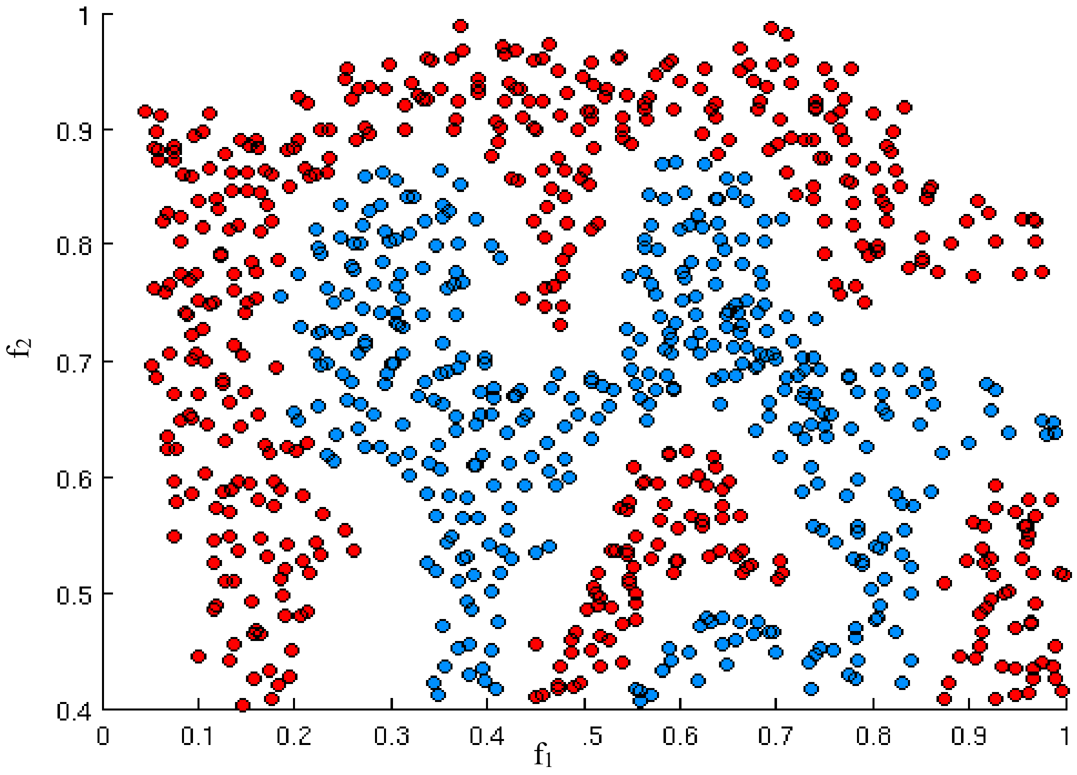
Assignment 4 – "I can't open the pod bay doors Dave…"

Part I: Questions (20 points each)

1. Given the two feature data set below where two classes are distinguished by color,

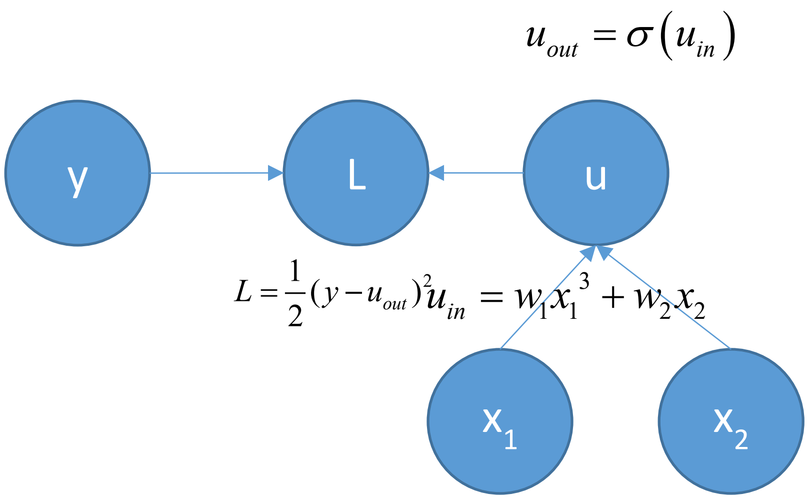


   would a single neuron neural net be more likely to have high bias or high variance? Justify your answer.

2. Briefly describe a task in your own life where unsupervised learning could be applicable.

3. Assume a 0-1 loss rule and a test set of examples: $e_1, e_2, \ldots, e_N$ with known labels $l_1, l_2, \ldots, l_N$. Assume that we have a learning function h^* such that $h^*(e_i) = \hat{l}_i$ where $\hat{l}_i$ is the predicted label. Write pseudocode for function estimate_loss(examples, labels, h*) that determines the empirical loss (also known as the empirical risk) of a test set of N examples.

4. Suppose we have the following simple network:



$u_{out} = \sigma(u_{in})$
y
L
u
$L = \frac{1}{2}(y - u_{out})^2$
$u_{in} = w_1 x_1^3 + w_2 x_2$
x_1
x_2

where $\sigma(\cdot)$ is the sigmoid function: $\sigma(z) = \frac{1}{1+e^{-z}}$.

a. Compute the following partial derivatives: $\frac{\partial L}{\partial u_{out}}, \frac{\partial u_{out}}{\partial u_{in}}$, and $\frac{\partial u_{in}}{\partial u_{w_2}}$.

b. Estimate the gradient of the loss with respect to node u's weight w_2: $\frac{\partial L}{\partial w_2}$.

Hints: 1) Use the chain rule and 2) while you could derive $\frac{\partial \sigma(z)}{\partial z}$, it is:

$$\frac{\partial \sigma(z)}{\partial z} = \sigma(z)\big(1 - \sigma(z)\big)$$

Part II: Decision tree classifiers – 100 points

You are to construct a decision tree classifier. The following Python modules will require you to write code:

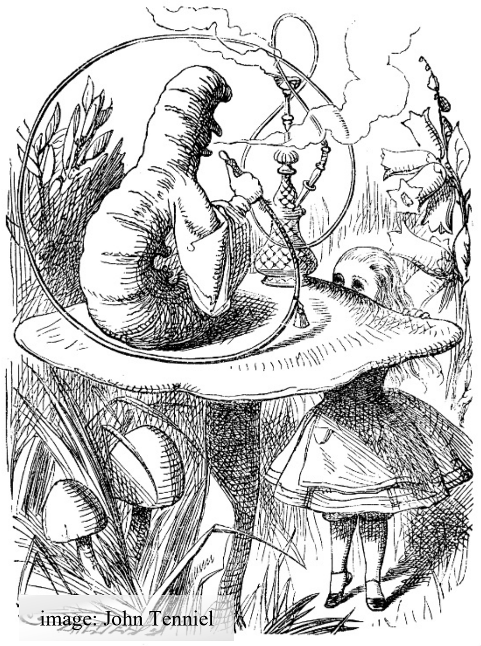


image: John Tenniel

driver.py – Entry point into your program. You will be classifying the mushroom data set and the zoo data set, both of which are provided to you. Use the provided cross validation class to conduct two 10-fold cross-validation decision tree experiments. One should be conducted without pruning and the other with pruned trees at a p-value of 0.05. Two miniscule datasets that correspond to things we did in class are available as well: restaurant, and tiny_animal_set. You do not need to test these in your program, but you may find them very useful for debugging as you know the correct behavior of many of the operations you need to implement as we discussed these in class.

The driver should print the mean error and standard deviation of the zoo and mushroom datasets using both unpruned and pruned decision trees. In addition, you should print out one unpruned decision tree and one pruned decision tree for each class. Call method chi_annotate on the tree before you print it so that you can see the χ^2 statistic for each decision node.

decision_tree.py – Contains skeleton code for a DecisionTree class. The constructor expects a dataset to be learned. Some of the methods are written for you. Do not change the names of any of the predefined variables or interfaces, they will be examined in the unit test code and can cause your execution to fail. The class contains methods to predict the class of examples, estimation entropy, estimate χ^2statistics, etc. Read through the code to determine what is provided and what you need to write. The bulk of your assignment is modifying this module.

Several modules are provided for you:

ml_lib/decision_tree_support.py – Classes to handle branches in decision trees (the root of your decision tree should be an instance of DecisionFork) as well as leaf nodes of decision trees. These classes require several things at instantiation. For DecisionFork's you will need the index of the attribute on which the fork depends (what question was asked), a tuple or list with the counts of each class that were active when this node of the tree was generated, the attribute name (as opposed to a number) to aid your understanding when the tree is displayed, and the parent node. DecisionLeaf simply needs the class associated with the node (e.g. bird or mammal for our example in class), the class counts, and the parent. Read through the code to determine how to add branches. Note that when printing the tree, if you have called chi_annotate on the tree, which will set a chi2 attribute on each DecisionTree, the tree will be printed with χ^2statistics for each branch.

ml_lib/crossval.py – Contains a function for performing k-fold cross validation. Make sure that you understand this code, cross-validation is an important concept for evaluating machine learning algorithms.

ml_lib/ml_util.py – Contains a number of useful classes and functions. You will not need everything in this module. Especially useful classes/functions:

- DataSet: Class used to parse and represent the comma separated value data that are in ml_lib/aima_data. To load in one of the datasets in the ml_lib/aima-data directory, use the keyword name, e.g. DataSet(name=”zoo”).
  Important: *Pay attention to what you are supposed to predict.* By default, the constructor assumes that the target classification attribute is the last column. This is true in the zoo data set, but not in the mushrooms data set where the goal is to determine if eating the mushroom will be hazardous to your health or not.
  DataSet also allows you to exclude certain attributes. In the zoo data set, the first column is the animal name. If we branch on the animal name, we very quickly end up with one item per branch, a trivial solution that we do not want. In this case, we would tell DataSet to exclude attribute zero. One nice feature of DataSet

is that it sets attribute values to contain a list of lists. Each item is all of the data values that were seen for a specific attribute. Read the class to learn more.
- normalize: Given a sequence such as (3, 4, 3) converts to probabilities (.3, .4, .3).

Note: These are relatively simple data sets, and you will not see a tremendous benefit to pruning (pay attention to the variance). You should however keep in mind that for more complicated data, pruning is very important.

To turn in:
As usual, parts I and II are to be turned in separately along with your affidavit. Part II should have a text file "classifier.txt" with your classifier output for the mushroom and zoo data sets.