CS 165B – Machine Learning, Spring 2016

Assignment #2
Due Friday, April 22 by 4:30pm

Notes:

- *This assignment is to be done individually. You may discuss the problems at a general level with others in the class (e.g., about the concepts underlying the question, or what lecture or reading material may be relevant), but the work you turn in must be solely your own.*
- Justify every answer you give – **show the work** that achieves the answer or **explain** your response.
- Be sure to re-read the "Policy on Academic Integrity" on the course syllabus.
- Be aware of the late policy in the course syllabus – i.e., *late submissions will not be accepted*, so turn in what you have by the due time.
- Any updates or corrections will be posted on the Assignments page (of the course web site), so check there occasionally.
- Turning in the assignment:
  - There are two options for turning in problems 1-5 of your assignment:
    1. Deliver a hardcopy to the homework box in HFH 2108
    2. Submit a typeset PDF version to Gauchospace – NO scanned or photographed submissions accepted!
  - For the programming problem (#6), turn in your source code and output file (as described in the problem) using the **`turnin`** command on CSIL. (Give this a try in advance so you're not stuck trying to figure it out at the last minute! Older turn-ins are overwritten, so you can test it with anything.)

Problem #1 [8 points]
A ranking classifier ranks 20 training examples $\{x_i\}$, from highest to lowest rank, in the following order:

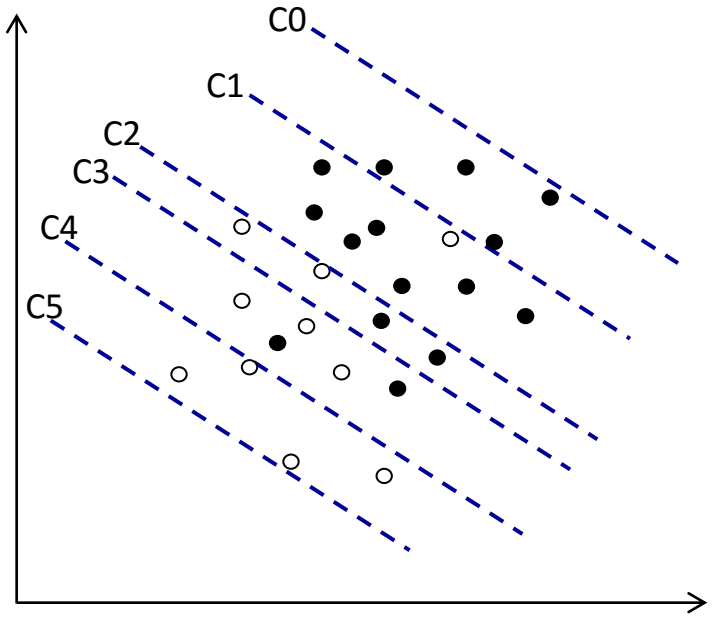
$$x_2, x_1, x_3, x_8, x_{13}, x_6, x_5, x_7, x_9, x_{12}, x_{10}, x_{11}, x_{15}, x_4, x_{14}, x_{16}, x_{17}, x_{20}, x_{18}, x_{19}$$

The true rank (ground truth) of the data is indicated by their subscript value (x_1 has a true rank of 1, x_2 has a true rank of 2, etc.). Examples x_1 through x_{12} are in the positive class; examples x_{13} through x_{20} are in the negative class.

(a) How many ranking errors are there?
(b) What is the ranking error rate?
(c) What is the ranking accuracy?
(d) Draw the coverage curve for the ranking classifier on this dataset.

Problem #2 [14 points]
The figure below shows training data with two features, with each example labeled as being in the positive (filled-in points) or negative (open points) class. Proposed linear discriminant functions (C0 through C5) are shown as dotted lines, each one indicating a different classifier for this data. Each classifier classifies points to the upper-right of its dotted line as positive and points to the lower-left of its dotted line as negative.

(a) Draw the coverage plot for this data and plot the different classifiers (and label them as C0, C1, etc.).
(b) Draw the ROC plot and label the classifiers.
(c) Which classifiers have the highest and lowest accuracy?
(d) Which classifiers have the highest and lowest average recall?
(e) Which classifiers (if any) are complete?
(f) Which classifiers (if any) are consistent?



Problem #3 [16 points]
The linear discriminant function for a binary classification problem with three features is the plane defined by the equation $2x_1 + x_2 + 3x_3 = 12$, where the features are x_1, x_2, and x_3. The discriminant plane separates the positive hypotheses ($2x_1 + x_2 + 3x_3 \geq 12$) from the negative hypotheses ($2x_1 + x_2 + 3x_3 < 12$).

The training set consists of the following labeled examples:

- (2, 2, 3) + (positive class)
- (3, 3, 2) +
- (1, 2, 3) +
- (1, 4, 1) +
- (4, 4, 4) +
- (2, 2, 2) +
- (1, 1, 1) – (negative class)
- (0, 4, 2) –
- (4, 0, 0) –
- (3, 3, 1) –
- (3, 3, 3) –

For each data point, give (a) its margin, (b) its 0-1 loss, (c) its hinge loss, and (d) its squared loss (clamped to zero above 1).

Problem #4 [10 points]
In a 5-class classification problem, 25 training examples are supplied that have the following class labels:

3	1	2	2	3
5	5	3	3	5
5	3	5	3	5
2	5	5	5	2
3	5	1	3	5

From the training data, we wish to estimate the probabilities of each class. Empirically estimate each class probability using (a) relative frequency, (b) Laplace correction, (c) m-estimate with m=5 and even distribution of the pseudocounts, and (d) m-estimate with m=20 and even distribution of the pseudocounts.

Plot these empirical probabilities, with class number on the x-axis and estimated probability on the y-axis – i.e., four plots (one for each approach), each of which consists of five connected points (the estimated class probabilities). Plot all four distributions on separate graphs.

Describe the trend as we go from (a) to (b) to (c) to (d) – that is, what does increasing the number of pseudocounts do (in general) to the probability distribution?

Problem #5 [10 points]
We'd like to learn a Boolean function that separates the people in the Hatfield family from people in the McCoy family. We know the following information about a given person:

Age status: { *Child, Young Adult, Adult, Elderly* }
Residency: { *West Virginia, Kentucky* }
Sympathizes with: { *Union, Confederate, Neither* }
Occupation: { *Miner, Bootlegger, Other, None* }

(a) For this simple problem, if testing a classification hypothesis takes a nanosecond, how long would it take to test every possible hypothesis?
(b) How long would it take if we used a conjunctive hypothesis space representation?
(c) How long would it take if we used a conjunctive hypothesis space with internal disjunctions?

Problem #6 [40 points]
Write a program called "triclassify" that classifies instances, using linear discriminating functions, among three classes, using the following method:

Training (using the training data set):
1. Compute the centroid of each class (A, B, and C).
2. Construct a discriminant function between each pair of classes (A/B, B/C, and A/C), halfway between the two centroids and orthogonal to the line connecting the two centroids.

Testing (using the testing data set):
1. For each instance, use the discriminant function to decide "A or B" and then (depending on that answer) to decide "A or C" or "B or C." (Ties should give priority to class A, then B, then C.)
2. Keep track of true positives, true negatives, false positives, and false negatives.

The training and testing data sets are available on the Assignments page of the course web site, along with a description of their formats. The syntax of your program should be:

```
% triclassify <training_data_file> <testing_data_file>
```

As output, the program should print out the averages of the true positive rate, the false positive rate, the error rate, the accuracy, and the precision – e.g., as shown here:

```
% triclassify training_file testing_file
True positive rate = 0.80
False positive rate = 0.27
Error rate = 0.44
Accuracy = 0.60
```

```
Precision = 0.90
```

Information on computing these averages is included on the Assignments page.

Run the **`triclassify`** program on the training and testing sets provided on the Assignments page, and include a single text file called **`outputs.txt`** that shows the program being run on each.

You may use C/C++, Java, Python, or Matlab for the assignment. Using the "turnin" command, submit the **`outputs.txt`** file and a subdirectory called **`src`**. In that subdirectory, include a file called **`readme.txt`** that describes specifically how to prepare (compile, load, etc.) and run the program. Your solution must run on the CSIL machines – double check that this is true before submitting.

The "turnin" submission should look like this:

```
% turnin hw2@cs165b outputs.txt src
```