indicates this computation is not scheduled, D is the list of computers on which the computation can be done ($D_{ij} \subseteq \{1,2,3\}$). Let T_{ij} be a second set of variables with domain $T = \{1,2,\ldots k\}$ - the set of days available. If $C_{ij} \neq -1$ then T_{ij} is the day scheduled for the computation.

The fairness constraint can be expressed as: $\sum_{j=1}^{n} C_{ij} > 0 \forall i$. The timing constraint is more complicated: if $T_{ij} = T_{i'j'} = T_{i''j''}$ then $C_{ij} \neq C_{i'j'} \wedge C_{i''j''} \neq C_{i'j'} \wedge C_{ij} \neq C_{i''j''}$

(b) Your manager also wants to minimize the total time in which the machines are idle (keeping all the same constraints as above). What kind of problem do you have? Specify one suitable solution method and motivate your choice in one sentence.

Answer: This is now an optimization problem, which could be solved, e.g. by simulated annealing or genetic algorithms (see lecture slides for reasons).

9. **Logic**

(a) Translate the following sentences in first-order logic.

i. Star Trek, Star Wars and The Matrix are science fiction movies.
Answer: $SciFi(StarTrek) \wedge SciFi(StartWars) \wedge SciFi(Matrix)$

ii. Every AI student loves Star Trek or Star Wars.
Answer: $\forall x AIStudent(x) \rightarrow Loves(x, StarTrek) \vee Loves(x, StartWars)$

iii. Some AI students do not love Star Trek.
Answer: $\exists x AIStudent(x) \wedge \neg Loves(x, StarTrek)$

iv. All AI students who love Star Trek also love The Matrix.
Answer: $\forall x AIStudent(x) \wedge Loves(x, StarTrek) \rightarrow Loves(x, Matrix)$

v. Every AI student loves some science fiction movie.
Answer: $\forall x AIStudent(x) \rightarrow (\exists y SciFi(y) \wedge Loves(x,y)$

vi. No science fiction movie is loved by all AI students.
Answer: $\neg(\exists y SciFi(y) \wedge (\forall x AIStudent(x) \rightarrow Loves(x,y)))$

vii. There is an AI student who loves all science fiction movies.
Answer: $\exists x AIStudent(x) \wedge (\forall y SciFi(y) \rightarrow Loves(x,y))$

(b) Based on the knowledge base above, prove formally that there exists some AI student who loves Star Wars.

Answer: We can re-write the first statement as:

$$\neg AIStudent(x) \vee Loves(x, StarTrek) \vee Loves(x, StartWars)$$

The second statement can be re-written through skolemization as:

$$AIStudent(X0) \wedge \neg Loves(X0, StarTrek)$$

By unification and resolution between these two statements, we get:

$$Loves(X0, StartWars)$$

which proves the conclusion