Mateus Toppin
EK381 Homework 10
28 April 2023

dimensionality_reduction.m

```
%This function takes in a data matrix Xrun, mean vector mu,
%eigenvector matrix V, and eigenvalues D, and dimension k.
%It selects the k eigenvectors corresponding to the k largest
%eigenvalues, centers the data by subtracting mu, and projects
%the centered data to k dimensions by multiplying by the matrix
%of k eigenvectors.
function Xrun_reduced = dimensionality_reduction(Xrun,mu,V,D,k)
[sD,sI] = sort(diag(D),'descend');
eigenk = [V(:,sI(1:k))];
Xrun_reduced = (Xrun - mu) * eigenk;
end
```

nearest_neighbor.m

```
%This function takes in a data matrix Xrun as well a training data matrix
%Xtrain and a labeled output Ytrain and produces a column vector guesses,
%correspoding to the nearest neighbor classifier.
function guesses = nearest_neighbor(Xrun,Xtrain,Ytrain)
dist = [];
guesses = [];
for i = 1:size(Xrun, 1)
   for k = 1:size(Xtrain, 1)
       dist(k)  = sum((Xrun(i,:) - Xtrain(k,:)).^2);
   end
   minimum_dist = min(dist);
   small_index = find(dist == minimum_dist);
   guesses(i) = [Ytrain(small_index)];
end
guesses = guesses';
end
```

labeled_mean_cov.m

```
%This function takes in a data matrix X, corresponding vector
%of labels Y, and a desired label. It outputs the the number
%of samples with desiredlabel as n_label as well as the sample
%mean vector mu_label (as a row vector) and sample covariance
%matrix sigma_label for the data in X whose labels in Y are equal
%to the desired label.
function [n_label,mu_label,sigma_label] = labeled_mean_cov(X,Y,desiredlabel)
Y_index = find(Y == desiredlabel);
```

```
n_label = length(Y_index);
mu_label = mean(X(Y_index,:)); % X mean
sigma_label = cov(X(Y_index,:)); % Y sigma
end
```

LDA.m

```
%This function takes in a data matrix Xrun as well the mean vectors mu0, mu1
%and the covariance matrices sigma0, sigma1 estimated from the training data
%and produces a column vector guesses, corresponding to the ML rule for
Gaussian vectors
%with different means and the same covariance matrix, which is referred to as
%Linear Discriminant Analysis (LDA) in machine learning.
function guesses = LDA(Xrun,mu0,mu1,sigmapooled)
guesses = []; % initializing it
for i = 1:size(Xrun)
   currentdata = Xrun(i,:);
   guesses(i) = ((currentdata - mu0) * pinv(sigmapooled) * (currentdata -
mu0)') >= ((currentdata - mu1) * pinv(sigmapooled) * (currentdata - mu1)');
end
guesses = guesses';
end
```

QDA.m

```
%This function takes in a data matrix Xrun as well the mean vectors mu0, mu1
%and the covariance matrices sigma0, sigma1 estimated from the training data
%and produces a column vector guesses, corresponding to the ML rule for
Gaussian vectors
%with different means and different covariance matrices, which is referred to
as
%Quadratic Discriminant Analysis (QDA) in machine learning.
function guesses = QDA(Xrun,mu0,mu1,sigma0,sigma1)
guesses = [];
for i = 1:size(Xrun)
   currentdata = Xrun(i,:);
   var1 = sum(log(eig(sigma1))) + ((currentdata - mu1)) * pinv(sigma1) *
(currentdata - mu1)';
   var2 = sum(log(eig(sigma0))) + ((currentdata - mu0)) * pinv(sigma0) *
(currentdata - mu0)';
   guesses(i) = var1 <= var2;
end
guesses = guesses';
end
```

hw10_matlab.m

What happens to training error rate as the dimension increases?

As the dimension increases, LDA's timing rate remains constant while QDA's timing rate continuously decreases overtime.

What happens to the test error rate as the dimension increases?

As the dimension increases, LDA's test error rate exponentially increases and QDA's test error rate shortly decreases then slowly increases.

For LDA, determine the values of k that yields the lowest training error and the lowest test error. Record these values of k as well as the resulting training and test error rates.

Lowest training error: k= 500

Lowest training error rate: k= 0.027

—---

Lowest test error: k= 250

Lowest test error rate: k= 0.0845

For QDA, determine the values of k that yields the lowest training error and the lowest test error. Record these values of k as well as the resulting training and test error rates.

Lowest training error: k= 500

Lowest training error rate: k= 0

—---

Lowest test error: k= 250

Lowest test error rate: k= 0.0495

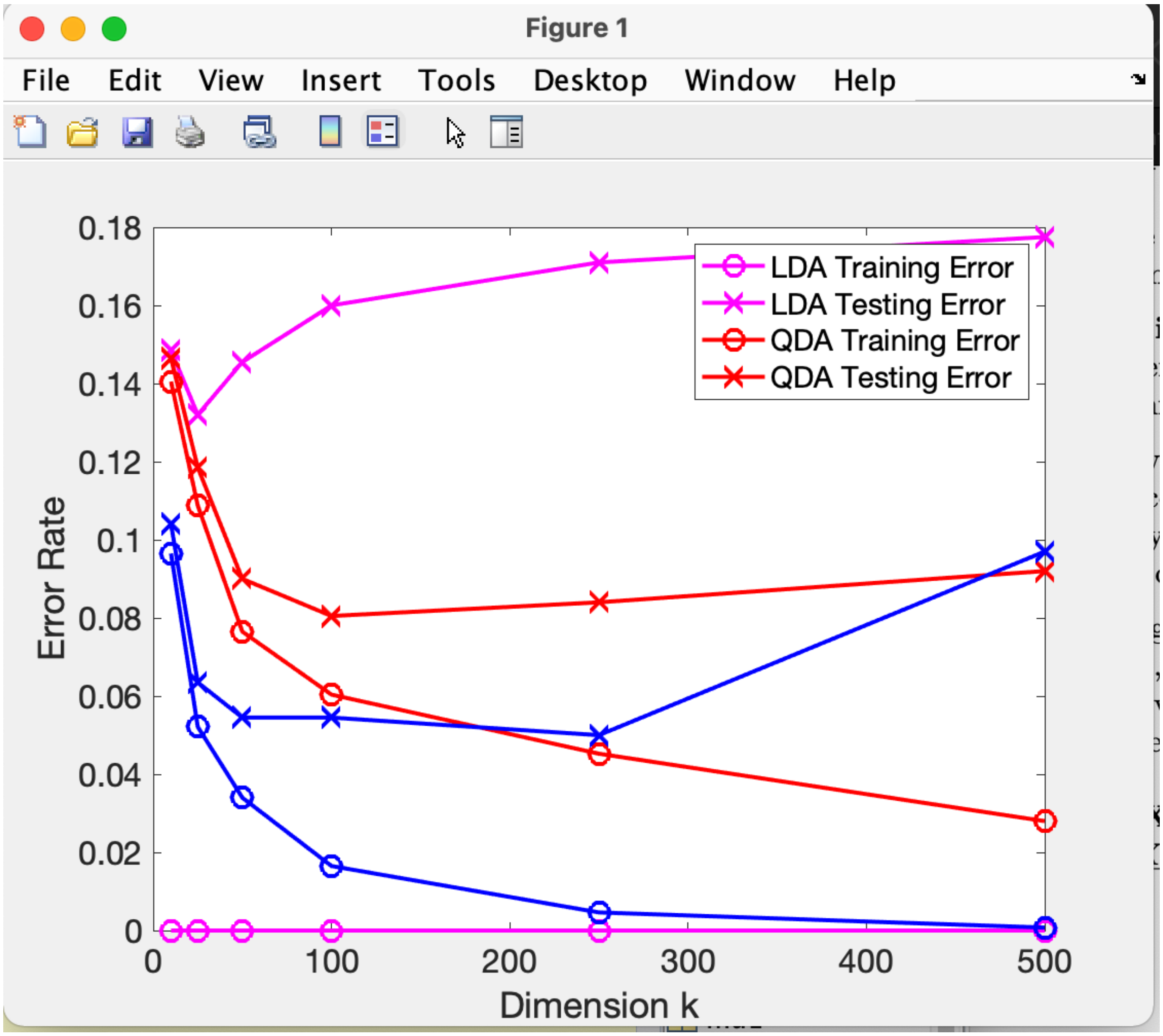
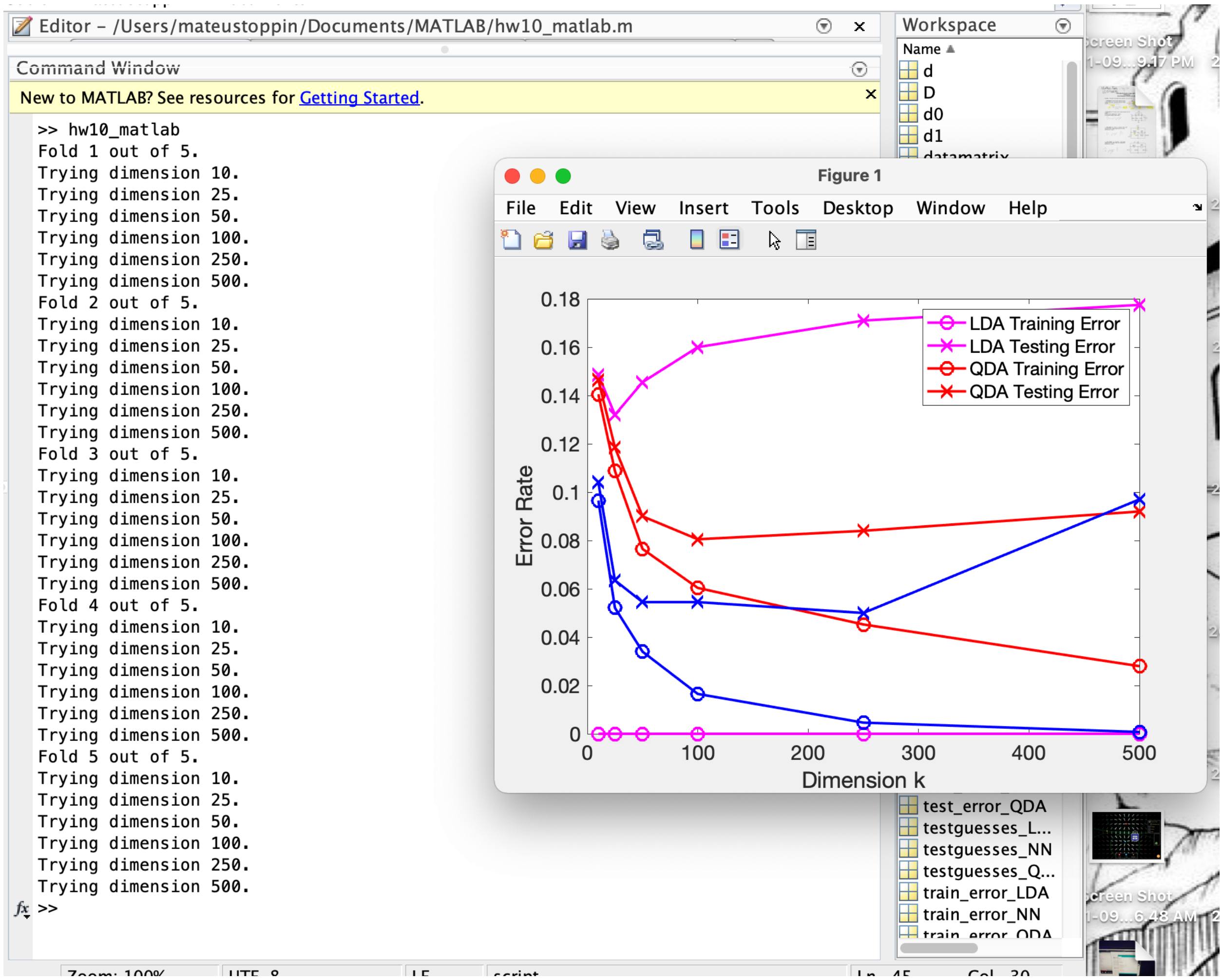


Figure 1
File
Edit
View
Insert
Tools
Desktop
Window
Help
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
Error Rate
0
100
200
300
400
500
Dimension k
LDA Training Error
LDA Testing Error
QDA Training Error
QDA Testing Error


Editor – /Users/mateustoppin/Documents/MATLAB/hw10_matlab.m
Command Window
New to MATLAB? See resources for Getting Started.
>> hw10_matlab
Fold 1 out of 5.
Trying dimension 10.
Trying dimension 25.
Trying dimension 50.
Trying dimension 100.
Trying dimension 250.
Trying dimension 500.
Fold 2 out of 5.
Trying dimension 10.
Trying dimension 25.
Trying dimension 50.
Trying dimension 100.
Trying dimension 250.
Trying dimension 500.
Fold 3 out of 5.
Trying dimension 10.
Trying dimension 25.
Trying dimension 50.
Trying dimension 100.
Trying dimension 250.
Trying dimension 500.
Fold 4 out of 5.
Trying dimension 10.
Trying dimension 25.
Trying dimension 50.
Trying dimension 100.
Trying dimension 250.
Trying dimension 500.
Fold 5 out of 5.
Trying dimension 10.
Trying dimension 25.
Trying dimension 50.
Trying dimension 100.
Trying dimension 250.
Trying dimension 500.
>>
Workspace
Name
d
D
d0
d1
Figure 1
File
Edit
View
Insert
Tools
Desktop
Window
Help
LDA Training Error
LDA Testing Error
QDA Training Error
QDA Testing Error
Error Rate
Dimension k
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
100
200
300
400
500
test_error_QDA
testguesses_L...
testguesses_NN
testguesses_Q...
train_error_LDA
train_error_NN