Lecture 5: Linear Discriminant Analysis (LDA)

School of Mathematical Sciences, Xiamen University

1. Setting

- The data considered in this lecture are assumed to be labeled, or annotated, in the sense that each data point belongs to one and only one class, and that the class-belonging information is available.
- Let $\mathcal{D}$ denote the data set partitioned into k nonintersecting subsets,

$$\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \cdots \mathcal{D}_k, \quad \mathcal{D}_j \cap \mathcal{D}_k = \emptyset, \quad \forall j \neq k,$$

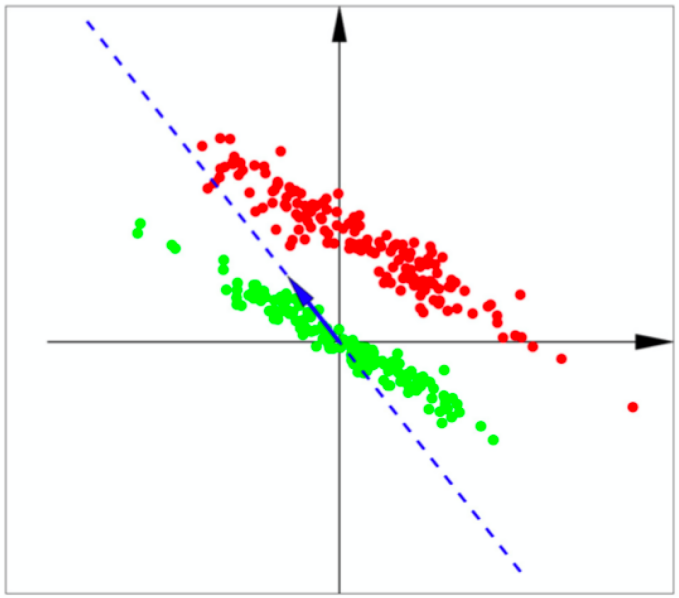
and let

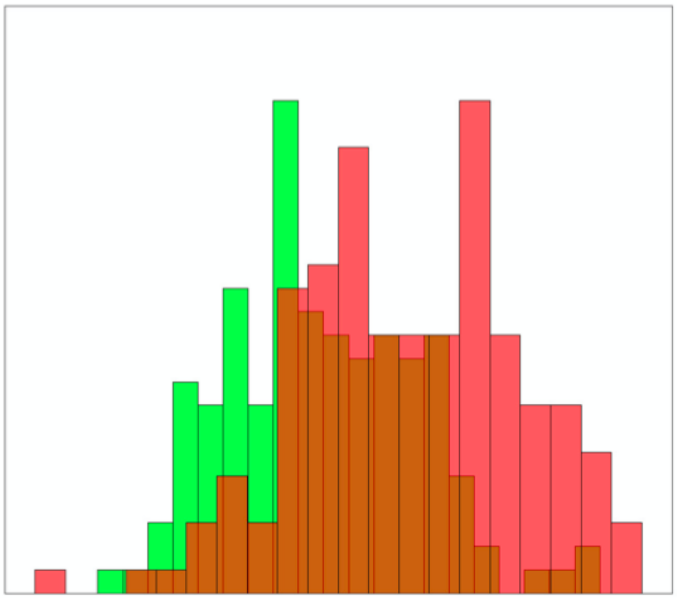
$$I_j = \ell \text{ equivalent to } \mathbf{x}^{(j)} \in \mathcal{D}_\ell, \quad 1 \leq j \leq p,$$

define the annotation vector $I \in \{1, 2, \cdots, k\}^p$ that contains the labels assigning each data vector to one and only one subset.

- We refer to the subsets $\mathcal{D}_\ell$ as clusters, regardless of whether or not we know that the groups are distinguishable from each other.

- In the following, we do not question the annotation of the data, but rather look for ways of assessing how separate the given clusters really are.
- Can we find a few directions with the property that when the data are projected along these directions, the clusters are maximally separated from each other?
- Data consisting of two separate classes with different labels. In this case, the first principal direction (the blue arrow) is not useful.





2. Scatter matrices and spread

- Given a data matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$, centered or noncentered, the components of the data vectors $\mathbf{x}^{(j)}$ on a given direction vector $\mathbf{q} \in \mathbb{R}^n$, with $\|\mathbf{q}\|_2 = 1$, can be collected into a row vector $\mathbf{y}$ of length p,

$$\mathbf{y} = \mathbf{X}^\top \mathbf{q}.$$

Recalling that the square of the spread of the data in the direction $\mathbf{q}$ is the quantity

$$\text{spread}(\mathbf{y})^2 = \|\mathbf{y}\|_2^2,$$

and replacing $\mathbf{y}$ by its expression in terms of the data matrix, we obtain

$$\text{spread}(\mathbf{y})^2 = \mathbf{q}^\top \mathbf{X}\mathbf{X}^\top \mathbf{q} = \mathbf{q}^\top \mathbf{S}\mathbf{q}.$$

The matrix $\mathbf{S}$,

$$\mathbf{S} = \mathbf{X}\mathbf{X}^\top \in \mathbb{R}^{n \times n},$$

is called the *scatter matrix* of the data $\mathbf{X}$.

- Therefore, the direction of maximum (minimum) spread of the data is the vector that maximizes (minimizes) the quadratic expression $\mathbf{q}^\top \mathbf{S}\mathbf{q}$.
- Without loss of generality we can reorder the data so that the columns corresponding to points in the same cluster are adjacent, i.e.,

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_k \end{bmatrix} \in \mathbb{R}^{n \times p},$$

where the columns of $\mathbf{X}_\ell$ are the data vectors in cluster $\mathcal{D}_\ell$,

$$\mathbf{X}_\ell = \begin{bmatrix} \mathbf{x}^{(j_1)} & \mathbf{x}^{(j_2)} & \cdots & \mathbf{x}^{(j_{p_\ell})} \end{bmatrix} \in \mathbb{R}^{n \times p_\ell},$$

and

$$I_\ell = \begin{bmatrix} j_1 & j_2 & \cdots & j_{p_\ell} \end{bmatrix}.$$

We have

$$p_1 + p_2 + \cdots + p_k = p.$$

We define the cluster centroids as

$$\mathbf{c}^{(\ell)} = \frac{1}{p_\ell} \sum_{j \in I_\ell} \mathbf{x}^{(j)}, \quad 1 \leq \ell \leq k.$$

The global centroid of the data is defined by the formula

$$\mathbf{c} = \frac{1}{p} \sum_{j=1}^{p} \mathbf{x}^{(j)}.$$

Define the centered data matrices for each cluster,

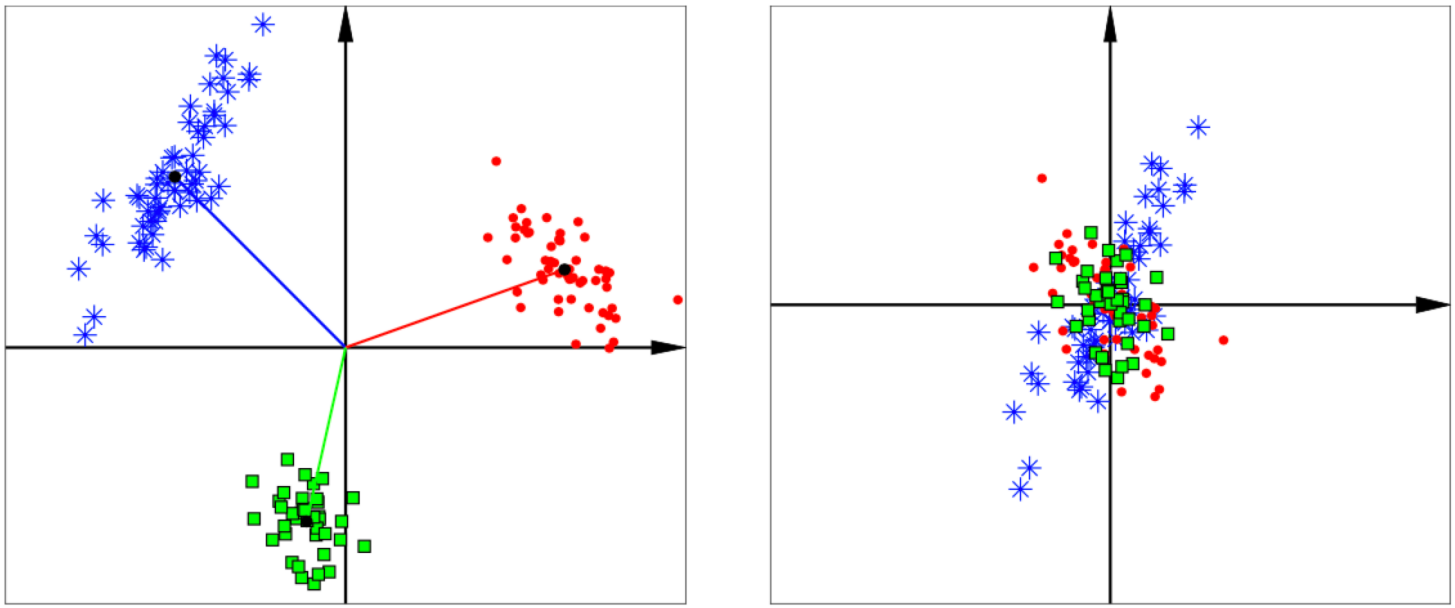
$$\mathbf{X}_{\ell,c} = \begin{bmatrix} \mathbf{x}^{(j_1)} - \mathbf{c}^{(\ell)} & \mathbf{x}^{(j_2)} - \mathbf{c}^{(\ell)} & \dots & \mathbf{x}^{(j_{p_\ell})} - \mathbf{c}^{(\ell)} \end{bmatrix} \in \mathbb{R}^{n \times p_\ell}.$$

Centering the data in a cluster is tantamount to moving the cluster so that its centroid coincides with the coordinate origin.

- The centered matrices are then collected in the within-cluster centered data matrix,

$$\mathbf{X}_w = \begin{bmatrix} \mathbf{X}_{1,c} & \mathbf{X}_{2,c} & \cdots & \mathbf{X}_{k,c} \end{bmatrix} \in \mathbb{R}^{n \times p}.$$

Geometrically, $\mathbf{X}_w$ provides a representation of the data where each cluster has been separately centered around the origin.



The left panel shows the original clustered data, with the respective cluster centroids, while the right panel shows the data corresponding to the matrix $\mathbf{X}_w$.

- Define the within-cluster scatter matrix $\mathbf{S}_w \in \mathbb{R}^{n \times n}$ for the centered data $\mathbf{X}_w$ by the formula

$$\mathbf{S}_w = \mathbf{X}_w \mathbf{X}_w^\top.$$

We also have

$$\mathbf{S}_w = \sum_{\ell=1}^{k} \mathbf{S}_\ell,$$

where $\mathbf{S}_\ell$ is the scatter matrix of the ℓth cluster,

$$\mathbf{S}_\ell = \mathbf{X}_{\ell,c} \mathbf{X}_{\ell,c}^\top = \sum_{j \in I_\ell} (\mathbf{x}^{(j)} - \mathbf{c}^{(\ell)})(\mathbf{x}^{(j)} - \mathbf{c}^{(\ell)})^\top \in \mathbb{R}^{n \times n}.$$

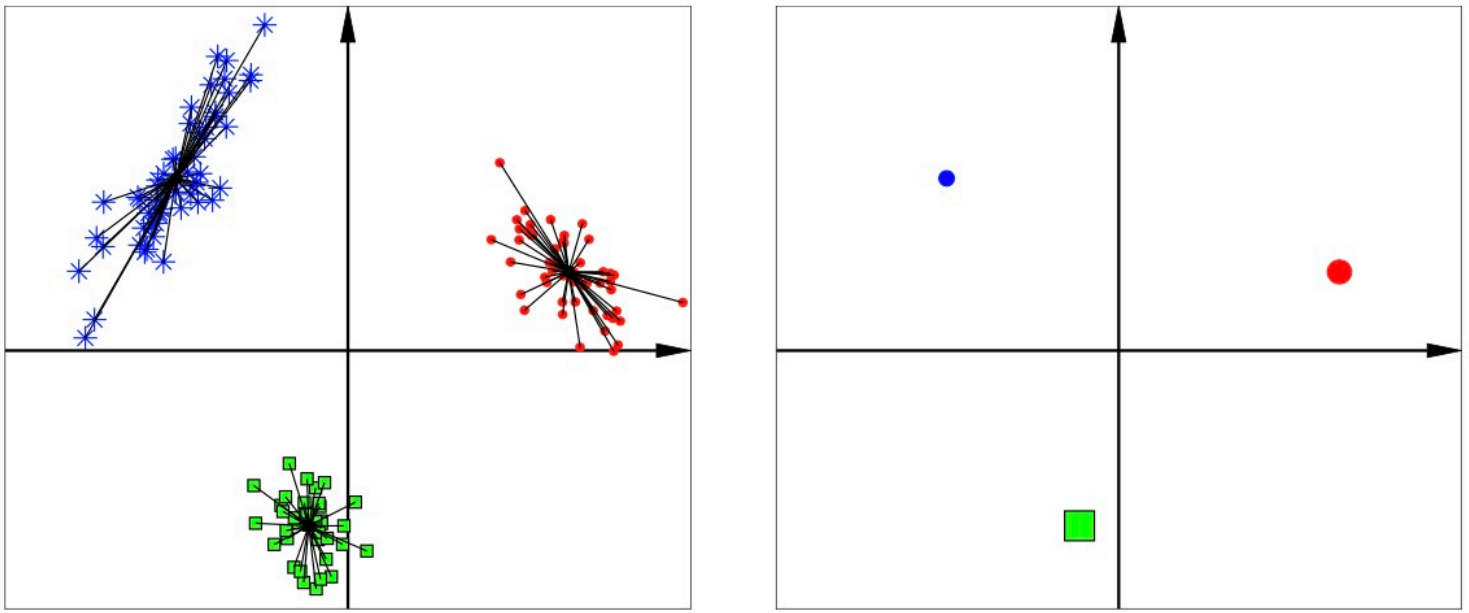
- We need a measure for how the clusters themselves are scattered. Therefore, we define a centroid approximation of the data matrix $\mathbf{X}$ by simply replacing each data vector $\mathbf{x}^{(j)}$ by the centroid of its cluster. Denote by $\overline{\mathbf{X}}_\ell$ the centroid approximation of $\mathbf{X}_\ell$,

$$\overline{\mathbf{X}}_\ell = \begin{bmatrix} \mathbf{c}^{(\ell)} & \mathbf{c}^{(\ell)} & \cdots & \mathbf{c}^{(\ell)} \end{bmatrix} \in \mathbb{R}^{n \times p_\ell}.$$

- Collect the centroid approximation matrices for all the clusters into the matrix

$$\overline{\mathbf{X}} = \begin{bmatrix} \overline{\mathbf{X}}_1 & \overline{\mathbf{X}}_2 & \cdots & \overline{\mathbf{X}}_k \end{bmatrix} \in \mathbb{R}^{n \times p},$$

whose rank is at most k. A schematic rendering of the meaning of $\overline{\mathbf{X}}_\ell$ and $\overline{\mathbf{X}}$ is shown in the following.



- Center the matrix $\overline{\mathbf{X}}$ by subtracting the global centroid from each column,

$$\overline{\mathbf{X}}_c = \overline{\mathbf{X}} - \begin{bmatrix} \mathbf{c} & \mathbf{c} & \cdots & \mathbf{c} \end{bmatrix},$$

and define the between-cluster scatter matrix $\mathbf{S}_b \in \mathbb{R}^{n \times n}$ by

$$\mathbf{S}_b = \overline{\mathbf{X}}_c \overline{\mathbf{X}}_c^\top,$$

which can be expressed also in the form

$$\begin{aligned} \mathbf{S}_b &= \sum_{\ell=1}^{k} \sum_{j=1}^{p_\ell} (\mathbf{c}^{(\ell)} - \mathbf{c})(\mathbf{c}^{(\ell)} - \mathbf{c})^\top \\ &= \sum_{\ell=1}^{k} p_\ell (\mathbf{c}^{(\ell)} - \mathbf{c})(\mathbf{c}^{(\ell)} - \mathbf{c})^\top. \end{aligned}$$

3. Optimizing the spread between clusters

- The goal of this section is to find a few projection directions in the data space along which the within-cluster spread is as small as possible and the between-cluster spread is as large as possible.
- The basic idea here is that after projecting the data onto such directions, the clusters will appear as compact as possible, while the separation between clusters is emphasized.
- The objective function guiding the search for vectors determining the directions with the desired properties is the ratio

$$H(\mathbf{q}) = \frac{\mathbf{q}^\top \mathbf{S}_b \mathbf{q}}{\mathbf{q}^\top \mathbf{S}_w \mathbf{q}}.$$

- We argue that any nonzero vector $\mathbf{q}$ that maximizes this ratio is a desired direction. For the time being, we assume that the matrix $\mathbf{S}_w$ is symmetric positive definite. The case of symmetric positive semidefinite $\mathbf{S}_w$ will be addressed later.

- Since for every $\alpha > 0$, $H(\alpha\mathbf{q}) = H(\mathbf{q})$, then we can scale the vector $\mathbf{q}$, without changing the value of the objective function, to guarantee that $\mathbf{q}^\top \mathbf{S}_w \mathbf{q} = 1$. The objective function becomes

$$H(\mathbf{q}) = \frac{\mathbf{q}^\top \mathbf{S}_b \mathbf{q}}{\mathbf{q}^\top \mathbf{S}_w \mathbf{q}} = \mathbf{q}^\top \mathbf{S}_b \mathbf{q}.$$

- By assumption, the matrix $\mathbf{S}_w$ is symmetric positive definite, it admits the Cholesky factorization,

$$\mathbf{S}_w = \mathbf{K}^\top \mathbf{K},$$

where $\mathbf{K}$ is upper triangular and invertible. Hence, we can write the constraint as

$$\mathbf{q}^\top \mathbf{S}_w \mathbf{q} = \mathbf{q}^\top \mathbf{K}^\top \mathbf{K} \mathbf{q} = \|\mathbf{K}\mathbf{q}\|_2^2 = 1,$$

or, by defining the vector $\mathbf{w}$ of unit length by

$$\mathbf{K}\mathbf{q} = \mathbf{w}, \text{ or } \mathbf{q} = \mathbf{K}^{-1}\mathbf{w}.$$

- The problem can be restated as

$$\text{maximize} \quad \mathbf{w}^\top \mathbf{K}^{-\top} \mathbf{S}_b \mathbf{K}^{-1} \mathbf{w} \quad \text{subject to} \quad \|\mathbf{w}\|_2 = 1.$$

- There are two equivalent ways to solve this maximization problem.

  (1) The method of Lagrange multipliers. Define a Lagrange function

$$L_\lambda(\mathbf{w}) = \mathbf{w}^\top \mathbf{K}^{-\top} \mathbf{S}_b \mathbf{K}^{-1} \mathbf{w} - \lambda(\|\mathbf{w}\|_2^2 - 1),$$

  where $\lambda \in \mathbb{R}$ is a Lagrange multiplier. We have

$$\nabla_{\mathbf{w}} L_\lambda(\mathbf{w}) = 2(\mathbf{K}^{-\top} \mathbf{S}_b \mathbf{K}^{-1} \mathbf{w} - \lambda \mathbf{w}) = \mathbf{0}.$$

  (2) SPD matrix eigendecomposition.

$$\mathbf{K}^{-\top} \mathbf{S}_b \mathbf{K}^{-1} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^\top.$$

- We have $\mathbf{w} = \mathbf{v}^{(1)}$, the eigenvector corresponding to the largest eigenvalue.

- Having found the vector $\mathbf{w}$, we may then solve $\mathbf{K}\mathbf{q} = \mathbf{w}$ for $\mathbf{q}$. We also note that

$$\mathbf{K}^{-1}\mathbf{K}^{-\top}\mathbf{S}_b\mathbf{K}^{-1}\mathbf{w} = \lambda\mathbf{K}^{-1}\mathbf{w}$$

or

$$\mathbf{S}_w^{-1}\mathbf{S}_b\mathbf{q} = \lambda\mathbf{q},$$

that is, the vectors $\mathbf{q}$ are eigenvectors of the nonsymmetric matrix $\mathbf{S}_w^{-1}\mathbf{S}$, which has thus been shown to have all real eigenvalues.

- If there are more than two clusters in the data, it may be useful to find more than just one projection direction to represent the data. A natural way is to find the eigenvectors $\mathbf{v}^{(\ell)}$, $\ell > 1$, set $\mathbf{w}^{\ell} = \mathbf{v}^{(\ell)}$, and define the projection directions as

$$\mathbf{q}^{(\ell)} = \mathbf{K}^{-1}\mathbf{w}^{(\ell)}.$$

Observe that while the vectors $\mathbf{v}^{(\ell)}$ are mutually orthogonal, the vectors $\mathbf{q}^{(\ell)}$ are usually not.

- So far, we have assumed that the matrix $\mathbf{S}_w$ is symmetric positive definite, and therefore invertible. If this assumption is not valid, the argument above does not work, as $\mathbf{S}_w$ will not admit the Cholesky factorization. On the other hand, being symmetric positive semidefinite, the matrix $\mathbf{S}_w$ could be factored as

$$\mathbf{S}_w = \mathbf{S}_w^{1/2}\mathbf{S}_w^{1/2},$$

but since the factors would not be invertible, the argument does not work. To overcome this problem, we replace $\mathbf{S}_w$ by a nearby positive definite approximation by setting

$$\mathbf{S}_{w,\varepsilon} = \mathbf{S}_w + \varepsilon\mathbf{I}_n,$$

where $\varepsilon > 0$ is a small scalar. For any $\mathbf{v} \in \mathbb{R}^n$, $\mathbf{v} \neq \mathbf{0}$, we have

$$\mathbf{v}^\top\mathbf{S}_{w,\varepsilon}\mathbf{v} = \mathbf{v}^\top\mathbf{S}_w\mathbf{v} + \varepsilon\|\mathbf{v}\|_2^2 \geq \varepsilon\|\mathbf{v}\|_2^2 > 0.$$

So the existence of the Cholesky decomposition is warranted.

4. Computational considerations

- A feasible rule of thumb is to set ε proportional to the largest eigenvalue of $\mathbf{S}_w$, which can be efficiently calculated by the power method, and set

$$\varepsilon = \tau \lambda_{\max}(\mathbf{S}_w),$$

  where $\tau > 0$ is a small parameter, e.g., $\tau = 10^{-10}$.

- Since only the eigenvectors associated with a few of the dominant eigenvalues are needed, the computations use the power method, and only products of vectors with the matrix $\mathbf{K}^{-\top}\mathbf{S}_b\mathbf{K}^{-1}$ need to be computed, not the matrix itself. The computation of a product of the form $\mathbf{p} = \mathbf{K}^{-\top}\mathbf{S}_b\mathbf{K}^{-1}\mathbf{z}$ can be performed by first solving the triangular linear system $\mathbf{K}\mathbf{y} = \mathbf{z}$ for $\mathbf{y}$, then solving the triangular linear system $\mathbf{K}^\top\mathbf{p} = \mathbf{S}_b\mathbf{y}$. In this manner the inverse of $\mathbf{K}$ is never explicitly computed, and the computation of each product only requires the solution of two triangular linear systems.

- Another important question is how many LDA separating directions should be computed. If we have k clusters, then $\mathbf{S}_b$ is the sum of k rank 1 matrices. However, since the data are centered, the k cluster centers define a subspace of dimensions $k-1$, and therefore,

$$\text{rank}(\mathbf{S}_b) \leq k-1 \quad \text{(proof?)}$$

and

$$\text{rank}(\mathbf{K}^{-\top}\mathbf{S}_b\mathbf{K}^{-1}) = \text{rank}(\mathbf{S}_b) \leq k-1.$$

Therefore, the number of nonzero eigenvalues of the matrix is at most $k-1$, and not only is there no point in looking for more than $k-1$ eigenvectors, but, in fact, one needs to be aware that the subsequent eigendirections are associated with zero eigenvalues, hence projection of the data along them may be misleading.

Algorithm: Linear discriminant analysis (LDA)

1. Given the data partitioned into k cluster matrices, $\mathbf{X}_1, \cdots, \mathbf{X}_k$
2. Compute the cluster means $\mathbf{c}_1, \cdots, \mathbf{c}_k$ and the global mean $\mathbf{c}$.
3. Compute the within-cluster scatter matrix $\mathbf{S}_w$
4. Compute the between-cluster scatter matrix $\mathbf{S}_b$
5. Compute the largest eigenvalue $\lambda_{\max}$ of $\mathbf{S}_w$, and set $\varepsilon = \tau\lambda_{\max}$ with a small $\tau > 0$, e.g., $\tau = 10^{-10}$.
6. Compute the Cholesky factor $\mathbf{K}$ satisfying $\mathbf{S}_w + \varepsilon\mathbf{I}_n = \mathbf{K}^\top\mathbf{K}$.
7. Compute the $k-1$ largest eigenpairs $(\lambda_j, \mathbf{w}^{(j)})$ of $\mathbf{K}^{-\top}\mathbf{S}_b\mathbf{K}^{-1}$.
8. Compute the eigenvectors $\mathbf{q}^{(j)}$ of the matrix $(\mathbf{S}_w + \varepsilon\mathbf{I}_n)^{-1}\mathbf{S}_b$ by $\mathbf{K}\mathbf{q}^{(j)} = \mathbf{w}^{(j)}$.
9. Compute the LDA-reduced data matrices $\mathbf{Z}_\ell = \mathbf{Q}^\top\mathbf{X}_\ell$ where $\mathbf{Q} = \begin{bmatrix}\mathbf{q}^{(1)} & \cdots & \mathbf{q}^{(k-1)}\end{bmatrix}$.

- The LDA-reduced data can be visualized by plotting either one-dimensional histograms or two-dimensional pairwise scatter plots of the LDA components.
- The geometric interpretation of LDA plots is not as straightforward as that of PCA plots. The PCA feature vectors are by construction mutually orthogonal, and therefore the PCA scatter plots are simply shadow images of the data on the planes spanned by pairs of the feature vectors.
- In the LDA algorithm, the vectors $\mathbf{w}^{(j)}$ are eigenvectors of a symmetric matrix $\mathbf{K}^{-\top}\mathbf{S}_b\mathbf{K}^{-1}$, thus mutually orthogonal. However, there is no guarantee of mutual orthogonality of vectors $\mathbf{q}^{(j)}$; in fact, usually they are not orthogonal. Consequently, the LDA components $\mathbf{z} = \mathbf{Q}^\top\mathbf{x}$ are not the components in the basis $\mathbf{Q}$ of the vector $\mathbf{x}$ projected on the plane spanned by the columns of $\mathbf{Q}$, but pairs of individual orthogonal projections on the vectors $\mathbf{q}^{(j)}$.

- To establish the connection, the data $\mathbf{X}$ can be projected orthogonally onto the subspace spanned by the LDA vectors,

$$\mathcal{H} = \text{span}\{\mathbf{q}^{(1)}, \cdots, \mathbf{q}^{(k-1)}\}.$$

  The orthogonal projection is given by

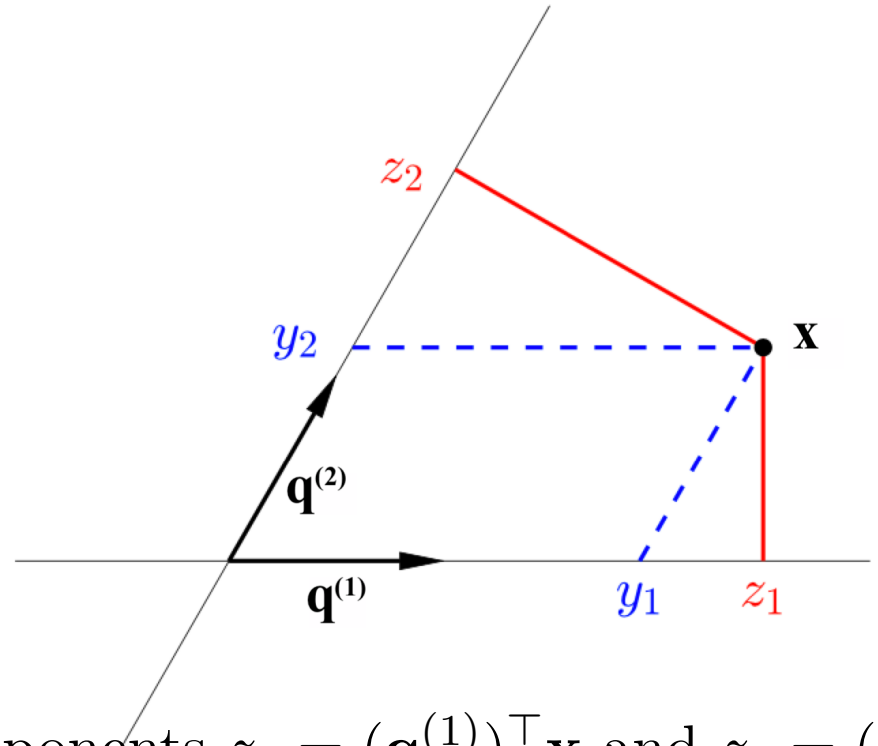
$$\mathbf{PX} = \mathbf{Q}(\mathbf{Q}^\top \mathbf{Q})^{-1}\mathbf{QX} = \mathbf{QY},$$

  where the coordinates with respect to the basis $\mathbf{Q}$ of $\mathcal{H}$ are

$$\mathbf{Y} = (\mathbf{Q}^\top \mathbf{Q})^{-1}\mathbf{QX} = (\mathbf{Q}^\top \mathbf{Q})^{-1}\mathbf{Z}.$$

- The difference between the LDA components and the components of the projection in the basis of the columns $\mathbf{Q}$ is illustrated graphically in next page..

- Schematic picture of the LDA projections when $\mathbf{x}$ lies in the plane spanned by the vectors $\mathbf{q}^{(1)}$ and $\mathbf{q}^{(2)}$.

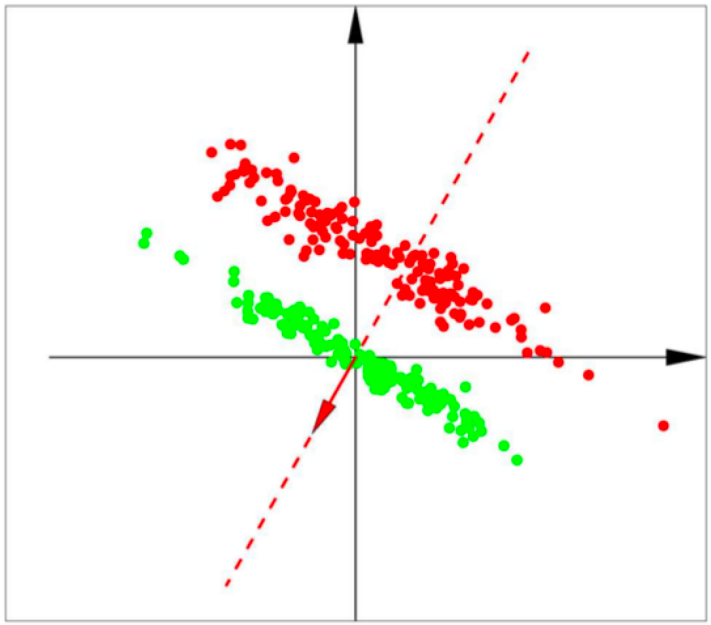


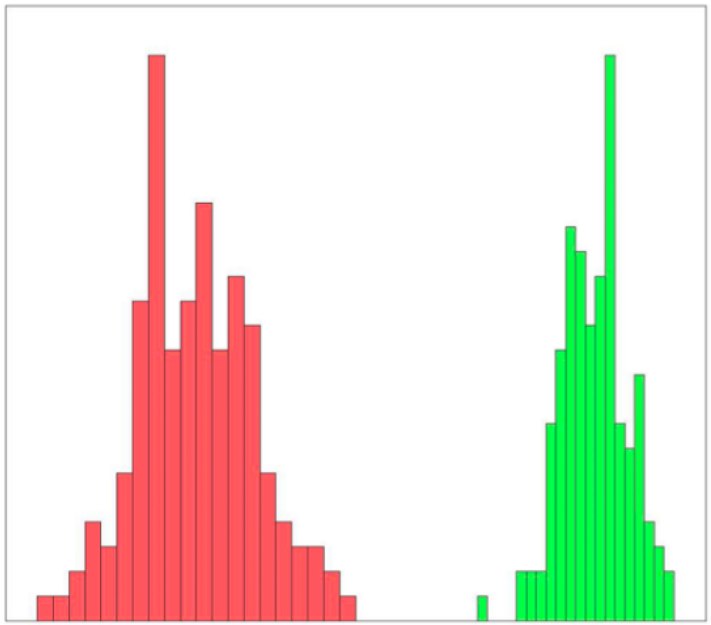
The LDA components $z_1 = (\mathbf{q}^{(1)})^\top \mathbf{x}$ and $z_2 = (\mathbf{q}^{(2)})^\top \mathbf{x}$ correspond to orthogonal projections of $\mathbf{x}$ onto these directions, while the coordinates y_1 and y_2 are the coordinates of $\mathbf{x}$ in the basis $\{\mathbf{q}^{(1)}, \mathbf{q}^{(2)}\}$, that is,

$$\mathbf{x} = y_1 \mathbf{q}^{(1)} + y_2 \mathbf{q}^{(2)}.$$

5. Computed examples

- Example 1: A simple low-dimensional example to demonstrate the idea of the LDA analysis. The red arrow is the principal LDA direction.





- Example 2: This example demonstrates the viability of the LDA algorithm to see cluster structure in high-dimensional data that is difficult to visualize.

  (1). From the data set of handwritten digits, select the vectors corresponding to three digits that are easy to confound, e.g., "0", "6", and "9".

  (2). Perform the PCA. Plot the first two principal components.

  (3). Compute the within-cluster centered data $\mathbf{X}_w$, form the scatter matrices $\mathbf{S}_w$ and $\mathbf{S}_b$.

  (4). Perform the LDA. Plot the scatter plot of the projections of the data onto the first two LDA directions.

  This example suggests that the LDA algorithm could be used for classifying data.

- Example 3. Application to facial data analysis. Yale face data set.