CS 188 Spring 2006 | Introduction to AI Dan Klein

Midterm Solutions

1. (20 pts.) True/False

Each problem is worth 2 points. Incorrect answers are worth 0 points. Skipped questions are worth 1 point.

(a) *True/False*: The action taken by a rational agent will always be a deterministic function of the agent's current percepts.

False. First, non-deterministic functions are useful in many settings such as multi-agent environments. Second, the agent must often consider its percept history and other sources of information.

(b) *True/False*: An optimal solution path for a search problem with positive costs will never have repeated states.

True. For any solution path with repeated states, removing the cycle will also be a solution path with lower cost.

(c) *True/False*: Depth-first graph search is always complete on problems with finite search graphs.

True. It will reach all reachable states in finite time because graph search does not repeat states.

(d) *True/False*: If two search heuristics $h_1(s)$ and $h_2(s)$ have the same average value, the heuristic $h_3(s) = \max(h_1(s), h_2(s))$ could give better A* efficiency than h_1 or h_2.

True. h_1 could have accurate values for early states and h_2 have accurate values for later states. h_3 could then have more accurate values than either component heuristic.

(e) *True/False*: For robots with only rotational joints, polygonal obstacles will have polygonal images in configuration space.

False. See example in homework 1.

(f) *True/False*: There exist constraint satisfaction problems which can be expressed using trinary (3-variable) constraints, but not binary constraints.

False. Proof that all CSPs have a binary formulation was also in homework 1.

(g) *True/False*: For all random variables W, X, Y and Z

$$\frac{P(Y,Z|W,X)P(W,X)}{P(Y,Z)} = P(W,X|Y,Z)$$

True. This is Bayes' rule if you replace (Y,Z) with A and (W,X) with B.

(h) *True/False*: For any set of attributes, and any training set generated by a deterministic function f of those attributes, there is a decision tree which is consistent with that training set.

True. Decision trees can represent any deterministic function.

(i) *True/False*: There is no hypothesis space which can be PAC learned with one example.

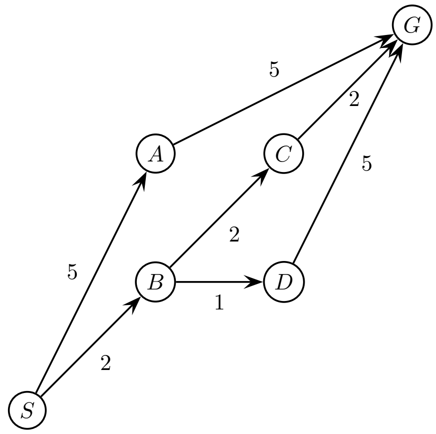
True. A hypothesis space containing only one or two values (e.g., $\{T, F\}$) can be PAC learned from one example.

(j) *True/False*: Smoothing is needed in a Naive Bayes classifier to compute the maximum likelihood estimates of $P(\mathit{feature} \mid \mathit{class})$.

False. Smoothing converts maximum likelihood estimates into smoothed estimates.

2. (24 pts.) Search

Consider the following search problem:



Node	h_0	h_1	h_2
S	0	5	6
A	0	3	5
B	0	4	2
C	0	2	5
D	0	5	3
G	0	0	0

(a) (3 points) Which of the heuristics shown are admissible? h_0 and h_1. $h_2(C) > h^*(C)$, so h_2 is not admissible.

(b) (8 points) Give the path A* search will return using each of the three heuristics? (Hint: you should not actually have to execute A* to know the answer in some cases.)

h_0: $S \to B \to C \to G$

h_1: $S \to B \to C \to G$

h_2: $S \to B \to D \to G$

(c) (3 points) What path will greedy best-first search return using h_1? $S \to A \to G$

Mazeworld: From the MazeWorld domain of Project 1, consider the following maze configuration:

```
 -----------------
|                 |
|                 |
|      S          |
|                 |
|~~~~~~~~~~~~     |
|~~~~~~~~~~~~     |
|~~~~~~~~~~~~ E   |
 -----------------
```

Recall that states are cells of this maze. The agent starts at `S` and tries to reach the goal `E`. Valid actions are moves into adjacent cells that are either clear (step cost of 1) or marked as `~` (step cost of 5).

Below are the outputs of the function `mazeworld.testAgentOnMaze`, tested with several agents that implemented graph search algorithms. Overlaid on the original maze, an `x` is placed on a cell that is used along the solution path, and an `o` denotes a cell that was expanded but not was not part of the solution path. For this problem, DFS and BFS explore successors according to a fixed ordering of moves: right, down, left, up. Greedy and A^* search use the Manhattan distance heuristic.

A different search strategy (BFS, DFS, UCS, Greedy, or A^*) was used to produce each of the outputs below. Specify which strategies were used (2 points each):

```
 -----------------
|ooooooooooooooo  |
|oooooooooooooooo |
|ooooooSoooooooooo|
|ooooooxxxxxxxooo |
|~~ooooooooo~xooo |
|~~~~~~~~~~~~xo   |
|~~~~~~~~~~~~xE   |
 -----------------
```

Strategy: UCS

```
 -----------------
|                 |
|                 |
|      S          |
|      x          |
|~~~~~~x~~~~~     |
|~~~~~~x~~~~~     |
|~~~~~~xxxxxxxE   |
 -----------------
```

Strategy: Greedy

```
 -----------------
|                 |
|                 |
|      Sxxxxxxxxxx|
|                x|
|~~~~~~~~~~~~    x|
|~~~~~~~~~~~~    x|
|~~~~~~~~~~~~ Exxx|
 -----------------
```

Strategy: DFS

```
 -----------------
|                 |
|                 |
|      Soooooo    |
|      xxxxxxxo   |
|~~~~~~~~~~~~xo   |
|~~~~~~~~~~~~xo   |
|~~~~~~~~~~~~xE   |
 -----------------
```

Strategy: A*

```
 -----------------
|oooooooooooooooo |
|ooooooooooooooooo|
|ooooooSoooooooooo|
|ooooooxooooooooo |
|ooooooxoooooooo  |
|ooooooxooooooo   |
|ooooooxxxxxxxE   |
 -----------------
```

Strategy: BFS

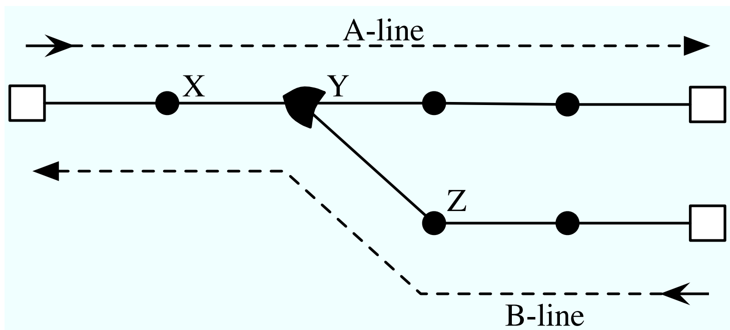
3. (16 pts.) CSPs

Two trains, the A-line and the B-line, use the simple rail network shown below. We must schedule a single daily departure for each train. Each departure will be on an hour, between 1pm and 7pm, inclusive. The trains run at identical, constant speeds, and each segment takes a train one hour to cover.

We can model this problem as a CSP in which the variables represent the departure times of the trains from their source stations:

Variables: A, B
Domains: $\{1, 2, 3, 4, 5, 6, 7\}$

For example, if $A = 1$ and $B = 1$, then both trains leave at 1pm.



The complication is that the trains cannot pass each other in the region of the track that they share, and will collide if improperly scheduled. The *only* points in the shared region where trains trains can pass or touch without a collision are the terminal stations (squares) and the intersection Y.

Example 1: The A-line and B-line both depart at 4pm. At 6pm, the A-line will have reached Y, clearing the shared section of the track. The B train will have only reached Z. No collision will occur.

Example 2: The A-line leaves at 4pm and the B-line at 2pm. At 5pm, the A-line will be at node X and the B-line at the intersection Y. They will then collide around 5:30pm.

Example 3: The A-line leaves at 4pm and the B-line at 3pm. At 5pm, the A train will be at node X and the B train at node Z. At 6pm they will pass each other safely at the intersection Y with *no collision.*

(a) (2 points) If the B-line leaves at 1pm, list the times that the A-line can safely leave without causing a collision.

1, 2, 6, 7

(b) (6 points) Implicitly state the binary constraint between the variables A and B for this CSP. Your statement should be precise, involving variables and inequalities, not a vague assertion that the trains should not collide.

$A < B + 2$ OR $A > B + 4$

(c) (2 points) Imagine the A-line must leave between 4pm and 5pm, inclusive, and the B-line must leave between 1pm and 7pm, inclusive. State the variables' domains after these unary constraints have been imposed.

$$A \in \{4, 5\}$$

$$B \in \{\xcancel{1}, \xcancel{2}, 3, 4, 5, 6, 7\}$$

(d) (6 points) Put an X through any domain value above that is removed by applying arc consistency to these domains.

4. (20 pts.) Probability and Naive Bayes

You've been hired to build a quality control system to decide whether car engines coming off an assembly line are *bad* or *ok*. However, this decision must be based on three noisy boolean observations: the engine may be *wobbly* (motion sensor), *rumbly* (sound sensor), or *hot* (heat sensor). Each sensor gives a Boolean observation: *true* or *false*.

You formalize the problem using the following variables and domains:

Cause: $B \in \{bad,\ ok\}$
Evidence: $W, R, H \in \{true,\ false\}$

You reason that a naive Bayes classifier is appropriate, and build one which predicts $P(B|W, R, H)$.

(a) (4 points) Under the naive Bayes assumption, write an expression for $P(b|w, r, h)$ in terms of *only* probabilities of the forms $P(b)$, $P(w|b)$, $P(r|b)$, and $P(h|b)$.

$$P(b|w,r,h) = \frac{P(w|b)P(r|b)P(h|b)P(b)}{\sum_{b'} P(w|b')P(r|b')P(h|b')P(b')}$$

(b) (3 points) The company has records for several engines which recently came off the assembly line:

B	W	R	H
ok	*false*	*false*	*true*
ok	*false*	*true*	*false*
ok	*false*	*true*	*false*
ok	*true*	*false*	*false*
ok	*false*	*false*	*false*
bad	*true*	*false*	*false*
bad	*true*	*true*	*false*
bad	*false*	*true*	*true*

Given this data, circle EITHER the top row of tables OR the bottom row to indicate which are the correct maximum likelihood parameters for the naive Bayes model of this domain.

The correct estimates ensure that each conditional table sums to 1.

$\hat{P}(B)$

bad	3/8
ok	5/8

$\hat{P}(W \mid B)$

true	*bad*	2/3
false	*bad*	1/3
true	*ok*	1/5
false	*ok*	4/5

$\hat{P}(R \mid B)$

true	*bad*	2/3
false	*bad*	1/3
true	*ok*	2/5
false	*ok*	3/5

$\hat{P}(H \mid B)$

true	*bad*	1/3
false	*bad*	2/3
true	*ok*	1/5
false	*ok*	4/5

Tables repeated for convenience:

$\hat{P}(B)$

bad	3/8
ok	5/8

$\hat{P}(W \mid B)$

true	*bad*	2/8
false	*bad*	1/8
true	*ok*	1/8
false	*ok*	4/8

$\hat{P}(R \mid B)$

true	*bad*	2/8
false	*bad*	1/8
true	*ok*	2/8
false	*ok*	3/8

$\hat{P}(H \mid B)$

true	*bad*	1/8
false	*bad*	2/8
true	*ok*	1/8
false	*ok*	4/8

$\hat{P}(B)$

bad	3/8
ok	5/8

$\hat{P}(W \mid B)$

true	*bad*	2/3
false	*bad*	1/3
true	*ok*	1/5
false	*ok*	4/5

$\hat{P}(R \mid B)$

true	*bad*	2/3
false	*bad*	1/3
true	*ok*	2/5
false	*ok*	3/5

$\hat{P}(H \mid B)$

true	*bad*	1/3
false	*bad*	2/3
true	*ok*	1/5
false	*ok*	4/5

(c) (6 points) Suppose you are given an observation of a new engine: *W=false*, *R=true*, *H=false*. What prediction would your naive Bayes classifier make: *bad* or *ok*?

Let b signify $B = bad$ and $\neg b$ signify $B = ok$.

$P(b|\neg w, r, \neg h) \propto \frac{3}{8} \cdot \frac{1}{3} \cdot \frac{2}{3} \cdot \frac{2}{3} = \frac{1}{18}$
$P(\neg b|\neg w, r, \neg h) \propto \frac{5}{8} \cdot \frac{4}{5} \cdot \frac{2}{5} \cdot \frac{4}{5} = \frac{4}{25}$

(d) (3 points) What would the posterior probability of that prediction be?

Normalizing, we have

$$\frac{4 \cdot 18}{4 \cdot 18 + 25} = \frac{72}{97}$$

(e) (4 points) The company informs you that, according to their utility function, the utility of rejecting an engine is -1, regardless of whether or not it is actually bad. However, the utility of accepting an engine is 2 for ok engines and -20 for bad engines. Given these utilities, what is the minimum posterior probability your agent needs to have for $B = ok$ before accepting an engine becomes the rational action?

Let $p = P(B = ok|W, R, B)$, then $EU(reject) = -1$ and $EU(accept) = p \cdot 2 + (1-p) \cdot -20$. Accepting would be a rational decision if $EU(accept) \geq EU(reject)$. These expected utilities reach equality when:

$$-1 = p \cdot 2 + (1-p) \cdot -20$$

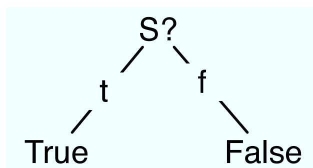
Or when $p = \frac{19}{22}$.

5. (20 pts.) Classification

Consider the following data, which shows observations of three Boolean variables: A indicates whether or not your friend aces his exam, S indicates whether or not he studied, and H indicates whether he wore his lucky hat. In this problem, 1 indicates the variable was true in that observation. You decide to build a decision tree to see which variable is actually more predictive of his grades.

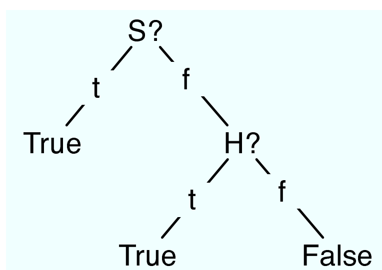
A	S	H
true	*true*	*false*
true	*true*	*true*
true	*true*	*false*
true	*true*	*true*
true	*true*	*false*
true	*false*	*true*
false	*false*	*false*
false	*false*	*true*

(a) (3 points) Draw the decision stump (single-level decision tree) which will be chosen using the information gain criterion presented in class.



The entropy is decreased more by splitting on S.

(b) (4 points) Draw the full decision tree which will be learned using the information gain-guided recursive splitting learner presented in class.



A second recursive split expands the right sub-tree.

(c) (2 points) Which, if either, of the two trees is consistent with the training data? Justify your answer.

Neither. Some training example is misclassified by each tree.

For the following questions, *your answers should fit in a few sentences, possibly one.*

Scenario: Imagine you grow a very large, complex decision tree from a training set which contains many features (attributes).

(d) (4 points) You find that the training set accuracy for your tree is very high, but the test set accuracy (as measured on held out data) is very low. Explain why the accuracy on the training data could be so much higher than on the test data.

Decision trees are very expressive, and so often overfit the training data.

(e) (4 points) You decide to use pruning to create a series of successively smaller subtrees from your full tree (e.g. χ-squared pruning, or truncating the tree at smaller depths). As you prune more and more of the tree, the test accuracy goes up at first, but then starts to drop again after extensive pruning. Explain the rise and then fall of test accuracy in terms of specific terminology and concepts discussed in class.

Pruning the full tree will reduce overfitting, which will yield a tree that more accurately generalizes to unseen data. Continuing to prune the tree will yield a classifier that is too simple to represent the patterns in the training data, so accuracy will eventually decrease.

(f) (3 points) Why is it important to use accuracy on held out data rather than accuracy on training data in order to decide when to stop pruning a decision tree?

Pruning using the training data would typically not have any effect because the recursive algorithm generates a tree that classifies the training data well. That is, deciding to stop when training accuracy decreased would cause an immediate stop. When pruning yields a decline in held-out data accuracy, on the other hand, the overfitting subtrees have been pruned away but the classifier is still expressive enough to model trends in the data.

End of Exam