Linear Discriminant Analysis (LDA)

Dr. Sriparna Saha
Assistant Proffesor
IIT Patna

Introduction

1. Linear Discriminant Analysis (LDA) is most commonly used as dimensionality reduction technique.
2. The goal of an LDA is to project a feature space (a dataset n-dimensional samples) onto a smaller subspace k (where $k \leq n-1$) while maintaining the class-discriminatory information.

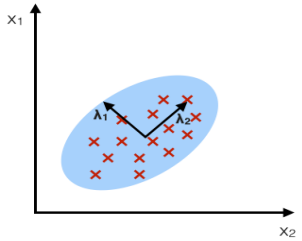
Principal Component Analysis vs. Linear Discriminant Analysis

	Principal Component Analysis (PCA)	**Linear Discriminant Analysis (LDA)**
Uses	Commonly used for dimensionality reduction.	Commonly used for dimensionality reduction.
Technique	PCA can be described as an "unsupervised" algorithm. Its goal is to find the directions that maximize the variance in a dataset.	LDA is "supervised" and computes the directions ("linear discriminants") that will represent the axes that maximize the separation between multiple classes.
Goals	The PCA accounts for the most variance in the whole dataset.	The LDA gives us the axes that account for the most variance between the individual classes.

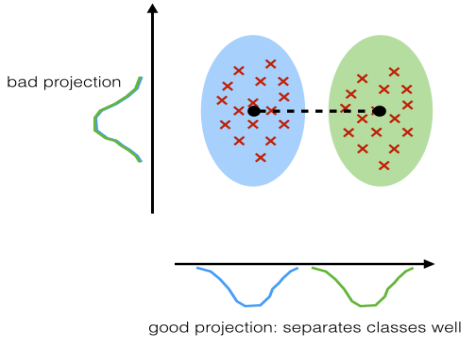
Principal Component Analysis vs. Linear Discriminant Analysis

Although it might sound intuitive that LDA is superior to PCA for a multi-class classification task where the class labels are known, but PCA tends to outperform LDA if the number of samples per class is relatively small.

PCA:
component axes that maximize the variance



LDA:
maximizing the component axes for class-separation



Summarizing the LDA approach

Listed below are the 5 general steps for performing a linear discriminant analysis; we will explore them in more detail in the following sections.

1. Compute the *d*-dimensional mean vectors for the different classes from the dataset.
2. Compute the scatter matrices (in-between-class and within-class scatter matrix).
3. Compute the eigenvectors (e_1,e_2,...,e_d) and corresponding eigenvalues (λ_1,λ_2,...,λ_d) for the scatter matrices.
4. Sort the eigenvectors by decreasing eigenvalues and choose *k* eigenvectors with the largest eigenvalues to form a $d \times k$ dimensional matrix *W* (where every column represents an eigenvector).
5. Use this $d \times k$ eigenvector matrix to transform the samples onto the new subspace. This can be summarized by the matrix multiplication: $Y=X \times W$ (where *X* is a $n \times d$-dimensional matrix representing the *n* samples, and *y* are the transformed $n \times k$-dimensional samples in the new subspace).

Normality assumptions

It should be mentioned that LDA assumes normal distributed data, features that are statistically independent, and identical covariance matrices for every class.

Steps of LDA:
About the dataset used in example

The iris dataset contains measurements for 150 iris flowers from three different species.

The three classes in the Iris dataset:

1. Iris-setosa (n=50)
2. Iris-versicolor (n=50)
3. Iris-virginica (n=50)

The four features of the Iris dataset:

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm

Step 1: Computing the d-dimensional mean vector

In this first step, we will start off with a simple computation of the mean vectors $m_{i,}$ (i=1,2,3) of the 3 different flower classes:

Step 2: Computing the Scatter Matrices

Now, we will compute the two *4x4*-dimensional matrices: The within-class and the between-class scatter matrix.

2.1 Within-class scatter matrix S_W

The **within-class scatter** matrix S_W is computed by the following equation:

$$S_W = \sum_{i=1}^{c} S_i$$

Where

$$S_i = \sum_{\boldsymbol{x} \in D_i}^{n} (\boldsymbol{x} - \boldsymbol{m}_i)\,(\boldsymbol{x} - \boldsymbol{m}_i)^T$$

(scatter matrix for every class)

$$\boldsymbol{m}_i = \frac{1}{n_i} \sum_{\boldsymbol{x} \in D_i}^{n} \boldsymbol{x}_k$$

$\boldsymbol{m}_i$ **is the mean vector**

Step 2: Computing the Scatter Matrices

2.1 Between-class scatter matrix S_B

The **between-class scatter** matrix *SB* is computed by the following equation:

$$S_B = \sum_{i=1}^{c} N_i (\boldsymbol{m}_i - \boldsymbol{m})(\boldsymbol{m}_i - \boldsymbol{m})^T$$

where

$\boldsymbol{m}$ is the overall mean, and $\boldsymbol{m}_i$ and $\boldsymbol{N}_i$ are the sample mean and sizes of the respective classes.

Step 3: Solving the generalized eigenvalue problem for matrix $S_W^{-1}S_B$

Next, we will solve the generalized eigenvalue problem for the matrix $S_W^{-1}S_B$ to obtain the linear discriminants.

A quick check that the eigenvector-eigenvalue calculation is correct and satisfy the equation:

$$\boldsymbol{Av} = \lambda \boldsymbol{v}$$

where

$$\boldsymbol{A} = S_W^{-1} S_B$$
$$\boldsymbol{v} = \text{Eigenvector}$$
$$\lambda = \text{Eigenvalue}$$

Step 4: Selecting linear discriminant for the new feature subspace

4.1. Sorting the eigenvectors by decreasing eigenvalues

The eigenvectors with the lowest eigenvalues bear the least information about the distribution of the data, and those are the ones we want to drop. The common approach is to rank the eigenvectors from highest to lowest corresponding eigenvalue and choose the top *k* eigenvectors.

4.2. Choosing *k* eigenvectors with the largest eigenvalues

After sorting the eigenpairs by decreasing eigenvalues, it is now time to construct our $k \times d$-dimensional eigenvector matrix ***W***.

Step 5: Transforming the samples onto the new subspace

In the last step, we use the matrix ***W*** that we just computed to transform our samples onto the new subspace via the equation

$$Y = X \times W$$

(where ***X*** is a ***n×d***-dimensional matrix representing the ***n*** samples, and ***Y*** are the transformed ***n×k***-dimensional samples in the new subspace).