AMATH 482 Homework 4

Ariel Luo

Mar 8, 2020

Abstract

We are going to analyze the given data set with both training and testing set and labels to try and identify the digit images by performing Singular Value Decomposition and Linear Discriminant Analysis.

1 Introduction and Overview

In this homework, we are working with 4 data sets which are training sets, training labels, test sets and test labels of images of digits. The labels are the digits showing in the images. We are trying to separate and reconstruct the images by using Linear Discriminant Analysis. We will first start with two digits, then we will move on to identify three. We will then identify the two easiest and hardest digits to separate by looking at the accuracy of the separation with LDA on the test data.

2 Theoretical Background

In order to train our computer to become able to tell the differences between objects in pictures, we are going to use the Linear Discriminant Analysis. As we learned from our lecture, LDA will project our data sets onto new bases. Our final goal is to find a good projection on a subspace that maximizes the distances between different data. For a 2-class LDA, we need a projection w such that:

$$w = argmax \frac{w^T S_B w}{w^T S_W w} \tag{1}$$

where

$$S_B = (\mu_2 - \mu_1)(\mu_2 - \mu_1)^T \tag{2}$$

$$S_W = \sum_{j=1}^{2} \sum_{x} (x - \mu_j)(x - \mu_j)^T \tag{3}$$

These are the variance of the data sets and the variance of the difference in means. We are going to use

$$S_B w = \lambda S_W w \tag{4}$$

to find the maximum eigenvalue. The eigenvector of that value will the projection basis. We will apply LDA after the SVD decomposition.

3 Algorithm Implementation and Development

The algorithm can be summarized and separated into two main parts:

For the first part, we find the coordinates(indices) of the moving mass by looking at the white dot located on the mass. For the second part, we apply PCA to plot the data.

Algorithm 1: Finding the projection

Find data with labels(0,1) or (0,1,2)
calculate mean
for $f = 1$:number of frames **do**
  Calculate sw
end for
Use for loop twice/three times to calculate sw
Calculate sb
Calculate V and D from eig
Calculate w(projection), then v0, v1, and v2 from w

Algorithm 2: Projecting on PCA SVD

Load data
Normalize data set
Calculate U, S and V by using SVD
S*V'

4 Computational Results

1. Before SVD, I reshape the data set from 28x28x60000 to 784x60000.
2. Since the rank of the digit space is 10, 10 modes are necessary for good image reconstruction
3. U's columns corresponds to the features, S corresponds to the strength of the projection, and V corresponds to the projected data of digits
4. See 1

I chose to build a classifier on digits 0 and 1. When I check to projections, we can see from 2 that there are still parts of 0 and 1 that are overlapping. Therefore, we can't pick a perfect threshold that separates the data perfectly. 3 is plot of the data and the threshold I picked. As we can see in the graph, there are only a small parts of the data that are wrong which means the threshold works. After the model was trained, I used the test sets to see what I could pull out from the data. As we can see from 4, we can almost tell that upper left image is a 1 but it's clear that the upper right image is a 1. The zeroes are also pretty clear to see.

I chose to build classifier on digits 0, 1 and 2 after that. I did almost the same as what I did for 2-class LDA. From 5, we can see that a lot of 1 values are overlapped with 0 and 2. that I calculated mean for all 3 data sets instead of 2. To find the threshold, I first calculated the threshold for 0 and 1, then the threshold for 1. From 7, the threshold for digit 2 seems a little off comparing to 0 and 1. But the final result **??** surprisingly turned out not too bad.

5 Summary and Conclusions

In conclusion, I was able identify 0, 1, and 2 from the test data set by building LDA on our training data set as we can see from the result. I was not able to define the most difficult and the easiest digits in the data set in MATLAB. Overall, LDA is a very effective tool to classify objects/animals from data. It's also sufficient since we are using a certain number of features instead of the whole data set.

Algorithm 3: Calculating threshold

sort v0, v1, v2
Find the threshold where we can have the same number of mistakes for all data set
then

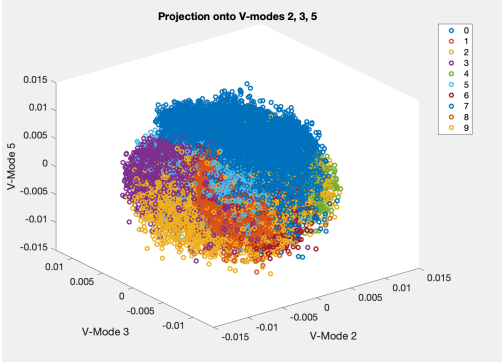

Figure 1: Projections on to V modes 2,3,and 5

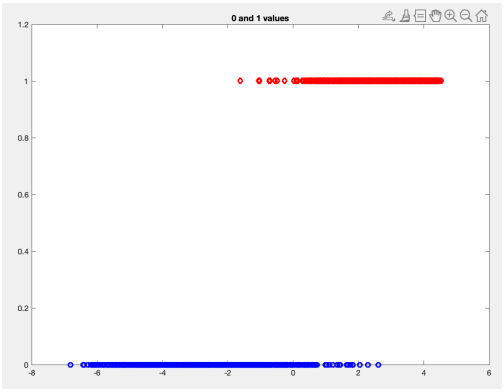

Figure 2: Values of 0 and 1

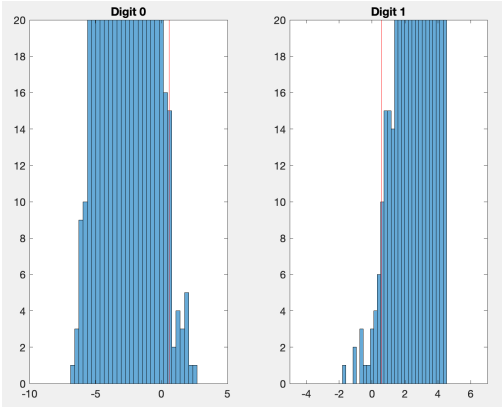

Figure 3: 0 and 1 values with threshold

Figure 4: Reconstructing 1 and 0 from test set

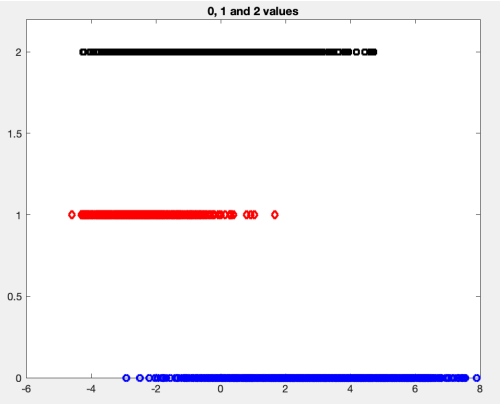

Figure 5: Values of 0, 1, 2

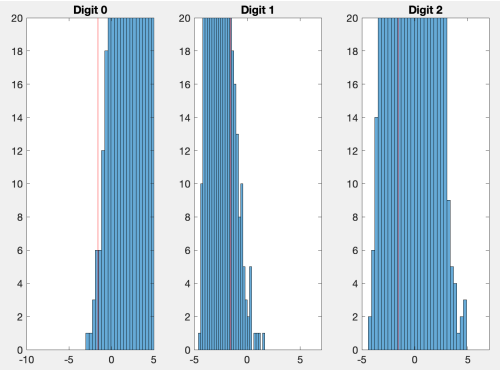

Figure 6: 0, 1, 2 values with threshold

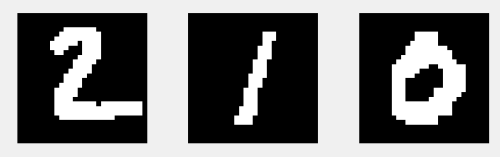

Figure 7: Reconstructing 2, 1, 0 from test set

Appendix A MATLAB Functions

Add your important MATLAB functions here with a brief implementation explanation.

- `sz = size(A)` returns a row vector whose elements are the lengths of the corresponding dimensions of A. For example, if A is a 3-by-4 matrix, then size(A) returns the vector [3 4].
- `[S = sum(A)` returns the sum of the elements of A along the first array dimension whose size does not equal 1.
- `M = mean(A)` returns the mean of the elements of A along the first array dimension whose size does not equal 1.
- `B = repmat(A,n)` returns an array containing n copies of A in the row and column dimensions. The size of B is size(A)*n when A is a matrix.
- `[U,S,V] = svd(A)` performs a singular value decomposition of matrix A, such that A = U*S*V'.
- `plot(Y)` creates a 2-D line plot of the data in Y versus the index of each value.

Appendix B MATLAB Code

```
clear all; close all;
[images, labels] = mnist_parse('train-images-idx3-ubyte', 'train-labels-idx1-ubyte');
images = double(reshape(images,784,60000));
[m,n]=size(images);
mn=mean(images,2);
im=images-repmat(mn,1,n);
[U,S,V]=svd(im/sqrt(n-1),'econ');
%% Projection onto 3 V-modes
for label=0:9
    label_indices = find(labels == label);
    plot3(V(label_indices, 2), V(label_indices, 3), V(label_indices, 5),...
        'o', 'DisplayName', sprintf('%i',label), 'Linewidth', 2)
    hold on
end
xlabel('V-Mode 2'), ylabel('V-Mode 3'), zlabel('V-Mode 5')
title('Projection onto V-modes 2, 3, 5')
legend
set(gca,'Fontsize', 14)
%%
z1=S*V';
ind=find(labels==0);
zero=z1(:,ind);
ind=find(labels==1);
one=z1(:,ind);
ind=find(labels==2);
two=z1(:,ind);

n0=size(zero,2);
n1=size(one,2);


lab0=zero(1:10,1:n0);
lab1=one(1:10,1:n0);
```

```
lab2=two(1:10,1:n0);
m1=mean(lab1,2);
m0=mean(lab0,2);
m2=mean(lab2,2);
sw=0;
for k = 1:n0
    sw=sw+(lab0(:,k)-m0)*(lab0(:,k)-m0)';
end
for k = 1:n0
    sw=sw+(lab1(:,k)-m1)*(lab1(:,k)-m1)';
end
sb=(m0-m1)*(m0-m1)';
[V2,D]=eig(sb,sw);
[lambda, ind] = max(abs(diag(D)));
w = V2(:,ind);
w = w/norm(w,2);
v0=w'*lab0;
v1=w'*lab1;
if mean(v0)>mean(v1)
    w = -w;
    v0 = -v0;
    v1 = -v1;
end

%%
figure(2)
plot(v0,zeros(size(v0)),'ob','Linewidth',2)
hold on
plot(v1,ones(size(v1)),'dr','Linewidth',2)
title('0 and 1 values')
ylim([0 1.2])

%%
sort0 = sort(v0);
sort1 = sort(v1);
t0=length(sort0);
t1=1;
while sort0(t0) > sort1(t1)
    t0=t0-1;
    t1=t1+1;
end
threshold=(sort0(t0)+sort1(t1))/2;

%%
figure(3)
subplot(1,2,1)
histogram(sort0,30); hold on, plot([threshold threshold], [0 20],'r')
set(gca,'Xlim',[-10 5],'Ylim',[0 20],'Fontsize',14)
title('Digit 0')
subplot(1,2,2)
histogram(sort1,30); hold on, plot([threshold threshold], [0 20],'r')
set(gca,'Xlim',[-5 7],'Ylim',[0 20],'Fontsize',14)
title('Digit 1')
```

```
%%
[testim, testlb] = mnist_parse('t10k-images-idx3-ubyte', 't10k-labels-idx1-ubyte');
testim = double(reshape(testim,784,10000));

%%
ppro=U(:,1:10)'*testim;
%%
ind=find((testlb==1)|(testlb==0));
test=testim(:,ind);
testd=ppro(:,ind);
testl=testlb(ind);
%%
pval=w'*testd;
resVec=(pval>threshold);
err = abs(resVec - testl);
errNum = sum(err);
sucRate = 1 - errNum/2115;
%%
k = 1;
figure(4)
for j = 1:2115
    if resVec(j) ~= testl(j)
        S = reshape(test(:,j),28,28);
        subplot(1,2,k)
        imshow(S)
        k = k+1;
    end
end
%%
sw3=sw;
for k = 1:n0
    sw3=sw3+(lab2(:,k)-m2)*(lab2(:,k)-m2)';
end
m12=mean([lab0 lab1 lab2],2);
sb3=(m0-m12)*(m0-m12)'+(m1-m12)*(m1-m12)'+(m2-m12)*(m2-m12)';
[V3, D3] = eig(sb3,sw3);
[lambda, ind] = max(abs(diag(D3)));
w3 = V3(:,ind);
w3 = w3/norm(w3,2);
v0_3 = w3'*lab0;
v1_3 = w3'*lab1;
v2_3 = w3'*lab2;

%%
figure(5)
plot(v0_3,zeros(size(v0_3)),'ob','Linewidth',2)
hold on
plot(v1_3,ones(size(v1_3)),'dr','Linewidth',2)
hold on
plot(v2_3,2*ones(size(v2_3)),'sk','Linewidth',2)
title('0, 1 and 2 values')
ylim([0 2.2])

%%
```

```
sort0_3 = sort(v0_3);
sort1_3 = sort(v1_3);
sort2_3 = sort(v2_3);
t0_3=length(sort0_3);
t1_3=1;
while sort0_3(t0_3) > sort1_3(t1_3)
    t0_3=t0_3-1;
    t1_3=t1_3+1;
end
threshold1=(sort0_3(t0_3)+sort1_3(t1_3))/2;
t1_3=length(sort1_3);
t2_3=1;
while sort1_3(t1_3) > sort2_3(t2_3)
    t1_3=t1_3-1;
    t2_3=t2_3+1;
end
threshold2=(sort1_3(t1_3)+sort2_3(t2_3))/2;
threshold_3=(threshold1+threshold2)/2;
%%
figure(6)
subplot(1,3,1)
histogram(sort0_3,30); hold on, plot([threshold_3 threshold_3], [0 20],'r')
set(gca,'Xlim',[-10 5],'Ylim',[0 20],'Fontsize',14)
title('Digit 0')
subplot(1,3,2)
histogram(sort1_3,30); hold on, plot([threshold_3 threshold_3], [0 20],'r')
set(gca,'Xlim',[-5 7],'Ylim',[0 20],'Fontsize',14)
title('Digit 1')
subplot(1,3,3)
histogram(sort2_3,30); hold on, plot([threshold_3 threshold_3], [0 20],'r')
set(gca,'Xlim',[-5 7],'Ylim',[0 20],'Fontsize',14)
title('Digit 2')
%%
ppro_3=U(:,1:10)'*testim;
%%
ind=find((testlb==1)|(testlb==0)|(testlb==2));
test3=testim(:,ind);
testd3=ppro_3(:,ind);
testl3=testlb(ind);
pval=w3'*testd3;
resVec3=(pval>threshold_3);
%%
k = 1;
figure(6)
for j = 1:3147
    if resVec3(j) ~= testl3(j)
        S = reshape(test3(:,j),28,28);
        subplot(1,3,k)
        imshow(S)
        k = k+1;
    end
end
```

Code for HW4