10-601 Machine Learning, Fall 2011: Homework 4

Tom Mitchell and Aarti Singh

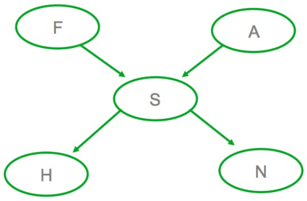
Machine Learning Department
Carnegie Mellon University

Due: October 25

Instructions There are 2 questions on this assignment. Please submit your completed homework to Sharon Cavlovich (GHC 8215) by noon, Tuesday, October 25. Submit your homework as 2 **separate** sets of pages, one for each question (so the TA's can easily split it up for grading). Include your name and email address on each set.

1 Expectation Maximization [30 points]

Consider the following graphical model, which defines a joint probability distribution over five Boolean variables. In this question, you will apply EM to train this Bayesian network, given training data in which the variables F, S, H and N are fully observed, and where the variable A is sometimes unobserved.



1. What are the conditional probability distributions associated with each of the five random variables in this network?
2. During the E step of the EM algorithm, we estimate the probability distribution over each unobserved value of each training example. Given that our only unobserved variable is A, the E step involves calculating $P(A|F, S, H, N)$. Write an expression for $P(A = 1|F, S, H, N)$ in terms of the conditional probability distributions available for this Bayesian network. *Hint: Start by writing it out based on the definition of conditional probability.*
3. What variables are in the Markov blanket for variable A?
4. Given that all variables in the Markov blanket of A are observed, it should be possible to compute the distribution over A based on only these variables. Simplify your expression from part (2) so that it uses only the variables in the Markov blanket of A.
5. During the M step, the parameters of the network are re-calculated using the observed training data plus the distributions calculated during the E step for the training values that are unobserved. Some parameters in our network can be estimated based solely on the observed variables F, S, H and N. Other parameters in our network depend on the inferred distributions over the unobserved variables calculated during the E step. List the parameters that depend on the E step. To refer to a particular parameter, simply write down the probability it represents (e.g., $P(N = 1|S = 0)$).

Example	F	S	H	N	A	$P(A=1\|F,S,H,N)$
1.	0	1	0	1	1	–
2.	0	0	1	0	1	–
3.	1	0	1	1	0	–
4.	0	0	0	1	?	0.8
5.	0	1	0	0	?	0.4

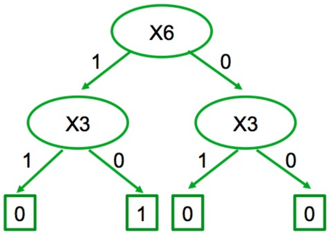
Table 1: Training examples, and E-step results for unobserved values of A

6. Consider the above data table, showing the values of observed variables in the training examples, and showing the inferred distribution over unobserved values from the E step during the 3rd iteration of the EM algorithm. What estimate will be produced during the M step for the parameter that defines $P(A=1)$? What value will be assigned to the parameter defining $P(S=1|F=0,A=1)$?

2 PAC Learning and Overfitting [30 points]

In Homework 3, you considered how the training and test error of a decision tree vary with the complexity of the tree. In this question, you will use PAC learning theory to examine the same question.

Consider a concept learning task defined over the set of instances X, where each instance $x \in X$ is described by $n = 40$ Boolean variables. Consider a decision tree learner that uses hypothesis space H_d. H_d contains only decision trees of full depth d; that is, each leaf node in each decision tree is exactly d edges away from the tree root. Furthermore, unlike general decision trees which can test different attributes along different paths from the root of the tree, trees in H_d test the same sequence of attributes along every path from the root to a leaf. Finally, any combination of labels is allowed on the leaf nodes of the tree. For example, the tree below, from H_2, tests the sequence of attributes X_6, X_3 along each path from the root to a leaf. Notice the number of functionally distinct decision trees in H_d is less than or equal to $(n \text{ choose } d)2^{2^d}$ [1]. Note also that the set of functions represented by H_{d+1} includes the entire set of functions represented by H_d.



Throughout this question, we make the usual assumptions of PAC learning: the function we are attempting to learn is deterministic, m training examples are drawn at random, i.i.d, from an unknown $P(X)$, then labelled by an error-free trainer, and the learner outputs a hypothesis h satisfying $h = \arg\min_{h \in H_d} error_{train}(h)$.

Throughout this question, let us also assume the target function we are attempting to learn, $f : X \rightarrow \{0,1\}$, is perfectly described by some decision tree in H_{10}, though the learning algorithm is unaware of this. Finally, we assume there are $m = 100,000$ training examples, and $n = 40$ Boolean variables defining each instance x.

Part A. This part of the question does not require you to use any equations. You should be able to answer the questions in this part based on informal analysis and common sense.

Which of the following statements must be true in this problem setting? Answer True or False for each, and give a *one sentence* explanation.

1. If the learner uses hypothesis space H_{10} then its learned h will satisfy $error_{train}(h) = 0$.

[1] There are (n choose d) different ways to choose which d variables to include in the tree, and there are 2^{2^d} ways to label the 2^d leaves of the tree.

2. If the learner uses H_{10} its learned h will satisfy $error_{true}(h) = 0$.

3. If one learner (we'll call this learner L$_{10}$) uses H_{10} and outputs a depth 10 decision tree h_{10}, and a second learner (which we will call L$_5$) uses H_5 to output h_5 based on exactly the same set of m training examples, then
$$error_{train}(h_{10}) \leq error_{train}(h_5)$$

4. If learner L$_{15}$ uses H_{15} and outputs a depth 15 decision tree h_{15}, and learner L$_{10}$ uses H_{10} to output h_{10} based on exactly the same set of m training examples, then
$$error_{train}(h_{15}) \leq error_{train}(h_{10})$$

Based on your analysis above, draw a plot showing *training accuracy* as a function of d (from $d = 1$ to 20), for learners using H_d, assuming that the number of training examples m is held constant. Note this plot is similar to the plot we provided to you in Homework 3, but you should draw it consistent with your answers to the above questions. For some ranges of d you won't know the exact training accuracy, so just draw an approximate curve in those intervals, and indicate on your plot which range of d has accuracies you are certain of.

Part B. Now let us see what we can say about $error_{true}(h_d)$ as a function of d. For this part, we will use the formal PAC results discussed in class.

1. Consider learner L$_{10}$ which uses H_{10} and outputs hypothesis h_{10}. Given that each draw of m training examples will be somewhat different, we cannot predict $error_{true}(h_{10})$ with certainty. However, we can use the result we proved in class to generate a probabilistic bound on $error_{true}(h_{10})$. Give an upper bound on $error_{true}(h_{10})$ that will be achieved with 0.9 probability if we train on 100,000 training examples (where the probability is taken over different draws of training sets of this size). Justify your answer. Enter this result in your plot, as the first point on a new curve showing a lower *bound on true accuracy.* [*Hint:* if you type the search query "17 choose 8" to Google, it will give you the answer.]

2. Answer the same question as above, this time for a learner that uses H_{12}, and also for a learner that uses H_{15}. Plot these two additional points on the "bound on true accuracy" curve.

3. Does your bound on $error_{true}(h_d)$ increase or decrease with d? With m?

4. Now consider the $error_{true}(h_d)$ when $d < 10$. Notice in this case, we cannot use the result we proved in class which assumes the correct target function is in the learner's hypothesis space H. However, we can use the agnostic bound on the distance between $error_{true}(h_d)$ and $error_{train}(h_d)$; that is, the degree of overfitting. Calculate a bound on the quantity $(error_{true}(h_d) - error_{train}(h_d))$ that will hold with probability 0.9, where this probability is taken over different draws of $m = 100,000$ training examples. Calculate your bound for a learner that uses H_5, and for a learner that uses H_8. Plot both points on your plot of true accuracy, relative to your estimated training accuracy curve.

5. How does your bound on overfitting grow or shrink with d? With m?

6. We can still use the agnostic bound even when $d \geq 10$. Use it to give a bound on $error_{test}(h_{10})$ for the learner L$_{10}$. Is this tighter or weaker than the bound you derived above? Plot it as well.

7. Optional: Any interesting observations you would like to volunteer?