Homework #7: Chapters 16, 18

The following exercises are due at the beginning of class on Thursday, April 24. Note, this homework is continued on the reverse side of the paper.

1. *[25 pts.]* Consider a student who has the choice to buy or not buy a textbook for a course. Assume that we have three Boolean random variables: *B* indicating whether the student chooses to buy the book, *M* indicating whether the student has mastered the material in the book, and *P* indicating whether the student passes the course. Since the course has an open-book final, *P* is **not** independent of *B* given *M*. Assume that the following conditional probabilities hold:

   $P(p|b,m) = 0.9$ $\quad$ $P(m|b)=0.9$
   $P(p|b,\neg m) = 0.5$ $\quad$ $P(m|\neg b)=0.7$
   $P(p|\neg b,m) = 0.8$
   $P(p|\neg b,\neg m) = 0.3$

   a) Draw a Bayesian network for the problem.
   b) The student's utility is a function of both buying the book and of passing. A common way to represent utilities that depend on multiple attributes in cases such as this is to use an **additive utility function,** e.g. $U(s)=U_{Buy}(s) + U_{Pass}(s)$. For this student, $U_{Buy}(s) = -100$ he buys the book and 0 if not, and $U_{Pass}(s) = 2000$ if he passes the course and 0 if he does not. Compute the expected utility of buying the book and of not buying it.
   c) What is the rational thing for the student do?

2. *[25 pts.]* Consider the training set in Table 1 below.

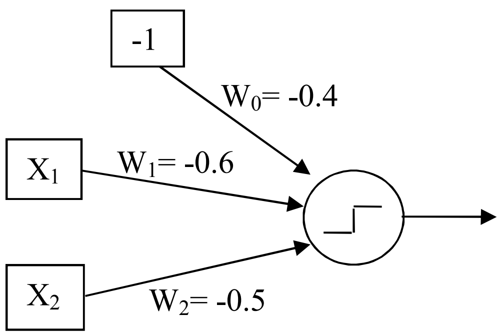
Example	Type	Garage	Bedrooms	Bathrooms	Goal Predicate
X_1	TownHouse	Yes	2	1.5	Yes
X_2	Condo	No	2	2	No
X_3	Apartment	Yes	2	1	No
X_4	TownHouse	Yes	4	2	No
X_5	TownHouse	Yes	3	1.5	Yes
X_6	Condo	Yes	1	1	No
X_7	TownHouse	No	3	1.5	No
X_8	Apartment	Yes	3	1	No

Table 1. Housing Training Set

   a) *[4 pts.]* Calculate the information requirement for the training set.
   b) *[16 pts.]* Calculate the information gain for each of the four attributes: *type*, *garage*, *bedrooms*, and *bathrooms*. **Hint:** If you don't have a calculator capable of doing base 2 logarithms, you can calculate them using the natural logarithm: $\log_2 x = \ln x / \ln 2$.
   c) *[5 pts.]* Based on your findings in part b), draw a partial decision tree that includes the attribute on which the first test should be performed and its immediate child nodes. You may use a question mark for the attribute of any nodes that cannot be completely classified by the first attribute test.

3. *[10 pts.]* Suppose you are running a learning experiment on a new algorithm for Boolean classification. You have a data set consisting of 100 positive examples and 100 negative examples. You plan to use leave-one-out cross-validation and compare your algorithm to a baseline function, a simple majority classifier. Note, a majority classifier is given a set of training data and then always outputs the class that is in the majority in the training set, regardless of the input. You expect the majority classifier to score about 50% on leave-one-out-cross-validation, but to your surprise, it scores zero every time. Can you explain why?

4. *[25 pts.]* Use the perceptron learning algorithm to teach the perceptron below to recognize the boolean NAND function. Assume that a **threshold activation function** is being used (i.e., the function returns 1 when its input is ≥ 0 and returns 0 otherwise). For initial weights, use $W_0 = -0.4$, $W_1 = -0.6$, and $W_2 = -0.5$. For the learning rate, use $\alpha = 0.1$. Use only the examples in the table to the right of the network in your learning process (note, Y is the correct output for the example). Stop the training once the weights remain unchanged for one full pass through the examples. The examples must be used in the order given by the table below. Start again with the first example whenever you exhaust all of the examples but have not yet reached the stopping criteria. Show all of the intermediate calculations and values (not just the answer or the updated weights after each example).



Training Set		
X_1	X_2	Y
0	1	1
1	1	0
0	0	1
1	0	1

5. *[15 pts.]* Construct by hand (i.e., not following the steps of any particular algorithm) a feed-forward neural network that computes the exclusive or (XOR) function of two inputs. Recall that XOR is true exactly when either input is true and the other is false. Assume that 1 = true and 0 = false for all inputs and outputs. Be sure to say which activation function you are using.