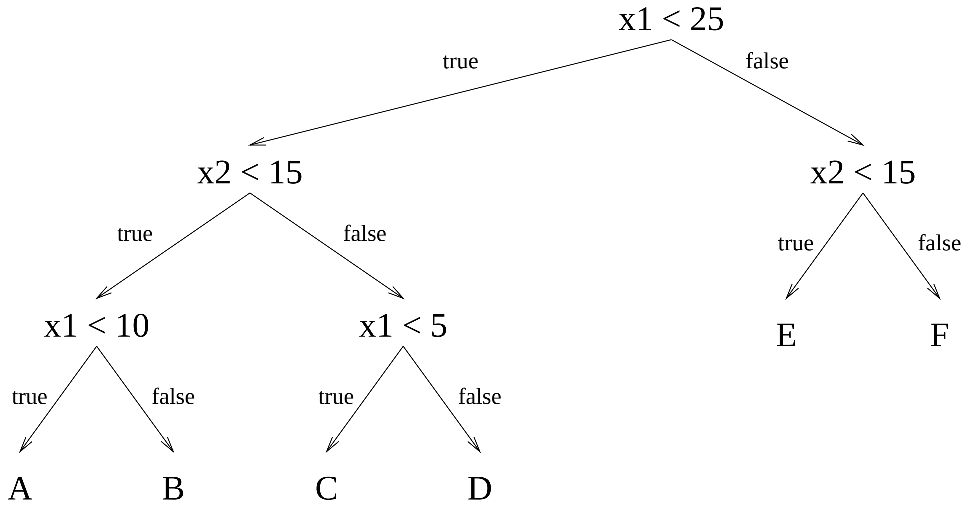
CS534 — Homework Assignment 2 — Due Friday in class, April 29th

Written assignment

1. Consider the following decision tree:



   (a) Draw the decision boundaries defined by this tree. Each leaf of the tree is labeled with a letter. Write this letter in the corresponding region of input space.

   (b) Give another decision tree that is syntactically different but defines the same decision boundaries. This demonstrates that the space of decision trees is syntactically redundant. Is this redundancy a statistical problem (i.e., does it affect the accuracy of the learned trees)? Is it a computational problem (i.e., does it increase or decrease the computational complexity of finding an accurate tree)?

2. In the basic decision tree algorithm, we choose the feature/value pair with the maximum mutual information as the test to use at each internal node of the decision tree. Suppose we modified the algorithm to choose at random from among those feature/value combinations that had non-zero mutual information, but that we kept all other parts of the algorithm unchanged.

   (a) Prove that if a splitting feature/value combination has non-zero mutual information at an internal node, then at least one training example must be sent to each of the child nodes.

   (b) What is the maximum number of leaf nodes that such a decision tree could contain if it were trained on m training examples?

   (c) What is the maximum number of leaf nodes that a decision tree could contain if it were trained on m training examples using the original maximum mutual information version of the algorithm? Is it bigger, smaller, or the same as your answer to (b)?

   (d)How do you think this change would affect the accuracy of the decision trees produced on average? Why?

3. Consider the following training set:

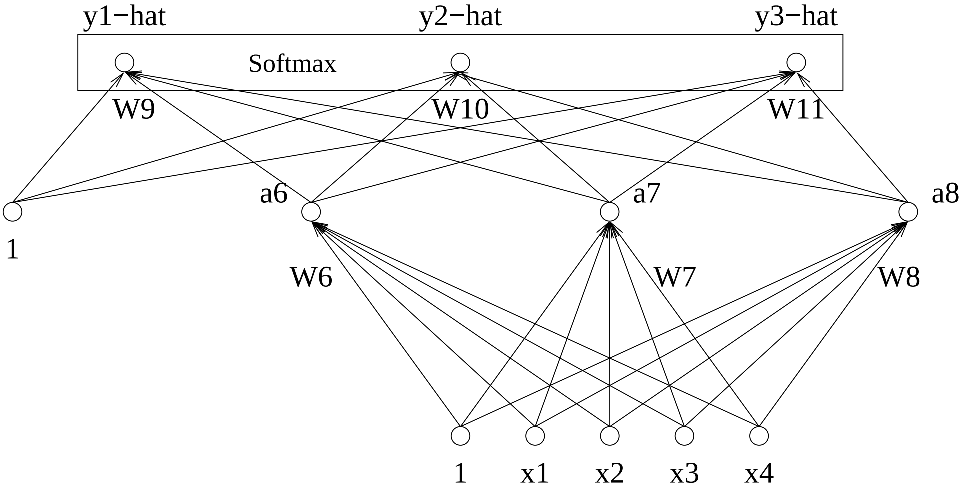
A	B	C	Y
0	1	1	0
1	1	1	0
0	0	0	0
1	1	0	1
0	1	0	1
1	0	1	1

a. Learn a Naive Beyes classifier by estimating all necessary probabilities. Make prediction for (A=1, B=0, C=0).

b. Suppose we know that A, B and C are independent random variables, can we say that the Naive Bayes assumption is valid? (Note that the particular data set is irrelevant for this question). If your answer is yes, please explain why; if you answer is no please give an counter example.

c. Learn a decision tree from the training set shown above using the Mutual Information criterion.

4. Consider a neural network diagram that uses a softmax activation function for its output layer. Its outputs can be interpreted as posterior probabilities $P(y|x)$ for a categorical target variable y. Consider the following neural network with three output units. The softmax activation function is defined as: $\hat{y}_i = \frac{\exp(x_i)}{\sum_{j=1}^{3}\exp(x_j)}$ (as opposed to what we saw in class $\hat{y}_i = \frac{1}{1+\exp(-x_i)}$), where x_i is the net input for the activation function of the output node i. Note that $\hat{y}_1 + \hat{y}_2 + \hat{y}_3 = 1$, making it a valid posterior probability. In this problem, we will compute the derivatives needed for the backpropagation algorithm for this kind of network.



a. Write down the log likelihood objective function $J(\mathbf{w})$ for this network, where $\mathbf{w}$ is the concatenation of $W6, W7, W8, W9, W10$, and $W11$. You may assume that each training example has the form $(\mathbf{x}, y)$, where $\mathbf{x} = (1, x_1, x_2, x_3, x_4)$ and $y = (y_1, y_2, y_3)$. There are only three possible y values: $y = (1,0,0)$, $y = (0,1,0)$, and $y = (0,0,1)$.

b. Compute the partial derivative

$$\frac{\partial J(\mathbf{w})}{\partial w_{9,6}}$$

c. Compute the partial derivative

$$\frac{\partial J(\mathbf{w})}{\partial w_{6,3}}$$

d. Generalize your answers to (b) and (c) and write the pseudo-code for the backpropagation algorithm using them.

5. Cubic Kernels. In class, we showed that the quadratic kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^2$ was equivalent to mapping each $\mathbf{x}$ into a higher dimensional space where

$$\Phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1)$$

for the case where $\mathbf{x} = (x_1, x_2)$. Now consider the cubic kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^3$. What is the corresponding Φ function (again, for the special case where $\mathbf{x} = (x_1, x_2)$)?

Implementation assignment

I. In this assignment you will implement the Naive Bayes classifier for document classification and apply it to the 20newsgroup data. Specifically, we have a vocabulary V, which contains all the words of interest. Consider a document that contains n words, represented as $\{X_1, X_2, \cdots, X_n\}$. The value of X_i is the word found in position i in the document. To predict the label Y (as one of m possible classes) of the document, we use the following:

$$P(Y|X_1, \cdots, X_n) \propto P(X_1, \cdots, X_n|Y)P(Y) = P(Y)\prod_{i=1}^{n} P(X_i|Y)$$

Further assuming that $P(X_i|Y)$ follows a categorical distribution over the fixed vocabulary V. For this assignment, you will estimate $P(Y)$ using MLE. For $P(X|Y)$, the conditional distribution of words given class, you will apply laplace smoothing:

$$P(X|Y) = \frac{\text{\# of times word X appeared in documents of class Y} + 1}{\text{total \# of words in documents of class Y} + |V|}$$

Note that this is also called add-one smoothing as it can be viewed as adding a pseudo-count of one appearance for each word of the vocabulary V.

One useful thing to note is that when calculating the probability, it can and will become overly small and you should operate with log of the probabilities to avoid underflow issues.

Data set information: The data set is the classic 20-newsgroup data set. There are six files.

- vocabulary.txt is a list of the words that may appear in documents. The line number is word's id in other files. That is, the first word ('archive') has wordId 1, the second ('name') has wordId 2, etc.
- newsgrouplabels.txt is a list of newsgroups from which a document may have come. Again, the line number corresponds to the label's id, which is used in the .label files. The first line ('alt.atheism') has id 1, etc.
- train.label contains one line for each training document specifying its label. The document's id (docId) is the line number.
- test.label specifies the labels for the testing documents.
- train.data describes the counts for each of the words used in each of the documents. It uses a sparse format that contains a collection of tuples "docId wordId count". The first number in the tuple species the document ID, and the second number is the word ID, and the final number species the number of times the word with id wordId in the training document with id docId. For example "5 3 10" species that the 3rd word in the vocabulary appeared 10 times in document 5.
- test.data is exactly the same as train.data, but contains the counts for the test documents.

Need to report:

1. Report the overall testing accuracy.

2. Report the confusion matrix C, where C_{ij} species the total number of times that a class i document is classified as class j.

Bonus exploration: For bonus points, design and test a heuristic to reduce the vocabulary size and improve the classification performance. This is intended to be open-ended exploration. There will be no punishment if the exploration leads to negative impact on performance.

II. Implement a fixed depth decision tree algorithm. In particular, the input to your algorithm will include the training data set and the maximum depth of the tree. For example, if the depth is set to one, you will learn a decision tree with one test node, which is also called a decision stump. Test your implementation, with depth=1, and 2 respectively, on the following data set as described below (train on the training data and test with the testing data set).

Data set information: This data set is extracted from the UCI zoo data set. Note that there are 16 features (the first 16 columns) and the class labels are in the last column. There are 7 classes (numerically specified as class 1 to 7). All features are binary except for feature 13, which is a categorical variable with possible values 0,2,4,5,6,8. Note that to create binary split, please use the one-vs-rest approach.

Need to report:

1. Report the learned decision tree (depth 1 and depth 2)
2. Report the test set accuracy and confusion matrices for both trees.