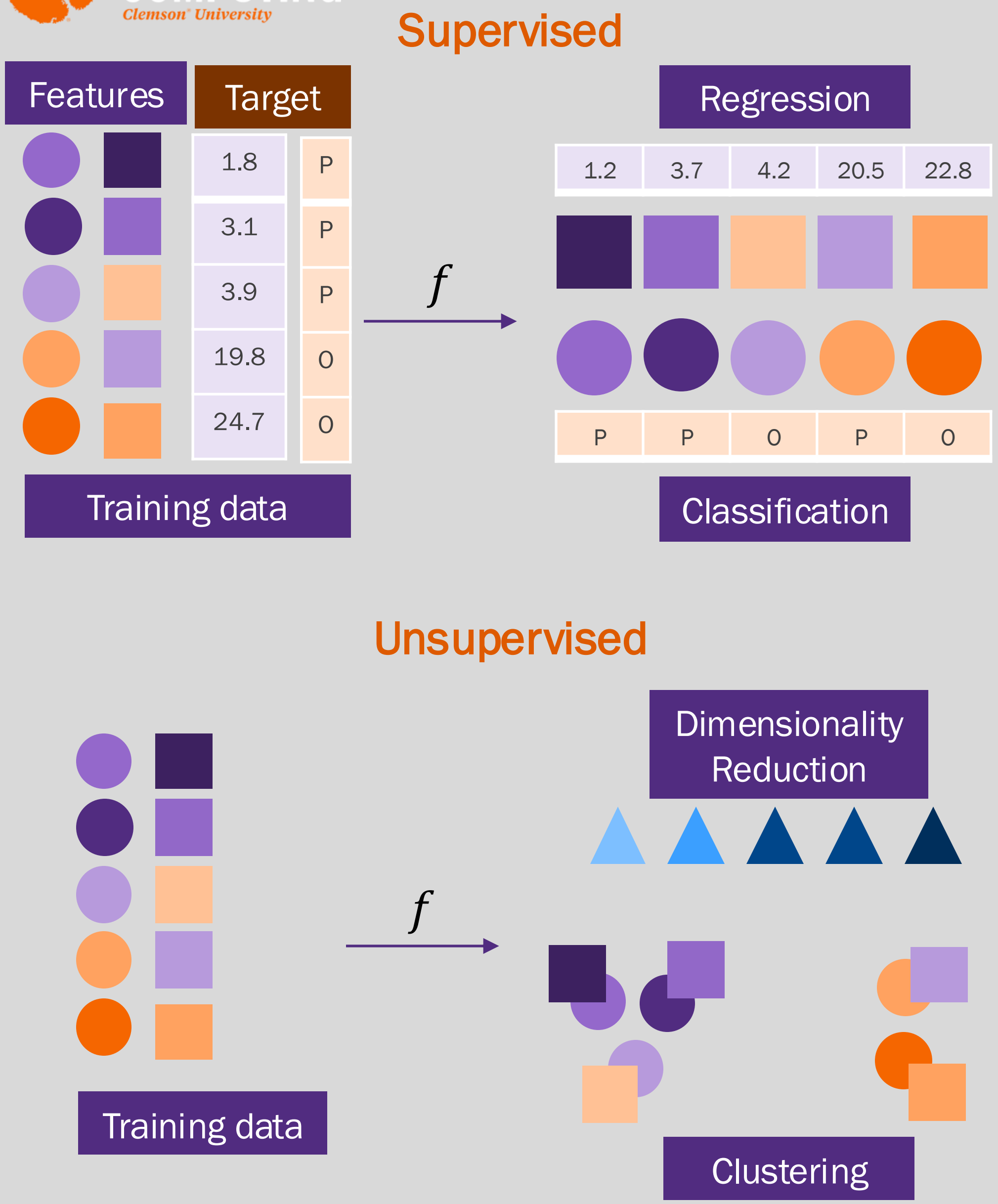
Dimensionality Reduction

Aaron J. Masino, PhD
Associate Professor, School of Computing

Outline

- Unsupervised learning review
- Principal components analysis
  - Method
  - Applications
- Other dimensionality reduction methods

Unsupervised learning

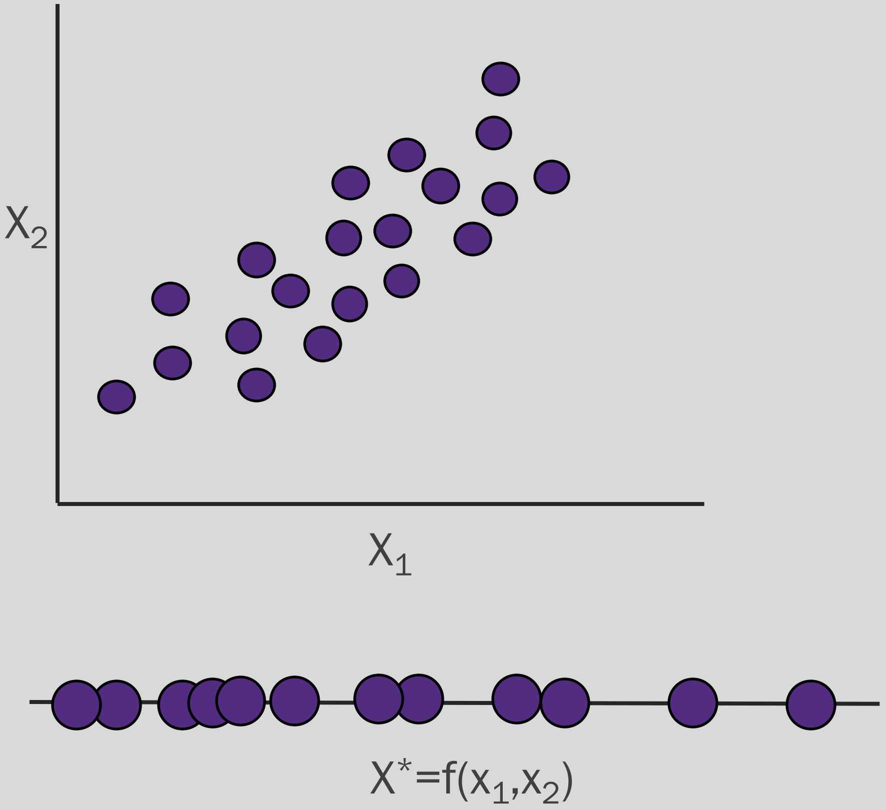


Unsupervised Learning

$$\boldsymbol{X} = \begin{bmatrix} x_{1,1}, x_{1,2}, \cdots, x_{1,p} \\ x_{2,1}, x_{2,2}, \cdots, x_{2,p} \\ \vdots \\ x_{n,1}, x_{n,2}, \cdots, x_{n,p} \end{bmatrix}$$

Posit $f(\boldsymbol{X}, \boldsymbol{\Theta}) = \widetilde{\boldsymbol{X}}$
subject to constraints $\boldsymbol{C}$

- Given an observed dataset
- <u>No target variable for prediction</u>
- Goal is to identify *interesting* characteristics in the observed data
- Most unsupervised learning goals are either:
  - *dimensionality reduction* – can be viewed as an unsupervised form of regression
  - *clustering* – can be viewed as an unsupervised form of classification



Dimensionality Reduction

- Goal: Find a lower dimensional representation that preserves information.
- Why?
  - Data visualization
  - Missing data imputation
  - More efficient supervised learning

Principal Components Analysis Method

Principal Components Analysis (PCA)

- Assume the observed data is composed of n samples each with p features
- The data will be transformed into n samples with $m < p$ features
- New dimensions (principal components) are:
  - Linear combinations of the original p features
  - Uncorrelated with each other (orthogonal)
  - Normalized (i.e., sum of squares of the coefficients is equal to one)
  - Can be ordered such that the higher components capture more variance of the original data than lower components
- Assume each column of the observed data is standardized so that total variance is given by

$$Var(\boldsymbol{X}) = \frac{1}{n}\sum_{j=1}^{p}\sum_{i=1}^{n} x_{i,j}^2$$

We seek principal components that explain the variance in the data

Principal Components Analysis (PCA)

- The principal component transformations, $\boldsymbol{Z_m}$ take the form

$$\boldsymbol{Z_1} = f(\boldsymbol{X}|\boldsymbol{\phi_1}) = \phi_{1,1}\boldsymbol{X_1} + \phi_{2,1}\boldsymbol{X_2} + \cdots + \phi_{p,1}\boldsymbol{X_p}$$

$$\vdots$$

$$\boldsymbol{Z_m} = f(\boldsymbol{X}|\boldsymbol{\phi_m}) = \phi_{1,m}\boldsymbol{X_1} + \phi_{2,m}\boldsymbol{X_2} + \cdots + \phi_{p,m}\boldsymbol{X_p}$$

- The vector, $\boldsymbol{\phi}_m = \left[\phi_{1,m}, \phi_{2,m}, \cdots, \phi_{p,m}\right]$ are the loadings for the m^{th} principal component
- The vector $\boldsymbol{Z_m}$ is the score for the m^{th} principal component
- There are at most $min(n-1, p)$ principal components
- In matrix form, we have the projections of $\boldsymbol{X}$ onto the principal components given by the $n \times m$ matrix

$$\boldsymbol{Z} = \boldsymbol{X\phi}$$

where $\boldsymbol{\phi}$ is the $p \times m$ matrix of loading values

Principal Components Analysis

- How do we find the principal components?
- We want the loading vectors $\boldsymbol{\phi}_m = [\phi_{1,m}, \phi_{2,m}, \cdots, \phi_{p,m}]$ for $m \in [1, M]$that, for each m, maximize

$$\frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{j,m}x_{ij}\right)^2$$

  subject to constraints $\sum_{j=1}^{p}\phi_{j,m}^2 = 1$ and $\boldsymbol{\phi}_j \cdot \boldsymbol{\phi}_k = 0 \; \forall j \neq k$
- This can be formulated as an eigenvalue problem where we find the SVD $\boldsymbol{S} = \boldsymbol{\phi}^T \boldsymbol{\Lambda} \boldsymbol{\phi}$ of the covariance matrix

$$\boldsymbol{S} = \frac{1}{n-1}\boldsymbol{X}^T\boldsymbol{X}$$

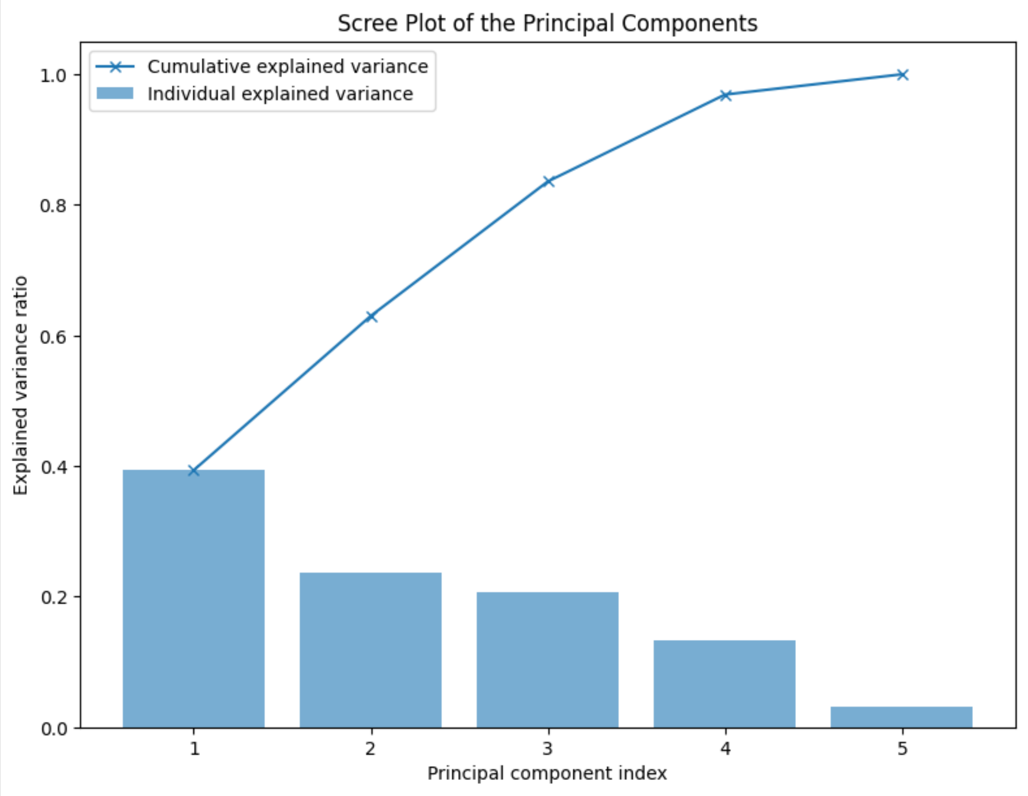
  Ordering the eigenvectors by their corresponding eigenvalues gives the ordered principal components

Principal Components Analysis

- What proportion of the variance is explained by each principal component?
- After some math, we find that the variance explained by m^{th} principal component is given by

$$\frac{\sum_{i=1}^{n} z_{i,m}^2}{\sum_{j=1}^{p} \sum_{i=1}^{n} x_{i,j}^2}$$

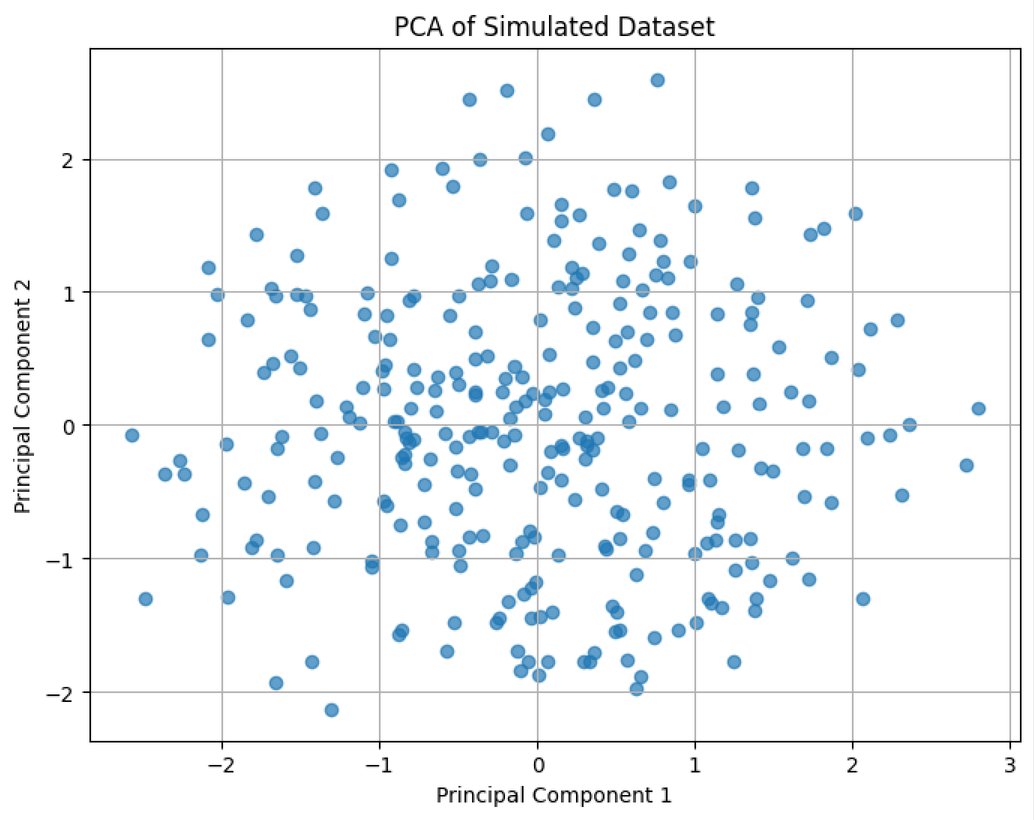
- In applications, we would want most of the variance to be explained by the first few principal components



Principal components analysis Applications

Data visualization

- Given an observed dataset with greater than 3 features, we cannot view the data in a single plot
- With PCA we can transform the original p dimensional data, $\boldsymbol{X}$, to 2 or 3 dimensions in $\boldsymbol{Z}$ for plotting
- Provided the first 2 or 3 dimensions account for a *reasonable* amount of the data variance, the plot is likely to reveal interesting characteristics of the data



Data imputation

- Assume we have n observations of samples defined by p features where for a given sample some of the features are missing *at random*
- We can *impute* estimates for the missing samples with PCA

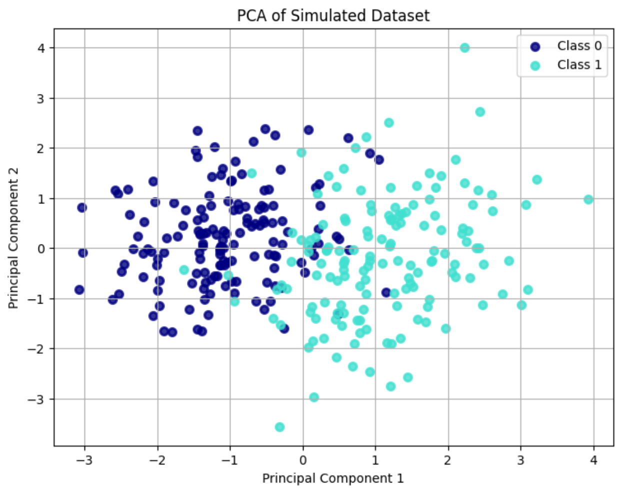
Imputation with PCA

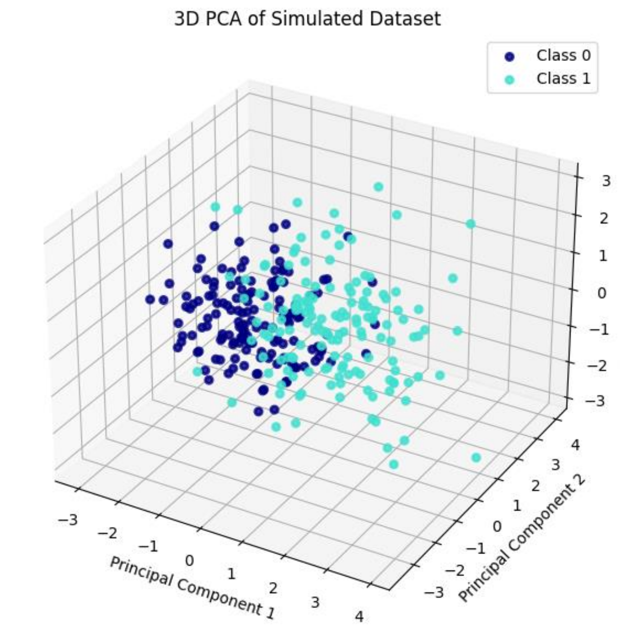
1. Create a completed data matrix $\widetilde{\boldsymbol{X}}$ of dimension $n \times p$ by filling in missing values of a given feature with the mean of the observed values of that feature
2. Repeats steps (a)-(c) until the objective does not decrease
   a) Compute the principal components of $\widetilde{\boldsymbol{X}}$
   b) Replace the missing elements of X with $\tilde{x}_{i,j} \leftarrow \sum_{m=1}^{M} z_{i,m}\phi_{j,m}$ to form a new $\widetilde{\boldsymbol{X}}$
   c) Compute the objective over the complete samples, $\mathcal{O}$,

$$\sum_{(i,j)\epsilon\mathcal{O}} \left(x_{i,j} - \sum_{m=1}^{M} z_{i,m}\phi_{j,m} \right)^2$$

Improved supervised learning

- PCA is often applied in supervised learning when the data is noisy or there are many more features than samples available for training
- If the input features truly have a linear relationship with the target value (*regression)* or there is a linear boundary (*classification*) and the first few principal components capture a significant portion of the data variance:
    - PCA transformation will make the supervised learning problem easier to solve
    - Explainability is typically reduced however, as the principal components are not always easy to interpret





PCA – How many components to keep?

$k = 1$

$k = 2$

$k = 4$

$k = 8$

$k = 16$

$k = 32$

$k = 50$

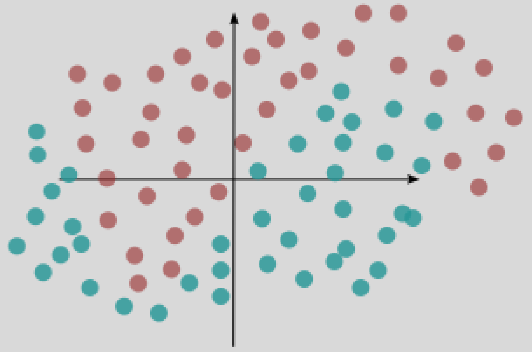
$k = 64$

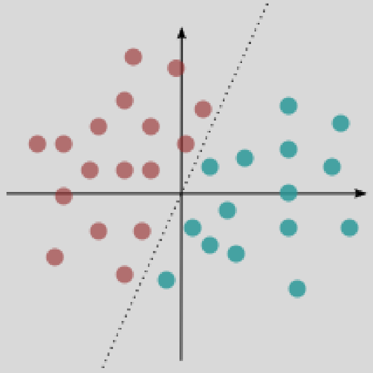
original

Other dimensionality reduction methods

Kernel PCA

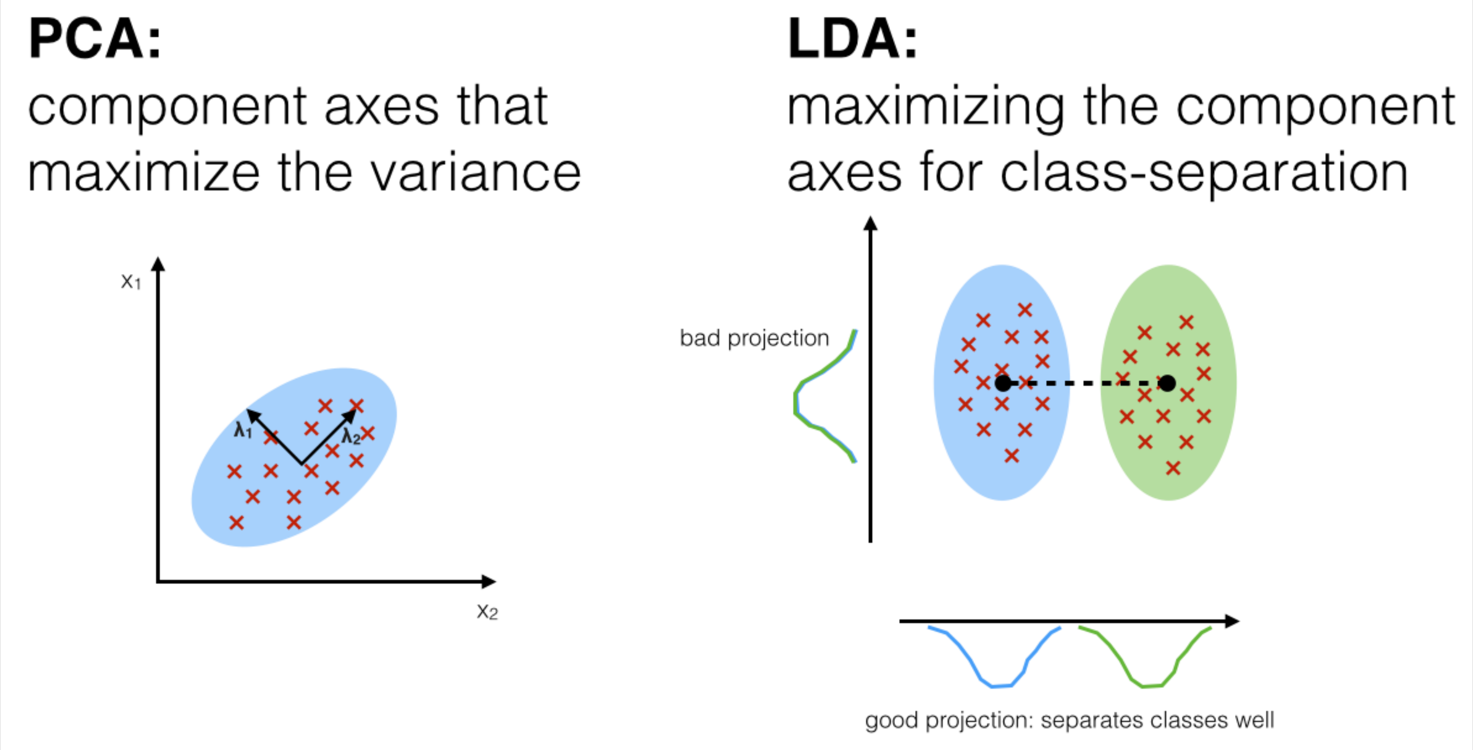
- Standard PCA is a linear transformation
- Kernel PCA extends PCA to handle non-linear data by employing kernel functions
- Compute the eigenvectors of the kernel matrix derived from applying a kernel function to the original data (Sigmoid, GaussRBF, etc.)
- The kernel matrix contains pairwise similarities between data points in a higher-dimensional feature space induced by the kernel function.
- The principal components obtained from Kernel PCA are non-linear combinations of the original features





Linear discriminant analysis (LDA)

- Similar to PCA but leverages available class membership information
- Seeks to identify components that account for data variance while also maximizing class separation



LDA Method

1. Class Means: For each class in the dataset, calculate the mean vector (average value of each feature for that class)
2. Within-Class Scatter Matrix: Sum of the outer product of the differences between each data point and its class mean. Represents deviation of samples from class mean.
3. Between-Class Scatter Matrix: Sum of the outer product of the differences between each class mean and the overall mean.
4. Eigenvectors and Eigenvalues: Find the eigenvectors and corresponding eigenvalues of the matrix resulting from the inverse of the within-class scatter matrix multiplied by the between-class scatter matrix.
   - eigenvectors represent the directions (or axes) along which the data is best separated
   - eigenvalues represent the amount of variance explained by each eigenvector.

Scatter Matrix

Given n samples of p dimensional data

$$\overline{\boldsymbol{x}} = \frac{1}{n}\sum_{j=1}^{n} \boldsymbol{x}_j$$

where $\boldsymbol{x}_j$ is the j^{th} sample

The scatter matrix, $\boldsymbol{S}$, is the $p \times p$ matrix

$$\boldsymbol{S} = \sum_{j=1}^{n} (\boldsymbol{x}_j - \overline{\boldsymbol{x}}) \otimes (\boldsymbol{x}_j - \overline{\boldsymbol{x}})$$

where the outer product, $\otimes$, is defined as:

$$\boldsymbol{u} = [u_1, \cdots, u_m]$$

$$\boldsymbol{v} = [v_1, \cdots, v_n]$$

$$\boldsymbol{u} \otimes v = \begin{bmatrix} u_1 v_1 & u_1 v_2 & \cdots & u_1 v_n \\ u_2 v_1 & u_2 v_2 & \cdots & u_2 v_n \\ \vdots & \vdots & \ddots & \vdots \\ u_m v_1 & u_m v_2 & \cdots & u_m v_n \end{bmatrix}$$