Lecture Summary

Linear Discriminant Analysis

Siddat Nesar, Kaveen Liyanage

Linear discriminant analysis (LDA) is a supervised dimensionality technique. Which takes into account class label information to find a vector w that maximizes the separability of the classes when projected. In this lecture, we will be covering the basic mathematical foundation and the algorithmic implementation of the LDA.

Let, given dataset $\mathbf{D} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \dots \mathbf{x}_n\}$, where $\mathbf{x}_i \in \mathbf{R}^d$, and class label, $\boldsymbol{y}_i \in \{\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3 \dots \mathbf{c}_k\}$

Let $\mathbf{D}_i$ denotes the subset of points labelled with class $\boldsymbol{c}_i$, i.e. $\mathbf{D}_i = \{\mathbf{x}_j^T \mid \boldsymbol{y}_j = \boldsymbol{c}_i\}$

Let $|\mathbf{D}_i| = \boldsymbol{n}_i$ denotes the number of points with class $\boldsymbol{c}_i$

We assume there are two classes (k = 2), thus **D** is divided into $\mathbf{D}_1$ and $\mathbf{D}_2$

Let **w** is a unit vector, $\mathbf{w}^T\mathbf{w} = \mathbf{1}$ [**w** is normal]

The projection of d-dimensional point $\mathbf{x}_i$ onto **w** is: $\mathbf{x}_i' = \left(\frac{\mathbf{w}^T\mathbf{x}_i}{\mathbf{w}^T\mathbf{w}}\right)\mathbf{w} = \left(\mathbf{w}^T\mathbf{x}_i\right)\mathbf{w} = a_i\mathbf{w}$

a_i is the scalar projection of $\mathbf{x}_i$ on **w** $a_i = \mathbf{w}^T\mathbf{x}_i$

a_i is the projected point. Thus n projected points is: $\{a_1, a_2, a_3 \dots a_n\}$ maps from $\mathbf{R}^d$ to **R**

Hence, d-dimensional space is converted to 1-dimensional space!

The projected mean: $m_1 = \frac{1}{n_1}\sum_{\mathbf{x}_i \in \mathbf{D}_1} a_i = \frac{1}{n_1}\sum_{\mathbf{x}_i \in \mathbf{D}_1} \mathbf{w}^T\mathbf{x}_i = \mathbf{w}^T\left(\frac{1}{n_1}\sum_{\mathbf{x}_i \in \mathbf{D}_1} \mathbf{x}_i\right) = \mathbf{w}^T\boldsymbol{\mu}_1$

Here, $\boldsymbol{\mu}_1$is the mean of all points in $\mathbf{D}_1$. Similarly, $m_2 = \mathbf{w}^T\boldsymbol{\mu}_2$

The goal of LDA is to maximize the separation between the clusters. The variance of $\boldsymbol{a}_i$'s for each class should be small. A large variance will result to overlap. So, the scatter s_i^2 needs to be small.

$$s_i^2 = \sum_{\mathbf{x}_j \in \mathbf{D}_i} (a_j - m_i)^2$$

Scatter is also the total squared deviation from mean $s_i^2 = n_i\sigma_i^2$

Here, $\boldsymbol{n}_i = |\mathbf{D}_i|$ is the size, and σ^2 is the variance for class $\mathbf{c}_i$

LDA Criteria are to maximize the distance between projected means and minimize the sum of projected scatter. This is also called ***Fisher LDA*** *objective*.

$$\max_{\mathbf{w}} J(\mathbf{w}) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2}$$

w is also called the optimal linear discriminant.

We can rewrite $(m_1 - m_2)^2$ as:

$$\begin{aligned}(m_1 - m_2)^2 &= \left(\mathbf{w}^T(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\right)^2 \\ &= \mathbf{w}^T\left((\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T\right)\mathbf{w} \\ &= \mathbf{w}^T\mathbf{B}\mathbf{w}\end{aligned}$$

Where, $\mathbf{B} = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T$ the is dxd matrix

B is called *between-class scatter matrix*

The projected scatter for class $\mathbf{c_1}$ can be computed as:

$$s_1^2 = \sum_{x_i \in D_i} (a_i - m_1)^2$$
$$= \sum_{x_i \in D_i} (w^T x_i - w^T \mu_1)^2$$
$$= \sum_{x_i \in D_i} \left(w^T (x_i - \mu_1)\right)^2$$
$$= w^T \left(\sum_{x_i \in D_i} (x_i - \mu_1)(x_i - \mu_1)^T \right) w$$
$$= w^T S_1 w$$

Here, $\mathbf{S_1}$ is the scatter matrix for $\mathbf{D_1}$.

Similarly, for $\mathbf{D_2}$

$$s_2^2 = w^T S_2 w$$

$$S_1 = \sum_{x_i \in D_i} (x_i - \mu_1)(x_i - \mu_1)^T$$

Note: Scatter matrix is the same as covariance matrix, but instead of taking average deviation, it takes total deviation.

$$\mathbf{S}_i = n_i \boldsymbol{\Sigma}_i$$

Combining,

$$s_1^2 + s_2^2 = \mathbf{w}^T \mathbf{S}_1 \mathbf{w} + \mathbf{w}^T \mathbf{S}_2 \mathbf{w} = \mathbf{w}^T (\mathbf{S}_1 + \mathbf{S}_2) \mathbf{w} = \mathbf{w}^T \mathbf{S} \mathbf{w}$$

Where, $\mathbf{S} = \mathbf{S_1} + \mathbf{S_2}$ is called within-class scatter matrix

$$\max_{\mathbf{w}} J(\mathbf{w}) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2} \rightarrow \max_{\mathbf{w}} J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{B} \mathbf{w}}{\mathbf{w}^T \mathbf{S} \mathbf{w}}$$

For any function $f(x)$ and $g(x)$:

$$\frac{d}{dx}\left(\frac{f(x)}{g(x)}\right) = \frac{f'(x)g(x) - g'(x)f(x)}{g(x)^2}$$

Thus,

$$\frac{d}{d\boldsymbol{w}} J(\boldsymbol{w}) = \frac{2\boldsymbol{B}\boldsymbol{w}(\boldsymbol{w}^T \boldsymbol{S}\boldsymbol{w}) - 2\boldsymbol{S}\boldsymbol{w}(\boldsymbol{w}^T \boldsymbol{B}\boldsymbol{w})}{(\boldsymbol{w}^T \boldsymbol{S}\boldsymbol{w})^2} = \mathbf{0}$$

$$\boldsymbol{B}\boldsymbol{w}(\boldsymbol{w}^T \boldsymbol{S}\boldsymbol{w}) = \boldsymbol{S}\boldsymbol{w}(\boldsymbol{w}^T \boldsymbol{B}\boldsymbol{w})$$
$$\boldsymbol{B}\boldsymbol{w} = \boldsymbol{S}\boldsymbol{w}\left(\frac{\boldsymbol{w}^T \boldsymbol{B}\boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{S}\boldsymbol{w}}\right)$$
$$\boldsymbol{B}\boldsymbol{w} = J(\boldsymbol{w})\boldsymbol{S}\boldsymbol{w}$$
$$\boldsymbol{B}\boldsymbol{w} = \boldsymbol{\lambda}\boldsymbol{S}\boldsymbol{w}$$

If **S** is non-singular,

$$\boldsymbol{S^{-1}Bw = \lambda S^{-1}Sw}$$

$$\boldsymbol{(S^{-1}B)w = \lambda w}$$

$\boldsymbol{\lambda} = J(\boldsymbol{w})$ is an eigenvalue, **w** is an eigenvector of matrix $\mathbf{S}^{-1}\mathbf{B}$. We need the largest $\boldsymbol{\lambda}$ and the corresponding **w** will be the best separator.

LDA Algorithm

The algorithm we will be following is algorithm 20.1 from chapter 20 of the Data Mining and Machine Learning textbook.

Algorithm 20.1: Linear Discriminant Analysis

LINEARDISCRIMINANT (D):

1 $\mathbf{D}_i \leftarrow \left\{\mathbf{x}_j^T \mid y_j = c_i, j = 1, \ldots, n\right\}, i = 1, 2$ // class-specific subsets
2 $\boldsymbol{\mu}_i \leftarrow \text{mean}(\mathbf{D}_i), i = 1, 2$ // class means
3 $\mathbf{B} \leftarrow (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T$ // between-class scatter matrix
4 $\overline{\mathbf{D}}_i \leftarrow \mathbf{D}_i - \mathbf{1}_{n_i}\boldsymbol{\mu}_i^T, i = 1, 2$ // center class matrices
5 $\mathbf{S}_i \leftarrow \overline{\mathbf{D}}_i^T \overline{\mathbf{D}}_i, i = 1, 2$ // class scatter matrices
6 $\mathbf{S} \leftarrow \mathbf{S}_1 + \mathbf{S}_2$ // within-class scatter matrix
7 $\lambda_1, \mathbf{w} \leftarrow \text{eigen}(\mathbf{S}^{-1}\mathbf{B})$ // compute dominant eigenvector

Where x_j is the j th element of $\in R^d$, c_i is i the class where $i = 1, .., n$ with n is the number of classes and y_j is the class label of the j th element. The algorithm starts by dividing the dataset D into n subsets according to the class labels of the data points. In step two the class means values are computed. In step three, the between-class scatter matrix B is calculated by taking the outer product of the class mean difference $(\mu_1 - \mu_2)^T$. In general $B_i = (\mu_i - \mu)(\mu_i - \mu)^T$ and $B = \sum_{i=1}^{n} B_i$. This matrix corresponds to the scattering or the covariance of the distribution of the class means in the input domain. Then in the fourth step, center class matrices $\bar{D}_i$, are calculated by subtracting the corresponding means from the class data subsets. Then class scatters matrices S_i, are calculated by taking the inner product of the center class matrices $\bar{D}_i^T \bar{D}_i$. This corresponds to how much the data of the specific class is scattered in the input domain. Next, within-class scatter matrix S, is calculated by taking the sum of the class scatter matrices $S = \sum_{i=1}^{n} S_i$. Finally, the dominant eigenvector of the term $S^{-1}B$ is calculated to get the discriminative w vector. Then all the data points can be projected to the w vector to get a discriminative representation.

The following figures show an example calculation on the IRIS dataset

Between class scatter matrix

$$B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$$

$$= \begin{bmatrix} 1.5775 & -0.6983 \\ -0.6983 & 0.3091 \end{bmatrix}$$

$$\bar{D}_i = D_i - \mathbf{1}_n \mu_i^T$$

Class Scatter matrix

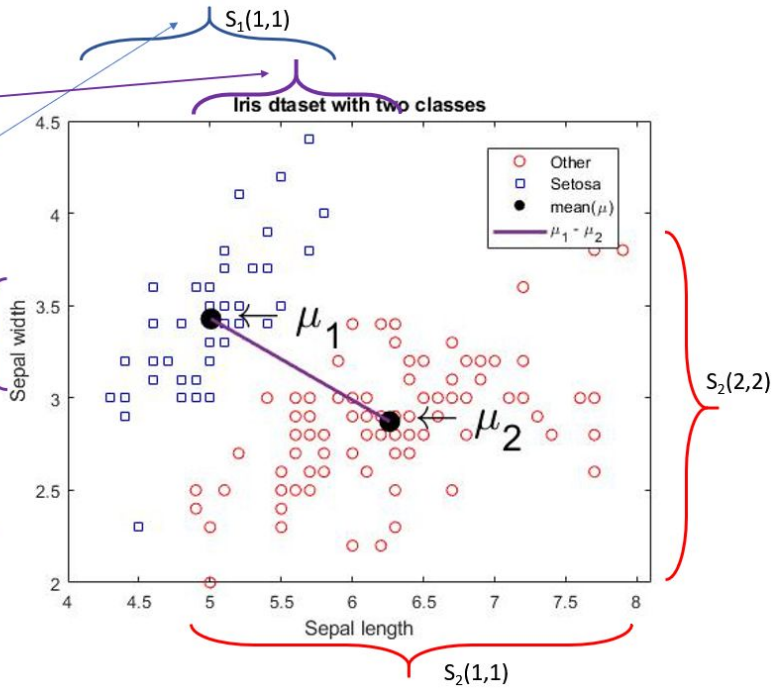
$$S_i = \bar{D}_i^T \bar{D}_i$$

$$S_1 = \begin{bmatrix} 6.0882 & 4.8616 \\ 4.8616 & 7.0408 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 43.4956 & 12.0936 \\ 12.0936 & 10.9616 \end{bmatrix}$$

Within-class scatter matrix

$$S = S_1 + S_2$$



$$\lambda_1, w \leftarrow eigen(S^{-1}B)$$

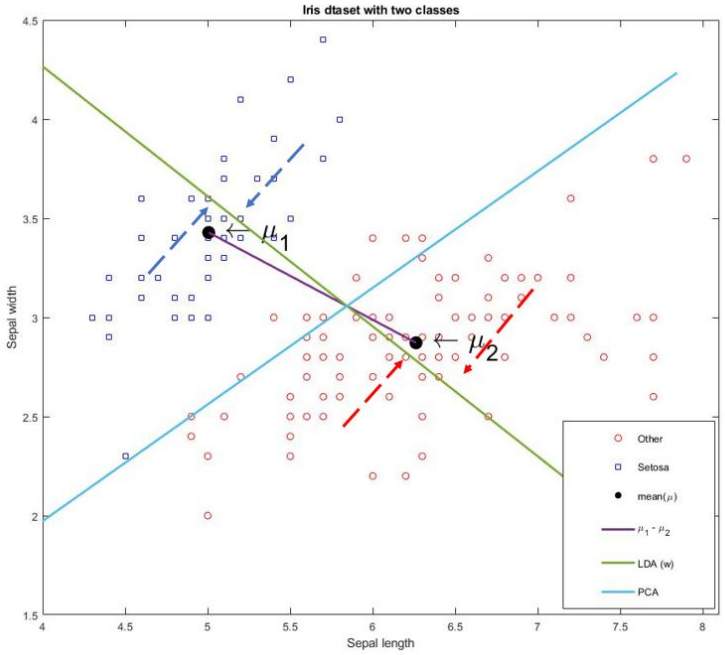
$$\Lambda = \begin{bmatrix} 0.114 & 0 \\ 0 & 0.00 \end{bmatrix}$$

$$w = \begin{bmatrix} 0.5483 & 0.4048 \\ -0.8363 & 0.9144 \end{bmatrix}$$

For two classes, **If** *S* is nonsingular

$$w = S^{-1}(\mu_1 - \mu_2)$$

$$w = \frac{w}{\|w\|}$$



For more than two classes

Between class scatter matrix

$$B_i = (\mu_1 - \mu)(\mu_1 - \mu)^T$$

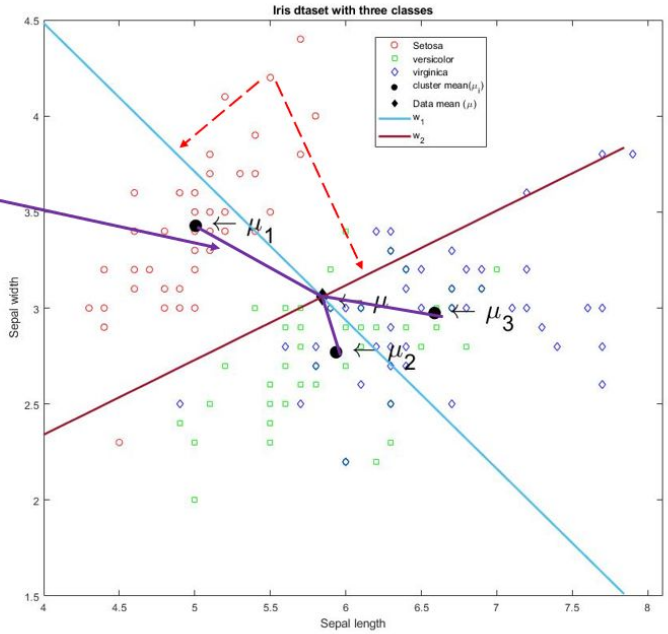
$$B = \sum_{i=1}^{c} B_i$$

$$\lambda_1, w \leftarrow eigen(S^{-1}B)$$

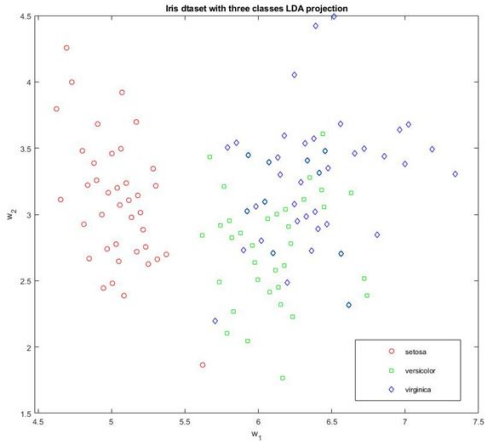
$$\Lambda = \begin{bmatrix} 0.834 & 0 \\ 0 & 0.032 \end{bmatrix}$$

$$w = \begin{bmatrix} 0.6118 & 0.3625 \\ -0.7910 & 0.9320 \end{bmatrix}$$

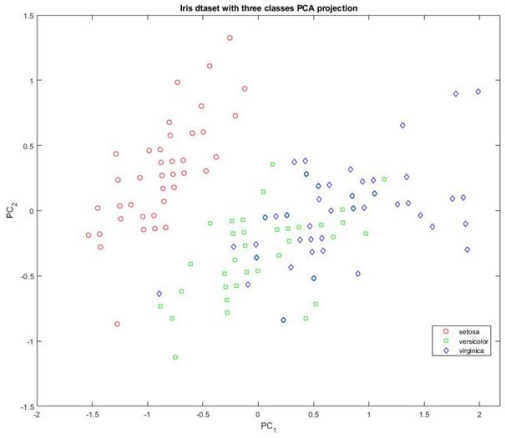
Since three mean points a considered,
The final w will be a plane



This calculation is finding the optimal vector by both the increase of the class means distance and the decrease of the within-class scatter by using the class label information. In contrast, in Principal Component Analysis (PCA) the algorithm does not take into account the class label information and to finds the vector which maximizes the whole data variance. Hence for classification tasks, the LDA is better suited as a dimensionality reduction technique.



LDA



PCA

Reference textbook: Mohammed J. Zaki, Wagner Meira, Jr., Data Mining, and Machine Learning: Fundamental Concepts and Algorithms, 2nd Edition, Cambridge University Press, March 2020. ISBN: 978-1108473989.