EE 5841 Machine Learning

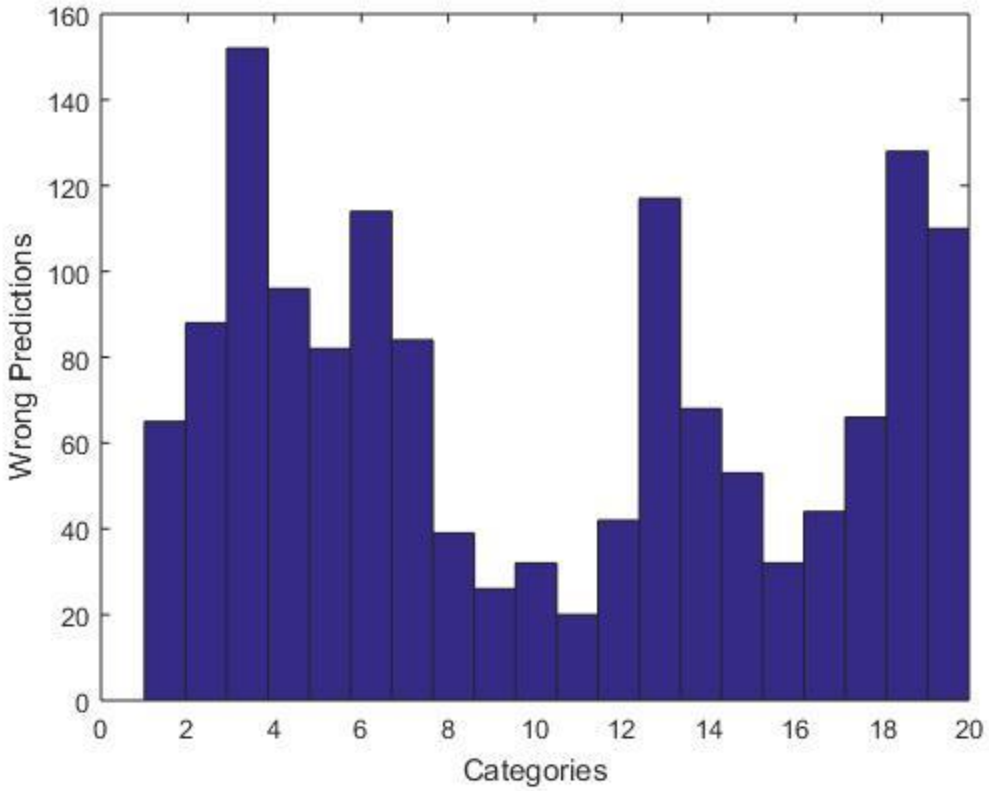
Project 2

Kirk D’Souza

a) Code

```
clear all;
% Loading training data sets

trainData = load ('train.data');
trainLabel = load ('train.label');


% Initialization of empty matrices for faster calculations
DocWordCount = zeros(11269, 53975);

% Creating Doc*word_index Count matrix for train data set
for i = 1:1467345
    DocWordCount(trainData(i,1), trainData(i,2)) = trainData(i,3);
end;


% Clearing variables that won't be used again to speed up the program


clear trainData;

% Initialization of empty matrices for faster calculations
WordIndexCountForClasses = zeros(20,53975);
TotalWordsForClasses = zeros(20,1);
PriorClass = zeros(20,1);


%% Calculating Probabilities

for i = 1:20
    for j  = 1:11269
        % Adding to empty matrices if categories match with trainLabel values


        if trainLabel(j) == i
        % Creating a sum of words for each category
        WordIndexCountForClasses(i,:) = WordIndexCountForClasses(i,:) + DocWordCount(j, :);
        % Total words in each Category
        TotalWordsForClasses(i,:) = TotalWordsForClasses(i,:) + sum(DocWordCount(j,:));

        end
    end
end
% Calculating Probability of Word given Categories
PWordCat = WordIndexCountForClasses./repmat(TotalWordsForClasses,[1,53975]);
```

```matlab
% Calculating log likelihood with a Smoothing Factor of 0.1
likelihood = log(((1 - 0.1)*PWordCat) + (0.1/53975));
% Calculating Prior
prior = log(TotalWordsForClasses./sum(TotalWordsForClasses));

clear WordIndexCountForClasses TotalWordsForClasses;

% ProbablityPriorClass = (SumEachClass/11269);
clear SumEachClass trainLabel DocWordCount trainLabel;

clear WordIndexCountForClasses TotalWordsForClasses;
% Loading Testing data
testData = load ('test.data');

docWordCountTest = zeros(7505, 61188);
for i = 1:967874
    docWordCountTest(testData(i,1), testData(i,2)) = testData(i,3);
end;

% Ignoring the new extra words
usedDocWordCountTest = docWordCountTest(:,1:53975);
% Calculating the Probability of Documents given Categories and taking log
likelihood^usedDocWordCountTest
PDocCat = likelihood * usedDocWordCountTest';
% Scaling Prior
bigPrior = repmat(prior,[1,7505]);
% Calculating the Posterior and taking log to calculate it as a summation instead of Product
posterior = bigPrior + PDocCat;

% Initialization of empty matrices for faster calculations
confidence = zeros(7505,1);
predictions = zeros(7505,1);

% Obtaining max values for posterior for every category
for i = 1:7505
    [con,pred] = max(posterior(:,i));
    predictions(i,1) = pred;
    confidence(i,1) = con*100;
end;

testLabel = load('test.label');
% Calculating accuracy of predications
acc = sum(testLabel == predictions)/length(testLabel)*100;

% Calculating the misclassified predictions and ploting them
wrong = testLabel(find(testLabel ~= predictions));
hist(wrong,20)
```



Published with MATLAB® R2016b

b) The accuracy for the test classification is 80.57%

c)

The predictions are obtained from the max values of the posterior matrix for every category. i.e. the row index of the max value in the 5th document gives the most likely category the document belongs to. The smoothing factor is taken as 0.1 which prevents the probabilities from going to 0.

It can be seen from the histogram that the documents are most wrongly classified for the 3rd category. We now from the map file that category 3 (comp.os.ms-windows.misc) may have a lot of common words with the other computer related categories or even the electronic category. This can confuse the classifier which would make the wrong prediction. Moreover, for the testing set there are approximately 7000 – 8000 new words used. But our classifier is only trained for 53975 words, hence some of the words in the test set documents are simply ignored. This would factor in for some of the misclassified predictions.

From the predictions it can be seen that several of the category 1 (alt.atheism) documents are misclassified as category 16 (soc.religion.christian) or 20 (talk.religion.misc) documents and category 8 (rec.autos) and 9 (rec.motorcycles) documents and category 10 (rec.sport.baseball) and category 11 (rec.sport.hockey) are often misclassified as each other. This makes sense because the topics will obviously have a lot of common words