10-601 Machine Learning, Midterm Exam: Spring 2008
SOLUTIONS

- Please put your name on this cover sheet
- 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, open notes, no applets, no wireless communication
- **There are 80 points total on the entire exam, and you have 80 minutes to complete it. Good luck!**

Name:
Andrew ID:

Q	Topic	Max. Score	Score
1	Short answer questions	20	
2	*d*-separation	10	
3	On-line learning	10	
4	Learning methods we studied	10	
5	A new learning method	15	
6	Naive Bayes and Decision Trees	15	
	Total	80	

Part I. Short-answer questions (20 pts)

1. (4 pts) Basic probability.

   (a) You have received a shiny new coin and want to estimate the probability θ that it will come up heads if you flip it. A priori you assume that the most probable value of θ is 0.5. You then flip the coin 3 times, and it comes up heads twice. Which will be higher, your maximum likelihood estimate (MLE) of θ, or your maximum a posteriori probability (MAP) estimate of θ?. Explain intuitively why, *using only one sentence*.

   MLE- setting a prior at 0.5 will make the MAP estimate closer to 0.5, while the MLE will be $p=\frac{2}{3}$.

   (b) Show, using a proof or an example, that if $\Pr(A|B,C) = \Pr(A|C)$, then $\Pr(A,B|C) = \Pr(A|C)\Pr(B|C)$.

   $Pr(A,B|C) = Pr(A|B,C)Pr(B|C) = P(A|C)P(B|C)$

2. (6 pts) Bayes network structures.

   (a) Using the chain rule of probability (also called the "product rule") it is easy to express the joint distribution over A, B and C,$\Pr(A,B,C)$, in several ways. For example,

   i. $\Pr(A,B,C) = \Pr(A|B,C)\Pr(C|B)\Pr(B)$
   ii. $\Pr(A,B,C) = \Pr(B|A,C)\Pr(C|A)\Pr(A)$

   Write the Bayes net structures corresponding to each of these two ways of factoring $\Pr(A,B,C)$ (i.e., the Bayes nets whose Conditional Probability Tables correspond to the factors). Indicate which is which.

   $Pr(A|B,C)Pr(C|B)Pr(B)$ - *B is parent of both C and A, C is parent only of A*
   $Pr(B|A,C)Pr(C|A)Pr(A)$ - *A is parent of both C and B, C is parent only of B*

   (b) How many distinct Bayes Network graph structures can you construct to correctly describe the joint distribution $\Pr(A,B,C)$ assuming there are no conditional independencies? Please give the exact number, plus a *one sentence* explanation. (You do not need to show the networks).

   6- the only possible structure in order to have no conditional independencies is to use the chain rule structures in the previous part, but permuting the roles of A, B *and* C.

3. (4 pts) Statistical significance.

   (a) You have trained a logistic regression classifier to predict movie preferences from past movie rentals. Over the 100 training examples, it correctly classifies 90. Over a separate set of 50 test examples, it correctly classifies 43. Give an *unbiased* estimate of the true error of this classifier, and a 90% confidence interval around that estimate (you may give your confidence interval in the form of an expression).

   The thing to notice here is that using the training data results (10 incorrect out of 100) will yield a biased estimate, and so you may only use the test results (7 errors out of 50). Some people confused accuracy and error (which is asked for in the question). The answer is the following (where $Z_{90} = 1.64$):

$$\frac{7}{50} \pm Z_{90} * \sqrt{\frac{0.14 * (1 - 0.14)}{50}}$$

   (b) Show that you really know what "confidence interval" means, by describing in *one sentence* a simple experiment you could perform repeatedly, where you would expect in 90% of the experiments to observe an accuracy that falls inside your confidence interval.

   If we repeated 100 times the experiment "collect 50 randomly drawn test examples and measure the fraction misclassified by the above trained classifier," then we expect that the confidence intervals derived by the above expression would contain the true error in at least 90 of these experiments."

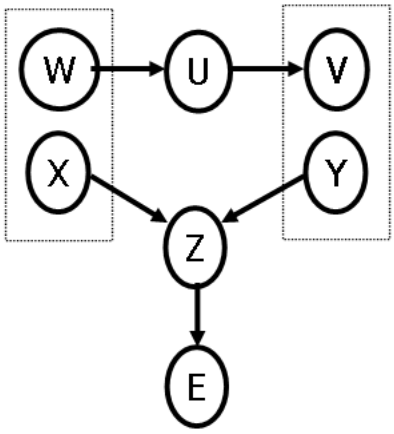
4. (3 pts) PAC learning.

   You want to learn a classifier over examples described by 49 real-valued features, using a learning method that is guaranteed to always find a hyperplane that perfectly separates the training data (whenever it is separable). You decide to use PAC learning theory to determine the number m of examples that suffice to assure your learner will with probability 0.9 output a hypothesis with accuracy at least 0.95. Just as you are about to begin, you notice that the number of features is actually 99, not 49. How many examples must you use now, compared to m? Give a *one sentence* justification for your answer.

   One thing to notice here is that the class of hypotheses are hyperplanes, and therefore we can use the formula relating the dimensionality of the data to its VC dimension, ie, $|VC|$ = dimensionality+1. Therefore, going from 49 to 99 features doubles the VC dimension from 50 to 100. Since m is linear in $|VC|$ (as opposed to log in $|H|$), this will also double m.

5. (3 pts) Bayes rule.

   Suppose that in answering a question on a true/false test, an examinee either knows the answer, with probability p, or she guesses with probability $1 - p$. Assume that the probability of answering a question correctly is 1 for an examinee who knows the answer and 0.5 for the examinee who guesses. What is the probability that an examinee knew the answer to a question, given that she has correctly answered it? *Classic application of Bayes rule. Common mistake was to assume $p = .5$, although this isn't stated in the problem. We get:* $p(knew|correct) = \frac{p(correct|knew)p(knew)}{p(correct)} = \frac{p(correct|knew)p(knew)}{p(correct|knew)p(knew)+p(correct|guessed)p(guessed)}$. *Notice in the denominator we had to consider the two ways she can get a question correct: by knowing, or by guessing. Plugging in we get:* $\frac{1*p}{1*p+.5*(1-p)} = \frac{2p}{p+1}$.



Part II. *d*-**separation (10 pts)**
Consider the Bayes network above.

1. (2 pts) Is the set $\{W, X\}$ d-separated from $\{V, Y\}$ given $\{E\}$? Why or why not?

   No. $W \rightarrow U \rightarrow V$ *is not blocked since* $U \in E$
   $X \rightarrow Z \rightarrow Y$ *is not blocked since a descendent of* Z *is in* E.

2. (4 pts) Let S be a set of variables such that $\{W, X\}$ is d-separated from $\{V, Y\}$ given S. What variables must be in S and why? What variables must **not** be in S and why?

   In S: U.
   Not in S: Z *and* E

3. (4 pts) Consider the subgraph containing only X, Y, Z and E. Is $X \perp Y|E$? If so, prove it. If not, give an example of CPTs such that $X \not\perp Y|E$.

   $X \not\perp Y|E$.
   One example: $Pr(X) = Pr(Y) = 0.5$.
   $Z \asymp X \cup Y$
   $E \asymp Z$

Part III. **On-line learning (10 pts)**
Consider an on-line learner B and a teacher A. Recall that learning proceeds as follows:

- For each round $t = 1, \ldots, T$
  - A sends B an instance x_t, withholding the associated label y_t
  - B predicts a class $\hat{y}_t$ for x_t, and is assessed an "error" if $\hat{y}_t \neq y_t$.
  - A sends B the correct label y_t for x_t.

1. (5 pts) Let H be a set of hypotheses, and assume that A is *reasonable for* H in the following sense: for every sequence $(x_1, y_t), \ldots, (x_T, y_T)$ presented by A to B, there exists some $h_* \in H$ such that $\forall t, y_t = h_*(x_t)$. Show A can still force the learner B to make at least d mistakes, where d is the VC dimension of H.

   $VCdim = d$ means that there is a set S such that $\forall S' \subset S \exists h \in H : h \cap S = S'$. Present each element of S to B; whatever B predicts, A can provide the opposite prediction and still be consistent with some h.

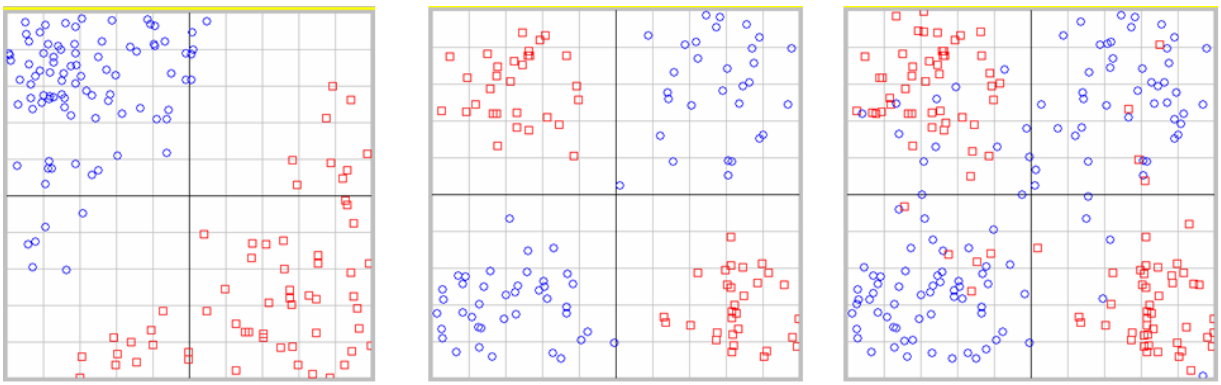
2. (5 pts) Suppose $D = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ is a dataset where

   - each $\mathbf{x}_i$ represents a document d_i, where d has no more than 100 English words (and for the purpose of this question there are no more than 100,000 English words);
   - each $\mathbf{x}_i$ is a sparse vector $(b_1, \ldots, b_{100,000})$, where $b_w = 1$ if word w is in the document d_i, and $b_w = 0$ otherwise;
   - y_1 is a label ($+1$ or -1) indicating the class of the document;
   - there exists some vector $\mathbf{u}$ such that $\forall i, y_i \mathbf{u} \cdot \mathbf{x}_i > 1$

   Write down an upper bound on R, the maximum distance of an example $\mathbf{x}_i$ from the origin, and m, the number of mistakes made by a perceptron before converging to a correct hypothesis $\mathbf{v}_k$, when trained on-line using data from D.

   $m < \frac{R}{\gamma^2}$.
   Plugging in, $R \leq ||\bar{x}||^2 \leq 100$
   $\gamma = 1$ (as $y_n \bar{u} \bar{x_n} > 1$. It's also ok to make an assumption about $||n||$)
   So $m < 100$.

Part IV. Understanding the performance of previously-studied learning methods (10 pts)



Consider the three datasets (A), (B), and (C) above, where the circles are positive examples (with two numeric features – the coordinate of the circle in 2D space) and the squares are negative examples. Now consider the following four learning methods: Naive bayes (NB), logistic regression (LR), decision trees with pruning (DT+p) , and decision trees without pruning (DT-p).

1. (5pts) For each of the three datasets, list the learning system(s) that you think will perform well, and say why.

   There was some freedom here as to how many methods you listed, and in which order, etc. The general idea, though, is that (A) is linearly separable and should be learnable by NB, LR, DT-p and DT+p. (B) is not linearly separable (and so can't be learned by NB or LR), but can be perfectly classified with two lines, ie, a decision tree like DT-p or DT+p. Finally, (C) is a noisy version of (B) and so we need DT+p's pruning to deal with the noise (DT-p would overfit the noise).

2. (5pts) For each of the three datasets, list the learning system(s) that you think will perform poorly, and say why.

   As above, all methods should do well on (A), NB and LR should fail (B) due to decision boundary form, and NB, LR and DT-p should fail (C) due to noise.

Part V. Understanding a novel learning method: Uniform Naive Bayes (15 pts)

Suppose you are building a Naive Bayes classifier for data containing two continuous attributes, X_1 and X_2, in addition to the boolean label Y. You decide to represent the conditional distributions $p(X_i|Y)$ as uniform distributions; that is, to build a *Uniform Naive Bayes Classifier*. Your conditional distributions would be represented using the parameters

$$p(X_i = x|Y = y_k) = \frac{1}{b_{ik} - a_{ik}}$$

where, $a_{ik} < b_{ik}$. (of course $P(X_i = x|Y = y_k) = 0$ when $x < a_{ik}$ or $x > b_{ik}$.)

1. (5 pts) Given a set of data, how would we choose MLEs for the parameters a_{ik} and b_{ik}?

   Take a_i, b_i to be the minimum and maximum for each class i. (See homework 2.1)

2. (5 pts) What will the decision surface look like? (If you like you can describe it in words).

   Two rectangles that may or may not intersect. If they intersect, there is a "tie".

3. (5 pts) Sometimes it is desirable to use a classifier to find examples with highly confident predictions—e.g., to rank examples $\mathbf{x}$ according to $P(Y = +|\mathbf{x}, \theta)$. If you're using a classifier for this, it is usually undesirable for the scores for two examples to be tied (e.g., for $P(Y = +|\mathbf{x}_1, \theta) = P(Y = +|\mathbf{x}_2, \theta)$.

   Are ties more or less likely for Uniform Naive Bayes, compared to Gaussian Naive Bayes, and why?

   More likely– Under UNB, in the intersection of the rectangles, as well as any area outside (where probability is 0 for both classes), there will be ties. In GNB ties only occur along the (linear) decision boundary.

Part VI. Understanding a novel learning method: Naive Bayes/Decision tree hybrids (15 pts)

Suppose you were to augment a decision tree by adding a Naive Bayes classifier at each leaf. That is, consider learning a decision tree of depth k, (where k is smaller than the number of variables n), where each leaf contains not a class, but a Naive Bayes classifier, which is based on the $n - k$ variables not appearing on the path to that leaf.

1. (5 pts) Briefly outline a learning algorithm for this representation, assuming that k is a fixed parameter. *The simplest idea was to grow a decision tree by the usual method to depth k. Then learn a NB model over the examples in each of the leaves at level k. More sophisticated versions of this method, eg. iterating one round of decision tree growth with one round of NB learning, were also acceptable.*

2. (5 pts) Will this result in having the same NB classifier at each leaf? Why or why not? If not, how will they be different?

   The NB classifiers learned at each leaf will be different because they will have been trained on different examples, based on the different paths the examples take through the tree and the corresponding conditions each example satisfies or doesn't.

3. (5 pts) Briefly describe a plausible means of selecting a value for k.

   Two common answers here: either a pruning-like method, based on information gain or something similar, or a cross validation method, ie, growing the tree for various values of k until performance on held-out data starts to fall. We didn't consider the time complexity for these various methods, but we did penalize for not explicitly using held-out data for cross-validation, eg. choosing k so as to maximize performance directly on test data was not acceptable.