AMATH 482 Homework 4

Tommy Dong

March 10, 2021

Abstract

This project will implement fundamental machine learning technique. In particular, one will implement Linear Discrimination Analysis (LDA) to distinguish between different objects. After utilize LDA to distinguish between two and three objects, Decision Tree and Support Vector Machine (SVM) will also be used for the same prediction and compare the fidelity of different machine laerning methods.

1 Introduction and Overview

One is provided with the MNIST data set, which contains thousands of handwritten digits images. The goal is to develop a LDA algorithm that can distinguish the differences between different digits. In the meantime, we are also going to do digits prediction with Decision Tree and Support Vector Machine (SVM), and then compare the accuracy of digits prediction to see which machine learning model are the best.

1.1 Data Description

The data used in this project are from MNIST, which are consists of 60000 images of handwritten digits for training the model, and 10000 images of handwritten digits images for testing the fidelity of model. All these images have the same size (28×28).

1.2 Goal

1.2.1 Principal Component Analysis

Initially, we want to do SVD decomposition of the training data. The reason to implement this procedure is to determine how many modes are necessary for good image reconstruction.

1.2.2 Develop LDA algorithm

With rank r determined from the step above, now we want to apply a LDA algorithm that can classify different handwritten digits. We will develop a 2-digits LDA and a 3-digits LDA and compare the accuracy of both models. Also, by implementing 2-digits LDA for all possible pairs of digits, we will determine the pair which has the highest accuracy and the pair who are the worst among predictions.

1.2.3 Decision Tree and SVM

After we find the highest accurate and lowest accurate digit pairs from 2-digits LDA, we put them into Decision Tree and SVM model to see how the predictions go in order to judge which model best predicts the handwritten digit images.

2 Theoretical Background

2.1 Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) decompose some matrix A into unitary matrices U and V and a diagonal matrix Σ.

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

Σ is the singular value matrix A, U are the left singular vectors of A and V is the right singular vectors of A.

2.2 Linear Discrimination Analysis (LDA)

Linear Discrimination Analysis find a suitable projection that maximizes the distance between the inter-class data while minimizing the intra-class data. In order to achieve this agenda, one need to find the right subspace to project onto.
The matrix below is the between-class scatter matrix, where μ are the means for each of the groups for each feature. This measures the variance between the groups.

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

Then we define within-class scatter matrix. This is a measure of the variance within each group.

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

We want to get vector w such that

$$w = argmax \frac{w^T S_B w}{w^T S_w w} \tag{4}$$

The equation above is hard to solve, but w turns out to be the eigenvector corresponding to the largest eigenvalue of the problem

$$S_B w = \lambda S_w w \tag{5}$$

We can easily access this with MATLAB command.

As for LDA for more than 3 dimensions, we simply need to alter the two scatter matrix into the form below:

$$S_B = \sum_{j}^{N} (\mu_j - \mu)(\mu_j - \mu)^T \tag{6}$$

$$S_w = \sum_{j=1}^{N} \sum_{x} (x - \mu_j)(x - \mu_j)^T \tag{7}$$

And then apply the same procedure as two dimension LDA.

3 Algorithm Implementation and Development

1. Principal Component Analysis (PCA) Principal Component Analysis will decompose study object into three parts. In this project in particular, U contains the entire principal components of the data, S stores the energy of the data, which represent how much information is sealed in each principal component, and V are the modes we project onto.

   In this project, we first transform each image from training set into a column vector, and then put all these columns in a matrix. Then we performed SVD with this matrix to retrieve critical information. From U and S, we want to determine how many modes are necessary for LDA to develop effective algorithms. As for V, we want to see how good the project on certain V-modes is.

2. Linear Discrimination Analysis (LDA) Linear Discrimination Analysis aimed to distinct two different objects by condense data inside each group in the meantime maximize the margin between different groups. With the help of SVD, we can project objects on to a linear line which stratifies the condition that minimize the gap between data inside and maximize the margin for different group outside.

   For this project, we pick two digits and perform SVD. Then we find the line that both maximize the distance between two digits and compress data points in each digit group with theoretical supports from equations listed in Theoretical Background. Then we calculate a threshold value for these two digits, and take training and testing data to verify the efficiency of LDA. Similarly, we also develop a LDA algorithm for 3 digits, which basically just need us to project these three dimension on to a line and discover two thresholds instead of one.

4 Computational Results

4.1 The first 10 principal components

From both Figure 1 and Figure 2, first 10 modes clearly extract some important features of digits, and the energy also suggested that the first 10 contained most of the information necessary to tell the difference between digits. So we pick 10 feature for further LDA algorithms.

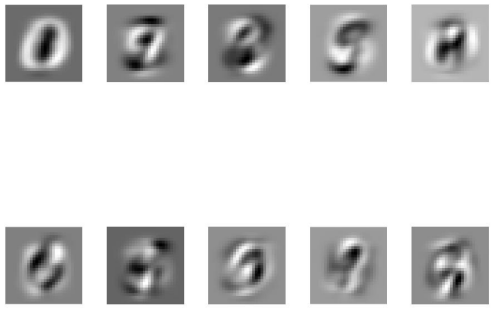

Figure 1: First 10 principal components

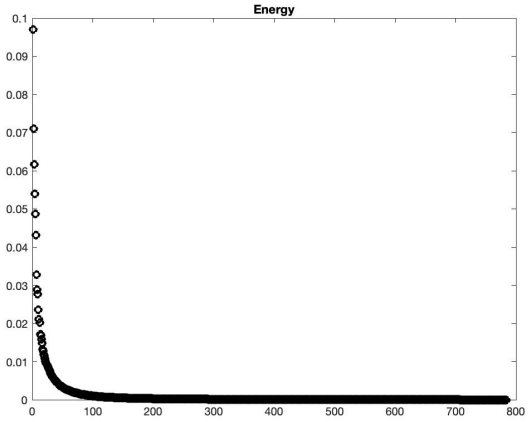

Figure 2: Energy Plot

4.2 Projection onto V-modes 2,3,5

For V-modes projection, we pick the 2,3,5 columns from V, and draw a 3D projection plot. The result are fair for all digits are clustered in some place, though there exist some overlap between different digits, we can still easily distinct different clusters by eyeballing.

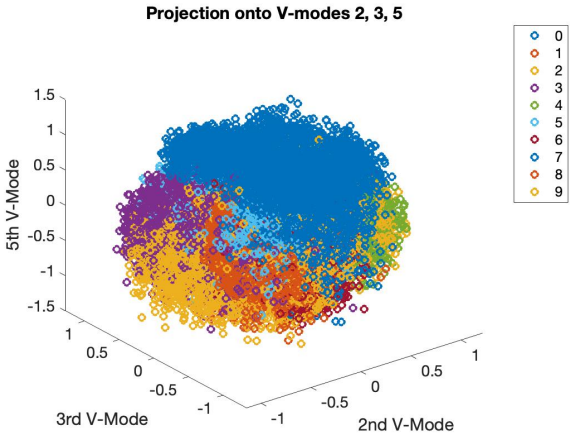

Figure 3: Projection on V-modes 2,3,5

4.3 Accuracy Comparison for 2 digits

1. 2 digit LDA both Train and Test's highest pair: 0,1 accuracy rate: 0.9974(train), 0.9976(test)
   Decision Tree on 0,1 accuracy rate: 0.9989(train), 0.9853(test)
   SVM on 0,1 accuracy rate: 1(train), 0.9915(test)

2. 2 digit LDA both Train and Test's lowest pair: 3,5 accuracy rate: 0.0985(train), 0.8198(test)
   Decision Tree on 3,5 accuracy rate: 0.9898(train), 0.4942(test)
   SVM on 3,5 accuracy rate: 0.8565(train), 0.8423(test)

(For the entire accuracy of all pairs, the table image is attached in appendix A).

4.4 Accuracy for 3 digit LDA

I picked 1,2,4 as my digits. The accuracy turned out to be 0.6453.

5 Summary and Conclusions

From the computational result on prediction accuracy, we can see that LDA is good at 2 digits distinction rather than 3 digits. The reason might be it is harder to find a perfect line for three digits to nicely distinguished between each other, and there might exists some overlap between digits, but that problem seldom exists between 2 digits LDA when there exists clear distinction between the 2 digits.

Compare to other methods, LDA perform fair. It did good on easily distinct pairs, but Decision Tree and SVM perform better. On the other hand, LDA did a excellent job on distinguish the most difficult pairs. The Decision Tree outrun LDA at training, but its testing is terrible, and my personal guess the reason why Decision Tree has only 0.49 accuracy for testing is because of the overfitting problem. The performance of Decision Tree are high depend on how many branches it develop, but sometimes too much branched created by training may lead to overfitting, causing ridiculous outcome for testing. SVM also defeated at the most difficult pair, but returned with a relatively decent result around 0.85 for both training and testing.

Appendix A Full Accuracy Table for 2 digits LDA

	1	2	3	4	5	6	7	8	9	10
1	0	0.9974	0.9769	0.9817	0.9925	0.9610	0.9852	0.9916	0.9816	0.9850
2	0	0	0.9787	0.9781	0.9921	0.9823	0.9924	0.9823	0.9687	0.9893
3	0	0	0	0.9463	0.9678	0.9638	0.9557	0.9679	0.9448	0.9712
4	0	0	0	0	0.9865	0.9085	0.9843	0.9661	0.9357	0.9562
5	0	0	0	0	0	0.9702	0.9802	0.9725	0.9786	0.9221
6	0	0	0	0	0	0	0.9645	0.9802	0.9330	0.9670
7	0	0	0	0	0	0	0	0.9947	0.9794	0.9941
8	0	0	0	0	0	0	0	0	0.9749	0.9221
9	0	0	0	0	0	0	0	0	0	0.9514
10	0	0	0	0	0	0	0	0	0	0

Figure 4: LDA 2 digit training set accuracy

	1	2	3	4	5	6	7	8	9	10
1	0	0.9976	0.9856	0.9839	0.9903	0.9621	0.9840	0.9890	0.9693	0.9859
2	0	0	0.8551	0.8214	0.9920	0.9753	0.9919	0.9325	0.4618	0.9771
3	0	0	0	0.9496	0.9429	0.9298	0.9337	0.9248	0.9158	0.9735
4	0	0	0	0	0.9789	0.8191	0.9741	0.9141	0.8881	0.9460
5	0	0	0	0	0	0.9621	0.9778	0.9716	0.8860	0.8769
6	0	0	0	0	0	0	0.9503	0.9839	0.6136	0.9474
7	0	0	0	0	0	0	0	0.9909	0.9493	0.9914
8	0	0	0	0	0	0	0	0	0.8232	0.7879
9	0	0	0	0	0	0	0	0	0	0.8311
10	0	0	0	0	0	0	0	0	0	0

Figure 5: LDA 2 digit testing set accuracy

Appendix B MATLAB Code

Add your MATLAB code here. This section will not be included in your page limit of six pages.

```
%% HW4
% Clean workspace
clear all; close all; clc

%% Import Data
[images, labels] = mnist_parse('train-images-idx3-ubyte', 'train-labels-idx1-ubyte');
[test_images, test_labels] = mnist_parse('t10k-images-idx3-ubyte', 't10k-labels-idx1-ubyte');

M = zeros(784,60000);
for i=1:60000
   M(:,i)=reshape(images(:,:,i),[784,1]);
end
Mtest = zeros(784,10000);
for i=1:10000
   Mtest(:,i)=reshape(test_images(:,:,i),[784,1]);
end

[m,n]=size(M);
mn=mean(M,2);
M=M-repmat(mn,1,n);
[U,S,V] = svd(M, 'econ');

sigma = diag(S);
energy = zeros(1,length(sigma));
for j = 1:length(sigma)
    energy(j) = sigma(j)^2/sum(sigma.^2);
end
figure(1)
for j=1:10
   subplot(2,5,j)
   ut1 = reshape(U(:,j),28,28);
   ut2 = rescale(ut1);
   imshow(ut2)
end

% Singular Value Spectrum
figure(2)
plot(energy,'ko','Linewidth',2)
title('Energy')

% Projection onto 3 V-modes
figure(3)
for label=0:9
    label_indices = find(labels == label);
    plot3(V(label_indices, 4)*100, V(label_indices, 5)*100, V(label_indices, 6)*100,...
        'o', 'DisplayName', sprintf('%i',label), 'Linewidth', 2)
    hold on
end
xlabel('2nd V-Mode'), ylabel('3rd V-Mode'), zlabel('5th V-Mode')
title('Projection onto V-modes 2, 3, 5')
legend
set(gca,'Fontsize', 14)
clc

%% 2 digits
LDAsuc = zeros(10);
LDAsuct = zeros(10);
for i = 1:10
   train_data{i} = M(:,find(labels==i-1));
   test_data{i} = Mtest(:,find(test_labels==i-1));
```