Comparison of PCA and Linear Regression alongside Logistic Regression method in terms of accuracy for Diagnosing Breast Cancer

Department of Electrical and Computer Engineering
ECE 503 Optimization for Machine Learning

PROJECT REPORT

Submitted by

Shaima Patel, V00949940

University of Victoria

Date: 21st December,2019

Table of Contents

Topics **Page Number**

List of Figures

Table Name **Page Number**

Abstract

As the name clearly suggests, in this work, I will be comparing the accuracy/efficiency of PCA, and Linear Regression Method using Logistic Regression for diagnosing Breast Cancer. The main aim of this lab experiment is to build computer program to automatic detect breast cancer. We will optimize parameters of logistic regression by using gradient regression.

I will test data samples, by applying these three methods on data sets introduced in our ECE – 503 course. In this project, I will be considering datasets namely: Wisconsin Diagnostic Breast Cancer dataset.

PCA, Logistic Regression and Linear Regression method, are applied on the above considered datasets. We know that, PCA was already applied in Lab - 1 of this course on MNIST dataset and Linear Regression was applied in Lab -2 on IRIS dataset, wheras we applied Logistic Regression in Lab-4 for identifying Cancer. Which is the reason why, I am using these experiments implementations in my work for comparison.

Now, I will apply,Linear Regression and PCA by using help from the studied lab and course materials. Note that I will use Logistic Regression which was applied on Breast Cancer dataset in Lab - 4. So my work includes applying both PCA and Linear Regression method on the Breast Cancer dataset. This dataset contains 480 training samples(180 Benign and 300 Malignant) and 89 testing samples, and these numbers are same for both PCA and Linear Regression. Note that number of features are same for these methods like as in Lab – 1, Lab – 2 and Lab – 4.

But I observed that the training dataset of Breast Cancer Dataset (D_bc_tr), which was used in Lab – 4 is not a structured dataset. If a traing dataset is not structured, then it will be difficult to apply PCA and Linear Regression. Which is the reason why, I arranged the dataset such that first 180 of my training samples are benign and next 300 samples are malignant. The MATLAB code I used for this arrangement is also included in Appendix.

1. Introduction Describing Technical Background of the Problem

The dataset to be used in this experiment collects these 30 feature values from each of 569 patients of which 357 patients were diagnosed as benign while the other 212 patients were diagnosed as malignant. In what follows, the dataset will be referred to as WDBC which stands for Wisconsin Diagnostic Breast Cancer. [2]

Using this dataset, the objective of this lab experiment is to develop a classifier to classify a new and unused sample either to benign or to malignant.

The problem of interest here is diagnosing Cancer samples problem. It is considered as a special case of the prediction problem. Here, the implementation/algorithm will be receiving a new data point x outside the training set and the algorithm which is implemented will predict a class to which that new data point belongs to. Such type of problems are called classification problems.

I agree and believe that this project can be useful in delivering/giving a generalized statement that the performance of PCA is better than that of Linear Regression, which is evident from the results obtained from this project. This is the main aim of my project. As a part of my project, I will be 89 testing samples of Cancer dataset using PCA, Logistic and Linear Regression using help from Logistic Regression.

The good thing about this project is that, I will be doing a 2-class/category classification using Linear Regression on Breast Cancer dataset. However, if number of classes > 2, then it is better to choose vector labels instead of scalar labels.

Further technical details about this classification problem are provided in the next section.

2. Formulation of Problem at hand as an Optimization Problem

Classification is done using:

2.1 Principal Component Analysis

In order to extract features in a lower dimension for a dataset, we need to calculate mean of the data i.e.

$$\boldsymbol{\mu} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{x}_i$$

to generate the centralized dataset, i.e.

$$\hat{X} = \{\boldsymbol{x}_i - \boldsymbol{\mu}, i = 1, 2, \ldots, m\}$$

If X contains multiple data classes, the above step should be applied to all classes as follows:

$$A = \begin{bmatrix} \boldsymbol{x}_1 - \boldsymbol{\mu} & \boldsymbol{x}_2 - \boldsymbol{\mu} & \cdots & \boldsymbol{x}_m - \boldsymbol{\mu} \end{bmatrix}$$

Calculate q-dimensional features *fi* using:

$$\boldsymbol{f}_i = \boldsymbol{U}_q^T (\boldsymbol{x}_i - \boldsymbol{\mu}))$$

In order to represent the point *x* in the jth class, we can represent it as:

$$\hat{\boldsymbol{x}}_j = \boldsymbol{U}_q^{(j)} \boldsymbol{f}_j + \boldsymbol{\mu}_j$$

The Eucledian distance between x and its representation is computed using:

$$e_j = \| \boldsymbol{x} - \hat{\boldsymbol{x}}_j \|_2$$

The sample x is classifiedto a class, whose error value is minimum. In this way, we classify the data. Since this method involves calculation of principal components and features, this method is called Principal Component Analysis.

2.2 Linear Regression (Prediction and Classification)

In order to understand/view classification as an optimization problem, we need to undetstand the basics of Prediction. The theory required for Prediction is as follows:

A training dataset considered and the dimensions of its entities are as follows:

$$\mathcal{D} = \{(\boldsymbol{x}_n, \boldsymbol{y}_n), n = 1, 2, \ldots, N\} \text{ with } \boldsymbol{x}_n \in R^{d\times 1} \text{ and } \boldsymbol{y}_n \in R^{l\times 1}$$

We consider a linear model *f(x,**W**,**b**)* to predict or classify the new data with Weight ***W*** and bias ***b.***

$$f(\boldsymbol{x}, \boldsymbol{W}, \boldsymbol{b}) = \boldsymbol{W}^T \boldsymbol{x} + \boldsymbol{b} \; ; \; \boldsymbol{W} \in R^{d\times l} \text{ and } \boldsymbol{b} \in R^{l\times 1} \; ;$$

The optimization function is:

$$\underset{W,b}{\text{minimize}} \quad \sum_{n=1}^{N} \| f(\boldsymbol{x}_n, \boldsymbol{W}, \boldsymbol{b}) - \boldsymbol{y}_n \|_2^2$$

Where:

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_1 & \boldsymbol{w}_2 & \cdots & \boldsymbol{w}_l \end{bmatrix}, \ \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_l \end{bmatrix}, \text{ and } \ \boldsymbol{y}_n = \begin{bmatrix} y_1^{(n)} \\ y_2^{(n)} \\ \vdots \\ y_l^{(n)} \end{bmatrix}$$

Where:

$$\hat{\boldsymbol{w}}_i = \begin{bmatrix} \boldsymbol{w}_i \\ b_i \end{bmatrix}, \ \ \hat{\boldsymbol{y}}_i = \begin{bmatrix} y_i^{(1)} \\ y_i^{(2)} \\ \vdots \\ y_i^{(N)} \end{bmatrix}$$

And:

$$\begin{bmatrix} \hat{\boldsymbol{w}}_1^* & \hat{\boldsymbol{w}}_2^* & \cdots & \hat{\boldsymbol{w}}_l^* \end{bmatrix} = \left(\hat{\boldsymbol{X}}^T \hat{\boldsymbol{X}} \right)^{-1} \hat{\boldsymbol{X}}^T [\hat{\boldsymbol{y}}_1 \quad \hat{\boldsymbol{y}}_2 \quad \cdots \quad \hat{\boldsymbol{y}}_l] = \left(\hat{\boldsymbol{X}}^T \hat{\boldsymbol{X}} \right)^{-1} \hat{\boldsymbol{X}}^T \begin{bmatrix} \boldsymbol{y}_1^T \\ \boldsymbol{y}_2^T \\ \vdots \\ \boldsymbol{y}_N^T \end{bmatrix}$$

And:

$$\boldsymbol{W}^* = \begin{bmatrix} \boldsymbol{w}_1^* & \boldsymbol{w}_2^* & \cdots & \boldsymbol{w}_l^* \end{bmatrix} \text{ and } \ \boldsymbol{b}^* = \begin{bmatrix} b_1^* \\ b_2^* \\ \vdots \\ b_l^* \end{bmatrix}$$

The optimal linear model will be:

$$f(\boldsymbol{x}, \boldsymbol{W}^*, \boldsymbol{b}^*) = \boldsymbol{W}^{*T} \boldsymbol{x} + \boldsymbol{b}^* \quad \text{and} \quad f(\boldsymbol{x}) = \boldsymbol{W}^{*T} \boldsymbol{x} + \boldsymbol{b}^*$$

This function is called discriminant function. For classification, the discriminant function becomes:

$$f(\boldsymbol{x}) = \boldsymbol{w}^{*T} \boldsymbol{x} + b^*$$

Where:

$$\hat{\boldsymbol{w}} = \begin{bmatrix} \boldsymbol{w} \\ b \end{bmatrix} \ \text{ and } \ \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

When doing 2-class classification, the new point x which is outside the dataset D:

$$\text{belongs to class } \mathcal{P} \text{ if } \boldsymbol{x}^T \boldsymbol{w}^* + b^* > 0$$

$$\text{belongs to class } \mathcal{N} \text{ if } \boldsymbol{x}^T \boldsymbol{w}^* + b^* < 0$$

And the decision boundary is given by:

$$f(\boldsymbol{x}) = \boldsymbol{w}^{*T} \boldsymbol{x} + b^* = 0$$

This term will define a flat plane which divides the input space into two parts. The points on the side of the plane will have *f(x)*values with same sign and the points on the other side of the plane will have *f(x)* values with opposite sign. In this way, classification is done in Linear Regression Method. Similarly, this can be extended for multi-class case.

2.3 Logistic Regression

Given a dataset D, logistic regression studied in the course notes is applicable to two-category classification problems. In what follows, it is assumed that dataset has been

normalized, and for notation simplicity it is still denoted by D = {(x_n,y_n) for n=1,2,...,N} with N = 480. The classification is performed in two steps.
(i) Minimize the objective function

$$f(\hat{\boldsymbol{w}}) = \frac{1}{N} \sum_{n=1}^{N} \ln\left(1 + e^{-y_n \hat{\boldsymbol{w}}^T \hat{\boldsymbol{x}}_n}\right)$$

with respect to parameter $\hat{\boldsymbol{w}} \in R^{31\times1}$ where

$$\hat{\boldsymbol{w}} = \begin{bmatrix} \boldsymbol{w} \\ b \end{bmatrix} \text{ and } \hat{\boldsymbol{x}}_n = \begin{bmatrix} \boldsymbol{x}_n \\ 1 \end{bmatrix}$$

and x_n and y_n for n = 1, 2, ..., N are provided by dataset D.
To apply GD algorithm for minimizing f (wˆ) , the gradient $\nabla_{w^\wedge}$ f (wˆ) is evaluated in closed-form as

$$\nabla_{\hat{\boldsymbol{w}}} f(\hat{\boldsymbol{w}}) = -\frac{1}{N} \sum_{n=1}^{N} \frac{y_n e^{-y_n \hat{\boldsymbol{w}}^T \hat{\boldsymbol{x}}_n}}{\left(1 + e^{-y_n \hat{\boldsymbol{w}}^T \hat{\boldsymbol{x}}_n}\right)} \hat{\boldsymbol{x}}_n$$

Note that objective function f (wˆ) is strictly convex, hence it admits a unique global minimizer and this minimizer is characterized by its gradient $\nabla_{\mathrm{W}^{\wedge}}$ f (wˆ) being zero. The convexity of f (wˆ)

also assures that the GD algorithm is insensitive to the choice of initial point wˆ $_{0}$.

(ii) If we call the subset of training samples with label “–1” class N and the subset of training

samples with label “1” class P, then minimizer {w*,b*} obtained from step (i) can be used to classify a new data point x outside the training data to class P or class N in accordance with

$$\begin{cases} \boldsymbol{x} \in \mathcal{N} & \text{if } \boldsymbol{w}^{*T}\boldsymbol{x} + b^{*} < 0 \\ \boldsymbol{x} \in \mathcal{P} & \text{if } \boldsymbol{w}^{*T}\boldsymbol{x} + b^{*} > 0 \end{cases}$$

3. Solution Methods

As I said, I will be applying PCA,Logistic Regression and Linear Regression on above mentioned 3 datasets. The solution methods are discussed in detail as follows for all the datasets.

3.1: For Breast Cancer Dataset:

PCA (Newly Implemented Algorithm): As I mentioned in the abstract, the training data of Breast cancer dataset is not ordered. Which means, benign and malignant data entities are completely shuffled and are arranged randomly in the matrix D_bc_tr. Based on the label, which is available in the training matrix, I have fetched the data and arranged it such that I can do PCA and Linear Regression. I know that there are 180 benign samples and 300 malignant samples (evident from labels). Which is why, first I have fetched 180 columns and then 300 columns and then I applied PCA on them. Which is just like/similar to taking 1600 columns each time and repeating it for 10 times for applying PCA in Lab – 1.

Linear Regression (Newly Implemented Algorithm): The above mentioned pre-processing steps are required in this method also. Rest all the steps will be similar and can be found from other Linear Regression methods. More over, this is two class classification using Linear Regression.

Logistic Regression: I have implemented the same experiment in the Lab – 4 for this section. It helped me compare my above implemented results.

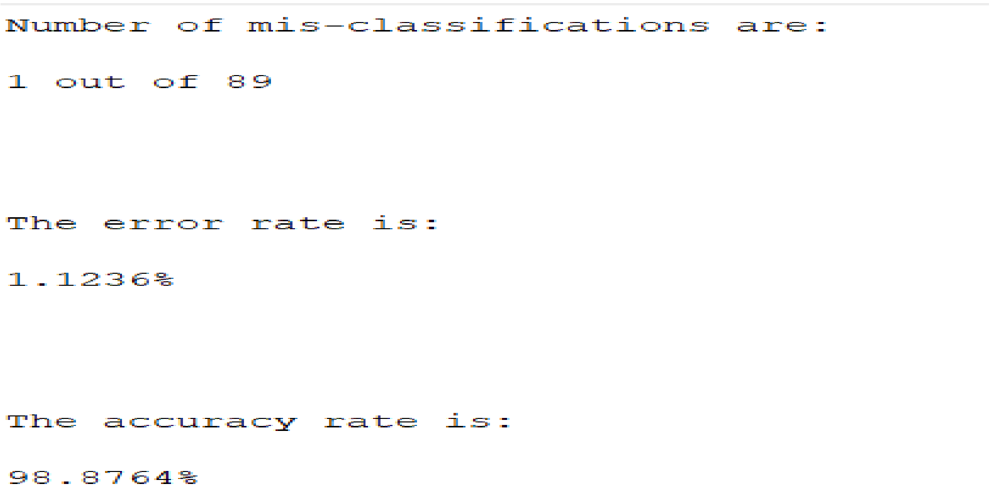
The results of my algorithms/implementations are presented in the next section.

4. Computer Simulations and Numerical Results

Computer simulations and numerical results for existing algorithms are already known and are done in the labs (Numerical results for these algorithms are available in the above section). But, for newly implemented algorithms, the resuls are as follows:

4.1: PCA (Newly Implemented Algorithm):

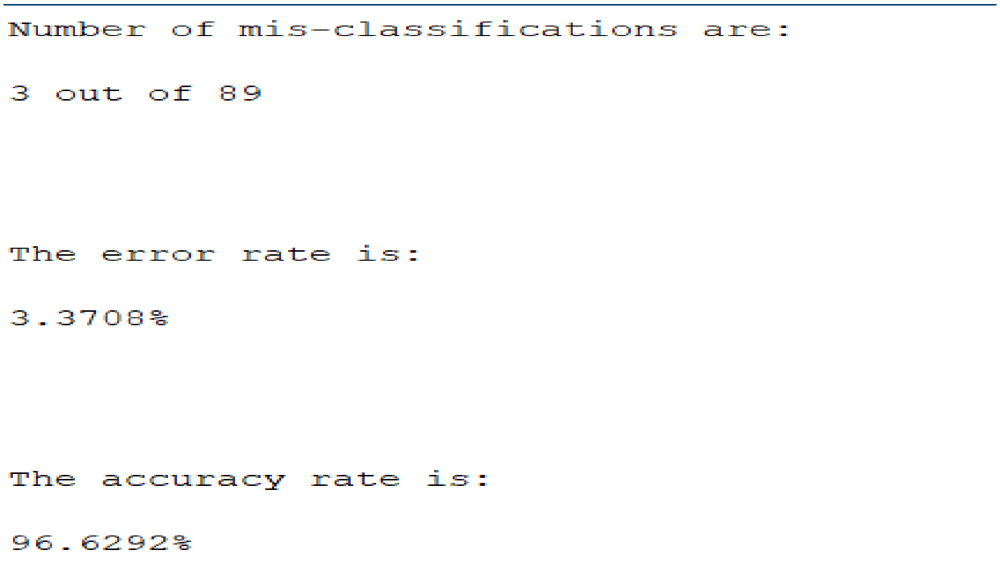
As I mentioned, first I made the dataset into a structured format. And then, I applied PCA on that dataset as explained above. The number of mis-classifications were: 1 out of 89. The error rate and accuracy rate values are: 1.1236% and 98.8764% respectively. The result obtained is of vital importance and is better than its Linear Regression method.



```
Number of mis-classifications are:

1 out of 89



The error rate is:

1.1236%



The accuracy rate is:

98.8764%
```

Fig – 1: Output for PCA

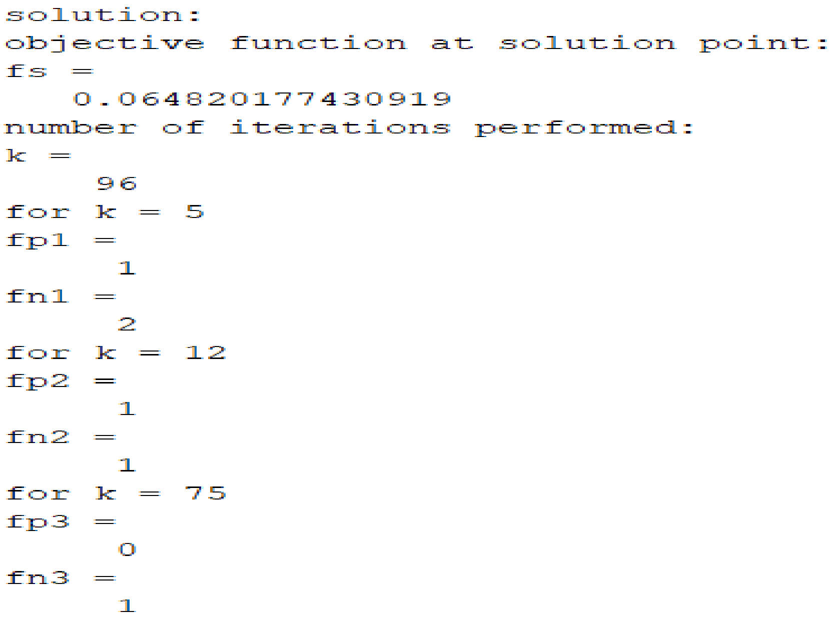
4.2 : Linear Regression (Newly Implemented Algorithm): This Linear Regression is a simple 2 class classification problem. As the dataset contains only 2 classes. The number of mis-classifications were: 3 out of 89. The error rate and accuracy rate values are: 3.3708% and 96.6292% respectively.



```
Number of mis-classifications are:

3 out of 89



The error rate is:

3.3708%



The accuracy rate is:

96.6292%
```

Fig – 2: Output for Linear Regression

<u>4.3 Logistic Regression (Newly Implemented Algorithm):</u> This Using this experiment I have tried to compare the best algorithm amongst PCA and Linear Regression.



```
solution:
objective function at solution point:
fs =
   0.064820177430919
number of iterations performed:
k =
    96
for k = 5
fp1 =
     1
fn1 =
     2
for k = 12
fp2 =
     1
fn2 =
     1
for k = 75
fp3 =
     0
fn3 =
     1
```

5. Conclusion

In this section, we will compare all the results obtained from both existing and newly implemented algorithms. Remember that our main aim of this project is to find which method is better and appropriate to use, in terms of accuracy (PCA or Linear Regression). Accordingly, we can give a generalized statement that the method which gives best possible accuracy all the time, is considered to be the most appropriate and best method to use.

Breast Cancer Resuls:

PCA :

- Misclassifications – 1/89
- Error Rate(%) – 1.1236
- Accuracy – 98.8764

Linear Regression :

- Misclassifications – 3/89
- Error Rate(%) – 3.3708
- Accuracy – 96.6292

Logistic Regression:

We analysed accuracy of logistic regression at steps those were given in procedure i.e. 5, 12, 75. In total we had 96 steps to form Ie-2 from epsilon. During training this model gave some interesting results as shown in below drawn table. Gradient Descent showed a false negative at later phases.

	False positive	False Negative
K=5	1	2
K = 12	1	1
K = 75	0	1

We see that in datasets, PCA is performing better than Linear Regression in terms of accuracy. PCA is giving better results and it is trustworthy. Hence we can say that PCA is a better technique than Linear Regression.

Appendix

I am including the codes for all the newly implemented algorithms. Since codes for existing algorithms are already presnt/done in the labs.

For Dataset – Breast Cancer Dataset:

1. PCA(Newly Implemented):

```
clc
clear all
close all

load D_bc_tr.mat
load D_bc_te.mat

B = [];
M = [];

for i = 1 : 480

if D_bc_tr(31,i) == -1

be = D_bc_tr(:,i);
B = [B be];

end

if D_bc_tr(31,i) == 1

ma = D_bc_tr(:,i);
M = [M ma];

end

end

X1 = B(1:30,:);
X2 = M(1:30,:);
y1 = [ones(300,1);-ones(180,1)];
y2 = [ones(180,1);-ones(300,1)];

X = [X1 X2];
X11_Ori = [X1 X2];
X11_Ori(31,:) = [ones(1,300) ones(1,180)];

X11 = X11_Ori';
w1_cap_star = pinv(X11)*y1;
```

```
w1 = w1_cap_star(1:30,:);
b1 = w1_cap_star(31,:);

X22_Ori = [X2 X1];
X22_Ori(31,:) = [ones(1,180) ones(1,300)];

X22 = X22_Ori';
w2_cap_star = pinv(X22)*y2;
w2 = w2_cap_star(1:30,:);
b2 = w2_cap_star(31,:);

W_star = [w1 w2];
b_star = [b1;b2];

Label = D_bc_te(31,:);

L1 = [];
L2 = [];
XTe1 = [];
XTe2 = [];

for i = 1 : 89

if Label(:,i) == -1

l2  = -1;
L2 = [L2 l2];
Xte = D_bc_te(1:30,i);
XTe1 = [XTe1 Xte];

end

if Label(:,i) == 1

l1  = 1;
L1 = [L1 l1];
Xte = D_bc_te(1:30,i);
XTe2 = [XTe2 Xte];

end

end

L = [L1 L2];
XTe = [XTe1 XTe2];

Xte = [XTe];
Label = [ones(1,57) ones(1,32)+1];
```

```
ek = zeros(2,89);
mis_class = 0;

for col = 1:89

f = W_star'*Xte(:,col)+b_star;
[maxvalue index] = max(f);
ek(index,col)  = 1;

if index ~= Label(col)

mis_class = mis_class +1;

else

end

end


ek;
mis_class;

error_rate = mis_class/89*100;
accuracy = 100 - error_rate;


fprintf('Number of mis-classifications are:\n\n%d out of 89\n', mis_class);
fprintf('\n\n\n\nThe error rate is:\n\n%2.4f%% \n', error_rate);
fprintf('\n\n\n\nThe accuracy rate is:\n\n%2.4f%% \n', accuracy);
```

2. Linear Regression(Newly Implemented):

```
clc
clear all
close all

load D_bc_tr.mat
load D_bc_te.mat
b = [];
a = [];

for i = 1 : 480

if D_bc_tr(31,i) == 1

be = D_bc_tr(:,i);
b = [b be];
```

```
end

if D_bc_tr(31,i) == -1

ma = D_bc_tr(:,i);
a = [a ma];

end

end

Xtr = [b(1:30,:) a(1:30,:)];

%% Select 40 Samples at a time.

SD = 48;
N = 8;
MU = [];
U = [];
T = [];

M1 = [];
U1 = [];
T1 = [];

M2 = [];
U2 = [];
T2 = [];

for j = 0 : 1

if j == 0

SD = 180;
X = Xtr(:,1:180);
mu = mean(X,2);
A = X - mu;
C = 1/SD*(A*A');

[uq, Sq] = eigs(C,N);
M1 = [M1 mu];
U1 = [U1 uq];
T1 = [T1 X];

end

if j == 1
```

```
SD = 300;
X = Xtr(:,181:end);
mu = mean(X,2);
A = X - mu;
C = 1/SD*(A*A');

[uq, Sq] = eigs(C,N);
M2 = [M2 mu];
U2 = [U2 uq];
T2 = [T2 X];

end
end

U = [U1 U2];
T = [T1 T2];
MU = [M1 M2];
D_test = D_bc_te(1:30,:);
t = cputime;
tic;
for j = 1:89
for i = 0:1

Uq = U(:,(i*N)+1:(i+1)*N);
f_j = Uq'*(D_test(:,j)-MU(:,i+1));
x_cap_j = Uq*f_j+MU(:,i+1);
e(i+1,j) = norm(D_test(:,j)-x_cap_j);

end
end

Label = D_bc_te(31,:);
Label_As_Per_Index = zeros(1,89);

for i = 1 : 89

if Label(:,i) == -1

Label_As_Per_Index(1,i) = 2;

end


if Label(:,i) == 1

Label_As_Per_Index(1,i) = 1;

end
end
```

```
Label = Label_As_Per_Index;

yz = toc;
[k,ind] = min(e);
t1 = cputime - t;
cmp = Label - (ind);

n = nnz(cmp);
error_rate = (n*100)/length(Label);
accuracy = 100 - error_rate;

fprintf('Number of mis-classifications are:\n\n%d out of 89\n', n)
fprintf('\n\n\n\nThe error rate is:\n\n%2.4f%% \n', error_rate)
fprintf('\n\n\n\nThe accuracy rate is:\n\n%2.4f%% \n', accuracy)
```

3. **Logistic Regression(My Lab - 4)**

```
load D_bc_te.mat
load D_bc_tr.mat

trn = D_bc_tr;
test = D_bc_te;

X_trn = trn(1:30, :);
Y_train = trn(31, :);

X_tst = test(1:30, :);
Y_tst = test(31, :);
for i = 1:30
 xi = X_trn(i,:);
 mi = mean(xi);
 vi = sqrt(var(xi));
 X_trn(i,:) = (xi - mi)/vi;
end
X_trn = [X_trn; ones(1, 480); Y_train];
for i = 1:30
 xi = X_tst(i,:);
 mi = mean(xi);
 vi = sqrt(var(xi));
 X_tst(i,:) = (xi - mi)/vi;
end
X_tst = [X_tst; ones(1, 89)];
w = zeros(31,1);
[xs,fs,k, xs1, xs2, xs3] = grad_desc('f_logistic', 'g_logistic', w, 1e-2, X_trn);
[fp1, fn1] = classify(X_tst, Y_tst, xs1);
[fp2, fn2] = classify(X_tst, Y_tst, xs2);
[fp3, fn3] = classify(X_tst, Y_tst, xs3);
```

```
disp('for k = 5');
fp1
fn1

disp('for k = 12');
fp2
fn2

disp('for k = 75');
fp3
fn3

function [fp, fn] = classify(X_test,Y_test, xs)
    result = zeros(89, 1);
    fp = 0;
    fn = 0;
    for i = 1:89
        y = xs' * X_test(:, i);
        if y > 0
            result(i) = 1;
        else
            result(i) = -1;
        end
        if result(i) == 1 && result(i) ~= Y_test(i)
            fn = fn + 1;
        end
        if result(i) == -1 && result(i) ~= Y_test(i)
            fp = fp + 1;
        end
    end
end
```

Refrences

[1] 'ECE-403/ 503-Course Notes', by Dr. Wu-Sheng Lu, 2019.

[2] 'ECE-403/ 503-Lab Manual', by Dr. Wu-Sheng Lu, 2019.

[3] 'Application of Principal Component Analysis in Data Classification', by W. Hernadez, 2003.

[4] UCI Machine Learning, http://archive.ics.uci.edu/ml, University of California Irvine, School of Information and Computer Science.

[5] ‘On Breast Cancer Detection: An Application of Machine Learning Algorithms’, Abien et al, 2019.

[6]W. N. Street, W. H. Wolberg, and O. L. Mangasarian, “Nuclear feature extraction for breast tumor diagnosis,” in IS?T/SPIE Int. Symp. Electronic Imaging: Science and Technology, vol. 1905, pp. 861-870, San Jose, CA., 1993.

[7] O. L. Mangasarian, W. N. Street, and W. H. Wolberg, “Breast cancer diagnosis and prognosis via linear programming,” AAAI Tech. Report SS-94-01, 1994.