ECE 175A
Zihe Deng
A14224990

HW 4

Part 1:

Given Y, the MLE of $\hat{x}$ is:

$$\hat{x} = \arg\min_{x} (y - ax)^T \Sigma^{-1} (y - ax) + N \log(2\pi|\Sigma|)$$
$$= \arg\min_{x} (y - ax)^T \Sigma^{-1} (y - ax)$$
$$= \arg\min_{x} (Sy - Sax)^T (Sy - Sax)$$
$$= \arg\min_{x} \| Sy - Sax \|^2$$

Note that: $S = \sqrt{\Sigma^{-1}}$

$$\hat{x} = \arg\max_{x} \| Sy - Sax \|^2 \quad ; \; V = Sy, \; M = Sa$$
$$\hat{x} = \arg\max_{x} \| V - Mx \|^2$$

take first order derivative: $\frac{\partial f}{\partial x} = 0$, when $M^T(V - Mx) = 0$

$$\hat{x} = (M^T M)^{-1} M^T V$$
$$= (a^T \Sigma^{-1} a)^{-1} \cdot a^T \Sigma^{-1} y$$

2nd order derivative: $\frac{\partial^2 f}{\partial x^2} > 0$ holds, we have a minimum which is maximum of the log-likelihood.

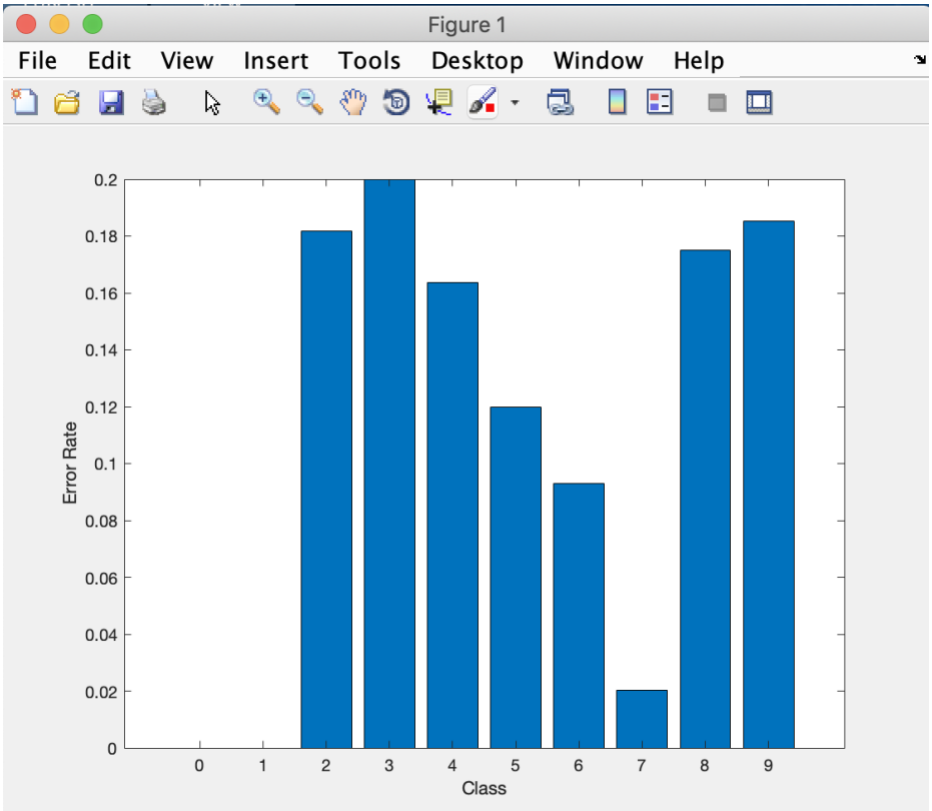
Same logic applys ($Y = a \cdot x$), substitude a with x: $\hat{a} = (x^T x)^{-1} \cdot (x^T \cdot y)$

The maximum likelihood estimation of a can be represented by: $(\boldsymbol{X^T} * \boldsymbol{Y}) * (\boldsymbol{X^T} * \boldsymbol{X})^{-1}$

a	0.6796

Part 2:

After normalized all test images by dividing **a**, we implement classification using least square distance metric, and we got the flowing plot, table, and total error rate:



T

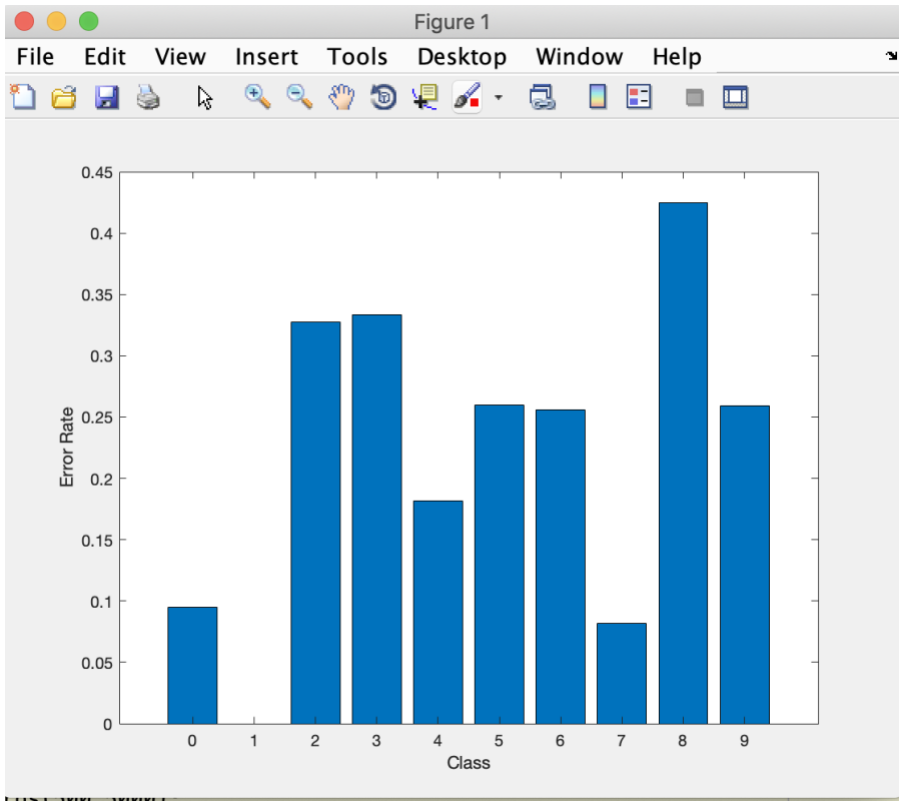
10x5 table

	1 Class	2 Total_num_of_images	3 Correctly_classified	4 Incorrectly_classified	5 Error_rate
1	'0'	42	42	0	0
2	'1'	67	67	0	0
3	'2'	55	45	10	0.1818
4	'3'	45	36	9	0.2000
5	'4'	55	46	9	0.1636
6	'5'	50	44	6	0.1200
7	'6'	43	39	4	0.0930
8	'7'	49	48	1	0.0204
9	'8'	40	33	7	0.1750
10	'9'	54	44	10	0.1852

total_error_rate 0.1120

Part 3:

Doing NN classifier without normalization process, we got:



T

10x5 table

	1 Class	2 Total_num_of_images	3 Correctly_classified	4 Incorrectly_classified	5 Error_rate
1	'0'	42	38	4	0.0952
2	'1'	67	67	0	0
3	'2'	55	37	18	0.3273
4	'3'	45	30	15	0.3333
5	'4'	55	45	10	0.1818
6	'5'	50	37	13	0.2600
7	'6'	43	32	11	0.2558
8	'7'	49	45	4	0.0816
9	'8'	40	23	17	0.4250
10	'9'	54	40	14	0.2593

total_error_rate 0.2120

Appendix

/Users/tylerdeng/Desktop/Matlab/ECE 175/hw4/MLE.m

```
%% (1)
clear all;
sampletrain = double(imread('sampletrain.png'));
sampletest = double(imread('sampletest.png'));


sampletest = sampletest(:);
sampletrain = sampletrain(:);

% Sample Amplifier:
a = (sampletrain.' * sampletest)/(sampletrain.' * sampletrain);

%% (2) classification using least square distance metric:
load 'data.mat';
%normalization:
imageTestNew = imageTestNew./a;

% Euclidean distance
EDistance = zeros(500,5000);
smallest = zeros(500,4);
for i = 1:500
    for j = 1:5000
        difference = (imageTestNew(:,:,i) - imageTrain(:,:,j)).^2;
        summation = sum(difference(:));
        EDistance(i,j) = sqrt(summation);
        if j==1
            smallest(i,1) = EDistance(i,1);
            smallest(i,2) = i;
            smallest(i,3) = j;
        end
        if EDistance(i,j)<smallest(i,1)
            smallest(i,1) = EDistance(i,j);
            smallest(i,2) = i;
            smallest(i,3) = j;
        end
    end
end
smallest(:,4) = labelTrain(smallest(:,3)); % columns in 'smallest': shortest_distance, test_idex, train

% total_error:
total_error = find(smallest(:,4)~=labelTestNew);
total_error_rate = size(total_error,1)/500;

error_num = zeros(10,1);
class_num = zeros(10,1);
 for i = 1:10
     error_num(i) = size(find(smallest(labelTestNew(:)==i-1,4)~=i-1),1);
     class_num(i) = size(find(labelTestNew(:)==i-1),1);
 end
error_rate_each_class = error_num./class_num;

% Plot:
index = (0:1:9);
bar(index,error_rate_each_class(:,1));
xlabel('Class');
ylabel('Error Rate');
%Error_Table:
Class = {'0';'1';'2';'3';'4';'5';'6';'7';'8';'9'};
Correctly_classified = class_num - error_num;
Incorrectly_classified = error_num;
Total_num_of_images = class_num;
Error_rate = error_rate_each_class(:,1);
T = table(Class,Total_num_of_images,Correctly_classified,Incorrectly_classified,Error_rate);
```

```matlab
%% (3) NN classifier without MLE
clear all;
load 'data.mat';
%normalization:
%imageTestNew = imageTestNew./a;
% Euclidean distance
EDistance = zeros(500,5000);
smallest = zeros(500,4);
for i = 1:500
    for j = 1:5000
        difference = (imageTestNew(:,:,i) - imageTrain(:,:,j)).^2;
        summation = sum(difference(:));
        EDistance(i,j) = sqrt(summation);
        if j==1
            smallest(i,1) = EDistance(i,1);
            smallest(i,2) = i;
            smallest(i,3) = j;
        end
        if EDistance(i,j)<smallest(i,1)
            smallest(i,1) = EDistance(i,j);
            smallest(i,2) = i;
            smallest(i,3) = j;
        end
    end
end
smallest(:,4) = labelTrain(smallest(:,3)); % columns in 'smallest': shortest_distance, test_idex, train

% total_error:
total_error = find(smallest(:,4)~=labelTestNew);
total_error_rate = size(total_error,1)/500;

error_num = zeros(10,1);
class_num = zeros(10,1);
 for i = 1:10
     error_num(i) = size(find(smallest(labelTestNew(:)==i-1,4)~=i-1),1);
     class_num(i) = size(find(labelTestNew(:)==i-1),1);
 end
error_rate_each_class = error_num./class_num;

% Plot:
index = (0:1:9);
bar(index,error_rate_each_class(:,1));
xlabel('Class');
ylabel('Error Rate');
%Error_Table:
Class = {'0';'1';'2';'3';'4';'5';'6';'7';'8';'9'};
Correctly_classified = class_num - error_num;
Incorrectly_classified = error_num;
Total_num_of_images = class_num;
Error_rate = error_rate_each_class(:,1);
T = table(Class,Total_num_of_images,Correctly_classified,Incorrectly_classified,Error_rate);
```

script Ln 76 Col 1