CSE 573 Exam Solutions – November 18, 2010

Name:

Scores					
Q.1 (14)	**Q.2 (15)**	**Q.3 (27)**	**Q.4 (21)**	**Q.4 (28)**	**Total (105)**

You have 80 minutes to answer 4 questions. Questions are weighted differently, and their point values are specified next to them. Each question has easier and harder parts, so try to answer at least the easier parts of all questions.

If you show your work and *briefly* describe your approach to the longer questions, I will happily give partially credit, where possible.

There are 12 pages in this exam. Please write your answers in the space provided. The last page is a tear-off with the figures for questions 3 and 4, for your convenience. You don't need to hand it in.

[This page was intentionally left blank]

[This page was intentionally left blank]

Question 1 – True/False – 14 points

Circle the correct answer each True / False question.

1. True / False – All consistent heuristics are admissible. (2 pt)
   Answer: True. Definition from class, given without proof.

2. True / False – A* Graph Search with an admissible heuristic always returns the optimal solution. (2 pt)
   Answer: False. The heuristic must be consistent.

3. True / False – Optimal Alpha-Beta pruning, on average, reduces the size of the search tree by an exponential factor. (2 pt)
   Answer: False. The change is from $O(b^m)$ to $O(b^{m/2})$. Note: This question was poorly worded and I gave everyone credit.

4. True / False – Value Iteration always find the optimal policy, when run to convergence. (2 pt)
   Answer: True. This is a result of the Bellman backups being a contraction, as discussed.

5. True / False – Policy Iteration has been empirically observed to converge more slowly than Value Iteration. (2 pt)
   Answer: False. Although there is no theoretical guarantee, if often converges faster, as we discussed in lecture.

6. True / False – Q-learning with linear function approximation (features) will always converge to the optimal policy. (2 pt)
   Answer: False. It may not even be able to represent the optimal policy.

7. True / False – The number of parameters in a Bayesian network is exponential in the total number of arcs in the graph. (2 pt)
   Answer: False. It is exponential in the number of parent for the node with the most parents.

Question 2 – Short Answer – 15 points

These short answer questions can be answered with one or two sentences.

1. Short Answer – We say that a search heuristic h_1 *dominates* (is not worse than) another heuristic h_2 if a certain property holds. Give the property and describe the effect it will have on the running time of A* when using h_1 vs. h_2. (3 pts)

   Answer: h_1 dominates h_2 if, for all s, $h_1(s) \geq h_2(s)$. The number of nodes expanded when using h_1 will not be larger than with h_2.

2. Short Answer – For HMM filtering, when would you prefer to use exact inference (the forward algorithm) and when would we prefer to use the particle filtering algorithm? Hint: Think about the computational complexity of each algorithm. (3 pts)

   Answer: Exact inference is preferred if the number of states is small enough.

3. Short Answer – For Q-learning to converge we need to correctly manage the exploration vs. exploitation tradeoff. What property needs to be hold for the exploration strategy? (3 pts)

   Answer: In the limit, every action needs to be tried sufficiently often in every possible state. This can be guaranteed with an sufficiently permissive exploration strategy.

4. Short Answer – Which search algorithm (BFS, DFS, UCS, A*, ID, etc.) best managed the worst-case asymptotic time and space complexity tradeoff? How did it achieve this result? (3 pts)

   Answer: Iterative deepening is the best. It saves space, like depth first search, by not requiring the list to represent the frontier. It also never builds paths that are longer than the length of the shortest solution.

5. Short Answer – Describe the differences between an HMM and a more general Bayesian Network? (3 pts)

   Answer: An HMM is a time series model, where each random variable has at most one parent. It has a specific structure, determined by the parameterization according the prior, transition, and observation distributions. An arbitrary BN can be any acyclic graph, along with the associated CPTs.

Question 3 – Tree Search – 27 points

Look at figure 2 on the last page (tear the page off to have the figures in front of you as you answer). The state transition graph defines a problem that was solved with four different tree search algorithms.

Assume, as we did in class, that: (1) The numbers next to nodes correspond to the priority that the algorithm assigned to them in the expansion queue (if any); (2) Ties in expansion order are broken according to alphabetical order; and, (3) The goal node that was found is highlighted.

1. For each of the search trees in the figure 2 (5 points per tree), say which algorithm was used, out of this list: Depth-Fist Search, Breadth-First Search, Iterative Deepening DFS, Uniform Cost Search, Best-First (greedy) Search, A*.

Specify also whether a heuristic was used, and if so, whether it was heuristic H1 or H2 (defined next to the graph on the figure), and whether the search found the optimal path to the goal. Write your answers in the space provided.

search tree 1: (5pts)
Algorithm? BFS
Heuristic function, if any? None
Optimal path? Why or why not?

No. BFS is only guaranteed to find the shortest path.

search tree 2: (5pts)
Algorithm? A*
Heuristic function, if any? H2
Optimal path? Why or why not?

Yes. Because H2 is admissible.

search tree 3: (5pts)
Algorithm? UCS
Heuristic function, if any? None
Optimal path? Why or why not?

Yes. UCS is always optimal.

search tree 4: (5pts)
Algorithm? DFS
Heuristic function, if any? None
Optimal path? Why or why not?

No. DFS will often find a path that is very long.

2. Are heuristics H1 and H2 consistent? Are they admissible? (4 points)

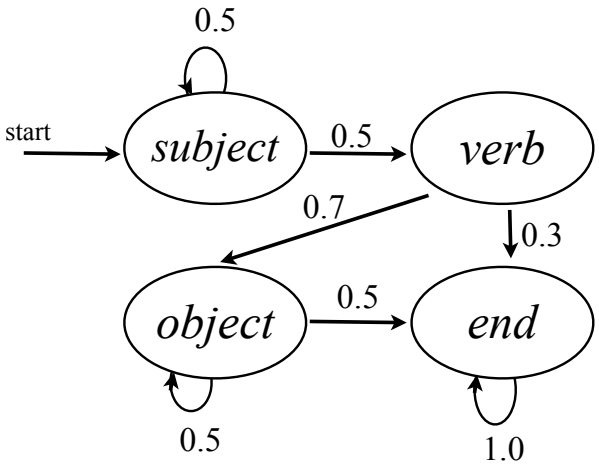
Answer: Both are admissible but only H2 is consistent.

3. For tree search A*, is one of H1 or H2 guaranteed to perform better than the other? If so, identify the better heuristic and explain why. (3 points)

Answer: Both will return the optimal solution, but H2 dominates H1 and will not expand more nodes of the search tree.

Question 4 – Temporal Reasoning with Uncertainty – 21 points

Consider this very simple model of the grammatical structure of an English sentence. A sentence always starts with a subject phrase, which can be made of one or several words (including punctuation), and at some point transitions to a verb. The sentence can stop there (with probability .3) but is more likely to transition to an object phrase, which has the same possible composition as the subject phrase. The figure below gives a compact HMM representation of this model, with all transition and output conditional probabilities specified. (This figure is reproduced on the tear-off sheet for your convenience.)



output word probabilities:	*subject*	*verb*	*object*	*end*
boat	.2	0	.2	0
man	.2	.2	.2	0
old	.2	0	.2	0
rows	.1	.8	.1	0
the	.3	0	.3	0
.	0	0	0	1

The output probabilities take into account that some words (like *man* and *rows*) can be both nouns (and so part of the subject and object phrases) and verbs. We observe the following beginning of a sentence "The man...".

1. What is the hidden variable and the observed variable (evidence) in this HMM? What are their respective domains? (5 points)

   The hidden variable is the word type, which can take values: subject, verb, object, and end. The observed variable is the word, which ranges over "boat," "man," "old," "rows," "the" and ".".

2. What type of word (subject, verb, object, or end) is most likely to appear next, given that we've observed "The man...". What the probability of this word type, conditioned on the evidence so far. (8 points)

   To solve this problem, run the forward algorithm. First, compute the distribution given the initial evidence: $P(X_1|e_1) \propto P(e_1|X_1)P(X_1)$ Since we start in the subject state with probability one, the observation does not change anything:

X=	subj.	verb	obj.	end
$P(X_1)$	1.0	0.0	0.0	0.0
$P(X_1\|E_1 =$"The")	1.0	0.0	0.0	0.0

   Now, compute one full filtering step, by
   (1) updating for time (transition): $P(X_t|e_{1:t-1}) = \sum_{x_{t-1}} P(X_t|x_{t-1})P(x_{t-1}|e_{1:t-1})$, and
   (2) updating for evidence (observe): $P(X_t|e_{1:t}) \propto P(X_t|e_{1:t-1})P(e_t|X_t)$

   The transition can take you to the subject or verb states (prob. 0.5 for each), which are equally likely to produce the word "man." So, the result is uniform over these two states, after normalizing.

X=	subj.	verb	obj.	end
$P(X_2\|Y_1 =$"The")	0.5	0.5	0.0	0.0
$P(X_2\|E_1 =$"The", $E_2 =$"man")	0.5	0.5	0.0	0.0

   Finally, one more transition, but no evidence to incorporate this time:

X=	subj.	verb	obj.	end
$P(X_3\|E_1 =$"The", $E_2 =$"man")	0.25	0.25	0.35	0.15

   The most likely next word type is object, with probability 0.35.

3. The next word we observe is "rows." What is the distribution over the type of the word "rows," given our sentence so far "The man rows."? (8 points)

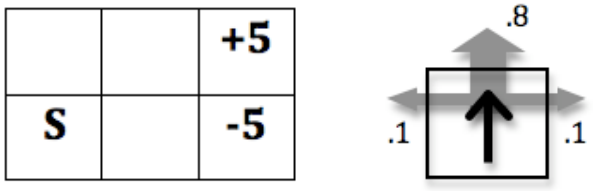
   Here, we just need to compute $P(X_3|Y_1 =$"The", $Y_2 =$"man", $Y_3 =$"rows"). We continue the forward computation, given the calculations above, by multiplying in the observation probabilities and re-normalizing:

X=	subj.	verb	obj.	end
$P(X_3\|E_1 =$"The", $E_2 =$"man") =	0.25	0.25	0.35	0.15
$P(X_3\|E_1 =$"The", $E_2 =$"man", $E_3 =$"rows") $\propto$	0.25(0.1)	0.25(0.8)	0.35(0.1)	0.15(0.0)
$P(X_3\|E_1 =$"The", $E_2 =$"man", $E_3 =$"rows") =	0.096	.769	0.135	0.0

   Given how likely the verb state is to generate the words "rows," we have changed out beliefs after incorporating evidence.

Question 5 – MDPs and Reinforcement Learning – 28 points

This gridworld MDP operates like to the one we saw in class. The states are grid squares, identified by their row and column number (row first). The agent always starts in state (1,1), marked with the letter S. There are two terminal goal states, (2,3) with reward +5 and (1,3) with reward -5. Rewards are 0 in non-terminal states. (The reward for a state is received as the agent moves into the state.) The transition function is such that the intended agent movement (North, South, West, or East) happens with probability .8. With probability .1 each, the agent ends up in one of the states perpendicular to the intended direction. If a collision with a wall happens, the agent stays in the same state.



(a) (b)

Figure 1: (a) Gridworld MDP. (b) Transition function.

1. Draw the optimal policy for this grid? (5 points)

S =	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)
$\pi^*(S) =$	Up	Left	NA	Right	Right	NA

2. Suppose the agent knows the transition probabilities. Give the first two rounds of value iteration updates for each state, with a discount of 0.9. (Assume V_0 is 0 everywhere and compute V_i for times $i = 1, 2$). (8 points)

   Apply the Bellman backups $V_{i+1}(s) = \max_a(\sum_{s'} T(s, a, s')(R(s, a, s') + \gamma V_i(s')))$ twice. I will show the computations for the max actions. Most of the terms will be zero, which are omitted here for compactness.

S =	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)
$V_0(S) =$	0	0	0	0	0	0
$V_1(S) =$	0	0	0	0	$0.8 \times 5.0 = 4.0$	0
$V_2(S) =$	0	$0.9 \times 0.8 \times 4$ $+0.1 \times -5 = 2.38$	0	$0.8 \times 0.9 \times 4.0 = 2.88$	$0.8 \times 5.0 = 4.0$	0

3. Suppose the agent does not know the transition probabilities. What does it need to be able do (or have available) in order to learn the optimal policy? (5 points)

   The agent must be able to explore the world by taking actions and observing the effects.

4. The agent starts with the policy that always chooses to go right, and executes the following three trials: 1) (1,1)–(1,2)–(1,3), 2) (1,1)–(1,2)–(2,2)–(2,3), and 3) (1,1)–(2,1)–(2,2)–(2,3). What are the monte carlo (direct utility) estimates for states (1,1) and (2,2), given these traces? (5 points)

   To compute the estimates, average the rewards received in the trajectories that went through the indicates states.
   $V((1,1)) = (-5+5+5)/3 = 5/3 = 1.666$
   $V((2,2)) = (5+5)/2 = 5$

5. Using a learning rate of .1 and assuming initial values of 0, what updates does the TD-learning agent make after trials 1 and 2, above? (5 points)

   The general TD-learning update is (the other form from lecture is also acceptable):

$$V(s) = V(s) + \alpha(r + \gamma V(s') - V(s))$$

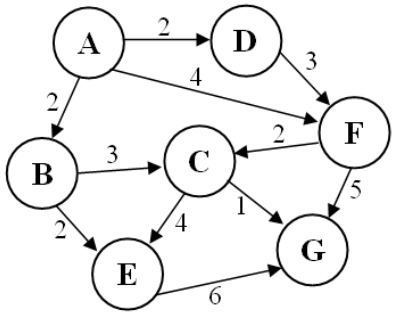
   After trial 1, all of the updates will be zero, expect for:

$$V((1,2)) = 0 + .1\,(-5 + 0.9 \times 0 - 0) = -0.5$$

   After trial 2, the updates will be:

$$\begin{aligned} V((1,1)) &= 0 + .1\,(0 + 0.9 \times -0.5 - 0) = -0.045 \\ V((1,2)) &= -0.5 + .1\,(0 + 0.9 \times 0 + 0.5) = -0.45 \\ V((2,2)) &= 0 + .1\,(5 + 0.9 \times 0 - 0) = 0.5 \end{aligned}$$

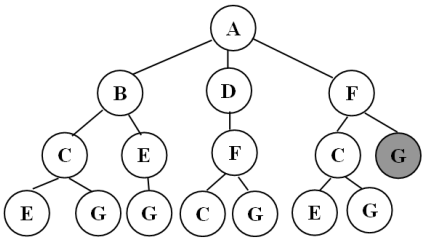
Figure 2: Question 1 – Search: graph, heuristics, and search trees.



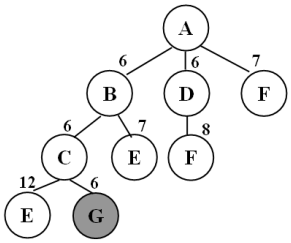
Question 1 graph

	H1	**H2**
A	4	4
B	4	4
C	1	1
D	1	4
E	3	3
F	3	3
G	0	0

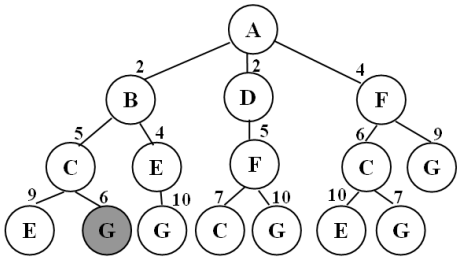
Heuristic functions



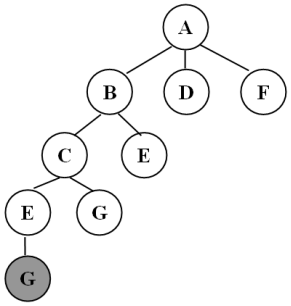
search tree 1



search tree 2

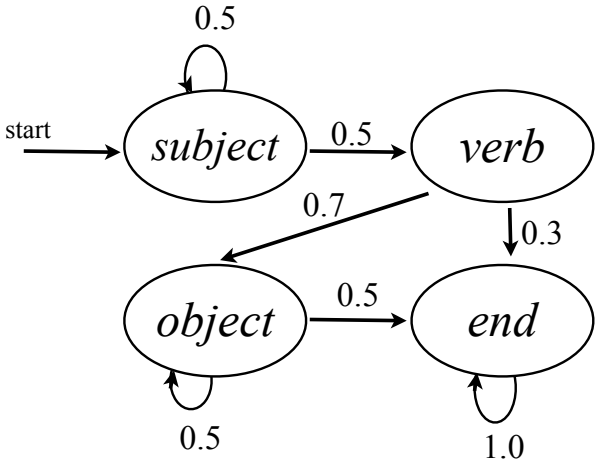


search tree 3



search tree 4

Figure 3: Sentence HMM with conditional probabilities for question 2 (tear-off).



output word probabilities:	*subject*	*verb*	*object*
boat	.2	0	.2
is	0	.5	0
man	.2	.1	.2
old	.2	0	.2
rows	.1	.4	.1
the	.3	0	.3