Machine Learning 15-681
Avrim Blum and Tom Mitchell, Fall 1996

Homework 7
Due December 5, 1996

NOTE: this homework is worth double credit. It is a very good way to study for the final! (Chapters 8-13 of the textbook are available from Jean in Wean 5313 if you have not yet picked up your copy).

1. *Bayesian interpretation of learning algorithms.* Consider the concept learning algorithm $FindG$, which outputs a maximally general consistent hypothesis (e.g., some maximally general member of the version space).
   (a) Specify distributions $P(h)$ and $P(D|h)$ under which $FindG$ is guaranteed to output a MAP hypothesis.
   (b) Specify distributions $P(h)$ and $P(D|h)$ under which $FindG$ is not guaranteed to output a MAP hypothesis.
   (c) Specify distributions $P(h)$ and $P(D|h)$ under which $FindG$ is guaranteed to output a ML hypothesis but not a MAP hypothesis.

2. *Sequential covering algorithms.* Consider the options for implementing the LEARN-ONE-RULE algorithm described as part of the sequential covering algorithm in Chapter 10. In particular, consider the following attributes of the search
   - generate-and-test, or data-driven
   - general-to-specific, or specific-to-general
   - sequential cover, or simultaneous cover

   Discuss the benefits of the choice made by the algorithm in Tables 10.1 and 10.2 in the textbook. For each of these three attributes of the search strategy, discuss the (positive and negative) impact of choosing the alternative option.

3. *Explanation based learning.* Consider the domain theory in the example of Table 11.1 in the textbook. One assertion in the domain theory of this example is:

   $Weight(x, w) \leftarrow Volume(x, v), Density(x, d), Equal(w, times(v, d))$

   Consider the following weaker assertions that might be available in place of the above assertion. For each of these alternatives, (a) describe the most general expression that follows from the training example of Table 11.1 along with the domain theory modified to include this assertion, and (b) if you can, suggest a way to extend PROLOG-EBG to accommodate assertions of this weaker form.

   (a) *Weight* increases monotonically with *Volume* and *Density*.

   (b) *Weight* depends solely on *Volume* and *Density*, though the nature of this dependency is unknown.

   (c) *Weight* is statistically correlated with *Volume* and *Density*.

4. *KBANN*

   A. Consider learning the target concept *GoodCreditRisk* defined over instances described by the four attributes *HasStudentLoan, HasSavingsAccount, IsStudent, OwnsCar*. Give the initial network created by KBANN for the following domain theory, including all network connections and weights.

$$GoodCreditRisk \leftarrow Employed, LowDebt$$
$$Employed \leftarrow \neg IsStudent$$
$$LowDebt \leftarrow \neg HasStudentLoan, HasSavingsAccount$$

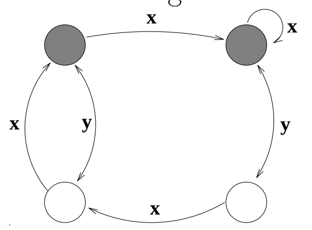
   B. KBANN converts a set of propositional Horn clauses into an initial neural network. Consider the class of *n–of–m clauses*, which are Horn clauses containing m literals in the preconditions (antecedents), and an associated parameter n where $n \leq m$. The preconditions of an n–of–m Horn clause are considered to be satisfied if at least n of its m preconditions are satisfied. For example, the clause

$$Student \leftarrow LivesInDorm, Young, Studies; \quad n = 2$$

asserts that one is a *Student* if at least two of these three preconditions are satisfied.

Give an algorithm similar to that used by KBANN, that accepts a set of propositional n–of-m clauses, and constructs a neural network consistent with the domain theory.

5. *Homing sequences.* Consider the following finite state environment:



A. A *Homing Sequence* is a sequence of actions such that the sequence of observations you see uniquely determines your *ending* state. In other words, if there are two different starting states where running the homing sequence produces the same sequence of observations, then it must be that they both lead to the same ending state. All (reduced) finite state environments have homing sequences.

   Write down a Homing Sequence for the above environment.

B. A *Distinguishing Sequence* is a sequence of actions such that the sequence of observations you see uniquely determines your *starting* state. In other words, there are no two starting states that result in the same sequence of observations. Not all finite state environments have distinguishing sequences.

   Does there exist a distinguishing sequence for the above environment? If so, give one. If not, explain why not.

C. An *Adaptive Distinguishing Sequence* is a rule where the next action you do may depend on what you've seen so far. For instance: First do the X action. If you see "gray", then do the Y action. On the other hand, if you see "white", then do the sequence XXY. In other words, you can think of this as a decision tree where nodes are labeled with observations and edges are labeled with action sequences.

   Can you find an Adaptive Distinguishing Sequence for the above environment?