Solutions to 15-781 Midterm, Fall 2002

YOUR ANDREW USERID IN CAPITAL LETTERS:

YOUR NAME:

- There are 5 questions.
- Questions 1-5 are worth 20 points each.
- The maximum possible total score is 100.
- Unless otherwise stated there is no need to show your working.

1 Decision Trees (20 points)

Master Yoda is concerned about the number of Jedi apprentices that have turned to the Dark Side, so he's decided to train a decision tree on some historical data to help identify problem cases in the future. The following table summarizes whether or not each of 12 initiates turned to the Dark Side based on their age when their Jedi training began, whether or not they completed their training, their general disposition, and their species.

Dark Side	Age Started Training	Completed Training	Disposition	Species
0	5	1	Happy	Human
0	9	1	Happy	Gungan
0	6	0	Happy	Wookiee
0	6	1	Sad	Mon Calamari
0	7	0	Sad	Human
0	8	1	Angry	Human
0	5	1	Angry	Ewok
1	9	0	Happy	Ewok
1	8	0	Sad	Human
1	8	0	Sad	Human
1	6	0	Angry	Wookiee
1	7	0	Angry	Mon Calamari

(a) *(3 points)* What is the initial entropy of *Dark Side*?

$-\frac{5}{12}\log_2\frac{5}{12} - \frac{7}{12}\log_2\frac{7}{12} = \mathbf{0.979868756651153}$

(b) *(3 points)* Which attribute would the decision-tree building algorithm choose to use for the root of the tree?

Completed Training

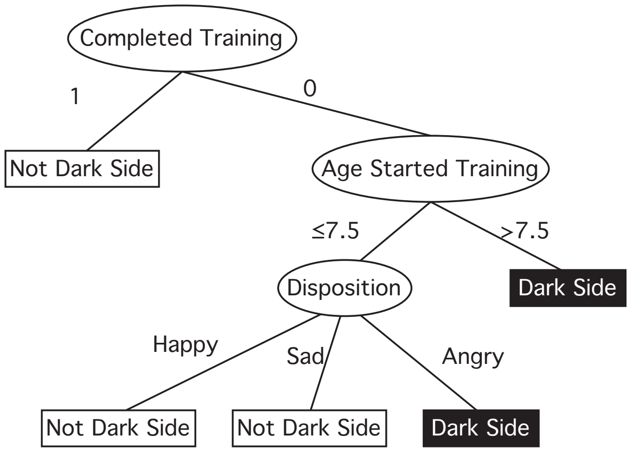
(c) *(3 points)* What is the information gain of the attribute you chose to split on in the previous question?

$\mathbf{a} - (\frac{5}{12}(-\frac{0}{5}\log_2\frac{0}{5} - \frac{5}{5}\log_2\frac{5}{5}) + \frac{7}{12}(-\frac{5}{7}\log_2\frac{5}{7} - \frac{2}{7}\log_2\frac{2}{7})) = \mathbf{0.476381758320618}$

where a is the answer to part (a)

(Note that $log0$ **is** $-\infty$**, but we define** $0log0 = 0$**.)**

(d) *(3 points)* Draw the full decision tree that would be learned for this data (with no pruning).



(e) *(2 points)* Consider the possibility that the input data above is noisy and not completely accurate, so that the decision tree you learned may not accurately reflect the function you want to learn. If you were to evaluate the three initiates represented by the data points below, on which one would you be most confident of your prediction, and why?

Name	Age Started Training	Completed Training	Disposition	Species
Ardath	5	0	Angry	Human
Barbar	8	0	Angry	Gungan
Caldar	8	0	Happy	Mon Calamari

Barbar. The rule we learned is that you turn to the Dark Side if you did not complete your training and you either were too old or angry. Barbar falls under both clauses of the OR part, so even if one half of the rule learned is wrong, he still goes to the Dark Side. A variety of answers were accepted provided they had suitable justification.

(f) *(3 points)* Assume we train a decision tree to predict Z from A, B, and C using the following data (with no pruning):

Z	A	B	C
0	0	0	0
0	0	0	1
0	0	0	1
0	0	1	0
0	0	1	1
1	0	1	1
0	1	0	0
1	1	0	1
1	1	1	0
1	1	1	0
0	1	1	1
1	1	1	1

What would be the training set error for this dataset? Express your answer as the number of records out of 12 that would be misclassified.

2. We have four pairs of records with duplicate input variables, but only two of these have contradictory output values. One item of each of these two pairs will always be misclassified.

(g) *(3 points)* Consider a decision tree built from an arbitrary set of data. If the output is discrete-valued and can take on k different possible values, what is the maximum training set error (expressed as a fraction) that any data set could possibly have?

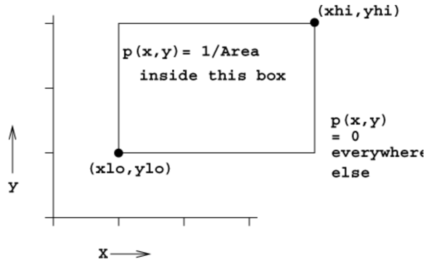
$\frac{k-1}{k}$ **Consider a set of data points with identical inputs but with outputs evenly distributed among the k possible values. The tree will label all these points as a single class which will be wrong for the ones in the other $k-1$ classes. Increasing the relative amount of any one class will guarantee that that class will be chosen as the label for all the points, so the error fraction will decrease (as that class now represents more than $\frac{1}{k}$ of the points.**

2 Probability and Bayes Classifiers (20 points)

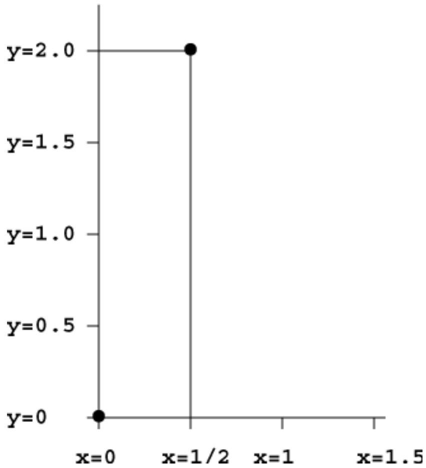
This figure illustrates a simple class of probability density functions over pairs of real-valued variables. We call it the Rectangle PDF.

$(x, y) \sim Rect(x_{lo}, y_{lo}, x_{hi}, y_{hi})$
means

$$p(x,y) = \frac{1}{(x_{hi}-x_{lo})(y_{hi}-y_{lo})} \text{ if } x_{lo} \le x \le x_{hi} \text{ and } y_{lo} \le y \le y_{hi}$$
$$= 0 \text{ otherwise}$$



(a) *(2 points)* Assuming $(x, y) \sim Rect(0, 0, 0.5, 2)$ (as shown in the diagram to the right), compute the value of the density $p(x = \frac{1}{4}, y = \frac{1}{4})$




```
p(x=1/4,y=1/4) = 1/Area = 1/(0.5 * 2) = 1
```

(b) *(3 points)* Under the same assumptions, compute the density $p(y = \frac{1}{4})$

```
The marginal p(y) is constant between 0 through 2 and zero everywhere
else. To integrate to 1 it must have height 0.5. So, since 0 <= 1/4 <= 2,
we have p(y) = 0.5
```

(c) *(3 points)* Under the same assumptions, compute the density $p(x = \frac{1}{4})$

```
2
```

(d) *(3 points)* Under the same assumptions, compute the density $p(x = \frac{1}{4} | y = \frac{1}{4})$

```
x and y are independent so p(x=1/4|y) = p(x=1/4) = 2
```

Maximum Likelihood Estimation of Rectangles

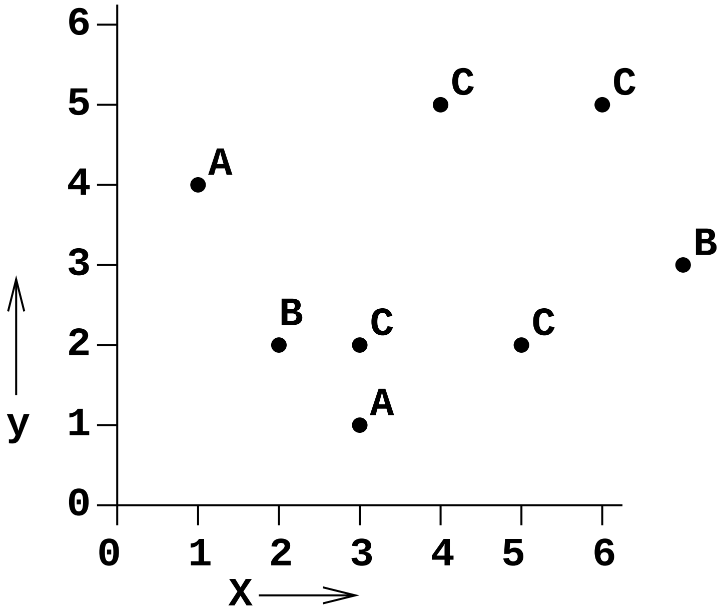
Assume we have R datapoints $(x_1, y_1), (x_2, y_2)...(x_R, y_R)$ where each datapoint is drawn independently from $Rect(x_{lo}, y_{lo}, x_{hi}, y_{hi})$

Suppose we want to find the MLE parameters $(x_{lo}, y_{lo}, x_{hi}, y_{hi})$ that maximize the likelihood of the datapoints. It turns out (no proof given or required) that these MLE values define the bounding box of the datapoints:

$x_{lo}^{MLE} = min_k x_k$
$y_{lo}^{MLE} = min_k y_k$
$x_{hi}^{MLE} = max_k x_k$
$y_{hi}^{MLE} = max_k y_k$

Now, suppose that we use the rectangle distribution as the density estimator for each class of a Bayes Classifier that we're about to learn. The data is

x	y	Class
1	4	A
3	1	A
2	2	B
7	3	B
3	2	C
4	5	C
5	2	C
6	5	C



Assuming we use the Rectangle Bayes Classifier learned from the data, what value will the classifier give for:

(e) *(3 points)* $P(Class = A|x = 1.5, y = 3)$

```
P(A|1.5,3) =              p(1.5,3|A) P(A)
              ---------------------------------------------------
              p(1.5,3|A) P(A) + p(1.5,3|B) P(B) + p(1.5,3|C) P(C)

which is clearly 1, since p(1.5,3|B) = p(1.5,3|C) = 0
```

(f) *(3 points)* $P(Class = A|x = 2.5, y = 2.5)$

```
P(A|2.5,2.5) =              p(2.5,2.5|A) P(A)
              ---------------------------------------------------
              p(2.5,2.5|A) P(A) + p(2.5,2.5|B) P(B) + p(2.5,2.5|C) P(C)

               =           1/6 * 1/4          =      1/6    =    5    =   5
                   ----------------------        ---------     -----     ---
                   1/6 * 1/4 + 1/5 * 1/.4        1/5 + 1/6     6 + 5      11
```

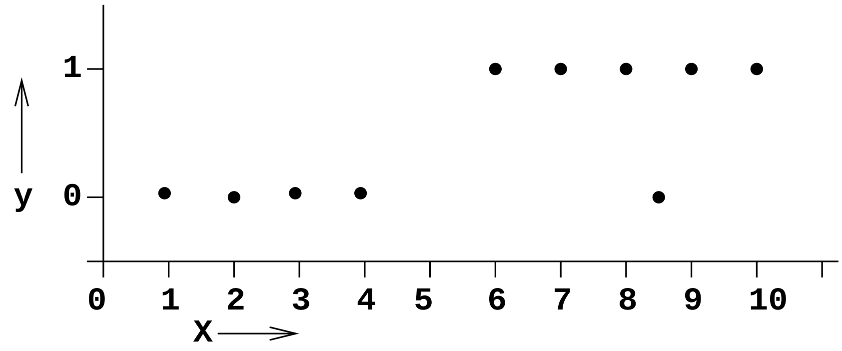
(g) *(3 points)* $P(Class = A|y = 5)$

```
0 (because P(y=5|A) = 0)
```

3 Cross Validation (20 points)

Suppose we are learning a classifier with binary output values Y=0 and Y=1. There is one real-valued input X. Here is our data:

X	Y
1	0
2	0
3	0
4	0
6	1
7	1
8	1
8.5	0
9	1
10	1



Assume we will learn a decision tree on this data. Assume that when the decision tree splits on the real valued attribute x, it puts the split threshold halfway between the attributes that surround the split. For example, using information gain as the splitting criterion, the decision tree would initially choose to split at x = 5, which is halfway between the x = 4 and x = 6 datapoints.

Let Algorithm DT2 be the method of learning a decision tree with only two leaf nodes (i.e. only one split).

Let Algorithm DT* be the method of learning a decision tree fully with no pruning.

(a) *(5 points)* What will be the training set error of DT2 on our data? In this part, and all future parts, you can express your answer as the number of misclassifications out of 10.

```
1/10, because the decision tree will split at x = 5 and will make one
mistake at the right branch
```

(b) *(5 points)* What will be the leave-one-out-cross-validation error of DT2 on our data?

```
1/10, because the decision tree will split at approximately
x = 5 on each fold and the left-out-point will be consistent with
the prediction in all folds except for the ``leave out x = 8.5'' fold
```

(c) *(5 points)* What will be the training set error of DT* on our data?

```
0/10 because there will be no inconcistencies in any leaves
```

(d) *(5 points)* What will be the leave-one-out-cross-validation error of DT* on our data?

```
3/10. The leave-one-out points that will be wrongly predicted
are x = 8, x = 8.5 and x = 9. For example when x=8 is left out
the decision tree that will be learned is
if x < 5    predict 0
if x > 5    if x < 7.75 (halfway point between 7 and 8.5) predict 1
            if x > 7.75   if x < 8.75  predict 0
                          if x > 8.75  predict 1

which wrongly predicts 0 for the left-out point
```

4 Computational learning theory (20 points)

True or false: For a-d, if false, give a counter example. If true, give a 1 sentence justification.

(a) *(3 points)* Within the setting of the PAC model it is impossible to assure with probability 1 that the concept will be learned perfectly (i.e., with true error=0), regardless of how many training examples are provided.

`Answer: true. In this setting instances are drawn at random, and we therefore can never be certain the training examples sufficient to learn the concept will be seen within any finite sample of instances.`

(b) *(3 points)* If the Halving Algorithm has made exactly $\lfloor log_2|H| \rfloor$ mistakes, and H contains the target concept, then it must have learned a hypothesis with true error=0, regardless of what training sequence we presented and what hypothesis space H it considered.

`Answer: true. After each mistake the size of the version space will be reduced to at most half its initial size. Hence, after floor(log2(|H|)) mistakes, there can be only one hypothesis remaining in the version space.`

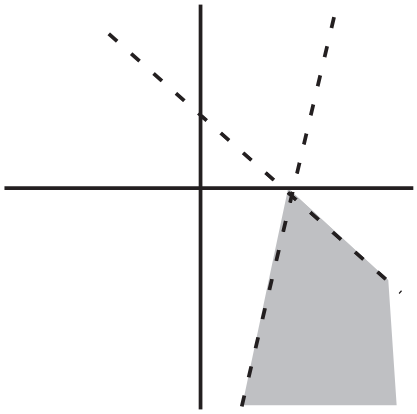
(c) *(3 points)* It is impossible for the Halving Algorithm to learn any concept without it making at least VC(H) mistakes, regardless of what training sequence we present, and what hypothesis space H it considers.

`Answer: false. As we discussed in class: for some sequences of training examples the Halving Algorithm can converge while making zero mistakes, because individual hypotheses will be removed from the version space even if the majority of hypotheses votes correctly.`

(d) *(3 points)* The PAC bounds make a worst case assumption about the probability distribution over the instances X, but it is possible to learn from fewer examples for some distributions over X.

`Answer: true. Consider the probability distribution that assigns probability 1 to a single instance in X, and probability 0 to all other instances. After one training example the concept will be perfectly learned (with error=0).`

Consider the class of concepts H2p defined by conjunctions of two arbitrary perceptrons. More precisely, each hypothesis $h(x) : X \rightarrow \{0, 1\}$ in H2p is of the form $h(x) = p_1(x)$ AND $p_2(x)$, where $p_1(x)$ and $p_2(x)$ are any **two-input** perceptrons. The figure illustrates one such possible classfier in two dimensions.



(e) *(4 points)* Draw a set of three points in the plane that cannot be shattered by H2p.

```
Note each hypothesis forms a ''V'' shaped surface in the plane, where points
within the V are labeled positive.  Three colinear points cannot be
shattered, because no V can capture the case that includes the two outermost
points while excluding the inner point.
```

(f) *(4 points)* What is the VC dimension of H2p? (Partial credit will be given if you can bound it, so show your reasoning!)

```
The VC dimension is 5.  You can shatter a set of 5 points spaced out evenly
on the circumference of a circle.  Note you cannot shatter a set of 6 points
spaced evenly on the circle because you cannot capture the case where the
labels alternate +-+-+-.  (note this doesn't really prove that there exists
*no* set of 6 points that can be shattered, but full credit was given to
anybody who gave this answer).
```

5 Regression and neural networks (20 points)

(a) *(8 points)* Derive a gradient descent training algorithm that minimizes the sum of squared errors for a variant of a perceptron where the output o of the unit depends on its inputs x_i as follows:

$o = w_0 + w_1 x_1 + w_1 x_1^3 + w_2 x_2 + w_2 x_2^3 + ... + w_n x_n + w_n x_n^3$

Give your answer in the form $w_i \leftarrow w_i + \ldots$ for $1 \leq i \leq n$. You do not need to give the update rule for w_0.

`To answer this, calculate the gradient in a fashion analogous to that shown on pages 91-92 of the textbook.  The answer in this case is`

$w_i \leftarrow w_i + \eta \sum_{d \in D} (t_d - o_d)(x_{id} + x_{id}^3)$