15-781 Midterm, Fall 2001

YOUR ANDREW USERID IN CAPITAL LETTERS:

YOUR NAME:

Andrew Moore

Solutions created in haste. Email AWM@CS.CMU.EDU if/when I've made mistakes.

- There are 6 questions.
- Questions 1-5 are worth 19 points each.
- Question 6 is potentially time-consuming and worth only 5 points. Only attempt it if you are certain there aren't any missing parts or errors in your answers to the other questions.
- The maximum possible total score is 100.

1 Decision Trees (19 points)

The following dataset will be used to learn a decision tree for predicting whether a person is happy (H) or sad (S) based on the color of their shoes, whether they wear a wig and the number of ears they have.

Color	Wig	Num. Ears	(Output) Emotion
G	Y	2	S
G	N	2	S
G	N	2	S
B	N	2	S
B	N	2	H
R	N	2	H
R	N	2	H
R	N	2	H
R	Y	3	H

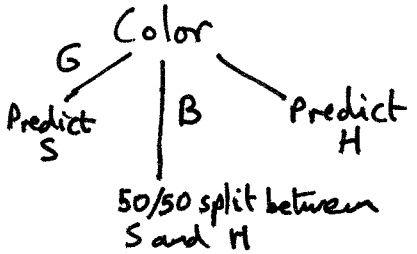
(a) *(2 points)* What is H(Emotion|Wig=Y)? 1

(b) *(2 points)* What is H(Emotion|Ears=3)? 0

(c) *(3 points)* Which attribute would the decision-tree building algorithm choose to use for the root of the tree (assume no pruning).

Color (by inspection: it nicely breaks apart emotion, but the others hardly affect entropy in either branch)

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



(e) *(3 points)* What would be the training set error for this dataset? Express your answer as the percentage of records that would be misclassified.

12½ %

The next two parts do not use the previous example, but are still about decision tree classifiers.

(f) *(3 points)* Assuming that the output attribute can take two values (i.e. has arity 2) what is the maximum training set error (expressed as a percentage) that any dataset could possibly have?

50%

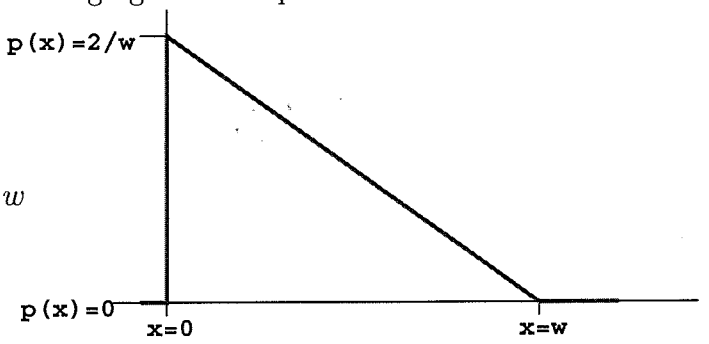
(g) *(3 points)* Construct an example dataset that achieves this maximum percentage training set error. (It must have two or fewer inputs and five or fewer records).

X_1	Y
0	0
0	1
1	0
1	1

2 Probability Density Functions (19 points)

Consider the PDF shown in the following figure and equations

$$p(x) = 0 \quad \text{if } x < 0$$
$$p(x) = \frac{2}{w} - \frac{2x}{w^2} \quad \text{if } 0 \le x \le w$$
$$p(x) = 0 \quad \text{if } w < x$$



(a) *(3 points)* Which one of the following expressions is true? (note—exactly one is true). Write your answer (a choice between 1 to 12) here:

(1) $E[X] = \int_{x=-\infty}^{\infty} (\frac{2}{w} - \frac{2x}{w^2})dx$ (2) $E[X] = \int_{x=-\infty}^{\infty} x(\frac{2}{w} - \frac{2x}{w^2})dx$ (3) $E[X] = \int_{x=-\infty}^{\infty} w(\frac{2}{w} - \frac{2x}{w^2})dx$

(4) $E[X] = \int_{x=0}^{w} (\frac{2}{w} - \frac{2x}{w^2})dx$ (5) $E[X] = \int_{x=0}^{w} x(\frac{2}{w} - \frac{2x}{w^2})dx$ (6) $E[X] = \int_{x=0}^{w} w(\frac{2}{w} - \frac{2x}{w^2})dx$

(7) $E[X] = \int_{w=-\infty}^{\infty} (\frac{2}{w} - \frac{2x}{w^2})dw$ (8) $E[X] = \int_{w=-\infty}^{\infty} x(\frac{2}{w} - \frac{2x}{w^2})dw$ (9) $E[X] = \int_{w=-\infty}^{\infty} w(\frac{2}{w} - \frac{2x}{w^2})dw$

(10) $E[X] = \int_{w=0}^{x} (\frac{2}{w} - \frac{2x}{w^2})dw$ (11) $E[X] = \int_{w=0}^{x} x(\frac{2}{w} - \frac{2x}{w^2})dw$ (12) $E[X] = \int_{w=0}^{x} w(\frac{2}{w} - \frac{2x}{w^2})dw$

(b) *(4 points)* What is $P(x = 1|w = 2)$? 0 (non probability mass)

(c) *(4 points)* What is $p(x = 1|w = 2)$? $\frac{2}{2} - \frac{2}{4} = \frac{1}{2}$

(d) *(4 points)* What is $p(x = 0|w = 1)$? $\frac{2}{1} - \frac{0}{4} = 2$

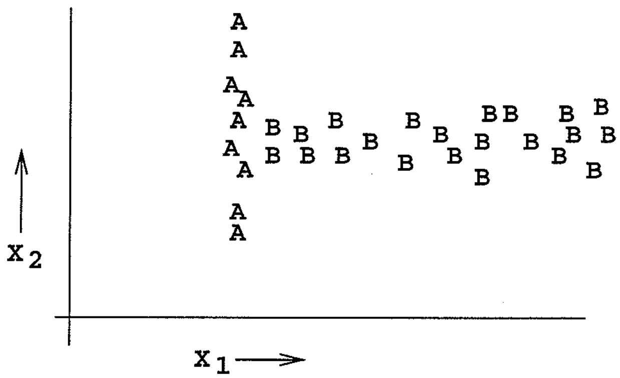
(e) *(4 points)* Suppose you don't know the value of w but you observe one sample from the distribution: $x = 3$. What is the maximum likelihood estimate of w?

$$p(x=3|w) = \frac{2}{w} - \frac{6}{w^2} \quad \text{if } w \ge 3$$
$$= 0 \quad \text{otherwise}$$

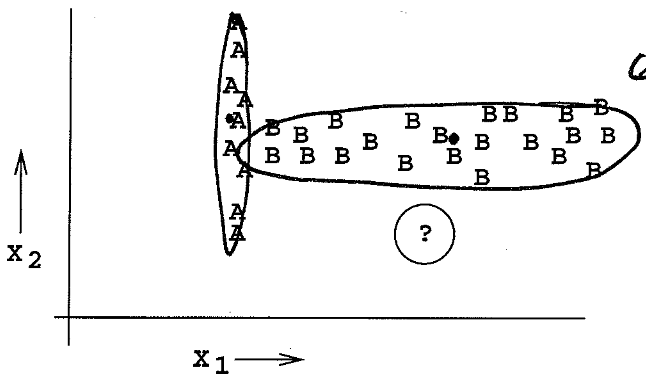
$$\text{MLE } w = \operatorname{argmax}_{w \ge 3} p(x=3|w) = w \text{ s.t. } \frac{\partial}{\partial w} p(x=3|w) = 0$$

$$= w \text{ s.t. } -\frac{2}{w^2} + \frac{12}{w^3} = 0 = w \text{ s.t. } 2 = \frac{12}{w} = 6$$

In a completely different Bayes Classifier example, suppose you trained a Bayes Classifier using General Gaussians on the following data that has two real-valued inputs (X_1 and X_2) and one two-valued categorical output Y.



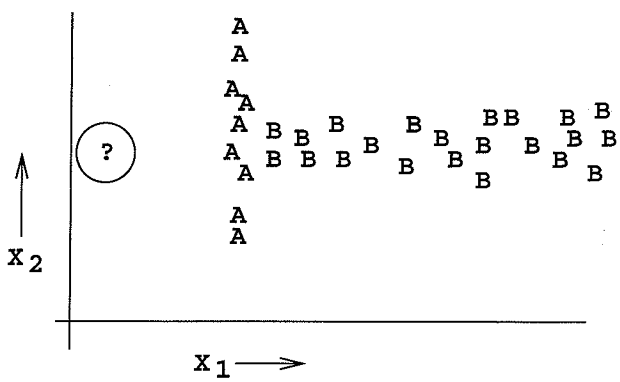
(f) *(3 points)* What class would the Bayes Classifier predict for inputs at the location shown by a question mark in the following figure? (Note: I strongly advise you to answer this by common sense and "eyeballing"—don't waste precious time calculating the Bayes Classifier.



Drawing guess at 2-sigma contours of classes

B: ~~this~~ ? is far closer in Mahal distance to B gaussian

(g) *(3 points)* And what class would it predict for the following location?



B: Only about $3\sigma_{x_1}$ from center of B gaussian, but maybe $10\sigma_{x_1}$ from center of A gaussian.

3 Gaussian Bayes Classifiers (19 points)

(a) *(2 points)* Suppose you have the following training set with one real-valued input X and a categorical output Y that has two values.

X	Y
0	A
2	A
3	B
4	B
5	B
6	B
7	B

You must learn the Maximum Likelihood Gaussian Bayes Classifier from this data. Write your answers in this table:

$\mu_A =$	1	$\sigma_A^2 =$	1	$P(Y=A) =$	2/7
$\mu_B =$	5	$\sigma_B^2 =$	$\frac{4+1+0+1+4}{5} = 2$	$P(Y=B) =$	5/7

I considered asking you to compute $p(X = 2|Y = A)$ using the parameters you had learned. But I decided that was too fiddly. So in the remainder of the question you can give your answers in terms of α and β, where:

$$\alpha = p(X = 2|Y = A)$$
$$\beta = p(X = 2|Y = B)$$

(b) *(2 points)* What is $p(X = 2 \wedge Y = A)$ (answer in terms of α)?

$= P(X=2|Y=A)P(Y=A) = \frac{2}{7}\alpha$

(c) *(2 points)* What is $p(X = 2 \wedge Y = B)$ (answer in terms of β)?

$= P(X=2|Y=B)P(Y=B) = \frac{5}{7}\beta$

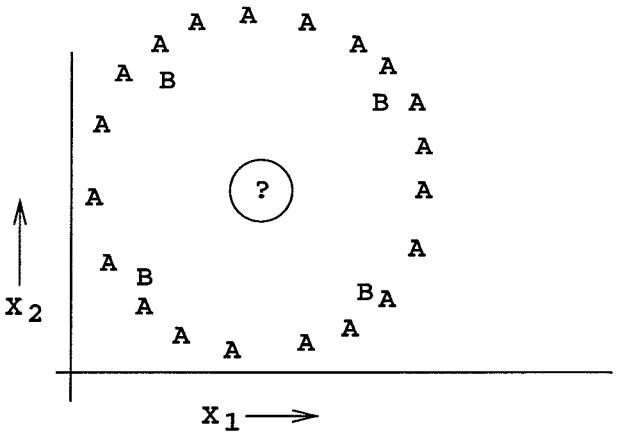
(d) *(2 points)* What is $p(X = 2)$ (answer in terms of α and β)?

$= \frac{1}{7}(2\alpha + 5\beta)$

(e) *(2 points)* What is $P(Y = A|X = 2)$ (answer in terms of α and β)?

$$= \frac{P(Y=A \wedge X=2)}{P(X=2)} = \frac{\frac{2}{7}\alpha}{\frac{1}{7}(2\alpha+5\beta)} = \frac{2\alpha}{2\alpha+5\beta}$$

(h) *(3 points)* Finally, consider the following figure. If you trained a new Bayes Classifier on this data, what class would be predicted for the query location?



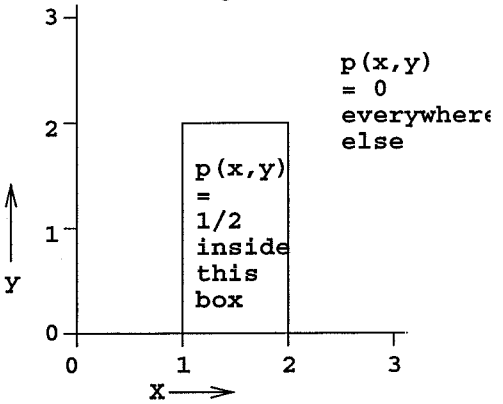
~~A and A~~

Both classes have same centroid, and A has slightly larger variance in each dimension, but the critical fact is that A's class prior is much higher, so we predict (A)

4 Bivariate PDFs (19 points)

Consider the following PDF defined by these equations and sketched in this figure:

$p(x,y) = \frac{1}{2}$ if $x \geq 1 \wedge x \leq 2 \wedge y \geq 0 \wedge y \leq 2$
$p(x,y) = 0$ Otherwise



(a) *(7 points)* What is $p(x)$? (your answer should be a function of x that integrates to 1)

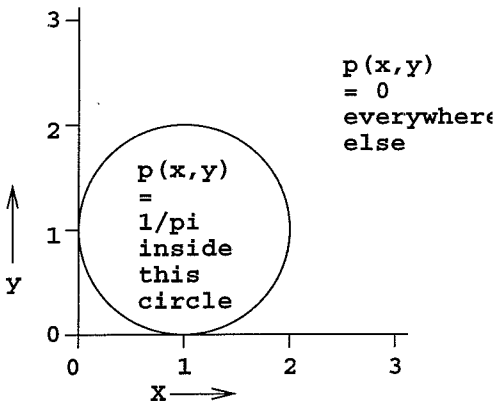
$p(x) =$ 0 if $x < 1$
1 if $1 \leq x \leq 2$
0 if $x > 2$

(b) *(6 points)* Is X independent of Y?

YES

Now consider a different joint distribution over (x, y):

$p(x,y) = \frac{1}{\pi}$ if $(x-1)^2 + (y-1)^2 \leq 1$
$p(x,y) = 0$ Otherwise



(c) *(6 points)* Is X independent of Y?

No. E.G. If $x = 0.01$ then I'm sure y is close to 1. But if $x = 1$ then y could be anywhere between 0 and 2

5 Bayes Rule (19 points)

(a) *(4 points)* I give you the following fact:

$$P(A|B) = 2/3$$

Do you have enough information to compute $P(B|A)$? If not, write "not enough info". If so, compute the value of $P(B|A)$.

Not Enough Info

(b) *(5 points)* Instead, I give you the following facts:

$$P(A|B) = 2/3$$
$$P(A|\sim B) = 1/3$$

Do you now have enough information to compute $P(B|A)$? If not, write "not enough info". If so, compute the value of $P(B|A)$.

Not enough Info

(c) *(5 points)* Instead, I give you the following facts:

$$P(A|B) = 2/3$$
$$P(A|\sim B) = 1/3$$
$$P(B) = 1/3$$

Do you now have enough information to compute $P(B|A)$? If not, write "not enough info". If so, compute the value of $P(B|A)$.

$$P(B|A) = \frac{P(A|B)P(B)}{P(A|B)P(B) + P(A|\sim B)P(\sim B)} = \frac{\frac{2}{3} \times \frac{1}{3}}{\frac{2}{3} \times \frac{1}{3} + \frac{1}{3} \times \frac{2}{3}} = \frac{1}{2}$$

(d) *(5 points)* Instead, I give you the following facts:

$$P(A|B) = 2/3$$
$$P(A|\sim B) = 1/3$$
$$P(B) = 1/3$$
$$P(A) = 4/9$$

Do you now have enough information to compute $P(B|A)$? If not, write "not enough info". If so, compute the value of $P(B|A)$.

* Still $\frac{1}{2}$ (of course)

6 Drunk Squirrels (5 points)

- A drunk squirrel is dropped onto a 1-dimensional branch of an oak tree at location s. s is drawn from a Gaussian: $s \sim N(\mu_s = 0, \sigma_s^2 = 2^2)$.
- The squirrel makes a step. It moves to the right by distance d, where $d \sim N(0, 1)$. (If d is negative, it moves to the left of course). If we write f as the final location of the squirrel, we see $f \sim N(s, 1)$.
- d is independent of s.

The squirrel ends up at location location $f = 2$. What is the most likely location s that the squirrel landed on the branch initially?

$$\begin{pmatrix} s \\ f \end{pmatrix} \sim N\left(\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \Sigma = \begin{pmatrix} 4 & 4 \\ 4 & 5 \end{pmatrix} \right)$$

$$\mu_{s|f} = \mu_s + \frac{\Sigma_{sf}}{\Sigma_{ff}}(f - \mu_f) = 0 + \frac{4}{5}(f - 0) = \frac{4}{5}f$$

Most likely s given $f=2$

$$= \underset{s}{\text{argmax}}\ P(s \mid f=2) = \mu_{s|f=2} = \frac{4}{5} \times 2 = 1.6$$

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*?

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

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

Completed Training

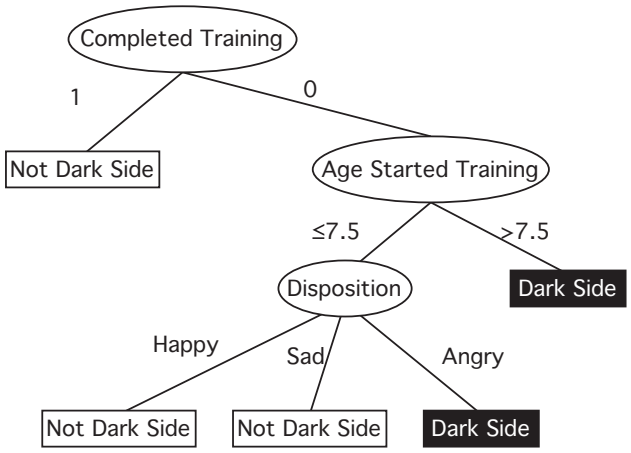
(c) *(3 points)* What is the information gain of the attribute you chose to split on in the previous question?

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

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.

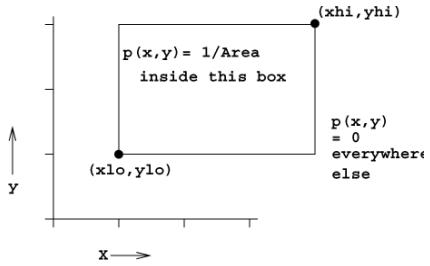
(g) *(3 points)* Consider a decision tree built from an arbitrary set of data. If the output is discrete-valued and can take on k different possible values, what is the maximum training set error (expressed as a fraction) that any data set could possibly have?

$\frac{k-1}{k}$ 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.

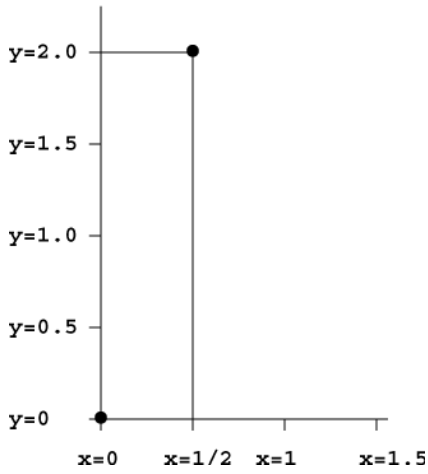
2 Probability and Bayes Classifiers (20 points)

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

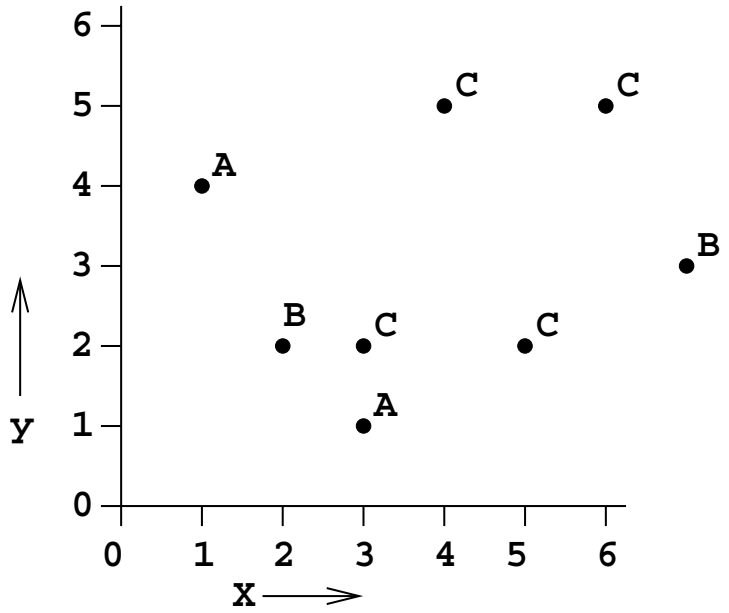
Assume we have R datapoints $(x_1, y_1), (x_2, y_2)...(x_R, y_R)$ where each datapoint is drawn independently from $Rect(x_{lo}, y_{lo}, x_{hi}, y_{hi})$

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
```

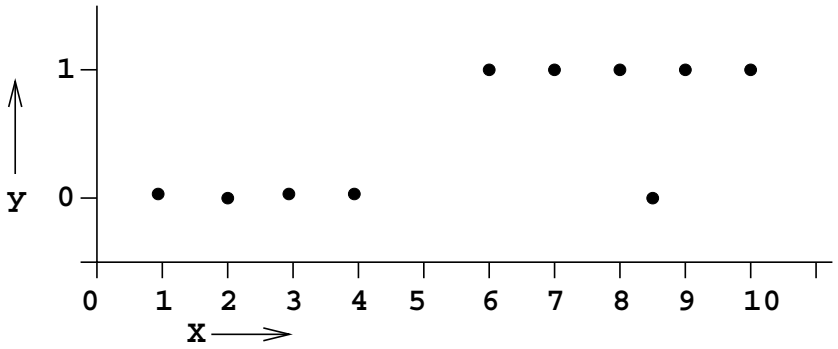
(g) *(3 points)* $P(Class = A | y = 5)$

```
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.

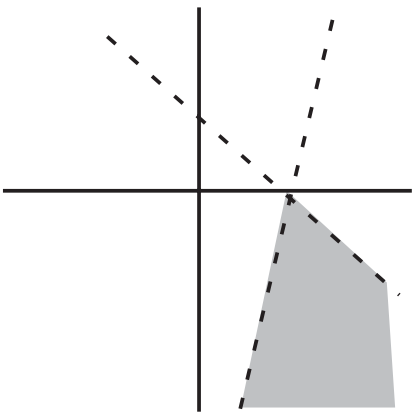
(c) *(3 points)* It is impossible for the Halving Algorithm to learn any concept without it making at least VC(H) mistakes, regardless of what training sequence we present, and what hypothesis space H it considers.

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:

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

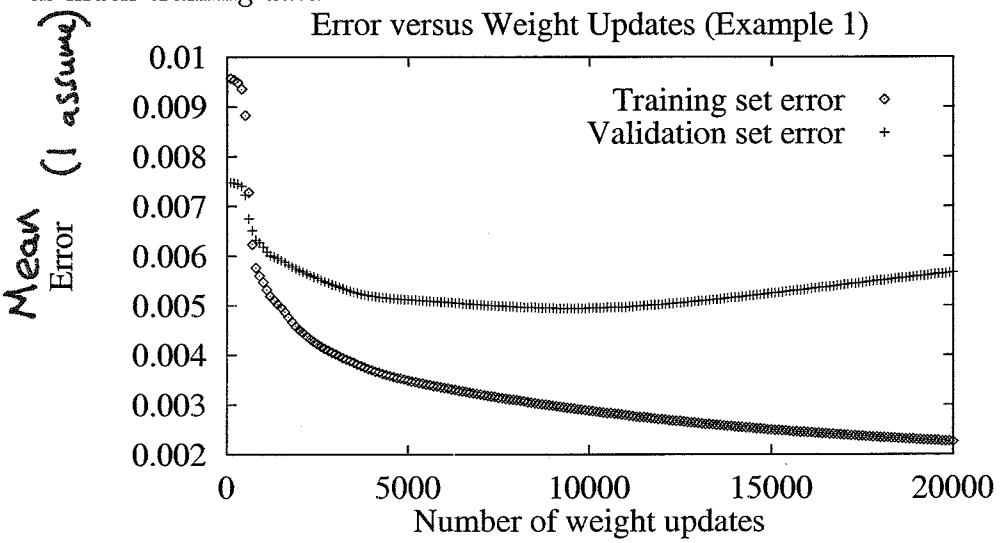
Give your answer in the form $w_i \leftarrow w_i + \ldots$ for $1 \leq i \leq n$. You do not need to give the update rule for w_0.

`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`

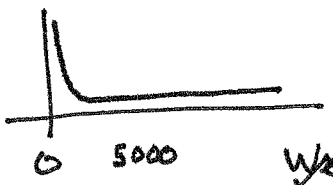
$w_i \leftarrow w_i + \eta \sum_{d \in D} (t_d - o_d)(x_{id} + x_{id}^3)$

CAVEAT: THESE ARE ANDREW'S ANSWERS WHICH MIGHT DIFFER FROM THE OPINION OF THE ORIGINAL QUESTION SETTER'S

Consider the following plot showing training set error and validation set error for the Backpropagation algorithm training a neural network for a particular medical diagnosis problem. Note that the training error decreases monotonically with increasing gradient descent steps, whereas the validation error does not. Suppose now that we were to retrain the same neural network using exactly the same algorithm, but using ten times as much training data.



(b) *(6 points)* Would you expect the training curve to be different? If so, draw what you would expect. In either case, explain your reasoning in **at most** three sentences.



If we are doing batch learning and keep the same learning rate the steps will be bigger and so we'll learn faster (but possibly be unstable).

We are less likely to overfit, given the additional data, so the validation set error will get worse later (and maybe not at all)

(c) *(6 points)* Would you expect the validation set curve to be different? If so, draw what you would expect. In either case, explain your reasoning in **at most** three sentences.

If we ignore the above effect, (e.g. by reducing the learning rate by a factor of ten) then the training curve would remain pretty much the same as before initially, but would end up not overfitting and so might not go down so far at the right. And if we're not overfitting so much, then the validation curve will not increase so much (maybe not at all)

15-781 Midterm, Fall 2003

YOUR ANDREW USERID IN CAPITAL LETTERS:

YOUR NAME:

- There are 9 questions. The ninth may be more time-consuming and is worth only three points, so do not attempt 9 unless you are completely happy with the rest of your answers.
- The maximum possible total score is 100.
- Unless otherwise stated there is no need to show your working.

1 Decision Trees (16 points)

The following dataset will be used to learn a decision tree for predicting whether a mushroom is edible or not based on its shape, color and odor.

Shape	Color	Odor	Edible
C	B	1	Yes
D	B	1	Yes
D	W	1	Yes
D	W	2	Yes
C	B	2	Yes
D	B	2	No
D	G	2	No
C	U	2	No
C	B	3	No
C	W	3	No
D	W	3	No

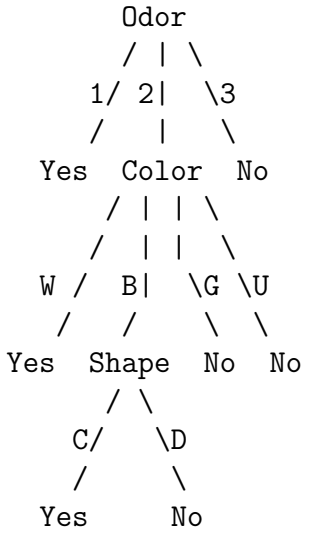
(a) *(4 points)* What is entropy $H(Edible|Order = 1 \text{ or } Odor = 3)$?

$H(Edible|Order = 1 \texttt{ or } Odor = 3) = -\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6} = 1$

(b) *(4 points)* Which attribute would the ID3 algorithm choose to use for the root of the tree (no pruning)?

`Odor`

(c) *(4 points)* Draw the full decision tree that would be learned for this data (no pruning).



(d) *(4 points)* Suppose we have a validation set as follows. What will be the training set error and validation set error of the tree? Express your answer as the number of examples that would be misclassified.

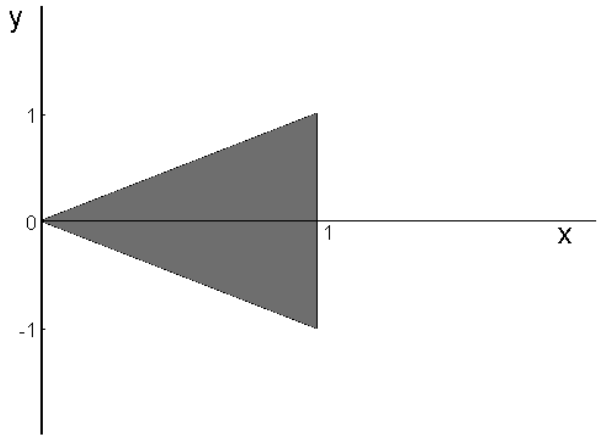
Shape	Color	Odor	Edible
C	B	2	No
D	B	2	No
C	W	2	Yes

```
training set error: 0
validation set error: 1
```

2 Probability Density Functions (8 points)

Suppose the joint Probability Density Function of a pair of random variables (x, y) is given by,

$$p(x,y) = 1 \quad |y| < x, 0 < x < 1$$
$$p(x,y) = 0 \qquad \text{otherwise}$$



(a) *(4 points)* What is $p(y|x = 0.5)$?

$$p(y|x = 0.5) = 1 \quad -0.5 < y < 0.5$$
$$p(y|x = 0.5) = 0 \qquad \texttt{otherwise}$$

(b) *(4 points)* Is x independent of y? (no explanation needed)

`Answer: No`

3 Bayes Classifiers (12 points)

Suppose you have the following training set with three boolean input x, y and z, and a boolean output U.

x	y	z	U
1	0	0	0
0	1	1	0
0	0	1	0
1	0	0	1
0	0	1	1
0	1	0	1
1	1	0	1

Supppose you have to predict U using a naive Bayes classifier,

(a) *(3 points)* After learning is complete what would be the predicted probability $P(U = 0|x = 0, y = 1, z = 0)$?

$$
\begin{aligned}
& P(U = 0|x = 0, y = 1, z = 0) \\
= & \frac{P(U = 0)P(X = 0|U = 0)P(Y = 1|U = 0)P(Z = 0|U = 0)}{P(X = 0, Y = 1, Z = 0)} \\
= & \frac{P(U = 0)P(X = 0|U = 0)P(Y = 1|U = 0)P(Z = 0|U = 0)}{P(U = 0)P(X = 0, Y = 1, Z = 0|U = 0) + P(U = 1)P(X = 0, Y = 1, Z = 0|U = 1)} \\
= & \frac{8}{35} \\
= & 0.229
\end{aligned}
$$

(b) *(3 points)* Using the probabilities obtained during the Bayes Classifier training, what would be the predicted probability $P(U = 0|x = 0)$?

$P(U = 0|x = 0) = \frac{1}{2}$

In the next two parts, assume we learned a Joint Bayes Classifier. In that case...

(c) *(3 points)* What is $P(U = 0|x = 0, y = 1, z = 0)$?

$P(U = 0|x = 0, y = 1, z = 0) = 0$

(d) *(3 points)* What is $P(U = 0|x = 0)$?

$P(U = 0|x = 0) = \frac{1}{2}$

4 Regression (9 points)

I have a dataset with R records in which the i^{th} record has one real-valued input attribute x_i and one real-valued output attribute y_i.

We have the following model with one unknown parameter w which we want to learn from data.

$$y_i \sim N(\exp(wx_i), 1)$$

Note that the variance is known and equal to one.

(a) *(3 points) (no explanation required)* Is the task of estimating w

A. a linear regression problem?

B. a non-linear regression problem?

`Answer: B`

(b) *(6 points) (no explanation required)* Suppose you decide to do a maximum likelihood estimation of w. You do the math and figure out that you need w to satisfy one of the following equations. Which one?

A. $\sum_i x_i \exp(wx_i) = \sum_i x_i y_i \exp(wx_i)$

B. $\sum_i x_i \exp(2wx_i) = \sum_i x_i y_i \exp(wx_i)$

C. $\sum_i x_i^2 \exp(wx_i) = \sum_i x_i y_i \exp(wx_i)$

D. $\sum_i x_i^2 \exp(wx_i) = \sum_i x_i y_i \exp(\frac{wx_i}{2})$

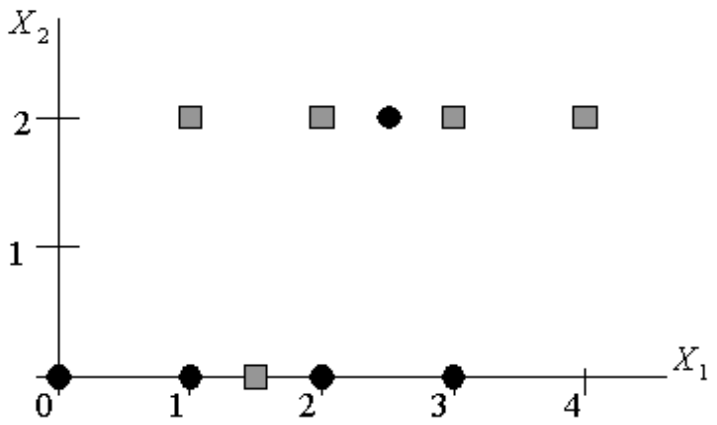
E. $\sum_i \exp(wx_i) = \sum_i y_i \exp(wx_i)$

`Answer: B`

5 Cross Validation (12 points)

Suppose we are learning a classifier with binary output values $Y = 0$ and $Y = 1$. There are two real-valued input attributes X_1 and X_2. Here is our data:

X_1	X_2	Y
0	0	0
1	0	0
2	0	0
2.5	2	0
3	0	0
1	2	1
1.5	0	1
2	2	1
3	2	1
4	2	1



Assume we will learn a decision tree using ID3 algorithm on this data. Assume that when the decision tree splits on the real-valued attributes, it puts the split threshold halfway between the values that surround the highest-scoring split location. For example, if X_2 is selected as the root attribute, the decision tree would choose to split at $X_2 = 1$, which is halfway between $X_2 = 0$ and $X_2 = 2$.

Let Algorithm DT2 be the method of learning a decision tree with only *two* leaf nodes (i.e. only one split), and Algorithm DT* be the method of learning a decision tree *fully* with no pruning.

(a) *(6 points)* What will be the training set error of DT2 and DT* on our data? Express your answer as the number of misclassifications out of 10.

```
Training set error
    DT2: 2 out of 10
    DT*: 0 out of 10
```

(b) *(6 points)* What will be the *leave-one-out* cross-validation error of DT2 and DT* on our data?

```
Leave-one-out cross validation error
    DT2: 2 out of 10
    DT*: 6 out of 10
```

6 Neural Nets (15 points)

(a) *(5 points)* Consider a single sigmoid threshold unit with three inputs, x_1, x_2, and x_3.

$$y = g(w_0 + w_1x_1 + w_2x_2 + w_3x_3) \quad \text{where} \quad g(z) = \frac{1}{1+\exp(-z)}$$

We input values of either 0 or 1 for each of these inputs. Assign values to weights w_0, w_1, w_2 and w_3 so that the output of the sigmoid unit is greater than 0.5 if an only if (x_1 AND x_2) OR x_3.

`There are many solutions. One of them is:`
$w_0 = -0.75$
$w_1 = w_2 = 0.5$
$w_3 = 1$

(b) *(10 points)* Answer the following true or false. (No explanation required).

A. One can perform linear regression using either matrix algebra or using gradient descent.

`Answer: True`

B. The error surface followed by the gradient descent Backpropagation algorithm changes if we change the training data.

`Answer: True`

C. Incremental gradient descent is always a better idea than batch gradient descent.

`Answer: False`

D. Given a two-input sigmoid unit with weights w_0, w_1, and w_2, we can negate the value of the unit output by negating all three weights.

`Answer: This question is ambiguous as the meaning of "negate the value of the unit output" is not clear. On one hand, the value of the unit ouput can never be negative, on the other hand, as` $\frac{1}{1+\exp(-(-x))} = 1 - \frac{1}{1+\exp(-x)}$`, the value of the unit output can be "negated" according to 1. Thus answering either True or False will be deemed as correct.`

E. The gradient descent weight update rule for a unit whose output is $w_0 + w_1(x_1 + 1) + w_2(x_2^2)$ is:

$$\begin{aligned}
\Delta w_0 &= \eta \sum_d (t_d - o_d) \\
\Delta w_1 &= \eta \sum_d [(t_d - o_d)x_{d1} + (t_d - o_d)] \\
\Delta w_2 &= \eta \sum_d [(t_d - o_d)2x_{d2}]
\end{aligned}$$

where

- t_d is the target output for the dth training example
- o_d is the unit output for the d^{th} example.
- x_{d1} is the value of x_1 for the dth training example
- x_{d2} is the value of x_2 for the dth training example

`Answer: False.`
`The formula for calculating` Δw_2 `is wrong. It should be`

$$\Delta w_2 = \eta \sum_d [(t_d - o_d) x_{d2}^2]$$

7 PAC Learning of Interval Hypotheses (15 points)

In this question we'll consider learning problems where each instance x is some integer in the set $X = \{1, 2, \ldots, 125, 126, 127\}$, and where each hypothesis $h \in H$ is an interval of the form $a \leq x \leq b$, where a and b can be any integers between 1 and 127 (inclusive), so long as $a \leq b$. A hypothesis $a \leq x \leq b$ labels instance x positive if x falls into the interval defined by a and b, and labels the instance negative otherwise. Assume throughout this question that the teacher is only interested in teaching concepts that can be represented by some hypothesis in H.

(a) *(3 points)* How many distinct hypotheses are there in H? (hint: when $b = 127$ there are exactly 127 possible values for a). (No explanation required)

$$\frac{(1 + 127) \times 127}{2} = 8128$$

(b) *(3 points)* Assume the teacher provides just one training example: x=64, label=+, then allows the student to query the teacher by generating new instances and asking for their label.

Assuming the student uses the optimal querying algorithm for this case, how many queries will they need to make? No explanation is required, you don't need to write down the optimal algorithm, and we will not be concerned if your answer is wrong by a count of one or two.

$\log_2 63 + \log_2(127 - 64) \approx 12$

(c) *(3 points)* Suppose the teacher is trying to teach the specific target concept $32 \leq x \leq 84$. What is the minimum number of training examples the teacher must present to guarantee that any consistent learner will learn this concept exactly?

`Answer: 4.`
`The training examples are:` $(a - 1, -)$, $(a, +)$, $(b, +)$, $(b + 1, -)$

(d) *(3 points)* Suppose now that instances are drawn at random according to a particular probability distribution $P(X)$, which is unknown to the learner. Each training example is generated by drawing an instance at random according to $P(X)$ then labeling it.

How many such training examples suffice to assure with probability 0.95 that any consistent learner will output a hypothesis whose true error is at most 0.10?

$$\begin{aligned} m &\geq \frac{1}{\epsilon}(\ln |H| + \ln \frac{1}{\delta}) \\ &= \frac{1}{0.1}(\ln(8128) + \ln(\frac{1}{1-0.95})) \\ &\approx 120 \end{aligned}$$

(e) *(3 points)* True or False (no explanation needed). In the above statement, the phrase "to assure with probability 0.95" means that if we were to run the following experiment a million times, then in roughly 950,000 cases or more, the consistent learner will output a hypothesis whose true error is at most 0.10. Each experiment involves drawing the given number of training instances at random (drawn i.i.d. from $P(X)$) and then running the consistent learner.

`Answer: True`

8 Mistake Bounds (9 points)

Assume that we have the predictions below of five experts, as well as the correct answer.

(a) *(3 points)* Using the Weighted-Majority algorithm (with $\beta = 0.5$) in order to track the best expert, show how the weight given to each expert changes after each example. Show your work.

Expert	1	2	3	4	5	Correct Answer
	T	T	T	F	F	F
Weights	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	1	
	F	T	F	T	T	T
Weights	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$	1	1	
	T	F	F	F	T	F
Weights	$\frac{1}{8}$	$\frac{1}{2}$	$\frac{1}{4}$	1	$\frac{1}{2}$	

(b) *(3 points)* Suppose we run the Weighted-Majority algorithm using n experts and $\beta = 0.5$. We find out that the average number of mistakes made by each expert is m but the best expert makes no mistake. Circle below the expression for the bound on mistakes made by the Weighted-Majority algorithm.

A) $O(n)$ B) $O(log_2 n + m)$ C) $O(log_2 n)$ D) none of the above

`Answer: C`

(c) *(3 points)* Notice that since there is an expert who made zero mistakes, we could use the Halving Algorithm instead (which of course corresponds to Weighted-Majority algorithm when $\beta = 0$). Circle below the bound on mistakes made by the Halving algorithm when given the same n experts.

A) $O(n)$ B) $O(log_2 n + m)$ C) $O(log_2 n)$ D) none of the above

`Answer: C`

9 Decision Trees - Harder Questions (4 points)

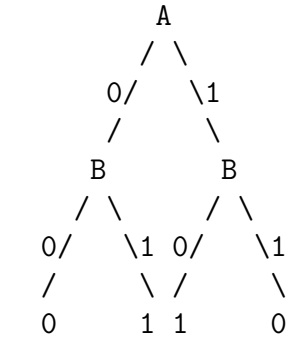
(Only 4 points, so only attempt this if you are happy with all your other answers).

(a) *(2 points)* Suppose we have three binary attributes (A, B and C) and 4 training examples. We are interested in finding a *minimum-depth* decision tree consistent with the training data. Please give a target concept and a *noise-free* training set for which ID3 (no pruning) will not find the decision tree with the minimum depth.

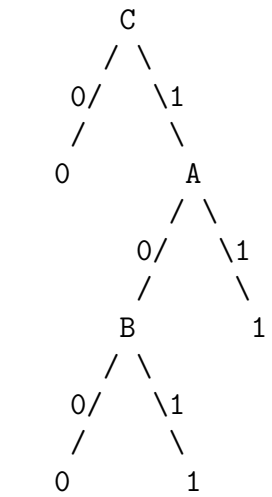
A	B	C	Class
1	1	0	0
1	0	1	1
0	1	1	1
0	0	1	0

Target concept: A XOR B

Minimum-depth tree:



Tree learned by ID3:



(b) *(2 points)* Suppose we learned a decision tree from a training set with binary output values (class = 0 or class = 1). We find that for a leaf node l, (1) there are M training examples falling into it; and (2) its entropy is H. Sketch a simple algorithm which takes

as input M and H and that outputs the number of training examples misclassified by leaf node l.

The entropy function H can be approximated using

$$1 - 4 \times (0.5 - p)^2 \ \ (0 \leq p \leq 1).$$

So $p = 0.5 - \sqrt{\frac{1-H}{4}}$ (here $0 \leq p \leq 0.5$).
The number of misclassified training examples is roughly

$$M \times p = M \times (0.5 - \sqrt{\frac{1-H}{4}}).$$

Solution to 10-701/15-781 Midterm Exam
Fall 2004

1 Introductory Probability and Statistics (12 points)

(a) *(2 points)* If A and B are disjoint events, and $Pr(B) > 0$, what is the value of $Pr(A|B)$?
Answer: 0 (Note that: $A \wedge B = \emptyset$)

(b) *(2 points)* Suppose that the p.d.f of a random variable X is as follows:

$$f(x) = \begin{cases} \frac{4}{3}(1-x^3), & \text{for } 0 \leq x \leq 1 \\ 0, & \text{otherwise} \end{cases}$$

Then $Pr(X < 0) =$?
Answer: 0 (Note that: $0 \leq x \leq 1$)

(c) *(4 points)* Suppose that X is a random variable for which $E(X) = \mu$ and $Var(X) = \sigma^2$, and let c be an arbitrary constant. Which **one** of these statements is true:

A. $E[(X-c)^2] = (\mu - c)^2 + \sigma^2$ D. $E[(X-c)^2] = (\mu - c)^2 + 2\sigma^2$

B. $E[(X-c)^2] = (\mu - c)^2$ E. $E[(X-c)^2] = \mu^2 + c^2 + 2\sigma^2$

C. $E[(X-c)^2] = (\mu - c)^2 - \sigma^2$ F. $E[(X-c)^2] = \mu^2 + c^2 - 2\sigma^2$

Answer: A

$$E[(X-c)^2] = E[X^2] - 2cE[X] + c^2 = Var(X) + [E(X)]^2 - 2c\mu + c^2 = (\mu - c)^2 + \sigma^2$$

(d) *(4 points)* Suppose that k events $B_1, B_2, ..., B_k$ form a partition of the sample space S. For $i = 1, ..., k$, let $Pr(B_i)$ denote the prior probability of B_i. There is another event A that $Pr(A) > 0$. Let $Pr(B_i|A)$ denote the posterior probability of B_i given that the event A has occurred.
Prove that if $Pr(B_1|A) < Pr(B_1)$, then $Pr(B_i|A) > Pr(B_i)$ for at least one value of i $(i = 2, ..., k)$.
(Hint: one or more of these tricks might help: $P(B_i|A)P(A) = P(B_i \wedge A)$, $\sum_{i=1}^{k} P(B_i) = 1$, $\sum_{i=1}^{k} P(B_i|A) = 1$, $P(B_i \wedge A) + P(B_i \wedge \neg A) = P(B_i)$, $\sum_{i=1}^{k} P(B_i \wedge A) = P(A)$)

Answer: We need to prove that if $Pr(B_1|A) < Pr(B_1)$, then $Pr(B_i|A) > Pr(B_i)$ for at least one value of i $(i = 2, ..., k)$.

Proof: We know that $\sum_{i=1}^{k} Pr(B_i) = 1$ and $\sum_{i=1}^{k} Pr(B_i|A) = 1$,
Suppose that for all i $(i = 2, ..., k)$, we have $Pr(B_i|A) \leq Pr(B_i)$, then we can get that
$\sum_{i=1}^{k} Pr(B_i) = Pr(B_1) + \sum_{i=2}^{k} Pr(B_i)$
$> Pr(B_1|A) + \sum_{i=2}^{k} Pr(B_i) > Pr(B_1|A) + \sum_{i=2}^{k} Pr(B_i|A)$
So we get that $1 > 1$. Confliction!.

2 Linear Regression (12 points)

We have a dataset with R records in which the i^{th} record has one real-valued input attribute x_i and one real-valued output attribute y_i.

(a) *(6 points)* First, we use a linear regression method to model this data. To test our linear regressor, we choose at random some data records to be a training set, and choose at random some of the remaining records to be a test set.
Now let us increase the training set size gradually. As the training set size increases, what do you expect will happen with the mean training and mean testing errors? (No explanation required)

- Mean Training Error: A. Increase; B. Decrease

- Mean Testing Error: A. Increase; B. Decrease

Answer:

The training error tends to increase. As more examples have to be fitted, it becomes harder to 'hit', or even come close, to all of them.

The test error tends to decrease. As we take into account more examples when training, we have more information, and can come up with a model that better resembles the true behavior. More training examples lead to better generalization.

(b) *(6 points)* Now we change to use the following model to fit the data. The model has one unknown parameter w to be learned from data.

$$y_i \sim N(log(wx_i), 1)$$

Note that the variance is known and equal to one. (no explanation required) Suppose you decide to do a maximum likelihood estimation of w. You do the math and figure out that you need w to satisfy one of the following equations. Which one?

A. $\sum_i x_i \log(wx_i) = \sum_i x_i y_i \log(wx_i)$

B. $\sum_i x_i y_i = \sum_i x_i y_i \log(wx_i)$

C. $\sum_i x_i y_i = \sum_i x_i \log(wx_i)$

D. $\sum_i y_i = \sum_i \log(wx_i)$

Answer: D.

Very similar with the problem 4 in our homework2, we perform Maximum Likelihood estimation.

$$y_i \sim N(log(wx_i), 1)$$

We could write the log likelihood as:

$$LL = log(\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(y_i - log(wx_i))^2}{2\sigma^2})) = \sum_{i=1}^{n} log(\frac{1}{\sqrt{2\pi}} exp(-\frac{(y_i - log(wx_i))^2}{2}))$$

$$\frac{\partial LL}{\partial w} = 0 \Rightarrow \frac{\partial \sum_{i=1}^{n} (y_i - log(wx_i))^2}{\partial w} = 0$$

$$\Rightarrow \sum_{i=1}^{n} \frac{x_i}{wx_i} * (y_i - log(wx_i)) = 0 \Rightarrow \sum_{i=1}^{n} y_i = \sum_{i=1}^{n} log(wx_i)$$

3 Decision Trees (11 points)

For this question, you're going to answer a couple questions regarding the dataset shown below. You'll be trying to determine whether Andrew finds a particular type of food appealing based on the food's temperature, taste, and size.

Appealing	Temperature	Taste	Size
No	Hot	Salty	Small
No	Cold	Sweet	Large
No	Cold	Sweet	Large
Yes	Cold	Sour	Small
Yes	H	Sour	Small
No	H	Salty	Large
Yes	H	Sour	Large
Yes	Cold	Sweet	Small
Yes	Cold	Sweet	Small
No	H	Salty	Large

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

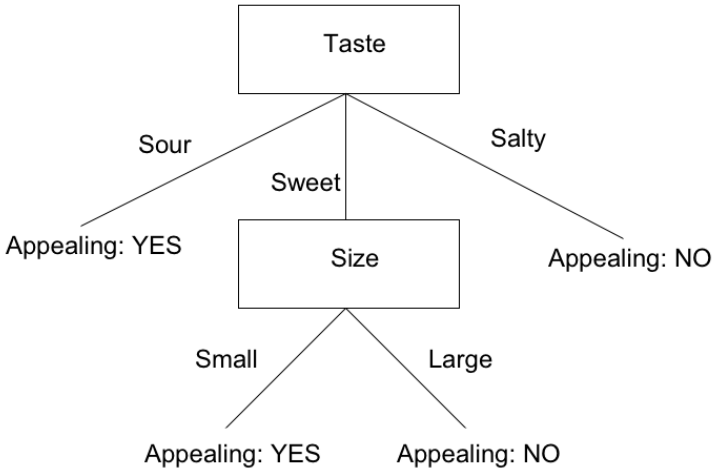
Answer: 1. $(-5/10 \cdot log(5/10) - 5/10 \cdot log(5/10))$**.**

(b) *(3 points)* Assume that *Taste* is chosen for the root of the decision tree. What is the information gain associated with this attribute?

Answer: 3/5. $(1 - 4/10 \cdot (2/4 \cdot log(2/4) + 2/4 \cdot log(2/4)) - 6/10 \cdot 0)$**.**

(c) *(5 points)* Draw the full decision tree learned for this data (without any pruning).

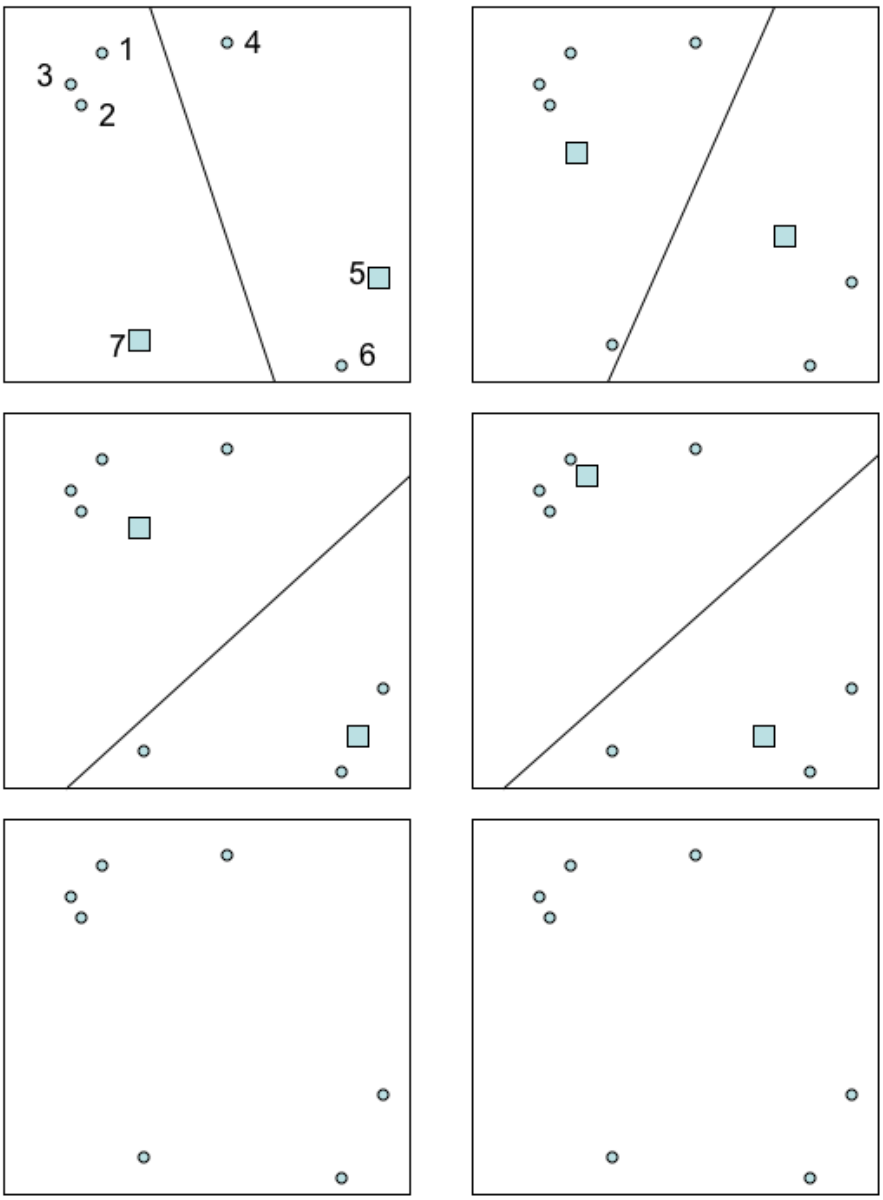
Answer:



4 K-means and Hierarchical Clustering (10 points)

(a) *(6 points)* Perform K-means on the dataset given below. Circles are data points and there are two initial cluster centers, at data points 5 and 7. Draw the cluster centers (as squares) and the decision boundaries that define each cluster. If no points belong to a particular cluster, assume its center does not change. Use as many of the pictures as you need for convergence.

Answer:



(b) *(4 points)* Give one advantage of hierarchical clustering over K-means clustering, and one advantage of K-means clustering over hierarchical clustering.

Answer: Many possibilities.

Some advantages of hierarchical clustering:

1. **Don't need to know how many clusters you're after**
2. **Can cut hierarchy at any level to get any number of clusters**
3. **Easy to interpret hierarchy for particular applications**
4. **Can deal with long stringy data**

Some advantages of K-means clustering:

1. **Can be much faster than hierarchical clustering, depending on data**
2. **Nice theoretical framework**
3. **Can incorporate new data and reform clusters easily**

5 Maximum Likelihood Estimates (9 points)

(a) *(9 points)* Suppose $X_1, \ldots, X_n$ are iid samples from $U(-w, w)$ That is,

$$p(x) = \begin{cases} 0, & x < -w \\ \frac{1}{2w}, & -w \leq x \leq w \\ 0, & x > w \end{cases}$$

Write down a formula for an MLE estimate of w.

Answer: $\hat{w} = \max(|X_1|, |X_2|, \ldots, |X_n|)$

Let $\hat{w}$ denote an MLE estimate of w. From MLE principle $\hat{w} = \arg\max_w p(X_1, \ldots, X_n|w)$.
Since $X_1, \ldots, X_n$ are iid: $\hat{w} = \arg\max_w \prod_{i=1}^n p(X_i|w)$.
Let $X_M = \max(|X_1|, |X_2|, \ldots, |X_n|)$.
If $w < X_M$, then $\prod_{i=1}^n p(X_i|w) = 0$ from the equation of $p(x)$.
Thus, $w \geq X_M$.

Given this, we have $p(X_i|w) = \frac{1}{2w}$, and thus
$\hat{w} = \arg\max_{w \geq X_M} \prod_{i=1}^n p(X_i|w)$
$= \arg\max_{w \geq X_M} \frac{1}{(2w)^n} =$
$\arg\max_{w \geq X_M} log(\frac{1}{(2w)^n}) =$
$\arg\max_{w \geq X_M} log1 - nlog(2w) =$
$\arg\max_{w \geq X_M} -nlog(2w) =$
$\arg\min_{w \geq X_M} nlog(2w) =$
$\arg\min_{w \geq X_M} log(w) =$
$\arg\min_{w \geq X_M} w =$
X_M

6 Bayes Classifiers (10 points)

Suppose we are given the following dataset, where A, B, C are input binary random variables, and y is a binary output whose value we want to predict.

A	B	C	y
0	0	1	0
0	1	0	0
1	1	0	0
0	0	1	1
1	1	1	1
1	0	0	1
1	1	0	1

(a) *(5 points)* How would a **naive** Bayes classifier predict y given this input:
$A = 0, B = 0, C = 1$. Assume that in case of a tie the classifier always prefers to predict 0 for y.

Answer: The classifier will predict 1

$P(y = 0) = 3/7; P(y = 1) = 4/7$
$P(A = 0|y = 0) = 2/3; P(B = 0|y = 0) = 1/3; P(C = 1|y = 0) = 1/3$
$P(A = 0|y = 1) = 1/4; P(B = 0|y = 1) = 1/2; P(C = 1|y = 1) = 1/2$

Predicted y maximizes $P(A = 0|y)P(B = 0|y)P(C = 1|y)P(y)$
$P(A = 0|y = 0)P(B = 0|y = 0)P(C = 1|y = 0)P(y = 0) = 0.0317$
$P(A = 0|y = 1)P(B = 0|y = 1)P(C = 1|y = 1)P(y = 1) = 0.0357$
Hence, the predicted y is 1.

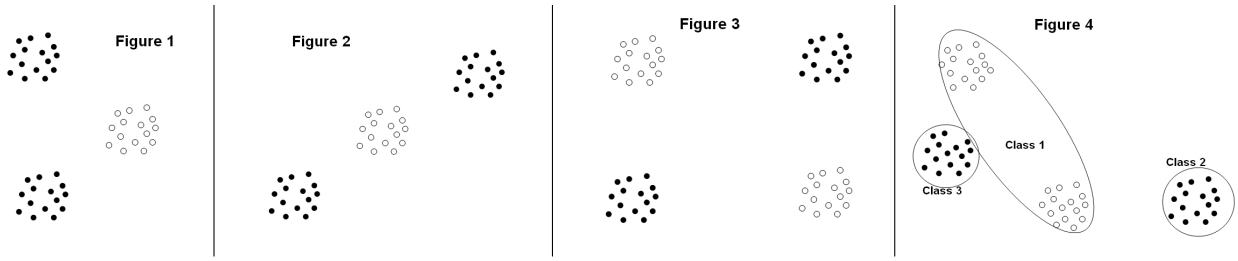
(b) *(5 points)* Suppose you know for fact that A, B, C are independent random variables. In this case is it possible for any other classifier (e.g., a decision tree or a neural net) to do better than a naive Bayes classifier? (The dataset is irrelevant for this question)

Answer: Yes

The independency of A, B, C does not imply that they are independent within each class (in other words, they are not necessarily independent when conditioned on y). Therefore, naive Bayes classifier may not be able to model the function well, while a decision tree might.
Thus, for example, $y = A$ XOR B, is an example where A, B might be independent variables, but a naive Bayes classifier will not model the function well since for a particular class (say, $y = 0$), A and B are dependent.

7 Classification (12 points)

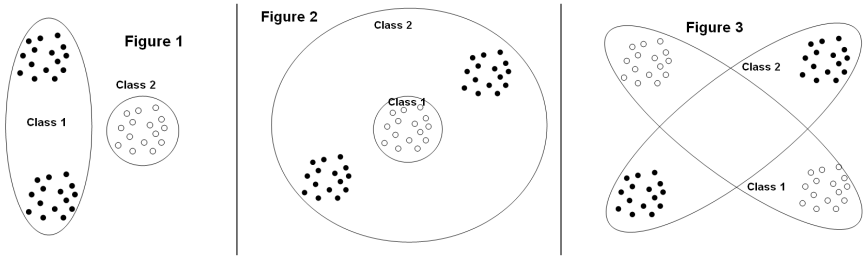
Figures 1, 2 and 3 present points from two different clusters: A (solid points) and B (hollow points). We would like to learn a classifier that achieves zero training error on this data. To do that we allow each classifier to divide the data into more than two classes, however, for each classifier there must be a subset of the classes that perfectly match class A and the complementary set of classes must match cluster B. For example, in Figure 4 classes 2 and 3 contain all of A's points and class 1 contains all of B's points and so this classification is a legitimate solution to this problem.



(a) *(6 points)* For a Gaussian Bayes classifier and for each of the three figures state the **minimum** number of classes required to achieve the above goal. For all figures you can assume equal class priors, that is P(A) = P(B).

	minimum number of classes
Figure 1	
Figure 2	
Figure 3	
Figure 4	3

Answer: The number of classes is 2 for all of the cases.



(b) *(6 points)* For the following figures, do we need a full covariance matrix for the classification or would a diagonal covariance matrix be enough

Figure 2? **Answer: Diagonal is enough. Note that the variance of the two clusters is different. A has a large variance for both the x and the y axis while B's variance is low in both direction. Thus, even though both have the same mean, the variance terms are enough to separate them.**

Figure 3? **Answer: Full is required. In this case, both the mean and the variance terms are same for both clusters. The only difference is in the covariance terms.**

8 Neural Nets and Regression (12 points)

Suppose we want to learn a quadratic model:

$$
\begin{array}{lllllllllll}
y = & w_0 & + & w_1 x_1 & + & w_2 x_2 & + & w_3 x_3 & + & \ldots & w_k x_k & + \\
 & & & w_{11} x_1^2 & + & w_{12} x_1 x_2 & + & w_{13} x_1 x_3 & + & \ldots & w_{1k} x_1 x_k & + \\
 & & & & & w_{22} x_2^2 & + & w_{23} x_2 x_3 & + & \ldots & w_{2k} x_2 x_k & + \\
 & & & \vdots & & & & \vdots & & & \vdots & \\
 & & & & & & & & w_{k-1,k-1} x_{k-1}^2 & + & w_{k-1,k} x_{k-1} x_k & + \\
 & & & & & & & & & + & w_{k,k} x_k^2 &
\end{array}
$$

Suppose we have a fixed number of records and k input attributes.

(a) *(6 points)* In big-O notation what would be the computational complexity in terms of k of learning the MLE weights using matrix inversion?

Answer: $O(k^6)$

$O(k^6)$ since it is $O([\text{number of basis functions}]^3)$ to solve the normal equations, and the number of basis functions is $\frac{1}{2}(k+1)(k+2)$.

(b) *(6 points)* What would be the computational complexity of one iteration of gradient descent? (The "batch" gradient descent method, NOT the online method).

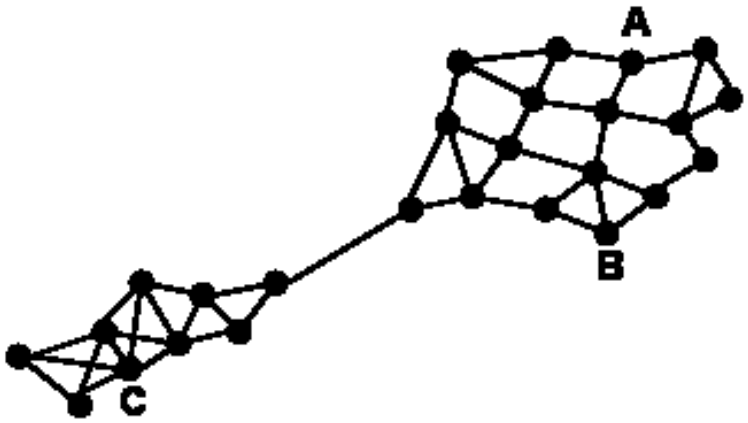
Answer: $O(k^2)$

$O(k^2)$ since work of computing δ_k for each datapoint involves $\frac{1}{2}(k+1)(k+2)$ operations and then there is one weight update for each weight.

Interesting note: If we had also included R as the number of records in the complexity then the answers are:
(a) $O(Rk^4+k^6)$, where the first term is for building an $X^T X$ matrix, and the second term is for matrix inversion.
(b) $O(Rk^2)$

9 Spectral clustering (12 points)

Consider the graph below. Let W be the distance matrix for this graph where $w_{i,j} = 1$ iff there is an edge between nodes i and j and otherwise $w_{i,j} = 0$. We will define the matrices D and P as we did in class by setting $D_{i,i} = \sum_j w_{i,j}$ and $P = D^{-1}W$. As we mentioned in class, P is the probability transition matrix for this graph. We denote by $P^t_{i,j}$ the i, j entry in the matrix P raised to the power of t.



For each of the expressions below, replace ? with either $<$, $>$ or $=$ and briefly explain your reasoning.

(a) *(3 points)* $P^{20}_{A,C}$? $P^{100}_{A,C}$

Answer: $P^{20}_{A,C} < P^{100}_{A,C}$
As the power of P increases it is more likely to transition to another cluster. Since A and C are in different clusters, it is more likely to end up in C when we take 100 steps than when we take 20 steps.

(b) *(3 points)* $P^{20}_{A,B}$? $P^{100}_{A,B}$

Answer: $P^{20}_{A,B} > P^{100}_{A,B}$
A and B are in the same cluster. It is more likely to stay in the same cluster when the power of P is low (few steps) than for higher powers of P (many steps).

(c) *(3 points)* $\sum_j P^{20}_{A,j}$? $\sum_j P^{100}_{A,j}$

Answer: $\sum_j P^{20}_{A,j} = \sum_j P^{100}_{A,j}$
P^t for any t is a probability transition matrix and so its rows always sum to 1.

(d) *(3 points)* $P^{\infty}_{B,A}$? $P^{\infty}_{B,C}$

Answer: $P^{\infty}_{B,A} < P^{\infty}_{B,C}$
At the limit, the point we end at is independent of the point we started at. Thus, we need to evaluate the (fixed) probability of ending in A vs. the probability of ending in C. In class, we have shown that this probability is proportional to the components of the first eigenvector of the symmetric matrix we

defined ($D^{-1/2}WD^{-1/2}$). In class (and in the problem set) we have derived the actual values for the entries of this vector. As we showed, these entries are the square root of the sum of the rows of W. Since in our case rows sum up to the out degree (or in degree) of the nodes, the probability that we will end up at a certain point is proportional to the connectivity of that point. Since C is connected to 5 other nodes whereas A is only connected to 3, $P^{\infty}_{B,A} < P^{\infty}_{B,C}$.

Solution to 10-701/15-781 Midterm Exam
Fall 2004

1 Introductory Probability and Statistics (12 points)

(a) *(2 points)* If A and B are disjoint events, and $Pr(B) > 0$, what is the value of $Pr(A|B)$?
Answer: 0 (Note that: $A \wedge B = \emptyset$)

(b) *(2 points)* Suppose that the p.d.f of a random variable X is as follows:

$$f(x) = \begin{cases} \frac{4}{3}(1-x^3), & \text{for } 0 \leq x \leq 1 \\ 0, & \text{otherwise} \end{cases}$$

Then $Pr(X < 0) =$?
Answer: 0 (Note that: $0 \leq x \leq 1$)

(c) *(4 points)* Suppose that X is a random variable for which $E(X) = \mu$ and $Var(X) = \sigma^2$, and let c be an arbitrary constant. Which **one** of these statements is true:

A. $E[(X-c)^2] = (\mu - c)^2 + \sigma^2$ D. $E[(X-c)^2] = (\mu - c)^2 + 2\sigma^2$

B. $E[(X-c)^2] = (\mu - c)^2$ E. $E[(X-c)^2] = \mu^2 + c^2 + 2\sigma^2$

C. $E[(X-c)^2] = (\mu - c)^2 - \sigma^2$ F. $E[(X-c)^2] = \mu^2 + c^2 - 2\sigma^2$

Answer: A

$$E[(X-c)^2] = E[X^2] - 2cE[X] + c^2 = Var(X) + [E(X)]^2 - 2c\mu + c^2 = (\mu - c)^2 + \sigma^2$$

(d) *(4 points)* Suppose that k events $B_1, B_2, ..., B_k$ form a partition of the sample space S. For $i = 1, ..., k$, let $Pr(B_i)$ denote the prior probability of B_i. There is another event A that $Pr(A) > 0$. Let $Pr(B_i|A)$ denote the posterior probability of B_i given that the event A has occurred.
Prove that if $Pr(B_1|A) < Pr(B_1)$, then $Pr(B_i|A) > Pr(B_i)$ for at least one value of i $(i = 2, ..., k)$.
(Hint: one or more of these tricks might help: $P(B_i|A)P(A) = P(B_i \wedge A)$, $\sum_{i=1}^{k} P(B_i) = 1$, $\sum_{i=1}^{k} P(B_i|A) = 1$, $P(B_i \wedge A) + P(B_i \wedge \neg A) = P(B_i)$, $\sum_{i=1}^{k} P(B_i \wedge A) = P(A)$)

Answer: We need to prove that if $Pr(B_1|A) < Pr(B_1)$, then $Pr(B_i|A) > Pr(B_i)$ for at least one value of i $(i = 2, ..., k)$.

Proof: We know that $\sum_{i=1}^{k} Pr(B_i) = 1$ and $\sum_{i=1}^{k} Pr(B_i|A) = 1$,
Suppose that for all i $(i = 2, ..., k)$, we have $Pr(B_i|A) \leq Pr(B_i)$, then we can get that
$\sum_{i=1}^{k} Pr(B_i) = Pr(B_1) + \sum_{i=2}^{k} Pr(B_i)$
$> Pr(B_1|A) + \sum_{i=2}^{k} Pr(B_i) > Pr(B_1|A) + \sum_{i=2}^{k} Pr(B_i|A)$
So we get that $1 > 1$. Confliction!.

2 Linear Regression (12 points)

We have a dataset with R records in which the i^{th} record has one real-valued input attribute x_i and one real-valued output attribute y_i.

(a) *(6 points)* First, we use a linear regression method to model this data. To test our linear regressor, we choose at random some data records to be a training set, and choose at random some of the remaining records to be a test set.
Now let us increase the training set size gradually. As the training set size increases, what do you expect will happen with the mean training and mean testing errors? (No explanation required)

- Mean Training Error: A. Increase; B. Decrease

- Mean Testing Error: A. Increase; B. Decrease

Answer:

The training error tends to increase. As more examples have to be fitted, it becomes harder to 'hit', or even come close, to all of them.

The test error tends to decrease. As we take into account more examples when training, we have more information, and can come up with a model that better resembles the true behavior. More training examples lead to better generalization.

(b) *(6 points)* Now we change to use the following model to fit the data. The model has one unknown parameter w to be learned from data.

$$y_i \sim N(log(wx_i), 1)$$

Note that the variance is known and equal to one. (no explanation required) Suppose you decide to do a maximum likelihood estimation of w. You do the math and figure out that you need w to satisfy one of the following equations. Which one?

A. $\sum_i x_i \log(wx_i) = \sum_i x_i y_i \log(wx_i)$

B. $\sum_i x_i y_i = \sum_i x_i y_i \log(wx_i)$

C. $\sum_i x_i y_i = \sum_i x_i \log(wx_i)$

D. $\sum_i y_i = \sum_i \log(wx_i)$

Answer: D.

Very similar with the problem 4 in our homework2, we perform Maximum Likelihood estimation.

$$y_i \sim N(log(wx_i), 1)$$

We could write the log likelihood as:

$$LL = log(\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(y_i - log(wx_i))^2}{2\sigma^2})) = \sum_{i=1}^{n} log(\frac{1}{\sqrt{2\pi}} exp(-\frac{(y_i - log(wx_i))^2}{2}))$$

$$\frac{\partial LL}{\partial w} = 0 \Rightarrow \frac{\partial \sum_{i=1}^{n} (y_i - log(wx_i))^2}{\partial w} = 0$$

$$\Rightarrow \sum_{i=1}^{n} \frac{x_i}{wx_i} * (y_i - log(wx_i)) = 0 \Rightarrow \sum_{i=1}^{n} y_i = \sum_{i=1}^{n} log(wx_i)$$

3 Decision Trees (11 points)

For this question, you're going to answer a couple questions regarding the dataset shown below. You'll be trying to determine whether Andrew finds a particular type of food appealing based on the food's temperature, taste, and size.

Appealing	Temperature	Taste	Size
No	Hot	Salty	Small
No	Cold	Sweet	Large
No	Cold	Sweet	Large
Yes	Cold	Sour	Small
Yes	H	Sour	Small
No	H	Salty	Large
Yes	H	Sour	Large
Yes	Cold	Sweet	Small
Yes	Cold	Sweet	Small
No	H	Salty	Large

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

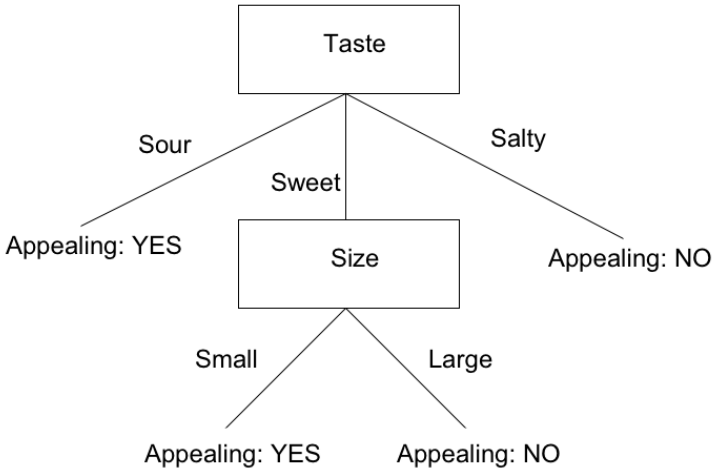
Answer: 1. $(-5/10 \cdot log(5/10) - 5/10 \cdot log(5/10))$**.**

(b) *(3 points)* Assume that *Taste* is chosen for the root of the decision tree. What is the information gain associated with this attribute?

Answer: 3/5. $(1 - 4/10 \cdot (2/4 \cdot log(2/4) + 2/4 \cdot log(2/4)) - 6/10 \cdot 0)$**.**

(c) *(5 points)* Draw the full decision tree learned for this data (without any pruning).

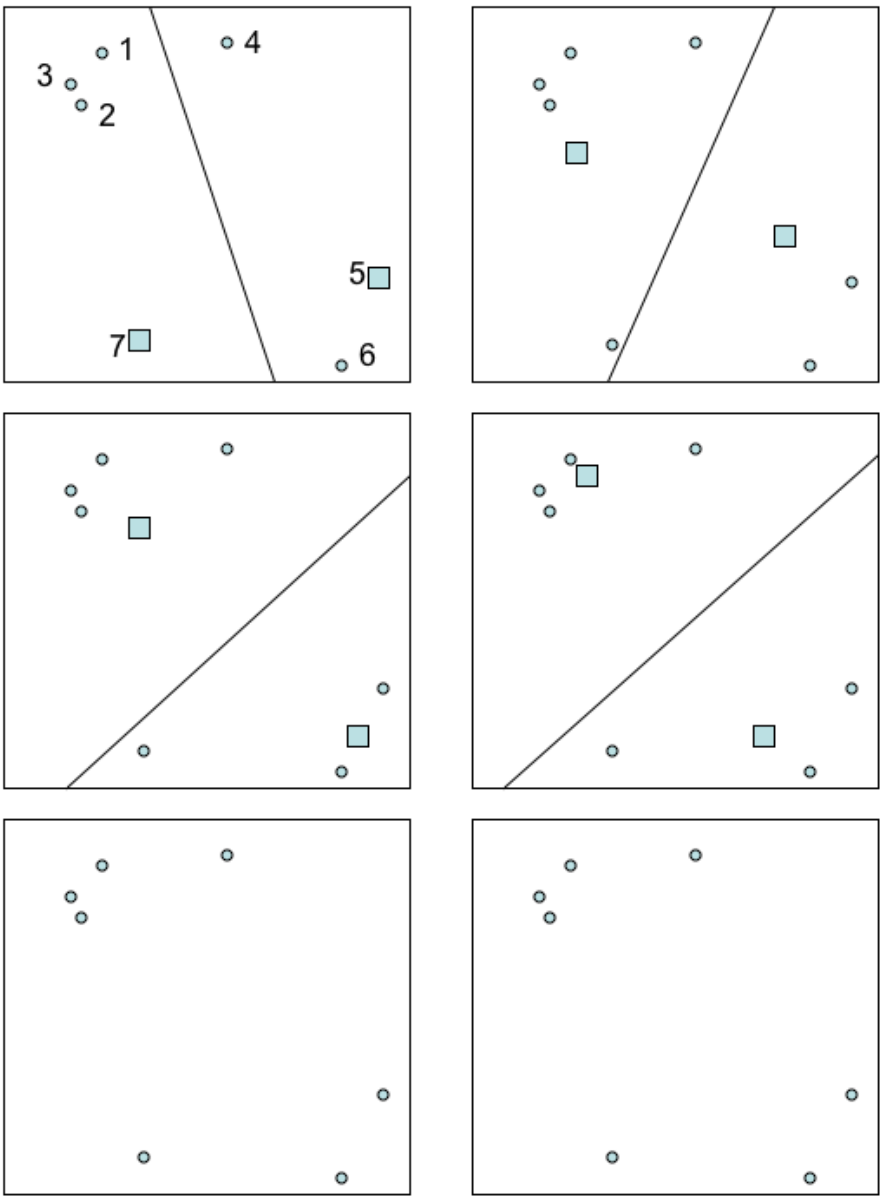
Answer:



4 K-means and Hierarchical Clustering (10 points)

(a) *(6 points)* Perform K-means on the dataset given below. Circles are data points and there are two initial cluster centers, at data points 5 and 7. Draw the cluster centers (as squares) and the decision boundaries that define each cluster. If no points belong to a particular cluster, assume its center does not change. Use as many of the pictures as you need for convergence.

Answer:



(b) *(4 points)* Give one advantage of hierarchical clustering over K-means clustering, and one advantage of K-means clustering over hierarchical clustering.

Answer: Many possibilities.

Some advantages of hierarchical clustering:

1. **Don't need to know how many clusters you're after**
2. **Can cut hierarchy at any level to get any number of clusters**
3. **Easy to interpret hierarchy for particular applications**
4. **Can deal with long stringy data**

Some advantages of K-means clustering:

1. **Can be much faster than hierarchical clustering, depending on data**
2. **Nice theoretical framework**
3. **Can incorporate new data and reform clusters easily**

5 Maximum Likelihood Estimates (9 points)

(a) *(9 points)* Suppose $X_1, \ldots, X_n$ are iid samples from $U(-w, w)$ That is,

$$p(x) = \begin{cases} 0, & x < -w \\ \frac{1}{2w}, & -w \leq x \leq w \\ 0, & x > w \end{cases}$$

Write down a formula for an MLE estimate of w.

Answer: $\hat{w} = \max(|X_1|, |X_2|, \ldots, |X_n|)$

Let $\hat{w}$ denote an MLE estimate of w. From MLE principle $\hat{w} = \arg\max_w p(X_1, \ldots, X_n|w)$.
Since $X_1, \ldots, X_n$ are iid: $\hat{w} = \arg\max_w \prod_{i=1}^n p(X_i|w)$.
Let $X_M = \max(|X_1|, |X_2|, \ldots, |X_n|)$.
If $w < X_M$, then $\prod_{i=1}^n p(X_i|w) = 0$ from the equation of $p(x)$.
Thus, $w \geq X_M$.

Given this, we have $p(X_i|w) = \frac{1}{2w}$, and thus
$\hat{w} = \arg\max_{w \geq X_M} \prod_{i=1}^n p(X_i|w)$
$= \arg\max_{w \geq X_M} \frac{1}{(2w)^n} =$
$\arg\max_{w \geq X_M} log(\frac{1}{(2w)^n}) =$
$\arg\max_{w \geq X_M} log1 - nlog(2w) =$
$\arg\max_{w \geq X_M} -nlog(2w) =$
$\arg\min_{w \geq X_M} nlog(2w) =$
$\arg\min_{w \geq X_M} log(w) =$
$\arg\min_{w \geq X_M} w =$
X_M

6 Bayes Classifiers (10 points)

Suppose we are given the following dataset, where A, B, C are input binary random variables, and y is a binary output whose value we want to predict.

A	B	C	y
0	0	1	0
0	1	0	0
1	1	0	0
0	0	1	1
1	1	1	1
1	0	0	1
1	1	0	1

(a) *(5 points)* How would a **naive** Bayes classifier predict y given this input:
$A = 0, B = 0, C = 1$. Assume that in case of a tie the classifier always prefers to predict 0 for y.

Answer: The classifier will predict 1

$P(y = 0) = 3/7; P(y = 1) = 4/7$
$P(A = 0|y = 0) = 2/3; P(B = 0|y = 0) = 1/3; P(C = 1|y = 0) = 1/3$
$P(A = 0|y = 1) = 1/4; P(B = 0|y = 1) = 1/2; P(C = 1|y = 1) = 1/2$

Predicted y maximizes $P(A = 0|y)P(B = 0|y)P(C = 1|y)P(y)$
$P(A = 0|y = 0)P(B = 0|y = 0)P(C = 1|y = 0)P(y = 0) = 0.0317$
$P(A = 0|y = 1)P(B = 0|y = 1)P(C = 1|y = 1)P(y = 1) = 0.0357$
Hence, the predicted y is 1.

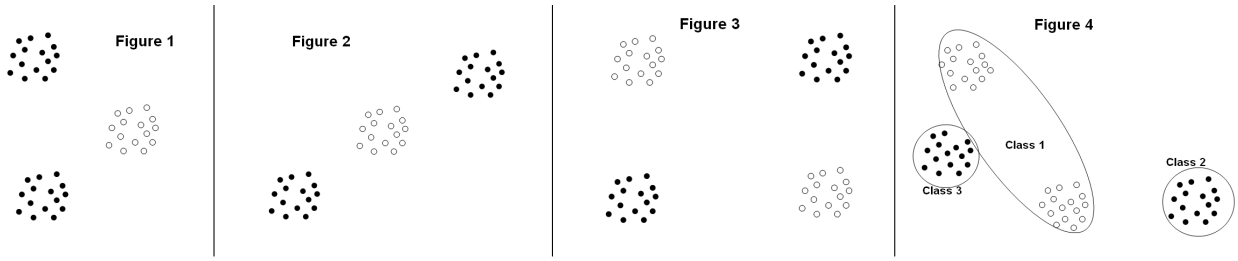
(b) *(5 points)* Suppose you know for fact that A, B, C are independent random variables. In this case is it possible for any other classifier (e.g., a decision tree or a neural net) to do better than a naive Bayes classifier? (The dataset is irrelevant for this question)

Answer: Yes

The independency of A, B, C does not imply that they are independent within each class (in other words, they are not necessarily independent when conditioned on y). Therefore, naive Bayes classifier may not be able to model the function well, while a decision tree might.
Thus, for example, $y = A$ XOR B, is an example where A, B might be independent variables, but a naive Bayes classifier will not model the function well since for a particular class (say, $y = 0$), A and B are dependent.

7 Classification (12 points)

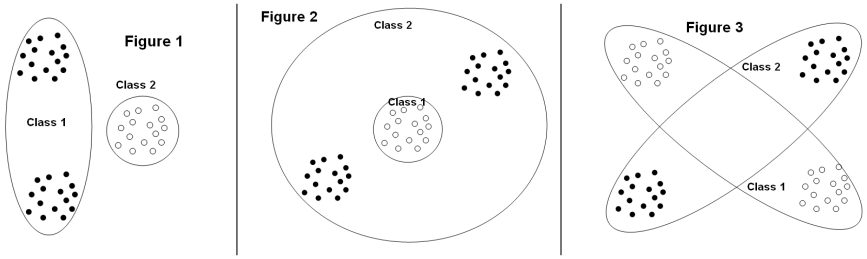
Figures 1, 2 and 3 present points from two different clusters: A (solid points) and B (hollow points). We would like to learn a classifier that achieves zero training error on this data. To do that we allow each classifier to divide the data into more than two classes, however, for each classifier there must be a subset of the classes that perfectly match class A and the complementary set of classes must match cluster B. For example, in Figure 4 classes 2 and 3 contain all of A's points and class 1 contains all of B's points and so this classification is a legitimate solution to this problem.



(a) *(6 points)* For a Gaussian Bayes classifier and for each of the three figures state the **minimum** number of classes required to achieve the above goal. For all figures you can assume equal class priors, that is P(A) = P(B).

	minimum number of classes
Figure 1	
Figure 2	
Figure 3	
Figure 4	3

Answer: The number of classes is 2 for all of the cases.



(b) *(6 points)* For the following figures, do we need a full covariance matrix for the classification or would a diagonal covariance matrix be enough

Figure 2? **Answer: Diagonal is enough. Note that the variance of the two clusters is different. A has a large variance for both the x and the y axis while B's variance is low in both direction. Thus, even though both have the same mean, the variance terms are enough to separate them.**

Figure 3? **Answer: Full is required. In this case, both the mean and the variance terms are same for both clusters. The only difference is in the covariance terms.**

8 Neural Nets and Regression (12 points)

Suppose we want to learn a quadratic model:

$$
\begin{array}{lllllllllll}
y = & w_0 & + & w_1 x_1 & + & w_2 x_2 & + & w_3 x_3 & + & \ldots & w_k x_k & + \\
 & & & w_{11} x_1^2 & + & w_{12} x_1 x_2 & + & w_{13} x_1 x_3 & + & \ldots & w_{1k} x_1 x_k & + \\
 & & & & & w_{22} x_2^2 & + & w_{23} x_2 x_3 & + & \ldots & w_{2k} x_2 x_k & + \\
 & & & \vdots & & & & \vdots & & & \vdots & \\
 & & & & & & & & w_{k-1,k-1} x_{k-1}^2 & + & w_{k-1,k} x_{k-1} x_k & + \\
 & & & & & & & & & + & w_{k,k} x_k^2 &
\end{array}
$$

Suppose we have a fixed number of records and k input attributes.

(a) *(6 points)* In big-O notation what would be the computational complexity in terms of k of learning the MLE weights using matrix inversion?

Answer: $O(k^6)$

$O(k^6)$ since it is $O([\text{number of basis functions}]^3)$ to solve the normal equations, and the number of basis functions is $\frac{1}{2}(k+1)(k+2)$.

(b) *(6 points)* What would be the computational complexity of one iteration of gradient descent? (The "batch" gradient descent method, NOT the online method).

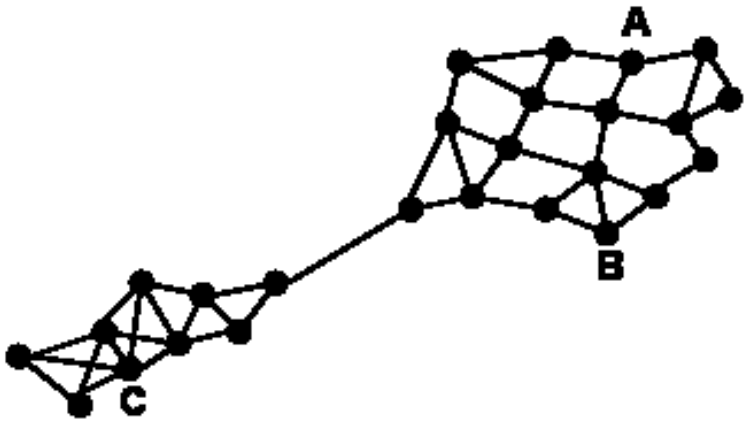
Answer: $O(k^2)$

$O(k^2)$ since work of computing δ_k for each datapoint involves $\frac{1}{2}(k+1)(k+2)$ operations and then there is one weight update for each weight.

Interesting note: If we had also included R as the number of records in the complexity then the answers are:
(a) $O(Rk^4+k^6)$, where the first term is for building an $X^T X$ matrix, and the second term is for matrix inversion.
(b) $O(Rk^2)$

9 Spectral clustering (12 points)

Consider the graph below. Let W be the distance matrix for this graph where $w_{i,j} = 1$ iff there is an edge between nodes i and j and otherwise $w_{i,j} = 0$. We will define the matrices D and P as we did in class by setting $D_{i,i} = \sum_j w_{i,j}$ and $P = D^{-1}W$. As we mentioned in class, P is the probability transition matrix for this graph. We denote by $P^t_{i,j}$ the i, j entry in the matrix P raised to the power of t.



For each of the expressions below, replace ? with either $<$, $>$ or $=$ and briefly explain your reasoning.

(a) *(3 points)* $P^{20}_{A,C}$? $P^{100}_{A,C}$

Answer: $P^{20}_{A,C} < P^{100}_{A,C}$
As the power of P increases it is more likely to transition to another cluster. Since A and C are in different clusters, it is more likely to end up in C when we take 100 steps than when we take 20 steps.

(b) *(3 points)* $P^{20}_{A,B}$? $P^{100}_{A,B}$

Answer: $P^{20}_{A,B} > P^{100}_{A,B}$
A and B are in the same cluster. It is more likely to stay in the same cluster when the power of P is low (few steps) than for higher powers of P (many steps).

(c) *(3 points)* $\sum_j P^{20}_{A,j}$? $\sum_j P^{100}_{A,j}$

Answer: $\sum_j P^{20}_{A,j} = \sum_j P^{100}_{A,j}$
P^t for any t is a probability transition matrix and so its rows always sum to 1.

(d) *(3 points)* $P^{\infty}_{B,A}$? $P^{\infty}_{B,C}$

Answer: $P^{\infty}_{B,A} < P^{\infty}_{B,C}$
At the limit, the point we end at is independent of the point we started at. Thus, we need to evaluate the (fixed) probability of ending in A vs. the probability of ending in C. In class, we have shown that this probability is proportional to the components of the first eigenvector of the symmetric matrix we

defined ($D^{-1/2}WD^{-1/2}$). In class (and in the problem set) we have derived the actual values for the entries of this vector. As we showed, these entries are the square root of the sum of the rows of W. Since in our case rows sum up to the out degree (or in degree) of the nodes, the probability that we will end up at a certain point is proportional to the connectivity of that point. Since C is connected to 5 other nodes whereas A is only connected to 3, $P^{\infty}_{B,A} < P^{\infty}_{B,C}$.

Q1 Probability and MLE [20 pts]

1. (a) Suppose we wish to calculate $P(H|E_1, E_2)$ and we have no conditional independence information. Which of the following sets of numbers are sufficient for the calculation ?

   i. $P(E_1, E_2), P(H), P(E_1|H), P(E_2|H)$
   ii. $P(E_1, E_2), P(H), P(E_1, E_2|H)$
   iii. $P(H), P(E_1|H), P(E_2|H)$

   Bayes' Rule: $P(H|E_1,E_2) = \frac{P(E_1,E_2|H)P(H)}{P(E_1,E_2)}$

   (b) Suppose we know that $P(E_1|H, E_2) = P(E_1|H)$ for all values of H, E_1, E_2. Now which of the above three sets are sufficient ?

   (i) because $P(E_1,E_2|H) = P(E_1|H)P(E_2|H)$
   (ii) it just ignores the given independence relations.

2. Which of the following statements are true ? If none of them are true, write NONE.

   (a) If X and Y are independent then $E[2XY] = 2E[X]E[Y]$ and $Var[X + 2Y] = Var[X] + Var[Y]$.
   ~~$Var[2XY] = Var[X] + Var[Y]$~~

   (b) If X and Y are independent and $X > 1$ then $Var[X+2Y^2] = Var[X]+4Var[Y^2]$ and $E[X^2-X] \geq Var[X]$.

   (c) If X are Y are not independent then $Var[X + Y] = Var[X] + Var[Y]$.

   (d) If X and Y are independent then $E[XY^2] = E[X]E[Y]^2$ and $Var[X + Y] = Var[X] + Var[Y]$.

   (e) If X and Y are not independent and $f(X) = X^2$ then $E[f(X)Y] = E[f(X)]E[Y]$ and $Var[X + 2Y] = Var[X] + 4Var[Y]$

   (b)

OVER FOR REASONS →

3. You are playing a game with two coins. Coin 1 has a θ probability of heads. Coin 2 has a 2θ probability of heads. You flip these coins several times and record your results:

Coin	Result
1	Head
2	Tail
2	Tail
2	Tail
2	Head

   (a) What is the log-likelihood of the data given θ ?

$$L(\theta) = P(\text{data}|\theta) = P(\text{coin 1} = \text{Head})\,[P(\text{coin 2} = \text{Tail})]^3\,P(\text{coin 2} = \text{Head})$$
$$= \theta(1-2\theta)^3 2\theta = 2\theta^2(1-2\theta)^3$$
$$\ell(\theta) = \log L(\theta) = \log 2 + 2\log\theta + 3\log(1-2\theta)$$

   (b) What is the maximum likelihood estimate for θ ?

$$0 = \frac{\partial \ell(\theta)}{\partial \theta} = \frac{2}{\theta} + \frac{3(-2)}{(1-2\theta)} \Rightarrow \frac{2(1-2\theta) - 6\theta}{\theta(1-2\theta)} = 0 \Rightarrow \hat{\theta}_{MLE} = 1/5$$

reminder: $\hat{\theta}_{MLE} = \arg\max_\theta L(\theta) = \arg\max_\theta \ell(\theta)$ b/c $\log(\cdot)$ is monotone! [maximizing $L(\theta)$ directly is hard]

2. Relevant properties

$E[aX] = aE[X] \quad a \in \mathbb{R}$

$Var[aX] = a^2 Var[X] \quad a \in \mathbb{R}$

if $f(X)$ is nonlinear then $E[f(X)] \neq f(E[X])$

If X and Y are independent

$E[XY] = E[X]E[Y]$

$Var[X+Y] = Var[X] + Var[Y]$

$E[X+Y] = E[X] + E[Y]$

If X and Y are not independent

$E[XY] \neq E[X]E[Y]$

$Var[X+Y] \neq Var[X] + Var[Y]$ (cf. $X \sim N(0, \sigma^2)$, $Y = -X$)

$E[X+Y] = E[X] + E[Y]$

These properties are enough to show that (a), (c), (d), (e) are false.

For (b) $E[X^2 - X] = E[X^2] - E[X]$

$Var[X] \triangleq E[(X - E[X])^2]$

$= E[X^2 - 2E[X]X + E[X]^2]$

$= E[X^2] - E[2E[X]X] + E[E[X]^2]$

$= E[X^2] - 2E[X]E[X] + E[X]^2$

$= E[X^2] - E[X]^2$

But since $X \geq 1$ $E[X]^2 \geq E[X]$ and so

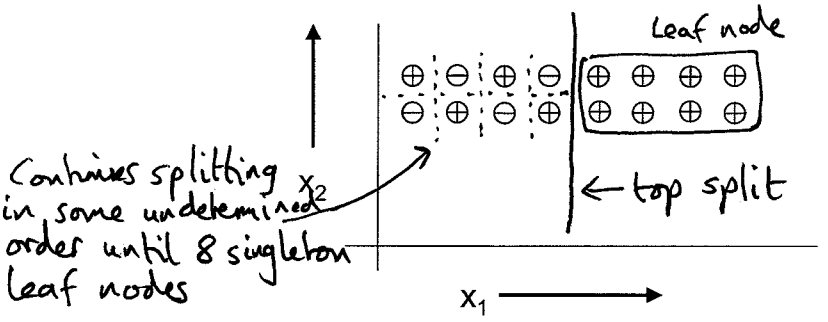
$$E[X^2] - E[X] \geq E[X^2] - E[X]^2$$

$$\| \qquad\qquad \| $$

$$E[X^2 - X] \geq Var[X]$$

Q2 Decision Trees [20 pts]

1. The figure below shows a dataset with two inputs X_1 and X_2 and one output Y, which can take on the values positive (+) or negative (-). There are 16 datapoints: 12 are positive and 4 are negative.



Answer to (a)

Assume we are testing two extreme decision tree learning algorithms. Algorithm OVERFIT builds a decision tree in the standard fashion, but never prunes. Algorithm UNDERFIT refuses to risk splitting at all, and so the entire decision tree is just one leaf node.

(a) Exactly how many leaf-nodes will be in the decision tree learned by OVERFIT on this data?

9 (see picture above)

(b) What is the leave-one-out classification error of using OVERFIT on our dataset? Report the total number of misclassifications.

Every point in the left half will be misclassified because it will be in a singleton leaf node owned by the opposing class. Every point on right will be fine. Answer = 8

(c) What is the leave-one-out classification error of using UNDERFIT on our dataset? Report the total number of misclassifications.

In all 16 folds, the ⊕ will be the majority class, so only errors will be on ⊖. There are 4 ⊖ nodes. Ans = 4

(d) Now, suppose we are learning a decision tree from a dataset with M binary-valued inputs and R training points. What is the maximum possible number of leaves in the decision tree. Circle one of the following answers:

If $R < 2^M$ then largest tree has a single point at each leaf, i.e. R leaves.

If $R \geq 2^M$ then splitting must stop after all M attributes have been tested. That makes 2^M leaves.

$R, \log_2(R), R^2, 2^R, M, \log_2(M), M^2, 2^M,$

$\min(R, M), \min(R, \log_2(M)), \min(R, M^2), \min(R, 2^M),$

$\min(\log_2(R), M), \min(\log_2(R), \log_2(M)), \min(\log_2(R), M^2), \min(\log_2(R), 2^M),$

$\min(R^2, M), \min(R^2, \log_2(M)), \min(R^2, M^2), \min(R^2, 2^M),$

$\min(2^R, M), \min(2^R, \log_2(M)), \min(2^R, M^2), \min(2^R, 2^M),$

$\max(R, M), \max(R, \log_2(M)), \max(R, M^2), \max(R, 2^M),$

$\max(\log_2(R), M), \max(\log_2(R), \log_2(M)), \max(\log_2(R), M^2), \max(\log_2(R), 2^M),$

$\max(R^2, M), \max(R^2, \log_2(M)), \max(R^2, M^2), \max(R^2, 2^M),$

$\max(2^R, M), \max(2^R, \log_2(M)), \max(2^R, M^2), \max(2^R, 2^M)$

Thus answer = $\min(R, 2^M)$

Q3

Linear Regression

Consider fitting the linear regression model for these data

x	-1	0	2
y	1	-1	1

(b) Fit $Y_i = \beta_0 + \epsilon_i$ (degenerated linear regression), find β_0.

β_0 =argmin $\sum(Y_i - \beta_0)^2$

$\beta_0 = 1/3$

(b) Fit $Y_i = \beta_1 X_i + \epsilon_i$ (linear regression without the constant term), find β_0 and β_1.

β_1 =argmin $\sum(Y_i - \beta_1 X_i)^2$

$\beta_1 = \sum X_i Y_i / \sum X_i^2 = 1/5$

Q4 Conditional Independence [5 pts]

1. Consider the following joint distribution over the random variables A, B, and C.

A	B	C	P(A,B,C)
0	0	0	1/8
0	1	0	1/8
0	0	1	1/8
0	1	1	1/8
1	0	0	1/8
1	1	0	1/8
1	0	1	1/8
1	1	1	1/8

(a) True or False: A is conditionally independent of B given C.

True, because $\forall_{ijk}\ P(A=i|B=j,C=k) = P(A=i|C=k)$

(b) If you answered part (a) with TRUE, make a change to the top two rows of this table to create a joint distribution in which the answer to (a) is FALSE.

If you answered part (a) with FALSE, make a change to the top two rows of this table to create a joint distribution in which the answer to (a) is TRUE.

One possible change is

A	B	C	P(ABC)
0	0	0	0
0	1	0	1/4

note any change made to these two rows must still result in the table representing a joint probability distribution whose probabilities sum to one.

Q5 Generative vs Discriminative Classifiers [15 pts]

1. You wish to train a classifier to predict the gender (a boolean variable, G) of a person based on that person's weight (a continuous variable, W) and whether or not they are a graduate student (a boolean variable, S). Assume that W and S are conditionally independent given G. Also, assume that the variance of the probability distribution $P(Weight|Gender = female)$ equals the variance for $P(Weight|Gender = male)$.

   (a) Is it reasonable to train a Naive Bayes classifier for this task?

   Yes. W and S are conditionally independent given G.

   (b) If not, explain why not, and describe how you might reformulate this problem to allow training a naive Bayes classifier. If so, list every probability distribution your classifier must learn, what form of distribution you would use for each, and give the total number of parameters your classifier must estimate from the training data.

   We must estimate 6 parameters:

   P(G) Bernoulli → $P(G=1) \equiv \pi$ (note $P(G=0)$ need not be estimated separately. It is $1 - P(G=1)$)

   P(S|G) Bernoulli → $P(S=1 \mid G=1) \equiv \theta_1$

   Bernoulli → $P(S=1 \mid G=0) \equiv \theta_0$

   P(W|G) Normal →
   - σ_W - variance for the Normal distributions governing W
   - $\mu_{W|G=1}$ - mean for $P(W|G=1)$
   - $\mu_{W|G=0}$ - mean for $P(W|G=0)$

   (c) Note one difference between the above $P(Gender|Weight, Student)$ problem and the problems we discussed in class is that the above problem involves training a classifier over a *combination* of boolean and continuous inputs. Now suppose you would like to train a discriminative classifier for this problem, to directly fit the parameters of $P(G|W, S)$, under the conditional independence assumption. Assuming that W and S are conditionally independent given G, is it correct to assume that $P(G = 1|W, S)$ can be expressed as a conventional logistic function:

$$P(G = 1|W, S) = \frac{1}{1 + \exp(w_0 + w_1 W + w_2 S)}$$

   If not, explain why not. If so, prove this.

   Yes. This can be shown by combining the derivation in Tom's Naive Bayes chapter draft (which covers the case of Normal variables) with the solution to a question from homework 2 (which covers Boolean variables).

from eq 19 in Tom's handout, using our variables G, W, + S, we have:

$$P(G=1|WS) = \frac{1}{1 + \exp\left(\ln\frac{1-\pi}{\pi} + \ln\frac{P(W|G=0)}{P(W|G=1)} + \ln\frac{P(S|G=0)}{P(S|G=1)}\right)}$$

from Tom's handout this equals: $W\left(\frac{\mu_0 - \mu_1}{\sigma^2}\right) + \left(\frac{\mu_1^2 - \mu_0^2}{2\sigma^2}\right)$

from HW2, this is: $S \ln\frac{\theta_0(1-\theta_1)}{\theta_1(1-\theta_0)} + \ln\frac{1-\theta_0}{1-\theta_1}$

therefore:

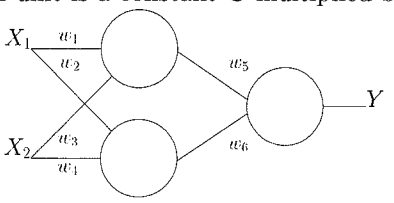
$$w_0 = \ln\frac{1-\pi}{\pi} + \ln\frac{1-\theta_0}{1-\theta_1} + \frac{\mu_1^2 - \mu_0^2}{2\sigma^2}$$

$$w_1 = \frac{\mu_0 - \mu_1}{\sigma^2}$$

$$w_2 = \ln\frac{\theta_0(1-\theta_1)}{\theta_1(1-\theta_0)}$$

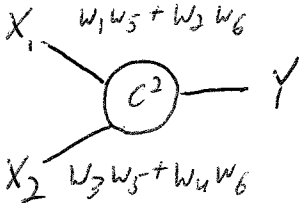
Q6 Neural Networks [20 pts]

1. For this question, suppose we have a Neural Network (shown below) with linear activation units. In other words, the output of each unit is a constant **C** multiplied by the weighted sum of inputs.



(a) Can any function that is represented by the above network also be represented by a single unit ANN (or perceptron). If so, draw the equivalent perceptron, detailing the weights and the activation function. Otherwise, explain why not.

Yes



This answer uses c^2 as the activation function.
Any answer that provided equivalent function by a correct linear combination of weights was acceptable.

(b) Can the space of functions that is represented by the above ANN also be represented by linear regression? (Yes/No)

Yes Any function in the network above has the form:

$$Y = \underbrace{c^2(w_1w_5 + w_2w_6)}_{\beta_1} X_1 + \underbrace{c^2(w_3w_5 + w_4w_6)}_{\beta_2} X_2$$

This is a linear regression on X_1, X_2 with coefficients β_1, β_2.

2. Consider the XOR function: $Y = (X_1 \wedge \neg X_2) \vee (\neg X_1 \wedge X_2)$. We can also express this as:

$$Y = \begin{cases} > \frac{1}{2} & X_1 \neq X_2 \\ < \frac{1}{2} & otherwise \end{cases}$$

It is well known that XOR cannot be implemented by a single perceptron. Draw a fully connected three unit ANN that has binary inputs X_1, X_2, 1 and output Y.

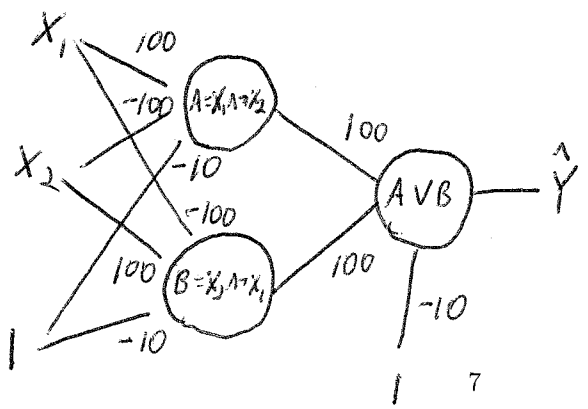
Select weights that implement $\mathbf{Y = (X_1 \text{ XOR } X_2)}$.

For this question, assume the sigmoid activation function:

$$y = \frac{1}{1 + \exp(-(w_0 + w_1x_1 + w_2x_2))}$$

We need to implement the following truth table:

X_1	X_2	Y
0	0	0
0	1	1
1	0	1
1	1	0



We use the decomposition above

$$Y = \underbrace{(X_1 \wedge \neg X_2)}_{A} \vee \underbrace{(\neg X_1 \wedge X_2)}_{B}$$

The first layer implements A & B
The last node implements "OR"

- keep in mind: weighted sum is (-) negative => sigmoid < .5 Otherwise sigmoid > .5
- If relative magnitude of weights is skewed, the output may also be skewed.

10-701 Midterm Exam, Spring 2005

1. Write your name and your email address below.

   Name:

   Email address:

2. There should be 15 numbered pages in this exam (including this cover sheet).

3. Write your name at the top of EVERY page in the exam.

4. You may use any and all books, papers, and notes that you brought to the exam, but not materials brought by nearby students. No laptops, PDAs, or Internet access.

5. If you need more room to work out your answer to a question, use the back of the page and clearly mark on the front of the page if we are to look at what's on the back.

6. Work efficiently. Some questions are easier, some more difficult. Be sure to give yourself time to answer all of the easy ones, and avoid getting bogged down in the more difficult ones before you have answered the easier ones.

7. Note there is one extra-credit question. The grade curve will be made without considering students' extra credit points. The extra credit will then be used to try to bump your grade up without affecting anyone else's grade.

8. You have 80 minutes.

9. Good luck!

Question	Number of points	Score
1. Big Picture	10	
2. Short Questions	15	
3. Learning Algorithms	16	
4. Decision Trees	16	
5. Loss Fns. and SVMs	23	
6. Learning Theory	20	
Total	100	
Extra credit		
7. Bias-Variance Trade-off	18	

1 [10 points] Big Picture

Following the example given, add 10 edges to Figure 1 relating the pair of algorithms. Each edge should be labeled with one characteristic the methods share, and one difference. These labels should be short and address basic concepts, such as types of learning problems, loss functions, and hypothesis spaces.

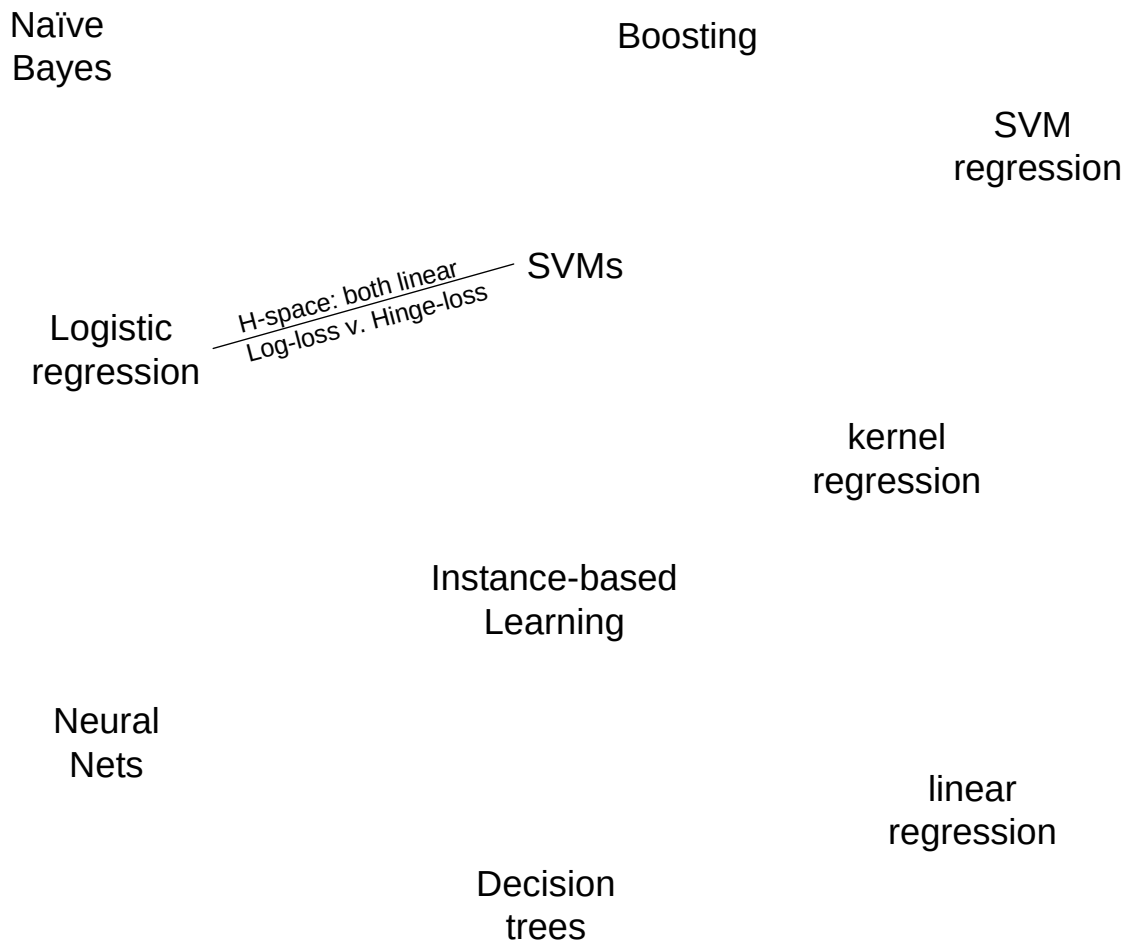


Figure 1: Big picture.

One solution is shown below, there are many others.

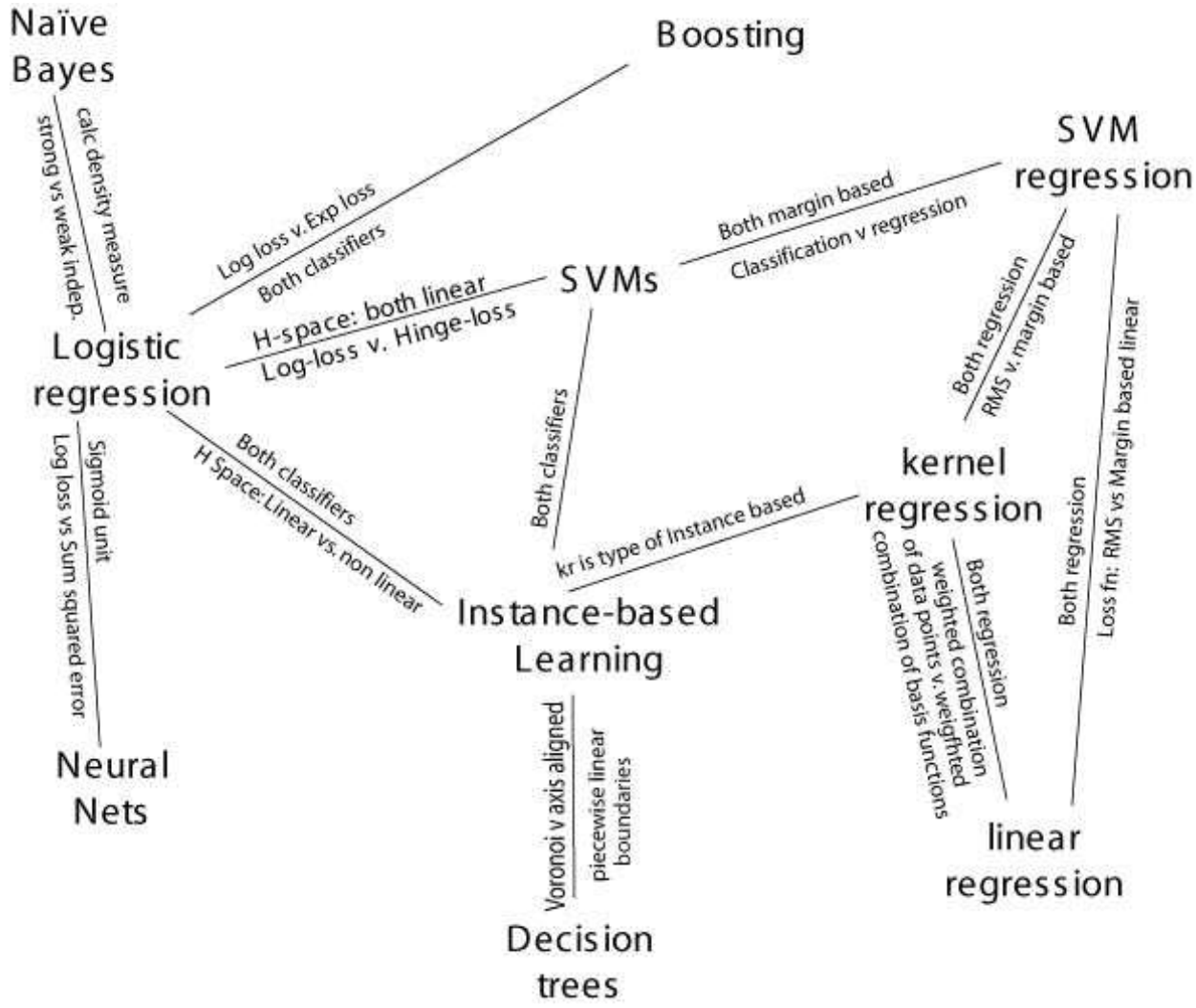


Figure 2: Big picture solutions.

2 [15 points] Short Questions

(a) [3 points] Briefly describe the difference between a *maximum likelihood* hypothesis and a *maximum a posteriori* hypothesis.

Solutions:
ML: maximize the data likelihood given the model, i.e., $\arg\max_W P(Data|W)$
MAP: $\arg\max_W P(W|Data)$

(b) [4 points] Consider a naive Bayes classifier with 3 boolean input variables, X_1, X_2 and X_3, and one boolean output, Y.

- How many parameters must be estimated to train such a naive Bayes classifier? (you need not list them unless you wish to, just give the total)

  Solutions:
  For a naive Bayes classifier, we need to estimate P(Y=1), $P(X1 = 1|y = 0), P(X2 = 1|y = 0), P(X3 = 1|y = 0), P(X1 = 1|y = 1), P(X2 = 1|y = 1), P(X3 = 1|y = 1)$. *Other probabilities can be obtained with the constraint that the probabilities sum up to 1.*
  So we need to estimate 7 parameters.

- How many parameters would have to be estimated to learn the above classifier if we do *not* make the naive Bayes conditional independence assumption?

  Solutions:
  Without the conditional independence assumption, we still need to estimate P(Y=1). For Y=1, we need to know all the enumerations of (X1,X2,X3), i.e., 2^3 *of possible (X1,X2,X3). Consider the constraint that the probabilities sum up to 1, we need to estimate* $2^3 - 1 = 7$ *parameters for Y=1.*
  Therefore the total number of parameters is $1 + 2(2^3 - 1) = 15$.

[8 points] True or False? If true, explain why in *at most two sentences*. If false, explain why or give a brief counterexample in *at most two sentences*.

- **(True or False?)** The error of a hypothesis measured over its training set provides a pessimistically biased estimate of the true error of the hypothesis.

  Solutions:
  False. The training error is optimisticly biased since it's biased while usually smaller than the true error.

- **(True or False?)** If you are given m data points, and use half for training and half for testing, the difference between training error and test error decreases as m increases.

  Solutions:
  True. As we have more and more data, training error increases and testing error decreases. And they all converge to the true error.

- **(True or False?)** Overfitting is more likely when the set of training data is small

  Solutions:
  True. With small training dataset, it's easier to find a hypothesis to fit the training data exactly,i.e., overfit.

- **(True or False?)** Overfitting is more likely when the hypothesis space is small

  Solutions:
  False. We can see this from the bias-variance trade-off. When hypothesis space is small, it's more biased with less variance. So with a small hypothesis space, it's less likely to find a hypothesis to fit the data very well,i.e., overfit.

3 [16 points] Learning Algorithms

Consider learning a target function of the form $f : \Re^2 \rightarrow \{A, B, C\}$ that is, a function with 3 discrete values defined over the 2-dimensional plane. Consider the following learning algorithms:

- Decision trees
- Logistic regression
- Support Vector Machine
- 1-nearest neighbor

Note each of these algorithms can be used to learn our target function f, though doing so might require a common extension (e.g., in the case of decision trees, we need to utilize the usual method for handling real-valued input attributes).

For each of these algorithms,

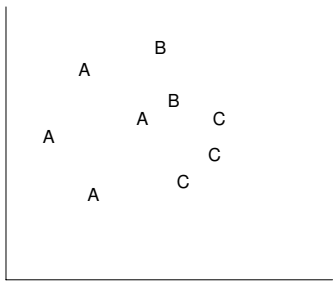
A. Describe any assumptions you are making about the variant of the algorithm you would use

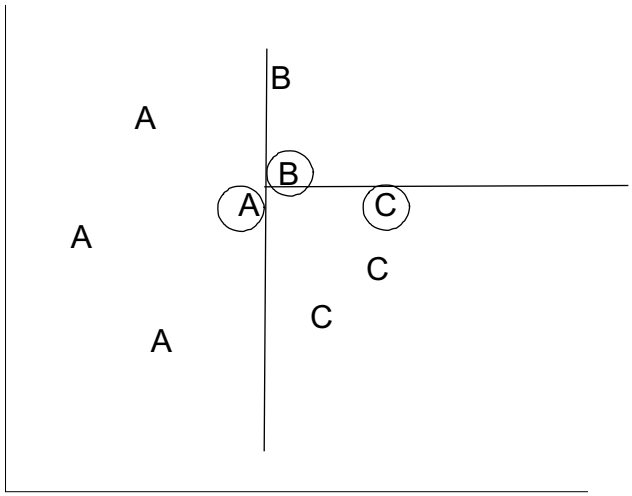
B. Draw in the decision surface that would be learned given this training data (and describing any ambiguities in your decision surface)

C. Circle any examples that would be misclassified in a leave-one-out evaluation of this algorithm with this data. That is, if you were to repeatedly train on n-1 of these examples, and use the learned classifier to label the left out example, will it be misclassified?

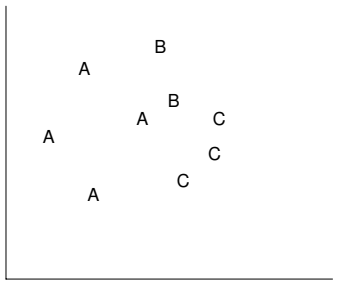
Solutions:
the assuptions are as follows:
Decision trees: Handle real valued attributes by discretizing;
Logistic regression: Handle non-binary classification;
SVM: Use one against all approach and a linear kernel;
1-NN: x-axis features and y-axis features are non-weighted.
Please see the figures on the right for decision surface and misclassified examples by leave-one-out evaluation.

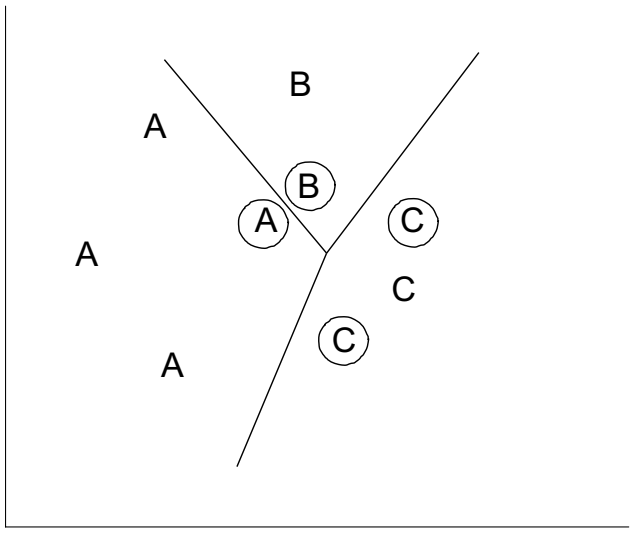
Decision trees



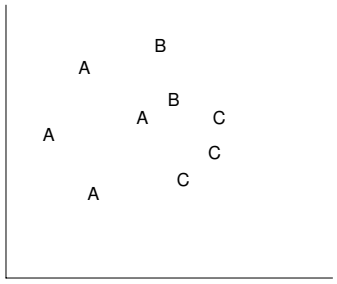


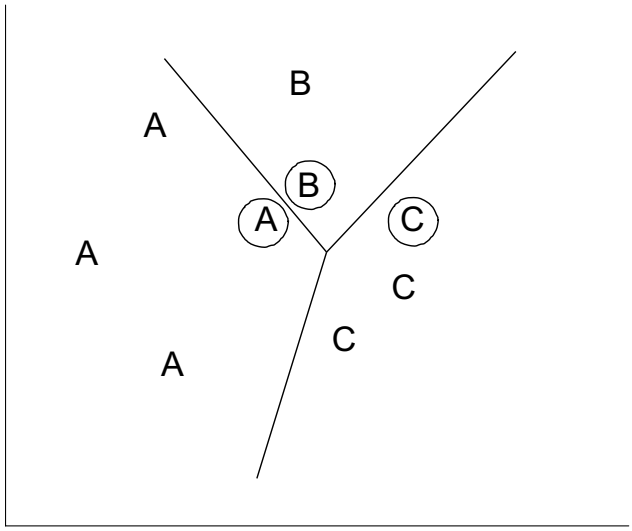
Logistic regression



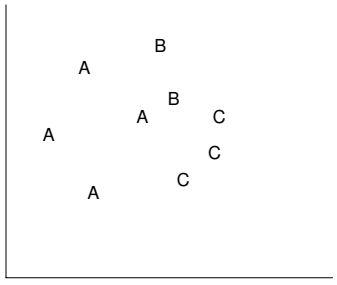


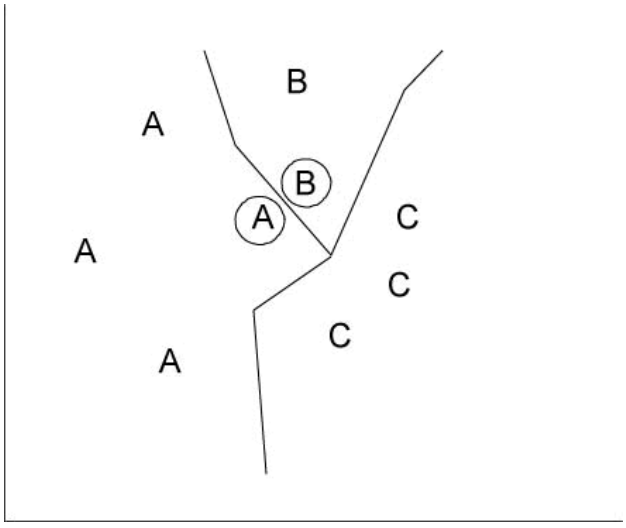
Support Vector Machine





1-nearest neighbor



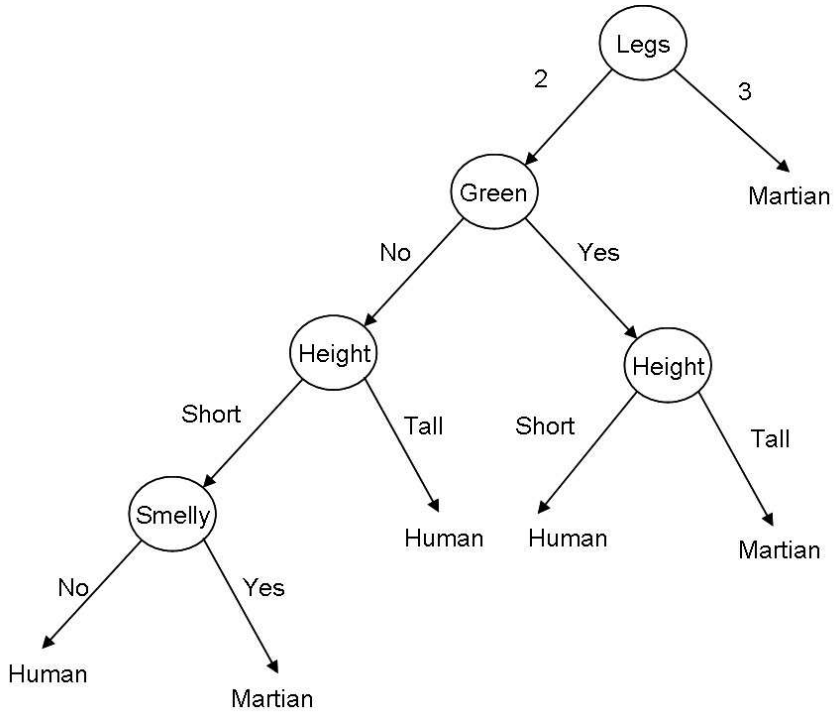


4 [16 points] Decision Trees

NASA wants to be able to discriminate between Martians (M) and Humans (H) based on the following characteristics: Green $\in \{N, Y\}$, Legs $\in \{2, 3\}$, Height $\in \{S, T\}$, Smelly $\in \{N, Y\}$. Our available training data is as follows:

	Species	Green	Legs	Height	Smelly
1)	M	N	3	S	Y
2)	M	Y	2	T	N
3)	M	Y	3	T	N
4)	M	N	2	S	Y
5)	M	Y	3	T	N
6)	H	N	2	T	Y
7)	H	N	2	S	N
8)	H	N	2	T	N
9)	H	Y	2	S	N
10)	H	N	2	T	Y

a)[8 points] Greedily learn a decision tree using the ID3 algorithm and draw the tree.
See the following figure for the ID3 decision tree:

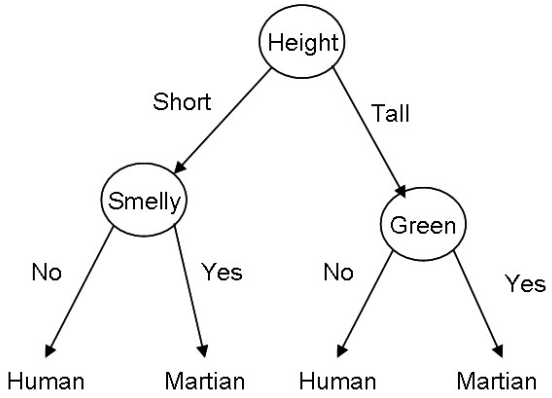


b) i) [3 points] Write the learned concept for Martian as a set of conjunctive rules (e.g., if (green=Y and legs=2 and height=T and smelly=N), then Martian; else if ... then Martian; ...; else Human).

Only the disjunction of conjunctions for Martians was required.
(Legs=3) $\vee$
(Legs=2 $\wedge$ *Green=Yes* $\wedge$ *Height=Tall)* $\vee$
(Legs=2 $\wedge$ *Green=No* $\wedge$ *Height=Short* $\wedge$ *Smelly=Yes)*

ii) [5 points] The solution of part b)i) above uses up to 4 attributes in each conjunction. Find a set of conjunctive rules using only 2 attributes per conjunction that still results in zero error in the training set. Can this simpler hypothesis be represented by a decision tree of depth 2? Justify.

We allowed a little variation on this one because the question could be interpreted as allowing conjunctions with up to two terms. In fact, only two two-term conjunctions are necessary:
(Green=Yes $\wedge$ *Height=Tall)* $\vee$ *(Smelly=Yes* $\wedge$ *Height=Short)*
These conjunctive rules share the height term, so a depth-2 tree is possible. See the figure below.



Notice how ID3 finds a tree that is much longer than the optimal tree. This is due to the greediness of the ID3 algorithm.

5 [23 points] Loss functions and support vector machines

In homework 2, you found a relationship between ridge regression and the maximum a posteriori (MAP) approximation for Bayesian learning in a particular probabilistic model. In this question, you will explore this relationship further, finally obtaining a relationship between SVM regression and MAP estimation.

(a) Ridge regression usually optimizes the squared (L_2) norm:

$$\widehat{\mathbf{w}}_{L_2} = \arg\min_{\mathbf{w}} \sum_{j=1}^{N} (t_j - \sum_i w_i h_i(x_j))^2 + \lambda \sum_i w_i^2. \tag{1}$$

The L_2 norm minimizes the squared residual $(t_j - \sum_i w_i h_i(x_j))^2$, thus significantly weighing outlier points. (An outlier is a data point that falls far away from the prediction $\sum_i w_i h_i(x_j)$.) An alternative that is less susceptible to outliers is to minimize the "sum of absolute values" (L_1) norm:

$$\widehat{\mathbf{w}}_{L_1} = \arg\min_{\mathbf{w}} \sum_{j=1}^{N} |t_j - \sum_i w_i h_i(x_j)| + \lambda \sum_i w_i^2. \tag{2}$$

(i)[2 points] Plot a sketch of the L_1 loss function, do not include the regularization term in your plot. (The x-axis should be the residual $t_j - \sum_i w_i h_i(x_j)$ and the y-axis is the loss function.)

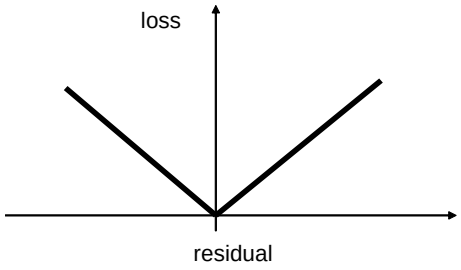


Figure 3: L_1 loss.

(ii)[2 points] Give an example of a case where outliers can hurt a learning algorithm.

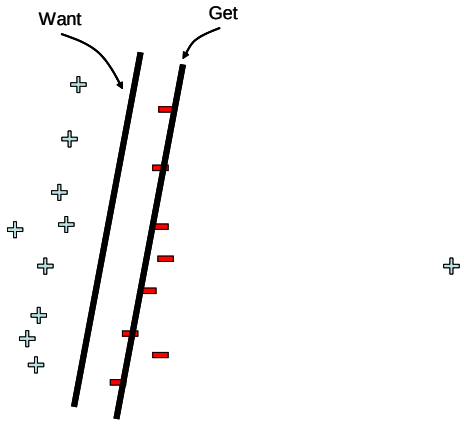


Figure 4: Outlier example.

(iii)[2 points] Why do you think L_1 is less susceptible to outliers than L_2?

L_2 penalizes the square of the residual, so an outlier with residual r will have a loss of r^2. On the other hand, L_1 will have a loss of only $|r|$. Therefore, if $|r| > 1$, this outlier will have a larger influence on the L_2 loss than L_1, and, thus, a greater effect on the solution.

(vi)[2 points] Are outliers always bad and we should always ignore them? Why? (Give one short reason for ignoring outliers, and one short reason against.)

Outliers are often "bad" data, caused by faulty sensors or errors entering values; in such cases, the outliers are not part of the function we want to learn and should be ignore. On the other hand, an outlier could be just an unlikely sample from the true distribution of the function of interest; in these cases, the data point is just another sample and should not be ignored.

(v)[4 points] As with ridge regression in Equation 1, the regularized L_1 regression in Equation 2 can also be viewed a MAP estimator. Explain why by describing the prior $P(\mathbf{w})$ and the likelihood function $P(t \mid \mathbf{x}, \mathbf{w})$ for this Bayesian learning problem. Hint: The p.d.f. of the Laplace distribution is:

$$P(x) = \frac{1}{2b} e^{-|x-\mu|/b}.$$

As with ridge regression, the prior over each parameter is zero-mean Gaussian with variance $1/\lambda$:

$$P(w_i) \sim \mathcal{N}(0; 1/\lambda).$$

The parameters have independent priors:

$$P(\mathbf{w}) = \prod_i P(w_i).$$

The likelihood function is Laplacian with mean $\mathbf{x} \cdot \mathbf{w}$:

$$P(t \mid \mathbf{x}, \mathbf{w}) = \frac{1}{2} e^{-|t-\mathbf{x}\cdot\mathbf{w}|}.$$

(b) As mentioned in class, SVM regression is a margin-based regression algorithm that takes two parameters, $\epsilon > 0$ and $C \geq 0$, as input. In SVM regression, there is no penalty for points that are within ϵ of the hyperplane. Points that are further than ϵ are penalized using the hinge loss. Formally, the SVM regression QP is:

$$\begin{array}{ll} \widehat{\mathbf{w}}_{SVM} = \min_{\mathbf{w},\xi,\bar{\xi}} & \frac{1}{2}\mathbf{w}.\mathbf{w} + C\sum_{j=1}^{m}(\xi_j + \bar{\xi}_j) \\ \text{s.t.} & t_j - \sum_i w_i h_i(x_j) \leq \epsilon + \xi_j \\ & \sum_i w_i h_i(x_j) - t_j \leq \epsilon + \bar{\xi}_j \\ & \xi_j \geq 0, \quad \bar{\xi}_j \geq 0, \quad \forall j \end{array}$$

(i)[4 points] Plot a sketch of the loss function function used by SVM regression. Again, the x-axis should be the residual $t_j - \sum_i w_i h_i(x_j)$ and the y-axis the loss function. However, do not include the $\frac{1}{2}\mathbf{w}.\mathbf{w}$ term in this plot of the loss function.

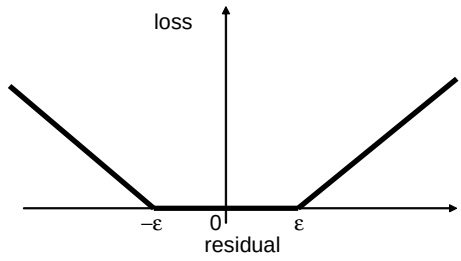


Figure 5: Margin loss.

(ii)[2 points] Compared to L_2 and L_1, how do you think SVM regression will behave in the presence of outliers?
The margin loss is very similar to L_1, so the margin loss will be less susceptible to outliers than L_2. When compared to L_1, the margin loss has an small region ($[-\epsilon, \epsilon]$) with zero penalty, thus, seemingly, margin loss should be less susceptible to outliers than L_1. However, ϵ is usually much smaller than the outlier residual, thus, L_1 and the margin loss will usually have very similar behavior.

(iii)[5 points] SVM regression can also be view as a MAP estimator. What is the prior and the likelihood function for this case?

As with ridge regression, the prior over each parameter is zero-mean Gaussian, but now with variance $2C$:

$$P(w_i) \sim \mathcal{N}(0; 2C).$$

The parameters have independent priors:

$$P(\mathbf{w}) = \prod_i P(w_i).$$

The likelihood function is constant in $[-\epsilon, \epsilon]$ and Laplacian with mean $\mathbf{x} \cdot \mathbf{w}$ elsewhere:

$$P(t \mid \mathbf{x}, \mathbf{w}) = \begin{cases} \frac{1}{2+2\epsilon}, & \text{for } |t - \mathbf{x} \cdot \mathbf{w}| \leq \epsilon; \\ \frac{1}{2+2\epsilon} e^{-(|t-\mathbf{x}\cdot\mathbf{w}|-\epsilon)}, & \text{for } |t - \mathbf{x} \cdot \mathbf{w}| > \epsilon. \end{cases}$$

6 [20 points] Learning Theory

This question asks you to consider the relationship between the VC dimension of a hypothesis space H and the number of queries the learner must make (in the worst case) to assure that it exactly learns an arbitrary target concept in H.

More precisely, we have a learner with a hypothesis space H containing hypotheses of the form $h : X \rightarrow \{0, 1\}$. The target function $c : X \rightarrow \{0, 1\}$ is one of the hypotheses in H. Training examples are generated by the learner posing a query instance $x_i \in X$, and the teacher then providing its label $c(x_i)$. The learner continues posing query instances until it has determined exactly which one of its hypothesis in H is the target concept c.

Show that in the worst case (i.e., if an adversary gets to choose $c \in H$ based on the learner's queries thus far, and wishes to maximize the number of queries), then the number of queries needed by the learner will be at least VC(H), the VC dimension of H. Put more formally, let $MinQueries(c, H)$ be the minimum number of queries needed to guarantee learning target concept c exactly, when considering hypothesis space H. We are interested in the worst case number of queries, $WorstQ(H)$, where

$$WorstQ(H) = max_{c \in H}[MinQueries(c, H)]$$

You are being asked to prove that

$$WorstQ(H) \geq VC(H)$$

You will break this down into two steps:

(a) [8 points] Consider the largest subset of instances $S \subset X$ that can be shattered by H. Show that regardless of its learning algorithm, in the worst case the learner will be forced to pose each instance $x \in S$ as a separate query.

Because S is shattered by H, there will be at least one subset $H^* \subset H$, where each $h \in H^*$ assigns one of the $2^{|S|}$ possible labelings to S. Suppose the adversary chooses a target function c such that $c \in H^*$.

The problem statement says the learner must pose queries until it determines exactly which one of its hypothesis in H is the target concept. Let us assume the learner poses fewer than $|S|$ queries. We will show the learner cannot in this case have converged to just a single consistent candidate hypothesis. Let $x_i \in S$ be one of the instances from S it has not used as a query, and let $A \subset S$ be the set of all instances from S the learner has queried. Because H^* shatters S there are at least two hypotheses $h_1 \in H^*$ and $h_2 \in H^*$ such that both h_1 and h_2 label A correctly, but for which $h_1(x_i) \neq h_2(x_i)$. Therefore, the learner will not have determined which one of the hypotheses in H (or even in H^*) is the target cocept.

(b) [5 points] Use the above to argue that $WorstQ(H) \geq VC(H)$.

We just showed in part (a) that $WorstQ(H) \geq |S|$. By definition, $VC(H) = |S|$. Therefore, $WorstQ(H) \geq VC(H)$.

(c) [7 points] Is there a better case? In other words, if the learner knows that a friend (not an adversary) will be choosing $c \in H$, and that the friend wishes to *minimize* the number of learning queries, is it possible for the friend to choose a c that allows the learner to avoid querying all of the points in S? More formally, if we define

$$BestQ(H) = min_{c \in H}[MinQueries(c, H)]$$

then is the following statement true or false?

$$BestQ(H) \geq VC(H)$$

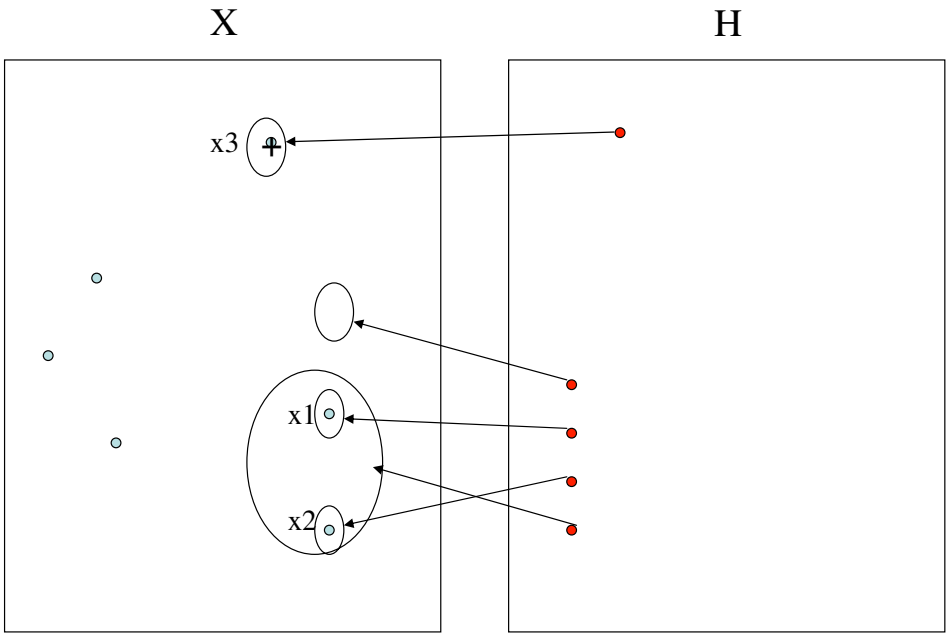
Justify your answer.

False. In fact, the answer will depend on the exact X and H, and is therefore false in general. To see why, consider the figure below, where X contains exactly 6 instances, and H contains exactly 5 hypotheses. In this diagram,the circle associated with each h indicates which members of X it labels positive. Notice VC(H)=2 in this figure, because the four hypotheses in the lower left of H shatter points $x1$ and $x2$.

Suppose here that the learner first asks for the label of $x3$, and the teacher/friend responds that the label of $x3$ is positive. There is only one hypothesis in H that labels $x3$ positive, so the learner has exactly learned the target function from one example, despite the fact that $VC(H) = 2$.

Notice an adversary could, in this case, respond that the label of $x3$ is negative, thereby forcing the learner to continue to consider the 4 hypotheses that shatter $x1$ and $x2$.

While $BestQ(H) \geq VC(H)$ does not hold for this X and H, it will hold in other cases. For example, if we add hypotheses to H in this example so that it shatters the entire instance space X, then we will have $BestQ(H) = VC(H)$.



10-701/15-781, Fall 2006, Midterm

- There are 7 questions in this exam (11 pages including this cover sheet).
- Questions are not equally difficult.
- If you need more room to work out your answer to a question, use the back of the page and clearly mark on the front of the page if we are to look at what's on the back.
- This exam is open book and open notes. Computers, PDAs, cell phones are not allowed.
- You have 1 hour and 20 minutes. Good luck!

Name:
Andrew ID:

Q	Topic	Max. Score	Score
1	Conditional Independece, MLE/MAP, Probability	12	
2	Decision Tree	12	
3	Neural Network and Regression	18	
4	Bias-Variance Decomposition	12	
5	Support Vector Machine	12	
6	Generative vs. Discriminative Classifier	20	
7	Learning Theory	14	
	Total	100	

1 Conditional Independence, MLE/MAP, Probability (12 pts)

1. (4 pts) Show that $\Pr(X, Y|Z) = \Pr(X|Z)\Pr(Y|Z)$ if $\Pr(X|Y, Z) = \Pr(X|Z)$.

$$\Pr(X,Y|Z) = \Pr(X|Y,Z)\Pr(Y|Z) \quad \text{(chain rule)}$$
$$= \Pr(X|Z)\Pr(Y|Z)$$

common mistake: $\Pr(X|Y,Z) = \Pr(X|Z) \Rightarrow X \perp Y$ given Z
$\Rightarrow \Pr(X,Y|Z) = \Pr(X|Z)\Pr(Y|Z)$

the first $\Rightarrow$ does not hold if the equation is not for all possible values of X, Y, Z.

2. (4 pts) If a data point y follows the Poisson distribution with rate parameter θ, then the probability of a single observation y is

$$p(y|\theta) = \frac{\theta^y e^{-\theta}}{y!}, \quad \text{for } y = 0, 1, 2, \cdots.$$

You are given data points $y_1, \cdots, y_n$ independently drawn from a Poisson distribution with parameter θ. Write down the log-likelihood of the data as a function of θ.

$$\sum_{i=1}^{n} \left(y_i \log\theta - \theta - \log y_i! \right)$$
$$= \left(\sum_{i=1}^{n} y_i \right) \log\theta - n\theta - \log\left(\prod_{i=1}^{n} y_i! \right)$$

3. (4 pts) Suppose that in answering a question in a multiple choice test, an examinee either knows the answer, with probability p, or he guesses with probability $1 - p$. Assume that the probability of answering a question correctly is 1 for an examinee who knows the answer and $1/m$ for the examinee who guesses, where m is the number of multiple choice alternatives. What is the probability that an examinee knew the answer to a question, given that he has correctly answered it?

$$P(\text{know answer} \mid \text{correct}) = \frac{P(\text{know answer, correct})}{P(\text{correct})}$$
$$= \frac{p}{p + (1-p)\frac{1}{m}}$$

2 Decision Tree (12 pts)

The following data set will be used to learn a decision tree for predicting whether students are lazy (L) or diligent (D) based on their weight (Normal or Underweight), their eye color (Amber or Violet) and the number of eyes they have (2 or 3 or 4).

Weight	Eye Color	Num. Eyes	Output
N	A	2	L
N	V	2	L
N	V	2	L
U	V	3	L
U	V	3	L
U	A	4	D
N	A	4	D
N	V	4	D
U	A	3	D
U	A	3	D

The following numbers may be helpful as you answer this problem without using a calculator: $\log_2 0.1 = -3.32, \log_2 0.2 = -2.32, \log_2 0.3 = -1.73, \log_2 0.4 = -1.32, \log_2 0.5 = -1$.

*You don't need to show the derivation for your answers in this problem.

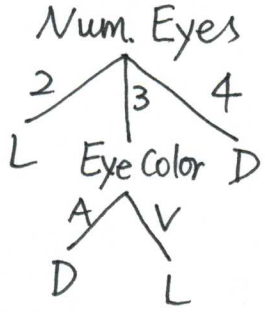
1. (3 pts) What is the conditional entropy $H(EyeColor|Weight = N)$?

$$-(0.4\log_2 0.4 + 0.6\log_2 0.6)$$
$$= 0.4 \times 1.32 + 0.6 \times (1.73 - 1) = 0.966$$

2. (3 pts) What attribute would the ID3 algorithm choose to use for the root of the tree (no pruning)?

Num. Eyes

3. (4 pts) Draw the full decision tree learned for this data (no pruning).



4. (2 pts) What is the training set error of this unpruned tree?

0

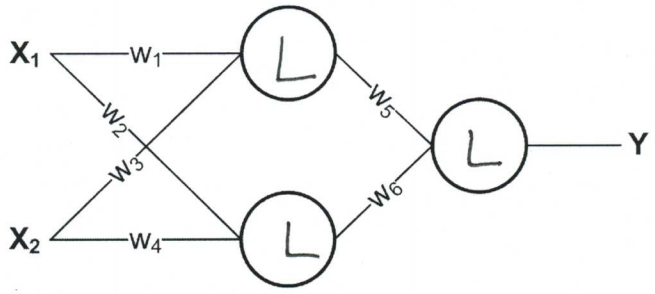
3 Neural Network and Regression (18 pts)

Consider a two-layer neural network to learn a function $f : X \rightarrow Y$ where $X = \langle X_1, X_2 \rangle$ consists of two attributes. The weights, $w_1, \cdots, w_6$, can be arbitrary. There are two possible choices for the function implemented by each unit in this network:

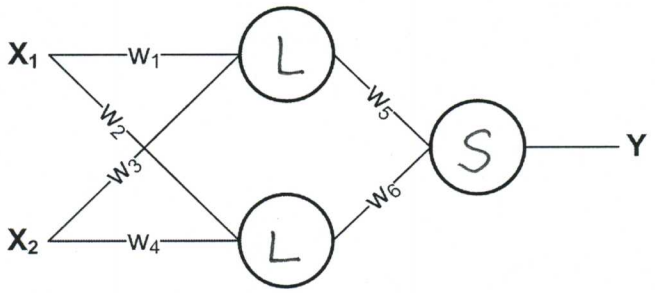
- **S**: signed sigmoid function $S(a) = sign[\sigma(a) - 0.5] = sign[\frac{1}{1+\exp(-a)} - 0.5]$
- **L**: linear function $L(a) = c\,a$

where in both cases $a = \sum_i w_i X_i$

1. (4 pts) Assign proper activation functions (**S** or **L**) to each unit in the following graph so this neural network simulates a linear regression: $Y = \beta_1 X_1 + \beta_2 X_2$.



2. (4 pts) Assign proper activation functions (**S** or **L**) for each unit in the following graph so this neural network simulates a binary logistic regression classifier: $Y = \arg\max_y P(Y = y|X)$, where $P(Y = 1|X) = \frac{\exp(\beta_1 X_1 + \beta_2 X_2)}{1+\exp(\beta_1 X_1 + \beta_2 X_2)}$, $P(Y = -1|X) = \frac{1}{1+\exp(\beta_1 X_1 + \beta_2 X_2)}$.

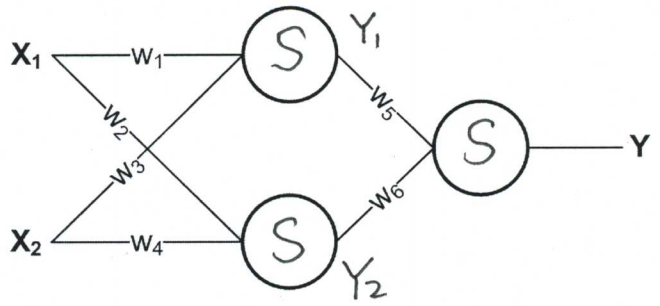


3. (3 pts) Following problem 3.2, derive β_1 and β_2 in terms of $w_1, \cdots, w_6$.

$\beta_1 = c(w_1 w_5 + w_2 w_6)$

$\beta_2 = c(w_3 w_5 + w_4 w_6)$

4. (4 pts) Assign proper activation functions (**S** or **L**) for each unit in the following graph so this neural network simulates a boosting classifier which combines two logistic regression classifiers, $f_1 : X \to Y_1$ and $f_2 : X \to Y_2$, to produce its final prediction: $Y = sign[\alpha_1 Y_1 + \alpha_2 Y_2]$. Use the same definition in problem 3.2 for f_1 and f_2.



5. (3 pts) Following problem 3.4, derive α_1 and α_2 in terms of $w_1, \cdots, w_6$.

$\alpha_1 = w_5$ #

$\alpha_2 = w_6$

4 Bias-Variance Decomposition (12 pts)

1. (6 pts) Suppose you have regression data generated by a polynomial of degree 3. Characterize the bias-variance of the estimates of the following models on the data with respect to the true model by circling the appropriate entry.

	Bias	Variance
Linear regression	low/high	low/high
Polynomial regression with degree 3	low/high	low/high
Polynomial regression with degree 10	low/high	low/high

2. Let $Y = f(X) + \epsilon$, where ϵ has mean zero and variance σ_ϵ^2. In k-nearest neighbor (kNN) regression, the prediction of Y at point x_0 is given by the average of the values Y at the k neighbors closest to x_0.

   (a) (2 pts) Denote the ℓ-nearest neighbor to x_0 by $x_{(\ell)}$ and its corresponding Y value by $y_{(\ell)}$. Write the prediction $\hat{f}(x_0)$ of the kNN regression for x_0 in terms of $y_{(\ell)}, 1 \leq \ell \leq k$.

$$\hat{f}(x_0) = \frac{1}{k} \sum_{\ell=1}^{k} y_{(\ell)}$$

   (b) (2 pts) What is the behavior of the bias as k increases?

increase

   (c) (2 pts) What is the behavior of the variance as k increases?

decrease

5 Support Vector Machine (12 pts)

Consider a supervised learning problem in which the training examples are points in 2-dimensional space. The positive examples are $(1,1)$ and $(-1,-1)$. The negative examples are $(1,-1)$ and $(-1,1)$.

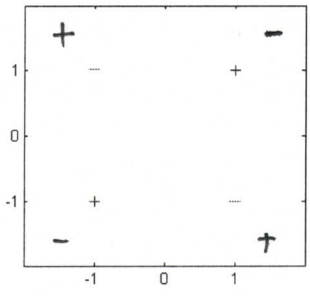
1. (1 pts) Are the positive examples linearly separable from the negative examples in the original space?

   No

2. (4 pts) Consider the feature transformation $\phi(x) = [1, x_1, x_2, x_1 x_2]$, where x_1 and x_2 are, respectively, the first and second coordinates of a generic example x. The prediction function is $y(x) = w^T * \phi(x)$ in this feature space. Give the coefficients, w, of a maximum-margin decision surface separating the positive examples from the negative examples. (You should be able to do this by inspection, without any significant computation.)

   $w = (0,0,0,1)^T$

3. (3 pts) Add one training example to the graph so the total five examples can no longer be linearly separated in the feature space $\phi(x)$ defined in problem 5.2.



4. (4 pts) What kernel $K(x, x')$ does this feature transformation ϕ correspond to?

   $1 + x_1 x_1' + x_2 x_2' + x_1 x_2 x_1' x_2'$

6 Generative vs. Discriminative Classifier (20 pts)

Consider the binary classification problem where class label $Y \in \{0, 1\}$ and each training example X has 2 binary attributes $X_1, X_2 \in \{0, 1\}$.

In this problem, we will always assume X_1 and X_2 are conditional independent given Y, that the class priors are $P(Y = 0) = P(Y = 1) = 0.5$, and that the conditional probabilities are as follows:

$P(X_1\|Y)$	$X_1 = 0$	$X_1 = 1$
$Y = 0$	0.7	0.3
$Y = 1$	0.2	0.8

$P(X_2\|Y)$	$X_2 = 0$	$X_2 = 1$
$Y = 0$	0.9	0.1
$Y = 1$	0.5	0.5

The expected error rate is the probability that a classifier provides an incorrect prediction for an observation: if Y is the true label, let $\hat{Y}(X_1, X_2)$ be the predicted class label, then the expected error rate is

$$P_{\mathcal{D}}\left(Y = 1 - \hat{Y}(X_1, X_2)\right) = \sum_{X_1=0}^{1} \sum_{X_2=0}^{1} P_{\mathcal{D}}\left(X_1, X_2, Y = 1 - \hat{Y}(X_1, X_2)\right).$$

Note that we use the subscript $\mathcal{D}$ to emphasize that the probabilities are computed under the true distribution of the data.

*You don't need to show all the derivation for your answers in this problem.

1. (4 pts) Write down the naïve Bayes prediction for all the 4 possible configurations of X_1, X_2. The following table would help you to complete this problem.

X_1	X_2	$P(X_1, X_2, Y = 0)$	$P(X_1, X_2, Y = 1)$	$\hat{Y}(X_1, X_2)$
0	0	$0.7 \times 0.9 \times 0.5$	$0.2 \times 0.5 \times 0.5$	0
0	1	$0.7 \times 0.1 \times 0.5$	$0.2 \times 0.5 \times 0.5$	1
1	0	$0.3 \times 0.9 \times 0.5$	$0.8 \times 0.5 \times 0.5$	1
1	1	$0.3 \times 0.1 \times 0.5$	$0.8 \times 0.5 \times 0.5$	1

2. (4 pts) Compute the expected error rate of this naïve Bayes classifier which predicts Y given both of the attributes $\{X_1, X_2\}$. Assume that the classifier is learned with infinite training data.

$$\begin{aligned} & 0.2 \times 0.5 \times 0.5 \\ + & 0.7 \times 0.1 \times 0.5 \\ + & 0.3 \times 0.9 \times 0.5 \\ + & 0.3 \times 0.1 \times 0.5 \\ = & \ 0.235 \end{aligned}$$

3. (4 pts) Which of the following two has a smaller expected error rate?

- (•) the naïve Bayes classifier which predicts Y given X_1 only
- ( ) the naïve Bayes classifier which predicts Y given X_2 only

Prediction $X_2=1 \mapsto \hat{Y}=1$, $X_2=0 \to \hat{Y}=0$

$P_D(X_2=1, Y=0)+P_D(X_2=0, Y=1)$
$=0.1\times0.5+0.5\times0.5=0.3$

Prediction
$X_1=1 \to \hat{Y}=1$
$X_1=0 \to \hat{Y}=0$

$P_D(X_1=0, Y=0)+P_D(X_1=0, Y=1)$
$=0.3\times0.5+0.2\times0.5=0.25$

4. (4 pts) Now, suppose that we create a new attribute X_3, which is a deterministic copy of X_2. What is the expected error rate of the naïve Bayes which predicts Y given all the attributes (X_1, X_2, X_3) now? Assume that the classifier is learned with infinite training data.

X_1	X_2	$X_3=X_2$	$P_{NB}(X_1,X_2,X_3,Y=0)$	$P_{NB}(X_1,X_2,X_3,Y=1)$	$\hat{Y}(X_1,X_2,X_3)$
0	0	0	$0.7\times0.9\times0.9\times0.5$	$0.2\times0.5\times0.5\times0.5$	0
0	1	1	$0.7\times0.1\times0.1\times0.5$	$0.2\times0.5\times0.5\times0.5$	1
1	0	0	$0.3\times0.9\times0.9\times0.5$	$0.8\times0.5\times0.5\times0.5$	0
1	1	1	$0.3\times0.1\times0.1\times0.5$	$0.8\times0.5\times0.5\times0.5$	1

Error rate $= 0.2\times0.5\times0.5$ $P_D(X_1=0, X_2=0, Y=1)$
$+ 0.7\times0.1\times0.5$ $P_D(X_1=0, X_2=1, Y=0)$
$+ 0.8\times0.5\times0.5$ $P_D(X_1=1, X_2=0, Y=1)$
$+ 0.3\times0.1\times0.5$ $P_D(X_1=1, X_2=1, Y=0)$
$= 0.3$

5. (4 pts) Explain what is happening with naïve Bayes in problem 6.4? Does logistic regression suffer from the same problem? Why?

The conditional independence assumption of naïve Bayes classifier does not hold. X_2 is overcounted leading to an error prediction when $X_1=0, X_2=0$.

LR does not suffer because it does not make such conditional independence assumption.

7 Learning Theory (14 pts)

You read in the paper that the famous bird migration website, Netflocks, is offering a $1M prize for accurately recommending movies about penguins. Furthermore, it is providing a training data set containing 100,000,000 labeled training examples. Each training example consists of a set of 100 real-valued features describing a movie, along with a boolean label indicating whether to recommend this movie to a person.

You determine that the $1M can be yours if you can train a *linear* Support Vector Machine with a true accuracy of 98%. Of course you understand that PAC learning theory provides only probabilistic bounds, so you decide to enter only if you can prove you have at least a 0.9 probability of achieving an accuracy of 98%.

1. (8 pts) Can you use PAC learning theory to decide whether you can meet your performance objective? If yes, give an expression for the number of training examples sufficient to meet your performance objective. If not, explain why not, then provide the minimum set of additional assumptions needed so that PAC learning theory can be applied, and give an expression of the number of training examples sufficient under your assumptions. (you may leave your expression as an unsolved arithmetic expression, but it should contain only constants - no variables).

No. Need to assume the true concept $C \in H$ to apply PAC learning theory.

$$m = \frac{1}{\varepsilon}\left(4\log_2\frac{1}{\delta} + 8\,VC(H)\log_2\frac{13}{\varepsilon}\right)$$

with

$$\varepsilon = 0.02,\quad \delta = 0.1,\quad VC(H) = 100+1 = 101$$

then

$$m = 50\left(4\log_2 10 + 808\log_2 650\right) << 100{,}000{,}000.$$

2. (3 pts) Consider the PAC-style statement "we can achieve true accuracy of at least 98% with probability 0.9." What is the meaning of "with probability 0.9"? Answer this by describing a randomized experiment which you could perform repeatedly to test whether the statement is true.

We could do M repititions of the following experiments (M is a large number, say 10^6).

generate 100,000,000 training data from the true data distribution D, train a linear SVM, then compute its accuracy on the true distribution D. If the accuracy is greater than 98%, report a positive result; otherwise, report a negative one.

If we get a number greater than 0.9M of positive results, we validate the PAC statement

3. (3 pts) Your friend already has a private dataset of 100,000,000 labeled movies, so she will end up with twice as much training data as you. You train using the Netflocks data to produce a classifier h_1. She uses the same learning algorithm, but trains with twice as much data to produce her output hypothesis, h_2. You are interested in how well the training errors of h_1 and h_2 predict their true errors. Consider the ratio

$$\frac{error_{train}(h_1) - error_{true}(h_1)}{error_{train}(h_2) - error_{true}(h_2)}$$

Which of these is the most likely value for this ratio? Circle the answer and give a *one-sentence* explanation.

$$4, 2, \textcircled{\sqrt{2}}, 1, -1, \frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{4}$$

$$error_{true}(h) < error_{train}(h) + \sqrt{\frac{VC(H)\left(\ln\frac{2m}{VC(H)}+1\right)+\ln\frac{4}{\delta}}{m}}$$

the difference is approx. $\propto m^{-\frac{1}{2}}$

* Any resemblance to real persons, animals, or organizations, living or dead, is purely coincidental.

10-701 Midterm Exam Solutions, Spring 2007

1. Personal info:
   - Name:
   - Andrew account:
   - E-mail address:
2. There should be 16 numbered pages in this exam (including this cover sheet).
3. You can use any material you brought: any book, class notes, your print outs of class materials that are on the class website, including my annotated slides and relevant readings, and Andrew Moore's tutorials. You cannot use materials brought by other students. Calculators are allowed, but no laptops, PDAs, phones or Internet access.
4. If you need more room to work out your answer to a question, use the back of the page and clearly mark on the front of the page if we are to look at what's on the back.
5. Work efficiently. Some questions are easier, some more difficult. Be sure to give yourself time to answer all of the easy ones, and avoid getting bogged down in the more difficult ones before you have answered the easier ones.
6. Note there are extra-credit sub-questions. The grade curve will be made without considering students' extra credit points. The extra credit will then be used to try to bump your grade up without affecting anyone else's grade.
7. You have 80 minutes.
8. Good luck!

Question	Topic	Max. score	Score
1	Short questions	21 + 0.911 extra	
2	SVM and slacks	16	
3	GNB	8	
4	Feature Selection	10	
5	Irrelevant Features	14 + 3 extra	
6	Neural Nets	16 + 5 extra	
7	Learning theory	15	

1 [21 Points] Short Questions

The following short questions should be answered with at most two sentences, and/or a picture. For the **(true/false)** questions, answer true or false. If you answer true, provide a short justification, if false explain why or provide a small counterexample.

1. [1 point] **true/false** A classifier trained on less training data is less likely to overfit.

   ★ **SOLUTION:** This is false. A specific classifier (with some fixed model complexity) will be more likely to overfit to noise in the training data when there is less training data, and is therefore more likely to overfit.

2. [1 point] **true/false** Given m data points, the training error converges to the true error as $m \to \infty$.

   ★ **SOLUTION:** This is true, if we assume that the data points are i.i.d. A few students pointed out that this might not be the case.

3. [1 point] **true/false** The maximum likelihood model parameters (α) can be learned using linear regression for the model: $y_i = \alpha_1 x_1 x_2^3 + \epsilon_i$ where $\epsilon_i \sim N(0, \sigma^2)$ iid noise.

   ★ **SOLUTION:** This is true. y is linear in α_1, so it can be learned using linear regression.

4. [2 points] **true/false** The maximum likelihood model parameters (α) can be learned using linear regression for the model: $y_i = x_1^{\alpha_1} e^{\alpha_2} + \epsilon_i$ where $\epsilon_i \sim N(0, \sigma^2)$ iid noise.

   ★ **SOLUTION:** This is false. y is not linear in α_1 and α_2, and no simple transformation will make it linear ($\log[x_1^{\alpha_1} e^{\alpha_2} + \epsilon_i] \neq \alpha_1 \log x_1 + \alpha_2 + \epsilon_i$).

5. [2 points] **true/false** The maximum likelihood model parameters (α) can be learned using linear regression for the model: $y_i = \log(x_1^{\alpha_1} e^{\alpha_2}) + \epsilon_i$ where $\epsilon_i \sim N(0, \sigma^2)$ iid noise.

   ★ **SOLUTION:** This is true. $y_i = \log(x_1^{\alpha_1} e^{\alpha_2}) + \epsilon_i = \alpha_1 \log x_1 + \alpha_2 + \epsilon_i$, which is linear in α_1 and α_2. Also, assuming $x_1 > 0$.

6. [2 points] **true/false** In AdaBoost weights of the misclassified examples go up by the same multiplicative factor.

   ★ **SOLUTION:** True, follows from the update equation.

7. [2 points] **true/false** In AdaBoost, weighted training error ϵ_t of the t^{th} weak classifier on training data with weights D_t tends to increase as a function of t.

   ★ **SOLUTION:** True. In the course of boosting iterations the weak classifiers are forced to try to classify more difficult examples. The weights will increase for examples that are repeatedly misclassified by the weak classifiers. The weighted training error ϵ_t of the t^{th} weak classifier on the training data therefore tends to increase.

8. [2 points] **true/false** AdaBoost will eventually give zero training error regardless of the type of weak classifier it uses, provided enough iterations are performed.

   ★ **SOLUTION:** Not if the data in the training set cannot be separated by a linear combination of the specific type of weak classifiers we are using. For example consider the EXOR example(In hw2 we worked with a *rotated* EXOR toy dataset) with decision stumps as weak classifiers. No matter how many iterations are performed zero training error will not be achieved.

9. [2 points] Consider a point that is correctly classified and distant from the decision boundary. Why would SVM's decision boundary be unaffected by this point, but the one learned by logistic regression be affected?

   ★ **SOLUTION:** The hinge loss used by SVMs gives zero weight to these points while the log-loss used by logistic regression gives a little bit of weight to these points.

10. [2 points] Why does the kernel trick allow us to solve SVMs with high dimensional feature spaces, without significantly increasing the running time?

★ **SOLUTION:** In the dual formulation of the SVM, features only appear as dot products which can be represented compactly by kernels.

11. [2 points] Consider a learning problem with 2D features. How are the decision tree and 1-nearest neighbor decision boundaries related?

    ★ **SOLUTION:** In both cases, the decision boundary is piecewise linear. Decision trees do axis-aligned splits while 1-NN gives a voronoi diagram.

12. [2 points] You are a reviewer for the International Mega-Conference on Algorithms for Radical Learning of Outrageous Stuff, and you read papers with the following experimental setups. Would you accept or reject each paper? Provide a one sentence justification. (This conference has short reviews.)

    - **accept/reject** "My algorithm is better than yours. Look at the training error rates!"

      ★ **SOLUTION:** Reject - the training error is optimistically biased.

    - **accept/reject** "My algorithm is better than yours. Look at the test error rates! (Footnote: reported results for $\lambda = 1.789489345672120002$.)"

      ★ **SOLUTION:** Reject - A λ with 15 decimal places suggests a highly tuned solution, probably looking at the test data.

    - **accept/reject** "My algorithm is better than yours. Look at the test error rates! (Footnote: reported results for best value of λ.)"

      ★ **SOLUTION:** Reject - Choosing λ based on the test data?

    - **accept/reject** "My algorithm is better than yours. Look at the test error rates! (Footnote: reported results for best value of λ, chosen with 10-fold cross validation.)"

      ★ **SOLUTION:** Accept - Cross validation is the appropriate method for selecting parameters.

13. [Extra credit: 0.911 points] You have designed the ultimate learning algorithm that uses physical and metaphysical knowledge to learn and generalize beyond the quantum P-NP barrier. You are now given the following test example:

    What label will your algorithm output?

(a) Watch a cartoon.

(b) Call the anti-terrorism squad.

(c) Support the Boston Red Sox.

(d) All labels have equal probability.

★ **SOLUTION:** Watching a cartoon earned you 0.39 points. 0.2005 points were given for supporting the Boston Red Sox. 0.666 points were given for calling the anti-terrorism squad. 0.911 points were given for "all labels have equal probability."

■ **COMMON MISTAKE :** Some students skipped this question; perhaps a Moonite Marauders is also scary on paper...

2 [16 Points] SVMs and the slack penalty C

The goal of this problem is to correctly classify test data points, given a training data set. You have been warned, however, that the training data comes from sensors which can be error-prone, so you should avoid trusting any specific point too much.

For this problem, assume that we are training an SVM with a **quadratic kernel**– that is, our kernel function is a polynomial kernel of degree 2. You are given the data set presented in Figure 1. The slack penalty C will determine the location of the separating hyperplane. Please answer the following questions *qualitatively*. Give a one sentence answer/justification for each and draw your solution in the appropriate part of the Figure at the end of the problem.

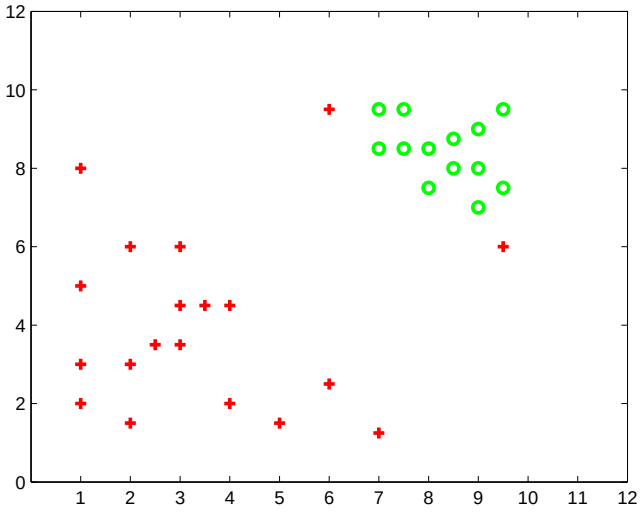


Figure 1: Dataset for SVM slack penalty selection task in Question 2.

1. [4 points] Where would the decision boundary be for very large values of C (i.e., $C \to \infty$)? (remember that we are using an SVM with a quadratic kernel.) Draw on the figure below. Justify your answer.

   ★ **SOLUTION:** For large values of C, the penalty for misclassifying points is very high, so the decision boundary will perfectly separate the data if possible. See below for the boundary learned using libSVM and $C = 100000$.

   ■ **COMMON MISTAKE 1:** Some students drew straight lines, which would not be the result with a quadratic kernel.

■ **COMMON MISTAKE 2:** Some students confused the effect of C and thought that a large C meant that the algorithm would be more tolerant of misclassifications.

2. [4 points] For $C \approx 0$, indicate in the figure below, where you would expect the decision boundary to be? Justify your answer.

★ **SOLUTION:** The classifier can maximize the margin between most of the points, while misclassifying a few points, because the penalty is so low. See below for the boundary learned by libSVM with $C = 0.00005$.

3. [2 points] Which of the two cases above would you expect to work better in the classification task? Why?

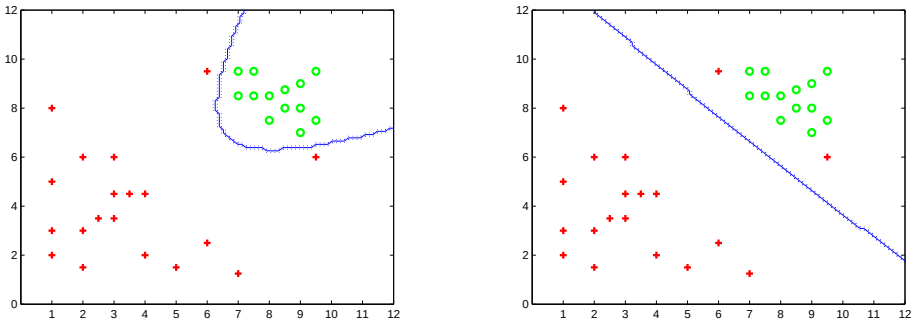
★ **SOLUTION:** We were warned not to trust any specific data point too much, so we prefer the solution where $C \approx 0$, because it maximizes the margin between the dominant clouds of points.

4. [3 points] Draw a data point which will not change the decision boundary learned for very large values of C. Justify your answer.

★ **SOLUTION:** We add the point circled below, which is correctly classified by the original classifier, and will not be a support vector.

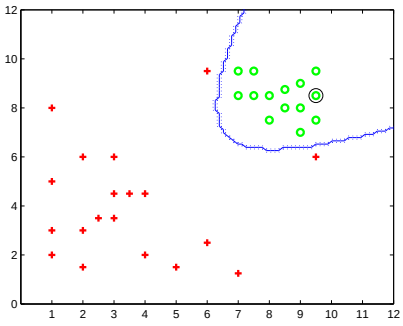
5. [3 points] Draw a data point which will significantly change the decision boundary learned for very large values of C. Justify your answer.

★ **SOLUTION:** Since C is very large, adding a point that would be incorrectly classified by the original boundary will force the boundary to move.

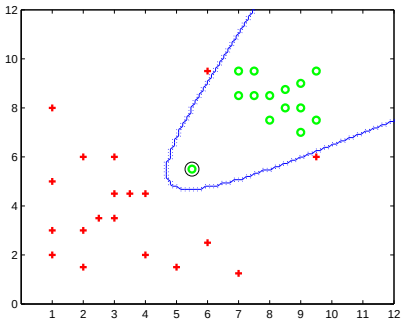


(a) Part 1

(b) Part 2



(c) Part 4



(d) Part 5

Figure 2: Solutions for Problem 2

3 [10 points] Feature selection with boosting

Consider a text classification task, such that the document X can be expressed as a binary feature vector of the words. More formally $X = [X_1, X_2, X_3, ..., X_m]$, where $X_j = 1$ if word j is present in document X, and zero otherwise. Consider using the AdaBoost algorithm with a simple weak learner, namely

$$h(X;\theta) = yX_j$$
$$\theta = \{j, y\} \; j \text{ is the word selector ; } y \text{ is the associated class}$$
$$y \in \{-1, 1\}$$

More intuitively, each weak learner is a word associated with a class label. For example if we had a word `football`, and classes {`sports`,`non-sports`}, then we will have two weak learners from this word, namely

- *Predict* **sports** *if document has word* **football**
- *Predict* **non-sports** *if document has word* **football**.

1. [2 points] How many weak learners are there ?

★ **SOLUTION:** Two weak learners for each word, i.e. 2m weak learners.

2. This boosting algorithm can be used for feature selection. We run the algorithm and select the features in the *order in which they were identified* by the algorithm.

   (a) [4 points] Can this boosting algorithm select the same weak classifier more than once? Explain.

   ★ **SOLUTION:** The boosting algorithm optimizes each new α by assuming that all the previous votes remain fixed. It therefore does not optimize these coefficients jointly. The only way to correct the votes assigned to a weak learner later on is to introduce the same weak learner again. Since we only have a discrete set of possible weak learners here, it also makes sense to talk about selecting the exact same weak learner again.

   (b) [4 points] Consider ranking the features based on their individual mutual information with the class variable y, i.e. $\hat{I}(y; X_j)$. Will this ranking be more informative than the ranking returned by AdaBoost ? Explain.

   ★ **SOLUTION:** The boosting algorithm generates a linear combination of weak classifiers (here features). The algorithm therefore evaluates each new weak classifier (feature) relative to a linear prediction based on those already included. The mutual information criterion considers each feature individually and is therefore unable to recognize how multiple features might interact to benefit linear prediction.

4 [8 points] Gaussian Naive Bayes classifier

Consider the datasets **toydata1** in figure **3(A)** and **toydata2** in figure **3(B)**.

- In each of these datasets there are two classes, '+' and 'o'.
- Each class has the same number of points.
- Each data point has two real valued features, the X and Y coordinates.

For each of these datasets, draw the decision boundary that a Gaussian Naive Bayes classifier will learn.

★ **SOLUTION:** For **toydata1** the crucial detail is that GNB learns diagonal covariance matrices yielding axis aligned Gaussians. In figure 4(A) the two circles are the gaussians learned by GNB, and hence the decision surface is the tangent through the point of contact.

For **toydata2** GNB learns two Gaussians , one for the circle inside with small variance , and one for the circle outside with a much larger variance, and the decision surface is roughly shown in figure 4(B).

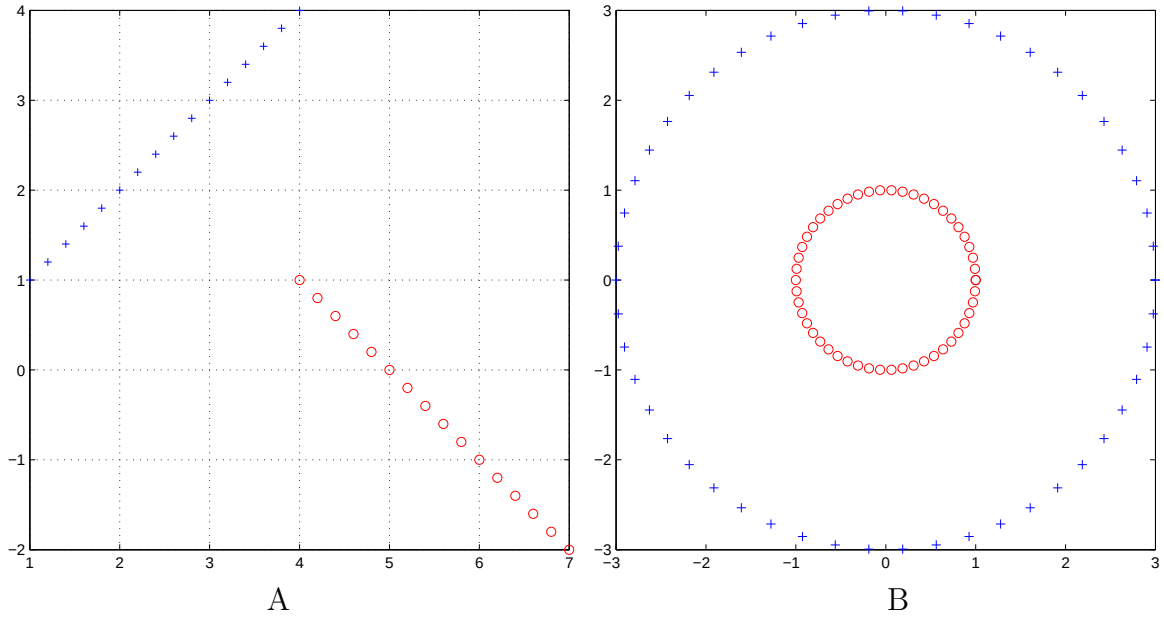


Figure 3: A. **toydata1** in Question 4, B. **toydata2** in Question 4

Remember that a very important piece of information was that all the classes had the *same* number of points, and so we don't have to worry about the prior.

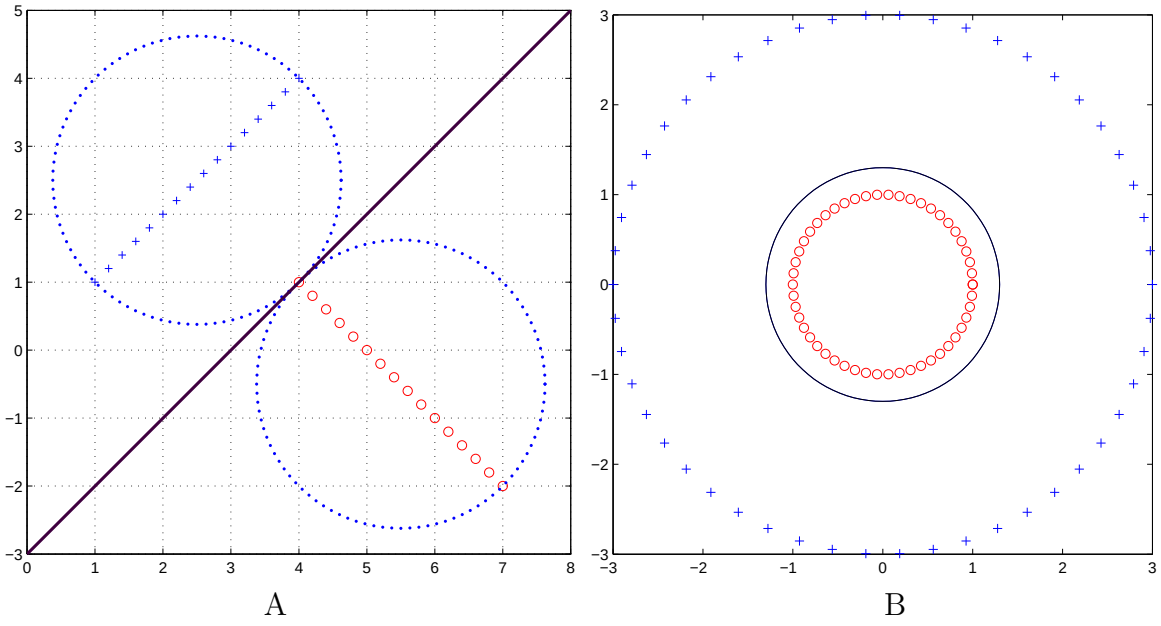
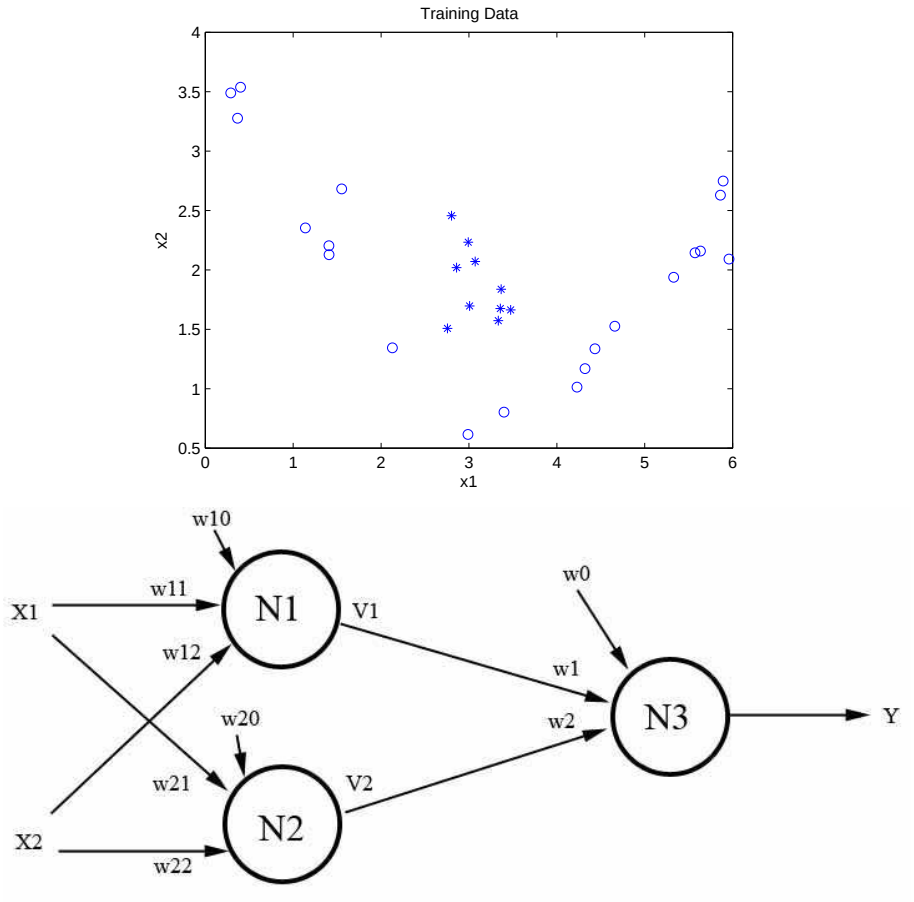


Figure 4: Solutions for A. **toydata1** in Question 4, B. **toydata2** in Question 4

5 [16 Points] Neural Networks

Consider the following classification training data (where "*" = true or 1 and "O" = false or 0) and neural network model that uses the **sigmoid** response function $\left(g(t) = \frac{1}{1+e^{-t}}\right)$.

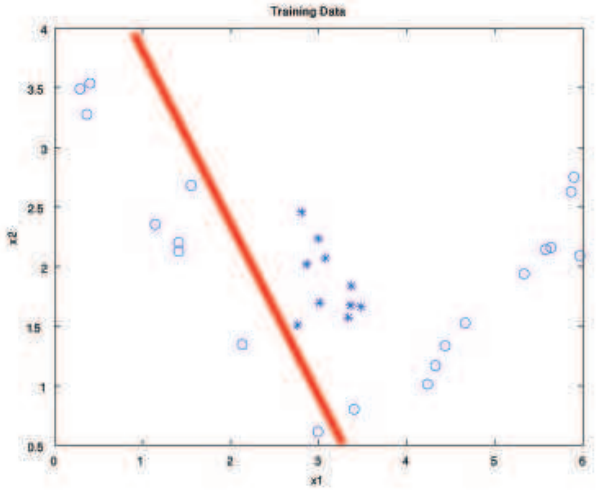


5.1 Weight choice

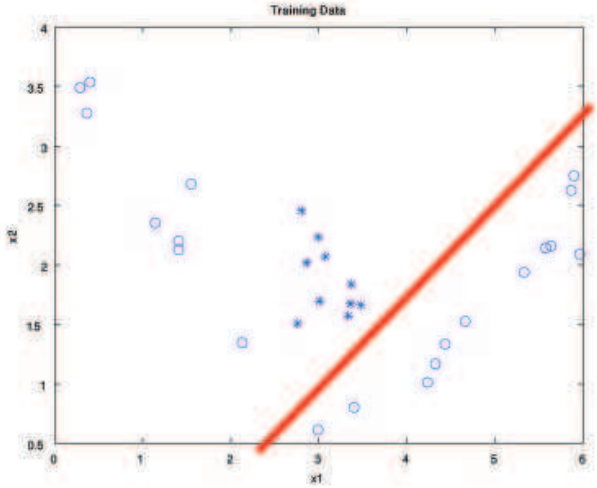
[8 points] We would like to set the weights (w) of the neural network so that it is capable of correctly classifying this dataset. Please plot the decision boundaries for N_1 and N_2 (e.g., for neuron N_1, the line where $w10 + w11 * X1 + w12 * X2 = 0$) on the first two graphs. In the third graph, which has axes V_2 and V_1, plot $\{V_1(x_1, x_2), V_2(x_1, x_2)\}$ for a few of the training points and provide a decision boundary so that the neural net will correctly classify the training data.

All graphs are on the following page!

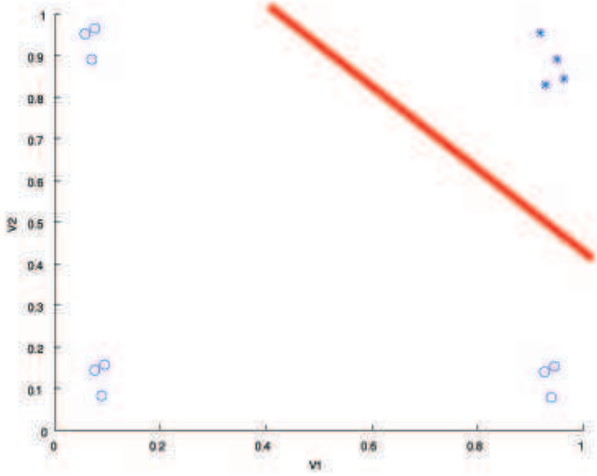
N1 (2 points)



N2 (2 points)



N3 (4 points)



5.2 Regularized Neural Networks

[8 points]

One method for preventing the neural networks' weights from overfitting is to add regularization terms. You will now derive the update rules for the regularized neural network. **Note:** $Y = out(x)$

Recall that the non-regularized gradient descent update rule for w_1 is:

$$w_1^{t+1} \leftarrow w_1^t + \eta \sum_j \left[\left(y^{(j)} - out(x^{(j)}) \right) out(x^{(j)})(1 - out(x^{(j)})) * V_1(x^{(j)}) \right] \tag{1}$$

[4 points] Derive the update rule for w_1 in the regularized neural net loss function which penalizes based on the square of each weight. Use λ to denote the magic regularization parameter.

★ **SOLUTION:** The regularization term is $\lambda(\sum_i w_i^2)$. Differentiating with respect to w_1 yields $2\lambda w_1$. The update rule is

$$w_1^{t+1} \leftarrow w_1^t + \eta(\sum_j \left[\left(y^{(j)} - out(x^{(j)}) \right) out(x^{(j)})(1 - out(x^{(j)})) * V_1(x^{(j)}) \right] - 2\lambda w_1)$$

[4 points] Now, re-express the regularized update rule so that the only difference between the regularized setting and the unregularized setting above is that the old weight w_1^t is scaled by some constant. Explain how this scaling prevents overfitting.

★ **SOLUTION:**

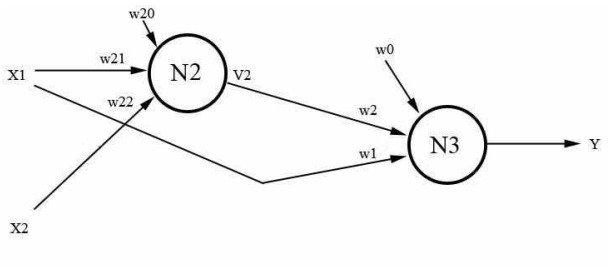
$$w_1^{t+1} \leftarrow w_1^t(1 - 2\eta\lambda) + \eta \sum_j \left[\left(y^{(j)} - out(x^{(j)}) \right) out(x^{(j)})(1 - out(x^{(j)})) * V_1(x^{(j)}) \right]$$

At each update the weight is kept closer to zero by the $(1 - 2\eta\lambda)$ term. This prevents the weights from becoming very large, which corresponds to overfitting.

5.3 Neural Net Simplification [Extra Credit (5 points)]

Please provide a feed-forward neural network with a smaller architecture (i.e., fewer neurons and weights) that is able to correctly predict the entire training set. Justify your answer.

★ **SOLUTION:** One solution follows from the observation that the decision boundary for N1 could be $x_1 = 3.7$. In fact, $N1$ can be removed entirely from the model. This yields a similar decision boundary for $N3$ except that $V1 = x_1$ ranges from 0 to 6.



Other possible solutions are to change the input feature space (e.g., by adding x_1^2 as an input to a single neuron along with x_1, x_2).

6 [14 Points] The Effect of Irrelevant Features

1. (a) [3 points] Provide a 2D dataset where 1-nearest neighbor (1-NN) has lower leave-one-out cross validation error (LOO error) than SVMs.

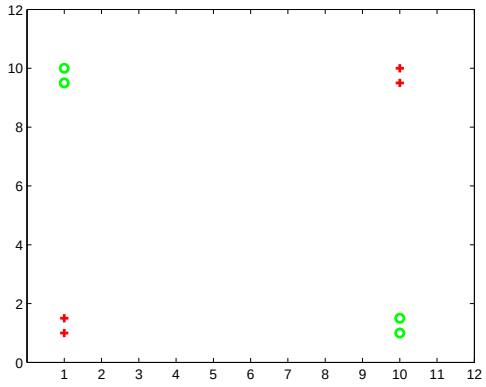


Figure 5: Dataset where 1-NN will do better than SVM in LOOCV.

(b) [3 points] Provide a 2D dataset where 1-NN has higher LOO error than SVMs.

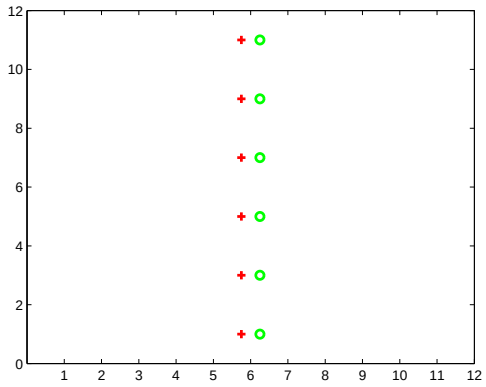


Figure 6: Dataset where SVM will do better than 1-NN in LOOCV.

2. [8 points] You will now generate a dataset to illustrate SVMs' robustness to irrelevant features. In particular, create a 2D dataset with features X_1 and X_2, here X_2 will be the irrelevant feature, such that:

   - If you only use X_1, 1-NN will have lower LOO error than SVMs,
   - but if you use both X_1 and X_2, the SVM LOO error will remain the same, but LOO error for 1-NN will increase significantly.

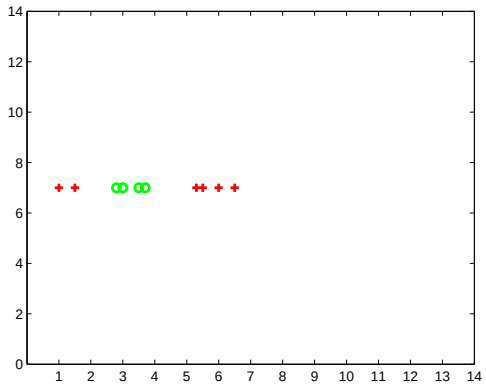


Figure 7: Here the horizontal axis is X_1, and we ignore X_2. 1-NN is perfect, and SVM will get the two points on the left wrong in LOOCV.

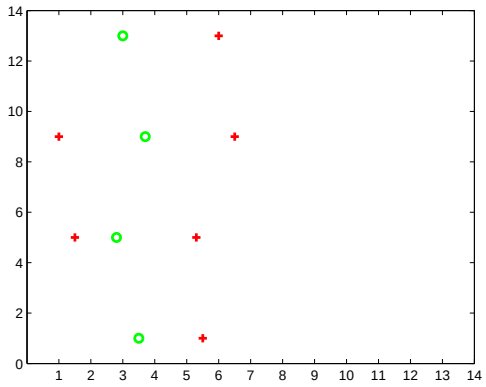


Figure 8: Here the horizontal axis is X_1, and the vertical axis is X_2. X_2 is the irrelevant feature. 1-NN gets every point wrong in LOOCV, while SVM has the same error.

You will receive extra credit if the 1-NN LOO error before adding the irrelevant feature is zero, but the error becomes 100% after adding the feature.

3. [Extra Credit (3 points)] SVMs tend to be robust to irrelevant features. Suppose we run SVMs with features $X_1, \ldots, X_n$, and then add a irrelevant feature X_{n+1} that cannot help increase the margin. How will SVMs automatically ignore this feature? Justify your answer formally.

★ **SOLUTION:** SVMs will automatically ignore this feature because it cannot possibly increase the margin, so giving it non-zero weight keeps the same margin but increases the regularization penalty. Therefore the solution with zero weight is superior to (i.e. has smaller objective function) all feasible solutions of the QP where this feature has non-zero weight.

7 [15 points] Learning Theory

Consider binary data-points X in n dimensions, with binary labels Y, i.e. $X \in \{0,1\}^n$; $Y \in \{0,1\}$. We wish to learn a mapping $X \to Y$ using a few different hypothesis classes, but are concerned about the tradeoff between the expressivity of our hypothesis space and the number of training examples required to learn the true mapping probably approximately correctly.

1. Consider the following hypothesis class H: decision stumps that choose a value for Y based on the value of one of the attributes of X. For example, there are two hypotheses in H that involve feature i:

$$h_i(X) = \begin{cases} 1 & \text{if } X_i = 1 \\ 0 & \text{otherwise} \end{cases} \qquad h_{\neg i}(X) = \begin{cases} 0 & \text{if } X_i = 1 \\ 1 & \text{otherwise;} \end{cases}$$

   (a) [3 points] What is the size of this hypothesis class?

   ★ **SOLUTION:** $H = 2n$

   (b) [3 points] For given ϵ, δ how many training examples are needed to yield a decision stump that satisfies the Haussler-PAC bound?

   ★ **SOLUTION:**

$$|H|e^{-m\epsilon} \leq \delta$$

$$\begin{aligned} m &\geq \frac{1}{\epsilon}\left(\log|H| + \log\frac{1}{\delta}\right) \\ &= \frac{1}{\epsilon}\left(\log(2n) + \log\frac{1}{\delta}\right) \end{aligned}$$

2. Now let us define another hypothesis class H', where each hypothesis is a majority over a set of simple decision stumps. Specifically, for each feature i, we either use h_i or $h_{\neg i}$, and the output is the result of a majority vote (in the case of a tie, we predict 1). For example, if we have 5 features, and we choose the stumps $\{h_{\neg 1}, h_2, h_3, h_4, h_{\neg 5}\}$, then the resulting hypothesis is:

$$h'(X) = \begin{cases} 1 & \text{if } h_{\neg 1}(X) + h_2(X) + h_3(X) + h_4(X) + h_{\neg 5}(X) \geq \frac{5}{2} \\ 0 & \text{otherwise} \end{cases}$$

(a) [4 points] What is the size of this hypothesis class?

★ **SOLUTION:** Each element of the hypothesis class here corresponds to picking a subset of n features. Hence:

$$|H| = 2^n$$

(b) [2 points] For given ϵ, δ how many training examples are needed to yield a hypothesis that satisfies the Haussler-PAC bound?

★ **SOLUTION:**

$$m \geq \frac{1}{\epsilon}\left(n \log 2 + \log \frac{1}{\delta}\right)$$

3. [3 points] What can we say about the amount of extra samples necessary to learn this voting classifier? Is this a concern?
Briefly explain the tradeoff between the expressive power of the hypothesis space and the number of training samples required for these two classifier.

★ **SOLUTION:** In the first part of the problem, the required number of samples scales as $O(\log n)$, and in the second part of the problem, it scales as $O(n)$. So we need more samples to get the PAC bound for the second hypothesis class, but scaling linearly in the number of features is probably acceptable.

The tradeoff illustrated here is that greater expressive power (as with H') necessitates more training samples. The smaller and less expressive class H requires fewer training examples.

10-701/15-781 Machine Learning
Mid-term Exam Solution

Your Name: _____

Your Andrew ID: _____

1 True or False (Give one sentence explanation) (20%)

1. (F) For a continuous random variable x and its probability distribution function $p(x)$, it holds that $0 \leq p(x) \leq 1$ for all x.

2. (F) Decision tree is learned by minimizing information gain.

3. (F) Linear regression estimator has the smallest variance among all unbiased estimators.

4. (T) The coefficients α assigned to the classifiers assembled by AdaBoost are always non-negative.

5. (F) Maximizing the likelihood of logistic regression model yields multiple local optimums.

6. (F) No classifier can do better than a naive Bayes classifier if the distribution of the data is known.

7. (F) The back-propagation algorithm learns a globally optimal neural network with hidden layers.

8. (F) The VC dimension of a line should be at most 2, since I can find at least one case of 3 points that cannot be shattered by any line.

9. (F) Since the VC dimension for an SVM with a Radial Base Kernel is infinite, such an SVM must be worse than an SVM with polynomial kernel which has a finite VC dimension.

10. (F) A two layer neural network with linear activation functions is essentially a weighted combination of linear separators, trained on a given dataset; the boosting algorithm built on linear separators also finds a combination of linear separators, therefore these two algorithms will give the same result.

2 Linear Regression (10%)

We are interested here in a particular 1-dimensional linear regression problem. The dataset corresponding to this problem has n examples $(x_1; y_1), \ldots, (x_n; y_n)$ where x_i and y_i are real numbers for all i. Let $\mathbf{w}^* = [w_0^*, w_1^*]^T$ be the least squares solution we are after. In other words, $\mathbf{w}^*$ minimizes

$$J(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i)^2.$$

You can assume for our purposes here that the solution is unique.

1. (5%) Check each statement that must be true if $\mathbf{w}^* = [w_0^*, w_1^*]^T$ is indeed the least squares solution.

$$\begin{array}{ll} \frac{1}{n}\sum_{i=1}^{n}(y_i - w_0^* - w_1^* x_i)y_i = 0 & (\quad) \\ \frac{1}{n}\sum_{i=1}^{n}(y_i - w_0^* - w_1^* x_i)(y_i - \bar{y}) = 0 & (\quad) \\ \frac{1}{n}\sum_{i=1}^{n}(y_i - w_0^* - w_1^* x_i)(x_i - \bar{x}) = 0 & (**) \\ \frac{1}{n}\sum_{i=1}^{n}(y_i - w_0^* - w_1^* x_i)(w_0^* + w_1^* x_i) = 0 & (**) \end{array}$$

   where $\bar{x}$ and $\bar{y}$ are the sample means based on the same dataset. (hint: take the derivative of $J(\mathbf{w})$ with respect to w_0^* and w_1^*)

   (sol.) Taking the derivative with respect to w_1 and w_0 gives us the following conditions of optimality

$$\frac{\partial}{\partial w_0}J(\mathbf{w}) = \frac{2}{n}\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i) = 0$$
$$\frac{\partial}{\partial w_1}J(\mathbf{w}) = \frac{2}{n}\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i)x_i = 0$$

   This means that the prediction error $(y_i - w_0 - w_1 x_i)$ does not co-vary with any linear function of the inputs (has a zero mean and does not co-vary with the inputs). $(x_i - \bar{x})$ and $(w_0^* + w_1^* x_i)$ are both linear functions of inputs.

2. (5%) There are several numbers (statistics) computed from the data that we can use to estimate $\mathbf{w}^*$. There are

$$\begin{array}{ll} \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i & (\quad) \\ \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i & (\quad) \\ C_{xx} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 & (**) \\ C_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) & (**) \\ C_{yy} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2 & (\quad) \end{array}$$

   Suppose we only care about the value of w_1^*. We'd like to determine w_1^* on the basis of ONLY two numbers (statistics) listed above. Which two numbers do we need for this? (hint: use the answers to the previous question)

(sol.) We need C_{xx} (spread of x) and C_{xy} (linear dependence between x and y). No justification was necessary as these basic points have appeared in the course. If we want to derive these more mathematically, we can, for example, look at one of the answers to the previous question:

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - w_0^* - w_1^* x_i)(x_i - \bar{x}) = 0,$$

which we can rewritte as

$$\left[\frac{1}{n}\sum_{i=1}^{n} y_i(x_i - \bar{x})\right] - w_0^*\left[\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})\right] - w_1^*\left[\frac{1}{n}\sum_{i=1}^{n} x_i(x_i - \bar{x})\right] = 0$$

By using the fact that $1/n\sum_i(x_i - \bar{x}) = 0$ we see that

$$\frac{1}{n}\sum_{i=1}^{n} y_i(x_i - \bar{x}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x}) = C_{xy}$$
$$\frac{1}{n}\sum_{i=1}^{n} x_i(x_i - \bar{x}) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x}) = C_{xx}$$

Substituting these back into our equation above gives

$$C_{xy} - w_1^* C_{xx} = 0$$

3 AdaBoost (15%)

Consider building an ensemble of decision stumps G_m with the AdaBoost algorithm,

$$f(x) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m G_m(x)\right).$$

Figure 1 dispalys a few labeled point in two dimensions as well as the first stump we have chosen. A stump predicts binary ± 1 values, and depends only on one coordinate value (the split point). The little arrow in the figure is the normal to the stump decision boundary indicating the positive side where the stump predicts $+1$. All the points start with uniform weights.

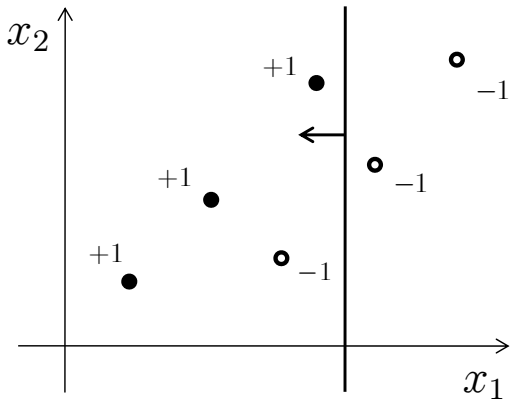


Figure 1: Labeled points and the first decision stump. The arrow points in the positive direction from the stump decision boundary.

1. (5%) Circle all the point(s) in Figure 1 whose weight will increase as a result of incorporating the first stump (the weight update due to the first stump).

   (sol.) The only misclassified negative sample.

2. (5%) Draw in the same figure a possible stump that we could select at the next boosting iteration. You need to draw both the decision boundary and its positive orientation.

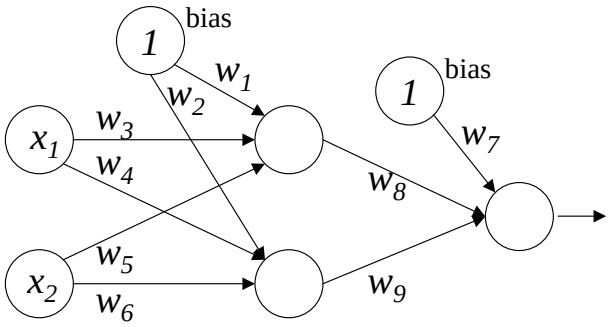
   (sol.) The second stump will also be a vertical split between the second positive sample (from left to right) and the misclassified negative smaple, as drawn in the figure.

3. (5%) Will the second stump receive higher coefficient in the ensemble than the first? In other words, will $\alpha_2 > \alpha_1$? Briefly explain your answer. (no calculation should be necessary).

   (sol.) $\alpha_2 > \alpha_1$ because the point that the second stump misclassifies will have a smaller relative weight since it is classified correctly by the first stump.

4 Neural Nets (15%)

Consider a neural net for a binary classification which has one hidden layer as shown in the figure. We use a linear activation function $h(z) = cz$ at hidden units and a sigmoid activation function $g(z) = \frac{1}{1+e^{-z}}$ at the output unit to learn the function for $P(\mathbf{y} = 1|\mathbf{x}, \mathbf{w})$ where $\mathbf{x} = (x_1, x_2)$ and $\mathbf{w} = (w_1, w_2, \ldots, w_9)$.



1. (5%) What is the output $P(\mathbf{y} = 1 \mid \mathbf{x}, \mathbf{w})$ from the above neural net? Express it in terms of x_i, c and weights w_i. What is the final classification boundary?

   (sol.)

$$\begin{aligned} & g(w_7 + w_8 h(w_1 + w_3 x_1 + w_5 x_2) + w_9 h(w_2 + w_4 x_1 + w_6 x_2)) \\ =& \frac{1}{1 + \exp(-(w_7 + cw_8 w_1 + cw_9 w_2 + (cw_8 w_3 + cw_9 w_4)x_1 + (cw_8 w_5 + cw_9 w_6)x_2))} \end{aligned}$$

   The classification boundary is :

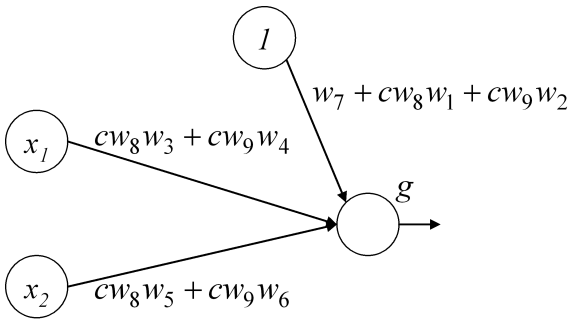
$$w_7 + cw_8 w_1 + cw_9 w_2 + (cw_8 w_3 + cw_9 w_4)x_1 + (cw_8 w_5 + cw_9 w_6)x_2 = 0$$

2. (5%) Draw a neural net with no hidden layer which is equivalent to the given neural net, and write weights $\tilde{\mathbf{w}}$ of this new neural net in terms of c and w_i.

   (sol.)

3. (5%) Is it true that any multi-layered neural net with linear activation functions at hidden layers can be represented as a neural net without any hidden layer? Briefly explain your answer.

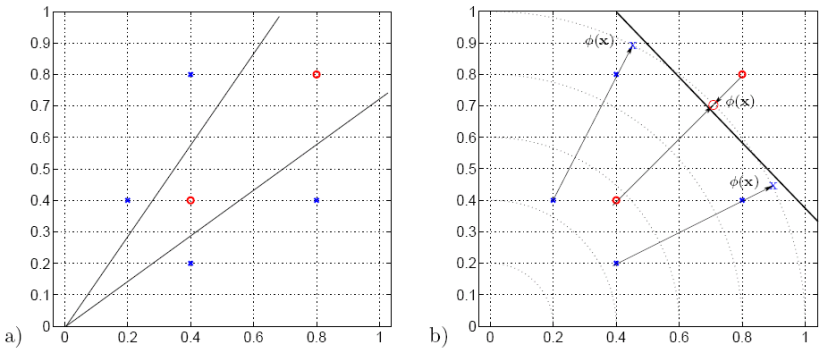
   (sol.) Yes. If linear activation functions are used for all the hidden units, output from hidden units will be written as linear combination of input features. Since these intermediate output serves as input for the final output layer, we can always find an equivalent neural net which does not have any hidden layer as seen in the example above.


1
$w_7 + cw_8w_1 + cw_9w_2$
x_1
$cw_8w_3 + cw_9w_4$
g
x_2
$cw_8w_5 + cw_9w_6$

5 Kernel Method (20%)

Suppose we have six training points from two classes as in Figure (a). Note that we have four points from class 1: $(0.2, 0.4), (0.4, 0.8), (0.4, 0.2), (0.8, 0.4)$ and two points from class 2: $(0.4, 0.4), (0.8, 0.8)$. Unfortunately, the points in Figure (a) cannot be separated by a linear classifier. The kernel trick is to find a mapping of $\mathbf{x}$ to some feature vector $\phi(\mathbf{x})$ such that there is a function K called kernel which satisfies $K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$. And we expect the points of $\phi(\mathbf{x})$ to be linearly separable in the feature space. Here, we consider the following normalized kernel:

$$K(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x}^T \mathbf{x}'}{||\mathbf{x}^T|| ||\mathbf{x}'||}$$



1. (5%) What is the feature vector $\phi(\mathbf{x})$ corresponding to this kernel? Draw $\phi(\mathbf{x})$ for each training point $\mathbf{x}$ in Figure (b), and specify from which point it is mapped.

$$\phi(\mathbf{x}) = \frac{\mathbf{x}}{||\mathbf{x}||}$$

2. (5%) You now see that the feature vectors are linearly separable in the feature space. The maximum-margin decision boundary in the feature space will be a line in $\mathbb{R}^2$, which can be written as $w_1 x + w_2 y + c = 0$. What are the values of the coefficients w_1 and w_2? (Hint: you don't need to compute them.)

   (sol.)

$$(w_1, w_2) = (1, 1)$$

3. (3%) Circle the points corresponding to support vectors in Figure (b).

4. (7%) Draw the decision boundary in the original input space resulting from the normalized linear kernel in Figure (a). Briefly explain your answer.

6 VC Dimension and PAC Learning(10%)

The VC dimension, $VC(H)$, of hypothesis space H defined over instance space X is the size of the largest largest number of points (in some configuration) that can be shattered by H. Suppose with probability $(1-\delta)$, a PAC learner outputs a hypothesis within error ϵ of the best possible hypothesis in H. It can be shown that the lower bound on the number of training examples m sufficient for successful learning, stated in terms of $VC(H)$ is

$$m \geq \frac{1}{\epsilon}(4\log_2(2/\delta) + 8VC(H)\log_2(13/\epsilon)).$$

Consider a learning problem in which $X = \mathcal{R}$ is the set of real numbers, and the hypothesis space is the set of intervals $H = \{(a < x < b)|a, b \in \mathcal{R}\}$. Note that the hypothesis labels points inside the interval as positive, and negative otherwise.

1. (5%) What is the VC dimension of H?

   (sol.) $VC(H) = 2$. Suppose we have two points x_1 and x_2, and $x_1 < x_2$. They can always be shattered by H, no matter how they are labled.

   (a) if x_1 positive and x_2 negative, choose $a < x_1 < b < x_2$;

   (b) if x_1 negative and x_2 positive, choose $x_1 < a < x_2 < b$;

   (c) if both x_1 and x_2 positive, choose $a < x_1 < x_2 < b$;

   (d) if both x_1 and x_2 negative, choose $a < b < x_1 < x_2$;

   However, if we have three points $x_1 < x_2 < x_3$ and if they are labeled as x_1 (positive) x_2 (negative) and x_3 (positive), then they cannot be shattered by H.

2. (5%) What is the probability that a hypothesis consistent with m examples will have error at least ϵ?

   (sol.) Use the above result. Substitute $VC(H) = 2$ into the inequality

   $$m \geq \frac{1}{\epsilon}(4\log_2(2/\delta) + 8 * 2\log_2(13/\epsilon)),$$

   we have

   $$\begin{aligned} \epsilon m &\geq 4\log_2(2/\delta) + 8 * 2\log_2(13/\epsilon) \\ \epsilon m - 16\log_2(13/\epsilon) &\geq 4\log_2(2/\delta) \\ \frac{2^{\epsilon m/4}}{(13/\epsilon)^4} &\geq 2/\delta \\ \delta &\geq \frac{\left(\frac{13}{\epsilon}\right)^4}{2^{\epsilon m/4-1}} \end{aligned}$$

7 Logistic Regression (10%)

We consider the following models of logistic regression for a binary classification with a sigmoid function $g(z) = \frac{1}{1+e^{-z}}$:

- Model 1: $P(Y = 1 \mid X, w_1, w_2) = g(w_1 X_1 + w_2 X_2)$
- Model 2: $P(Y = 1 \mid X, w_1, w_2) = g(w_0 + w_1 X_1 + w_2 X_2)$

We have three training examples:

$$x^{(1)} = [1, 1]^T \quad x^{(2)} = [1, 0]^T \quad x^{(3)} = [0, 0]^T$$
$$y^{(1)} = 1 \qquad y^{(2)} = -1 \qquad y^{(3)} = 1$$

1. (5%) Does it matter how the third example is labeled in Model 1? i.e., would the learned value of $\mathbf{w} = (w_1, w_2)$ be different if we change the label of the third example to -1? Does it matter in Model 2? Briefly explain your answer. (Hint: think of the decision boundary on 2D plane.)

   (sol.) It does not matter in Model 1 because $x^{(3)} = (0, 0)$ makes $w_1 x_1 + w_2 x_2$ always zero and hence the likelihood of the model does not depend on the value of $\mathbf{w}$. But it does matter in Model 2.

2. (5%) Now, suppose we train the logistic regression model (Model 2) based on the n training examples $x^{(1)}, \ldots, x^{(n)}$ and labels $y^{(1)}, \ldots, y^{(n)}$ by maximizing the penalized log-likelihood of the labels:

   $$\sum_i \log P(y^{(i)} \mid x^{(i)}, \mathbf{w}) - \frac{\lambda}{2}||\mathbf{w}||^2 = \sum_i \log g(y^{(i)} \mathbf{w}^T x^{(i)}) - \frac{\lambda}{2}||\mathbf{w}||^2$$

   For large λ (strong regularization), the log-likelihood terms will behave as linear functions of $\mathbf{w}$.

   $$\log g(y^{(i)} \mathbf{w}^T x^{(i)})) \approx \frac{1}{2} y^{(i)} \mathbf{w}^T x^{(i)}$$

   Express the penalized log-likelihood using this approximation (with Model 1), and derive the expression for MLE $\hat{\mathbf{w}}$ in terms of λ and training data $\{x^{(i)}, y^{(i)}\}$. Based on this, explain how $\mathbf{w}$ behaves as λ increases. (We assume each $x^{(i)} = (x_1^{(i)}, x_2^{(i)})^T$ and $y^{(i)}$ is either 1 or -1)

   (sol.)

   $$\log l(\mathbf{w}) \approx \sum_i \frac{1}{2} y^{(i)} \mathbf{w}^T x^{(i)} - \frac{\lambda}{2}||w||^2$$
   $$\frac{\partial}{\partial w_1} \log l(\mathbf{w}) \approx \frac{1}{2} \sum_i y^{(i)} x_1^{(i)} - \lambda w_1 = 0$$
   $$\frac{\partial}{\partial w_2} \log l(\mathbf{w}) \approx \frac{1}{2} \sum_i y^{(i)} x_2^{(i)} - \lambda w_2 = 0$$
   $$\therefore \quad \mathbf{w} = \frac{1}{2\lambda} \sum_i y^{(i)} \mathbf{x}^{(i)}$$

15-781 Midterm Example Questions

1 Short Answer

(a) **(True or False?)** If $P(A|B) = P(A)$ then $P(A \wedge B) = P(A)P(B)$.

True

(b) What is the entropy of the following probability distribution: $[0.0625, 0.0625, 0.125, 0.25, 0.5]$?

1 7/8

(c) **(True or False?)** Because decision trees learn to classify discrete-valued outputs instead of real-valued functions it is impossible for them to overfit.

False

(d) **(True or False?)** Assuming a fixed number of attributes, a Gaussian-based Bayes optimal classifier can be learned in time linear in the number of records in the dataset.

True

2 Decision Trees

You are stranded on a deserted island. Mushrooms of various types grow wildly all over the island, but no other food is anywhere to be found. Some of the mushrooms have been determined as poisonous and others as not (determined by your former companions' trial and error). You are the only one remaining on the island. You have the following data to consider.

Example	IsHeavy	IsSmelly	IsSpotted	IsSmooth	IsPoisonous
A	0	0	0	0	0
B	0	0	1	0	0
C	1	1	0	1	0
D	1	0	0	1	1
E	0	1	1	0	1
F	0	0	1	1	1
G	0	0	0	1	1
H	1	1	0	0	1
U	1	1	1	1	?
V	0	1	0	1	?
W	1	1	0	0	?

You know whether or not mushrooms A through H are poisonous, but you do not know about U through W. For the first couple of questions, consider only mushrooms A through H.

(a) What is the entropy of IsPoisonous?

```
-5/8 log 5/8 - 3/8 log 3/8 = 0.9544
```

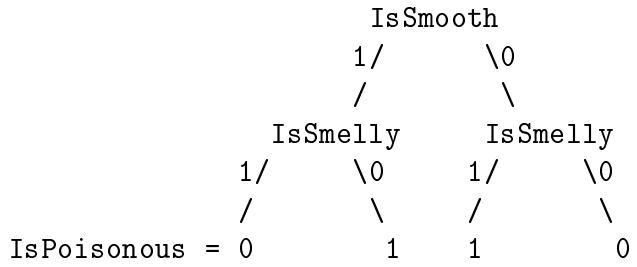
(b) Which attribute should you choose as the root of a decision tree? Hint: You can figure this out by looking at the data without explicitly computing the information gain of all four attributes.

```
IsSmooth
```

(c) What is the information gain of the attribute you chose in the previous question?

```
approximately 0.0487
```

(d) Build a decision tree to classify mushrooms as poisonous or not.



There are other valid solutions since it only asks for ‘‘a’’ decision tree and doesn’t ask for an ID3 decision tree.

(e) Classify mushrooms U, V, and W using this decision tree as poisonous or not poisonous.

U and V both classify as ‘‘not poisonous’’.
W classifies as ‘‘poisonous’’.

Your solution to this might be different depending on the decision tree of the previous question.

(f) If the mushrooms of A through H that you know are not poisonous suddenly became scarce, should you consider trying U, V, and W? Which one(s) and why? Or if none of them, then why not?

The answer we were going for here should have mentioned the small number of examples and that all of U, V, and W can be seen as ‘‘risky’’ due to the small training set. For example, there are other decision trees that are consistent with the training data (other than the one seen in the solution to part d above) for which the classifications of U, V, and W are different.

3 Gaussian Bayes Classifiers

1. Gaussian-based Bayes Classifiers assume that, given n classes, the kth datapoint was generated by first deciding the class of the kth datapoint according to the class prior probabilities, and then choosing the the kth input vector to be generated randomly by a Gaussian distribution with a mean and (usually) a covariance that is dependent on the choice of the class.

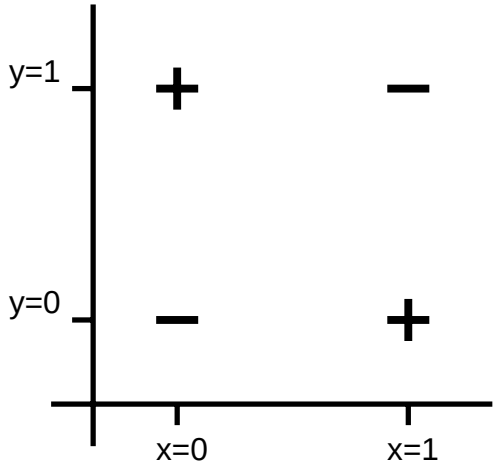
   Describe one or more ways in which this assumption could be wrong in practice.

Many real-valued distributions are not Gaussian

The assumption that they are independently distributed is often wrong. Frequently the next observation may depend on, or be correlated with, the previous observation.

Sometimes, the underlying probabilities drift around while the data is being generated.

2. Consider the XOR problem, in which there are two input attributes x and y which take on the values 0 and 1. The output class is positive if and only if $x \neq y$.



What will happen if you try to train a Gaussian-based Bayes Classifier on such a dataset? Assume that the classifier is able to learn arbitrary covariance matrices.

In theory it is fine...two diagonal thin Gaussians can model this perfectly. (That answer would get full points)

But it happens in this case that the MLE Gaussians would be ill-conditioned (zero determinant) so there are likely to be nasty numerical problems.

3. Is it possible to construct a Bayes classifier for one input x so that when it is used it will predict

   - Class 1 if $x < -1$
   - Class 2 if $-1 < x < 1$
   - Class 1 if $1 < x$?

   If so, how?

Easy. Just use two Gaussians with zero mean but different variances
S1^2 and S2^2. Let the class probabilities be P(Class=1) = P(Class=2) = 0.5.
Then choose the variances so that

S1^2 > S2^2

and

the probability densities are equal when X = 1.

3. Explain or sketch an example of a classification problem with
two real-valued inputs and one discrete output in which...

3a: Gaussian Bayes Classifiers would do well on a training set but
badly on a test set.

Any example in which the training points were unlucky and unrepresentative
of the true distribution.

OR, even if the training points aren't unlucky, any situation with
overfitting, e.g. only two training points per class

3b: Gaussian Bayes Classifiers would do well on a test set but
decision trees would do badly on a test set.

One example would have the class boundaries be diagonal, and have
relatively little data, so that decision trees did not have the
data support to allow them to grow the model boundaries well.

3c: Gaussian Bayes Classifiers would do badly on more than one third
of the test set but decision trees would do nearly perfectly on a test set.

One example: put an 8 by 8 checkerboard over the unit square and
1000000 datapoints uniformly randomly in the square with their color
according to the checkerboard color.

4. PDFS: Give an example of a probability density over a single real-valued
variable in which

p(x) > 0 for all x

P(X == 0) = 0

E[X] = 0

P(X == 1) > 0

One of many possible answers:

Let X be sampled using the following recipe:

With Prob 1/3 set it to -1
With Prob 1/3 set it to +1
With Prob 1/3, draw it from N(0,1)

5. You can expect a question like the Bayesian Gaussian MAP estimation ‘‘intellectual snobs’’ example.

6. A ‘‘Box’’ distribution of a scalar random variable is a PDF with two parameters: L and H (for LO and HIGH) in which

p(x) = 0 if x < L

p(x) = 1/(H-L) if L <= x <= H

p(x) = 0 if x > H

We’ll use the notation X ~ BOX(L,H) to mean that X is a random variable drawn from a Box distribution with parameters L and H

6a: If X ~ BOX(L,H) what is E[X]?

(L+H)/2

6b: If X ~ BOX(L,H) what is Var[X]?

(H-L) * (H-L) / 12

6c: Write P(x < q) as an if-then-else expression involving q, L and H

P(x < q) = 0 if q < L
= (x - L)/(H - L) if L <= q <= H
= 1 if q >= H

6d: Suppose you have data x_1 , x_2 , ... x_R i.i.d. ~ BOX(L,H) and suppose L and H are unknown. What are their MLE values? Explain. (Note this is a case where a careful few sentences explaining your answer may be better than an attempt at a proof by classic differentiation of log-likelihood)

MLE is L = min_i x_i
       H = max_i x_i

You can't increase L any further because you'd get a LL of -infinity
If you decrease it you'll just make the height of the box lower and
penalize all the log-likelihoods.

Similar argument for H.

7: Imagine you are going to learn a Naive Bayes classifier for
the following data. Imagine you'll use the Box distribution described
above for the real-valued parameter. Once you've learned the classifier,
what is P(Happy=True | Occupation=Professor ^ Age=36) according to the
classifier?

Inputs		Output
Age	Occupation	Happy
20	CTO	No
40	Prof	No
50	CTO	Yes
30	Prof	Yes
50	Prof	Yes

ANSWER: Our MLE Bayes Classifier is

p(age|happy) = 1/20 if age is in [30,50] and 0 otherwise
p(age|~happy) = 1/20 if age is in [20,40] and 0 otherwise
P(prof|happy) = 2/3
P(prof|~happy) = 1/2
P(happy) = 3/5

Thus we get (by making intensive use of the naive assumption)...

P(happy ^ prof ^ age=36) = P(prof|happy) * p(age=36|happy) * P(happy)
                         = 1/50

P(~happy ^ prof ^ age=36) = P(prof|~happy) * p(age=36|~happy) * P(~happy)
                          = 1/100

So P(happy | prof ^ age=36) = 1/50 / (1/50 + 1/100) = 2/3

10-701/15-781 Machine Learning - Midterm Exam, Fall 2010

Aarti Singh
Carnegie Mellon University

1. Personal info:
   - Name:
   - Andrew account:
   - E-mail address:
2. There should be **15** numbered pages in this exam (including this cover sheet).
3. You can use any material you brought: any book, class notes, your print outs of class materials that are on the class website, including annotated slides and relevant readings, and Andrew Moore's tutorials. You cannot use materials brought by other students. Calculators are not necessary. Laptops, PDAs, phones and Internet access are not allowed.
4. If you need more room to work out your answer to a question, use the back of the page and clearly mark on the front of the page if we are to look at what's on the back.
5. Work efficiently. Some questions are easier, some more difficult. Be sure to give yourself time to answer all of the easy ones, and avoid getting bogged down in the more difficult ones before you have answered the easier ones.
6. You have **90** minutes.
7. Good luck!

Question	Topic	Max. score	Score
1	Short questions	20	
2	Bayes Optimal Classification	15	
3	Logistic Regression	18	
4	Regression	16	
5	SVM	16	
6	Boosting	15	
	Total	100	

1 Short Questions [20 pts]

Are the following statements True/False? Explain your reasoning in only 1 sentence.

1. Density estimation (using say, the kernel density estimator) can be used to perform classification.

   True: Estimate the joint density $P(Y, X)$, then use it to calculate $P(Y|X)$.

2. The correspondence between logistic regression and Gaussian Naïve Bayes (with identity class covariances) means that there is a one-to-one correspondence between the parameters of the two classifiers.

   False: Each LR model parameter corresponds to a whole set of possible GNB classifier parameters, there is no one-to-one correspondence because logistic regression is discriminative and therefore doesn't model $P(X)$, while GNB does model $P(X)$.

3. The training error of 1-NN classifier is 0.

   True: Each point is its own neighbor, so 1-NN classifier achieves perfect classification on training data.

4. As the number of data points grows to infinity, the MAP estimate approaches the MLE estimate for all possible priors. In other words, given enough data, the choice of prior is irrelevant.

   False: A simple counterexample is the prior which assigns probability 1 to a single choice of parameter θ.

5. Cross validation can be used to select the number of iterations in boosting; this procedure may help reduce overfitting.

   True: The number of iterations in boosting controls the complexity of the model, therefore, a model selection procedure like cross validation can be used to select the appropriate model complexity and reduce the possibility of overfitting.

6. The kernel density estimator is equivalent to performing kernel regression with the value $Y_i = \frac{1}{n}$ at each point X_i in the original data set.

   False: Kernel regression predicts the value of a point as the weighted average of the values at nearby points, therefore if all of the points have the same value, then kernel regression will predict a constant (in this case, $\frac{1}{n}$) for all values.

7. We learn a classifier f by boosting weak learners h. The functional form of f's decision boundary is the same as h's, but with different parameters. (e.g., if h was a linear classifier, then f is also a linear classifier).

   False: For example, the functional form of a decision stump is a single axis-aligned split of the input space, but the functional form of the boosted classifier is linear combinations of decision stumps which can form a more complex (piecewise linear) decision boundary.

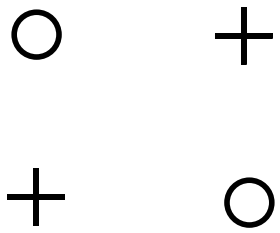
8. The depth of a learned decision tree can be larger than the number of training examples used to create the tree.

   False: Each split of the tree must correspond to at least one training example, therefore, if there are n training examples, a path in the tree can have length at most n.

   Note: There is a pathological situation in which the depth of a learned decision tree can be larger than number of training examples n - if the number of features is larger than n and there exist training examples which have same feature values but different labels. Points have been given if you answered true and provided this explanation.

For the following problems, circle the correct answers:

1. Consider the following data set:



   Circle all of the classifiers that will achieve zero training error on this data set. (You may circle more than one.)

   (a) Logistic regression

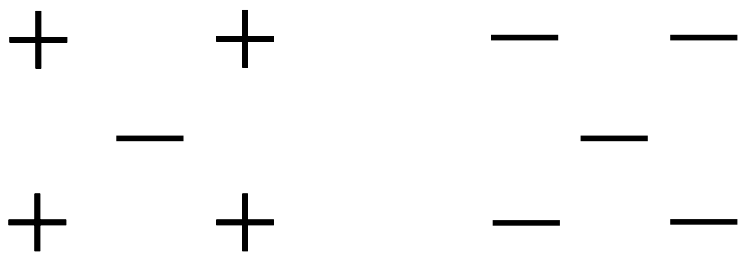
   (b) SVM (quadratic kernel)

   (c) Depth-2 ID3 decision trees

   (d) 3-NN classifier

   Solution: SVM (quad kernel) and Depth-2 ID3 decision trees

2. For the following dataset, circle the classifier which has larger Leave-One-Out Cross-validation error.



a) 1-NN

b) 3-NN

Solution: 1-NN since 1-NN CV err: 5/10, 3-NN CV err: 1/10

2 Bayes Optimal Classification [15 pts]

In classification, the loss function we usually want to minimize is the 0/1 loss:

$$\ell(f(x), y) = \mathbf{1}\{f(x) \neq y\}$$

where $f(x), y \in \{0, 1\}$ (i.e., binary classification). In this problem we will consider the effect of using an asymmetric loss function:

$$\ell_{\alpha,\beta}(f(x), y) = \alpha\mathbf{1}\{f(x) = 1, y = 0\} + \beta\mathbf{1}\{f(x) = 0, y = 1\}$$

Under this loss function, the two types of errors receive different weights, determined by $\alpha, \beta > 0$.

1. **[4 pts]** Determine the Bayes optimal classifier, i.e. the classifier that achieves minimum risk assuming $P(x, y)$ is known, for the loss $\ell_{\alpha,\beta}$ where $\alpha, \beta > 0$.

   Solution: We can write

$$\begin{aligned}
\arg\min_f \mathbb{E}\ell_{\alpha,\beta}(f(x), y) &= \arg\min_f \mathbb{E}_{X,Y}[\alpha\mathbf{1}\{f(X) = 1, Y = 0\} + \beta\mathbf{1}\{f(X) = 0, Y = 1\}] \\
&= \arg\min_f \mathbb{E}_X[\mathbb{E}_{Y|X}[\alpha\mathbf{1}\{f(X) = 1, Y = 0\} + \beta\mathbf{1}\{f(X) = 0, Y = 1\}]] \\
&= \arg\min_f \mathbb{E}_X[\int_y \alpha\mathbf{1}\{f(X) = 1, y = 0\} + \beta\mathbf{1}\{f(X) = 0, y = 1\} dP(y|x)] \\
&= \arg\min_f \int_x [\alpha\mathbf{1}\{f(x) = 1\}P(y = 0|x) + \beta\mathbf{1}\{f(x) = 0\}P(y = 1|x)] dP(x)
\end{aligned}$$

   We may minimize the integrand at each x by taking:

$$f(x) = \begin{cases} 1 & \beta P(y = 1|x) \geq \alpha P(y = 0|x) \\ 0 & \alpha P(y = 0|x) > \beta P(y = 1|x). \end{cases}$$

2. **[3 pts]** Suppose that the class $y = 0$ is extremely uncommon (i.e., $P(y = 0)$ is small). This means that the classifier $f(x) = 1$ for all x will have good risk. We may try to put the two classes on even footing by considering the risk:

$$R = P(f(x) = 1|y = 0) + P(f(x) = 0|y = 1)$$

   Show how this risk is equivalent to choosing a certain α, β and minimizing the risk where the loss function is $\ell_{\alpha,\beta}$.

   Solution: Notice that

$$\begin{aligned}
E\ell_{\alpha,\beta}(f(x), y) &= \alpha P(f(x) = 1, y = 0) + \beta P(f(x) = 0, y = 1) \\
&= \alpha P(f(x) = 1|y = 0)P(y = 0) + \beta P(f(x) = 0|y = 1)P(y = 1)
\end{aligned}$$

   which is same as the minimizer of the given risk R if $\alpha = \frac{1}{P(y=0)}$ and $\beta = \frac{1}{P(y=1)}$.

3. [**4 pts**] Consider the following classification problem. I first choose the label $Y \sim$ Bernoulli($\frac{1}{2}$), which is 1 with probability $\frac{1}{2}$. If $Y = 1$, then $X \sim$ Bernoulli(p); otherwise, $X \sim$ Bernoulli(q). Assume that $p > q$. What is the Bayes optimal classifier, and what is its risk?

Solution: Since label is equally likely to be 1 or 0, to minimize prob of error simply predict the label for which feature value X is most likely. Since $p > q$, $X = 1$ is most likely for $Y = 1$ and $X = 0$ is most likely for $Y = 0$. Hence $f^*(X) = X$. Baye's risk $= P(X \neq Y) = 1/2 \cdot (1 - p) + 1/2 \cdot q$.

Formally: Notice that since $Y \sim$ Bernoulli($\frac{1}{2}$), we have $P(Y = 1) = P(Y = 0) = 1/2$.

$$\begin{aligned} f^*(x) = \arg\max_y P(Y = y|X = x) &= \arg\max_y P(X = x|Y = y)P(Y = y) \\ &= \arg\max_y P(X = x|Y = y) \end{aligned}$$

Therefore, $f^*(1) = 1$ since $p = P(X = 1|Y = 1) > P(X = 1|Y = 0) = q$, and $f^*(0) = 0$ since $1 - p = P(X = 0|Y = 1) < P(X = 0|Y = 0) = 1 - q$. Hence $f^*(X) = X$. The risk is $R^* = P(f^*(X) \neq Y) = P(X \neq Y)$.

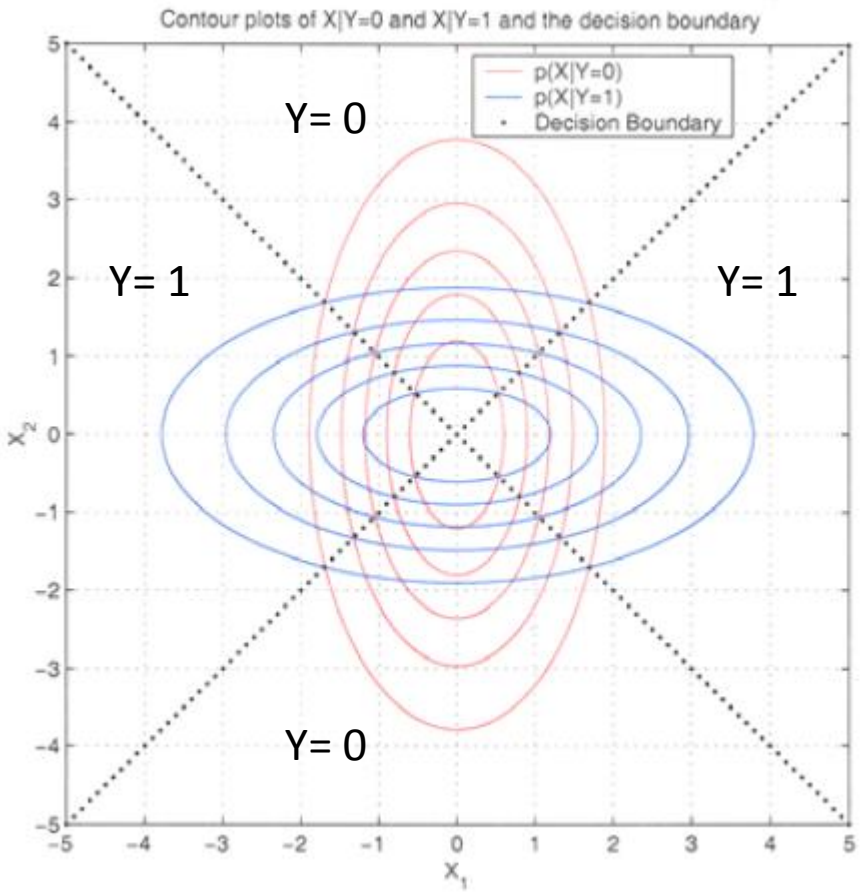
$$R^* = P(Y = 1)P(X = 0|Y = 1) + P(Y = 0)P(X = 1|Y = 0) = \frac{1}{2} \cdot (1 - p) + \frac{1}{2} \cdot q.$$

4. [**4 pts**] Now consider the regular 0/1 loss ℓ, and assume that $P(y = 0) = P(y = 1) = 1/2$. Also, assume that the class-conditional densities are Gaussian with mean μ_0 and co-variance Σ_0 under class 0, and mean μ_1 and co-variance Σ_1 under class 1. Further, assume that $\mu_0 = \mu_1$.

For the following case, draw contours of the level sets of the class conditional densities and label them with $p(x|y = 0)$ and $p(x|y = 1)$. Also, draw the decision boundaries obtained using the Bayes optimal classifier in each case and indicate the regions where the classifier will predict class 0 and where it will predict class 1.

$$\Sigma_0 = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}, \Sigma_1 = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}$$

Solution: next page


Contour plots of X|Y=0 and X|Y=1 and the decision boundary
p(X|Y=0)
p(X|Y=1)
Decision Boundary
Y= 0
Y= 1
Y= 1
Y= 0
X2
X1

3 Logistic Regression [18 pts]

We consider here a discriminative approach for solving the classification problem illustrated in Figure 1.

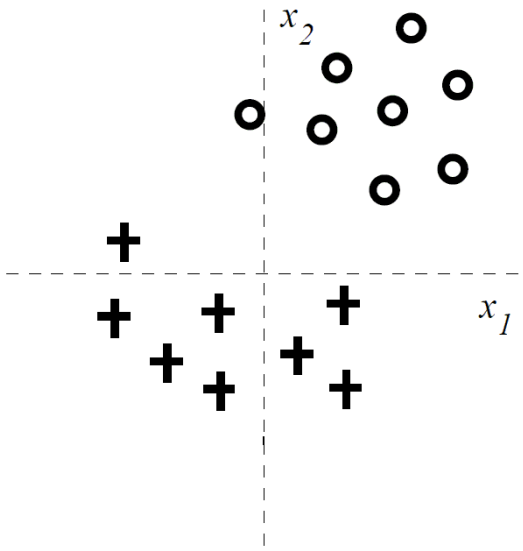


Figure 1: The 2-dimensional labeled training set, where '+' corresponds to class y=1 and 'O' corresponds to class $y = 0$.

1. We attempt to solve the binary classification task depicted in Figure 1 with the simple linear logistic regression model

$$P(y = 1|\vec{x}, \vec{w}) = g(w_0 + w_1x_1 + w_2x_2) = \frac{1}{1 + exp(-w_0 - w_1x_1 - w_2x_2)}.$$

Notice that the training data can be separated with *zero* training error with a linear separator.

Consider training *regularized* linear logistic regression models where we try to maximize

$$\sum_{i=1}^{n} \log\ (P(y_i|x_i, w_0, w_1, w_2)) - Cw_j^2$$

for very large C. The regularization penalties used in penalized conditional log-likelihood estimation are $-Cw_j^2$, where $j = \{0, 1, 2\}$. In other words, only one of the parameters is regularized in each case. Given the training data in Figure 1, how does the training error change with regularization of each parameter w_j? State whether the training error increases or stays the same (zero) for each w_j for very large C. Provide a brief justification for each of your answers.

(a) By regularizing w_2 **[2 pts]**

SOLUTION: Increases. When we regularize w_2, the resulting boundary can rely less and less on the value of x_2 and therefore becomes more vertical. For very large C, the training error increases as there is no good linear vertical separator of the training data.

(b) By regularizing w_1 **[2 pts]**

SOLUTION: Remains the same. When we regularize w_1, the resulting boundary can rely less and less on the value of x_1 and therefore becomes more horizontal and the training data can be separated with *zero* training error with a horizontal linear separator.

(c) By regularizing w_0 **[2 pts]**

SOLUTION: Increases. When we regularize w_0, then the boundary will eventually go through the origin (bias term set to zero). Based on the figure, we can *not* find a linear boundary through the origin with *zero* error. The best we can get is one error.

2. If we change the form of regularization to L1-norm (absolute value) and regularize w_1 and w_2 only (but not w_0), we get the following penalized log-likelihood

$$\sum_{i=1}^{n} \log P(y_i|x_i, w_0, w_1, w_2) - C(|w_1| + |w_2|).$$

Consider again the problem in Figure 1 and the same linear logistic regression model $P(y = 1|\vec{x}, \vec{w}) = g(w_0 + w_1x_1 + w_2x_2)$.

(a) **[3 pts]** As we increase the regularization parameter C which of the following scenarios do you expect to observe? (Choose only one) Briefly explain your choice:

( ) First w_1 will become 0, then w_2.

( ) First w_2 will become 0, then w_1.

( ) w_1 and w_2 will become zero simultaneously.

( ) None of the weights will become exactly zero, only smaller as C increases.

SOLUTION: First w_1 will become 0, then w_2.

The data can be classified with zero training error and therefore also with high log-probability by looking at the value of x_2 alone, i.e. making $w_1 = 0$. Initially we might prefer to have a non-zero value for w_1 but it will go to zero rather quickly as we increase regularization. Note that we pay a regularization penalty for a non-zero value of w_1 and if it does not help classification why would we pay the penalty? Also, the absolute value regularization ensures that w_1 will indeed go to *exactly* zero. As C increases further, even w_2 will eventually become zero. We pay higher and higher cost for setting w_2 to a non-zero value. Eventually this cost overwhelms the gain from the log-probability of labels that we can achieve with a non-zero w_2.

(b) **[3 pts]** For very large C, with the same L1-norm regularization for w_1 and w_2 as above, which value(s) do you expect w_0 to take? Explain briefly. (Note that the number of points from each class is the same.) (You can give a range of values for w_0 if you deem necessary).

SOLUTION: For very large C, we argued that both w_1 and w_2 will go to zero. Note that when $w_1 = w_2 = 0$, the log-probability of labels becomes a finite value, which is equal to n log(0.5), i.e. $w_0 = 0$. In other words, $P(y = 1|\vec{x}, \vec{w})$=$P(y = 0|\vec{x}, \vec{w})$=0.5. We expect so because the number of elements in each class is the same and so we would like to predict each one with the same probability, and w_0=0 makes $P(y = 1|\vec{x}, \vec{w})$=0.5.

(c) **[3 pts]** Assume that we obtain more data points from the '+' class that corresponds to y=1 so that the class labels become unbalanced. Again for very large C, with the same L1-norm regularization for w_1 and w_2 as above, which value(s) do you expect w_0 to take? Explain briefly. (You can give a range of values for w_0 if you deem necessary).

SOLUTION: For very large C, we argued that both w_1 and w_2 will go to zero. With unbalanced classes where the number of '+' labels are greater than that of 'o' labels, we want to have $P(y = 1|\vec{x}, \vec{w}) > P(y = 0|\vec{x}, \vec{w})$. For that to happen the value of w_0 should be greater than zero which makes $P(y = 1|\vec{x}, \vec{w}) > 0.5$.

4 Kernel regression [16 pts]

Now lets consider the non-parametric kernel regression setting. In this problem, you will investigate univariate locally linear regression where the estimator is of the form:

$$\widehat{f}(x) = \beta_1 + \beta_2 x$$

and the solution for parameter vector $\beta = [\beta_1 \ \beta_2]$ is obtained by minimizing the weighted least square error:

$$J(\beta_1, \beta_2) = \sum_{i=1}^{n} W_i(x)(Y_i - \beta_1 - \beta_2 X_i)^2 \quad \text{where} \quad W_i(x) = \frac{K\left(\frac{X_i - x}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right)},$$

where K is a kernel with bandwidth h. Observe that the weighted least squares error can be expressed in matrix form as

$$J(\beta_1, \beta_2) = (Y - A\beta)^T W (Y - A\beta),$$

where Y is a vector of n labels in the training example, W is a $n \times n$ diagonal matrix with weight of each training example on the diagonal, and

$$A = \begin{bmatrix} 1 \ X_1 \\ 1 \ X_2 \\ \cdots \\ 1 \ X_n \end{bmatrix}$$

1. [**4 pts**] Derive an expression in matrix form for the solution vector $\hat{\beta}$ that minimizes the weighted least square.

   Solution: Differentiating the objective function wrt β, we have:

   $$\frac{\partial J(\beta)}{\beta} = 2A^T W A\beta - 2A^T W^T Y.$$

   Therefore, the solution $\hat{\beta}$ satisfies the following normal equations:

   $$A^T W A\beta = A^T W^T Y$$

   And if $A^T W A$ is invertible, then the solution is $\hat{\beta} = (A^T W A)^{-1} A^T W^T Y$. (Note that $W = W^T$, so the solution can be written in terms of either).

2. [**3 pts**] When is the above solution unique?

   Solution: When $A^T W A$ is invertible. Since W is a diagonal matrix, $A^T W A = (W^{1/2}A)^T (W^{1/2}A)$ and hence rank$(A^T W A) = \min(n, 2)$ - Refer TK's recitation notes. Since a matrix is invertible if it is full rank, a unique solution exists if $n \geq 2$.

3. [**3 pts**] If the solution is not unique, one approach is to optimize the objective function J using gradient descent. Write the update equation for gradient descent in this case. Note: Your answer must be expressed in terms of the matrices defined above.

Solution: Let $\alpha > 0$ denote the step-size.

$$\begin{aligned}\beta^{(t+1)} &= \beta^{(t)} - \frac{\alpha}{2}\frac{\partial J(\beta)}{\partial \beta} \\ &= \beta^{(t)} - \alpha A^T W(A\beta - Y)\end{aligned}$$

4. [**3 pts**] Can you identify the signal plus noise model under which maximizing the likelihood (MLE) corresponds to the weighted least squares formulation mentioned above?

   Solution: $Y = \beta_1 + \beta_2 X + \epsilon$, where $\epsilon_i \overset{iid}{\sim} \mathcal{N}(0, \sigma_i^2)$ for $i = 1, \ldots, n$. Here $\sigma_i^2 \propto 1/W_i(x)$.

5. [**3 pts**] Why is the above setting non-parametric? Mention one advantage and one disadvantage of nonparametric techniques over parametric techniques.

   Solution: The above setting is non-parametric since it performs locally linear fits, therefore number of parameters scale with data. Notice that $W_i(x)$, and hence the solution $\hat{\beta}$, depends on x. Thus we are fitting the parameters to every point x - therefore total number of parameters can be larger than n.

   Nonparametric techniques do not place very strict assumptions on the form of the underlying distribution or regression function, but are typically computationally expensive and require large number of training examples.

5 SVM [16 pts]

5.1 L2 SVM

Let $\{(\mathbf{x}_i, y_i)\}_{i=1}^l$ be a set of l training pairs of feature vectors and labels. We consider binary classification, and assume $y_i \in \{-1, +1\}\ \forall i$. The following is the primal formulation of L2 SVM, a variant of the standard SVM obtained by squaring the hinge loss:

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\mathbf{w}^\top\mathbf{w} + \frac{C}{2}\sum_{i=1}^{l}\xi_i^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top\mathbf{x}_i + b) \ \geq\ 1 - \xi_i, \quad i \in \{1, \ldots, l\}, \\ & \xi_i \ \geq\ 0, \quad i \in \{1, \ldots, l\}. \end{aligned}$$

1. **[4 pts]** Show that removing the last set of constraints $\{\xi_i \geq 0\ \forall i\}$ does not change the optimal solution to the primal problem.

   Solution: Let $(\mathbf{w}^*, b^*, \boldsymbol{\xi}^*)$ be the optimal solution to the problem without the last set of constraints. It suffices to show that $\xi_i^* \geq 0\ \forall i$. Suppose it is not the case, then there exists some $\xi_j^* < 0$. Then we have

   $$y_j((\mathbf{w}^*)^\top\mathbf{x}_j + b^*) \ \geq\ 1 - \xi_j^* \ >\ 1,$$

   implying that $\xi_j' = 0$ is a feasible solution and yet gives a smaller objective value since $(\xi_j')^2 = 0 < (\xi_j^*)^2$, a contradiction to the assumption that ξ_j^* is optimal.

2. **[3 pts]** After removing the last set of constraints, we get a simpler problem:

   $$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\mathbf{w}^\top\mathbf{w} + \frac{C}{2}\sum_{i=1}^{l}\xi_i^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top\mathbf{x}_i + b) \ \geq\ 1 - \xi_i, \quad i \in \{1, \ldots, l\}. \end{aligned} \tag{1}$$

   Give the Lagrangian of (1).

   Solution: The Lagrangian is

   $$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) \ :=\ \frac{1}{2}\mathbf{w}^\top\mathbf{w} + \frac{C}{2}\sum_{i=1}^{l}\xi_i^2 - \sum_{i=1}^{l}\alpha_i(y_i(\mathbf{w}^\top\mathbf{x}_i + b) - 1 + \xi_i),$$

   where $\alpha_i \geq 0,\ \forall i$ are the Lagrange multipliers.

3. **[6 pts]** Derive the dual of (1). How is it different from the dual of the standard SVM with the hinge loss?

Solution: Taking partial derivatives of the Lagrangian wrt $\mathbf{w}$, b and ξ_i,

$$\nabla_{\mathbf{w}} L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) \;=\; 0 \iff \mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i \;,$$

$$\partial_b L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) \;=\; 0 \iff \sum_{i=1}^{l} \alpha_i y_i \;=\; 0,$$

$$\partial_{\xi_i} L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) \;=\; 0 \iff \xi_i = \alpha_i / C \;.$$

Plugging these back to the Lagrangian, rearranging terms and keeping constraints on the Lagrange multipliers we obtain the dual

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & -\frac{1}{2}\boldsymbol{\alpha}^\top (Q + I/C)\boldsymbol{\alpha} + \mathbf{1}^\top \boldsymbol{\alpha} \\ \text{s.t.} \quad & \mathbf{y}^\top \boldsymbol{\alpha} \;=\; 0, \quad \alpha_i \;\geq\; 0 \;\forall i, \end{aligned}$$

where $\mathbf{1}$ is a vector of ones, I is the identity matrix, $\mathbf{y}$ is the vector of labels y_i's, and Q is the l-by-l kernel matrix such that $Q_{ij} = y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$. Compared with the dual of the standard SVM, the quadratic term is regularized by an additional positive diagonal matrix, and thus has stronger convexity leading to faster convergence. The other difference is that the dual variables here are only bounded from below, but in the standard SVM the dual variables are bounded both from above (by C) and from below. In fact, for L2 svms the solution does not depend on the tradeoff parameter C.

5.2 Leave-one-out Error and Support Vectors

[3 pts] Consider the standard two-class SVM with the hinge loss. Argue that under a given value of C,

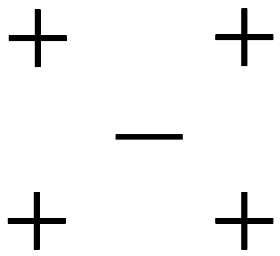
$$\text{LOO error} \;\leq\; \frac{\#\text{SVs}}{l},$$

where l is the size of the training data and #SVs is the number of support vectors obtained by training SVM on the entire set of training data.

Solution: Since the decision function only depends on the support vectors, removing a non-support vector from the training data and then re-training an SVM would lead to the same decision function. Also, non-support vectors must be classified correctly. As a result, errors found in the leave-one-out validation must be caused by removing the support vectors, proving the desired result.

6 Boosting [15 pts]

1. Consider training a boosting classifier using decision stumps on the following data set:

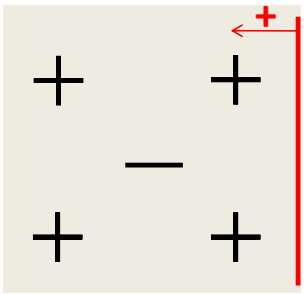


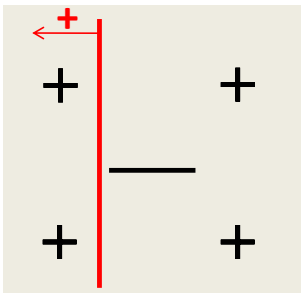
(a) **[3 pts]** Which examples will have their weights increased at the end of the first iteration? Circle them.

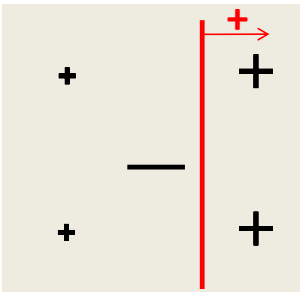
Solution: The negative example since the decision stump with least error in first iteration is constant over the whole domain. Notice this decision stump only predicts incorrectly on the negative example, whereas any other decision stump predicts incorrectly on at least two training examples.

(b) **[3 pts]** How many iterations will it take to achieve zero training error? Explain.

Solution: At least three iterations. The first iteration misclassifies the negative example, the second iteration misclassifies two of the positive examples as the negative one has large weight. The third iteration is needed since a weighted sum of the first two decision stumps can't yield zero training error, and misclassifies the other two positive examples. See Figures below.







(c) **[3 pts]** Can you add one more example to the training set so that boosting will achieve zero training error in two steps? If not, explain why.

Solution: No. Notice that the simplest case is adding one more negative example in center or one more positive example between any two positive examples, as it still yields three decision regions with axis-aligned boundaries. If only two steps were enough, then a linear combination of only two decision stumps $sign(\alpha_1 h_1(x) + \alpha_2 h_2(x))$

should be able to yield three decision regions. Also notice that at least one of h_1 or h_2 misclassifies two positive examples. If only h_2 misclassifies two positive examples, the possible decisions are (1) $sign(\alpha_1 - \alpha_2)$ on those two positive examples, (2) $sign(\alpha_1 + \alpha_2)$ on the remaining positive examples and (3) $sign(\alpha_1 - \alpha_2)$ on the negative examples - which don't yield zero training error since signs on (1) and (3) agree. If both h_1 and h_2 misclassify two positive examples, we have (1) $sign(\alpha_1 - \alpha_2)$ on two positive examples, (2) $sign(-\alpha_1 + \alpha_2)$ on the remaining positive examples and (3) $sign(-\alpha_1 - \alpha_2)$ on the negative - which again don't yield zero training error since signs on (1) and (2) don't agree.

2. [**2 pts**] Why do we want to use "weak" learners when boosting?

   Solution: To prevent overfitting, since the complexity of the overall learner increases at each step. Starting with weak learners implies the final classifier will be less likely to overfit.

3. [**4 pts**] Suppose AdaBoost is run on m training examples, and suppose on each round that the weighted training error ϵ_t of the t^{th} weak hypothesis is at most $1/2 - \gamma$, for some number $\gamma > 0$. After how many iterations, T, will the combined hypothesis H be consistent with the m training examples, i.e., achieves zero training error? Your answer should only be expressed in terms of m and γ. (Hint: What is the training error when 1 example is misclassified?)

   Solution: Training error when 1 example is misclassified = $1/m$. Therefore, we need to guarantee that training error is $< 1/m$. Since $\epsilon_t \leq 1/2 - \gamma$, from class notes we know that

   $$\text{Training err of the combined hypothesis } H \leq \exp(-2T\gamma^2)$$

   The upper bound is $< 1/m$ if $T > \ln m/2\gamma^2$.

10-701 Midterm Exam, Spring 2011

1. Personal info:
   - Name:
   - Andrew account:
   - E-mail address:
2. There are 14 numbered pages in this exam (including this cover sheet).
3. You can use any material you brought: any book, notes, and print outs. You cannot use materials brought by other students.
4. No computers, PDAs, phones or Internet access.
5. If you need more room to answer a question, use the back of the preceding page.
6. Work efficiently. Consider answering all of the easier questions first.
7. There is one *optional extra credit question*, which will *not* affect the grading curve. It will be used to bump your grade up, without affecting anyone else's grade.
8. You have 80 minutes, the test has 100 points. Good luck!

Question	Topic	Max. score	Score
1	Short Questions	20	
2	Bayes Nets	23	
3	Decision Surfaces and Training Rules	12	
4	Linear Regression	20	
5	Conditional Independence Violation	25	
6	[Extra Credit] Violated Assumptions	6	

1 [20 Points] Short Questions

1.1 True or False (Grading: Carl Doersch)

Answer each of the following True of False. If True, give a short justification. If False, a counter-example or convincing one-sentence explanation.

1. [2 pts] If we train a Naive Bayes classifier using infinite training data that satisfies all of its modeling assumptions (e.g., conditional independence), then it will achieve zero *training error* over these training examples.

   ★ **SOLUTION:** **False**. There will still be unavoidable error. In Naive Bayes, Y is probabilistic, so it is often impossible to predict Y even if the model's estimate of $P(Y)$ is perfect. Furthermore, Naive Bayes is linear, and so it can't necessarily even estimate $P(Y)$ perfectly: for example, in the distribution $Y = 1 \Leftrightarrow X_1 \text{XOR} X_2$.

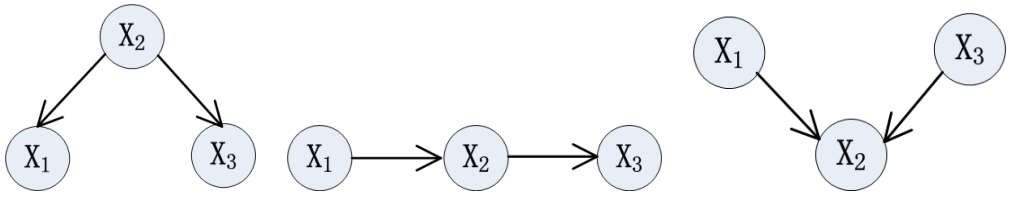
2. [2 pts] If we train a Naive Bayes classifier using infinite training data that satisfies all of its modeling assumptions (e.g., conditional independence), then it will achieve zero *true error* over test examples drawn from this same distribution.

   ★ **SOLUTION:** **False**, for the same reasons as above.

3. [2 pts] Every Bayes Net defined over 10 variables $\langle X_1, X_2, \ldots X_{10} \rangle$ tells how to factor the joint probability distribution $P(X_1, X_2, \ldots X_{10})$ into the product of exactly 10 terms.

   ★ **SOLUTION:** **True**, by the definition of Bayes Net.

   Consider the three Bayes Nets shown below:



A B C

4. [3 pts] True or false: Every joint distribution $P(X_1, X_2, X_3)$ that can be defined by adding Conditional Probability Distributions (CPD) to Bayes Net graph A can also be expressed by appropriate CPD's for Bayes Net graph B.

★ **SOLUTION:** **True**. If a distribution can be represented in graph A, it will factorize as $P(X_2)P(X_1|X_2)P(X_3|X_2)$. Using Bayes rule, this becomes $P(X_2)P(X_3|X_2)P(X_2|X_1)P(X_1)/P(X_2) = P(X_3|X_2)P(X_2|X_1)P(X_1)$.

5. [3 pts] True or false: Every joint distribution $P(X_1, X_2, X_3)$ that can be defined by adding Conditional Probability Distributions to Bayes Net graph A can also be expressed by appropriate CPD's for Bayes Net graph C.

★ **SOLUTION:** **False**. A can represent distributions where X_1 can depend on X_3 given no information about X_2, whereas graph C cannot.

1.2 Quick questions (Grading: Yi Zhang)

Answer each of the following in one or two sentences, in the space provided.

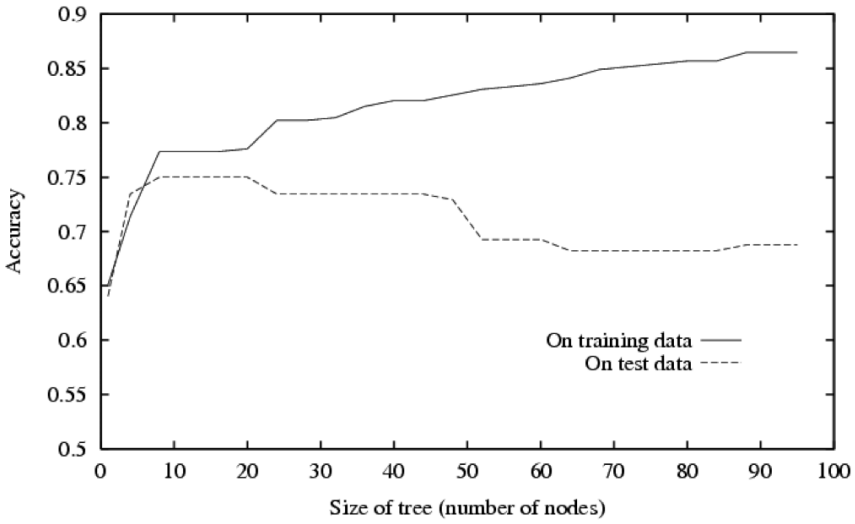
1. [2 pts] Prove that $P(X_1|X_2)P(X_2) = P(X_2|X_1)P(X_1)$. (*Hint:* This is a two-line proof.)

★ **SOLUTION:** $P(X_1|X_2)P(X_2) = P(X_1, X_2) = P(X_2|X_1)P(X_1)$

2. [3 pts] Consider a decision tree learner applied to data where each example is described by 10 boolean variables $\langle X_1, X_2, \ldots X_{10}\rangle$. What is the VC dimension of the hypothesis space used by this decision tree learner?

★ **SOLUTION:** The VC dimension is 2^{10}, because we can shatter 2^{10} examples using a tree with 2^{10} leaf nodes, and we cannot shatter $2^{10} + 1$ examples (since in that case we must have duplicated examples and they can be assigned with conflicting labels).

3. [3 pts] Consider the plot below showing training and test set accuracy for decision trees of different sizes, using the same set of training data to train each tree. Describe in one sentence how the training data curve (solid line) will change if the *number of training examples* approaches infinity. In a second sentence, describe what will happen to the test data curve under the same condition.

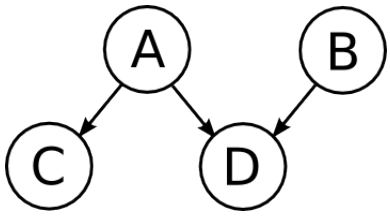


★ **SOLUTION:** The new training accuracy curve should be below the original training curve (since it's impossible for the trees to overfit infinite training data); the new testing accuracy curve should be above the original testing curve and become identical to the new training curve (since trees learned from infinite training data should perform well on testing data and do not overfit at all).

2 [23 Points] Bayes Nets (Grading: Carl Doersch)

2.1 [17 pts] Inference

In the following graphical model, A, B, C, and D are binary random variables.



1. [2 pts] How many parameters are needed to define the Conditional Probability Distributions (CPD's) for this Bayes Net?

   ★ **SOLUTION:** 8: 1 for A, 1 for B, 2 for C, and 4 for D.

2. [2 pts] Write an expression for the probability $P(A = 1, B = 1, C = 1, D = 1)$ in terms of the Bayes Net CPD parameters. Use notation like $P(C = 1|A = 0)$ to denote specific parameters in the CPD's.

   ★ **SOLUTION:**

   $$P(A = 1)P(B = 1)P(C = 1|A = 1)P(D = 1|A = 1, B = 1)$$

3. [3 pts] Write an expression for $P(A = 0|B = 1, C = 1, D = 1)$ in terms of the Bayes Net Conditional Probability Distribution (CPD) parameters.

   ★ **SOLUTION:**

   $$\frac{P(A = 0)P(B = 1)P(C = 1|A = 0)P(D = 1|A = 0, B = 1)}{P(A = 0)P(B = 1)P(C = 1|A = 0)P(D = 1|A = 0, B = 1)+ \\ P(A = 1)P(B = 1)P(C = 1|A = 1)P(D = 1|A = 1, B = 1)}$$

4. [2 pts] True or False (give brief justification): C is conditionally independent of B given D.

★ **SOLUTION:** **False**. There is one path from C to B, and this path isn't blocked at either node.

5. [2 pts] True or False (give brief justification): C is conditionally independent of B given A.

★ **SOLUTION:** **True**. The path is now blocked at both A and D.

Suppose we use EM to train the above Bayes Net from the partially labeled data given below, first initializing all Bayes net parameters to 0.5.

A	B	C	D
1	0	1	0
1	?	0	1
1	1	0	?
0	?	0	?
0	1	0	?

6. [2 pts] How many distinct quantities will be updated during the first M step?

★ **SOLUTION:** **5** or **8**, depending on your interpretation. In the M step we update the values of all parameters, and from part 1 there were 8 parameters. However, only 5 of them will actually be changed if your algorithm's initialization is clever.

7. [2 pts] How many distinct quantities will be estimated during the first E step?

★ **SOLUTION:** **5**. Every unknown value must be estimated.

8. [2 pts] When EM converges, what will be the final estimate for $P(C = 0|A = 1)$? [*Hint:* You do not need a calculator.]

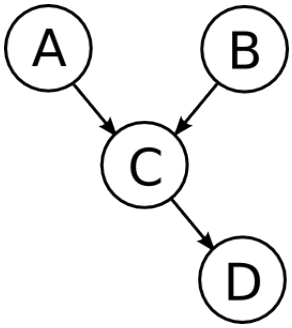
★ **SOLUTION:** **2/3**: the fraction of times when C=0 out of all examples where A=1.

2.2 [6 pts] Constructing a Bayes net

Draw a Bayes net over the random variables $\{A, B, C, D\}$ where the following conditional independence assumptions hold. Here, $X \perp Y | Z$ means X is conditionally independent of Y given Z, and $X \not\perp Y | Z$ means X and Y are not conditionally independent given Z, and $\emptyset$ stands for the empty set.

- $A \perp B | \emptyset$
- $A \not\perp D | B$
- $A \perp D | C$
- $A \not\perp C | \emptyset$
- $B \not\perp C | \emptyset$
- $A \not\perp B | D$
- $B \perp D | A, C$

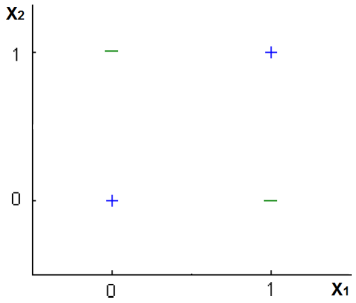
★ **SOLUTION:**



3 [12 Points] Decision Surfaces and Training Rules (Grading: Yi Zhang)

Consider a classification problem with two boolean variables $X_1, X_2 \in \{0, 1\}$ and label $Y \in \{0, 1\}$. In Figure 1 we show two positive ("+") and two negative ("-") examples.

Figure 1: Two positive examples and two negative examples.



Question [2 pts]: Draw (or just simply describe) a decision tree that can perfectly classify the four examples in Figure 1.

★ **SOLUTION:** Split using one variable (e.g., X_1) and then split using the other variable (e.g., X_2). Label each leaf node according to the assigned training example.

Question [3 pts]: In the class we learned the training rule to grow a decision tree: we start from a single root node and iteratively split each node using the "best" attribute selected by maximizing the information gain of the split. We will stop splitting a node if: 1) examples in the node are already pure; or 2) we cannot find any single attribute that gives a split with *positive* information gain. If we apply this training rule to the examples in Figure 1, will we get a decision tree that perfectly classifies the examples? Briefly explain what will happen.

★ **SOLUTION:** We will stop at a single root node and cannot grow the tree any more. This is because, at the root node, splitting on any single variable has zero information gain.

Question [5 pts]: Suppose we learn a Naive Bayes classifier from the examples in Figure 1, using MLE (maximum likelihood estimation) as the training rule. Write down all the parameters and their estimated values (note: both $P(Y)$ and $P(X_i|Y)$ should be Bernoulli distributions). Also, does this learned Naive Bayes perfectly classify the four examples?

★ **SOLUTION:** $P(Y = 1) = 0.5$ $(= P(Y = 0))$
$P(X_1 = 1|Y = 0) = P(X_1 = 1|Y = 1) = 0.5$ $(= P(X_1 = 0|Y = 0) = P(X_1 = 0|Y = 1))$
$P(X_2 = 1|Y = 0) = P(X_2 = 1|Y = 1) = 0.5$ $(= P(X_2 = 0|Y = 0) = P(X_2 = 0|Y = 1))$
This is a very poor classifier since for any X_1, X_2 it will predict $P(Y = 1|X_1, X_2) = P(Y = 0|X_1, X_2) = 0.5$. Naturally, it cannot perfectly classify the examples in the figure.

Question [2 pts]: Is there any logistic regression classifier using X_1 and X_2 that can perfectly classify the examples in Figure 1? Why?

★ **SOLUTION:** No, logistic regression only forms linear decision surface, but the examples in the figure are not linearly separable.

4 [20 Points] Linear Regression (Grading: Xi Chen)

Consider a simple linear regression model in which y is the sum of a deterministic linear function of x, plus random noise ϵ.

$$y = wx + \epsilon$$

where x is the real-valued input; y is the real-valued output; and w is a single real-valued parameter to be learned. Here ϵ is a real-valued random variable that represents noise, and that follows a Gaussian distribution with mean 0 and standard deviation σ; that is, $\epsilon \sim N(0, \sigma)$

(a) [3pts] Note that y is a random variable because it is the sum of a deterministic function of x, plus the random variable ϵ. Write down an expression for the probability distribution governing y, in terms of $N(), \sigma, w$ and x.

★ **SOLUTION:** y follows a Gaussian distribution with the mean wx and the standard deviation σ:

$$p(y|w, x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(y - wx)^2}{2\sigma^2}\right\}$$

(b) [3 pts] You are given n i.i.d. training examples $\{(x^1, y^1), (x^2, y^2), \ldots, (x^n, y^n)\}$ to train this model. Let $\mathcal{Y} = (y^1, \ldots, y^n)$ and $\mathcal{X} = (x^1, \ldots, x^n)$, write an expression for the conditional data likelihood: $p(\mathcal{Y}|\mathcal{X}, w)$.

★ **SOLUTION:**

$$\begin{aligned} p(\mathcal{Y}|\mathcal{X}, w) &= \prod_{i=1}^{n} p(y^i|x^i, w) \\ &= \left(\frac{1}{2\pi\sigma^2}\right)^{n/2} \prod_{i=1}^{n} \exp\left\{-\frac{(y^i - wx^i)^2}{2\sigma^2}\right\} \\ &= \left(\frac{1}{2\pi\sigma^2}\right)^{n/2} \exp\left\{-\frac{\sum_{i=1}^{n}(y^i - wx^i)^2}{2\sigma^2}\right\} \end{aligned}$$

(c) [9 pts] Here you will derive the expression for obtaining a MAP estimate of w from the training data. Assume a Gaussian prior over w with mean 0 and standard deviation τ (i.e. $w \sim N(0, \tau)$). Show that finding the MAP estimate w^* is equivalent to solving the following optimization problem:

$$w^* = \text{argmin}_w \frac{1}{2} \sum_{i=1}^{n} (y^i - wx^i)^2 + \frac{\lambda}{2} w^2;$$

Also express the regularization parameter λ in terms of σ and τ.

★ SOLUTION:

$$\begin{aligned} p(w|\mathcal{Y}, \mathcal{X}) &\propto p(\mathcal{Y}|\mathcal{X}, w) p(w|\mathcal{X}) \\ &\propto \exp\left\{ -\frac{\sum_{i=1}^{n} (y^i - wx^i)^2}{2\sigma^2} \right\} \exp\left\{ -\frac{w^2}{2\tau^2} \right\} \end{aligned}$$

$$\begin{aligned} w^* &= \text{argmax}_w \ln p(w|\mathcal{Y}, \mathcal{X}) \\ &= \text{argmax}_w - \frac{\sum_{i=1}^{n} (y^i - wx^i)^2}{2\sigma^2} - \frac{w^2}{2\tau^2} \\ &= \text{argmin}_w \frac{\sum_{i=1}^{n} (y^i - wx^i)^2}{2\sigma^2} + \frac{w^2}{2\tau^2} \\ &= \text{argmin}_w \frac{1}{2} \sum_{i=1}^{n} (y^i - wx^i)^2 + \frac{\sigma^2}{2\tau^2} w^2 \end{aligned}$$

We can see that $\lambda = \frac{\sigma^2}{\tau^2}$.

(d) [5pts] Above we assumed a zero-mean prior for w, which resulted in the usual $\frac{\lambda}{2} w^2$ regularization term for linear regression. Sometimes we may have prior knowledge that suggests w has some value other than zero. Write down the revised objective function that would be derived if we assume a Gaussian prior on w with mean μ instead of zero (i.e., if the prior is $w \sim N(\mu, \tau)$).

★ SOLUTION:

$$\begin{aligned} p(w|\mathcal{Y}, \mathcal{X}) &\propto p(\mathcal{Y}|\mathcal{X}, w) p(w|\mathcal{X}) \\ &\propto \exp\left\{ -\frac{\sum_{i=1}^{n} (y^i - wx^i)^2}{2\sigma^2} \right\} \exp\left\{ -\frac{(w-\mu)^2}{2\tau^2} \right\} \end{aligned}$$

$$\begin{aligned} w^* &= \text{argmax}_w \ln p(w|\mathcal{Y}, \mathcal{X}) \\ &= \text{argmax}_w - \frac{\sum_{i=1}^{n} (y^i - wx^i)^2}{2\sigma^2} - \frac{(w-\mu)^2}{2\tau^2} \\ &= \text{argmin}_w \frac{\sum_{i=1}^{n} (y^i - wx^i)^2}{2\sigma^2} + \frac{(w-\mu)^2}{2\tau^2} \\ &= \text{argmin}_w \frac{1}{2} \sum_{i=1}^{n} (y^i - wx^i)^2 + \frac{\sigma^2}{2\tau^2} (w-\mu)^2 \end{aligned}$$

5 [25 Points] Conditional Independence Violation (Grading: Yi Zhang)

5.1 Naive Bayes without Conditional Independence Violation

Table 1: $P(Y)$

$Y=0$	$Y=1$
0.8	0.2

Table 2: $P(X_1|Y)$

	$X_1=0$	$X_1=1$
$Y=0$	0.7	0.3
$Y=1$	0.3	0.7

Consider a binary classification problem with variable $X_1 \in \{0,1\}$ and label $Y \in \{0,1\}$. The true generative distribution $P(X_1, Y) = P(Y)P(X_1|Y)$ is shown as Table 1 and Table 2.

Question [4 pts]: Now suppose we have trained a Naive Bayes classifier, using *infinite* training data generated according to Table 1 and Table 2. In Table 3, please write down the predictions from the trained Naive Bayes for different configurations of X_1. Note that $\hat{Y}(X_1)$ in the table is the decision about the value of Y given X_1. For decision terms in the table, write down either $\hat{Y}=0$ or $\hat{Y}=1$; for probability terms in the table, write down the actual values (and the calculation process if you prefer, e.g., $0.8 * 0.7 = 0.56$).

Table 3: Predictions from the trained Naive Bayes

	$\hat{P}(X_1, Y=0)$	$\hat{P}(X_1, Y=1)$	$\hat{Y}(X_1)$
$X_1=0$	$0.8 \times 0.7 = 0.56$	$0.2 \times 0.3 = 0.06$	$\hat{Y}=0$
$X_1=1$	$0.8 \times 0.3 = 0.24$	$0.2 \times 0.7 = 0.14$	$\hat{Y}=0$

★ **SOLUTION:** The naive Bayes model learned from infinite data will have $\hat{P}(Y)$ and $\hat{P}(X_1|Y)$ estimated exactly as Table 1 and Table 2. The resulting predictions are shown in Table 3.

Question [3 pts]: What is the expected error rate of this Naive Bayes classifier on testing examples that are generated according to Table 1 and Table 2? In other words, $P(\hat{Y}(X_1) \neq Y)$ when (X_1, Y) is generated according to the two tables.
Hint: $P(\hat{Y}(X_1) \neq Y) = P(\hat{Y}(X_1) \neq Y, X_1 = 0) + P(\hat{Y}(X_1) \neq Y, X_1 = 1)$.

★ **SOLUTION:**

$$\begin{aligned} P(\hat{Y}(X_1) \neq Y) &= P(\hat{Y}(X_1) \neq Y, X_1 = 0) + P(\hat{Y}(X_1) \neq Y, X_1 = 1) \\ &= P(Y=1, X_1=0) + P(Y=1, X_1=1) \\ &= 0.06 + 0.14 \\ &= 0.2 \end{aligned}$$

5.2 Naive Bayes with Conditional Independence Violation

Consider two variables $X_1, X_2 \in \{0, 1\}$ and label $Y \in \{0, 1\}$. Y and X_1 are still generated according to Table 1 and Table 2, and then X_2 is created as a **duplicated copy** of X_1.

Question [6 pts]: Now suppose we have trained a Naive Bayes classifier, using *infinite* training data that are generated according to Table 1, Table 2 and the duplication rule. In Table 4, please write down the predictions from the trained Naive Bayes for different configurations of (X_1, X_2). For probability terms in the table, you can write down just the calculation process (e.g., one entry might be $0.8 * 0.3 * 0.3 = 0.072$, and you can just write down $0.8 * 0.3 * 0.3$ to save some time). Hint: the Naive Bayes classifier **does** assume that X_2 is conditionally independent of X_1 given Y.

Table 4: Predictions from the trained Naive Bayes

	$\hat{P}(X_1, X_2, Y = 0)$	$\hat{P}(X_1, X_2, Y = 1)$	$\hat{Y}(X_1, X_2)$
$X_1 = 0, X_2 = 0$	$0.8 \times 0.7 \times 0.7$	$0.2 \times 0.3 \times 0.3$	$\hat{Y} = 0$
$X_1 = 1, X_2 = 1$	$0.8 \times 0.3 \times 0.3$	$0.2 \times 0.7 \times 0.7$	$\hat{Y} = 1$
$X_1 = 0, X_2 = 1$	$0.8 \times 0.7 \times 0.3$	$0.2 \times 0.3 \times 0.7$	$\hat{Y} = 0$
$X_1 = 1, X_2 = 0$	$0.8 \times 0.3 \times 0.7$	$0.2 \times 0.7 \times 0.3$	$\hat{Y} = 0$

★ **SOLUTION:** The naive Bayes model learned from infinite data will have $\hat{P}(Y)$ and $\hat{P}(X_1|Y)$ estimated exactly as Table 1 and Table 2. However, it also has $\hat{P}(X_2|Y)$ incorrectly estimated as Table 2. The resulting predictions are shown in Table 4.

Question [3 pts]: What is the expected error rate of this Naive Bayes classifier on testing examples that are generated according to Table 1, Table 2 and the duplication rule?

★ **SOLUTION:** Note that the testing examples are generated according to the true distribution (i.e., where X_2 is a duplication). We have:

$$\begin{aligned} P(\hat{Y}(X_1, X_2) \neq Y) &= P(\hat{Y}(X_1, X_2) \neq Y, X_1 = X_2 = 0) + P(\hat{Y}(X_1, X_2) \neq Y, X_1 = X_2 = 1) \\ &= P(Y = 1, X_1 = X_2 = 0) + P(Y = 0, X_1 = X_2 = 1) \\ &= P(Y = 1, X_1 = 0) + P(Y = 0, X_1 = 1) \\ &= 0.06 + 0.24 \\ &= 0.3 \end{aligned}$$

Question [3 pts]: Compared to the scenario without X_2, how does the expected error rate change (i.e., increase or decrease)? In Table 4, the decision rule $\hat{Y}$ on **which** configuration is responsible to this change? What actually happened to this decision rule? (You need to *briefly* answer: increase or decrease, the responsible configuration, and what happened.)

★ **SOLUTION:** The expected error rate increases from 0.2 to 0.3, due to the incorrect decision $\hat{Y} = 1$ on the configuration $X_1 = X_2 = 1$. Basically the naive Bayes model makes the incorrect conditional independence assumption and considers both $X_1 = 1$ and $X_2 = 1$ as evidence.

5.3 Logistic Regression with Conditional Independence Violation

Question [2 pts]: Will logistic regression suffer from having an additional variable X_2 that is actually a duplicate of X_1? Intuitively, why (hint: model assumptions)?

★ **SOLUTION:** No. Logistic regression does not make conditional independence assumption. (Note: in the class we did derive the form $P(Y|X)$ of logistic regression from naive Bayes assumptions, but that does not mean logistic regression makes the conditional independence assumption).

Now we will go beyond the intuition. We have a training set $\mathbf{D}_1$ of L examples $\mathbf{D}_1 = \{(X_1^1, Y^1), \ldots, (X_1^L, Y^L)\}$. Suppose we generate another training set $\mathbf{D}_2$ of L examples $\mathbf{D}_2 = \{(X_1^1, X_2^1, Y^1), \ldots, (X_1^L, X_2^L, Y^L)\}$, where in each example X_1 and Y are the same as in $\mathbf{D}_1$ and then X_2 is a duplicate of X_1. Now we learn a logistic regression from $\mathbf{D}_1$, which should contain two parameters: w_0 and w_1; we also learn another logistic regression from $\mathbf{D}_2$, which should have three parameters: w_0', w_1' and w_2'.

Question [4 pts] : First, write down the training rule (maximum conditional likelihood estimation) we use to estimate (w_0, w_1) and (w_0', w_1', w_2') from data. Then, given the training rule, what is the relationship between (w_0, w_1) and (w_0', w_1', w_2') we estimated from $\mathbf{D}_1$ and $\mathbf{D}_2$? Use this fact to argue whether or not the logistic regression will suffer from having an additional duplicate variable X_2.

★ **SOLUTION:**

The training rule for (w_0, w_1) is to maximize:

$$\ln \prod_{l=1}^{L} P(Y^l | X_1^l, w_0, w_1) = \sum_{l=1}^{L} Y^l (w_0 + w_1 X_1^l) - \ln(1 + \exp(w_0 + w_1 X_1^l))$$

The training rule for (w_0', w_1', w_2') is to maximize:

$$\ln \prod_{l=1}^{L} P(Y^l | X_1^l, X_2^l, w_0', w_1', w_2') = \sum_{l=1}^{L} Y^l (w_0' + w_1' X_1^l + w_2' X_2^l) - \ln(1 + \exp(w_0' + w_1' X_1^l + w_2' X_2^l))$$

Since X_2 is a duplication of X_1, the training rule for (w_0', w_1', w_2') becomes maximizing:

$$\begin{aligned} & \sum_{l=1}^{L} Y^l (w_0' + w_1' X_1^l + w_2' X_1^l) - \ln(1 + \exp(w_0' + w_1' X_1^l + w_2' X_1^l)) \\ = & \sum_{l=1}^{L} Y^l (w_0' + (w_1' + w_2') X_1^l) - \ln(1 + \exp(w_0' + (w_1' + w_2') X_1^l)) \end{aligned}$$

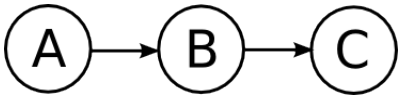
which is basically the same as the training rule for (w_0, w_1), with substitution $w_0 = w'_0$ and $w_1 = w'_1 + w'_2$. This is also the relationship between (w_0, w_1) and (w'_0, w'_1, w'_2) we estimated from $\mathbf{D}_1$ and $\mathbf{D}_2$. As a result, logistic regression will simply split the weight w_1 into $w'_1 + w'_2 = w_1$ when facing duplicated variable $X_2 = X_1$.

6 [Extra Credit 6 pts] Violated assumptions (Grading: Carl Doersch)

Extra Credit Question: This question is optional – do not attempt it until you have completed the rest of the exam. It will not affect the grade curve for the exam, though you will receive extra points if you answer it.

Let A, B, and C be boolean random variables governed by the joint distribution $P(A, B, C)$. Let D be a dataset consisting of n data points, each of which is an independent draw from $P(A, B, C)$, where all three variables are fully observed.

Consider the following Bayes Net, which does not necessarily capture the correct conditional independencies in $P(A, B, C)$.



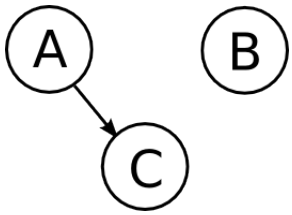
Let $\hat{P}$ be the distribution learned after this Bayes net is trained using D. Show that for any number ϵ, $0 < \epsilon \leq 1$, there exists a joint distribution $P(A, B, C)$ such that $P(C = 1|A = 1) = 1$, but such that the Bayes net shown above, when trained on D, will (with probability 1) learn CPTs where:

$$\hat{P}(C = 1|A = 1) = \sum_{b \in \{0,1\}} \hat{P}(C = 1|B = b)\hat{P}(B = b|A = 1) \leq \epsilon$$

as $|D|$ approaches ∞. Assume that the Bayes net is learning on the basis of the MLE.

You should solve this problem by defining a distribution with the above property. Your final solution may be either in the form of a fully specified joint distribution (i.e. you write out the probabilities for each assignment of the variables A, B, and C), or in the form of a Bayes net with fully specified CPTs. (Hint: the second option is easier.)

★ SOLUTION:



Let $P(A = 1) = \epsilon$, $P(C = 1|A = 1) = 1$, $P(C = 1|A = 0) = 0$. $P(B)$ can be any arbitrary value, as for all values of B, the Bayes net will estimate $P(C = 1|B = b) = \epsilon$.

10-601 Machine Learning, Midterm Exam

Instructors: Tom Mitchell, Ziv Bar-Joseph

Monday 22nd October, 2012

There are 5 questions, for a total of 100 points.
This exam has 16 pages, make sure you have all pages before you begin.
This exam is open book, open notes, but *no computers or other electronic devices*.

Good luck!

Name: _____

Andrew ID: _____

Question	Points	Score
Short Answers	20	
Comparison of ML algorithms	20	
Regression	20	
Bayes Net	20	
Overfitting and PAC Learning	20	
Total:	100	

Question 1. **Short Answers**

True False Questions.

(a) [1 point] We can get multiple local optimum solutions if we solve a linear regression problem by minimizing the sum of squared errors using gradient descent.
True False

Solution:
False

(b) [1 point] When a decision tree is grown to full depth, it is more likely to fit the noise in the data.
True False

Solution:
True

(c) [1 point] When the hypothesis space is richer, over fitting is more likely.
True False

Solution:
True

(d) [1 point] When the feature space is larger, over fitting is more likely.
True False

Solution:
True

(e) [1 point] We can use gradient descent to learn a Gaussian Mixture Model.
True False

Solution:
True

Short Questions.

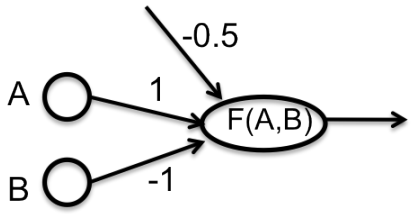
(f) [3 points] Can you represent the following boolean function with a single logistic threshold unit (i.e., a single unit from a neural network)? If yes, show the weights. If not, explain why not in 1-2 sentences.

A	B	f(A,B)
1	1	0
0	0	0
1	0	1
0	1	0

Solution:

Yes, you can represent this function with a single logistic threshold unit, since it is linearly separable. Here is one example.

$$F(A, B) = 1\{A - B - 0.5 > 0\}$$



(1)

(g) [3 points] Suppose we clustered a set of N data points using two different clustering algorithms: k-means and Gaussian mixtures. In both cases we obtained 5 clusters and in both cases the centers of the clusters are exactly the same. Can 3 points that are assigned to different clusters in the k-means solution be assigned to the same cluster in the Gaussian mixture solution? If no, explain. If so, sketch an example or explain in 1-2 sentences.

Solution:
Yes, k-means assigns each data point to a unique cluster based on its distance to the cluster center. Gaussian mixture clustering gives soft (probabilistic) assignment to each data point. Therefore, even if cluster centers are identical in both methods, if Gaussian mixture components have large variances (components are spread around their center), points on the edges between clusters may be given different assignments in the Gaussian mixture solution.

Circle the correct answer(s).

(h) [3 points] As the number of training examples goes to infinity, your model trained on that data will have:

A. Lower variance B. Higher variance C. Same variance

Solution:
Lower variance

(i) [3 points] As the number of training examples goes to infinity, your model trained on that data will have:

A. Lower bias B. Higher bias C. Same bias

Solution:
Same bias

(j) [3 points] Suppose you are given an EM algorithm that finds maximum likelihood estimates for a model with latent variables. You are asked to modify the algorithm so that it finds MAP estimates instead. Which step or steps do you need to modify:

A. Expectation B. Maximization C. No modification necessary D. Both

Solution:
Maximization

Question 2. **Comparison of ML algorithms**

Assume we have a set of data from patients who have visited UPMC hospital during the year 2011. A set of features (e.g., temperature, height) have been also extracted for each patient. Our goal is to decide whether a new visiting patient has any of diabetes, heart disease, or Alzheimer (a patient can have one or more of these diseases).

(a) [3 points] We have decided to use a neural network to solve this problem. We have two choices: either to train a *separate* neural network for each of the diseases or to train a single neural network with one output neuron for each disease, but with a shared hidden layer. Which method do you prefer? Justify your answer.

Solution:
1- Neural network with a shared hidden layer can capture dependencies between diseases. It can be shown that in some cases, when there is a dependency between the output nodes, having a shared node in the hidden layer can improve the accuracy.
2- If there is no dependency between diseases (output neurons), then we would prefer to have a separate neural network for each disease.

(b) [3 points] Some patient features are expensive to collect (e.g., brain scans) whereas others are not (e.g., temperature). Therefore, we have decided to first ask our classification algorithm to predict whether a patient has a disease, and if the classifier is 80% confident that the patient has a disease, then we will do additional examinations to collect additional patient features In this case, which classification methods do you recommend: neural networks, decision tree, or naive Bayes? Justify your answer in one or two sentences.

Solution:
We expect students to explain how each of these learning techniques can be used to output a confidence value (any of these techniques can be modified to provide a confidence value). In addition, Naive Bayes is preferable to other cases since we can still use it for classification when the value of some of the features are unknown.
We gave partial credits to those who mentioned neural network because of its non-linear decision boundary, or decision tree since it gives us an interpretable answer.

(c) Assume that we use a logistic regression learning algorithm to train a classifier for each disease. The classifier is trained to obtain MAP estimates for the logistic regression weights W. Our MAP estimator optimizes the objective

$$W \leftarrow \arg\max_W \ln[P(W) \prod_l P(Y^l|X^l, W)]$$

where l refers to the lth training example. We adopt a Gaussian prior with zero mean for the weights $W = \langle w_1 \dots w_n \rangle$, making the above objective equivalent to:

$$W \leftarrow \arg\max_W \ -C \sum_i w_i + \sum_l \ln P(Y^l|X^l, W)$$

Note C here is a constant, and we re-run our learning algorithm with different values of C. Please answer each of these true/false questions, and explain/justify your answer in no more than 2 sentences.

i. [2 points] The average log-probability of the *training data* can never increase as we increase C.
True False

Solution:
True. As we increase C, we give more weight to constraining the predictor. Thus it makes our predictor less flexible to fit to training data (over constraining the predictor, makes it unable to fit to training data).

ii. [2 points] If we start with $C = 0$, the average log-probability of *test data* will likely decrease as we increase C.
True False

Solution:
False. As we increase the value of C (starting from $C = 0$), we avoid our predictor to over fit to training data and thus we expect the accuracy of our predictor to be increased on the test data.

iii. [2 points] If we start with a very large value of C, the average log-probability of *test data* can never decrease as we increase C.
True False

Solution:
False. Similar to the previous parts, if we over constraint the predictor (by choosing very large value of C), then it wouldn't be able to fit to training data and thus makes it to perform worst on the test data.

(d) Decision boundary

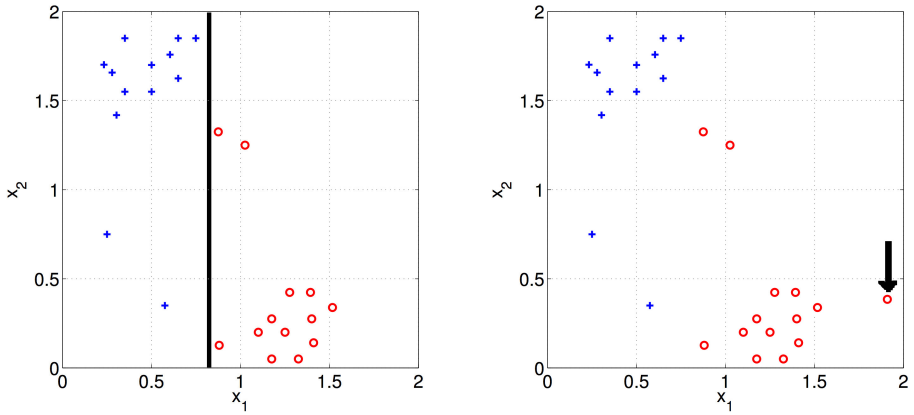


Figure 1: Labeled training set.

i. [2 points] Figure 1(a) illustrates a subset of our training data when we have only two features: X_1 and X_2. Draw the decision boundary for the logistic regression that we explained in part (c).

> **Solution:**
> The decision boundary for logistic regression is linear. One candidate solution which classifies all the data correctly is shown in Figure 1. We will accept other possible solutions since decision boundary depends on the value of C (it is possible for the trained classifier to miss-classify a few of the training data if we choose a large value of C).

ii. [3 points] Now assume that we add a new data point as it is shown in Figure 1(b). How does it change the decision boundary that you drew in Figure 1(a)? Answer this by drawing both the old and the new boundary.

> **Solution:**
> We expect the decision boundary to move a little toward the new data point.

(e) [3 points] Assume that we record information of all the patients who visit UPMC every day. However, for many of these patients we don't know if they have any of the diseases, can we still improve the accuracy of our classifier using these data? If yes, explain how, and if no, justify your answer.

> **Solution:**
> Yes, by using EM. In the class, we showed how EM can improve the accuracy of our classifier using both labeled and unlabeled data. For more details, please look at `http://www.cs.cmu.edu/~tom/10601_fall2012/slides/GrMod3_10_9_2012.pdf`, page 6.

Question 3. **Regression**

Consider real-valued variables X and Y. The Y variable is generated, conditional on X, from the following process:

$$\epsilon \sim N(0, \sigma^2)$$
$$Y = aX + \epsilon$$

where every ϵ is an independent variable, called a *noise* term, which is drawn from a Gaussian distribution with mean 0, and standard deviation σ. This is a one-feature linear regression model, where a is the only weight parameter. The conditional probability of Y has distribution $p(Y|X, a) \sim N(aX, \sigma^2)$, so it can be written as

$$p(Y|X, a) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(Y - aX)^2\right)$$

The following questions are all about this model.

MLE estimation

(a) [3 points] Assume we have a training dataset of n pairs (X_i, Y_i) for $i = 1..n$, and σ is known. Which ones of the following equations correctly represent the maximum likelihood problem for estimating a? Say yes or no to each one. More than one of them should have the answer "yes."

[Solution: *no*] $\arg\max_a \sum_i \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{1}{2\sigma^2}(Y_i - aX_i)^2)$

[Solution: *yes*] $\arg\max_a \prod_i \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{1}{2\sigma^2}(Y_i - aX_i)^2)$

[Solution: *no*] $\arg\max_a \sum_i \exp(-\frac{1}{2\sigma^2}(Y_i - aX_i)^2)$

[Solution: *yes*] $\arg\max_a \prod_i \exp(-\frac{1}{2\sigma^2}(Y_i - aX_i)^2)$

[Solution: *no*] $\arg\max_a \frac{1}{2}\sum_i (Y_i - aX_i)^2$

[Solution: *yes*] $\arg\min_a \frac{1}{2}\sum_i (Y_i - aX_i)^2$

(b) [7 points] Derive the maximum likelihood estimate of the parameter a in terms of the training example X_i's and Y_i's. We recommend you start with the simplest form of the problem you found above.

Solution:

Use $F(a) = \frac{1}{2}\sum_i (Y_i - aX_i)^2$ and minimize F. Then

$$0 = \frac{\partial}{\partial a}\left[\frac{1}{2}\sum_i (Y_i - aX_i)^2\right] \quad (2)$$

$$= \sum_i (Y_i - aX_i)(-X_i) \quad (3)$$

$$= \sum_i aX_i^2 - X_iY_i \quad (4)$$

$$a = \frac{\sum_i X_iY_i}{\sum_i X_i^2} \quad (5)$$

Partial credit: 1 point for writing a correct objective, 1 point for taking the derivative, 1 point for getting the chain rule correct, 1 point for a reasonable attempt at solving for a. 6 points for correct up to a sign error.

Many people got $\sum y_i / \sum x_i$ as the answer, by erroneously cancelling x_i on top and bottom. 4 points for this answer when it is clear this cancelling caused the problem. If they explicitly derived $\sum x_i y_i / \sum x_i^2$ along the way, 6 points. If it is completely unclear where $\sum y_i / \sum x_i$ came from, sometimes worth only 3 points (based on the partial credit rules above).

Some people wrote a gradient descent rule. We intended to ask for a closed-form maximum likelihood estimate, not an algorithm to get it. (Yes, it is true that lectures never said there exists a closed-form solution for linear regression MLE. But there is. In fact, there is a closed-form solution even for multiple features, via linear algebra.) But we gave 4 points for getting the rule correct; 3 points for correct with a sign error.

For gradient descent/ascent signs are tricky. If you are using the log-likelihood, thus maximization, you want gradient ascent, and thus add the gradient. If instead you're doing the minimization problem, and using gradient descent, need to subtract the gradient. Either way, it comes out to $a \leftarrow a + \eta \sum_i (y_i - ax_i)x_i$. Interpretation: $\sum_i (y_i - ax_i)x_i$ is the correlation of data against the residual. In the case of positive x,y, if the data still correlates with the residual, that means predictions are too low, so you want to increase a.

Here is a lovely book chapter by Tufte (1974) on one-feature linear regression:
`http://www.edwardtufte.com/tufte/dapp/chapter3.html`

MAP estimation

Let's put a prior on a. Assume $a \sim N(0, \lambda^2)$, so

$$p(a|\lambda) = \frac{1}{\sqrt{2\pi}\lambda} \exp(-\frac{1}{2\lambda^2}a^2)$$

The posterior probability of a is

$$p(a \mid Y_1, \ldots Y_n, X_1, \ldots X_n, \lambda) = \frac{p(Y_1, \ldots Y_n | X_1, \ldots X_n, a) p(a|\lambda)}{\int_{a'} p(Y_1, \ldots Y_n | X_1, \ldots X_n, a') p(a'|\lambda) da'}$$

We can ignore the denominator when doing MAP estimation.

(c) [3 points] Under the following conditions, how do the prior and conditional likelihood curves change? Do a^{MLE} and a^{MAP} become closer together, or further apart?

	$p(a\|\lambda)$ prior probability: wider, narrower, or same?	$p(Y_1 \dots Y_n \| X_1 \dots X_n, a)$ conditional likelihood: wider, narrower, or same?	$\|a^{MLE} - a^{MAP}\|$ increase or decrease?
As $\lambda \to \infty$	**[Solution:** wider]	**[Solution:** same]	**[Solution:** decrease]
As $\lambda \to 0$	**[Solution:** narrower]	**[Solution:** same]	**[Solution:** increase]
More data: as $n \to \infty$ (fixed λ)	**[Solution:** same]	**[Solution:** narrower]	**[Solution:** decrease]

(d) [7 points] Assume $\sigma = 1$, and a fixed prior parameter λ. Solve for the MAP estimate of a,

$$\arg\max_a [\ln p(Y_1..Y_n \mid X_1..X_n, a) + \ln p(a|\lambda)]$$

Your solution should be in terms of X_i's, Y_i's, and λ.

Solution:

$$\frac{\partial}{\partial a}[\log p(Y|X,a) + \log p(a|\lambda)] = \frac{\partial \ell}{\partial a} + \frac{\partial \log p(a|\lambda)}{\partial a} \tag{6}$$

To stay sane, let's look at it as maximization, not minimization. (It's easy to get signs wrong by trying to use the squared error minimization form from before.) Since $\sigma = 1$, the log-likelihood and its derivative is

$$\ell(a) = \log\left[\prod_i \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(Y_i - aX_i)^2\right)\right] \tag{7}$$

$$\ell(a) = -\log Z - \frac{1}{2}\sum_i (Y_i - aX_i)^2 \tag{8}$$

$$\frac{\partial \ell}{\partial a} = -\sum_i (Y_i - aX_i)(-X_i) \tag{9}$$

$$= \sum_i (Y_i - aX_i)X_i \tag{10}$$

$$= \sum_i X_i Y_i - aX_i^2 \tag{11}$$

Next get the partial derivative for the log-prior.

$$\frac{\partial \log p(a)}{\partial a} = \frac{\partial}{\partial a}\left[-\log(\sqrt{2\pi}\lambda) - \frac{1}{2\lambda^2}a^2\right] \tag{12}$$

$$= -\frac{a}{\lambda^2} \tag{13}$$

The full partial is the sum of that and the log-likelihood which we did before.

$$0 = \frac{\partial \ell}{\partial a} + \frac{\partial \log p(a)}{\partial a} \tag{14}$$

$$0 = \left(\sum_i X_i Y_i - a X_i^2 \right) - \frac{a}{\lambda^2} \tag{15}$$

$$a = \frac{\sum_i X_i Y_i}{(\sum_i X_i^2) + 1/\lambda^2} \tag{16}$$

Partial credit: 1 point for writing out the log posterior, and/or doing some derivative. 1 point for getting the derivative correct.

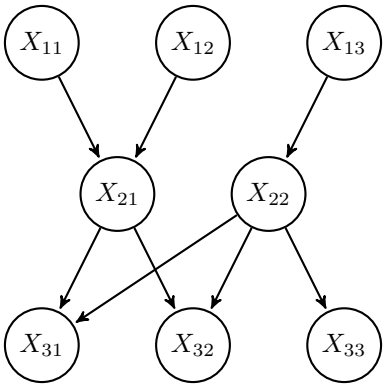
For full solution: deduct a point for a sign error. (There are many potential places for flipping signs). Deduct a point for having n/λ^2: this results from wrapping a sum around the log-prior. (Only the log-likelihood as a $\sum_i$ around it since it's the probability of drawing each data point. The parameter a is drawn only once.)

Some people didn't set $\sigma = 1$ and kept σ to the end. We simply gave credit if substituting $\sigma = 1$ gave the right answer; a few people may have derived the wrong answer but we didn't carefully check all these cases.

People who did gradient descent rules were graded similarly as before: 4 points if correct, deduct one for sign error.

Question 4. **Bayes Net**

Consider a Bayesian network B with boolean variables.



(a) [2 points] From the rule we covered in lecture, is there any variable(s) conditionally independent of X_{33} given X_{11} and X_{12}? If so, list all.

Solution:
X_{21}

(b) [2 points] From the rule we covered in lecture, is there any variable(s) conditionally independent of X_{33} given X_{22}? If so, list all.

Solution:
Everything but X_{22}, X_{33}.

(c) [3 points] Write the joint probability $P(X_{11}, X_{12}, X_{13}, X_{21}, X_{22}, X_{31}, X_{32}, X_{33})$ factored according to the Bayes net. How many parameters are necessary to define the conditional probability distributions for this Bayesian network?

Solution:
$P(X_{11}, X_{12}, X_{13}, X_{21}, X_{22}, X_{31}, X_{32}, X_{33})$
$= P(X_{11})P(X_{12})P(X_{13})P(X_{21}|X_{11}, X_{12})P(X_{22}|X_{13})P(X_{31}|X_{21}X_{22})P(X_{32}|X_{21}X_{22})P(X_{33}|X_{22})$
9 parameters are necessary.

(d) [2 points] Write an expression for $P(X_{13} = 0, X_{22} = 1, X_{33} = 0)$ in terms of the conditional probability distributions given in your answer to part **(c)**. Show your work.

Solution:
$P(X_{13} = 0)P(X_{22} = 1|X_{13} = 0)P(X_{33} = 0|X_{22} = 1)$

(e) [3 points] From your answer to **(d)**, can you say X_{13} and X_{33} are independent? Why?

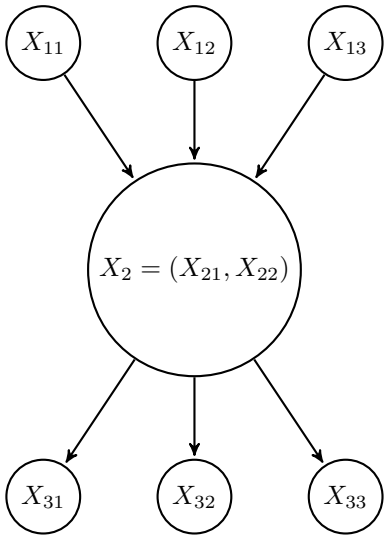
Solution:
No. Conditional independence doesn't imply marginal independence.

(f) [3 points] Can you say the same thing when $X_{22} = 1$? In other words, can you say X_{13} and X_{33} are independent given $X_{22} = 1$? Why?

Solution:
Yes. X_{22} is the only parent of X_{33} and X_{13} is a nondescendant of X_{33}, so by the rule in the lecture we can say they are independent given $X_{22} = 1$

(g) [2 points] Replace $X21$ and $X22$ by a single new variable $X2$ whose value is a pair of boolean values, defined as: $X2 = \langle X21, X22 \rangle$. Draw the new Bayes net B' after the change.

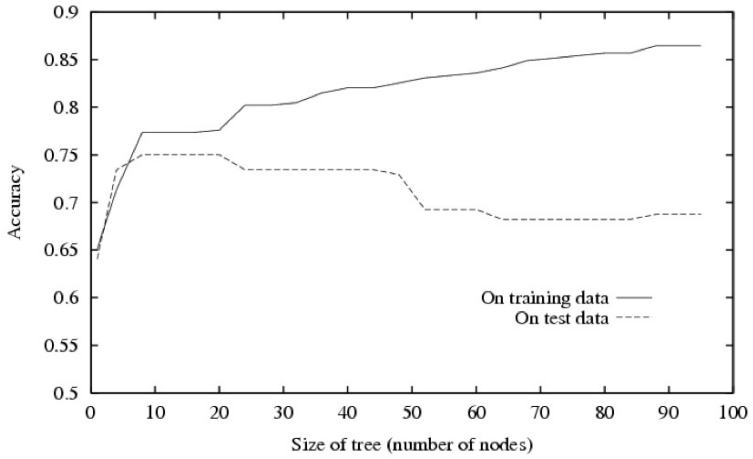
Solution:



(h) [3 points] Do all the conditional independences in B hold in the new network B'? If not, write one that is true in B but not in B'. Consider only the variables present in both B and B'.

Solution:
No. For instance, X_{32} is not conditionally independnt of X_{33} given X_{22} anymore.
* Note: We noticed the problem description was a bit ambiguous, so we also accepted yes as a correct answer

Question 5. **Overfitting and PAC Learning**



(a) Consider the training set accuracy and test set accuracy curves plotted above, during decision tree learning, as the number of nodes in the decision tree grows. This decision tree is being used to learn a function $f : X \rightarrow Y$, where training and test set examples are drawn independently at random from an underlying distribution $P(X)$, after which the trainer provides a noise-free label Y. Note error = 1 - accuracy. Please answer each of these true/false questions, and explain/justify your answer in *1 or 2 sentences.*

i. [2 points] T or F: Training error at each point on this curve provides an unbiased estimate of true error.

Solution:
False. Training error is an optimistically biased estimate of true error, because the hypothesis was chosen based on its fit to the training data.

ii. [1 point] T or F: Test error at each point on this curve provides an unbiased estimate of true error.

Solution:
True. The expected value of test error (taken over different draws of random test sets) is equal to true error.

iii. [1 point] T of F: Training accuracy minus test accuracy provides an unbiased estimate of the degree of overfitting.

Solution:
True. We defined overfitting as test error minus training error, which is equal to training accuracy minus test accuracy.

iv. [1 point] T or F: Each time we draw a different test set from $P(X)$ the test accuracy curve may vary from what we see here.

Solution:
True. Of course each random draw from $P(X)$ may vary from another draw.

v. [1 point] T or F: The variance in test accuracy will increase as we increase the number of test examples.

Solution:
False. The variance in test accuracy will *decrease* as we increase the size of the test set.

(b) Short answers.

i. [2 points] Given the above plot of training and test accuracy, which size decision tree would you choose to use to classify future examples? Give a one-sentence justification.

Solution:
The tree with 10 nodes. This has the highest test accuracy of any of the trees, and hence the highest expected true accuracy.

ii. [2 points] What is the amount of overfitting in the tree you selected?

Solution:
overfitting = training accuracy minus test accuracy = 0.77 - 0.74 = 0.03

Let us consider the above plot of training and test error from the perspective of agnostic PAC bounds. Consider the agnostic PAC bound we discussed in class:

$$m \geq \frac{1}{2\epsilon^2}(\ln |H| + \ln(1/\delta))$$

where ϵ is defined to be the difference between $error_{true}(h)$ and $error_{train}(h)$ for any hypothesis h output by the learner.

iii. [2 points] State in one carefully worded sentence what the above PAC bound guarantees about the two curves in our decision tree plot above.

Solution:
If we train on m examples drawn at random from $P(X)$, then with probability $(1 - \delta)$ the overfitting (difference between training and true accuracy) for each hypothesis in the plot will be less than or equal to ϵ. Note the the true accuracy is the expected value of the test accuracy, taken over different randomly drawn test sets.

iv. [2 points] Assume we used 200 training examples to produce the above decision tree plot. If we wish to reduce the overfitting to half of what we observe there, how many training examples would you suggest we use? Justify your answer in terms of the agnostic PAC bound, in *no more than two sentences*.

Solution:
The bound shows that m grows as $\frac{1}{2\epsilon^2}$. Therefore if we wish to halve ϵ, it will suffice to increase m by a factor of 4. We should use $200 \times 4 = 800$ training examples.

v. [2 points] Give a one sentence explanation of why you are not certain that your recommended number of training examples will reduce overfitting by exactly one half.

Solution:
There are several reasons, including the following. 1. Our PAC theory result gives a bound, not an equality, so 800 examples might decrease overfitting by more than half. 2. The "observed" overfitting is actually the test set accuracy, which is only an estimate of true accuracy, so it may vary from true accuracy and our "observed" overfitting will vary accordingly.

(c) You decide to estimate of the probability θ that a particular coin will turn up heads, by flipping it 10 times. You notice that if repeat this experiment, each time obtaining as new set of 10 coin flips, you get different resulting estimates. You repeat the experiment $N = 20$ times, obtaining estimates $\hat{\theta}^1, \hat{\theta}^2 \ldots \hat{\theta}^{20}$. You calculate the variance in these estimates as

$$var = \frac{1}{N} \sum_{i=1}^{i=N} (\hat{\theta}^i - \theta^{mean})^2$$

where θ^{mean} is the mean of your estimates $\hat{\theta}^1, \hat{\theta}^2 \ldots \hat{\theta}^{20}$.

i. [4 points] Which do you expect to produce a smaller value for var: a Maximum likelihood estimator (MLE), or a Maximum a posteriori (MAP) estimator that uses a Beta prior? Assume both estimators are given the same data. Justify your answer in one sentence.

Solution:
We should expect the MAP estimate to produce a smaller value for var, because using the Beta prior is equivalent to adding in a fixed set of "hallucinated" training examples that will *not* vary from experiment to experiment.