CMSC422: Practice Problems

UID: ____

RIGHT NOW: Write your UID (NOT NAME!) in the top right of every page!!! **This is worth 7 points!**

Test Information

You are *only* allowed to use the page of notes that you brought, and definitely not allowed to use any communicating device (i.e., no internet browsing, chatting, emails, texting, friends, telepathy, etc.). You *must* sign below to indicate that you understand this and as confirmation that you have not and will not break this rule. Not signing will result in a **ZERO** exam score.

____

Grading (do not write in this section)

UID	______ / 7
Total	______ / 100

Justifications (optional)

You might feel like you need to make additional assumptions in order to answer the questions, or to otherwise justify the answer you have given. Below, you may write justifications for your answers to *at most three* questions of your choosing. These will only be looked at after a first pass of grading, and only if you did not get full points on a question.

Does not apply to true/false.

Justification for Question #:

Justification for Question #:

Justification for Question #:

1 Problem: Easy vs. Hard Datasets

In this question, your task is to come up with examples of binary classification datasets that have the required properties. Illustrate each dataset in the 2-D plane.

1. Draw an example of a dataset with $-\infty$ margin.

2. Draw a dataset for which it is easy to learn a perceptron classifier. Include a hyperplane that separates the data.

3. Draw a linearly separable dataset that is harder to classify than the previous dataset (i.e., the perceptron will take longer to converge). Include a hyperplane that separates the data.

4. Draw a dataset that can be correctly classified using a decision tree, but cannot be classified with a perceptron. Draw a decision boundary for the decision tree.

5. Draw a dataset that can be correctly classified using a perceptron, but cannot be classified with a decision tree. Draw a decision boundary for the perceptron.

2 Problem: Perceptron Classifiers

2.1 Fill in the blank

For each of these statements, fill in the blank with a few words (1 to 4) or numbers. Be as specific as possible. For instance, given **"A depth k decision tree queries at most** [all] **features"**, an answer of "all" is incorrect (it's insufficiently specific). Each box is worth 2%.

1. Poorly ordered data points make perceptron training [ ] .
2. Assume you are given a data set D, with margin $\gamma > 0$. The order of the training examples will [ ] the maximum of updates needed for the perceptron algorithm to converge.
3. Early stopping is a technique used to avoid [ ] when training perceptron classifiers.
4. If a dataset is linearly separable, then the (vanilla) perceptron is guaranteed to converge to a/an [ ]
5. Suppose a perceptron encounters a training example $(\boldsymbol{x}, y = +1)$. It predicts $\hat{y} = -0.4$ and updates. The next example it sees is the same. The new prediction is $\geq$ [ ] (write the largest number guaranteed).

2.2 Practical Issues

Alice and Bob collected 1200 training examples for a binary classification task: given information about which courses a student has taken previously and their score on the midterm and final exams in those courses, predict whether or not they will like CMSC422. They implemented the standard perceptron classification algorithm, and obtained the curves in Figure 1 when applying it to their data, after randomizing the order of training examples.

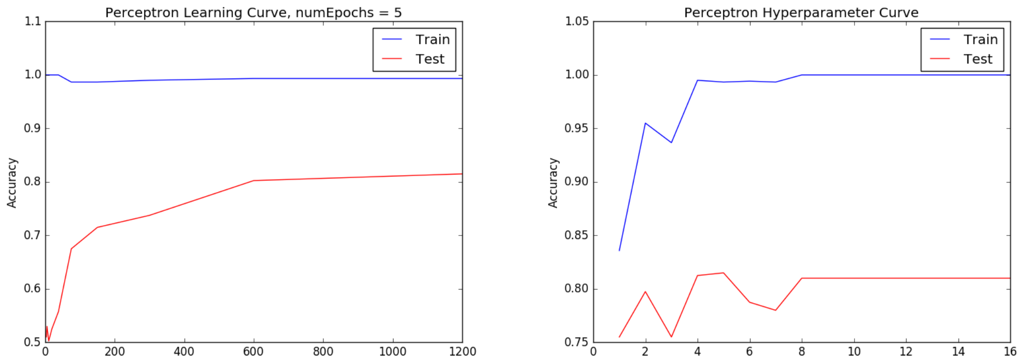


Figure 1: Curves produced by Alice and Bob using their implementation of the perceptron on the binary course recommendation task.

1. Bob suggests improving the performance of the perceptron by collecting more examples to train on. Alice thinks that would be a waste of time. Who do you agree with and why?

2. Alice observes that exams are harder in some courses than in others, and as a result grades might be distributed differently in different courses. She suggests improving the perceptron by scaling features: map all $\boldsymbol{x}$ to $\langle \frac{x_1}{m_1}, \frac{x_2}{m_2}, \ldots, \frac{x_D}{m_D} \rangle$, where m_d is the maximum absolute value of feature d in the training data, i.e., $m_d = \max \text{abs}\, x_d$ where the max ranges over training examples $\boldsymbol{x}$. Bob says that this would help improve performance with a decision tree classifier but will not benefit the perceptron. Who do you agree with and why?

3 Multiclass Linear Separability

(a) All Pairs (aka All-versus-All) is one way of mapping multiclass problems to binary problems. In All Pairs, we train $\binom{K}{2}$ classifiers, where f_{ij} distinguishes i (negative) from j (positive). Suppose we define a multiclass dataset to be "All Pairs Linearly Separable" if there exist linear binary classifiers f_{ij} (for $1 \leq i < j \leq K$) such that each All Pairs-induced binary problem is linearly separable.

Draw below a data set with three classes (call them A, B, C) and five examples per class that is All Pairs Linearly Separable. Draw the corresponding three decision boundaries and label them as f_{AB}, f_{AC} and f_{BC}. Be sure to show which direction each boundary points.

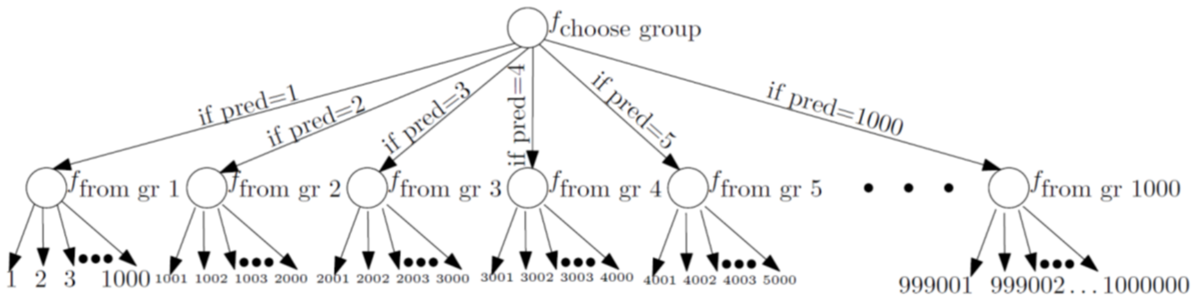
(b) One Against All is a way of mapping multiclass problems to binary problems. Define a multiclass dataset to be "OAA Linearly Separable" if there exist binary classifiers f_i (for $1 \leq i \leq K$) such that each OAA-induced binary problem is linearly separable.

Suppose that a data set (for $K = 3$) is All Pairs Linearly Separable. Is it guaranteed to be OAA Linearly Separable? If so, briefly state why (in English, one or two sentences). If not, draw a counter example.

(c) Consider four classes, with a tree-based multiclass reduction. In a tree-based multiclass reduction with four classes, the tree first splits classes c_1 and c_2 from classes c_3 and c_4. One branch then separates c_1 from c_2 and the other branch separates c_3 from c_4.

For a fixed tree τ, as before, say that a data set is τ-Linearly Separable if it's linearly separable for each of these binary problems. Consider two trees: τ_1 first separates A,B from C,D; while τ_2 first separates A,C from B,D. Draw a data set that is linearly separable for τ_1 but not for τ_2. Show the three relevant decision boundaries for τ_1.

(d) The trees we've used above are binary. We can alternatively use shallower, but bushier trees. Suppose you have $1,000,000$ classes. We can first divide these $1m$ classes into 1000 groups, where each group contains 1000 classes. This gives the shallow tree below, where leaves correspond to each of the $1m$ classes:



Here, there are 1001 multiclass classifiers. At the root, $f_{\text{choose group}}$ makes a first prediction, pred. If pred $= 3$, then the third child is traversed. Then a second 1000-way classification is made by $f_{\text{from gr 3}}$ amongst the 1000 classes sitting under node 3. Suppose each of these 1001 classifiers is implemented with OAA to reduce all the way down to binary $\{-1, +1\}$ classification.

Answer the following questions:

(1) At test time, how many binary predictions will you make to make a prediction on one test example?

(2) What happens in the case of a *single* false positive of any of the binary predictors (i.e., it predicts $+1$ when the answer should be -1) and what happens in the case of a *single* false negative of any of the binary predictors (predicts -1 when it should predict $+1$?

(3) How many binary errors can the tree tolerate and still have a reasonable chance ($> 50\%$) of making a correct prediction?