Machine Learning 15:681

Final Exam, Fall 1995

Professor Tom Mitchell, CMU

Instructions:

1. Write your name at the top of each page.
2. Answer questions in the space provided. If you need extra space, use the back of the preceding sheet.
3. Short, precise answers will receive the most credit. Long rambling answers are worth fewer points.
4. do not, do not, do not work on the extra credit question until you are finished with all other questions.

Good Luck!

1. [20 points] *One sentence answers.* Really. One sentence only.

- Explain why Occam's razor can lead you astray.

- Give a one-sentence defense of the Minimum Description Length Principle.

- Give your best one-sentence description of the inductive bias of:
  - Candidate elimination
  - Bayes rule (i.e., Bayes optimal classifier)
  - Prolog-EBG
  - KBANN

- Given that Backpropagation works so well, why use anything else? Give two reasons in two one-sentence answers. In each, suggest an alternative algorithm and a reason to prefer it over Backpropagation.

- Consider a learning task in which instances X are points in the x, y plane, f is some real-valued target function defined over x and y, and the hypothesis space H is the set of all linear functions defined over x and y. List the conditions under which the MAP hypothesis is the hypothesis with minimal sum of squared errors over the training data.

- True of false? If there exists a set of k instances that *cannot* be shattered by H, then the VC dimension of H is less than k.

- State two of the assumptions that most limit that practical utility of PAC learning theory. (ok, one sentence each)

2. Computational learning theory. [20 points]

A. Consider a learning task defined over some finite instance space X, and a learner using a hypothesis space H. If H contains a hypothesis corresponding to each of the possible subsets of X (i.e., H is the power set of X), give an expression for the VC dimension of H.

B. Consider the *mistake bound* problem setting in which a learner receives a sequence of training examples. Upon receiving each new instance x, the learner must first predict the target value $c(x)$, and is then provided the correct target value by the trainer. We are interested in the total number of mistakes the learner will make in these predictions before it *exactly* learns the target concept. It exactly learns the target concept when its hypothesis h is identical to the true target concept c.

Consider a learner that uses the version space candidate elimination algorithm and uses a hypothesis space H that contains that target concept. It predicts the value of each new instance x by a *majority vote* among the hypotheses in the current version space. *Give a worst case bound on the total number of mistakes that will be made by this algorithm before exactly learning the target concept, independent of the particular target concept c in C and the specific sequence of training examples. Explain your reasoning.*

(big hint: think about what mistakes do to the number of hypotheses remaining in the version space)

- (Extra credit). Extend your result to the situation in which prior probabilities $P(h)$ are known for h in H, and in which the learner makes predictions now by taking the *weighted* majority vote among all hypotheses, weighting each by its posterior probability. Consider first the case where $P(D|h)$ is either 0 or 1. Then consider when $P(D|h)$ may never be zero, then the learner will never be able to eliminate any hypothesis from consideration, so the argument you (probably) used above doesn't carry over. Can you say anything interesting in this case? (note this algorithm is an incremental Bayes optimal classifier)

3. [20 points] Bayes learning.

A. [8 pts] Consider learning a boolean concept using a hypothesis space H containing 100 equally probable hypotheses. Assume noise-free data, and assume the target concept is contained in H. The first training example seen by the learner is labelled a positive example. What is the posterior probability of

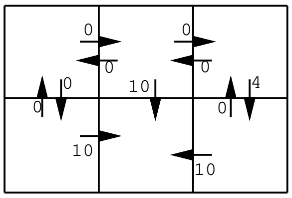
   - Each hypothesis consistent with this training example (assume there are n such hypotheses).

   - Each hypothesis that is inconsistent with this example (assume there are $100 - n$ of these).

B. [8 pts] Now consider the case of noisy training data. In particular, assume that the probability that any given training example will be incorrectly labelled (e.g., positive when it should be negative) is 0.1. Give answers to the same two questions under this new set of assumptions.

4. [20 points] Reinforcement learning.

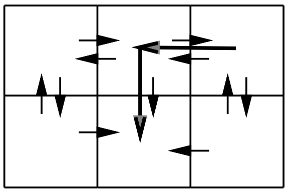
Consider Q learning for the state space and reward function shown below. Here the arrows indicate possible actions, and the numbers indicate the immediate reward $r(s, a)$ obtained by the agent for executing the corresponding action. (Note the bottom center state is an "absorbing" state in which no further actions or rewards are available to the agent).



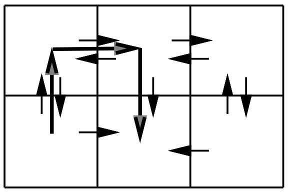
reward function $r(s, a)$.

Consider applying the Q learning algorithm (see Table 1 in the Reinforcement Learning chapter) to the sequence of training episodes shown below. Assume $\gamma = 0.8$.

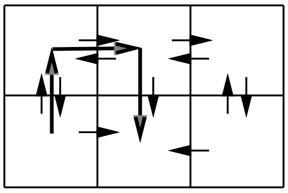
A. Show the learned $\hat{Q}$ function after each of the following training episodes is presented in sequence to the learner. (i.e., write in any values of $\hat{Q}$ that are updated based on the episode).



Episode 1

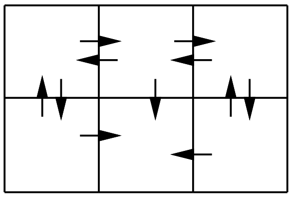


Episode 2

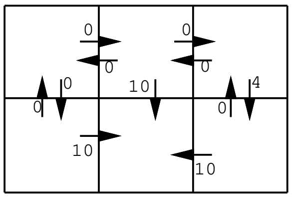


Episode 3

B. Write in the correct Q values for each state-action transition (i.e., the values to which Q learning will eventually converge.) Also draw in arrows indicating the optimal policy.



C. Suggest a change to the reward function $r(s, a)$ that will alter the $Q(s, a)$ values, but will not alter the optimal policy.



reward function $r(s, a)$.

D. Suggest a change to γ that will alter the optimal policy, and summarize what will change.

E. Consider the strategy used by the agent to select actions to perform during training. Discuss one advantage and one disadvantage of each of the following strategies:

Max: In state s the agent always executes the action a that *maximizes* $\hat{Q}(s, a)$.

LRU: In state s the agent always executes the action a it has tried least often (in the past) from this state.

Min: In state s the agent always executes the action a that *minimizes* $\hat{Q}(s, a)$.

5. [30 points] Combining analytical and inductive learning. Consider the following domain theory for the target concept $Pass\_Final$.

$$pass\_final(x) \leftarrow know\_material(x), wide\_awake(x)$$
$$know\_material(x) \leftarrow studied(x)$$
$$wide\_awake(x) \leftarrow slept\_last\_night(x)$$

Assume instances are described by the seven boolean attributes $slept\_last\_night$, $male$, $studied$, $ate\_breakfast$, $passed\_midterm$, $attended\_lectures$, and $read\_handouts$.

A. [6 pts.] Show the result of applying Prolog-EBG to the following positive training example:

$slept\_last\_night(Sue) \wedge \neg male(Sue) \wedge studied(Sue) \wedge \neg ate\_breakfast(Sue) \wedge \neg passed\_midterm(Sue) \wedge attended\_lectures(Sue) \wedge read\_handouts(Sue)$

In particular, show the explanation and the final horn clause.

If Prolog-EBG is given an incorrect domain theory, it may output an incorrect rule. For example, because the above domain theory happens to omit the important connection between $attended\_lectures$ and $know\_material$, the learned rule is overly general. *Assume throughout the remainder of this question that we know in advance the only domain theory errors present are such omitted preconditions. In this case the only possible error in any learned rule is that it might be overly general.*

The remainder of this question asks you to consider algorithms that accommodate such overly general domain theories by first learning, then repairing incorrect rules.

B. [7 pts.] Assume noise-free training examples are presented *incrementally* to the learner. Suggest an algorithm that uses version spaces to refine (when needed) each rule generated by Prolog-EBG. Remember that several distinct rules might be required to cover all positive examples of the target concept. Give a pseudo-code description of your algorithm in the space below.

C. [7 pts.] Now assume training examples are presented in a "batch," all available at once rather than incrementally presented. Suggest an algorithm that uses the mutual information measure as a basis for refining incorrect rules in this case. Again, give a pseudo-code description.

D. [10 pts.] Finally, consider how the KBANN algorithm would use the above domain theory. We can view KBANN's process of refining network weights as refining the initial overly general network derived directly from the domain theory. Compare your two approaches and KBANN by answering the following questions

- Assume noise-free training examples. For each of the three algorithms, discuss whether it is guaranteed to find optimal (or at least better) refinements to the initial Prolog-EBG rule. Either state conditions under which it will, or discuss the difficulties that make such claims impossible.

- Now assume training examples may contain noise (errors). For each of the three algorithms, indicate the impact of noisy training data on its usefulness.

- Consider removing the assumption that the domain theory is overly general (i.e., assume it may contain arbitrary errors). For each of the three algorithms, sketch *briefly* any problems this might cause, and how you would try to overcome them.