

6.034 Spring 2009

Midterm Solutions

March 18, 2009

PROBLEM	VALUE	SCORE
1	33	
2	15	
3	30	
TOTAL	78	

Problem 1 (33 points)

- (a) (2 points) Figure 1 illustrates the boundaries for two decision-tree classifiers trained on the same training set. One of the boundaries corresponds to an unpruned decision tree and the other corresponds to a pruned decision tree.

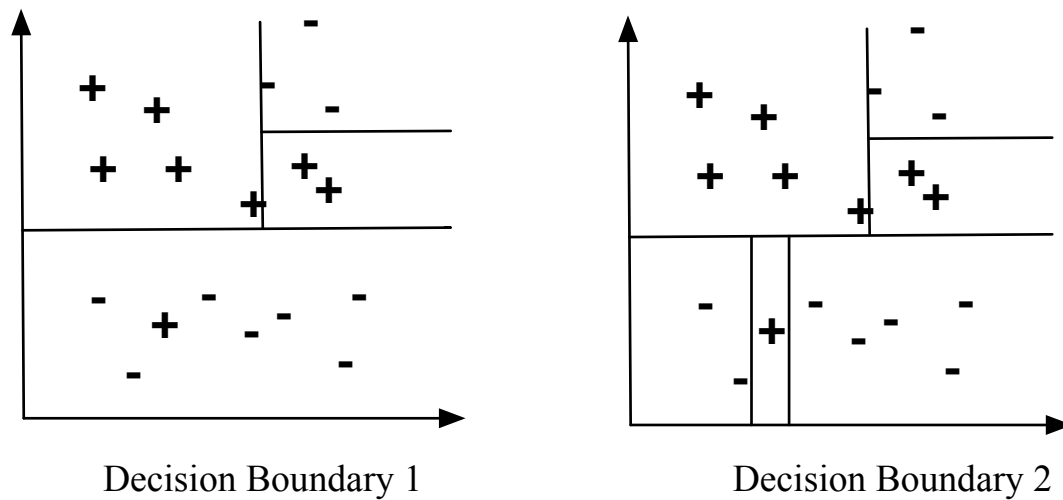


FIGURE 1

Circle the diagram corresponding to the **pruned** decision tree.

The figure on the left is pruned.

(b) (6 points) In class, we defined add- k Laplace correction, which is commonly used for smoothing Bayesian classifiers. Consider training classifiers with two values of k : 1 and 50.

- Would the two resulting classifiers have the same performance on the training set? If “no,” which would perform better? Explain briefly.

$k = 1$ will perform better on the training set than $k = 50$. When k is small, the classifier will fit the training data closely; as k increases, this fit diminishes.

- Would the two classifiers have the same performance on the test set? If “no,” which would perform better? Explain briefly.

We cannot say. This answer depends on the test set we are using. Multiple factors, like training set size and the similarity of the testing data to the training data, could tip the accuracy in favor of either $k = 1$ or $k = 50$.

- (c) (6 points) You want to develop a classifier that identifies the disease “psueditis” based on the results of several tests. Missing a true case of psueditis has grave consequences for a patient, so it is of crucial importance to minimize false negatives. In contrast, false positive identification of psueditis is a minor concern because additional tests can easily clarify the confusion. Explain how you can modify the Bayesian classifier presented in class to minimize the number of false negatives.

We want to introduce a bias in favor of classifying a patient as having psueditis. We could do this in at least two ways:

- Set the prior probability $P(\text{psueditis})$ to be relatively large; say, 0.90.
- Instead of giving back the classification with highest priority (i.e. using argmax), we could use a threshold to decide. We impose a relatively low threshold like 0.10, and only return a negative answer if the probability of psueditis is computed to be less than that threshold. Under this setup, even if the chance of the disease is 15%, we conservatively return “You have psueditis.”

(d) (15 points) Consider an algorithm for decision tree construction that randomly selects an attribute for splitting (rather than using the minimal entropy criterion presented in class).

- Is this algorithm guaranteed to generate a tree that does not test the same attribute twice? Explain briefly.

No. Unlike the minimum-entropy criterion, choosing randomly does not guarantee that you won't split on the same attribute twice. (Unless you explicitly track this in your code, of course.)

- Is this algorithm guaranteed to correctly separate the training data? Explain briefly.

Yes. This algorithm does not stop until it completely separates the training data; that's its stopping condition! If you were to implement some sort of time or depth cutoff, this guarantee goes away.

- Is this algorithm guaranteed to provide the same classification accuracy on the *training* set as the entropy-based algorithm presented in class? Explain briefly.

Yes. Both algorithms continue to split the data until it is completely separated, so both algorithms will have 100% classification accuracy on the training set.

- Is this algorithm guaranteed to provide the same classification accuracy on the *training* set as the entropy-based algorithm presented in class, once both algorithms are pruned? Explain briefly.

No. The training accuracy for both trees can only go down by pruning, but we have no way of saying in what way. There is certainly no guarantee that the pruned trees will have the same accuracy—we could only really say that if the two generated trees happened to be the same.

- Is this algorithm guaranteed to provide the same classification accuracy on the *test* set as the entropy-based algorithm presented in class, once both algorithms are pruned? Explain briefly.

No. We prune the minimum-entropy tree to avoid overfitting to the training data, so hopefully the testing accuracy of that tree has gone up. There's no telling if pruning the random tree will increase, decrease, or not change the test accuracy. Again, we can't make any certain comparison between the minimum-entropy tree and the randomized tree.

- (e) (4 points) You are given a set A of k points which is not linearly-separable. If the point b is removed from this set, then the set of the remaining $k - 1$ points becomes linearly-separable.

Consider training a perceptron algorithm on the original set of k points for n iterations. Would this algorithm correctly classify $k - 1$ points in the set $A - \{b\}$? Explain briefly.

We cannot guarantee that. Certainly, after running the Perceptron algorithm for n rounds, it could be getting all but 1 point correct, but there is no guarantee that 1 incorrect point is b . Furthermore, the perceptron might be getting any number of points wrong: 1, 2, 3, or more! We only know the number of misclassified points for sure when the algorithm converges naturally—to zero error.

Problem 2 (15 points)

DECISION TREE

- (a) (5 points) Write an expression for the average entropy of the test $x_2 > -0.5$ for the data below. You do not need to find the final numerical value, but do not leave any variables in your expression. (You can leave logs in your answer.)

i	x_1	x_2	y
1	-2	0	1
2	2	2	1
3	-2	-1	-1
4	-2	3	-1

The test $x_2 > -0.5$ splits the data into sets $\{1,2,4\}$ and $\{3\}$. We want to find the weighted average of the entropy in both sets. $\{3\}$ has 0 entropy, and $\{1,2,4\}$ is split with 2 +'s and 1 -. So the average entropy is

$$\frac{1 \times 0 + 3 \times \left(-\frac{1}{3} \log\left(\frac{1}{3}\right) - \frac{2}{3} \log\left(\frac{2}{3}\right)\right)}{4}$$

- (b) (5 points) For the same data, write a test (of the form $x_i > value$) considered by the decision tree algorithm that has average entropy equal to 1.

There are several options, like $x_2 > 1$. An example of a test that is *not* valid even though its entropy is 1 is $x_2 > -4$: the id-tree algorithm has no reason to consider such a test—it doesn't split the data at all!

PERCEPTRON

- (c) (5 points) The following table shows a data set and the number of times each point is misclassified during a run of the perceptron algorithm. What is the equation of the separating line found by the algorithm, as a function of x_1 and x_2 ? Assume that the learning rate is 1 and the initial weights are all 0.

i	x_1	x_2	y	times misclassified
1	-3	2	+1	1
2	-1	1	+1	0
3	-1	-1	-1	2
4	2	2	-1	1
5	1	-1	-1	0

The weights will be a linear combination of the data points, weighted by the number of times they are misclassified and their class. So only points 1,3, and 4 matter. Remember we're adding on an extra always-one feature for the offset b .

$$w = \langle b, w_1, w_2 \rangle = (1)(+1) \langle 1, -3, 2 \rangle + (2)(-1) \langle 1, -1, -1 \rangle + (1)(-1) \langle 1, 2, 2 \rangle$$

$$w = \langle b, w_1, w_2 \rangle = \langle 1, -3, 2 \rangle + \langle -2, 2, 2 \rangle + \langle -1, -2, -2 \rangle = \langle -2, -3, 2 \rangle$$

These weights lead to the equation for the separating line:

$$-2 - 3x_1 + 2x_2 = 0$$

Problem 3 (30 points)

In this problem, we are going to tackle the nursery rhyme Three Blind Mice as a FOL problem. For those of you unfamiliar, here are the words to the nursery rhyme:

Three blind mice. Three blind mice.
See how they run. See how they run.
They all ran after the farmer's wife,
Who cut off their tails with a carving knife,
Did you ever see such a sight in your life,
As three blind mice?

We are going to convert English sentences that encapsulate the characters and situations in this nursery rhyme into FOL sentences, convert these FOL sentences to clausal form, and then do a resolution-refutation proof concerning the mice in this nursery rhyme.

Please carefully follow the directions, making sure that you do everything asked in each part. Sometimes you will need to express an English sentence as FOL and convert to clausal form. Sometimes you will only need to convert a FOL sentence to clausal form. And sometimes we will give you the English sentence, the FOL sentence, and the clauses.

Also, be sure to copy the clauses from this page to the proof where applicable. We have noted the number of clauses you should get from each sentence.

For each English sentence below, fill in the missing steps in converting the English sentence to FOL logic to clausal form. Two have been done for you. Use the following predicates:

Mouse(x) = x is a mouse
Blind(x) = x is blind
FarmersWife(x) = x is a farmer's wife
Run(x) = x runs
Chase(x, y) = x chases y
Cut(x, y) = x cuts the tail off y
Tail(x) = x has a tail
Eq(x, y) = x is equal to y

(a) There are exactly three blind mice.

FOL:

Answer: $\exists x \exists y \exists z \text{Mouse}(x) \wedge \text{Blind}(x) \wedge \text{Mouse}(y) \wedge \text{Blind}(y) \wedge \text{Mouse}(z) \wedge \text{Blind}(z) \wedge \neg \text{Eq}(x,y) \wedge \neg \text{Eq}(y,z) \wedge \neg \text{Eq}(x,z) \wedge (\forall w \text{Mouse}(w) \wedge \text{Blind}(w) \rightarrow \text{Eq}(w,x) \vee \text{Eq}(w,y) \vee \text{Eq}(w,z))$

Clauses (10 clauses):

Answer:
Mouse(Huey)
Blind(Huey)
Mouse(Dewey)
Blind(Dewey)
Mouse(Louie)
Blind(Louie)
 $\neg \text{Eq}(\text{Huey}, \text{Dewey})$
 $\neg \text{Eq}(\text{Dewey}, \text{Louie})$
 $\neg \text{Eq}(\text{Huey}, \text{Louie})$
 $\neg \text{Mouse}(w) \vee \neg \text{Blind}(w) \vee \text{Eq}(w, \text{Huey}) \vee \text{Eq}(w, \text{Dewey}) \vee \text{Eq}(w, \text{Louie})$

(b) (4 points) All blind mice run.

FOL:

$\forall x. \text{Mouse}(x) \wedge \text{Blind}(x) \rightarrow \text{Run}(x)$

Clauses (1 clause):

$\neg \text{Mouse}(x) \vee \neg \text{Blind}(x) \vee \text{Run}(x)$

(c) (5 points) There is only one farmer's wife.

FOL:

$$\exists x. FarmersWife(x) \wedge (\forall y. FarmersWife(y) \rightarrow Eq(x, y))$$

Clauses (2 clauses):

$$FarmersWife(Rose) \\ \neg FarmersWife(y) \vee Eq(y, Rose)$$

(d) (3 points) All blind mice chase a farmer's wife.

FOL:

$$\text{Answer: } \forall x \text{ Mouse}(x) \wedge \text{Blind}(x) \rightarrow \exists y \text{ FarmersWife}(y) \wedge \text{Chase}(x, y)$$

Clauses (2 clauses):

$$\neg \text{Mouse}(x) \vee \neg \text{Blind}(x) \vee \text{FarmersWife}(f(x)) \\ \neg \text{Mouse}(x) \vee \neg \text{Blind}(x) \vee \text{Chase}(x, f(x))$$

(e) (5 points) All farmer's wives cut the tail off blind mice that chase them.

FOL:

$$\forall x \forall y \text{FarmersWife}(x) \wedge \text{Blind}(y) \wedge \text{Mouse}(y) \wedge \text{Chase}(y, x) \rightarrow \text{Cut}(x, y)$$

Clauses (1 clause):

$$\neg \text{FarmersWife}(x) \vee \neg \text{Blind}(y) \vee \neg \text{Mouse}(y) \vee \neg \text{Chase}(y, x) \vee \text{Cut}(x, y)$$

(f) If there is someone who cuts off your tail, you have no tail.

FOL:

$$\text{Answer: } \forall x \exists y \text{Cut}(y, x) \rightarrow \neg \text{Tail}(x)$$

Clauses (1 clause):

$$\text{Answer: } \neg \text{Cut}(f(x), x) \vee \neg \text{Tail}(x)$$

- (g) (5 points) We want to prove “All blind mice do not have tails.” Write this sentence as a FOL sentence.

FOL:

$$\forall x \text{Mouse}(x) \wedge \text{Blind}(x) \rightarrow \neg \text{Tail}(x)$$

Because we are going to use resolution refutation, negate this sentence and use the negated sentence to create the clauses that we will use in our proof.

Clauses (3 clauses from the *negation* of the FOL sentence you wrote above):

$$\begin{array}{l} \text{Mouse}(\text{Fred}) \\ \text{Blind}(\text{Fred}) \\ \text{Tail}(\text{Fred}) \end{array}$$

- (h) (8 points) Now fill all the clauses you derived above into the table below and do a resolution refutation proof to derive a contradiction. Write the numbers of the two clauses (from the step column) you are unifying into P1 and P2, and fill in the unifier you are using.

The table is longer than you should need.

Step	P1	P2	Clause	Unifier
1	From (a)		Mouse(Huey)	N/A
2	From (a)		Blind(Huey)	N/A
3	From (a)		Mouse(Dewey)	N/A
4	From (a)		Blind(Dewey)	N/A
5	From (a)		Mouse(Louie)	N/A
6	From (a)		Blind(Louie)	N/A
7	From (a)		$\neg \text{Eq}(\text{Huey}, \text{Dewey})$	N/A
8	From (a)		$\neg \text{Eq}(\text{Dewey}, \text{Louie})$	N/A
9	From (a)		$\neg \text{Eq}(\text{Huey}, \text{Louie})$	N/A
10	From (a)		$\neg \text{Mouse}(w) \vee \neg \text{Blind}(w) \vee \text{Eq}(w, \text{Huey}) \vee \text{Eq}(w, \text{Dewey}) \vee \text{Eq}(w, \text{Louie})$	N/A
11	From (b)		$\neg \text{Mouse}(x) \vee \neg \text{Blind}(x) \vee \text{Run}(x)$	N/A
12	From (c)		<i>FarmersWife</i> (Rose)	N/A
13	From (c)		$\neg \text{FarmersWife}(y) \vee \text{Eq}(y, \text{Rose})$	N/A
14	From (d)		$\neg \text{Mouse}(x) \vee \neg \text{Blind}(x) \vee \text{FarmersWife}(f(x))$	N/A
15	From (d)		$\neg \text{Mouse}(x) \vee \neg \text{Blind}(x) \vee \text{Chase}(x, f(x))$	N/A
16	From (e)		$\neg \text{FarmersWife}(x) \vee \neg \text{Blind}(y) \vee \neg \text{Mouse}(y) \vee \neg \text{Chase}(y, x) \vee \text{Cut}(x, y)$	N/A
17	From (f)		$\neg \text{Cut}(f(x), x) \vee \neg \text{Tail}(x)$	N/A
18	From (g)		<i>Mouse</i> (Fred)	N/A
19	From (g)		<i>Blind</i> (Fred)	N/A
20	From (g)		<i>Tail</i> (Fred)	N/A
21	14	18	$\neg \text{Blind}(\text{Fred}) \vee \text{FarmersWife}(f(\text{Fred}))$	x/Fred

Step	P1	P2	Clause	Unifier
22	19	21	$FarmersWife(f(Fred))$	
23	16	22	$\neg Mouse(y) \vee \neg Blind(y) \vee \neg Chase(y, f(Fred)) \vee Cut(f(Fred), y)$	$x/f(Fred)$
24	18	23	$\neg Blind(Fred) \vee \neg Chase(Fred, f(Fred)) \vee Cut(f(Fred), Fred)$	$x/Fred$
25	19	24	$\neg Chase(Fred, f(Fred)) \vee Cut(f(Fred), Fred)$	
26	15	18	$\neg Blind(Fred) \vee Chase(Fred, f(Fred))$	$x/Fred$
27	19	26	$Chase(Fred, f(Fred))$	
28	25	27	$Cut(f(Fred), Fred)$	$x/Fred$
29	17	28	$\neg Tail(Fred)$	
30	20	29	$false$	
31				
32				
33				
34				
35				
36				
37				
38				
39				
40				