CSE 5522: Homework 3
Due Friday, October 9, 11:59 pm

1. (15 points) A survey is taken of people in the class of favorite flavor of ice cream among a forced choice of chocolate, vanilla, or strawberry. 70% prefer chocolate, 20% prefer vanilla, and 10% prefer strawberry.

   a) What is the entropy of this distribution? (Show how you derived the entropy.)
   b) How far is this from the maximum amount of entropy that you could observe for a variable that has three values?
   c) Consider the following probability distributions:
   P(Flavor=<c,v,s>| Gender=female) = <.80,.12,.08>
   P(Flavor=<c,v,s>| Gender=male) = <.67,.22,.11>
   P(Flavor=<c,v,s>| StudentType=undergrad) = <.76,.16,.08>
   P(Flavor=<c,v,s>| StudentType=grad) = <.64,.24,.12>
   P(Gender=<f,m>)=<.25,.75>
   P(StudentType=<ug,gr>)=<.5,.5>

   What would an appropriate first decision be for the decision tree algorithm?

2. (60 points) My family likes to play games, but follows a weird set of rules to determine which game to play. I've created a data set of instances on deciding whether to play "Apples To Apples" or "Settlers of Catan" based on the following attributes:
   - dayOfWeek: Weekday, Saturday, Sunday
   - timeOfDay: morning, afternoon, evening
   - timeToPlay: <30, 30-60, >60
   - mood: silly, happy, tired
   - friendsVisiting: no, yes
   - kidsPlaying: no, yes
   - atHome: no, yes
   - snacks: no, yes
   - game (predicted value): SettersOfCatan, ApplesToApples

   I've given you two versions of the data set for training and testing (use whichever one is most helpful):
   - "attrdata" – a comma-separated list of values for each instance in text
   - "codedata" – the same data represented as integers (with the class as +/- 1)

   Your job is to build an ensemble classifier, following the AdaBoost algorithm, that uses decision stumps (1-question decision trees).

   a. Build a decision stump classifier that uses information gain to choose the best first question to ask in a decision tree. Report on the accuracy of that classifier on the test set. Also report on the average probability assigned to the correct class across the test set.
   b. Implement the AdaBoost algorithm (see attachment at the end of this assignment page). Note that you will need to adapt the decision stump classifiers in part (a) to accept weights for every training data point. Report on the accuracy on the test set as you increase the ensemble to 2,3,4, and 5 classifiers.
   *Note: you must write your own implementations of both decision stumps and AdaBoost*

   Extra credit (20 points): Change the implementation of AdaBoost from decision stumps to decision trees. Discuss, for this data set, the changes in accuracy. Does AdaBoost help for this data set?

3. (20 points) For the following functions, determine whether the function can be represented by a single perceptron, or if a multi-layer perceptron is needed. Assuming threshold units, provide a set of weights for either the single perceptron or the multi-layer perceptron that represents the function.
    a. The majority function: assuming 5 binary inputs $x_1 \ldots x_5$ (either 0 or 1), 3 or more inputs are 1.
    b. The odd parity function: is 1 if and only if an odd number of binary inputs are 1 (demonstrate with 5 inputs $x_1 \ldots x_5$).
    c. $x_1$ and $x_2$ are real valued; the function is only 1 when $x_2$ is more than twice $x_1$ (0 otherwise)
    d. $x_1$ and $x_2$ are real valued, and is only 1 when $x_1$ and $x_2$ are non negative but $x_1+x_2 < 10$.

4. (5 points) [Exercise 18.11 from AMIA] Suppose you are running a learning experiment on a new algorithm for Boolean classification. You have a data set consisting of 100 positive 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. (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-once-out cross-validation, but to your surprise, it scores zero every time. Can you explain why?

AdaBoost algorithm (Figure 18.34 AIMA)

**function** AdaBoost(*examples,L,K*) **returns** a weighted majority hypothesis
**inputs:** *examples*: set of *N* labeled examples $(x_1,y_1)...(x_N,y_N)$
*L*: a learning algorithm (decision stump)
*K*: the number of hypotheses in the ensemble

**local variables: w,** a vector of N example weights, initially 1/N
**h**, a vector of K hypotheses
**z**, a vector of K hypothesis weights

**for** $k$ = 1 **to** $K$ **do**
**h**[$k$]← L(*examples*,**w**)
*error* ← 0
**for** $j$ = 1 to N do
if **h**[k]($x_j$) ≠ $y_j$ then *error* ← *error* + **w**[j]
**for** j = 1 to N do
if **h**[k]($x_j$) = $y_j$ then **w**[j] ← **w**[j] • *error* / (1-*error*)
**w** ← normalize(**w**)
**z**[k] ←log ((1-*error*)/*error*)
**return** weighted-majority(**h**,**z**)

Notes:

normalize is a function (that you write) that makes the vector sum to one.

weighted-majority is a function that takes votes from the classifiers **h** in the ensemble and then weights the votes from the classifiers according to **z**

You do not need to implement this as a function that returns a combined classifier – just take the algorithm outline as a guideline. You do need to store the component classifiers and votes somehow so you can run on the test set.