LDA Fast Prototyping
Sabiha Barlaskar
Student ID: 917932501

PROCESS:

1. Acquired an initial set of 32X32 male and female images

```
path_train_male = dir('/Users/sabihabarlaskar/Documents/MATLAB/LDA_data/Train_Male/
small_*.TIF');
[female_n,female_col] = size(path_train_female);
tau_female = ones(1024,female_n);

for i =1:female_n
   filename_f = strcat('/Users/sabihabarlaskar/Documents/MATLAB/LDA_data/
Train_Female/',path_train_female(i).name);
   %Read the images %
   I_f = double(imread(filename_f));
   %Vectorize the images into a column vector of size N^2 X 1 and store it
   %in a matrix to form an image space
   V_I_f = I_f(:);
   tau_female(:,i) = V_I_f;
end

[male_n,male_col] = size(path_train_male);

tau_male = ones(1024,male_n);
for i = 1:male_n
   filename_m = strcat('/Users/sabihabarlaskar/Documents/MATLAB/LDA_data/
Train_Male/',path_train_male(i).name);
   %Read the images %
   I_m = double(imread(filename_m));
   %Vectorize the images into a column vector of size N^2 X 1 and store it
   %in a matrix to form an image space
   V_I_m = I_m(:);
   tau_male(:,i) = V_I_m;
end
```

2. Performed PCA on the images to find the top k eigen vectors

```
tau_all = [tau_female tau_male];
psi = mean(tau_all,2);
n = female_n + male_n;
phi = tau_all - repmat(psi,1,n);
size(phi);
%A is a matrix that contains the phi of all the images
A = phi;
C = A * A';
[eigvec,eigval] = eig(C);
eigval = diag(eigval);
[sortedeigval, eig_indices] = sort(abs(eigval),'descend');
Sorted_eig = eigvec(:,eig_indices);
size(sortedeigval);
total_eigval = sum(sortedeigval);
var_covered = sortedeigval(1);
k = 2;
while((var_covered/total_eigval)<0.96)
   var_covered = var_covered + sortedeigval(k);
   k = k + 1;
end
k
top_k_eig_vec = Sorted_eig(:,1:k);
```

3. Formed the PCA model by projecting the eigen vectors

```
projected_tau_all = top_k_eig_vec' * phi;
projected_tau_female = projected_tau_all(:,1:female_n);
projected_tau_male = projected_tau_all(:,female_n+1:n);
```

4. Calculated mean of the projected vectors

```
mean_all = mean(projected_tau_all,2);
mean_female= mean(projected_tau_female,2);
mean_male= mean(projected_tau_male,2);
```

5. Calculated the within class scatter matrix Sw for class separability

```
S_w = S_b_female + S_b_male;
```

6. Calculated the between class scatter matrix Sb for class separability

```
diff_female = projected_tau_female - repmat(mean_female,1,female_n);
diff_male = projected_tau_male - repmat(mean_male,1,male_n);
S_b_female = diff_female * diff_female';
S_b_male = diff_male * diff_male';
```

7. Calculated the top eigen vectors and calculated a single principle component for matrix inv(SW)SB to obtain the linear discriminants

```
S_mat = (inv(S_w))*(S_b);

[eigvec_lda,eigval_lda] = eig(S_mat);
eigval_lda = diag(eigval_lda);
[sortedeigval_lda, eig_indices_lda] = sort(abs(eigval_lda),'descend');

Sorted_eig_lda = eigvec_lda(:,eig_indices_lda);
principle_comp = Sorted_eig_lda(:,1);
```

The eigenvectors form the new axes of the new feature subspace.

8. Calculated the discriminant slope

```
discriminant_slope = principle_comp' * top_k_eig_vec';
```

We take the top k eigenvectors as it gives us the direction of maximum variance and give us the maximum information.

9. Calculated the discriminant intercept

```
discriminant_intercept = principle_comp' * ((mean_female + mean_male)/2);
```

10. Calculated the test condition that determines whether an image is male or female

```
test_condition_male = discriminant_slope * (tau_male - psi) - discriminant_intercept;
test_condition_female = discriminant_slope * (tau_female - psi) - discriminant_intercept
```

11. Calculated the accuracy of prediction of the model

```
for i = 1:predicted_females
    if test_condition_female(i)< 0
        count1 = count1 + 1;
    end
end
acc_female = count1/predicted_females
count2 = 0;
[n,predicted_males] = size(test_condition_male);
for i = 1:predicted_males
    if test_condition_male(i)> 0
        count2 = count2 + 1;

    end
end

acc_male = count2/predicted_males
```

VISUALIZATIONS:

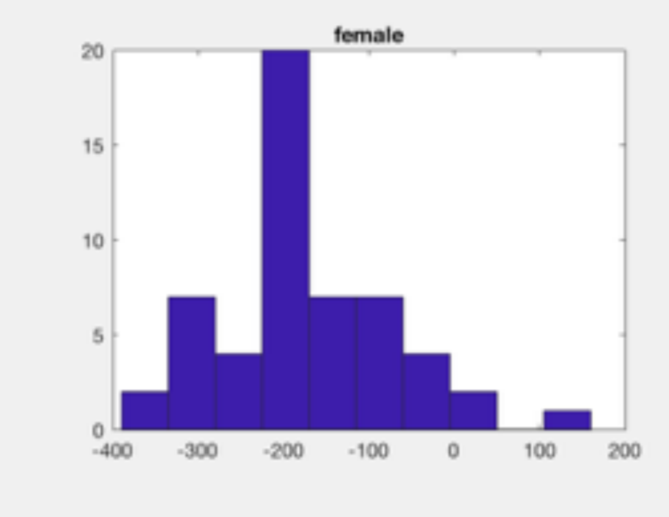

Figure 1: Histogram of the distribution of female (<0)

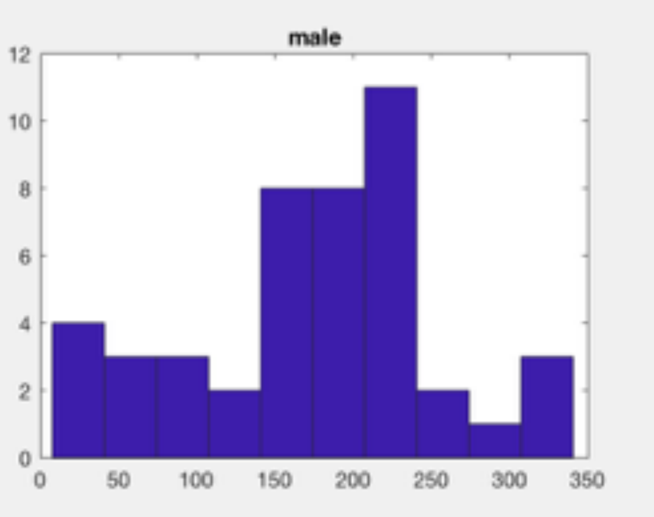

Figure 2: Histogram of the distribution of male (>0)

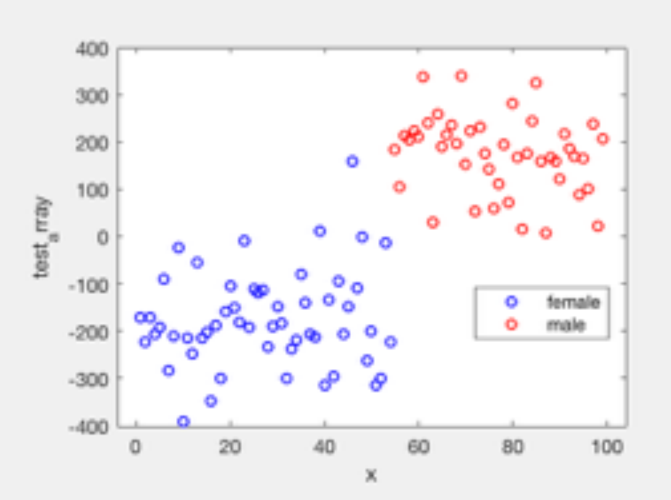

Figure 3: Scatter plot showing the classes of male and female

RESULTS:

Variance	K	Female Accuracy	Male Accuracy
90%	31	90.74%	95.56%
91%	34	90.74%	97.78%
92%	37	94.44%	97.78%
93%	41	96.3%	100%
94%	45	94.44%	97.78%
95%	50	94.44%	97.78%
96%	56	3.70%	2.22%

Figure 4:Table showing the different accuracies by changing variance and k values

For the training model, I got the best accuracy for k = 41 that covers 93% variance. On increasing the variance, the accuracy drops as k increases. This is a classic case of overfitting, as with the increase in number of features, the accuracy decreases.

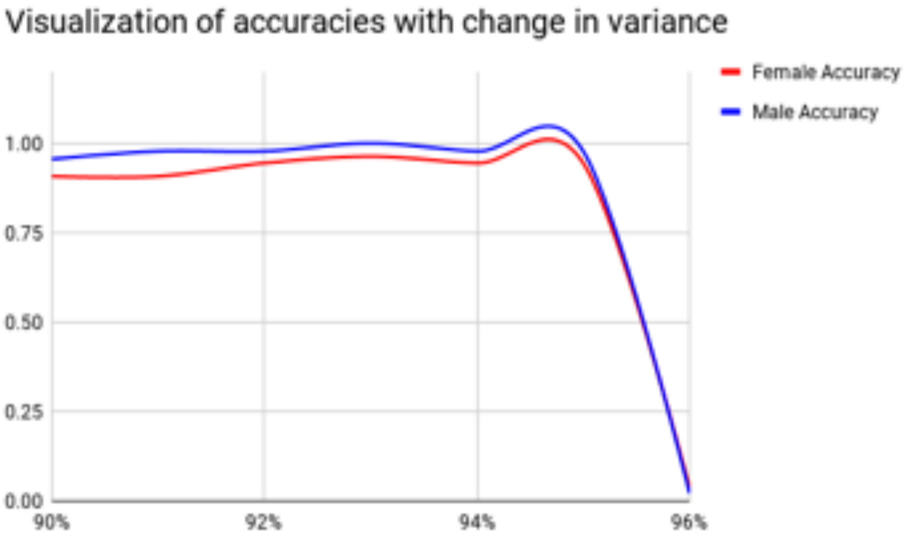

Figure 5

TESTING ON UNKNOWN DATA

Using this model, I tested for a new set of testing data.
For training purpose, I took a set of 44 female images and 37 male images. I used 10 female and 8 male images for test purpose.
I found the following results:

Variance	K	Female accuracy	Male accuracy
90%	28	40%	87.5%
91%	30	40%	37.5%
92%	33	50%	37.5%
93%	36	70%	75%
94%	39	70%	75%
95%	43	20%	37.5%
96%	48	20%	37.5%

Figure 6:Table showing the different accuracies by changing variance and k values

Again, I got the highest accuracy for k = 43, that covers 93% and 94% variance and the accuracy drops with higher variance and higher values of k.

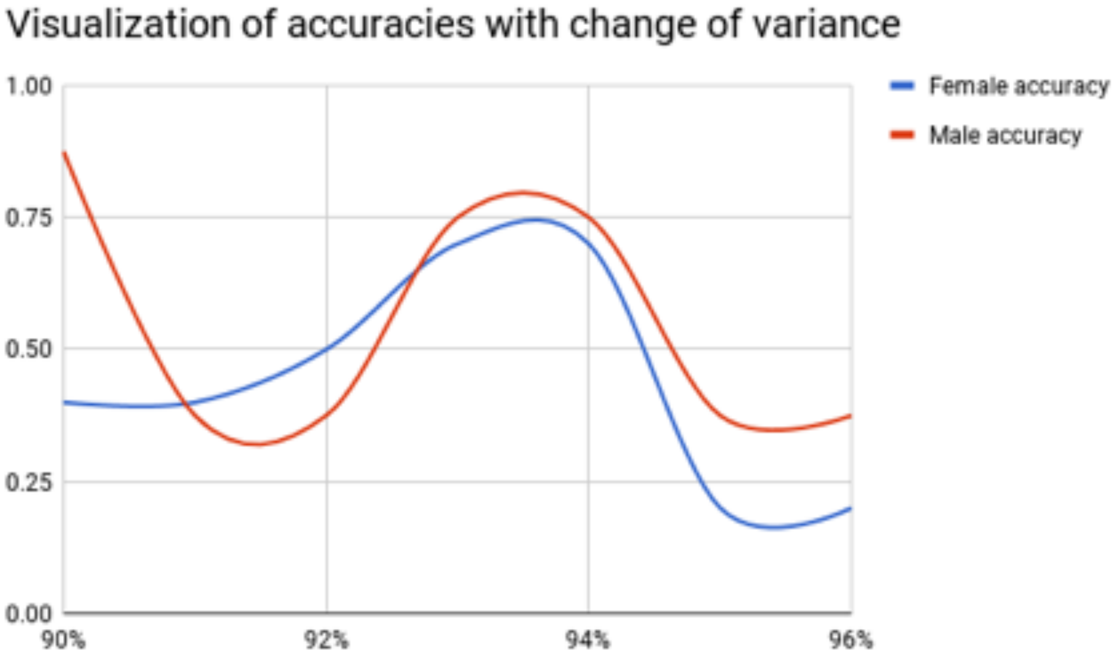

Figure 7

CODE FOR BUILDING THE TRAINING MODEL

```
path_female = dir('/Users/sabihabarlaskar/Documents/MATLAB/FaceClassification_Data/
Female/small_*.TIF');
path_male = dir('/Users/sabihabarlaskar/Documents/MATLAB/FaceClassification_Data/Male/
small_*.TIF');

tau_female = ones(1024,54);
for i =1:54
   filename_f = strcat('/Users/sabihabarlaskar/Documents/MATLAB/FaceClassification_Data/
Female/',path_female(i).name);
   %Read the images %
   I_f = double(imread(filename_f));
   %Vectorize the images into a column vector of size N^2 X 1 and store it
   %in a matrix to form an image space
   V_I_f = I_f(:);
   tau_female(:,i) = V_I_f;
end
%mean_female = mean(tau_female,2);

tau_male = ones(1024,45);
for i = 1:45
   filename_m = strcat('/Users/sabihabarlaskar/Documents/MATLAB/FaceClassification_Data/
Male/',path_male(i).name);
   %Read the images %
   I_m = double(imread(filename_m));
   %Vectorize the images into a column vector of size N^2 X 1 and store it
   %in a matrix to form an image space
   V_I_m = I_m(:);
   tau_male(:,i) = V_I_m;
end
%mean_male = mean(tau_male,2);
tau_all = [tau_female tau_male];
psi = mean(tau_all,2);

phi = tau_all - repmat(psi,1,99);
%A is a matrix that contains the phi of all the images
A = phi;
C = A * A';
[eigvec,eigval] = eig(C);
eigval = diag(eigval);
```

```
[sortedeigval, eig_indices] = sort(abs(eigval),'descend');
Sorted_eig = eigvec(:,eig_indices);
size(sortedeigval);
total_eigval = sum(sortedeigval);
var_covered = sortedeigval(1);
k = 2;
while((var_covered/total_eigval)<0.93)
   var_covered = var_covered + sortedeigval(k);
   k = k + 1;
end
k
top_k_eig_vec = Sorted_eig(:,1:k);
size(top_k_eig_vec);
% figure('NumberTitle','off','Name','Eigen faces')
%
% for i = 1:k
%     colormap('gray');
%     subplot(10,5,i);
%     imagesc(reshape(top_k_eig_vec(:, i), 32, 32));
% end
projected_tau_all = top_k_eig_vec' * phi;
size(projected_tau_all);
projected_tau_female = projected_tau_all(:,1:54);
projected_tau_male = projected_tau_all(:,55:99);

mean_all = mean(projected_tau_all,2);
mean_female= mean(projected_tau_female,2);
mean_male= mean(projected_tau_male,2);



diff_female = projected_tau_female - repmat(mean_female,1,54);
diff_male = projected_tau_male - repmat(mean_male,1,45);

S_female = diff_female * diff_female';
S_male = diff_male * diff_male';

S_w = S_female + S_male;

diff_female_from_mean = mean_female - mean_all;
diff_male_from_mean = mean_male - mean_all;

S_female_from_mean = diff_female_from_mean * diff_female_from_mean';
```

```
S_male_from_mean = diff_male_from_mean * diff_male_from_mean';

S_b = S_female_from_mean + S_male_from_mean;

S_mat = (inv(S_w))*(S_b);

[eigvec_lda,eigval_lda] = eig(S_mat);
eigval_lda = diag(eigval_lda);
[sortedeigval_lda, eig_indices_lda] = sort(abs(eigval_lda),'descend');

Sorted_eig_lda = eigvec_lda(:,eig_indices_lda);
principle_comp = Sorted_eig_lda(:,1);

discriminant_slope = principle_comp' * top_k_eig_vec';
size(discriminant_slope);
discriminant_intercept = principle_comp' * ((mean_female + mean_male)/2)

size(discriminant_slope);
size(discriminant_intercept);

test_female = tau_all(:,1);
test_male = tau_all(:,99);
test_condition_female = discriminant_slope * (tau_female - psi) - discriminant_intercept
test_condition_male = discriminant_slope * (tau_male - psi) - discriminant_intercept;
size(test_condition_male);

test_array = [test_condition_female test_condition_male];

x = (1:99);
groups=[ones(54,1);ones(45,1)+1];
c = linspace(1,10,length(x));

subplot(2,2,1);

hist(test_condition_female);
title("female");

subplot(2,2,2);
hist(test_condition_male);
title("male");
subplot(2,2,3);
gscatter(x,test_array,groups,'br','oo',5);
legend('female','male','north');
```

```
count1 = 0;
[n,predicted_females] = size(test_condition_female);
for i = 1:predicted_females
    if test_condition_female(i)< 0
        count1 = count1 + 1;

    end

end
%Accuracy of prediction of females
acc_female = count1/predicted_females
count2 = 0;
[n,predicted_males] = size(test_condition_male);
for i = 1:predicted_males
    if test_condition_male(i)> 0
        count2 = count2 + 1;

    end

end
%Accuracy of prediction of males
acc_male = count2/predicted_males
```

CODE FOR TESTING NEW DATA USING THE TRAINING MODEL

```
path_train_female = dir('/Users/sabihabarlaskar/Documents/MATLAB/LDA_data/Train_Female/small_*.TIF');
path_train_male = dir('/Users/sabihabarlaskar/Documents/MATLAB/LDA_data/Train_Male/small_*.TIF');
[female_n,female_col] = size(path_train_female);
tau_female = ones(1024,female_n);

for i =1:female_n
    filename_f = strcat('/Users/sabihabarlaskar/Documents/MATLAB/LDA_data/Train_Female/',path_train_female(i).name);
    %Read the images %
    I_f = double(imread(filename_f));
    %Vectorize the images into a column vector of size N^2 X 1 and store it
    %in a matrix to form an image space
    V_I_f = I_f(:);
    tau_female(:,i) = V_I_f;
end
```

```
[male_n,male_col] = size(path_train_male);

tau_male = ones(1024,male_n);
for i = 1:male_n
   filename_m = strcat('/Users/sabihabarlaskar/Documents/MATLAB/LDA_data/
Train_Male/',path_train_male(i).name);
   %Read the images %
   I_m = double(imread(filename_m));
   %Vectorize the images into a column vector of size N^2 X 1 and store it
   %in a matrix to form an image space
   V_I_m = I_m(:);
   tau_male(:,i) = V_I_m;
end
%mean_male = mean(tau_male,2);
tau_all = [tau_female tau_male];
psi = mean(tau_all,2);
n = female_n + male_n;
phi = tau_all - repmat(psi,1,n);
size(phi);
%A is a matrix that contains the phi of all the images
A = phi;
C = A * A';
[eigvec,eigval] = eig(C);
eigval = diag(eigval);
[sortedeigval, eig_indices] = sort(abs(eigval),'descend');
Sorted_eig = eigvec(:,eig_indices);
size(sortedeigval);
total_eigval = sum(sortedeigval);
var_covered = sortedeigval(1);
k = 2;
while((var_covered/total_eigval)<0.93)
   var_covered = var_covered + sortedeigval(k);
   k = k + 1;
end
k
top_k_eig_vec = Sorted_eig(:,1:k);
size(top_k_eig_vec);
% figure('NumberTitle','off','Name','Eigen faces')
%
% for i = 1:k
%     colormap('gray');
%     subplot(10,5,i);
%     imagesc(reshape(top_k_eig_vec(:, i), 32, 32));
% end
projected_tau_all = top_k_eig_vec' * phi;
projected_tau_female = projected_tau_all(:,1:female_n);
projected_tau_male = projected_tau_all(:,female_n+1:n);
%scatter(projected_tau_female,projected_tau_male);
size(projected_tau_male);
mean_all = mean(projected_tau_all,2);
mean_female= mean(projected_tau_female,2);
mean_male= mean(projected_tau_male,2);



diff_female = projected_tau_female - repmat(mean_female,1,female_n);
size(diff_female);
diff_male = projected_tau_male - repmat(mean_male,1,male_n);
%Step 6
S_b_female = diff_female * diff_female';
S_b_male = diff_male * diff_male';
%Step 6
S_w = S_b_female + S_b_male;

diff_female_from_mean = mean_female - mean_all;
diff_male_from_mean = mean_male - mean_all;
%Covariance of LDA
S_female_from_mean = diff_female_from_mean * diff_female_from_mean';
S_male_from_mean = diff_male_from_mean * diff_male_from_mean';

S_b = S_female_from_mean + S_male_from_mean;

S_mat = (inv(S_w))*(S_b);

[eigvec_lda,eigval_lda] = eig(S_mat);
eigval_lda = diag(eigval_lda);
[sortedeigval_lda, eig_indices_lda] = sort(abs(eigval_lda),'descend');

Sorted_eig_lda = eigvec_lda(:,eig_indices_lda);
principle_comp = Sorted_eig_lda(:,1);

discriminant_slope = principle_comp' * top_k_eig_vec';
discriminant_intercept = principle_comp' * ((mean_female + mean_male)/2);

size(discriminant_slope);
size(discriminant_intercept);
test_condition = discriminant_slope * (tau_male - psi) - discriminant_intercept;
%male(negative)
%female(positive)

%Testing

path_test_female = dir('/Users/sabihabarlaskar/Documents/MATLAB/LDA_data/Test_female/
small_*.TIF');
path_test_male = dir('/Users/sabihabarlaskar/Documents/MATLAB/LDA_data/Test_male/
small_*.TIF');
[female_t_n,t_col] = size(path_test_female);
tau_female_test = ones(1,female_t_n);
for i =1:female_t_n
   filename_f_test = strcat('/Users/sabihabarlaskar/Documents/MATLAB/LDA_data/
Test_female/',path_test_female(i).name);
   %Read the test images %
   I_f_test = double(imread(filename_f_test));
   V_I_f_test = I_f_test(:);
   %tau_female_test(:,i) = V_I_f_test;
   test_condition_female = discriminant_slope * (V_I_f_test - psi) - discriminant_intercept;
   tau_female_test(i) = test_condition_female;

end

[male_t_n,t_m_col] = size(path_test_male);
tau_male_test = ones(1,male_t_n);
for i =1:male_t_n
   filename_m_test = strcat('/Users/sabihabarlaskar/Documents/MATLAB/LDA_data/
Test_male/',path_test_male(i).name);
   %Read the test images %
   I_m_test = double(imread(filename_m_test));
   V_I_m_test = I_m_test(:);
   %tau_male_test(:,i) = V_I_m_test;
   test_condition_male = discriminant_slope * (V_I_m_test - psi) - discriminant_intercept;
   tau_male_test(i) = test_condition_male;
end


count1 = 0;
for i = 1:8
   if tau_male_test(i)<0
      count1 = count1+1;
   end
end
acc_male = count1/8

count2 = 0;
for i = 1:10
   if tau_female_test(i) > 0
      count2 =  count2+1;
   end
end

acc_female = count2/10

%test_condition = discriminant_slope * (V_I_f_test - psi) - discriminant_intercept
```

CONCLUSION:

By doing this fast prototyping exercise, I came to understand the difference between PCA and LDA and also the importance of eigen vectors for dimensionality reduction.
PCA finds the axes with maximum variance whereas LDA tries to find the axes for best class separability. In practice, often a LDA is done followed by a PCA for dimensionality reduction. LDA is a supervised classification method.

REFERENCES:

1. Daniel L. Swets and John (Juyang) Weng *Using Discriminant Eigenfeatures for Image Retrieval*
2. https://bidal.sfsu.edu/~kazokada/csc872/csc872_exe11.pdf
3. *PCA versus LDA, IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(2): 228-233 2001)*