Introduction

Linear discriminant analysis (LDA), normal discriminant analysis (NDA), or discriminant function analysis is a generalization of Fisher's linear discriminant, a method used in statistics, pattern recognition and machine learning to find a linear combination of features that characterizes or separates two or more classes of objects or events. The resulting combination**may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification**.

Comments:

- **Discriminant analysis is not discriminative learning algorithms, but are generative algorithm**.
- LDA is often used in supervised learning while PCA is often used in unsupervised learning. (see details later)

Quadratic and linear discriminant function

https://people.revoledu.com/kardi/tutorial/LDA/LDA%20Formula.htm (https://people.revoledu.com/kardi/tutorial/LDA/LDA%20Formula.htm). (see downloaded PDF in /fundamentals folder)

- If there are g groups, the total error of classification is minimized by assigning the object to group i which has the highest conditional probability where $P(i \mid x) > p(j \mid x), \forall j \neq i$.
- We use Bayes rule to calculate $P(i \mid x)$ instead of estimate it directly (see generative and discriminative algorithms in other notes)
$$p(i \mid x) = \frac{p(x \mid i)p(i)}{\sum_j p(x \mid j)P(j)}$$
- Thus the rule to minimize classification error becomes: assign the object to group i if
$$\frac{p(x \mid i)p(i)}{\sum_k p(x \mid k)P(k)} > \frac{p(x \mid j)p(j)}{\sum_k p(x \mid k)P(k)}, \forall j \neq i$$
This further becomes
$$p(x \mid i)p(i) > p(x \mid j)p(j), \forall j \neq i$$
- Assuming $P(x \mid i)$ is a multivariate Gaussian
$$p(x \mid i)p(i) = \frac{1}{(2\pi)^{\frac{n}{2}}|C_i|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}(x-\mu_i)^T C_i^{-1}(x-\mu_i)\right)$$
and plugging this into the previous inequality, we have the following equation after taking logarithm,

$$d_i(x) - 2\ln P(i) < d_j(x) - 2\ln P(j), \forall i \neq j$$

where $d_i(x) - \ln(|C_i|) + (x - \mu_i)^T C_i^{-1}(x - \mu_i)$. **This is the quadratic discriminant function.**

- If further assume $C = C_i = C_j$, then the above rule can be simplified as: assign object with measurement x to group i if
  $$f_i > f_j, \forall i \neq j$$
  where $f_i = \mu_i C^{-1} x^T - \frac{1}{2}\mu_i C^{-1}\mu_i^T + \ln P(i)$ is the **linear discrimination function**. Thus linear discriminant analysis has assumption of multivariate normal distribution and all groups have the same covariance matrix.
- So far we separate the two classes by requiring one term is bigger than the other. Now we also require that one term is bigger than the other term with a threshold T. With the notation as used in the following Wikipedia link https://en.wikipedia.org/wiki/Linear_discriminant_analysis (https://en.wikipedia.org/wiki/Linear_discriminant_analysis) and also requiring the conditions of LDA satisfied (i.e. Normal distribution and same covariance), the criterion now becomes
  $$w \cdot x > c$$
  where
  $$w = \Sigma^{-1}(\mu_1 - \mu_0)$$
  $$c = \frac{1}{2}(T - \mu_0^T \Sigma_0^{-1} \mu_0 + \mu_1^T \Sigma_1^{-1} \mu_1)$$
  Here $\Sigma_0 = \Sigma_0 = \Sigma = C$. In the next section, we may obtain w by maximizing a 'separation' defined by Fisher, where w is actually a unit vector.

Fisher's linear discriminant

http://www.csd.uwo.ca/~olga/Courses/CS434a_541a/Lecture8.pdf (http://www.csd.uwo.ca/~olga/Courses/CS434a_541a/Lecture8.pdf)

More details of Fisher's linear discriminant are provided below by following a slightly different notation. The main idea here is to find projection to a line subject to samples from different classes are well separated.

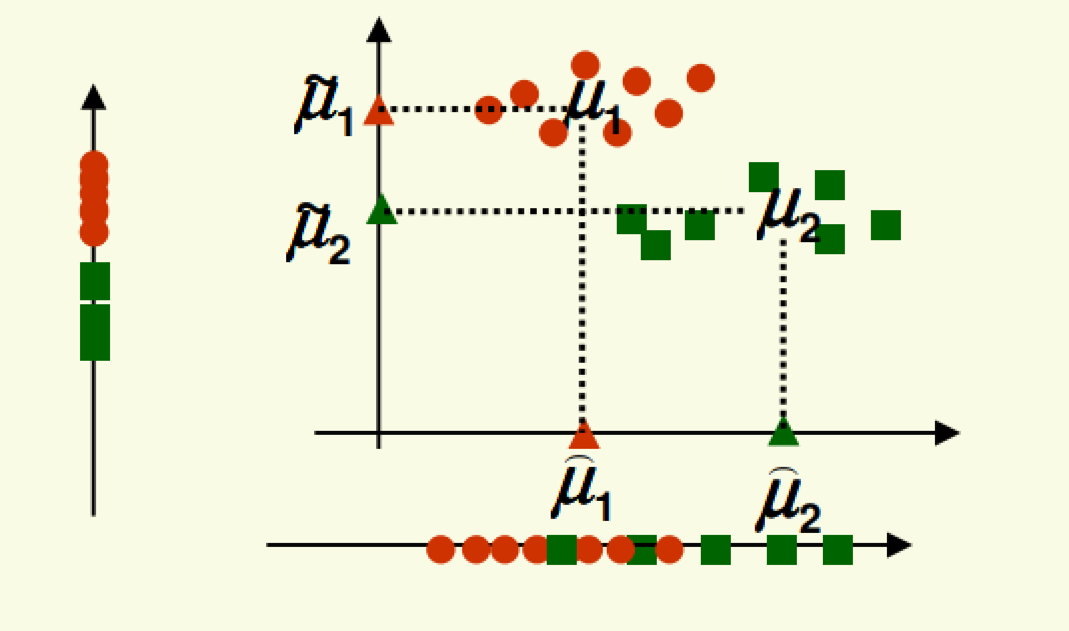
Assume we have two classes and d dimensional samples $x_1, x_2, \ldots x_n$ where n_1 and n_2 belong to first and second classes respectively.

The projection of sample x_i onto a line in direction v is given by $v^T x_i$. Let $\tilde{\mu_1}$ and $\tilde{\mu_2}$ be the means of projections of classes 1 and 2, while μ_1 and μ_2 be the means of the two classes.

A tentative measure for separation is $|\tilde{\mu_1} - \tilde{\mu_2}|$, where

$$\tilde{\mu_1} = \frac{1}{n_1}\sum_{x_i \in C_1}^{n_1} v^T x_i = v^T\left(\frac{1}{n_1}\sum_{x_i \in C_1}^{n_1} x_i\right) = v^T \mu_1$$

and similarly $\tilde{\mu_2} = v^T \mu_2$.



As shown in the figure above, it turns out that sometimes a bigger value of $|\tilde{\mu}_1 - \tilde{\mu}_2|$ is not necessarily a good measure for separation as it does not consider variance. We need normalize $|\tilde{\mu}_1 - \tilde{\mu}_2|$ by a factor which is proportional to variance.

For samples $z_1, \ldots z_n$ and sample mean $\mu_z = \frac{1}{n} \sum_i^n z_i$, define their **scatter** as

$$s = \sum_i^n (z_i - \mu_z)^2$$

, which is the sample variance multiplied by n. This quantity describes how 'scattered' the data around the mean and thus is called scatter. Now we normalize the former $|\tilde{\mu}_1 - \tilde{\mu}_2|$ by scatter.

Let $y_i = v^T x_i$ are the projected samples. Then the scatter for class 1 and 2 are respectively

$$\tilde{s}_1{}^2 = \sum_{y_i \in C_1} (y_i - \tilde{\mu}_1)^2$$

$$\tilde{s}_2{}^2 = \sum_{y_i \in C_2} (y_i - \tilde{\mu}_2)^2$$

Normalizing by scatter gives the Fisher linear discriminant

$$J(v) = \frac{(\tilde{\mu}_1 - \tilde{\mu}_2)^2}{\tilde{s}_1{}^2 + \tilde{s}_2{}^2}$$

We are finding v which maximizes $J(v)$. To do so, we define **scatter matrices** S_1 and S_2 for the two classes, which measure the scatter of original samples

$$S_1 = \sum_{x_i \in C_1} (x_i - \mu_1)(x_i - \mu_1)^T$$

$$S_2 = \sum_{x_i \in C_2} (x_i - \mu_2)(x_i - \mu_2)^T$$

From these definitions, $y_i = v^T x_i$, $\tilde{\mu}_1 = v^T \mu_1$ and $\tilde{\mu}_2 = v^T \mu_2$, we have

$$\tilde{s}_1^{\,2} = \sum_{y_i \in C_1} (y_i - \tilde{\mu}_1)^2 = \sum_{x_i \in C_1} (v^T x_i - v^T \mu_1)^2 = \ldots = v^T S_1 v$$

and similarly $\tilde{s}_2^{\,2} = v^T S_2 v$. Therefore,

$$\tilde{s}_1^{\,2} + \tilde{s}_2^{\,2} = v^T S_1 v + v^T S_2 v \equiv v^T S_w v$$

where S_w is defined as the **within the class scatter matrix**.

Further define the **between the class scatter matrix**

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

which measures separation between the means of two classes. With this definition, we can rewrite

$$(\tilde{\mu}_1 - \tilde{\mu}_2)^2 = (v^T \mu_1 - v^T \mu_2)^2 = v^T S_B v$$

Therefore, the objective function becomes

$$J(v) = \frac{(\tilde{\mu}_1 - \tilde{\mu}_2)^2}{\tilde{s}_1^{\,2} + \tilde{s}_2^{\,2}} = \frac{v^T S_B v}{v^T S_w v}$$

To maximize $J(v)$, take the derivative with respect to v and set it to zero. This will give the generalized eigenvalue problem $S_B v = \lambda S_w v$, and the final solutions are

$$v = S_w^{-1}(\mu_1 - \mu_2)$$

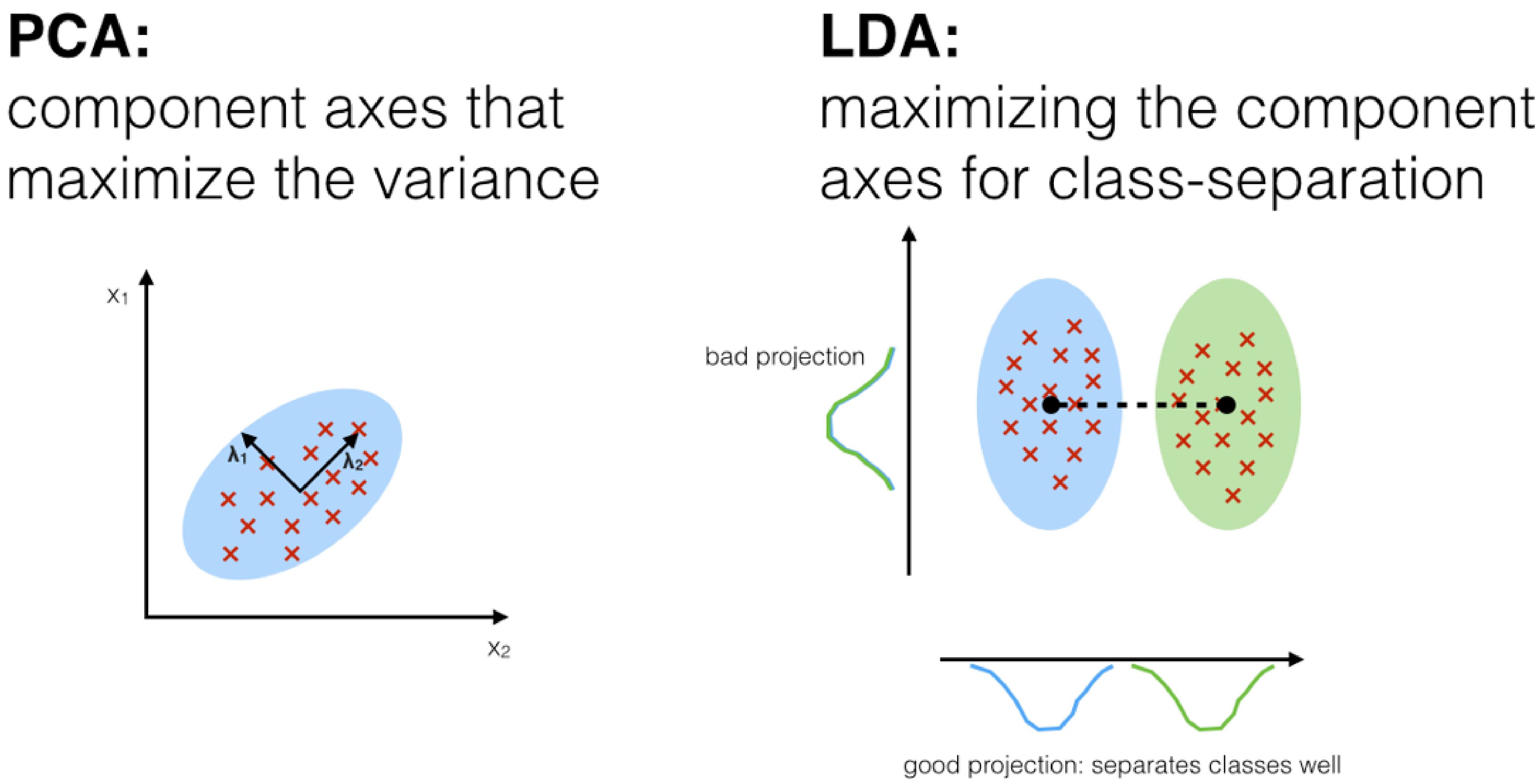
$$\lambda = (\mu_1 - \mu_2)^T x$$

Comments:

Comparing the results we obtained in the previous section $w = \Sigma^{-1}(\mu_1 - \mu_0)$, we find that the v obtained by maximizing Fisher's discriminant J is equivalent to the LDA results arrived earlier by introducing a threshold. What is behind this equivalence?

Application of LDA

- LDA is most commonly used as dimensionality reduction technique in the pre-processing step for pattern-classification and machine learning applications. The goal is to project a dataset onto a lower-dimensional space with good class-separability in order avoid overfitting and also reduce computational costs. https://sebastianraschka.com/Articles/2014_python_lda.html (https://sebastianraschka.com/Articles/2014_python_lda.html). In general, dimensionality reduction does not only help reducing computational costs for a given classification task, but it can also be helpful to avoid overfitting.
- LDA works when the measurements made on independent variables for each observation are continuous quantities. When dealing with categorical independent variables, the equivalent technique is discriminant correspondence analysis.
- It should be mentioned that LDA assumes normal distributed data, features that are statistically independent, and identical covariance matrices for every class. However, this **only applies for LDA as classifier** and LDA for dimensionality reduction can also work reasonably well if those assumptions are violated. And even for classification tasks LDA seems can be quite robust to the distribution of the data:
- Both Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) **are linear transformation techniques that are commonly used for dimensionality reduction.** PCA can be described as an "unsupervised" algorithm. In contrast, **LDA is "supervised" and computes the directions that will represent the axes that that maximize the separation between multiple classes.**
- Because LDA always tries to maximize separation, it might sound intuitive that LDA is superior to PCA for a multi-class classification task where the class labels are known. **This might not always be the case**. For example, comparisons between classification accuracies for image recognition after using PCA or LDA show that PCA tends to outperform LDA if the number of samples per class is relatively small. In practice, it is also not uncommon to use both LDA and PCA in combination: E.g., PCA for dimensionality reduction followed by an LDA.
- With kernel trick, LDA can handle nonlinear problems.


PCA:
component axes that maximize the variance
x1
λ1
λ2
x2
LDA:
maximizing the component axes for class-separation
bad projection
good projection: separates classes well

An implementation of LDA

https://sebastianraschka.com/Articles/2014_python_lda.html (https://sebastianraschka.com/Articles/2014_python_lda.html).

- The link above provides a python implementation of LDA step by step. It eventually handles the LDA by numerically solving a generalized eigenvalue problem. Compare this with the closed-form solution introduced earlier.
- Compute the d-dimensional mean vectors for the different classes from the dataset.
- Compute the scatter matrices (in-between-class and within-class scatter matrix).
- Compute the eigenvectors $e_1, e_2, \ldots e_d$ and corresponding eigenvalues $\lambda_1, \lambda_2, \ldots \lambda_d$ from the scatter matrices.
- Sort the eigenvectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues to form a $d \times k$ dimension matrix W, where column represent an eigenvector.
- Use this $d \times k$ eigenvector matrix to transform the samples onto the new subspace. This can be summarized by the matrix multiplication: $Y = XW$. Here, X is a $n \times d$ matrix representing n samples, and y are the transformed $n \times k$ samples in the new

space.