CSCI 5521: Intro to Machine Learning (Spring 2020)

Homework 2 Solution

1. (**30 points**) In this problem, you will implement a program to fit two multivariate Gaussian distributions to the 2-class data and classify the test data by computing the log odds $log\frac{P(C_1|x)}{P(C_2|x)}$. The priors $P(C_1)$ and $P(C_2)$ should be estimated from the training data. Three pairs of training data and test data are given. The parameters μ_1, μ_2, $\mathbf{S_1}$ and $\mathbf{S_2}$, the mean and covariance for class 1 and class 2, are learned in the following three models for each training data and test data pair,

   - **Model 1**: Assume independent $\mathbf{S_1}$ and $\mathbf{S_2}$ (the discriminant function is as equation (5.17) in the textbook).
   - **Model 2**: Assume $\mathbf{S_1} = \mathbf{S_2}$. In other words, shared $\mathbf{S}$ between two classes (the discriminant function is as equation (5.22) in the textbook).
   - **Model 3**: Assume $\mathbf{S_1}$ and $\mathbf{S_2}$ are diagonal and the diagonal entries are identical within $\mathbf{S_1}$ and $\mathbf{S_2}$: $\mathbf{S_1} = diag(\sigma_1)$, $\mathbf{S_2} = diag(\sigma_2)$. (You need to derive the discriminant function yourself).

   (a) (**10 points**) Write the likelihood function and derive $\mathbf{S_1}$ and $\mathbf{S_2}$ by maximum likelihood estimation of model 2 and model 3.

   (b) (**10 points**) Your program should return and print out the learned parameters $P(C_1), P(C_2)$, μ_1 and μ_2 of each data pair to either terminal or PyCharm console. Your implementation of model 1 and model 2 should return and print out the learned parameters $\mathbf{S_1}, \mathbf{S_2}$. Your implementation of model 3 will return and print out σ_1 and σ_2.

   (c) (**10 points**) For each test set, print out the error rates of each model to either terminal or PyCharm console (three models per each test set). Match each data pair to one of the models and justify your answer. Also, explain the difference in your results in the report.

Answer:

The discriminant function is calculated as:
$g_i(x) = \log p(x|C_i) + \log P(C_i)$,
where $p(x|C_i) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp\Big[-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)\Big]$, and $P(C_i)$ is given.
The decision by log odds is done such that:
If $\log\frac{P(C_1|x^t)}{P(C_2|x^t)} = \log P(C_1|x^t) - \log P(C_2|x^t) = g_1(x^t) - g_2(x^t) > 0$, then we classify x^t as class 1, otherwise class 2.

*[Part a : diag], [Part b : shared], and [Part c : different] in the below answers.
Model 1: $S_i = \frac{\sum_t (x^t-m_i)(x^t-m_i)^T}{N_i}$

Model 2: $L(m_1, m_2, S|\mathcal{X}) = \prod_{t=1}^{N_1} p(x^t|m_1, S) \prod_{t=1}^{N_2} p(x^t|m_2, S)$
Given the likelihood for class 1 $L_1(m_1, S|\mathcal{X}_1) = \prod_{t=1}^{N_1} p(x^t|m_1, S)$:
Its log-likelihood is given by:
$\mathcal{L}_1(m_1, S|\mathcal{X}_1) = -\frac{N_1}{2}(\log(2\pi) + \log|S|) - \frac{1}{2}\sum_{t=1}^{N1}(x^t - m_1)^T S^{-1}(x^t - m_1)$
$\frac{\mathcal{L}_1(m_1,S|\mathcal{X})}{\partial S^{-1}} = \frac{N_1}{2}S - \frac{1}{2}\sum_{t=1}^{N1}(x^t - m_1)^T(x^t - m_1)$
Similarly:
$\frac{\mathcal{L}_2(m_2,S|\mathcal{X})}{\partial S^{-1}} = \frac{N_2}{2}S - \frac{1}{2}\sum_{t=1}^{N2}(x^t - m_2)^T(x^t - m_2)$
Finally, combining both:
$\frac{\mathcal{L}(m_1,m_2,S|\mathcal{X})}{\partial S^{-1}} = \frac{N_1}{2}S - \frac{1}{2}\sum_{t=1}^{N_1}(x^t - m_1)^T(x^t - m_1) + \frac{N_2}{2}S - \frac{1}{2}\sum_{t=1}^{N_2}(x^t - m_2)^T(x^t - m_2) = 0$
Which implies:
$S = \frac{N_1}{N_1+N_2}(\frac{\sum_{t=1}^{N_1}(x^t-m_1)^T(x^t-m_1)}{N_1}) + \frac{N_2}{N_1+N_2}(\frac{\sum_{t=1}^{N_2}(x^t-m_2)^T(x^t-m_2)}{N_2}) = P(C_1)S_1 + P(C_2)S_2$

Model 3: $L(m_1, \sigma_1|\mathcal{X}) = \prod_{t=1}^{N_1}\frac{1}{(2\pi)^{d/2}}\prod_{j=1}^{d}\frac{1}{\sigma_{1j}}\exp\Big[-\frac{1}{2}\sum_{j=1}^{d}\frac{(x_j^t-m_{1j})^2}{\sigma_{1j}^2}\Big]$
$\mathcal{L}(m_1, \sigma_1|\mathcal{X}) = -\frac{dN_1}{2}\log(2\pi) - N_1\sum_{j=1}^{d}\log\sigma_{1j} - \frac{1}{2}\sum_{t=1}^{N_1}\sum_{j=1}^{d}\frac{(x_j^t-m_{1j})^2}{\alpha_1}$
$\frac{\mathcal{L}(m_1,\sigma_1|\mathcal{X})}{\sigma_{1j}} = -\frac{N_1}{2\sigma_{1j}^2} + \frac{1}{2}\sum_{t=1}^{N_1}\frac{(x_j^t-m_{1j})^2}{\sigma_{1j}^4} = 0 \implies \sigma_{1j}^2 = \frac{1}{N_1}\sum_{t=1}^{N_1}(x_j^t - m_{1j})^2$
Similarly: $\sigma_2 = \frac{1}{N_2}\sum_{t=1}^{N_2}(x_j^t - m_{2j})^2$

Test error of model 1 for dataset 1: 0.3000
Test error of model 2 for dataset 1: 0.2450
Test error of model 3 for dataset 1: 0.2500

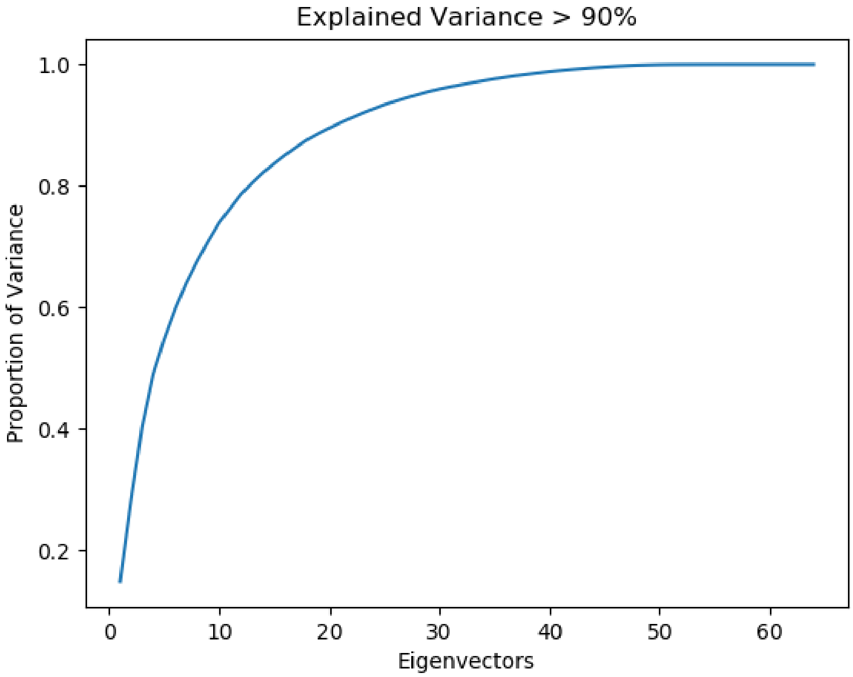
Test error of model 1 for dataset 2: 0.0450
Test error of model 2 for dataset 2: 0.2100
Test error of model 3 for dataset 2: 0.1450

Test error of model 1 for dataset 3: 0.2350
Test error of model 2 for dataset 3: 0.2550
Test error of model 3 for dataset 3: 0.2150

2. (a) Implement k-Nearest Neighbor (KNN) on the Optdigits dataset for $k = \{1, 3, 5, 7\}$. The error rates are contained in the following table:

	k=1	k=3	k=5	k=7
error rates	0.0539	0.0404	0.0438	0.0539

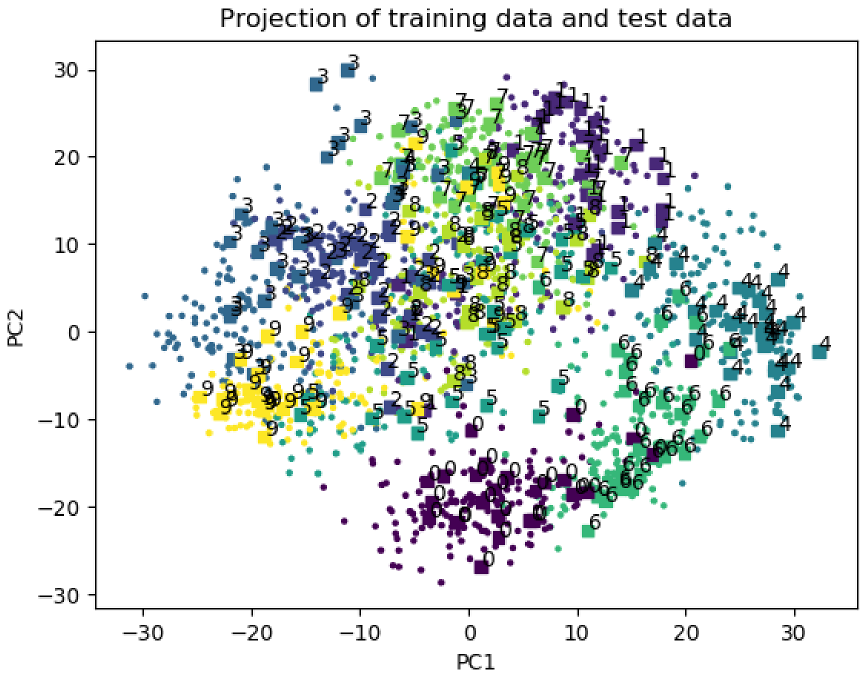
(b) The proportion of variance plot is as follows:



K=21 such that first K largest eigenvectors covers 90% of variance. And after we project data onto first 21 principal components, we can apply myKNN on projected data, and get

	k=1	k=3	k=5	k=7
error rates	0.0471	0.0471	0.0539	0.0539

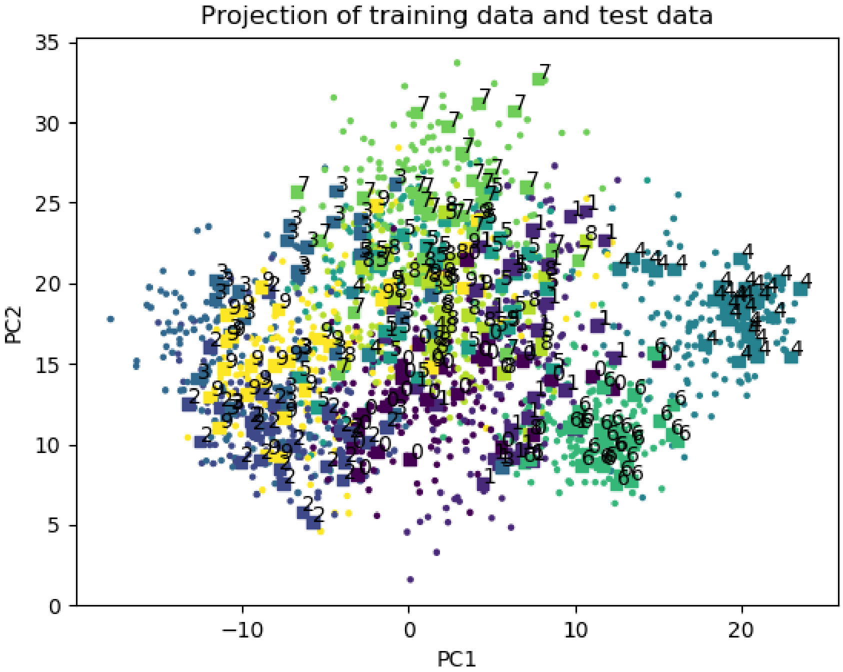
(c) Simply use parameter k=2 in myPCA and scatter plot all samples, with some of the samples labeled with different colors.



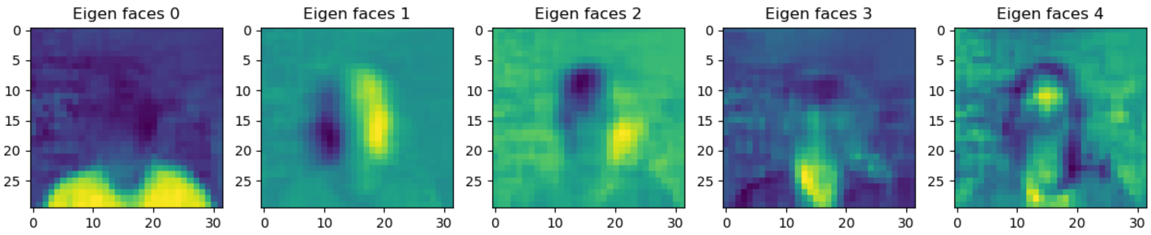
(d) After using LDA, we have the following results:

	k=1	k=3	k=5
error rates (L=2)	0.5791	0.5522	0.5354
error rates (L=4)	0.2795	0.2492	0.2391
error rates (L=9)	0.0976	0.1077	0.1010

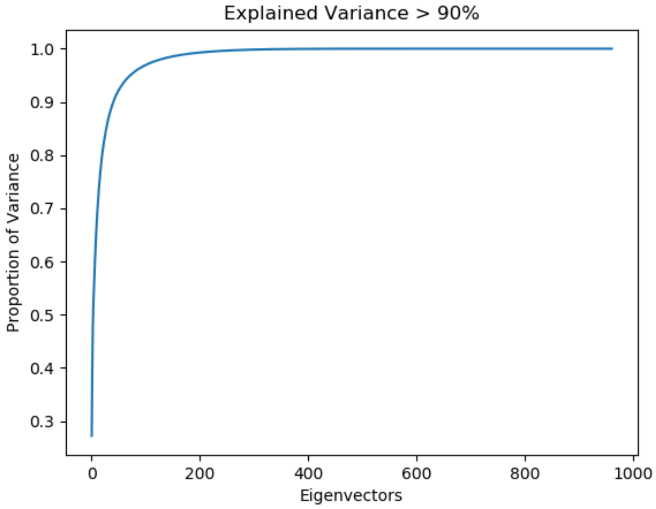
(e) Simply use parameter k=2 in myLDA and scatter plot all samples, with some of the samples labeled with different colors.


Projection of training data and test data
PC1
PC2
−10
0
10
20
0
5
10
15
20
25
30
35

3. (a) The 5 eigen faces obtained are as follows:



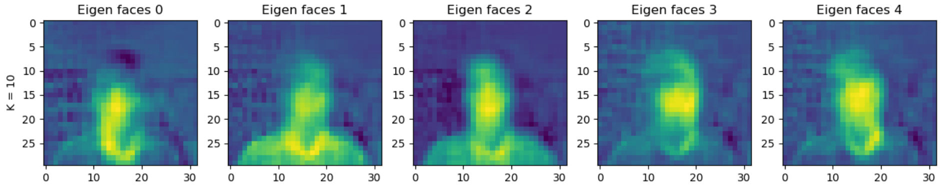
(b) The proportion of variance plot is as follows:

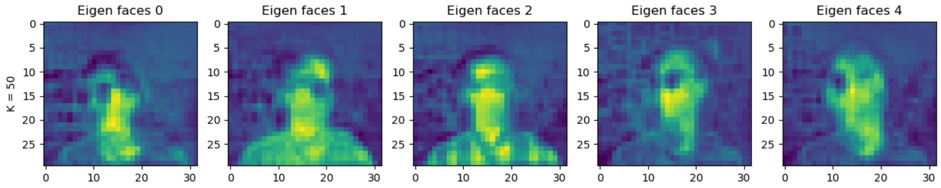


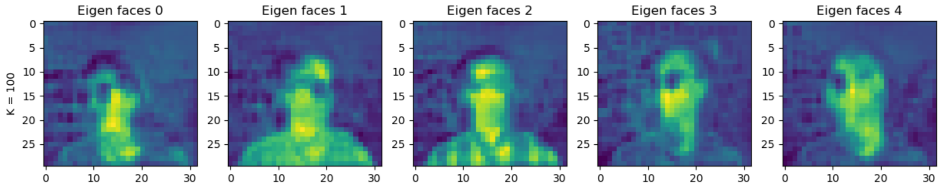
K=41 such that first K largest eigenvectors covers 90% of variance. And after we project data onto first 41 principal components, we can apply myKNN on projected data, and get

	k=1	k=3	k=5	k=7
error rates	0.1129	0.2339	0.4113	0.4355

(c) The number of principal components which were considered are 10,50 and 100. The following were the faces obtained.







Discussion:

- We see that the first-row images are the least clear and do not give much idea about the face of a person. This corresponds to the faces which were reconstructed using 10 principal components. Here we cannot distinguish a person as to wearing a sunglass or not.

- The second-row images correspond to the selection of number of principal components equal to 50. We see that to achieve 90% POV we need K = 41 components.

Hence, the clarity becomes better with 50 components. Here we can get an idea of the sunglasses but not sure yet.

- The third-row images provide with the best clarity and gives a good idea about the image and the contents of it as well. In this set of images, we get the best clarity and can easily identify if a person is wearing the sunglasses or not.