Pattern Recognition Assignment 1

James Jackson

November 2015

Question 1

```
close all
clear all
clc
% load dataset
load('Example_MNIST_digits.mat');
% loop over 10 digits
for i = 1: 10
     % 3/4 subplots
    subplot(3,4,i)
    % Get the average of each digit, convert it 8bit integers,
    % Reshape the row into a matrix and show the resulting image
    imshow(reshape(uint8(mean(b(labb == i, :), 1)), 28, 28)');
    % Show the title for the subplot
    title(i-1)
end
```

For output please see figure 1

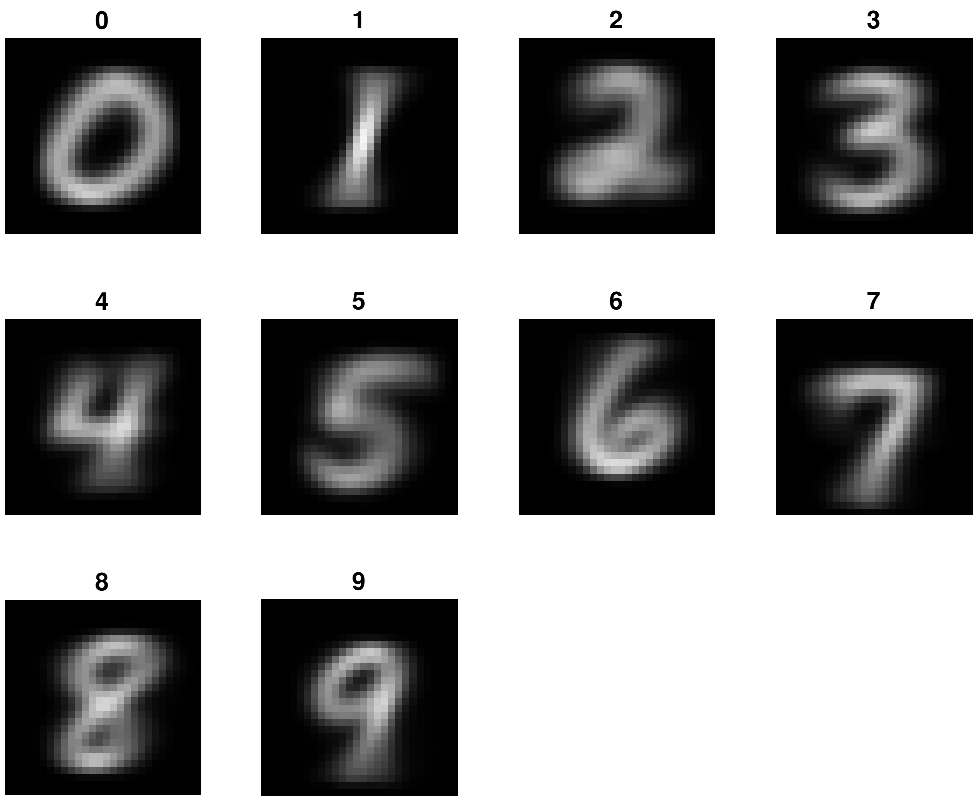

Figure 1: Average digit visualisation

Question 2

```
close all
clear all
clc

load('Example_MNIST_digits.mat');
% create D and Z variables
D = zeros(size(b));
Z = zeros(size(b,1), 2);
% 1 to number of rows in b
for i = 1:size(b, 1)
   % apply treshold
   D(i, :) = (b(i, :) <= 120);
   % transform each row into a 28X28 matrix
   c = reshape(D(i, :), 28, 28);
   % find all values in c that are 0
   [row, col] = find(~c);
   % find aspect ratio
   Z(i, 1) = (max(row)-min(row))/ ...
         (max(col)-min(col));
   % find proportion
   Z(i, 2) = numel(row)/numel(D(i, :));
end
```

Question 3

```
close all
clear all
clc

load('Example_MNIST_digits.mat');
D = b <= 120;
Z = zeros(size(b,1), 2);
% 1 to number of rows in b
for i = 1:size(b, 1)
    c = reshape(D(i, :), 28, 28);
    [row, col] = find(~c);
    % find aspect ratio
    Z(i, 1) = (max(row)-min(row))/ ...
        (max(col)-min(col));
    % find proportion

end
Z(:,2) = mean(~D, 2);

% part A
% Variable to hold errors
ers = zeros(9,1);
% Loop over 10 digits
for i = 0:9
    % use suplied classifier with the data and save the labels in nl
    nl= classifier(Z, labb == i + 1);
    % count the errors in the new labels
    ers(i + 1) = 1 - sum(nl == (labb == i + 1)) / size(nl,1);
end
figure
% plot the errors for the 10 digits for comparison further down
plot(0:9, ers, 'r-');
xlabel('Digit')
ylabel('Error rate')
print -depsc 3a

% part c
% Variable to hold errors
err = zeros(numel(unique(labb)), 1);
% Loop over 10 digits
for i = 1:10
    % current label(digit) is positive (1) everything else is negative(0)
    labba = labb == i;
    % lables array
    labels = zeros(size(Z,1),1);
    %loop over all data in Z
    for j = 1 : size(Z,1)
        % assign label for the left out element
        labels(j) = MyNMC(Z([1:j, j+1:end], :), ...
            labba([1:j, j+1:end]), Z(j, :));
    end
    % calculate the error rate
```

```
    err(i) = mean(labels ~= labba);
end
% plot the error rate for comparison
figure
plot(0:9, err, 'b-');
xlabel('Digit')
ylabel('Error rate')
print -depsc 3c
```

```
function NewLabels = MyNMC(Ztr, Ytr, Zts)
    % Get a variable holding the unique labels
    labels = unique(Ytr);
    % grpstats gets mean grouped by the labels
    mean = grpstats(Ztr, Ytr);
    for i = 1 : size(Zts,1);
        % Assign label index by getting the smallest distance when comparing
        % the current element with the means
        [~, NewLabels(i)] = min(pdist2(mean, Zts(i,:)));
    end
    % Assign label values by indexing the unique labels with
    % the supplied indeces
    NewLabels = labels(NewLabels);
end
```

Using the given classifier indicates that the digit 1 is a more distinct as it will have a lower aspect ratio and lower proportion of black as seen by figure 2

In task 3c we use the MyNMC classifier to classify the data and we notice that the error plot (figure 3) it very similar to that of the given classifier which leads to the conclusion that it is an NMC classifier. THe slight differences in error is down to the fact that in figure 3 a leaveone out NMC is being used, whereas in figure 2 the classifier is being trained on all of the data.

Actual error rates below.

```
    3a        3c
    0.3338    0.3338
    0.1894    0.1894
    0.3058    0.3058
    0.4680    0.4680
    0.4600    0.4600
    0.3324    0.3324
    0.4654    0.4654
    0.4068    0.4068
    0.4882    0.4882
    0.4138    0.4138
```

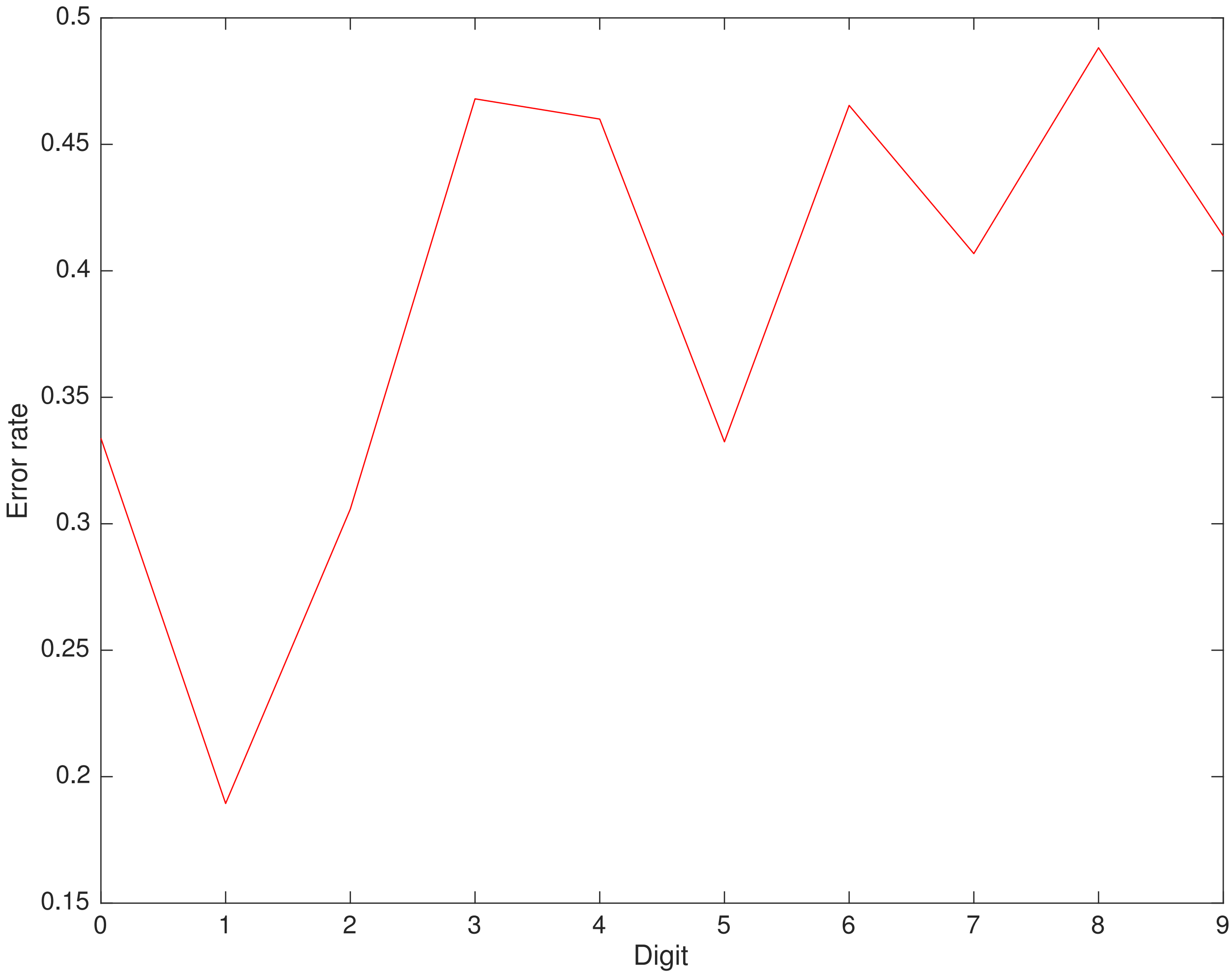

Figure 2: Plot of given classifier error (3a)

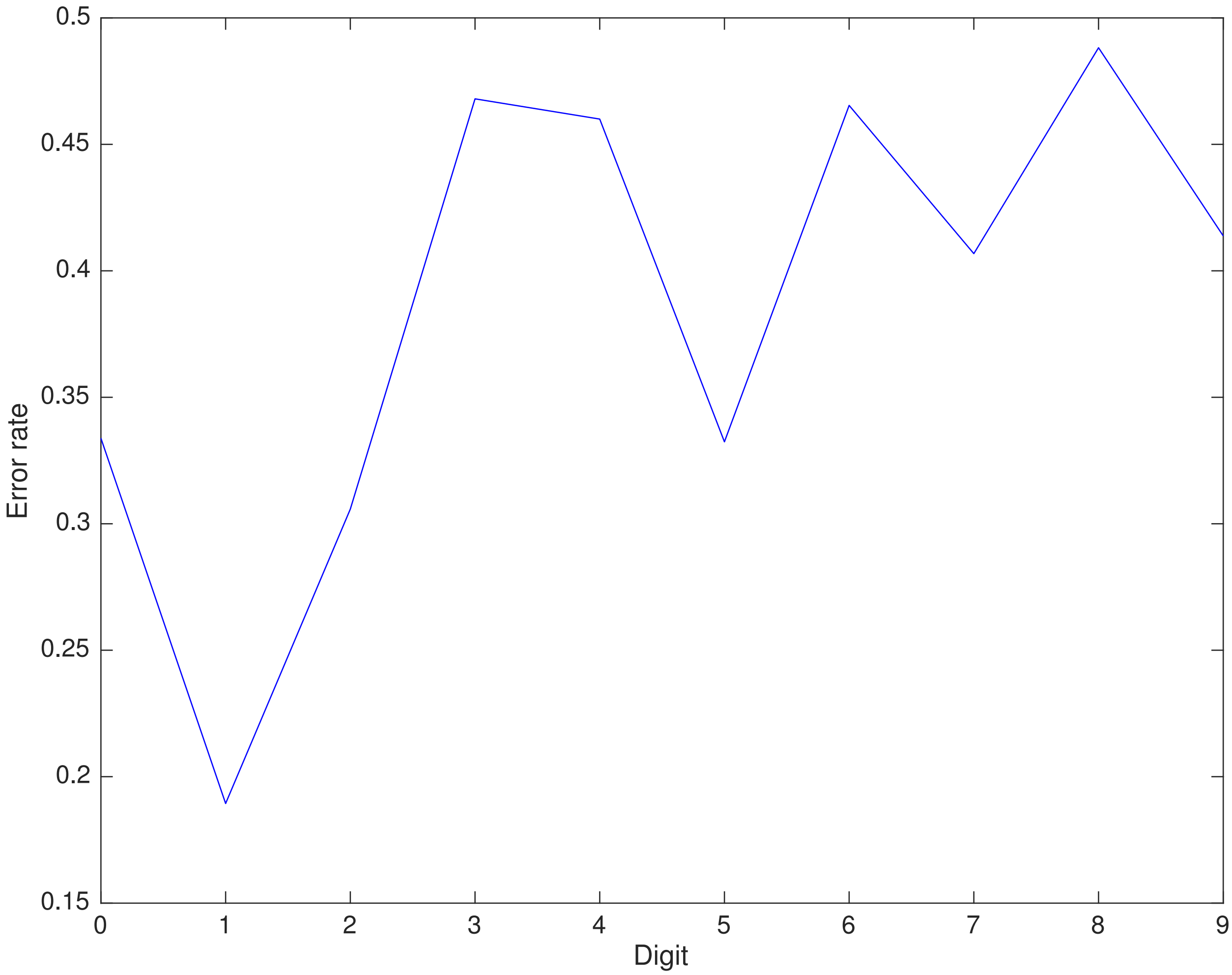

Figure 3: Plot of given classifier error (3c)

Question 4

```
close all
clear all
clc

load('Example_MNIST_digits.mat');
D = b <= 120;


% part a

ers = zeros(9,1);
for i = 1:10
    nl= classifier(b, labb == i);
    ers(i) = 1 - sum(nl == (labb == i)) / size(nl,1);
end

ers2 = zeros(9,1);
for i = 0:9
    nl= classifier(double(D), labb == i + 1);
    ers2(i + 1) = 1 - sum(nl == (labb == i + 1)) / size(nl,1);
end

figure
hold on
% plot the error rates for both classifier with Original Data and D
plot(0:9, ers, 'r-');
plot(0:9, ers2, 'b-');
xlabel('Digit')
ylabel('Error rate')
print -depsc plot4
```

When not considering aspect ratio and ink amount the NMC classifier returns different results in terms of error rate, there are differing peaks but better overall error rate as seen in figure 4

Error rates

```
    Original   Threshold
    0.0522     0.0528
    0.0646     0.0636
    0.0786     0.0770
    0.1002     0.0974
    0.1268     0.1262
    0.1780     0.1774
    0.0666     0.0660
    0.0784     0.0780
    0.1206     0.1220
    0.1828     0.1816
```

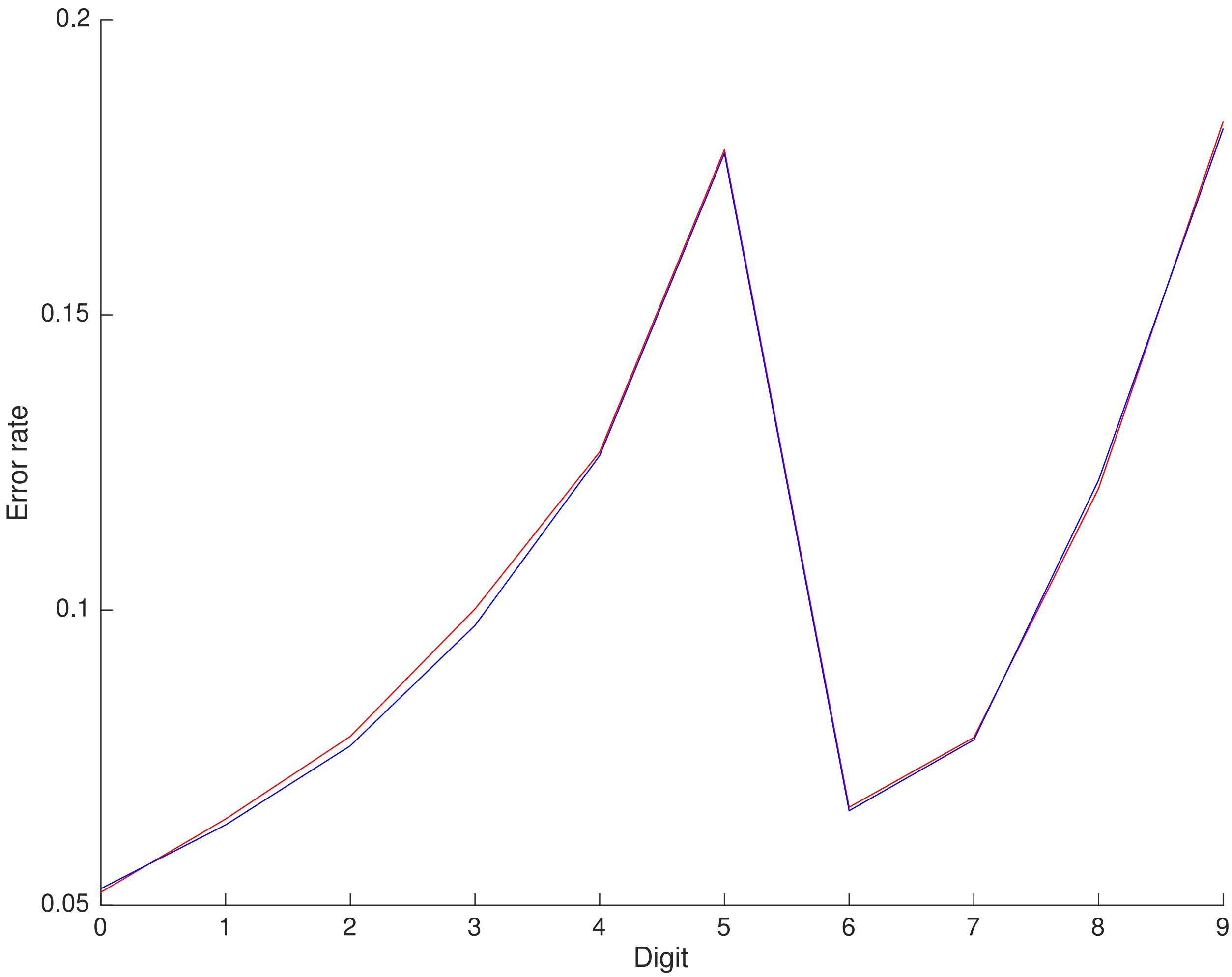

Figure 4: Plot of given classifier error (4)

Question 5

```
function [RocMatrix, Closest] = myRoc(data, labels)
[~,ind] = sort(data);
d = labels(ind);

RocMatrix = zeros(numel(d), 2);
    for i = 1:numel(d)
        % Get current position in the data (split)
        dstart = d(1:i);
        dend = d(i+1:end);
        % Find all the:
            % True Positives and Negatives and
            % False Positives and Negatives
        tPos = sum(dend == 2);
        tNeg = sum(dstart == 1);
        fPos = sum(dend == 1);
        fNeg = sum(dstart == 2);
        % Put the Sensitivity and specificity for each split
        % into the roc matrix in columns 1 and 2 respectively
        RocMatrix(i, 1) = tNeg/(tNeg+fPos);
        RocMatrix(i, 2) = tPos/(tPos+fNeg);
    end
    [~, ind] = min(pdist2([1 - RocMatrix(:,1) RocMatrix(:,2)], [0 1]));
    Closest = RocMatrix(ind,:);
end
```

Question 6

```
close all
clear all
clc

load('Example_MNIST_digits.mat');
D = b <= 120;
Z = zeros(size(b,1), 2);
Z(:,2) = mean(~D, 2);
% 1 to number of rows in b
for i = 1:size(b, 1)
    c = reshape(D(i, :), 28, 28);
    [row, col] = find(~c);
    % find aspect ratio
    Z(i, 1) = (max(row)-min(row))/ ...
        (max(col)-min(col));
    % find proportion

end

labbs = labb == 2;
% Use the MyRoc function to compare the Aspect Ratio
% and Proportion
[rocMatrixAspect, aClosest] = MyRoc(Z(:,1), ~labbs + 1);
[rocMatrixProportion, pClosest] = MyRoc(Z(:,2), ~labbs + 1);
figure
hold on
grid on
axis square
set(gca, 'ytick', 0:0.2:1)
plot(1 - rocMatrixAspect(:,1), rocMatrixAspect(:,2), 'r-');
plot(1 - rocMatrixProportion(:,1), rocMatrixProportion(:,2), 'b-');
xlabel('1-specifictiy')
ylabel('sensitivity')
legend('Aspect Ratio','Proportion Black')
print -depsc plot6
```

For output, see plot in figure 5 The closest points to [0,1] in each case are

```
    Aspect Ratio
    0.8315    0.8633
    Proportion
    0.8763    0.8204
```

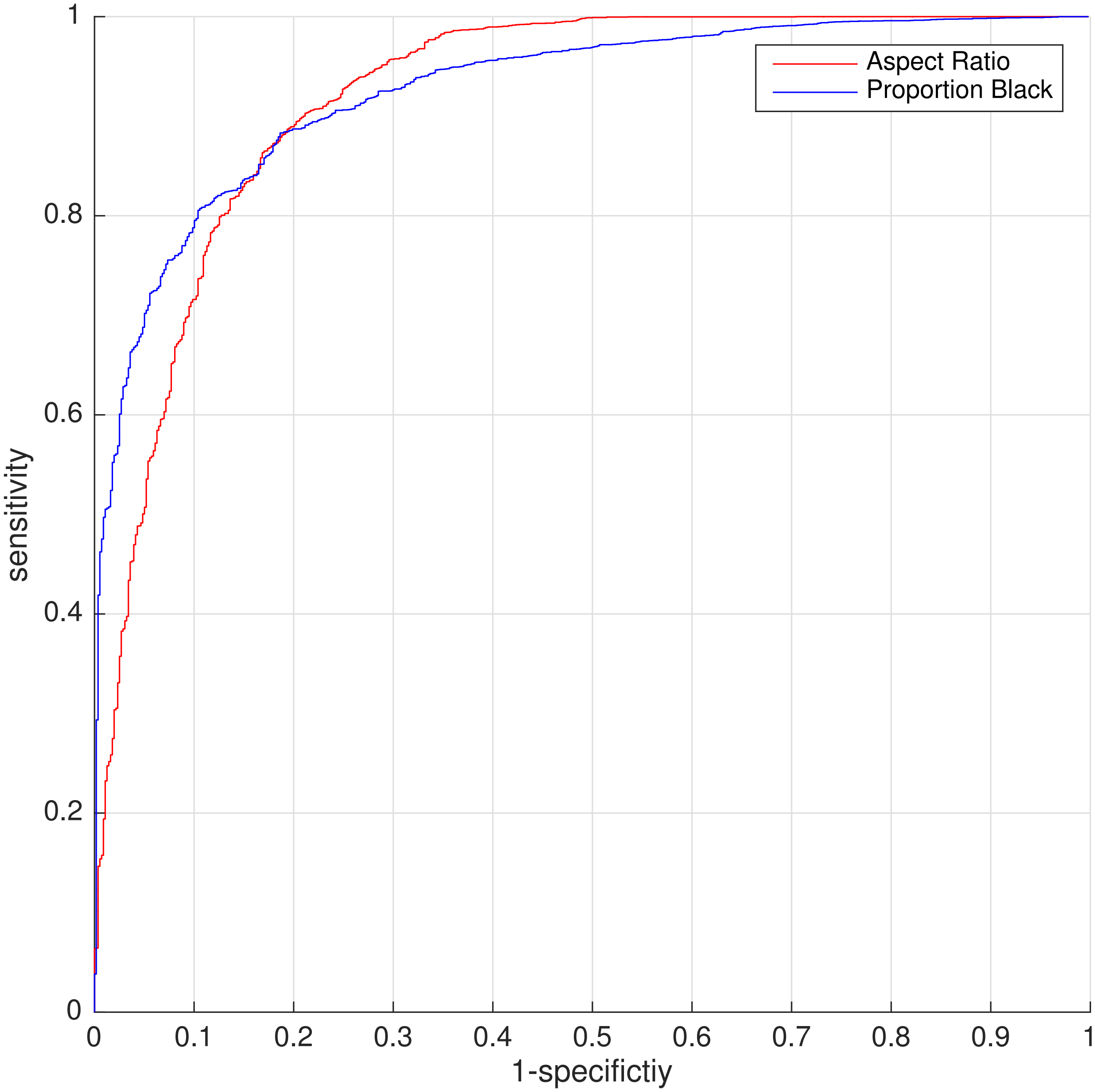

Figure 5: Plot rock curve for aspect ratio and proportion (6)

Question 7

```
close all
clear all
clc

load('Example_MNIST_digits.mat');
% create variable to hold all assigned labels in bagging
assignedLabs = zeros(5000,9);
for i = 1 : 9
    % Build an index of 5000 values with repitition
    % in the range 1 to length of dataset
    ind = randi(size(b, 1), size(b, 1), 1);
    % pass the indexed dataset to the MyNMC function
    % as well as the full dataset for testing
    assignedLabs(:,i) = MyNMC(b(ind, :), labb(ind), b);
end
% Get the assigned label by selecting the most assigned
% label from the output for each value
alabb = mode(assignedLabs,2);
% calculate the accuracy of the classifier ensemble
acc = mean(alabb == labb);

% construct the confusion matrix
c = zeros(10);
% loop over the rows in the
for i = 1 : size(alabb, 1);
  % increment the value at the posiition representing where the
  % assigned label is
  c(labb(i),alabb(i)) = c(labb(i),alabb(i)) + 1;
end

% used to test accuracy of confusion matrix
% con = confusionmat(labb, avgLabs);
% mean(c(:)==con(:))

% run single nmc on the data and output the accuracy
snmc = MyNMC(b,labb,b);
snmcAcc = mean(snmc==labb)
```

Confusion matrix

424	0	2	1	0	46	16	1	2	2
0	537	2	3	0	6	2	0	8	0
10	32	422	17	20	3	15	9	15	2
4	16	13	375	1	29	4	3	25	10
0	7	2	0	393	0	4	1	4	66
7	37	4	56	5	320	11	3	7	19
9	16	6	0	15	17	450	0	3	0
1	28	4	0	9	2	1	422	6	33
4	21	4	48	3	15	4	1	358	19
6	11	3	9	48	4	1	19	8	369

The accuracy for the ensemble is around 0.81 and the single NMC has a similar accuracy at around 0.81.