v. 2

# CS 165B – Machine Learning, Spring 2016

## Assignment #3
## Due Monday, May 9 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 1-3 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 (#4), turn in your source code and output file (as described in the problem) using the **`turnin`** command on CSIL.

---

Problems 1-3 use the following training data to learn the concept *WinTitle*, which decides whether or not an NBA team will win the championship based on several attributes:

Talent level = { High, Medium, Low }
Average height = { NBA_tall, NBA_medium, NBA_short }
Great coach = { Yes, No }
Team chemistry = { Okay, Great }

The training data for *WinTitle* is:

| *#* | *Talent* | *AveHeight* | *GreatCoach* | *TeamChemistry* | *WinTitle?* |
|---|---|---|---|---|---|
| 1 | Low | NBA_medium | Yes | Okay | No |
| 2 | Low | NBA_tall | No | Okay | No |
| 3 | High | NBA_medium | No | Okay | Yes |
| 4 | Low | NBA_medium | No | Great | Yes |
| 5 | Medium | NBA_medium | Yes | Great | Yes |
| 6 | Medium | NBA_short | No | Okay | Yes |
| 7 | High | NBA_medium | Yes | Great | Yes |
| 8 | Low | NBA_short | Yes | Okay | No |
| 9 | Low | NBA_short | Yes | Great | No |
| 10 | Medium | NBA_short | Yes | Okay | Yes |
| 11 | High | NBA_medium | Yes | Okay | Yes |
| 12 | High | NBA_tall | No | Okay | No |
| 13 | High | NBA_tall | No | Great | Yes |
| 14 | Medium | NBA_tall | No | Great | No |
| 15 | Medium | NBA_short | No | Okay | No |
| 16 | Low | NBA_tall | Yes | Great | Yes |

**Problem #1 [12 points]**
In a conjunctive hypothesis space learning approach applied to the *WinTitle* scenario:

(a) How many possible (conjunctive) hypotheses are there?

(b) What is the least general hypothesis for *WinTitle* after observing training data points 13 and 16 (only)?

(c) What is the most general hypothesis for *WinTitle* after observing training data points 13, 16, and 7 (only)?

(d) What is the least general hypothesis for *WinTitle* after observing training data points 10, 11, and 13 (only)?

**Problem #2 [20 points]**
Based on the GrowTree and BestSplit-Class algorithms and using the entropy impurity function, create a decision tree to learn the *WinTitle* concept. Show how each node is decided (based on comparing impurity measures), then draw the full decision tree.

Now apply this learned concept (decision tree) to the test data files posted on the Assignments page, and list the incorrectly classified examples (by number), if any, for each.

What is the error rate of the decision tree on the training data? On each test data set?

**Problem #3 [15 points]**
From your decision tree in problem 2, create a ranking tree based on empirical probabilities of the leaves, using Laplace correction. Draw the tree and label each leaf with its estimated *WinTitle* probability and its ranking.

Compute the error rate and the accuracy of applying the ranking tree to the training data and to the test data files posted on the Assignments page.

**Problem #4 [30 points]**
Write a program called **`linreg`** that creates a multivariate linear least-squares regressor from training data (a set of P N-dimensional points) and then runs it on test data (a set of Q N-dimensional points). (P, Q, and N are positive integers.) The program learns regression parameters $\boldsymbol{w} = (w_1, w_2, \dots, w_N, t)$ for a training set of data points, leading to the regression estimate $\hat{y} = \boldsymbol{w}^T\boldsymbol{x}$ (using homogeneous coordinates). The syntax of your program should be:

```
% linreg <training_data_file> <testing_data_file>
```

As output, the program should print out the regression parameters and the regression estimate (a scalar) for each data point in the testing set, as shown here for a 2D regression problem with 4 test points:

```
% linreg training1.txt testing1.txt
w: 2.3 0.6 -2.1
1.0 2.0 -- 1.4
0.0 0.0 -- -2.1
-1.0 -1.0 -- -5.0
2.0 0.0 -- 2.5
```

Run the **`linreg`** 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 CSIL “**`turnin`**” submission should look like this:

```
% turnin hw3@cs165b outputs.txt src
```