Massachusetts Institute of Technology

6.034 Artificial Intelligence

Examination Solutions # **2**

Problem 1 Framing the question (26 points)

Part A (6 points)

```
;;;; MAIN TREE
(define main
  (make-tree
   '(branch
     ;; COUNT THE CIRCUIT-BOARDS
     (count (object-description --> o)
           if-end-rtn (db-call `(db-report-count (db-count-items ,o))))
     (what are (object-description --> o) (attribute-list --> a-l)
           if-end-rtn (db-call `(db-show (db-project ,o '(over ,@a-l)))))

;;; ADDED BY BEN
     (    put (object-description --> o1)
          on  (object-description --> o2)
          if-end-rtn
          (db-call
          `(put-on ,o1 ,o2 )))

;;; ALTERNATIVELY ADDED BY BEN
     (    put (object-description --> o1)
          (place --> o2)
          if-end-rtn
          (db-call
          `(put-on ,o1 ,o2 )))

;;;;
;;;; END OF BEN'S ADDITIONS
     ((identify)
      branch
      ;; IDENTIFY THE CIRCUIT-BOARDS
      ((object-description --> o)
      if-end-rtn (db-call `(db-show ,o)))
     (stop if-end-rtn 'quit)
     (rtn (format #t "~%Sorry, I do not understand")
          #f))))
```

You got full credit for either answer, with or without the `db-call`, which is in the code to do an `eval` on the `put-on` expression.

Part B (6 points)

Yes, Ben must make other changes, to ensure that the `object-description` routine will handle the new objects that appear in the class and location fields.

Part C (6 points)

You will have to modify the transition tree network, changing `object-description`, so as to recognize, absorb, and construct relational-database access code from place phrases.

Part D (8 points)

Yes, the network will fail, because "on the workbench" will be absorbed by the `object-description` network, rather than `main`. Once you get into the `object-description` network, it will go ahead and absorb "on the workbench" before the top-level main branch that you implemented in part A even sees the word *on*.

But another answer was also accepted, with full credit:

Yes, the network will fail, because semantic transition trees have no facility for backup (called backtracking by some instructors).

The implementation given in the problem set has partial backup capability in that it tries multiple branches in the face of branch failure. However, the implementation never backs up into the operations of a previously completed routine. In implementations that do backup into the operations of a previously completed routine, the network would actually discover a path corresponding to the intended semantics.

Problem 2 Bayes Nets (31 points)

```
      Q
     / \
    /   \
   G     R
    \   /
     \ /
      A
```

Q has no dependencies:

Var Name	Var Name	Var Name	Cond Prob
			$P(Q \mid --,--,--)$
			1/3

G depends on Q only

Var Name	Var Name	Var Name	Cond Prob
Q			P(G | --,--,--)
T			0.95
F			0.5

R also depends on Q only

Var Name	Var Name	Var Name	Cond Prob
Q			P(R | --,--,--)
T			0.9
F			0.75

A depends on both G and R

Var Name	Var Name	Var Name	Cond Prob
G	R		P(A | --,--,--)
T	T		0.75
T	F		0.5
F	T		0.5
F	F		0.0

Part A (8 points)

The noisy-or model will work, as you can see by plugging numbers into the noisy-or formula.

It is especially convenient that P(A| G=F, R=F) = 0, for that tells us that the leak probability is zero. With a 0 leak, we can read the causal probabilities from the tables, without further computation.

P(A | G, R)

= 1 - (1 - P_C(A | G=T, R=F)) (1 - P_C(A | G=F, R=T)) (1 - P_C(A | leak)) =

1 - (1 - 0.5) (1 - 0.5) (1 - 0)

=

0.75

Then, note that P(A | G, R), given by the formula, is the same as the value given the table. Hence, noisy-or does model the situation.

Part B (6 points)

The following problems are independent of the ones above.

Given the following Bayes Net:

```
        ---> B
```

```
   /
  A
   \
    ---> C ----> D
```

By inspection, from the diagram, the dependencies indicate:

8 P(A=a,B=b,C=c,D=d) = P(D=d| C=c) P(B=b| A=a) P(C=c| A=a) P(A=a)

Part C (8 points)

By the definition of conditional probability:

P(B=T,D=F | A=T,C=F) = P(A=T, B=T, C=F, D=F) / P(A=T, C=F)

But

P(A=T, C=F) = P(A=T, B=T, C=F, D=T) +P(A=T, B=F, C=F, D=T) +P(A=T, B=T, C=F, D=F) +P(A=T, B=F, C=F, D=F)

Hence, given

1	P(A=T,B=T,C=T,D=T)	9	P(A=F,B=T,C=T,D=T)
2	P(A=T,B=T,C=T,D=F)	10	P(A=F,B=T,C=T,D=F)
3	P(A=T,B=T,C=F,D=T)	11	P(A=F,B=T,C=F,D=T)
4	P(A=T,B=T,C=F,D=F)	12	P(A=F,B=T,C=F,D=F)
5	P(A=T,B=F,C=T,D=T)	13	P(A=F,B=F,C=T,D=T)
6	P(A=T,B=F,C=T,D=F)	14	P(A=F,B=F,C=T,D=F)
7	P(A=T,B=F,C=F,D=T)	15	P(A=F,B=F,C=F,D=T)
8	P(A=T,B=F,C=F,D=F)	16	P(A=F,B=F,C=F,D=F)

Then,

P(B=T,D=F | A=T,C=F) = 4 / (3 + 7 + 4 + 8)

Problem 3 Identification Trees (20 points)

Part A (4 points)

For the test **SAT** > **c**, half of the samples end up in an evenly-balanced set, for which the disorder is 1, and half in a homogenous set, for which the disorder is 0. Hence, the average disorder is 0.5.

Part B (4 points)

If the test **SAT** > **c** is chosen as the test at the top of the tree, in if (**SAT** > **c**) is *false*, there is no further testing, because the set is homogeneous.

If (**SAT** > **c**) is *true*, you get the following table:

Test	Rank
SAT > d	3
Grades > a	1
Grades > b	1

The two rows marked 1 have symetric distributions of +s and -s; both put 4 of the 12 samples in a homogenous set, making it highly unlikely that the row marked one will do better, as it places no samples in a homogenous set.

Part C (8 points)

The tenth feature usefully splits the sample data into groups, but it splits the sample data much like the ninth feature, but not quite as well. Therefore, the tenth feature is always dominated by the ninth feature.

True. Obvious.

The tenth feature is useful, but only in combination with the ninth feature. Because ID tree construction does no look ahead or test combinations, the value of the tenth feature is overlooked.

True. It is easy to devise examples in which both of two test both split samples into completely heterogenous groups, but both together split into completely homogenous groups:

```
f1 -->  Y     N
f2  Y   8+    8-
    N   8-    8+
```

The values of the tenth feature are all tightly clustered around the mean. The feature can be useful, but you must first divide all values by the standard deviation of the values for the tenth feature in the sample set.

False. Because each dimension is handled separately, the normalization that is useful with nearest neighbors is not useful with identification trees.

The tenth feature is not useful. However, a new, eleventh feature can be derived from the tenth feature, and that new eleventh feature is useful.

True. Consider day of month versus day of the week. People often show up late on Monday morning, but that would not emerge from an attempt to find regularity in data that only provides day of the month information.

Part D (4 points)

He is more likely to be reacting to the tree built for Group A, for which the number of samples is just about the number of terminals. Each sample has ended up in a group by itself; and because of that individualized handling of each sample, the overfitting label comes to mind.

Problem 4 How they work (23 points)

Part A Neural Nets

A.1 (5 points)

The training software is not likely to work with the mirror-image threshold function. When you derive the weight-change formulas using the mirror-image threshold function, you see that the incorrect sign in the exponent propagates through the derivation, producing a change in sign in the weight changes. That is, the original weight-change formulas will drive the network in the wrong direction, away from the desired outputs.

A.2 (5 points)

The training software may work with the altered training function, but behavior will certainly be different. When you derive the weight-change formulas using the threshold function with the altered exponent, you see that the multiplier of 10^6 propagates through the derivation, producing a multiplicative effect on the weight changes. That is, the original weight-change formulas will drive the network in the right direction, but it is likely that the rate constant would need adjustment to get reasonable behavior.

A.3 (5 points)

If you divide all the weights by 1000, the output of each sigmoid box will be unchanged; thus dividing all weights by 1000 compensates exactly for the multiplier of 1000 in the exponent.

Part B Object Recognition (8 points)

“**We assume** that humans never tilt their heads; consequently, the problem is a problem of pure rotation. **That means** we can get by with just one picture.”

No. The pure rotation was explained in lecture and in the book. Two pictures are needed.

“**We assume** that our software never confuses one feature with another; hence there is never a feature-correspondence problem. **That means** we can get by with just one picture.”

No. The conclusion that you need two or more pictures presumes that points have been placed in perfect correspondence .

“**We assume** that the picture we have is taken with the person’s head in a known, standard orientation (say, the nose points forward and both ears are visible). **That means** we can get by with just one picture.”

No. The conclusion that you need two or more pictures follows whether or not one of the images is taken with the object in a known orientation.

“**We assume** that human faces are exactly modeled by faces painted on basketballs. Given that constraint, the two-dimensional position of a point, in the usual orthographic projection, tells you the three-dimensional coordinates of that point. From the three-dimensional coordinates of a point, we can determine where the point would be if we rotate the sphere. **That means** we can compute as many pictures as we need for Ullman’s recognition scheme from the one picture you supply.”

Yes. If you have enough constraint to determine what an object would look like from multiple perspectives, you do not need to take pictures. This is a powerful learning idea that has been reinvented several times