Aditi Mallavarapu

CS 412

Question 3

fid = fopen('SMSSpamCollection'); % read file

data = fread(fid);

fclose(fid);

lcase = abs('a'):abs('z');

ucase = abs('A'):abs('Z');

caseDiff = abs('a') - abs('A');

caps = ismember(data,ucase);

data(caps) = data(caps)+caseDiff; % convert to lowercase

data(data == 9) = abs(' '); % convert tabs to spaces

validSet = [9 10 abs(' ') lcase];

data = data(ismember(data,validSet)); % remove non-space, non-tab, non-(a-z) characters

data = char(data); % convert from vector to characters

words = strsplit(data'); % split into words

% split into examples

count = 0;

examples = {};

for (i=1:length(words))

if (strcmp(words{i}, 'spam') || strcmp(words{i}, 'ham'))

count = count+1;

examples(count).spam = strcmp(words{i}, 'spam');

examples(count).words = [];

else

examples(count).words{length(examples(count).words)+1} = words{i};

end

end

%split into training and test

random\_order = randperm(length(examples));

train\_examples = examples(random\_order(1:floor(length(examples)\*.8)));

test\_examples = examples(random\_order(floor(length(examples)\*.8)+1:end));

% count occurences for spam and ham

spamcounts = javaObject('java.util.HashMap');

numspamwords = 0;

hamcounts = javaObject('java.util.HashMap');

numhamwords = 0;

function [retaccuracy,retfscore] = probabilty\_calc(samples,spamcounts,numspamwords,hamcounts,numhamwords,alpha)

truepositive =0;

truenegative =0;

falsepositive =0;

falsenegative =0;

for ( i= 1: length(samples))

messageprobspam = 1;

messageprobham = 1 ;

for(j=1:length(samples(i).words))

pspam=0;

pham=0;

word = samples(i).words{j};

if(spamcounts.get(word) != 0)

pspam = spamcounts.get(word)/(numspamwords+(alpha\*20000));

else %unseen word in training set

pspam = alpha/(numspamwords+(alpha\*20000));

end

if(hamcounts.get(word) != 0)

pham = hamcounts.get(word)/(numhamwords+(alpha\*20000));

else %unseen word in training set

pham = alpha/(numspamwords+(alpha\*20000));

end

messageprobspam = messageprobspam \* pspam;

messageprobham = messageprobham \* pham;

end

if(messageprobspam >= messageprobham)

if(samples(i).spam == 1)

truepositive = truepositive+1;

else

falsepositive = falsepositive +1;

end

else

if(samples(i).spam == 0)

truenegative = truenegative +1;

else

falsenegative = falsenegative +1;

end

end

end

%accuracy,precision, recall, Fscore

retaccuracy = (truepositive+truenegative)\*100/length(samples);

precision = truepositive/(truepositive + falsepositive);

recall = truepositive/(truepositive+falsenegative);

retfscore = (2\*precision\*recall)/(precision+recall);

end

trainaccuracy =[];

testaccuracy = [];

testFscore = [];

trainFscore =[];

alpha =[0.03125,0.0625,0.125,0.25,0.5,1];

for (k= 1:length(alpha))

%calculate accuracy

%train the model: count the occurances in spam and ham for probabilities

for (i=1:length(train\_examples))

for (j=1:length(train\_examples(i).words))

word = train\_examples(i).words{j};

if (train\_examples(i).spam == 1)

numspamwords = numspamwords+1;

current\_count = spamcounts.get(word);

if (isempty(current\_count))

spamcounts.put(word, 1+alpha(k)); % initialize by including pseudo-count prior

else

spamcounts.put(word, current\_count+1); % increment

end

else

numhamwords = numhamwords+1;

current\_count = hamcounts.get(word);

if (isempty(current\_count))

hamcounts.put(word, 1+alpha(k)); % initialize by including pseudo-count prior

else

hamcounts.put(word, current\_count+1); % increment

end

end

end

end

[trainaccuracy(k), trainFscore(k)] = probabilty\_calc(train\_examples,spamcounts,numspamwords,hamcounts,numhamwords,alpha(k));

[testaccuracy(k), testFscore(k)] = probabilty\_calc(test\_examples,spamcounts,numspamwords,hamcounts,numhamwords,alpha(k));

end

%log alpha

%scale the axes

k=[-5,-4,-3,-2,-1,0];

figure,plot(k,trainaccuracy,"-;trainaccuracy;",k,testaccuracy,"-r;testaccuracy;");

xlabel('power of 2 for alpha');

ylabel('accuarcy');

figure,plot(k,trainFscore,"-;trainFscore;",k,testFscore,"-r;testFscore;");

xlabel('power of 2 for alpha');

ylabel('Fscore');

