Imperial College
London

COURSEWORK 4

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

Mathematics for Machine Learning

Author:
Arnold Cheung (CID: 01184493)

Date: December 2, 2019

Part I:

The file PCA.m, wPCA.m and LDA.m have been completed and used to produce the plots of the recognition error versus the number of components kept for each of the methods. It can be seen that LDA is the best performing method out of the three, achieving the lowest error rate (≈ 0.25) with the smallest number of components kept (≈ 50). This is due to the methods are tested for its classification ability, and LDA is an algorithm designed to maximise the separation between the classes within the total dataset, while the other two methods focuses on maximising the variance in the latent space. LDA can also be understood as a supervised learning technique, while the other two are unsupervised, which is more well trained in classification with labels provided as training data.

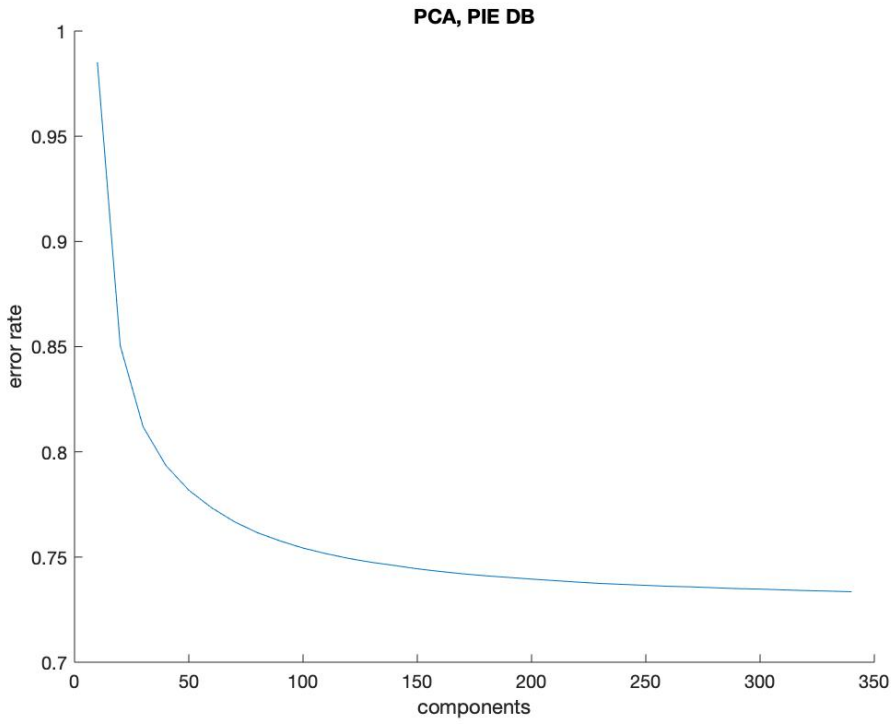

Figure 1: Recognition error versus the number of components kept for PCA

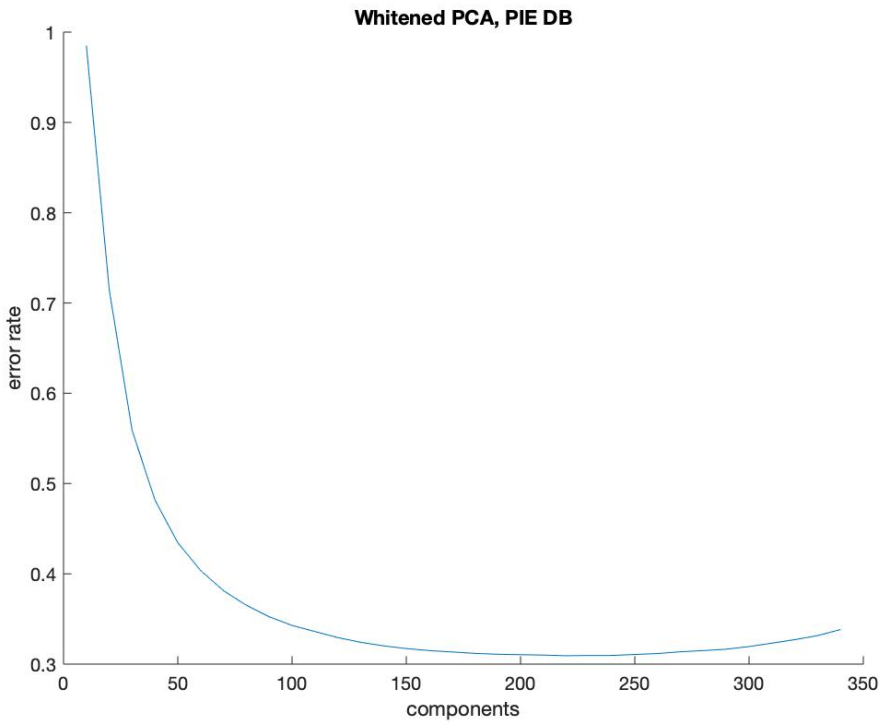

Figure 2: recognition error versus the number of components kept for wPCA

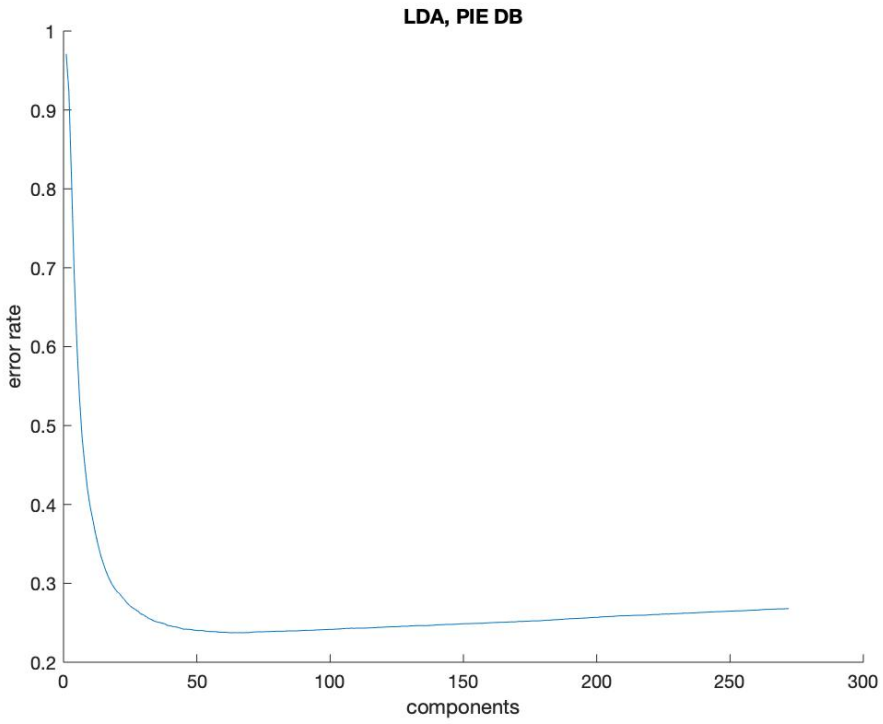

Figure 3: recognition error versus the number of components kept for LDA

Part II:

ia)

For the given optimisation problem, the Lagrangian is formulated as follows:

$$\mathscr{L}(\boldsymbol{w}, b, \xi_i, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\boldsymbol{w}^T \boldsymbol{S_t} \boldsymbol{w} + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i[y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^{N}\beta_i\xi_i \tag{1}$$

To minimise with respect to $\boldsymbol{w}, b$ and $\boldsymbol{\xi}$, set the derivative with respect to each variable and set to 0, the optimal $\boldsymbol{w}, b$ and $\boldsymbol{\xi}$ is achieved when the following conditions are satisfied:

$$\begin{aligned} \nabla_{\boldsymbol{w}}\mathscr{L} &= \boldsymbol{S_t}\boldsymbol{w} - \sum_{i=1}^{N}\alpha_i y_i \boldsymbol{x_i} = 0 \\ \Rightarrow \boldsymbol{w} &= \boldsymbol{S_t}^{-1}\sum_{i=1}^{N}\alpha_i y_i \boldsymbol{x_i} \end{aligned} \tag{2}$$

$$\begin{aligned} \frac{\partial\mathscr{L}}{\partial b} &= -\sum_{i=1}^{N}\alpha_i y_i = 0 \\ \Rightarrow \sum_{i=1}^{N}\alpha_i y_i &= 0 \end{aligned} \tag{3}$$

$$\begin{aligned} \nabla_{\boldsymbol{\xi}}\mathscr{L} &= C - \alpha_i - \beta_i = 0 \\ \Rightarrow \beta_i &= C - \alpha_i \end{aligned} \tag{4}$$

To formulate the dual optimisation problem (maximise with respect to α), apply the above conditions to the original Lagrangian:

$$\max_{\alpha} -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \boldsymbol{x_i}^T \boldsymbol{S_t}^{-1}\boldsymbol{x_j} + C\sum_{i=1}^{N}\xi_i + \sum_{i=1}^{N}\alpha_i - \sum_{i=1}^{N}\alpha_i\xi_i - \sum_{i=1}^{N}\beta_i\xi_i \tag{5}$$

Collecting the terms:

$$\max_{\alpha} -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \boldsymbol{x_i}^T \boldsymbol{S_t}^{-1}\boldsymbol{x_j} + \sum_{i=1}^{N}\alpha_i + \sum_{i=1}^{N}(C - \alpha_i - \beta_i)\xi_i \tag{6}$$

Since $\beta_i = C - \alpha_i$ (4), the last term becomes 0. Since β_i is also a Lagrangian multiplier and must be ≥ 0, $C \geq \alpha_i$ can be concluded, and constraints on the dual problem is therefore formulated:

$$\begin{aligned} \max_{\alpha} -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \boldsymbol{x_i}^T \boldsymbol{S_t}^{-1}\boldsymbol{x_j} + \sum_{i=1}^{N}\alpha_i \\ \text{s.t.} \qquad 0 \leq \alpha_i \leq C \\ \boldsymbol{\alpha}^T\boldsymbol{y} = 0 \end{aligned} \tag{7}$$

The problem can be simplified by changing the problem into a minimisation problem and defining $\boldsymbol{K_y} = [y_i y_j \boldsymbol{x_i}^T \boldsymbol{S_t}^{-1} \boldsymbol{x_j}]$:

$$\begin{aligned} \min_a \frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{K_y} \boldsymbol{\alpha} + \mathbf{1}^T \boldsymbol{\alpha} \\ \text{s.t.} \qquad 0 \leq \alpha_i \leq C \\ \boldsymbol{\alpha}^T \boldsymbol{y} = 0 \end{aligned} \tag{8}$$

After computing α, the optimal $\boldsymbol{w}*$ and $b*$ can be computed as follows:

$$\boldsymbol{w}^* = \boldsymbol{S_t}^{-1} \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x_i} \tag{9}$$

α_i is only greater than 0 for points $\boldsymbol{x_i}$ that lies on the margin, i.e. $\boldsymbol{w}^T \boldsymbol{x_i} + b = y_i$, y_i is know and can only take values from -1, 1, using the optimal $\boldsymbol{w}^*$, b^* can be recovered as follows:

$$\begin{aligned} b^* = y_i - \boldsymbol{w}^{*T} \boldsymbol{x_i} \\ \text{for } i \text{ where } \alpha_i > 0 \end{aligned} \tag{10}$$

The final b^* is the mean of all b retrieved to take into account numerical fluctuations.

ib)

The above optimisation problem has been solved numerically on using Matlab and the classification accuracy has been calculated to be 100% using the given train and test data. The code implementation and label prediction for each sample can be found in the in the file SVM.m.

ii)

$\boldsymbol{S_t}$ being singular can be dealt with by substituting $\boldsymbol{w} = \boldsymbol{U}\boldsymbol{Q}$ and choose $\boldsymbol{U}$ such that $\boldsymbol{U}^T \boldsymbol{S_t} \boldsymbol{U} = \boldsymbol{I}$. Which transforms the problem from:

$$\min \frac{1}{2}\boldsymbol{w}^T \boldsymbol{S_t} \boldsymbol{w} + C \sum_{i=1}^{N} \xi_i \tag{11}$$

into:

$$\min \frac{1}{2}\boldsymbol{Q}^T \boldsymbol{Q} + C \sum_{i=1}^{N} \xi_i \tag{12}$$

Which allowed the problem to be solved with respect to $\boldsymbol{Q}$ and avoiding the need to inverse $\boldsymbol{S_t}$. The desired $\boldsymbol{w}$ can be retrieved in the end by computing $\boldsymbol{w} = \boldsymbol{U}\boldsymbol{Q}$.

$\boldsymbol{U}$ can be chosen as follows:

$\boldsymbol{S_t} = \frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T$ is a symmetrical matrix and can therefore be diagonalised:

$$\boldsymbol{S_t} = \boldsymbol{P}^T\boldsymbol{\Lambda}\boldsymbol{P} \tag{13}$$

$$\boldsymbol{P}^T = \boldsymbol{P}^{-1} \tag{14}$$

$$\boldsymbol{U}^T\boldsymbol{S_t}\boldsymbol{U} = \boldsymbol{U}^T\boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^T\boldsymbol{U} = \boldsymbol{I} \tag{15}$$

Where $\boldsymbol{P}$ is the matrix whose columns are the eigenvectors of $\boldsymbol{S_t}$ and $\boldsymbol{\Lambda}$ is the diagonal matrix containing the eigenvalues of $\boldsymbol{S_t}$. The required condition is satisfied when $\boldsymbol{U}$ is chosen as:

$$\boldsymbol{U} = \boldsymbol{P}\boldsymbol{\Lambda}^{-\frac{1}{2}} \tag{16}$$

$$\boldsymbol{U}^T\boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^T\boldsymbol{U} = \boldsymbol{I}$$

$$(\boldsymbol{P}\boldsymbol{\Lambda}^{-\frac{1}{2}})^T\boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{\Lambda}^{-\frac{1}{2}} = \boldsymbol{I}$$

$$\boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{\Lambda}^{-\frac{1}{2}} = \boldsymbol{I}$$

$$\boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{\Lambda}\boldsymbol{\Lambda}^{-\frac{1}{2}} = \boldsymbol{I}$$

$$\boldsymbol{I} = \boldsymbol{I}$$

Therefore, by choosing $\boldsymbol{U} = \boldsymbol{P}\boldsymbol{\Lambda}^{-\frac{1}{2}}$ and substituting $\boldsymbol{w} = \boldsymbol{U}\boldsymbol{Q}$, the problem caused by the singularity of $\boldsymbol{S_t}$ can be avoided.