AMATH482
HW4 Classifying Digits
Dean Wang

1. Abstract

Given 6000 pieces of data, we need to classify them use Singular Value Decomposition, Linear Discriminant Analysis, Decision Tree, and Support Vector Machines. Finally, we need to compare the accuracy among these methods.

2. Introduction and Overview

We use SVD since it is a good method to analysis the matrix and help us reduce the matrix dimension to save time in the analysis. SVD returns complied matrix X with each image as a row. And we want to find the low rank approximation and the accuracy between different data.

3. Theoretical Background

3.1 SVD

We are given:

$$A = U\Sigma V^* \tag{1}$$

where we know that U and V are unitary matrices that simply lead to rotation and Σ scales an image as prescribed by the singular values. In addition, U and V are rotational matrix, and Σ is stretching matrix. By using this method, we could decrease the redundancy and change the single vector to matrix form.

3.2 LDA

LDA can help us know the distance between inter-class data point. We first define the between-class scatter:

$$S_W = \sum_{j=1}^{2} \sum (x - \mu_j)(x - \mu_j)^T \tag{2}$$

$$w = argmax \frac{w^T S_B w}{w^T S_w w} \tag{3}$$

4. Algorithm Implementation and Development

4.1 SVD

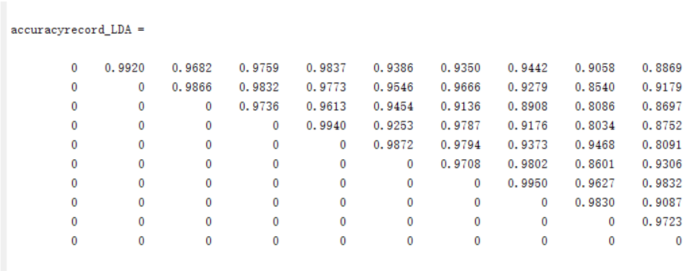
From the main command, I first call the function SVD_analysis to train the. This function analysis returns complied matrix X with each image as a row and original labels variable from train set. 3 plots will be generated: figure1 as a reformed U matrix rows; figure2 singular values spectrum; figure3 projection in V matrix modes. X is the merged matrix of all picture with each one as a column and U S V are the result of SVD analysis. X(:,i)=I reshape each image to a single column and merge them together. Moreover, this function plot of SVD result and plot of rows in U matrix by reshaping them back to 28X28.

4.2 LDA
In this section, I call the function LDA_classifer to choose number. The basic logic is to choose 3 numbers to use LDA classification and would generate the classification result if plotflag is set to 1. Then, I use LDA to identify two number and determined from previous plot of singular value. I separate out data of two choose numbers and make sure the analysis within class variance. The next step is analysis the linear discriminant and find threshold.

4.3 mnist_parse
Then, I load test set and literately run LDA on all combination of two numbers between 0-9 which represents the accuracy result is stored in accuracy matrix. This could run the SVM and decision tree classification result on all pairs of numbers and stored in this matrix if needed.

4.4 SVM & Tree classifier
I choose the number pairs here, the easiest pair is 8&9, and the hardest pair is3&9 according to result from LDA. By call the function tree_SVM_classifier, I input number 1 and 2 choose to classify num1, num2 and receive trained models of SVM method and decision tree method. I reshape each image to a single column and merge them together and use SVM and decision tree classifiers.

Computational Result



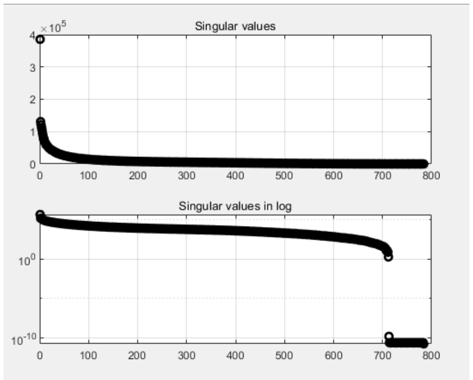
```
accuracyrecord_LDA =

         0    0.9920    0.9682    0.9759    0.9837    0.9386    0.9350    0.9442    0.9058    0.8869
         0         0    0.9866    0.9832    0.9773    0.9546    0.9666    0.9279    0.8540    0.9179
         0         0         0    0.9736    0.9613    0.9454    0.9136    0.8908    0.8086    0.8697
         0         0         0         0    0.9940    0.9253    0.9787    0.9176    0.8034    0.8752
         0         0         0         0         0    0.9872    0.9794    0.9373    0.9468    0.8091
         0         0         0         0         0         0    0.9708    0.9802    0.8601    0.9306
         0         0         0         0         0         0         0    0.9950    0.9627    0.9832
         0         0         0         0         0         0         0         0    0.9830    0.9087
         0         0         0         0         0         0         0         0         0    0.9723
         0         0         0         0         0         0         0         0         0         0
```

This graph represents accuracy between pairs. From top to bottom, from left to right are from 0 to 9.

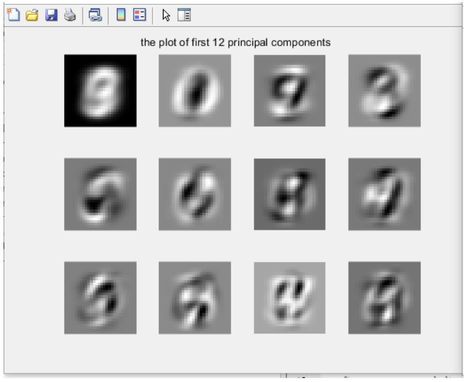
1. Do an SVD analysis of the digit images. You will need to reshape each image into a column vector and each column of your data matrix is a different image.
column of your data matrix is a diffferent image. SVD_analysis()

2. What does the singular value spectrum look like and how many modes are necessary for good image recon- struction? (i.e. what is the rank r of the digit space?)

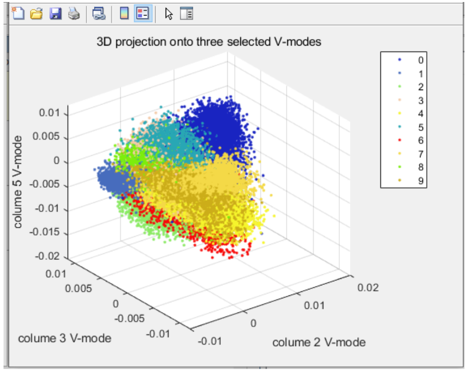


From the second graph, we could see the 712 modle

3. What is the interpretation of the U, Σ, and V matrices?

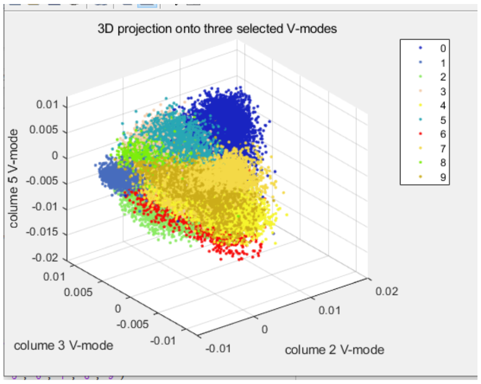


This is figure 1, which is the vector of U

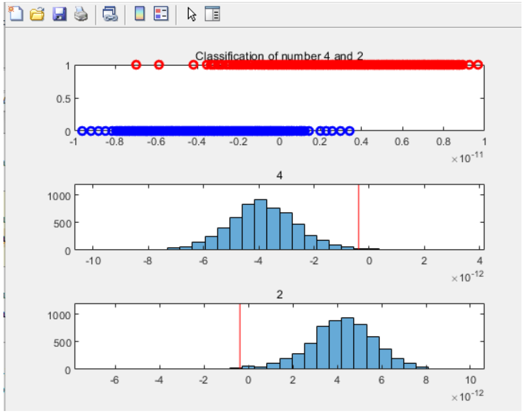


This is figure 3, which is projection of 3 vectors of V

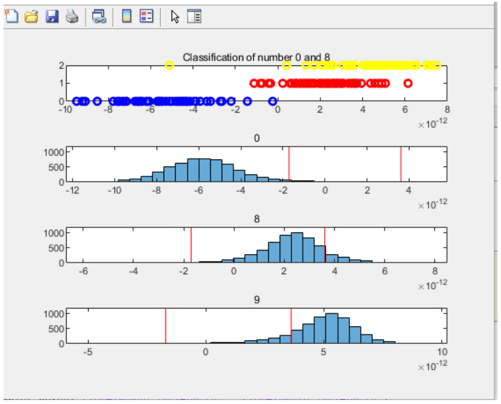
4. On a 3D plot, project onto three selected V-modes (columns) colored by their digit label. For example, columns 2,3, and 5.



• Pick two digits. See if you can build a linear classifier (LDA) that can reasonable identify them.
LDA_classifier(num1,num2,labels,X,plotflag)



• Pick three digits. Try to build a linear classifier to identify these three now.
LDA_classifier3(num1,num2,num3,labels,X,plotflag)



• Which two digits in the data set appear to be the most difficult to separate? Quantify the accuracy of the separation with LDA on the test data.
From accuracy record, we could see each pair's accuracy. We could see that pair 3 and 8 is the hardest which only has 80.34%

- Which two digits in the data set are most easy to separate? Quantify the accuracy of the separation with LDA on the test data.

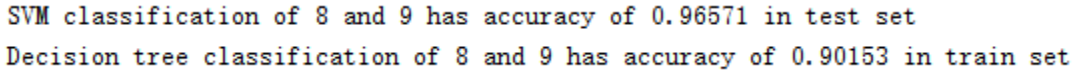
We could see that pair 6 and 7 is the highest which has 99.5%

- SVM (support vector machines) and decision tree classifiers were the state-of-the-art until about 2014. How well do these separate between all ten digits? (see code below to get started).

tree_SVM_classifier(num1,num2)

- Compare the performance between LDA, SVM and decision trees on the hardest and easiest pair of digits to separate (from above).

At the end of the init, the program will output the accuracy.



```
SVM classification of 8 and 9 has accuracy of 0.96571 in test set
Decision tree classification of 8 and 9 has accuracy of 0.90153 in train set
```

Summary and Conclusions

By using decision tree and SVM, we could see the black-box characteristic of machine learning. LAD give us the result of the accuracy between each two vectors. The shape and digits cause the difference in this output. We could see that pair 3 and 8 is the hardest which only has 80.34% and pair 6 and 7 is the highest which has 99.5%

Appendix1

LDA_classifier2(): choose 2 numbers to use LDA classification on would generate the classification result if plot flag is set to 1

LDA_classifier3(): choose 3 numbers to use LDA classification on would generate the classification result if plot flag is set to 1

SVD_analysis(): return complied matrix X with each image as a row and original labels variable from train set

tree_SVM_classifier(): `input number 1 and 2 choose to classify num1,num2 and output trained models of SVM method and decision tree method`

Appendix2

1. HW4init.m

```
%%
%runSVD analysis on the train set
[X,labels]=SVD_analysis();
%%
%LDA_clssifier 2 choosed number
[U2,S2,V2,threshold,w,sortnum1,sortnum2]=LDA_classifier(4,2,labels,X,1);

%%
%LDA_clssifier 3 choosed number
[U3,S3,V3,threshold1,threshold2,w,sortnum1,sortnum2]=LDA_classifier3(0,8,9,la
bels,X,1);

%%
%load test set and iterately run LDA on all combination of two numbers
between 0-9
%the accuracy result is stored in accuracy matrix
clc
[testimages, testlabels] = mnist_parse('t10k-images.idx3-ubyte', 't10k-
labels.idx1-ubyte');
accuracyrecord_LDA=zeros(10,10);
% accuracyrecord_SVM=zeros(10,10);
% accuracyrecord_dtree=zeros(10,10);
%could run the SVM and decision tree classification result on all pairs of
%numbers and stored in these matrix if needed


for num1=0:9
    for num2=num1+1:9

[U2,S2,V2,threshold,w,sortnum1,sortnum2]=LDA_classifier(num1,num2,labels,X,0)
;

ind1=find(testlabels==num1);
ind2=find(testlabels==num2);
testset1=testimages(:,:,ind1);
testset2=testimages(:,:,ind2);


testX=zeros(28*28,length(ind1)+length(ind2));
for i=1:length(ind1)
    I=reshape(testset1(:,:,i),28*28,1);
    testX(:,i)=I;
    %reshape each image to a singul colume and merge them together
end
for i=1:length(ind2)
    I=reshape(testset2(:,:,i),28*28,1);
    testX(:,length(ind1)+i)=I;
    %reshape each image to a singul colume and merge them together
end

truelabel=[testlabels(ind1)' testlabels(ind2)'];
projec2pca=U2'*testX;% PCA projection
pval=w'*projec2pca(1:length(w),:);
```

```
resvect=zeros(size(pval));
for i=1:length(pval)
   if pval(i)>threshold
      resvect(i)=num2;
   else
       resvect(i)=num1;
   end
end
err=abs(truelabel-resvect);
errnum=sum(err);

accuracy=1-errnum/length(pval);

disp([' LDA classification of ',num2str(num1),' and ',num2str(num2),' has
accuracy of ',num2str(accuracy),'in test set',newline])
accuracyrecord_LDA(num1+1,num2+1)=accuracy;




    end
end
accuracyrecord_LDA


%%
%%use SVM and decision tree classifiers
num1=8
num2=9
%choose the number pairs here, the most easy pair is 8&9,
%hardest pair is3&9 accoording to result from LDA

[treeModel,SVMModel]=tree_SVM_classifier(num1,num2);

ind1=find(testlabels==num1);
ind2=find(testlabels==num2);
testset1=testimages(:,:,ind1);
testset2=testimages(:,:,ind2);
testX=zeros(28*28,length(ind1)+length(ind2));
for i=1:length(ind1)
    I=reshape(testset1(:,:,i),28*28,1);
    testX(:,i)=I;
    %reshape each image to a singul colume and merge them together
end
for i=1:length(ind2)
    I=reshape(testset2(:,:,i),28*28,1);
    testX(:,length(ind1)+i)=I;
    %reshape each image to a singul colume and merge them together
end
truelabel=[testlabels(ind1)' testlabels(ind2)'];
SVMresult=predict(SVMModel,testX');
%testX is loaded from test set

err=(truelabel~=SVMresult');
errnum=sum(sum(err));
accuracy=1-errnum/length(SVMresult);
```

```
disp([' LDA classification of ',num2str(num1),' and ',num2str(num2),' has
accuracy of ',num2str(accuracyrecord_LDA(num1+1,num2+1)),' in test set'])
disp([' SVM classification of ',num2str(num1),' and ',num2str(num2),' has
accuracy of ',num2str(accuracy),' in test set'])
classError = kfoldLoss(treeModel);

disp([' Decision tree classification of ',num2str(num1),' and
',num2str(num2),' has accuracy of ',num2str(1-classError),' in train
set',newline])
```

2. SVD_analysis

```
function [X,labels]=SVD_analysis()
%SVD analysis,return complied matrix X with each image as a row and
%original labels variable from train set
%3 plots will be generated
%figure1 as a reformed U matrix rows
%figure2 singular values spectrum
%figure3 projection in V matrix modes


clc
clear
close all
% load('mnist_parse.m')
[images, labels] = mnist_parse('train-images.idx3-ubyte', 'train-labels.idx1-
ubyte');


%%
% X: the merged matrix of all picture with each one as a colume
% U S V:the result of SVD analysis


X=zeros(28*28,60000);
for i=1:60000
    I=reshape(images(:,:,i),28*28,1);
    X(:,i)=I;
    %reshape each image to a singul colume and merge them together
end

[U,S,V] = svd(X,'econ');
%%
%plot of SVD result
%plot of rows in U matrix by reshaping them back to 28X28
figure(1)
for k = 1:12

subplot(3,4,k)
ut1 = reshape(U(:,k),28,28);
ut2 = rescale(ut1);
```

```
imshow(ut2)
end
text(-80,-85,'the plot of first 12 principal components')


% plot the singular values,
figure(2)
subplot(2,1,1)
plot(diag(S),'ko','Linewidth',2)
title('Singular values')
grid on
subplot(2,1,2)
semilogy(diag(S),'ko','Linewidth',2)
title('Singular values in log')
grid on

%%
%plot of V-modes
% v1 v2 v3 : the selected colums of V
% colors: randomly generated 10 colors

figure(3)
v1=V(:,2);
v2=V(:,3);
v3=V(:,5);
colors=rand(10,3);
for i=0:9
index=find(labels==i);
plot3(v1(index),v2(index),v3(index),'.','Color',colors(i+1,:))
hold on
end
xlabel('colume 2 V-mode')
ylabel('colume 3 V-mode')
zlabel('colume 5 V-mode')
title('3D projection onto three selected V-modes')
grid on
legend('0','1','2','3','4','5','6','7','8','9')

end
```

3. LDA classifer3

```
function
[U3,S3,V3,threshold1,threshold2,w,sortnum1,sortnum2]=LDA_classifier3(num1,num
2,num3,labels,X,plotflag)
%choose 3 numbers to use LDA classification on
%would generate the classification result if plotflag is set to 1

%%
disp(['start LDA classification 3 numbers',num2str(num1),'  ,
',num2str(num2) ,' and ',num2str(num3)])
%use LDA to identify two number
feature=712;
%determined from previous plot of singular value
```

```
index1=find(labels==num1);
index2=find(labels==num2);
index3=find(labels==num3);
%seperate out data of two choosed numbers

X3=[X(:,index1) X(:,index2) X(:,index3)];
[U3,S3,V3]=svd(X3,'econ');
n1=length(index1);
n2=length(index2);
n3=length(index3);
Proj2PC=S3*V3';

n1m=Proj2PC(1:feature,1:n1);
n2m=Proj2PC(1:feature,n1+1:n1+n2);
n3m=Proj2PC(1:feature,n1+n2+1:n1+n2+n3);

mean1=mean(n1m,2);
mean2=mean(n2m,2);
mean3=mean(n3m,2);

Sw=0;%within class variance
for k=1:n1

    Sw=Sw+(n1m(:,k)-mean1)*(n1m(:,k)-mean1)';
end
for k=1:n2
    Sw=Sw+(n2m(:,k)-mean2)*(n2m(:,k)-mean2)';
end
for k=1:n3
    Sw=Sw+(n3m(:,k)-mean3)*(n3m(:,k)-mean3)';
end

Sb=(mean1-mean2)*(mean1-mean2)'+(mean3-mean2)*(mean3-mean2)'+(mean1-
mean3)*(mean1-mean3)';
%between class variance
%%
% linear discriminant analysis
[Vlda,D]=eig(Sb,Sw);
[lambda, ind] = max(abs(diag(D)));
w = Vlda(:,ind);
w = w/norm(w,2);
vn1=w'*n1m;
vn2=w'*n2m;
vn3=w'*n3m;
pvn1=vn1(1:100:n1);
pvn2=vn2(1:100:n2);
pvn3=vn3(1:100:n3);
pn1=length(pvn1);
pn2=length(pvn2);
pn3=length(pvn3);

figure()
subplot(4,1,1)
plot(pvn1,zeros(pn1),'bo','Linewidth',2)
hold on
plot(pvn2,ones(pn2),'ro','Linewidth',2)
hold on
```

```matlab
plot(pvn3,ones(pn3,1)*2,'yo','Linewidth',2)
title(['Classification of number ',num2str(num1),' and ',num2str(num2)])
%
% if plotflag
%     %only plot when plotflag is not 0
% figure()
% subplot(4,1,1)
% plot(vn1,zeros(n1),'bo','Linewidth',2)
% hold on
% plot(vn2,ones(n2),'ro','Linewidth',2)
% hold on
% plot(vn3,ones(n3,1)*2,'yo','Linewidth',2)
% title(['Classification of number ',num2str(num1),' and ',num2str(num2)])
% end
%%
%find thresh hold
sortnum1=sort(vn1);
sortnum2=sort(vn2);
sortnum3=sort(vn3);
vmeans=[mean(vn1) mean(vn2) mean(vn3)];
[meanmax,ind1]=max(vmeans);
[meanmin,ind2]=min(vmeans);
for i=1:3
   if i~=ind1&& i~=ind2
      meanmid=vmeans(i);
   end
end
threshold2=(meanmax+meanmid)/2;
threshold1=(meanmin+meanmid)/2;


if plotflag

disp('plotting the result of LDA, this could take a while')
subplot(4,1,2)
histogram(sortnum1,30); hold on,
plot([threshold1 threshold1], [0 1200],'r')
hold on,
plot([threshold2 threshold2], [0 1200],'r')
% set(gca,'Xlim',[-10 10],'Ylim',[0 1200],'Fontsize',14)
title(num2str(num1))
subplot(4,1,3)
histogram(sortnum2,30);
hold on,
plot([threshold1 threshold1], [0 1200],'r')
hold on,
plot([threshold2 threshold2], [0 1200],'r')
% set(gca,'Xlim',[-10 10],'Ylim',[0 1200],'Fontsize',14)
title(num2str(num2))

subplot(4,1,4)
histogram(sortnum3,30);
hold on,
plot([threshold1 threshold1], [0 1200],'r')
hold on,
plot([threshold2 threshold2], [0 1200],'r')
% set(gca,'Xlim',[-10 10],'Ylim',[0 1200],'Fontsize',14)
```

```
title(num2str(num3))
end

disp(['finish analyzing 3 numbers',num2str(num1),',  ',num2str(num2) ,' and
',num2str(num3)])
end
```

4. LDA classifer2

```
function
[U2,S2,V2,threshold,w,sortnum1,sortnum2]=LDA_classifier(num1,num2,labels,X,pl
otflag)
%choose 2 numbers to use LDA classification on
%would generate the classification result if plotflag is set to 1

%%
disp(['start LDA classification ',num2str(num1),' and ',num2str(num2)])

%use LDA to identify two number
feature=712;
%determined from previous plot of singular value


index1=find(labels==num1);
index2=find(labels==num2);
%seperate out data of two choosed numbers

X2=[X(:,index1) X(:,index2)];
[U2,S2,V2]=svd(X2,'econ');
n1=length(index1);
n2=length(index2);
Proj2PC=S2*V2';

n1m=Proj2PC(1:feature,1:n1);
n2m=Proj2PC(1:feature,n1+1:n1+n2);

mean1=mean(n1m,2);
mean2=mean(n2m,2);

Sw=0;%within class variance
for k=1:n1

    Sw=Sw+(n1m(:,k)-mean1)*(n1m(:,k)-mean1)';
end
for k=1:n2
    Sw=Sw+(n2m(:,k)-mean2)*(n2m(:,k)-mean2)';
end

Sb=(mean1-mean2)*(mean1-mean2)';%between class variance
%%
% linear discriminant analysis
[Vlda,D]=eig(Sb,Sw);
[lambda, ind] = max(abs(diag(D)));
w = Vlda(:,ind);
```

```
w = w/norm(w,2);
vn1=w'*n1m;
vn2=w'*n2m;

if mean(vn1)>mean(vn2)
   w=-w;
   vn1=-vn1;
   vn2=-vn2;
end
%make sure number 1 is always lower

if plotflag
    %only plot when plotflag is not 0
figure()
subplot(3,1,1)
plot(vn1,zeros(n1),'bo','Linewidth',2)
hold on
plot(vn2,ones(n2),'ro','Linewidth',2)
title(['Classification of number ',num2str(num1),' and ',num2str(num2)])
end
%%
%find thresh hold
sortnum1=sort(vn1);
sortnum2=sort(vn2);

t1=n1;
t2=1;
while sortnum1(t1)>sortnum2(t2)
   t1=t1-1;
   t2=t2+1;
end
threshold= (sortnum1(t1) + sortnum2(t2))/2;

if plotflag

disp('plotting the result of LDA, this could take a while')
subplot(3,1,2)
histogram(sortnum1,30); hold on,
plot([threshold threshold], [0 1200],'r')
% set(gca,'Xlim',[-10 10],'Ylim',[0 1200],'Fontsize',14)
title(num2str(num1))
subplot(3,1,3)
histogram(sortnum2,30); hold on,
plot([threshold threshold], [0 1200],'r')
% set(gca,'Xlim',[-10 10],'Ylim',[0 1200],'Fontsize',14)
title(num2str(num2))

end

disp(['finish analyzing ',num2str(num1),' and ',num2str(num2)])
end
```

5. mnist_parse.m

```
function [images, labels] = mnist_parse(path_to_digits, path_to_labels)

% The function is curtesy of stackoverflow user rayryeng from Sept. 20,
% 2016. Link: https://stackoverflow.com/questions/39580926/how-do-i-load-in-
the-mnist-digits-and-label-data-in-matlab

% Open files
fid1 = fopen(path_to_digits, 'r');

% The labels file
fid2 = fopen(path_to_labels, 'r');

% Read in magic numbers for both files
A = fread(fid1, 1, 'uint32');
magicNumber1 = swapbytes(uint32(A)); % Should be 2051
% fprintf('Magic Number - Images: %d\n', magicNumber1);

A = fread(fid2, 1, 'uint32');
magicNumber2 = swapbytes(uint32(A)); % Should be 2049
% fprintf('Magic Number - Labels: %d\n', magicNumber2);

% Read in total number of images
% Ensure that this number matches with the labels file
A = fread(fid1, 1, 'uint32');
totalImages = swapbytes(uint32(A));
A = fread(fid2, 1, 'uint32');
if totalImages ~= swapbytes(uint32(A))
    error('Total number of images read from images and labels files are not
the same');
end
% fprintf('Total number of images: %d\n', totalImages);

% Read in number of rows
A = fread(fid1, 1, 'uint32');
numRows = swapbytes(uint32(A));

% Read in number of columns
A = fread(fid1, 1, 'uint32');
numCols = swapbytes(uint32(A));

% fprintf('Dimensions of each digit: %d x %d\n', numRows, numCols);

% For each image, store into an individual slice
images = zeros(numRows, numCols, totalImages, 'uint8');
for k = 1 : totalImages
    % Read in numRows*numCols pixels at a time
    A = fread(fid1, numRows*numCols, 'uint8');

    % Reshape so that it becomes a matrix
    % We are actually reading this in column major format
    % so we need to transpose this at the end
    images(:,:,k) = reshape(uint8(A), numCols, numRows).';
end

% Read in the labels
labels = fread(fid2, totalImages, 'uint8');

% Close the files
fclose(fid1);
fclose(fid2);

end
```

6. tree_SVM_classifier

```
function [treeModel,SVMModel]=tree_SVM_classifier(num1,num2)
% input number 1 and 2 choosed to classify
% num1,num2: number 1 and 2 choosed to classify
% output:
% trained models of SVM method and decision treemethod



[images, labels] = mnist_parse('train-images.idx3-ubyte', 'train-labels.idx1-ubyte');
ind1=find(labels==num1);
ind2=find(labels==num2);
trainset1=images(:,:,ind1);
trainset2=images(:,:,ind2);
trainX=zeros(28*28,length(ind1)+length(ind2));
for i=1:length(ind1)
    I=reshape(trainset1(:,:,i),28*28,1);
    trainX(:,i)=I;
    %reshape each image to a singul colume and merge them together
end
for i=1:length(ind2)
    I=reshape(trainset2(:,:,i),28*28,1);
    trainX(:,length(ind1)+i)=I;
    %reshape each image to a singul colume and merge them together
end
truelabel=[labels(ind1)' labels(ind2)'];

%use SVM and decision tree classifiers
disp(['start SVM classification ',num2str(num1),' and ',num2str(num2),'this could take a while'])
SVMModel = fitcsvm(trainX',truelabel');
disp(['start decision tree classification ',num2str(num1),' and ',num2str(num2)])
treeModel = fitctree(trainX',truelabel','MaxNumSplits',3,'CrossVal','on');
% view(tree.Trained{1},'Mode','graph');

end
```