Probabilistic Foundations of Artificial Intelligence
Final Exam

Date:	28 January 2014
Time limit:	120 minutes
Number of pages:	16

You can use the back of the pages if you run out of space. Collaboration on the exam is strictly forbidden.

[1 point] Please fill in your name and student ID:

1 [10 points] Short Questions

For each of the questions below, decide whether they are true or false, and briefly justify your answer (i.e., 1-2 sentences should suffice).

(a) **[2 points]** Breadth-first search is complete if the state space has infinite depth but finite branching factor.

☐ True ☐ False

(b) **[2 points]** Suppose you are using the A^* algorithm to solve a search problem (e.g., an agent wants to get from state A to state B as quickly as possible). If h_1 and h_2 are two non-negative admissible heuristics for this problem, then $\sqrt{h_1 h_2}$ is also guaranteed to be admissible.

☐ True ☐ False

(c) **[2 points]** Assume a king can move one square in any direction on a chess board (8 directions in all). Manhattan distance (also known as L_1 distance) is then an admissible heuristic for the problem of moving the king from square A to square B.

☐ True ☐ False

(d) **[2 points]** Every search problem with a finite number of states can be expressed as an MDP with at most as many states as the original search problem.

☐ True ☐ False

(e) **[2 points]** Assume an MDP has a finite number of states. Value iteration is guaranteed to converge if the discount factor γ satisfies $0 < \gamma < 1$.

☐ True ☐ False

2 [15 points] Propositional and First-order Logic

James, Henry and David are working in a company. We know that they hold the jobs of manager, programmer and engineer; but we don't know which person has which job.

a) **[5 points]** Generate a first-order logic knowledge base considering the information and predicates below. Convert each sentence from KB to clausal form (you may assume database semantics).

Information of the company:

- James has borrowed money from the programmer.
- Manager is married.
- Manager doesn't like to borrow money from somebody else.
- David is single.

Predicates:

- Borrowed(X, Y) in which X is the name of a person and Y is a job. This predicate is true iff person X borrowed from the person holding job Y.
- Married(X) in which X is the name of a person, indicating that X is married.
- Job(X,Y) in which X is the name of a person and Y is his job, indicating that X holds job Y.

Moreover, for sake of brevity you can refer to the set of names and jobs as follows:

$$\text{Names} = \{J, H, D\}, \quad \text{Jobs} = \{P, M, E\}.$$

b) **[5 points]** We now prove statements about people's jobs.

[*Hint: Remember resolution in first order logic can be performed by propositionalization followed by propositional resolution. To simplify notation, only propositionalize the rules as necessary.*

As an example, consider the following KB:

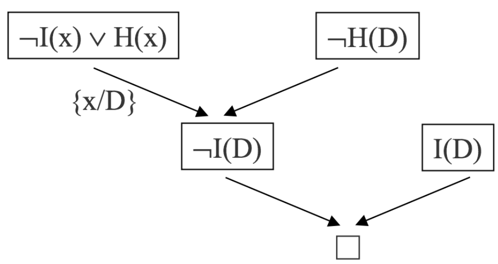
- *If something is intelligent, it has common sense*
- *Deep Blue does not have common sense*

We prove that Deep Blue is not intelligent.

We first convert the information in the KB to sentences in first order logic. Here predicate $I(x)$ means that x is intelligent and $H(x)$ means that x has common sense.

$$\forall x, I(x) \Rightarrow H(x) \quad \textit{and} \quad \neg H(D)$$

Then we convert the sentences to CNF and use resolution to prove that Deep Blue is not intelligent.]



Prove using resolution that David is not the manager.

c) **[5 points]** Now prove that James is not the manager.

3 [15 points] Whodunnit?

X finds his garden destroyed one day and immediately blames Y, who he believes had once before tried to ruin his orchid collection. Assume that if Y has a motive, he would have made an earlier attempt with probability 0.8. If Y does not have a motive, the probability that he made an earlier attempt is 0.05. That is,

$$P(\text{Historical Attempt} = \texttt{T} \mid \text{Motive} = \texttt{T}) = 0.8, \text{ and}$$
$$P(\text{Historical Attempt} = \texttt{T} \mid \text{Motive} = \texttt{F}) = 0.05$$

Evidence in this case is in the form of forensic analysis that confirms whether Y was in X's garden on the fateful day. Your task in this question to reason about Y's chance of conviction. You are also given a table from the hypothetical book "Solving Garden Crimes" (see Table 1) according to which both motive and evidence have to be proven in court to obtain a conviction. Also assume that Y has a 0.2 prior probability of having a motive to damage the garden and 0.8 prior probability of being around X's garden on any given day:

$$P(\text{Motive} = \texttt{T}) = 0.2, \text{ and } P(\text{Evidence} = \texttt{T}) = 0.8$$

(a) **[5 points]** Using the binary random variables E(**E**vidence), M(**M**otive), H(**H**istorical Attempt) and C(**C**onviction), draw the Bayesian network corresponding to the problem described above. You do not need to specify the conditional distribution tables in the Bayesian network.

Table 1: Solving Garden Crimes: Probability of Conviction

Evidence	Motive	P(Conviction)
T	T	0.9
T	F	0.2
F	T	0.2
F	F	0.1

(b) **[5 points]** Using the information given, compute the probability that Y is convicted given that forensic analysis confirms Y's presence in the garden (i.e., $P(C = \texttt{T} \mid E = \texttt{T})$)

(c) **[5 points]** Given that Y was not around the garden on that day, and further given the fact that Y is not convicted, what is the probability that Y has the motive to destroy the garden (i.e., $P(M = \texttt{T} \mid C = \texttt{F}, E = \texttt{F})$)?

4 [15 points] Bayesian Networks

Consider the given figure representing a Bayesian network (see Figure 1). Answer the following questions with true or false, and briefly justify your answer.

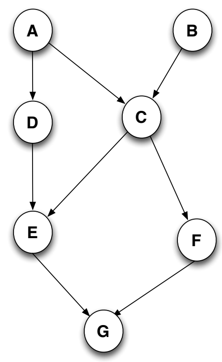


Figure 1: Bayesian network

(a) **[2 points]** F is conditionally independent of D given no information ($F \perp D$)

☐ True ☐ False

(b) **[2 points]** F is conditionally independent of D given B ($F \perp D \mid B$)

☐ True ☐ False

(c) **[2 points]** F is conditionally independent of D given A ($F \perp D \mid A$)

☐ True ☐ False

(d) **[4 points]** Consider the variable ordering A,G,C,F,E,B,D. For each iteration of the variable elimination algorithm, determine which factors are removed and introduced. What is the size of the largest factor that results from the given order?

(e) **[5 points]** Suggest a better ordering such that the size of the largest resulting factor is less than that for the ordering in (d). What is the size of the largest factor under this ordering?

5 [12 points] Temporal Models: Exploratory Rover

An exploratory rover is navigating through planet Vulcan, a deserted planet scattered with volcanic vents and radioactive valleys. The rover is equipped with a thermometer that registers only two levels, *hot* and *cold*. The rover sends back thermal responses $E =$ `hot` when it is at a **V**olcanic vent (V), or a **R**adioactive valley (R), and $E =$ `cold` when it is at **N**ormal area (N). There is no chance of a mistaken reading.

The rover can only stay in one area on any given day. It travels around according to the following transition probabilities:

	$P(X_t \mid X_{t-1} = N)$	$P(X_t \mid X_{t-1} = V)$	$P(X_t \mid X_{t-1} = R)$
$X_t = N$	0.7	0.6	0.2
$X_t = V$	0.2	0.3	0.2
$X_t = R$	0.1	0.1	0.6

(a) **[2 points]** Imagine that you observe the sequence $\{$`cold`, `hot`, `hot`$\}$. What is the probability that the rover were at normal area all three days?

(b) **[5 points]** What is the most likely sequence for the observation sequence $\{$`cold`, `hot`, `hot`, `cold`$\}$? *[Hint: you should not have to do extensive calculations.]*

Continue to next page for more space

(c) **[3 points]** Imagine you are monitoring the rover's state using the particle filtering algorithm, and on a given day you have 5 particles on N, 3 on V, and 2 on R, before making an observation on that day. If the rover sends back "`hot`", what weight will each of your particles have?

(d) **[2 points]** After resampling, what are the expected numbers of particles you will have on N, V and R ?

6 [22 points] MDPs and RL: The Evasion Plan

Pacman () is chased by a ghost () in a maze shown in Figure 3. There are four valid positions that they can stay, annotated with coordinates (0,0), (0,1), (1,0), (1,1). Initially, the ghost is in square (0,0), and Pacman is in square (1,1). At each step, Pacman can choose from two actions:

1. *Move* horizontally or vertically to his neighboring gray square, or
2. *Cross* the river in the middle, and end up in the diagonal position;

whereas the ghost can only move horizontally or vertically, and *cannot* cross the river. Whoever performs the action "move" always succeeds. When Pacman chooses to *cross* the river, however, he only succeeds with probability 0.5 of ending up on the diagonal square, and gets stuck in the original square otherwise. Both Pacman and the ghost act simultaneously.

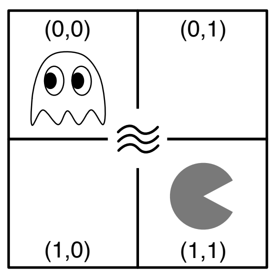


Figure 2: Pacman chased by a ghost in a maze

Distance between squares is measured by the Hamming distance of their coordinates, e.g., $d((0,0),(0,1)) = 1$ and $d((0,1),(1,0)) = 2$. The ghost always moves toward the direction that is closer to Pacman's current position. If moving horizontally and moving vertically have equal distance to Pacman, the ghost will choose either direction equally likely to move forward. When taking an action, Pacman gains a reward of $+4$ if he ends up 2 squares away from the ghost, a reward of $+1$ if he ends up 1 square away from the ghost, and a reward of -10 if he ends up getting caught. Once Pacman is caught by the ghost, the game stops, and no action will be taken (and hence Pacman receives 0 reward afterwards). The discount factor is $\gamma = 0.5$.

(a) **[7 points]** Draw an MDP for the problem described above, annotating the action-dependent transitions with transition probabilities and associated rewards. *[Hint: One way to model the above problem as an MDP is to create one state for each pair of Pacman-Ghost locations, which amounts to a total of* $2^4 = 16$ *states. However, you can represent the problem with far fewer states, by focusing on the* relative distance *between Pacman and the ghost.]*

(b) **[10 points]** In order to compute the optimal policy for the given MDP, we consider the policy iteration approach here.

Recall that policy iteration starts with an arbitrary initial policy π. Until convergence, it iteratively computes the value function $V_\pi(x)$ for the current policy and then updates the current policy to be the greedy policy π_g w.r.t. the computed $V_\pi(x)$. The greedy policy for a value function is given by

$$\pi_g = \arg\max_a \; r(x, a) + \gamma \sum_{x'} P(x'|x, \pi(x)) V_\pi(x'),$$

and the value function $V_\pi(x)$ is given by

$$V_\pi(x) = \sum_{x'} P(x'|x, \pi(x)) \left[r(x, \pi(x), x') + \gamma V_\pi(x') \right].$$

Compute the optimal policy and its value function for the above MDP. *[Hint: To save time, start with an initial guess for the optimal policy and prove that it's greedy w.r.t. the corresponding value function, i.e., policy iteration terminates.]*

(c) **[5 points]** Imagine that the transition probabilities in your MDP are unknown. To learn them, you choose to run Q-learning. Assume the learning rate is $\alpha = 0.5$, and all Q-values are initialized to 0. The discount factor is $\gamma = 0.5$. Write down the Q-values that result from the Q-learning update for the 3 transitions specified by the episode below (see Figure 3). *[Hint: you only need to write down the Q-values that have been updated (i.e., the non-zero values).]*

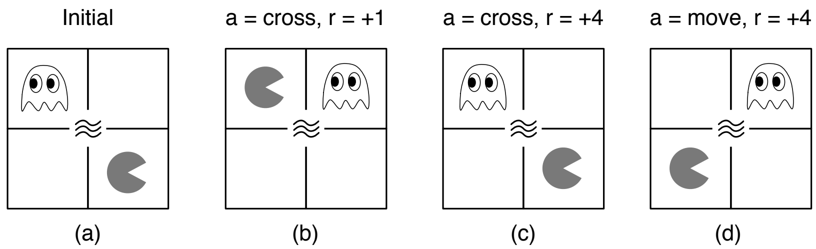


Figure 3: Pacman and the ghost within 3 transitions. Transition from (a) to (b): Pacman crosses and receives reward +1; from (b) to (c): Pacman crosses and receives reward +4; from (b) to (c): Pacman moves and gets reward +4.

7 [10 points] Bayesian Network Structure Learning

Suppose you would like to learn a Bayesian network on *binary* variables $X_1, \ldots, X_n$. You are given N complete observations, i.e., you observe a data set $D = \{(x_1^{(1)}, \ldots, x_n^{(1)}), \ldots, (x_1^{(N)}, \ldots, x_n^{(N)})\}$. For any given $\lambda \geq 0$, consider the following score function of a graph structure G:

$$S(G; D) = \sum_{i=1}^{n} \left\{ \hat{I}_D(X_i, \mathbf{Pa}_i) \right\} - \lambda \cdot \#params(G),$$

where $\hat{I}_D(X_i, \mathbf{Pa}_i)$ is the empirical mutual information between variable X_i and its parents $\mathbf{Pa}_i$ with respect to data set D, and $\#params(G)$ is the total number of parameters (in all CPTs combined) of the Bayesian network.

Suppose you know the causal ordering of the variables. Thus, you know for sure that the parents $\mathbf{Pa}_i$ of variable X_i can only be a subset of $\{X_1, \ldots, X_{i-1}\}$. Develop an efficient algorithm (whose running time can be exponential in k) that learns the optimal Bayesian network structure (w.r.t. $S(G; D)$) where each variable can have *at most k parents*. Write down the pseudocode for your algorithm, and explain your answer.