Fundamentals of Statistical Learning
Project Phase - 1

Upinderjit Singh
ASU ID: 1217042463
Master's in Computer Science
Arizona State University

1 Introduction

The phase-1 of the project is focused on multivariate density estimation using dimensionality reduction algorithm Principal Component Analysis (PCA) on a subset of images from the MNIST data. The density estimation is then used to doing minimum-error classification. The subset will only have images for digit "0" and digit "1". Below are the parts of this phase:

- Feature normalization.
- PCA using the training samples.
- Dimension reduction using PCA.
- Density estimation.
- Bayesian Decision Theory for optimal classification.

2 Summarization of the tasks

2.1 Feature normalization

The data is required to normalized using below formula:

$$y_{\mathrm{i}} = (x_{\mathrm{i}} - m_{\mathrm{i}})/s_{\mathrm{i}}$$

where, y_i - i^{th} normalized feature
x_i - i^{th} feature
m_i - i^{th} mean
m_i - i^{th} standard deviation

Normalised training data samples:

	0	1	2	3	4	5
0	-0.554279	0.271642	1.686317	-0.953446	1.439958	-0.937530
1	0.118035	-1.296516	1.841664	-0.967297	0.281496	0.565086
2	0.275565	-0.111632	-1.232155	-0.357323	0.273285	1.138107
3	-0.036853	-0.188358	-1.166385	0.316913	-0.948542	-1.228987
4	0.848663	-0.663974	-1.153143	-0.497127	-0.945557	0.098894
5	-0.872139	-0.557974	0.315410	-0.554405	3.266557	-0.702465
6	1.335613	-0.155619	-0.703976	-1.297396	1.248499	0.964522
7	-0.094783	0.375925	-1.174927	-0.751477	-1.278621	0.042849
8	0.045803	-0.285341	-0.690080	0.617221	-0.784177	1.121882
9	-0.678281	-0.795789	0.478442	0.428442	0.549620	0.370221
10	-0.075256	-0.884003	-0.580480	-0.824137	-0.534066	0.909913
11	-0.672231	-1.069213	0.633936	0.413441	-0.447727	2.092185
12	2.062093	1.985198	-0.900322	1.031916	2.166195	-1.285730

Figure 1: Normalised Training Data

Normalised testing data samples:

	0	1	2	3	4	5
0	0.154873	-0.773552	1.357609	-1.036275	-0.551014	-1.125258
1	-1.170483	-1.041824	-0.558667	-0.486641	-1.198400	-1.000783
2	-1.069378	-0.039907	-0.743704	-0.492287	-1.112618	-0.713358
3	-0.724609	-1.104978	-0.889706	-0.578002	-1.282929	-0.642399
4	-0.794437	-1.302783	-0.965014	-0.511059	-0.719124	0.021637
5	-1.089591	-0.703642	-0.212308	-0.931365	-0.413721	-1.319398
6	-0.961127	-0.682722	-0.093308	0.296927	0.061140	-1.053467
7	-0.163968	-0.626619	-0.833654	-1.020115	-0.645142	-0.475718
8	-1.169619	-0.542147	-1.265722	-0.505110	-1.278259	-1.060793
9	-0.199478	-0.519648	-0.291977	-0.427385	0.215175	-0.981369
10	-0.127342	-0.078070	-0.735466	-0.073250	-0.152556	-0.694538
11	-0.497972	-1.097920	-0.646006	-0.788406	-0.791152	-0.278181
12	-0.166423	-0.525642	-1.068605	-1.168987	0.193864	-0.483833

Figure 2: Normalised Testing Data

2.2 PCA using the training samples

The task included steps to compute co-variance matrix and doing eigen analysis and identifying the PCA components

- Co-variance matrix:

Covariance Matrix:

	0	1	2	3	4	5
0	1.000000	-0.001839	0.000205	-0.015194	0.009932	0.008065
1	-0.001839	1.000000	-0.007675	0.018140	-0.005898	-0.007214
2	0.000205	-0.007675	1.000000	-0.006408	0.000671	-0.000299
3	-0.015194	0.018140	-0.006408	1.000000	0.007503	0.009535
4	0.009932	-0.005898	0.000671	0.007503	1.000000	0.013734
5	0.008065	-0.007214	-0.000299	0.009535	0.013734	1.000000
6	0.004016	0.012288	-0.005989	0.001164	0.000206	0.004005
7	-0.006810	-0.003378	-0.013321	0.006490	0.002462	-0.004332
8	0.000543	-0.017595	0.008657	0.001230	-0.008750	0.004307
9	-0.002160	0.009828	0.004717	0.018549	-0.004297	-0.008647
10	0.004385	0.007985	0.011935	0.016190	0.007203	-0.006523
11	-0.005945	-0.000032	0.002835	-0.004479	-0.010221	-0.009122
12	-0.007706	0.002038	0.010427	0.005432	0.014125	0.013031

Figure 3: Covariance Matrix

- Eigen Values:

```
Variance on PCA 1: 99.46082192723023
Variance on PCA 2: 39.80304184115658
Variance on PCA 3: 26.210644896417836
Variance on PCA 4: 23.364092984184893
```

Figure 4: Eigen Values

- Eigen Vectors:

Eigen Vectors:

	0	1
0	0.000672	0.002183
1	0.000548	-0.001501
2	0.000655	0.000936
3	-0.000016	0.000695
4	0.000461	-0.000009

Figure 5: Eigen Vectors

2.3 Dimension reduction using PCA

Below are the representation of class distribution for training and testing data with the help of two principal components.

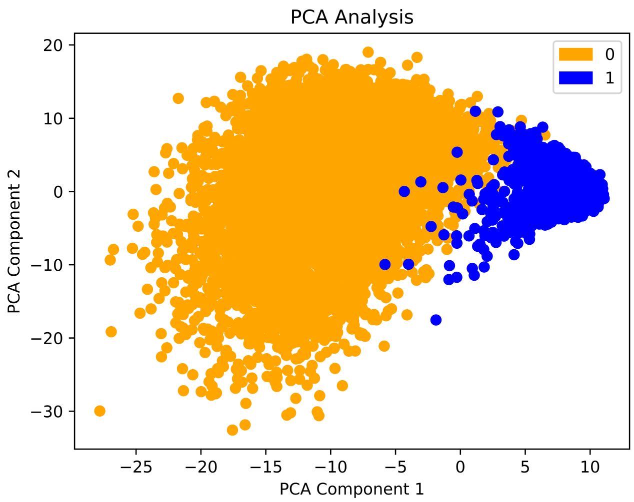

Figure 6: Class distribution for training data

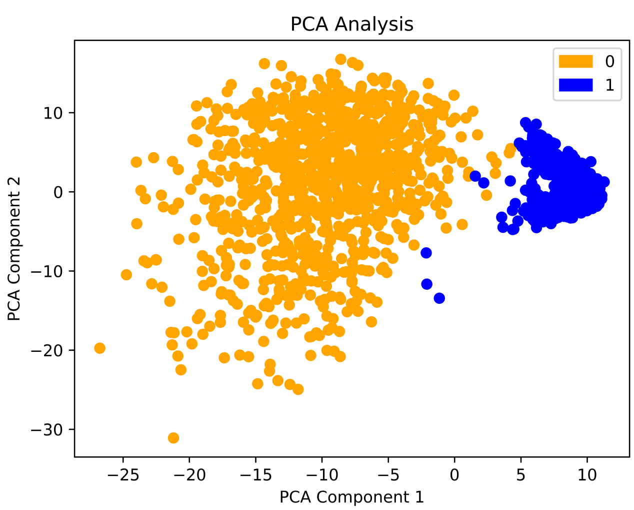

Figure 7: Class distribution for training data

2.4 Density estimation

For density estimation we needed to find mean and co-variance matrix for training data. These are then used to estimate the true probability density functions. Shown in figure 8,9.

2.5 Bayesian Decision Theory for optimal classification

Calculated the accuracy for training and testing data.

```
Means for both classes:
          0         1
-9.923454  8.717979
 0.851423 -0.747995
```

Figure 8: Mean for both classes

```
Co-variance for class 0:
            0          1
0  25.322606  15.898889
1  15.898889  79.106873

Co-variance for class 1:
           0         1
0  2.066604 -0.021484
1 -0.021484  4.083813
```

Figure 9: Co-variance matrix

```
(base) upsingh@DESKTOP-PMDRNGP:/mnt/u/
$ python project.py -train0 training0.

Accuracy on Training Data: 98.8788%
Accuracy on Testing Data: 99.1962%
```

Figure 10: Accuracy

3 Code

Code is available on below mentioned links and all the above result can be replicated using the instructions in README.md
https://github.com/Upinder3/CSE_569_FSL