Imperial College London

COURSEWORK

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

CO496 - Mathematics for Inference and Machine Learning

Author:
Daren Sin (CID: ds2912)

Date: November 30, 2016

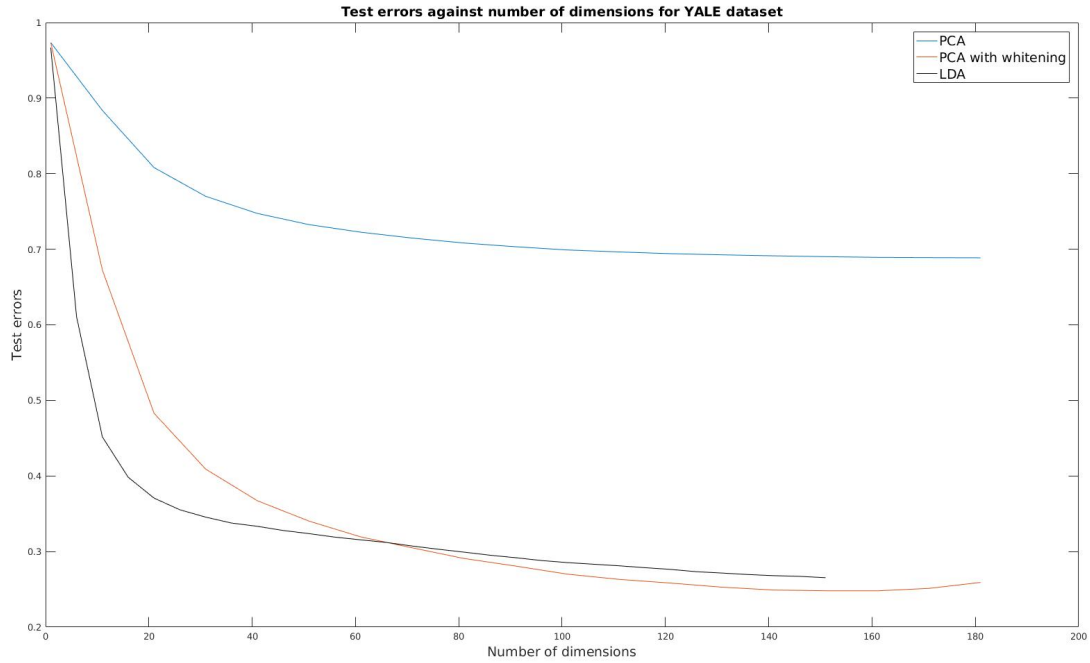

Figure 1: Graph of test errors against number of dimensions for the YALE dataset.

Part I

The plots for the YALE and PIE datasets can be found in Figure 1 and 2 respectively. First, for both datasets, LDA and PCA with whitening performed significantly better, with much lower test errors compared to the default PCA, across all dimensions.

Part II

1.) The Lagrangian is as follows:

$$L(R, \boldsymbol{a}, \xi_i, \lambda, r) = R^2 + C\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\lambda_i\left[(\boldsymbol{x}_i - \boldsymbol{a})^\top(\boldsymbol{x}_i - \boldsymbol{a}) - R^2 - \xi_i\right] - \sum_{i=1}^{n} r_i\xi_i \quad (1)$$

To find the dual, we need to minimise the Lagrangian with respect to R, $\boldsymbol{a}$, and ξ_i. We thus differentiate the Lagrangian with respect to these variables:

$$\frac{\partial L}{\partial R} = 2R - 2R\sum_{i=1}^{n}\lambda_i \overset{!}{=} 0 \quad \Rightarrow \quad 1 - \sum_{i=1}^{n}\lambda_i = 0 \quad \Rightarrow \quad \sum_{i=1}^{n}\lambda_i = 1 \quad (2)$$

$$\frac{\partial L}{\partial \boldsymbol{a}} = \sum_{i=1}^{n}\lambda_i\left[-2(\boldsymbol{x}_i - \boldsymbol{a})\right] \overset{!}{=} 0 \quad \Rightarrow \quad \sum_{i=1}^{n}\lambda_i(\boldsymbol{x}_i - \boldsymbol{a}) = 0 \quad (3)$$

$$\frac{\partial L}{\partial \xi_i} = C - \lambda_i - r_i \overset{!}{=} 0 \quad \Rightarrow \quad C = \lambda_i + r_i \quad (4)$$

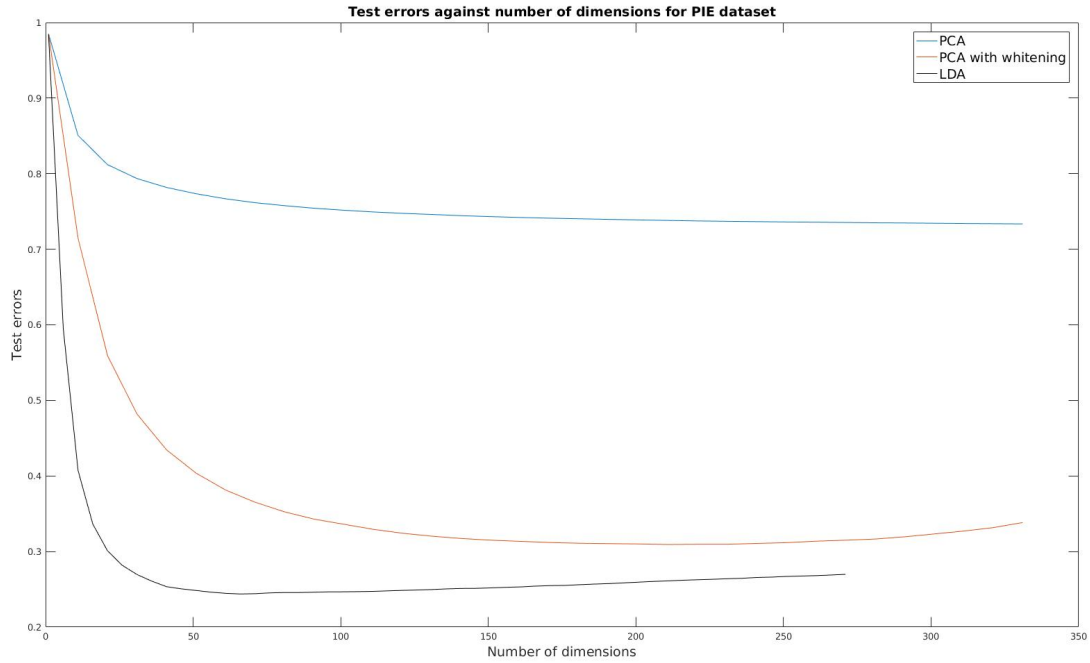

Figure 2: Graph of test errors against number of dimensions for the PIE dataset.

Furthermore, from equation 3, we also have the result:

$$\sum_{i=1}^{n}(\lambda_i \boldsymbol{x}_i - \boldsymbol{a}\lambda_i) = 0 \quad \Rightarrow \quad \sum_{i=1}^{n}(\lambda_i \boldsymbol{x}_i) - \boldsymbol{a} = 0 \quad \Rightarrow \quad \boldsymbol{a} = \sum_{i=1}^{n}\lambda_i \boldsymbol{x}_i \tag{5}$$

using the result that $\sum_{i=1}^{n}\lambda_i = 1$ from above. So, using the above equations, we can express the Lagrangian as such:

$$L(\lambda) = R^2 + \sum_{i=1}^{n}(\lambda_i + r_i)\xi_i + \lambda_i \sum_{i=1}^{n}(\boldsymbol{x}_i - \boldsymbol{a})^\top(\boldsymbol{x}_i - \boldsymbol{a}) - R^2\sum_{i=1}^{n}\lambda_i - \sum_{i=1}^{n}\lambda_i\xi_i - \sum_{i=1}^{n}r_i\xi_i$$

(6)

$$= R^2 + \sum_{i=1}^{n}(\lambda_i + r_i)\xi_i + \sum_{i=1}^{n}\left[\lambda_i(\boldsymbol{x}_i - \boldsymbol{a})^\top(\boldsymbol{x}_i - \boldsymbol{a})\right] - R^2 - \sum_{i=1}^{n}(\lambda_i + r_i)\xi_i \tag{7}$$

$$= \sum_{i=1}^{n}\left[\lambda_i(\boldsymbol{x}_i^\top - \boldsymbol{a}^\top)(\boldsymbol{x}_i - \boldsymbol{a})\right] \tag{8}$$

$$= \sum_{i=1}^{n}\lambda_i\left[\boldsymbol{x}_i^\top\boldsymbol{x}_i - \boldsymbol{x}_i^\top\boldsymbol{a} - \boldsymbol{a}^\top\boldsymbol{x}_i + \boldsymbol{a}^\top\boldsymbol{a}\right] \tag{9}$$

$$= \sum_{i=1}^{n}\left[\lambda_i\boldsymbol{x}_i^\top\boldsymbol{x}_i - \lambda_i\boldsymbol{x}_i^\top\boldsymbol{a} - \lambda_i\boldsymbol{a}^\top\boldsymbol{x}_i + \lambda_i\boldsymbol{a}^\top\boldsymbol{a}\right] \tag{10}$$

$$= \sum_{i=1}^{n} \left[\lambda_i \boldsymbol{x}_i^\top \boldsymbol{x}_i - \lambda_i \boldsymbol{x}_i^\top \boldsymbol{a} - \lambda_i \boldsymbol{a}^\top (\boldsymbol{x}_i - \boldsymbol{a}) \right] \tag{11}$$

$$= \sum_{i=1}^{n} \left[\lambda_i \boldsymbol{x}_i^\top \boldsymbol{x}_i - \lambda_i \boldsymbol{x}_i^\top \boldsymbol{a} \right] - \boldsymbol{a}^\top \underbrace{\sum_{i=1}^{n} \lambda_i (\boldsymbol{x}_i - \boldsymbol{a})}_{=0, \text{ from equation 3}} \tag{12}$$

We also note that, from equation 5, we have

$$\boldsymbol{a} = \sum_{j=1}^{n} \lambda_j \boldsymbol{x}_j$$

So,

$$L(\lambda) = \sum_{i=1}^{n} \left[\lambda_i \boldsymbol{x}_i^\top \boldsymbol{x}_i - \lambda_i \boldsymbol{x}_i^\top \boldsymbol{a} \right] \tag{13}$$

$$= \sum_{i=1}^{n} (\lambda_i \boldsymbol{x}_i^\top \boldsymbol{x}_i) - \sum_{i=1}^{n} \lambda_i \boldsymbol{x}_i^\top \left(\sum_{j=1}^{n} \lambda_j \boldsymbol{x}_j \right) \tag{14}$$

$$= \sum_{i=1}^{n} (\lambda_i \boldsymbol{x}_i^\top \boldsymbol{x}_i) - \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \boldsymbol{x}_i^\top \boldsymbol{x}_j \lambda_j \tag{15}$$

$$= \text{diag}(\boldsymbol{X}^\top \boldsymbol{X})^\top \lambda - \lambda^\top (\boldsymbol{X}^\top \boldsymbol{X}) \lambda \tag{16}$$

$$= \text{diag}(\boldsymbol{K}_x)^\top \boldsymbol{\lambda} - \boldsymbol{\lambda}^\top (\boldsymbol{K}_x) \boldsymbol{\lambda} \tag{17}$$

where $\boldsymbol{K}_x = [\boldsymbol{x}_i^\top \boldsymbol{x}_j]$. Hence, we can write the dual as:

$$\begin{aligned} \max_{\boldsymbol{\lambda}} \quad & \text{diag}(\boldsymbol{K}_x)^\top \boldsymbol{\lambda} - \boldsymbol{\lambda}^\top (\boldsymbol{K}_x) \boldsymbol{\lambda} \\ \text{subject to} \quad & \lambda_i \geq 0, \quad 0 \leq \lambda_i \leq C, \quad \text{for } i = 1, \ldots, n \\ & \sum_{i=1}^{n} \lambda_i = 1 \quad \Rightarrow \quad \mathbf{1}^\top \boldsymbol{\lambda} = 1 \end{aligned}$$

The second constraint, $0 \leq \lambda_i \leq C$, for $i = 1, \ldots, n$ can be derived from equation 4, which implies that $\lambda_i = C - r_i$. Since $r_i \geq 0$ and $\lambda_i \geq 0$, r_i can only take values from 0 to C, inclusive. We then have the constraint, $0 \leq \lambda_i \leq C$.

Furthermore, we can write the above optimisation problem as its equivalent minimisation problem:

$$\begin{aligned} -\min_{\boldsymbol{\lambda}} \quad & -\text{diag}(\boldsymbol{K}_x)^\top \boldsymbol{\lambda} + \boldsymbol{\lambda}^\top (\boldsymbol{K}_x) \boldsymbol{\lambda} \\ \text{subject to} \quad & 0 \leq \lambda_i \leq C, \quad \text{for } i = 1, \ldots, n \\ & \mathbf{1}^\top \boldsymbol{\lambda} = 1 \end{aligned}$$

2.) When using arbitrary positive definite kernels, we can write the above optimisation problem as the following:

$$\begin{aligned} -\min_{\boldsymbol{\lambda}} \quad & -\operatorname{diag}(\boldsymbol{K}_x)^\top \boldsymbol{\lambda} + \boldsymbol{\lambda}^\top (\boldsymbol{K}_x)\boldsymbol{\lambda} \\ \text{subject to} \quad & 0 \leq \lambda_i \leq C, \quad \text{for } i = 1, \ldots, n \\ & \mathbf{1}^\top \boldsymbol{\lambda} = 1 \end{aligned}$$

where $\boldsymbol{K}_x = [k(\boldsymbol{x}_i, \boldsymbol{x}_j)]$ and $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j)$ is the kernel.

3.) After identifying the parameters to the MATLAB function `quadprog` (see MATLAB code), we get an optimal $\boldsymbol{\lambda}^*$ per class, output by the function. We can then substitute the $\boldsymbol{\lambda}^*$ back into the objective function for the dual, to get, say, $-d^*$:

$$-d^* = -\operatorname{diag}(\boldsymbol{K}_x)^\top \boldsymbol{\lambda}^* + (\lambda^*)^\top (\boldsymbol{K}_x)\boldsymbol{\lambda}^*$$

Furthermore, the optimal solution to the primal, p^*, is equal to the optimal solution to the dual, $-d^*$, where the negative sign arises from converting our maximisation problem to a minimisation one. We also note that, at the optimal solution, the slack variables, ξ_i, go to 0. Hence, keeping $\boldsymbol{a}$ constant, p^* represents the optimal solution to the minimisation problem:

$$R^2 + C \sum_{i=1}^{n} \xi_i$$

As $\xi_i = 0$ for all $i = 1, \ldots, n$, the optimal p^* is simply the optimal R^2. We can thus find the radius of the optimal enclosing hypersphere by taking the square root of $-d^*$.

We can also find the vector $\boldsymbol{a}_k \in \mathbb{R}^2$, which represents the center of the optimal enclosing hypersphere of class k by:

$$\boldsymbol{a}_k = \sum_{j=1}^{n} \lambda_j^* \boldsymbol{x}_{j,k}$$

where $\boldsymbol{x}_{j,k} \in \mathbb{R}^2$ is the jth data point of class k. The plot of the optimal hyperspheres is in Figure 3.

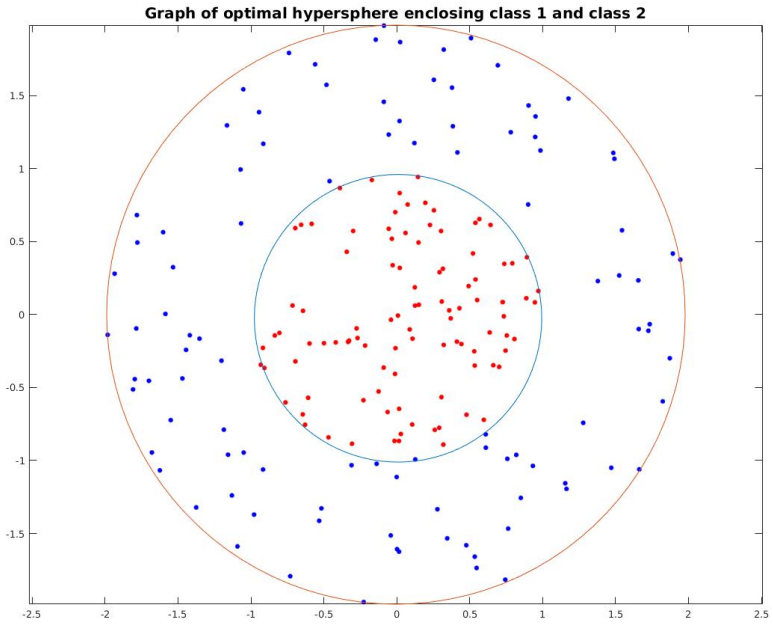

Figure 3: Plot of data points and the optimal enclosing hyperspheres for each class.