Mathematics for Machine Learning — Coursework 4

Balint Rikker

November 29, 2017

Part I

See fig. 1 for PCA, fig. 2 for whitened PCA, and fig. 3 for LDA.

Overall LDA seem to perform the best for Small Sample Size problems (error rate is less than 30% for number of components between 20 and 50), but it requires labels being available. For unsupervised problems, whitened PCA seem to perform a lot better than PCA. Error rates are mostly below 40% for wPCA with number of components less than 40, whereas PCA for the same range performs badly, with over 80-90% error rates. PCA's error rate only approaches 75% for a large number of components.

Figure 1: PCA recognition error vs number of components kept

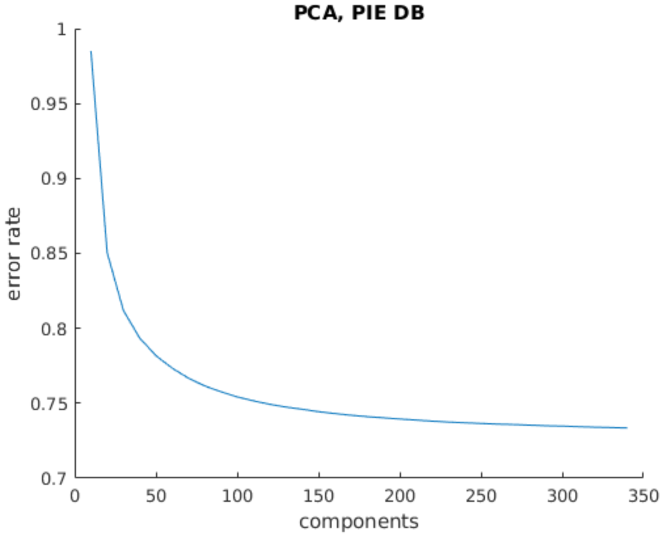


Figure 2: Whitened PCA recognition error vs number of components kept

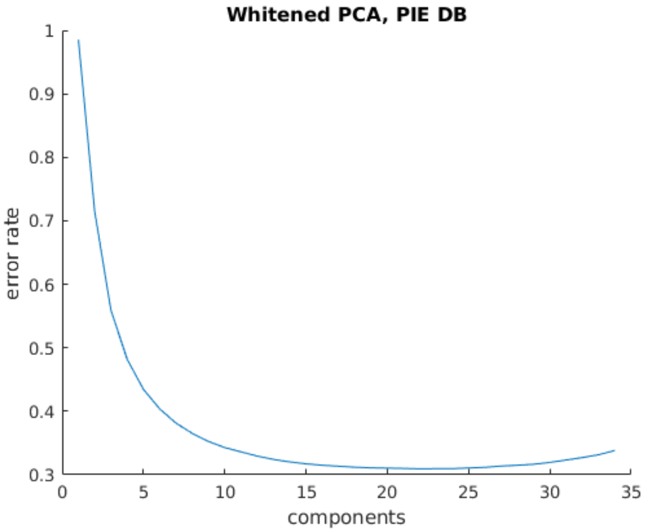
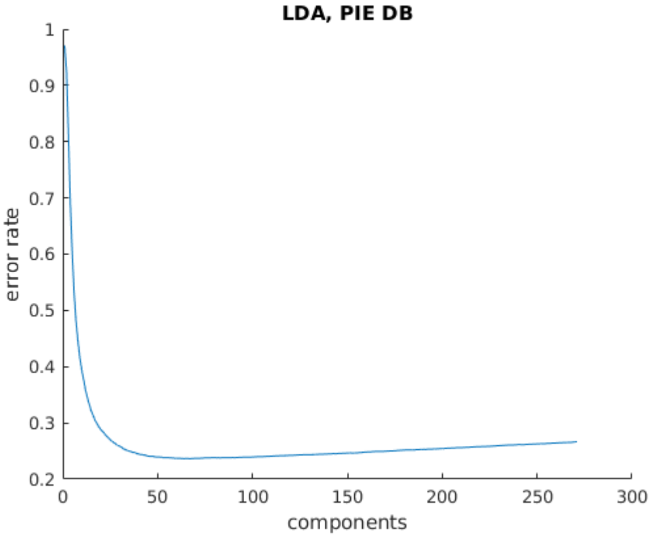


Figure 3: LDA recognition error vs number of components kept



Part II. i.

The Lagrangian of the optimization problem can be stated as follows (where $a_i \geq 0$ and $r_i \geq 0$ are Lagrangian multipliers):

$$L(\mathbf{w}, b, \xi_i, a_i, r_i) = \frac{1}{2}\mathbf{w}^T\mathbf{S}_t\mathbf{w} + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}a_i(y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 + \xi_i) - \sum_{i=1}^{n}r_i\xi_i \tag{1}$$

The dual problem to optimize is then:

$$\max_{a_i > 0}\min_{\mathbf{w}, b, \xi_i} L(\mathbf{w}, b, \xi_i, a_i, r_i)$$

The optimal $\mathbf{w}$, b and ξ_i will satisfy the condition that the partial derivatives with regards to these parameters will be 0.

$$\frac{\partial L}{\partial \mathbf{w}} = \frac{1}{2}(\mathbf{S}_t + \mathbf{S}_t^T)\mathbf{w} - \sum_{i=1}^{n}a_iy_i\mathbf{x}_i = \mathbf{S}_t\mathbf{w} - \sum_{i=1}^{n}a_iy_i\mathbf{x}_i = 0 \Leftrightarrow$$

$$\mathbf{S}_t\mathbf{w} = \sum_{i=1}^{n}a_iy_i\mathbf{x}_i \Leftrightarrow$$

$$\mathbf{w} = \mathbf{S}_t^{-1}\sum_{i=1}^{n}a_iy_i\mathbf{x}_i$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{n}a_iy_i = 0 \Leftrightarrow$$

$$\mathbf{a}^T\mathbf{y} = 0$$

$$\frac{\partial L}{\partial \xi_i} = C - a_i - r_i = 0 \Leftrightarrow$$

$$r_i + a_i = C$$

We can substitute these values back to (1):

$$L(\mathbf{a}) = \frac{1}{2}(\mathbf{S}_t^{-1}\sum_{i=1}^{n}a_iy_i\mathbf{x}_i)^T\mathbf{S}_t(\mathbf{S}_t^{-1}\sum_{i=1}^{n}a_iy_i\mathbf{x}_i) + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}a_i(y_i((\mathbf{S}_t^{-1}\sum_{i=1}^{n}a_iy_i\mathbf{x}_i^T)\mathbf{x}_i + b) - 1 + \xi_i) - \sum_{i=1}^{n}r_i\xi_i$$

$$L(\mathbf{a}) = \frac{1}{2}(\mathbf{S}_t^{-1}\sum_{i=1}^{n}a_iy_i\mathbf{x}_i)^T\sum_{i=1}^{n}a_iy_i\mathbf{x}_i + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}a_i(y_i((\mathbf{S}_t^{-1}\sum_{i=1}^{n}a_iy_i\mathbf{x}_i)^T\mathbf{x}_i + b)) + \sum_{i=1}^{n}a_i - \sum_{i=1}^{n}a_i\xi_i - \sum_{i=1}^{n}r_i\xi_i$$

$$L(\mathbf{a}) = \frac{1}{2}(\mathbf{S}_t^{-1}\sum_{i=1}^{n}a_iy_i\mathbf{x}_i)^T\sum_{i=1}^{n}a_iy_i\mathbf{x}_i + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}a_i(y_i((\mathbf{S}_t^{-1}\sum_{i=1}^{n}a_iy_i\mathbf{x}_i)^T\mathbf{x}_i + b)) + \sum_{i=1}^{n}a_i - \sum_{i=1}^{n}a_i\xi_i - \sum_{i=1}^{n}r_i\xi_i$$

$$L(\mathbf{a}) = \frac{1}{2}(\mathbf{S}_t^{-1}\sum_{i=1}^{n} a_i y_i \mathbf{x}_i)^T \sum_{i=1}^{n} a_i y_i \mathbf{x}_i - \sum_{i=1}^{n} a_i (y_i((\mathbf{S}_t^{-1}\sum_{i=1}^{n} a_i y_i \mathbf{x}_i)^T \mathbf{x}_i + b)) + \sum_{i=1}^{n} a_i + C\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \xi_i (a_i + r_i)$$

$$L(\mathbf{a}) = \frac{1}{2}(\mathbf{S}_t^{-1}\sum_{i=1}^{n} a_i y_i \mathbf{x}_i)^T \sum_{i=1}^{n} a_i y_i \mathbf{x}_i - \sum_{i=1}^{n} a_i (y_i((\mathbf{S}_t^{-1}\sum_{i=1}^{n} a_i y_i \mathbf{x}_i)^T \mathbf{x}_i + b)) + \sum_{i=1}^{n} a_i + C\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \xi_i C$$

$$L(\mathbf{a}) = \frac{1}{2}(\mathbf{S}_t^{-1}\sum_{i=1}^{n} a_i y_i \mathbf{x}_i)^T \sum_{i=1}^{n} a_i y_i \mathbf{x}_i - \sum_{i=1}^{n} a_i (y_i((\mathbf{S}_t^{-1}\sum_{i=1}^{n} a_i y_i \mathbf{x}_i)^T \mathbf{x}_i + b)) + \sum_{i=1}^{n} a_i$$

$$L(\mathbf{a}) = \frac{1}{2}(\mathbf{S}_t^{-1}\sum_{i=1}^{n} a_i y_i \mathbf{x}_i)^T \sum_{i=1}^{n} a_i y_i \mathbf{x}_i - \sum_{i=1}^{n} a_i y_i ((\mathbf{S}_t^{-1}\sum_{i=1}^{n} a_i y_i \mathbf{x}_i)^T \mathbf{x}_i) + \sum_{i=1}^{n} a_i y_i b + \sum_{i=1}^{n} a_i$$

$$L(\mathbf{a}) = \frac{1}{2}(\mathbf{S}_t^{-1}\sum_{i=1}^{n} a_i y_i \mathbf{x}_i)^T \sum_{i=1}^{n} a_i y_i \mathbf{x}_i - \sum_{i=1}^{n} a_i y_i ((\mathbf{S}_t^{-1}\sum_{i=1}^{n} a_i y_i \mathbf{x}_i)^T \mathbf{x}_i) + \sum_{i=1}^{n} a_i$$

$$L(\mathbf{a}) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_i y_i \mathbf{x}_i^T \mathbf{S}_t^{-1} a_j y_j \mathbf{x}_j + \sum_{i=1}^{n} a_i$$

This dual problem can be written as:

$$L(\mathbf{a}) = \mathbf{1}^T\mathbf{a} - \frac{1}{2}\mathbf{a}K_y\mathbf{a}^T$$

subject to $C \geq a_i \geq 0, i = 1..n, \mathbf{a}^T\mathbf{y} = 0$, with $\mathbf{K}_y = y_i \mathbf{x}_i^T \mathbf{S}_t^{-1} y_j \mathbf{x}_j$.

This problem can now be solved in Matlab using `quadprog`. The code solving the problem and calculating accuracy on the test set can be found in the file `SVM.m`.

The accuracy on the test set is 1.00.

Part II. ii.

Singular $\mathbf{S}_t$ arises when there is a linear interdependence between the variables, or geometrically, the number of dimensions is smaller than the number of samples. We can perform some sort of dimensionality reduction, e.g. PCA on our input dataset before applying the SVM algorithm. By keeping the number of dimensions equal to the number of non-zero eigenvalues of our covariance matrix, we can be sure that the covariance matrix of our resulting dataset will be invertible.