AMATH482 hoomework4

Rui Zhu

March 2021

Abstract

In this assignment we are given a set of 60000 hand-written digits pictures. And we will try using **SVD**, **Linear Discriminant Analysis** (LDA), **Support Vector Machines** (SVM), and **Decision Trees** to classify them. We will also compare the accuracy between digits and approaches.

1 Introduction and Overview

Singular Value Decomposition (SVD) is a way of factoring matrices into three separate components. It is widely used in the field of Data Analysis partly due to its dimensional reductions ability. **Principle Component Analysis** (PCA) is an application of SVD. Generally speaking, we will see what low-rank approximations mean geometrically when we are working with data matrices. These two strategies are applied in the last assignment: **PCA and a Spring-mass System**. In this assignment, we are also going to use SVD and PCA in order to perform **Linear Discriminant Analysis** on the given data set.

2 Theoretical Background

2.1 SVD

SVD factors a matrix into a product of three matrices. For example, if we apply SVD to a matrix A We have $A = U\Sigma V^*$ where U and V are unitary matrices that simply lead to rotation and Σ is a diagonal matrix that scales an image as prescribed by the singular values. There are many benefits of performing SVD, a major one is to remove redundancy and describe the signals or data with a maximal variance.

2.2 LDA

The main goal of performing LDA is to find a suitable projection that maximizes the distance between the inter-class data while minimizing the intra-class data.

2.2.1 Implementing LDA for 2 Datasets

We first calculate the means for each of our groups for each feature. As above, we will call these μ_1 and μ_2 where μ are column vectors. We then define the between-class scatter matrix (a measure of the variance between the groups (between the means)):

$$S_B = (\mu_2 - \mu_1)(\mu_2 - \mu_1)^T$$

Then we can define the within-class scatter matrix(a measure of the variance within each group):

$$S_w = \sum_{j=1}^{2} \sum_x (x - \mu_j)(x - \mu_j)^T$$

Finally we want to find a vector w such that:

$$w = argmax \frac{w^T S_B w}{w^T S_w w}$$

The vector w above has been proven to be the eigenvector corresponding to the largest eigenvalue of the generalized eigenvalue problem (which can be easily solved using MATLAB's command `eig()`):

$$S_B w = \lambda S_w w$$

2.2.2 Implementing LDA for more Datasets

Generally speaking, implementing LDA for more than 2 datasets is very similar, we just slightly change the way we calculate the between-class scatter matrix and the within-class scatter matrix:

$$S_B = \sum_{j=1}^{N} (\mu_2 - \mu_1)(\mu_2 - \mu_1)^T$$

$$S_w = \sum_{j=1}^{N} \sum_x (x - \mu_j)(x - \mu_j)^T$$

3 Algorithm Implementation and Development

3.1 Initial Setup

1. Load the train/test dataset using `mnist_parse()` command.
2. Obtain the `images` and `labels` matrix which contain the digit images and corresponding digit as number.
3. Reshape the train/test data and change the data type as `double`
4. For reshaped train/test data, subtract the rowwise mean for future SVD.

3.2 Apply and interpret SVD

1. Apply SVD using `svd(, 'econ')` command.
2. Plot principal components (U: the left singular vectors)

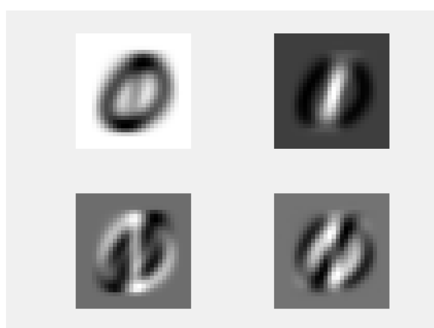

Figure 1: Plot of the first four principal components

   Here we use digit 0 images and digit 1 images. We can identify a clear '0' and '1' from the top two plots

3. Plot the singular values

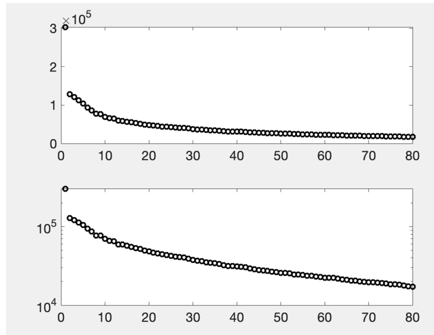

Figure 2: Plot of the singular values

It is clear that the first mode is dominant. But, we can also see that there isn't a sharp drop-off in the singular values. That is, we have a heavy-tail distribution, meaning there is still information in the later modes.

4. Plot the right singular vectors in V

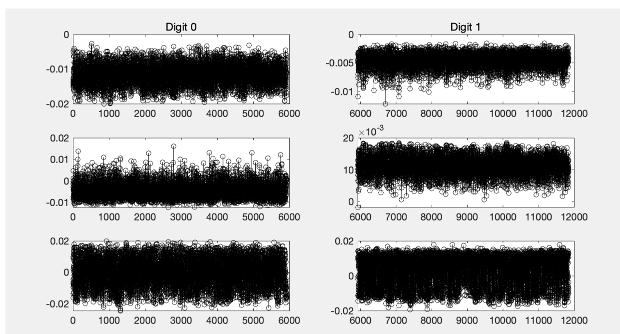

Figure 3: Plot of the right singular vectors

On inspection, the first and third modes are not that useful for differentiating between digit 0 and digit 1. The second one is though. The digit 0s are almost all negative, while the digit 1s have opposite sign. Recall that modes 1 and 2 both had very clear visualization of digit 0 and digit 1, but the digit 1 in mode 1 are black while the digit 1 in mode 2 are white. Loosely, we can understand what we are seeing in the following way. The first and second digit 0 modes above are both negative, thus 'flipping' the black and white cells in the first two principal components. In the case of digit 1, we have opposite signs in the first two right singular vectors above, meaning that when we multiply them by the respective principal components and add them together, the visualizations will both be white.

5. Project onto three selected V-modes

3.3 Apply and Interpret LDA

1. Build a linear classifier (LDA) that can reasonable identify digit 0 and digit 1
2. View the performance of LDA via histogram and calculate the accuracy

Algorithm 1:

Data: `trainData.mat, labels.mat`

Seperate the target digit from `trainData.mat` based on the `labels.mat` using `find()` command
Reshape the two digit data so that the longer one has the same length as the shorter one
Set the number of PCA modes (variable `feature`)
Project onto principal components: `// X = USV' --> U'X = SV'`
Calculate the scatter matrices
Find w using the `eig()` command
Project on w
Set a threshold for our classifier, and make first digit always below the threshold

4 Computational Results

4.1 Build a linear classifier (LDA) for digit 0 and 1

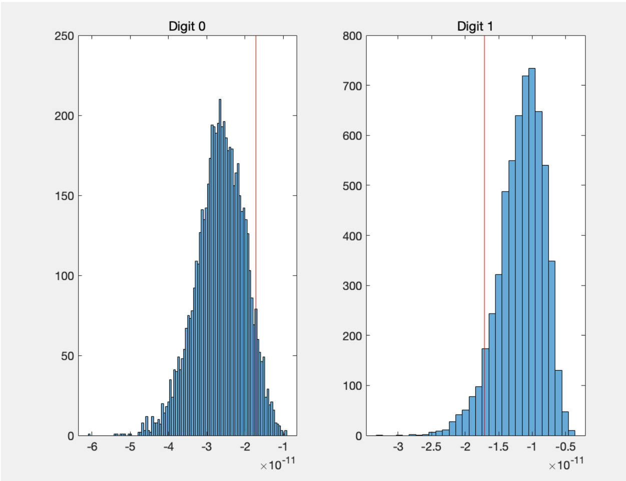

Figure 4: Histograms of the digit 0 and digit 1 values with decision- making threshold.

LDA did a pretty good job. Most of digit zeros can be correctly classified (below the threshold) and most of digit ones can be correctly classified as well (above the threshold).

4.2 Accuracy of LDA among 10 digits

Pairs	Accuracy	Pairs	Accuracy	Pairs	Accuracy	Pairs	Accuracy	Pairs	Accuracy	Pairs	Accuracy	Pai
0,1	93.52%	1,2	88.99%	2,3	54.29%	3,4	61.26%	4,5	62.56%	5,6	54.45%	6,7
0,2	62.32%	1,3	86.54%	2,4	65.56%	3,5	57.35%	4,6	58.44%	5,7	57.61%	6,8
0,3	65.59%	1,4	80.02%	2,5	61.04%	3,6	52.38%	4,7	53.85%	5,8	61.19%	6,9
0,4	75.88%	1,5	81.55%	2,6	56.52%	3,7	64.33%	4,8	65.78%	5,9	53.61%	
0,5	71.78%	1,6	85.10%	2,7	69.26%	3,8	53.80%	4,9	50.82%			
0,6	68.25%	1,7	76.98%	2,8	49.87%	3,9	61.44%					
0,7	68.19%	1,8	89.42%	2,9	65.84%							
0,8	62.06%	1,9	80.97%									
0,9	75.89%											

Table 1: Accuracy of LDA among 10 digits

Digit 2 and 8 are the hardest to separate (accuracy of 49.87%)
Digit 0 and 1 are the easiest to separate (accuracy of 93.52%)

4.3 Decision Trees and Support Vector Machines (SVM)

To perform Decision Trees, we adopt the MATLAB command `fitctree(), view(), and kfoldLoss()`
Decision Trees can successfully identify 54.44% of all 10 digits:

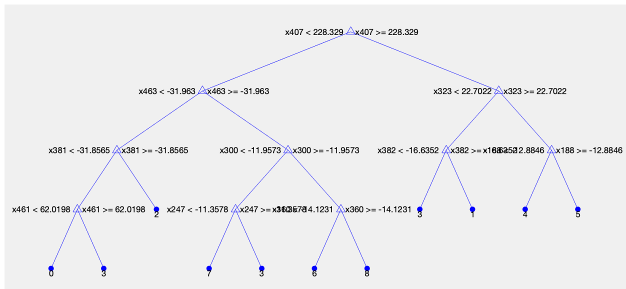

Figure 5: Decision tree classifiers of all 10 digits

Decision Tree gives us 84.84% accuracy of identifying digit 2 and 8; SVM gives 92.34% whereas LDA gives only 49.87%. Thus, SVM and Decision Trees performs well on the hardest duo.

Decision Tree gives us 99.31% accuracy of identifying digit 0 and 1; SVM gives 99.78% whereas LDA gives 93.52%. Thus, SVM and Decision Trees performs well on the easiest duo as well.

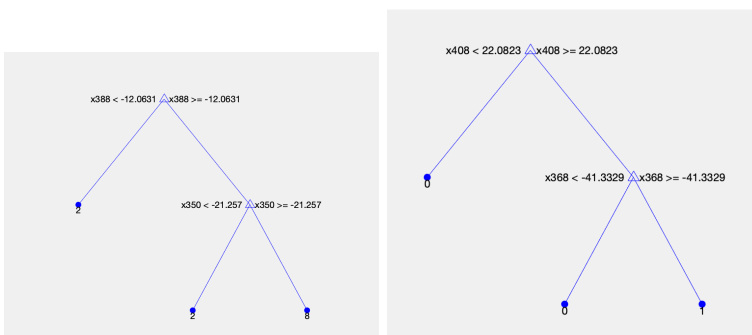

Figure 6: Decision tree classifiers on hardest duo and easiest duo

5 Summary and Conclusions

We implemented LDA to 45 different pairs of digit images to see the classification accuracy. And LDA gives us a variety of results (from 50% accuracy to 94% accuracy). This might be caused by the number of `feature` or the general shape of the digits themselves. We also compare the performance of LDA with two front-edge techeniques: Decision Trees and SVM on the easiest duo and hardest duo. Generally speaking, we successfully classified the digits and also see the so called black-box characteristic of machine learning.

Appendix A MATLAB Functions

- `k = find(X)` returns a vector containing the linear indices of each nonzero element in array X.
- `M = mean(A,dim)` returns the mean along dimension dim. For example, if A is a matrix, then mean(A,2) is a column vector containing the mean of each row.

- `[U,S,V] = svd(A)` performs a singular value decomposition of matrix A, such that $A = U * \Sigma * V'$.
- `fitctree()`Fit binary decision tree for multiclass classification.
- `fitcsvm()` Train support vector machine (SVM) classifier for one-class and binary classification.

Appendix B MATLAB Code

```
function [U,S,V,threshold,w,sort_first,sort_second] = digit_trainer(digitData1, digitData2, feature)

    feature = 20;
    n_first = size(digitData1,2);
    n_second = size(digitData2,2);
    [U, S, V] = svd([digitData1 digitData2], 'econ');
    digits = S*V'; % projection onto principal components: X = USV' --> U'X = SV'
    first = digits(1:feature,1:n_first);
    second = digits(1:feature,n_second+1:n_first+n_second);
    m_first = mean(digitData1,2);
    m_second = mean(digitData2,2);

    Sw = 0; % within class variances
    for k=1:n_first
        Sw = Sw + (digitData1(:,k) - m_first)*(digitData2(:,k) - m_first)';
    end

    for k=1:n_second
        Sw = Sw + (digitData2(:,k) - m_second)*(digitData2(:,k) - m_second)';
    end

    Sb = (m_first-m_second)*(m_first-m_second)'; % between class

    [V2, D] = eig(Sb,Sw); % linear discriminant analysis
    [lambda, ind] = max(abs(diag(D)));
    w = V2(:,ind);
    w = w/norm(w,2);

    v_first = w'*digitData1;
    v_second = w'*digitData2;

    if mean(v_first) > mean(v_second) %Make first digit always below
        w = -w;
        v_first = -v_first;
        v_second = -v_second;
    end

    sort_first = sort(v_first);
    sort_second = sort(v_second);
    t1 = length(sort_first);
    t2=1;

    while sort_first(t1)>sort_second(t2)
        t1 = t1-1;
        t2 = t2+1;
    end
```

```
    threshold = (sort_first(t1)+sort_second(t2))/2;

end

%% Load the train data set

[testimages, testlabels] = mnist_parse('t10k-images.idx3-ubyte', 't10k-labels.idx1-ubyte');

[images, labels] = mnist_parse('train-images.idx3-ubyte', 'train-labels.idx1-ubyte');
%% Reshape the train data
trainData = double(reshape(images, size(images,1)*size(images,2), []));
testData = double(reshape(testimages, size(testimages,1)*size(testimages,2), []));

% %% Edge detectation using Wavelet Transform
% [m,n] = size(trainData); % 784 * 60000
% pxl = sqrt(m);
% nw = m/4;
% imageData = zeros(nw,n);
% for k = 1:n
%     X = reshape(trainData(:,k),pxl,pxl);
%     [~,cH,cV,~] = dwt2(X,'harr');
%     cod_cH1 = rescale(abs(cH));
%     cod_cV1 = rescale(abs(cV));
%     cod_edge = cod_cH1 + cod_cV1;
%     imageData(:,k) = reshape(cod_edge,nw,1);
% end

%% Subtract the rowwise mean
[m,n] = size(trainData);
for i = 1:n
    a = mean(trainData(:,i));
    for j = 1:m
        trainData(j,i) = trainData(j,i) - a;
    end
end

%% SVD
[U,S,V] = svd(trainData, 'econ');

%% Plot principal components (U: the left singular vectors)
for k=1:4
subplot(2,2,k)
ut1 = reshape(U(:,k),28,28);
ut2 = rescale(ut1);
imshow(ut2)
end
%% Plot singular values (S)
figure(2)
subplot(2,1,1)
plot(diag(S),'ko','Linewidth',2)
set(gca,'Fontsize',16,'Xlim',[0 80])
subplot(2,1,2)
semilogy(diag(S),'ko','Linewidth',2)
```

```
set(gca,'Fontsize',16,'Xlim',[0 80])

%% Plot principal components (V: the right singular vectors)
% Projections onto the first 3 PCA modes (the first three columns)
figure(3)
for k=1:5
    subplot(5,1,k)
    plot(1:60,V(1:60,k),'ko-')
end

figure(3)
for k=1:3
subplot(3,2,2*k-1)
plot(1:5923,V(1:5923,k),'ko-')
subplot(3,2,2*k)
plot(5923:11846,V(5923:11846,k),'ko-')
end
subplot(3,2,1), set(gca,'Fontsize',12), title('Digit 0')
subplot(3,2,2), set(gca,'Fontsize',12), title('Digit 1')
subplot(3,2,3), set(gca,'Fontsize',12)
subplot(3,2,4), set(gca,'Fontsize',12)
subplot(3,2,5), set(gca,'Fontsize',12)
subplot(3,2,6), set(gca,'Fontsize',12)
%% Project onto three selected V-modes (columns) colored by their digit label

%%%%%%%%%%?%%%%%%%%%%%%%
plot3(V(1:60000,2), V(1:60000,3), V(1:60000,5))


%% Perform linear classifier (LDA) on digit 2 and 9
ind0=find(labels==0);
ind1=find(labels==1);
ind2=find(labels==2);
ind3=find(labels==3);
ind4=find(labels==4);
ind5=find(labels==5);
ind6=find(labels==6);
ind7=find(labels==7);
ind8=find(labels==8);
ind9=find(labels==9);

digit2=[];
digit9=[];
for i = 1:length(ind2)
    digit2 = [digit2; trainData(:,ind2(i))];
end
digit2Data = reshape(digit2,[784,5958]);
digit2Data = digit2Data(:,1:5949); %match the size to use SVD
for i = 1:length(ind9)
    digit9 = [digit9; trainData(:,ind9(i))];
end
digit9Data = reshape(digit9,[784,5949]);

%% LDA
```

```
feature = 20;
[U,S,V,threshold,w,sort2,sort9] = digit_trainer(digit2Data,digit9Data,feature);
%% Performance of our code
figure(5)
subplot(1,2,1)
histogram(sort2,30); hold on, plot([threshold threshold], [0 1000],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 2')
subplot(1,2,2)
histogram(sort9,30); hold on, plot([threshold threshold], [0 1000],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 9')

correct_29 = sum(sort2 < threshold);
accuracy_29 = correct_29 / 5949 % 65.84% of digit 2 and 9 are successfully identified

%% Find the easiest and hardest duos
digit0=[];
digit1=[];
digit3=[];
digit4=[];
digit5=[];
digit6=[];
digit7=[];
digit8=[];

for i = 1:length(ind0)
    digit0 = [digit0; trainData(:,ind0(i))];
end
digit0Data = reshape(digit0,[784,5923]);

for i = 1:length(ind1)
    digit1 = [digit1; trainData(:,ind1(i))];
end
digit1Data = reshape(digit1,[784,6742]);

for i = 1:length(ind3)
    digit3 = [digit3; trainData(:,ind3(i))];
end
digit3Data = reshape(digit3,[784,6131]);

for i = 1:length(ind4)
    digit4 = [digit4; trainData(:,ind4(i))];
end
digit4Data = reshape(digit4,[784,5842]);

for i = 1:length(ind5)
    digit5 = [digit5; trainData(:,ind5(i))];
end
digit5Data = reshape(digit5,[784,5421]);

for i = 1:length(ind6)
    digit6 = [digit6; trainData(:,ind6(i))];
end
```

```
digit6Data = reshape(digit6,[784,5918]);

for i = 1:length(ind7)
    digit7 = [digit7; trainData(:,ind7(i))];
end
digit7Data = reshape(digit7,[784,6265]);

for i = 1:length(ind8)
    digit8 = [digit8; trainData(:,ind8(i))];
end
digit8Data = reshape(digit8,[784,5851]);

%% Compare accuracy

%0 and 1
digit1Data_1 = digit1Data(:,1:5923); %match the size to use SVD
[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit0Data,digit1Data_1,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 0')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 800],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 1')


correct_01 = sum(sort1 < threshold);
accuracy_01 = correct_01 / 5923 % 93.52% of digit 1 and 2 are successfully identified

%% 0 and 2
digit2Data_1 = digit2Data(:,1:5923); %match the size to use SVD
[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit0Data,digit2Data_1,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 0')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 800],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 2')


correct_02 = sum(sort1 < threshold);
accuracy_02 = correct_02 / 5923 % 62.32% of digit 0 and 2 are successfully identified

%% 0 and 3

digit3Data_1 = digit3Data(:,1:5923); %match the size to use SVD
[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit0Data,digit3Data_1,20);
```

```
figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 0')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 800],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 3')


correct_03 = sum(sort1 < threshold);
accuracy_03 = correct_03 / 5923 % 65.59% of digit 0 and 3 are successfully identified

%% 0 and 4

digit0Data_1 = digit0Data(:,1:5842); %match the size to use SVD
[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit0Data_1,digit4Data,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 0')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 800],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 4')


correct_04 = sum(sort1 < threshold);
accuracy_04 = correct_04 / 5842 % 75.88% of digit 0 and 4 are successfully identified

%% 0 and 5

digit0Data_1 = digit0Data(:,1:5421); %match the size to use SVD
[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit0Data_1,digit5Data,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 0')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 800],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 5')


correct_05 = sum(sort1 < threshold);
accuracy_05 = correct_05 / 5421 % 71.18% of digit 0 and 3 are successfully identified

%% 0 and 6
```

```
digit0Data_1 = digit0Data(:,1:5918); %match the size to use SVD
[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit0Data_1,digit6Data,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 0')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 800],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 6')


correct_06 = sum(sort1 < threshold);
accuracy_06 = correct_06 / 5918 % 68.25% of digit 0 and 3 are successfully identified


%% 0 and 7

digit7Data_1 = digit7Data(:,1:5923); %match the size to use SVD
[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit0Data,digit7Data_1,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 0')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 800],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 7')


correct_07 = sum(sort1 < threshold);
accuracy_07 = correct_07 / 5923 % 68.19% of digit 0 and 3 are successfully identified

%% 0 and 8

digit0Data_1 = digit0Data(:,1:5851); %match the size to use SVD
[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit0Data_1,digit8Data,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 0')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 800],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 8')
```

```
correct_08 = sum(sort1 < threshold);
accuracy_08 = correct_08 / 5851 % 62.06% of digit 0 and 3 are successfully identified

%% 0 and 9

digit9Data_1 = digit9Data(:,1:5923); %match the size to use SVD
[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit0Data,digit9Data_1,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 0')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 800],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 9')


correct_09 = sum(sort1 < threshold);
accuracy_09 = correct_09 / 5923 % 75.89% of digit 0 and 3 are successfully identified


%% 1 and 2

digit1Data_1 = digit1Data(:,1:5949); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit1Data_1,digit2Data,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 1')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 2')


correct_12 = sum(sort1 < threshold);
accuracy_12 = correct_12 / 5949 % 88.99% of digit 1 and 2 are successfully identified


%% 1 and 3

digit1Data_1 = digit1Data(:,1:6131); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit1Data_1,digit3Data,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
```

```
title('Digit 1')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 3')


correct_13 = sum(sort1 < threshold);
accuracy_13 = correct_13 / 6131 % 86.54% of digit 1 and 3 are successfully identified

%% 1 and 4

digit1Data_1 = digit1Data(:,1:5842); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit1Data_1,digit4Data,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 1')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 4')


correct_14 = sum(sort1 < threshold);
accuracy_14 = correct_14 / 5842 % 80.02% of digit 1 and 4 are successfully identified

%% 1 and 5

digit1Data_1 = digit1Data(:,1:5421); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit1Data_1,digit5Data,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 1')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 5')


correct_15 = sum(sort1 < threshold);
accuracy_15 = correct_15 / 5421 % 81.55% of digit 1 and 5 are successfully identified


%% 1 and 6

digit1Data_1 = digit1Data(:,1:5918); %match the size to use SVD
```

```
[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit1Data_1,digit6Data,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 1')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 6')


correct_16 = sum(sort1 < threshold);
accuracy_16 = correct_16 / 5918 % 85.10% of digit 1 and 4 are successfully identified


%% 1 and 7

digit1Data_1 = digit1Data(:,1:6265); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit1Data_1,digit7Data,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 1')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 7')


correct_17 = sum(sort1 < threshold);
accuracy_17 = correct_17 / 6265 % 76.98% of digit 1 and 7 are successfully identified


%% 1 and 8

digit1Data_1 = digit1Data(:,1:5851); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit1Data_1,digit8Data,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 1')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 8')
```

```
correct_18 = sum(sort1 < threshold);
accuracy_18 = correct_18 / 5851 % 89.42% of digit 1 and 8 are successfully identified


%% 1 and 9

digit1Data_1 = digit1Data(:,1:5923); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit1Data_1,digit9Data,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 1')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 9')


correct_19 = sum(sort1 < threshold);
accuracy_19 = correct_19 / 5923 % 80.97% of digit 1 and 9 are successfully identified


%% 2 and 3

digit3Data_1 = digit3Data(:,1:5949); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit2Data,digit3Data_1,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 2')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 3')


correct_23 = sum(sort1 < threshold);
accuracy_23 = correct_23 / 5949 % 54.29% of digit 2 and 3 are successfully identified



%% 2 and 4

digit2Data_1 = digit2Data(:,1:5842); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit2Data_1,digit4Data,20);
```

```
figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 2')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 4')


correct_24 = sum(sort1 < threshold);
accuracy_24 = correct_24 / 5842 % 65.56% of digit 2 and 4 are successfully identified



%% 2 and 5

digit2Data_1 = digit2Data(:,1:5421); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit2Data_1,digit5Data,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 2')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 5')


correct_25 = sum(sort1 < threshold);
accuracy_25 = correct_25 / 5421 % 61.04% of digit 2 and 5 are successfully identified

%% 2 and 5

digit2Data_1 = digit2Data(:,1:5918); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit2Data_1,digit6Data,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 2')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 6')
```

```
correct_26 = sum(sort1 < threshold);
accuracy_26 = correct_26 / 5918 % 56.52% of digit 2 and 6 are successfully identified

%% 2 and 7

digit7Data_1 = digit7Data(:,1:5949); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit2Data,digit7Data_1,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 2')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 7')


correct_27 = sum(sort1 < threshold);
accuracy_27 = correct_27 / 5949 % 69.26% of digit 2 and 7 are successfully identified

%% 2 and 8

digit2Data_1 = digit2Data(:,1:5851); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit2Data_1,digit8Data,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 2')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 5')


correct_28 = sum(sort1 < threshold);
accuracy_28 = correct_28 / 5851 % 49.87% of digit 2 and 8 are successfully identified

%% 3 and 4

digit3Data_1 = digit3Data(:,1:5842); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit3Data_1,digit4Data,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 3')
```

```
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 4')


correct_34 = sum(sort1 < threshold);
accuracy_34 = correct_34 / 5842 % 61.26% of digit 3 and 4 are successfully identified

%% 3 and 5

digit3Data_1 = digit3Data(:,1:5421); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit3Data_1,digit5Data,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 3')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 5')


correct_35 = sum(sort1 < threshold);
accuracy_35 = correct_35 / 5421 % 57.35% of digit 3 and 5 are successfully identified

%% 3 and 6

digit3Data_1 = digit3Data(:,1:5918); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit3Data_1,digit6Data,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 3')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 6')


correct_36 = sum(sort1 < threshold);
accuracy_36 = correct_36 / 5918 % 52.38% of digit 3 and 6 are successfully identified

%% 3 and 7

digit3Data_1 = digit3Data(:,1:5949); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit3Data_1,digit7Data,20);
```

```
figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 3')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 7')


correct_37 = sum(sort1 < threshold);
accuracy_37 = correct_37 / 5949 % 64.33% of digit 3 and 7 are successfully identified

%% 3 and 8

digit3Data_1 = digit3Data(:,1:5851); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit3Data_1,digit8Data,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 3')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 8')


correct_38 = sum(sort1 < threshold);
accuracy_38 = correct_38 / 5851 % 53.80% of digit 3 and 8 are successfully identified

%% 3 and 9

digit3Data_1 = digit3Data(:,1:5949); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit3Data_1,digit9Data,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 3')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 9')


correct_39 = sum(sort1 < threshold);
accuracy_39 = correct_39 / 5949 % 61.44% of digit 3 and 9 are successfully identified
```

```
%% 4 and 5

digit5Data_1 = digit5Data(:,1:5842); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit4Data,digit5Data,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 4')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 5')


correct_45 = sum(sort1 < threshold);
accuracy_45 = correct_45 / 5842 % 62.56% of digit 3 and 7 are successfully identified

%% 4 and 6

digit6Data_1 = digit6Data(:,1:5842); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit4Data,digit6Data_1,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 4')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 6')


correct_46 = sum(sort1 < threshold);
accuracy_46 = correct_46 / 5842 % 58.44% of digit 4 and 6 are successfully identified

%% 4 and 7

digit7Data_1 = digit7Data(:,1:5842); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit4Data,digit7Data_1,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 4')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
```

```
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 7')


correct_47 = sum(sort1 < threshold);
accuracy_47 = correct_47 / 5842 % 53.85% of digit 4 and 7 are successfully identified


%% 4 and 8

digit8Data_1 = digit8Data(:,1:5842); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit4Data,digit8Data_1,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 4')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 8')


correct_48 = sum(sort1 < threshold);
accuracy_48 = correct_48 / 5842 % 65.78% of digit 4 and 8 are successfully identified

%% 4 and 9

digit9Data_1 = digit9Data(:,1:5842); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit4Data,digit9Data_1,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 4')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 9')


correct_49 = sum(sort1 < threshold);
accuracy_49 = correct_49 / 5842 % 50.82% of digit 4 and 9 are successfully identified

%% 5 and 6

digit6Data_1 = digit6Data(:,1:5421); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit5Data,digit6Data_1,20);
```

```matlab
figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 5')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 6')


correct_56 = sum(sort1 < threshold);
accuracy_56 = correct_56 / 5421 % 54.45% of digit 5 and 6 are successfully identified

%% 5 and 7

digit7Data_1 = digit7Data(:,1:5421); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit5Data,digit7Data_1,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 5')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 7')


correct_57 = sum(sort1 < threshold);
accuracy_57 = correct_57 / 5421 % 57.61% of digit 5 and 7 are successfully identified

%% 5 and 8

digit8Data_1 = digit8Data(:,1:5421); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit5Data,digit8Data_1,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 5')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 8')


correct_58 = sum(sort1 < threshold);
accuracy_58 = correct_58 / 5421 % 61.19% of digit 5 and 8 are successfully identified
```

```
%% 5 and 9

digit9Data_1 = digit9Data(:,1:5421); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit5Data,digit9Data_1,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 5')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 9')


correct_59 = sum(sort1 < threshold);
accuracy_59 = correct_59 / 5421 % 53.61% of digit 5 and 9 are successfully identified

%% 6 and 7

digit7Data_1 = digit7Data(:,1:5918); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit6Data,digit7Data_1,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 6')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 7')


correct_67 = sum(sort1 < threshold);
accuracy_67 = correct_67 / 5918 % 62.62% of digit 6 and 7 are successfully identified

%% 6 and 8

digit6Data_1 = digit6Data(:,1:5851); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit6Data_1,digit8Data,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 6')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
```

```
title('Digit 8')


correct_68 = sum(sort1 < threshold);
accuracy_68 = correct_68 / 5851 % 56.62% of digit 6 and 8 are successfully identified

%% 6 and 9

digit9Data_1 = digit9Data(:,1:5918); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit6Data,digit9Data_1,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 6')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 9')


correct_69 = sum(sort1 < threshold);
accuracy_69 = correct_69 / 5918 % 58.82% of digit 6 and 9 are successfully identified

%% 7 and 8

digit7Data_1 = digit7Data(:,1:5851); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit7Data_1,digit8Data,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 7')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 8')


correct_78 = sum(sort1 < threshold);
accuracy_78 = correct_78 / 5851 % 69.29% of digit 7 and 8 are successfully identified


%% 7 and 9

digit7Data_1 = digit7Data(:,1:5949); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit7Data_1,digit9Data,20);

figure(6)
```

```
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 7')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 9')


correct_79 = sum(sort1 < threshold);
accuracy_79 = correct_79 / 5949 % 54.85% of digit 7 and 9 are successfully identified

%% 8 and 9

digit9Data_1 = digit9Data(:,1:5851); %match the size to use SVD

[U,S,V,threshold,w,sort1,sort2] = digit_trainer(digit8Data,digit9Data_1,20);

figure(6)
subplot(1,2,1)
histogram(sort1,100); hold on, plot([threshold threshold], [0 250],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 8')
subplot(1,2,2)
histogram(sort2,30); hold on, plot([threshold threshold], [0 700],'r')
%set(gca,'Xlim',[-3 4],'Ylim',[0 10],'Fontsize',14)
title('Digit 9')


correct_89 = sum(sort1 < threshold);
accuracy_89 = correct_89 / 5851 % 65.80% of digit 8 and 9 are successfully identified

%% LDA Summary

% Digit 2 and 8 are the hardest to seperate (accuracy of 49.87%)
% Digit 0 and 1 are the easiest to seperate (accuract of 93.52%)

%% Decision Trees
tree = fitctree(trainData.',labels,'MaxNumSplits',10,'CrossVal','on');
view(tree.Trained{1},'Mode','graph');
classError = kfoldLoss(tree)
% Decision Trees can successfully identify 54.44% of all 10 digits

%% Decision Trees on easiest duo and hardest duo

hard = [digit2Data_1 digit8Data];
hardlabels = [2*ones(length(digit8Data),1); 8*ones(length(digit8Data),1)];
tree = fitctree(hard.',hardlabels,'MaxNumSplits',2,'CrossVal','on');
view(tree.Trained{1},'Mode','graph');
classError = kfoldLoss(tree);
1- classError
% Decision Trees can successfully identify 84.84% of hardest duo
```

```
digit1Data_1 = digit1Data(:,1:5923);
easy = [digit0Data digit1Data_1];
easylabels = [zeros(length(digit0Data),1); ones(length(digit0Data),1)];
tree = fitctree(easy.',easylabels,'MaxNumSplits',2,'CrossVal','on');
view(tree.Trained{1},'Mode','graph');
classError = kfoldLoss(tree);
1- classError

%% SVM
Mdl = fitcsvm(easy.',easylabels);

easyind0=find(testlabels==0);
easyind1=find(testlabels==1);


testdigit0=[];
testdigit1=[];
for i = 1:length(easyind0)
    testdigit0 = [testdigit0; testData(:,easyind0(i))];
end
test0 = reshape(testdigit0,[784,980]);

digit2Data = digit2Data(:,1:5949); %match the size to use SVD

for i = 1:length(easyind1)
    testdigit1 = [testdigit1; testData(:,easyind1(i))];
end
test1 = reshape(testdigit1,[784,1135]);

easytest = [test0 test1];

test_labels = predict(Mdl,easytest.');
```