So Far…

- It's time for
  - Unsupervised learning
    - We are only given inputs
    - Goal: find "interesting patterns"

    - Discovering clusters
      - Clustering
    - Discovering latent factors
      - Dimensionality reduction
      - Topic modeling
      - Matrix factorization

Dimensionality Reduction

Deng Cai (蔡登)

College of Computer Science
Zhejiang University

dengcai@gmail.com

Feature Representation

- For all the learning tasks (supervised, unsupervised), we need $\boldsymbol{x}$
- Better representation makes learning easier
- Minimum requirement:
  - $\boldsymbol{x}$ should contains relevant features
- But we don't know which features are useful.
  - As many features as possible
- Feature engineering problem:
  - Dimensionality reduction

What is Dimensionality Reduction?

- The Key:
  - Feature mapping from $\boldsymbol{x}$ to $\boldsymbol{z}$
  - The $\boldsymbol{x}$ is the original representation, usually with high dimensionality.
  - We believe the number of latent factors (degree of the freedoms) of the data is far less.
    - Handwritten digits example
  - Thus, the dimensionality of $\boldsymbol{z}$ is **usually** smaller than that of $\boldsymbol{x}$
  - This is the name DR comes from.

Linear and Nonlinear

$$\mathcal{F}(\boldsymbol{x} \in R^p) = \boldsymbol{z} \in R^d$$

$$f_1(\boldsymbol{x}) = z_1 \qquad f_i(\boldsymbol{x}) = z_i \qquad f_d(\boldsymbol{x}) = z_d$$

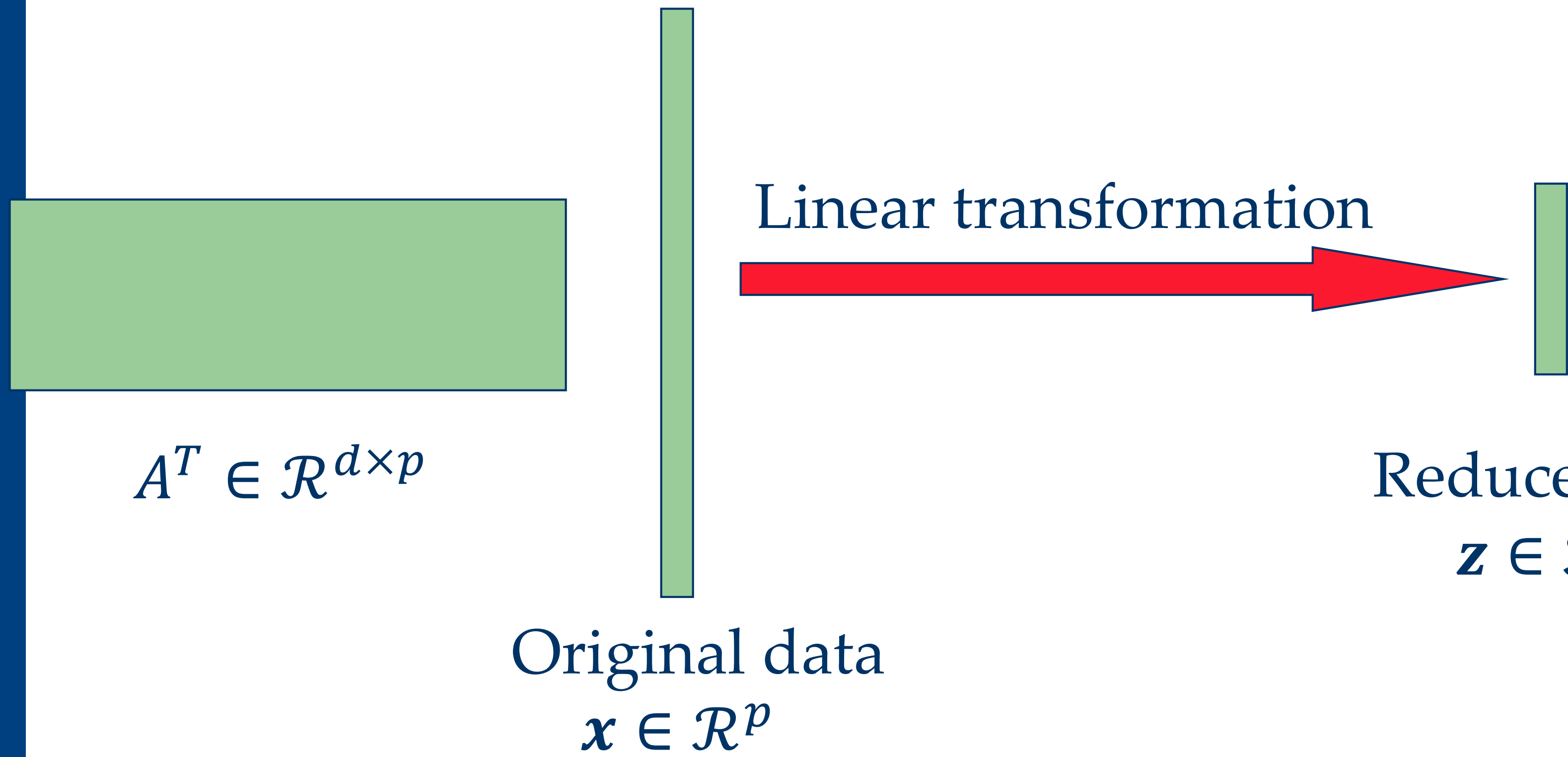
- All the methods (classification & clustering) can be seen as a DR approach (either supervised or unsupervised)
- If f is linear, linear dimensionality reduction

$$\boldsymbol{a}_1^T\boldsymbol{x} = z_1 \qquad \boldsymbol{a}_i^T\boldsymbol{x} = z_i$$

$$A \triangleq [\boldsymbol{a}_1, \cdots, \boldsymbol{a}_d] \qquad A^T\boldsymbol{x} = \boldsymbol{z}$$

- If f is nonlinear, nonlinear dimensionality reduction
  - We know the embedding function
  - We don't know the function

Linear Transformation



$$A \in \mathcal{R}^{p \times d}: \boldsymbol{x} \in \mathcal{R}^p \rightarrow \boldsymbol{z} = A^T \boldsymbol{x} \in \mathcal{R}^d$$

Feature Extraction vs Feature Selection

- Dimensionality reduction (Feature reduction)
  - Feature extraction
  - Feature selection

- Selection: choose a best subset of size d from the available p features

- Extraction: given p features (set X), extract d new features (set Z) by linear or non-linear combination of all the p features

$$A \in \mathcal{R}^{p \times d}: \boldsymbol{x} \in \mathcal{R}^p \rightarrow \boldsymbol{z} = A^T \boldsymbol{x} \in \mathcal{R}^d$$

- Selection: $A \in [0,1]^{p \times d}$, every column of A has only one 1.

- Extraction: $A \in \mathcal{R}^{p \times d}$

Dimensionality Reduction Algorithms

- Unsupervised
  - Latent Semantic Indexing (LSI): truncated SVD
  - Principal Component Analysis (PCA)
  - Independent Component Analysis (ICA)
  - Canonical Correlation Analysis (CCA)
- Supervised
  - Linear Discriminant Analysis (LDA)
- Semi-supervised
  - Semi-supervised Discriminant Analysis (SDA)

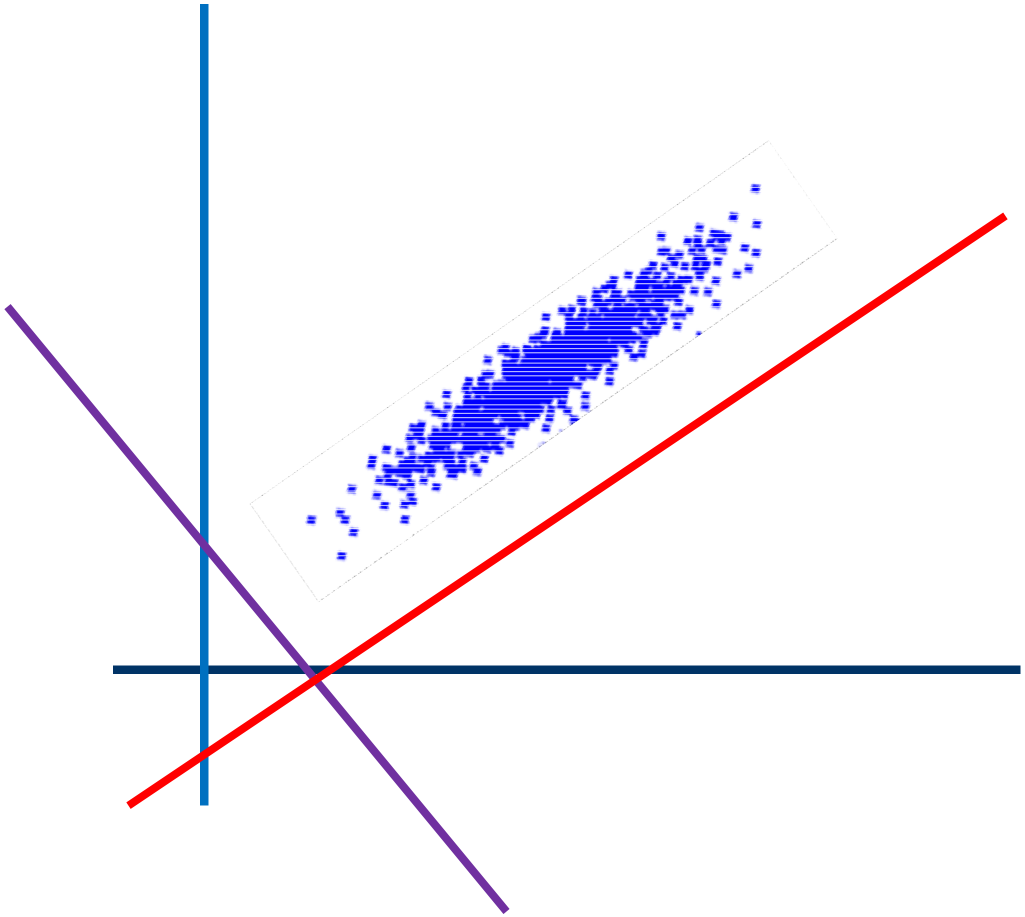
Dimensionality Reduction Algorithms

- Linear
  - Latent Semantic Indexing (LSI): truncated SVD
  - Principal Component Analysis (PCA)
  - Linear Discriminant Analysis (LDA)
  - Canonical Correlation Analysis (CCA)
- Nonlinear
  - Nonlinear feature reduction using kernels
  - Manifold learning

Algorithms

- Principal Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)
- Locality Preserving Projections (LPP)
- The framework of graph based dimensionality reduction.
- Laplacian Eigenmap

Principal Component Analysis



What is Principal Component Analysis?

- Principal component analysis (PCA)
  - Reduce the dimensionality of a data set by finding a new set of variables, smaller than the original set of variables
  - Retains most of the sample's information.
  - Useful for the compression and classification of data.

- By information we mean the variation present in the sample, given by the correlations between the original variables.
  - The new variables, called principal components (PCs), are uncorrelated, and are ordered by the fraction of the total information each retains.

Algebraic Derivation of PCs

- Given a sample of n observations on a vector of p variables

$$\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\} \in \mathcal{R}^p$$

- Define the first principal component of the sample by the linear transformation

$$z_i^{(1)} = \boldsymbol{a}_1^T \boldsymbol{x}_i, \qquad i = 1, \cdots n$$

is chosen such that $var\left(z^{(1)}\right)$ is maximum.

Algebraic Derivation of PCs

$$var(z^{(1)}) = E\left(\left(z^{(1)} - \bar{z}^{(1)}\right)^2\right)$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{a}_1^T\boldsymbol{x}_i - \boldsymbol{a}_1^T\overline{\boldsymbol{x}}\right)^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{a}_1^T(\boldsymbol{x}_i - \overline{\boldsymbol{x}})(\boldsymbol{x}_i - \overline{\boldsymbol{x}})^T\boldsymbol{a}_1 = \boldsymbol{a}_1^T S \boldsymbol{a}_1$$

Where $S = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \overline{\boldsymbol{x}})(\boldsymbol{x}_i - \overline{\boldsymbol{x}})^T$

is the covariance matrix and $\overline{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i$ is the mean.

Algebraic Derivation of PCs

$$\max_{\boldsymbol{a}_1} \boldsymbol{a}_1^T S \boldsymbol{a}_1$$

$$s.t. \quad \boldsymbol{a}_1^T \boldsymbol{a}_1 = 1$$

Let λ be a Lagrange multiplier

$$L = \boldsymbol{a}_1^T S \boldsymbol{a}_1 - \lambda\left(\boldsymbol{a}_1^T \boldsymbol{a}_1 - 1\right)$$

$$\frac{\partial L}{\partial \boldsymbol{a}_1} = 2S\boldsymbol{a}_1 - 2\lambda \boldsymbol{a}_1 = 0$$

$$S\boldsymbol{a}_1 = \lambda \boldsymbol{a}_1$$

therefor, $\boldsymbol{a}_1$is an eigenvector of S corresponding to the largest eigenvalue $\lambda = \lambda_1$.

Algebraic Derivation of PCs

$$\max_{\boldsymbol{a}_2} \boldsymbol{a}_2^T S \boldsymbol{a}_2$$

$$s.t. \quad \boldsymbol{a}_2^T \boldsymbol{a}_2 = 1, \mathrm{cov}\left(z^{(2)}, z^{(1)}\right) = 0$$

$$\mathrm{cov}\left(z^{(2)}, z^{(1)}\right) = \boldsymbol{a}_2^T S \boldsymbol{a}_1 = \lambda \boldsymbol{a}_2^T \boldsymbol{a}_1 = 0$$

$$S\boldsymbol{a}_2 = \lambda \boldsymbol{a}_2$$

$\boldsymbol{a}_2$ is an eigenvector of S corresponding to the second largest eigenvalue $\lambda = \lambda_2$.

Algebraic Derivation of PCs

- In general:

$$var\left(z^{(k)}\right) = \boldsymbol{a}_k^T S \boldsymbol{a}_k = \lambda_k$$

- The k^{th} largest eigenvalue of S is the variance of k^{th} PC.
- The k^{th} PC $z^{(k)}$ retains the k^{th} greatest fraction of the variation in the sample.

Principle Component Analysis

- Main steps for computing PCs:
  - Form the covariance matrix S.
  - Compute its eigenvectors: $\{\boldsymbol{a}_i\}_{i=1}^{p}$
  - Use the first d eigenvectors $\{\boldsymbol{a}_i\}_{i=1}^{d}$ to form the d PCs.
  - The transformation A is given by

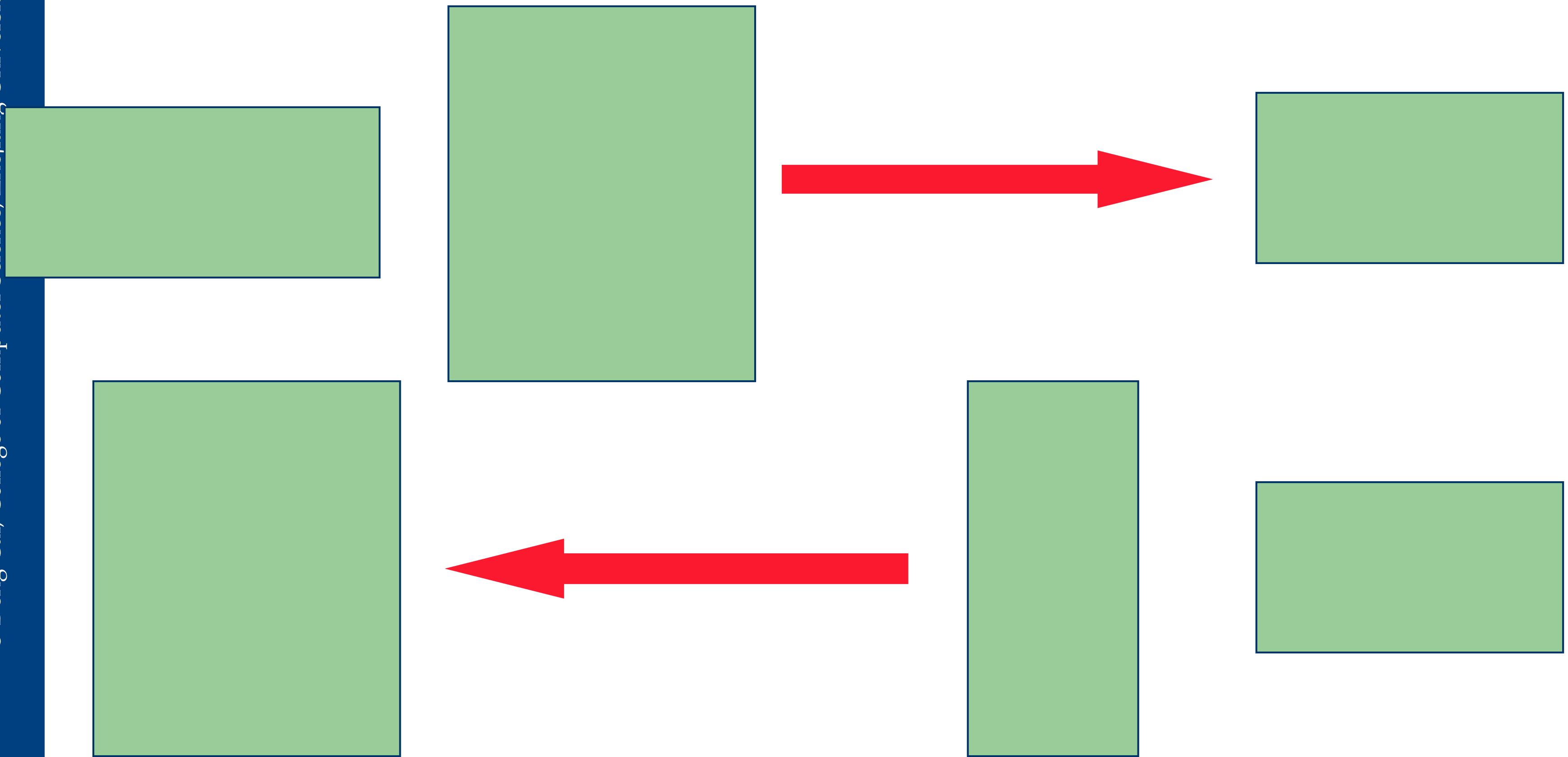
$$A = [\boldsymbol{a}_1, \cdots \boldsymbol{a}_d]$$

- A test point $\boldsymbol{x} \in \mathcal{R}^p \rightarrow A^T\boldsymbol{x} \in \mathcal{R}^d$

Optimality Property of PCA

Reconstruction

- Dimension reduction: $X \in \mathcal{R}^{p \times n} \to A^T X \in \mathcal{R}^{d \times n}$
- Original data: $A^T X \in \mathcal{R}^{d \times n} \to \bar{X} = A(A^T X) \in \mathcal{R}^{p \times n}$



Optimality Property of PCA

- Main theoretical result:
- The matrix A consisting of the first d eigenvectors of the covariance matrix S solves the following optimization problem:

$$\min_{A \in \mathcal{R}^{p \times d}} \left\| X - AA^T X \right\|_F^2 \quad s.t. A^T A = I_d$$

$$\left\| X - \hat{X} \right\|_F^2$$

$\boxed{\text{Reconstruction error}}$

- X is centered data matrix
- PCA projection minimizes the reconstruction error among all linear projections of size d.

PCA for Image Compression

d=1

d=2

d=4

d=8

d=16

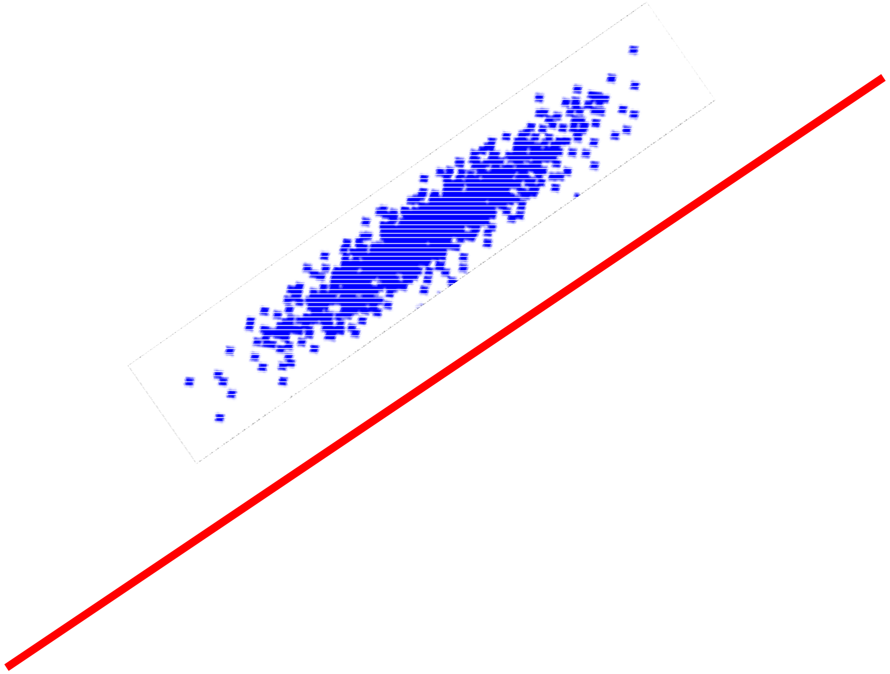
d=32

d=64

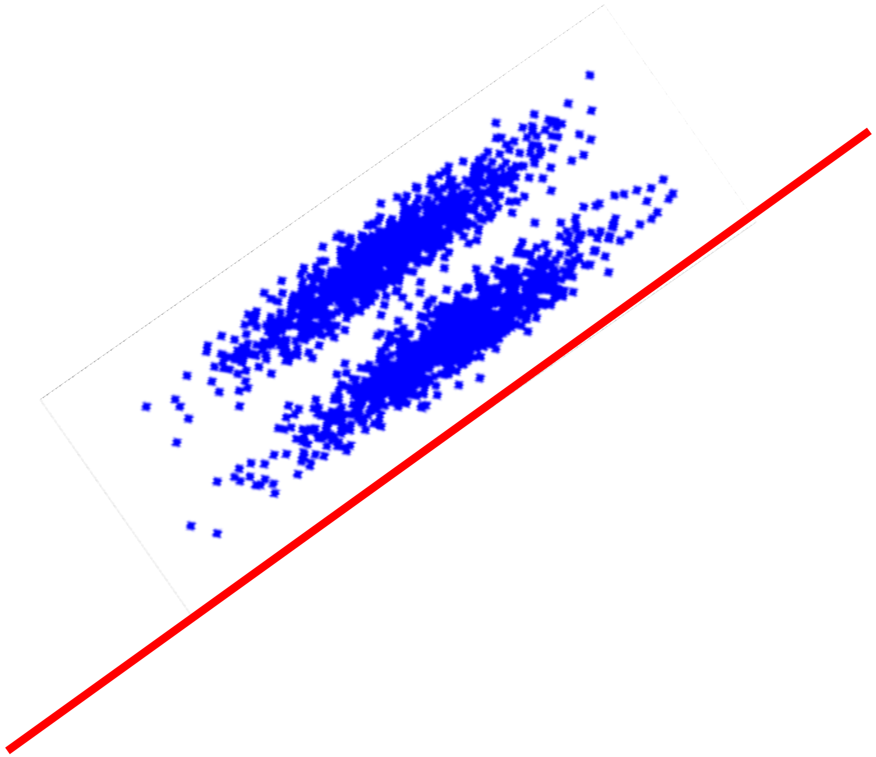
d=100

Original Image

Principal Component Analysis

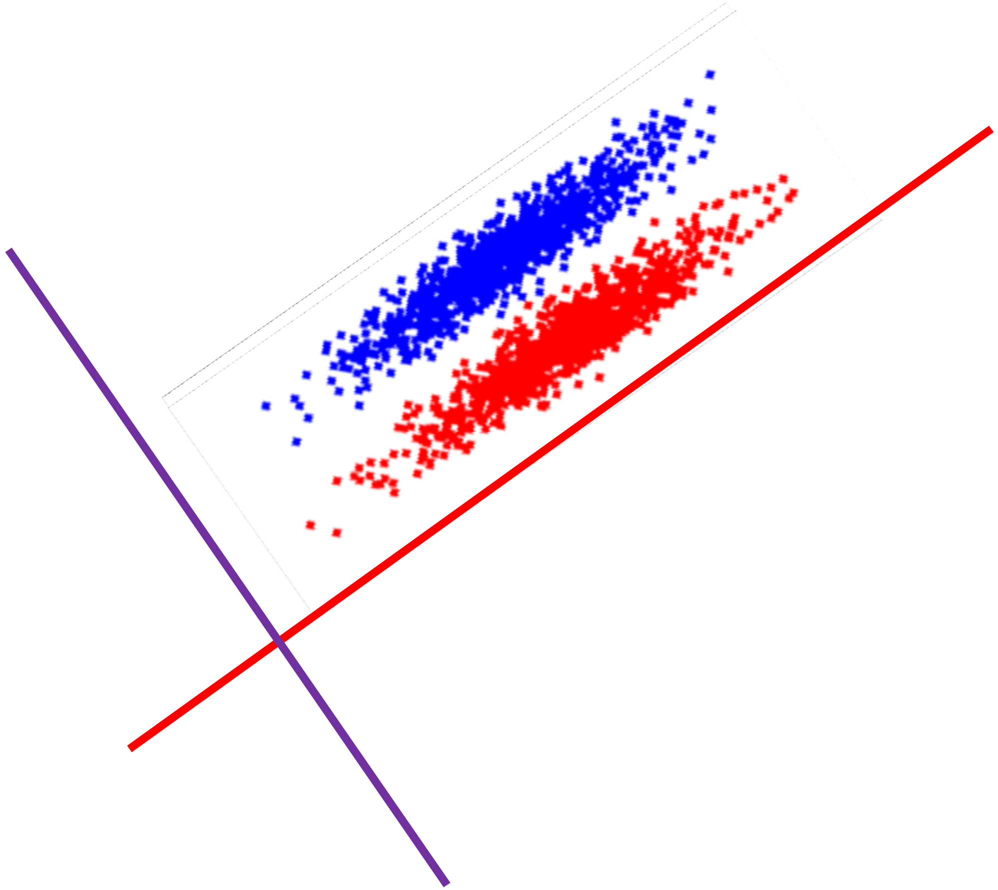


Principal Component Analysis

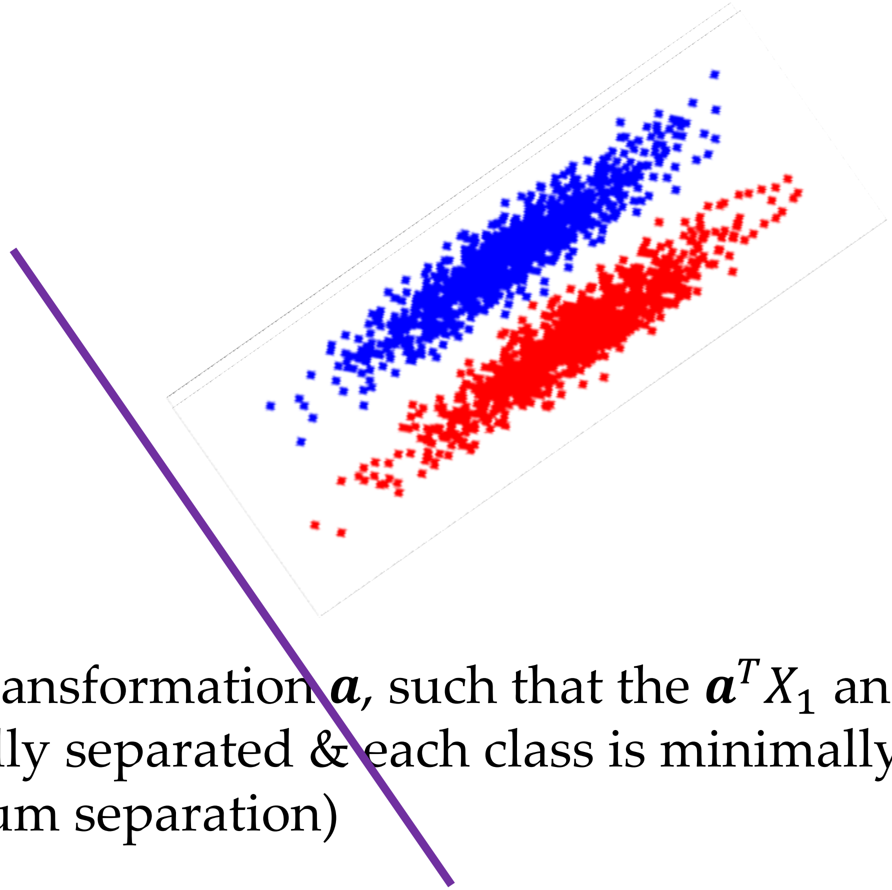


- Find a transformation $\boldsymbol{a}$, such that the $\boldsymbol{a}^T X$ $w^T x$ is dispersed the most (maximum distribution)

Principal Component Analysis



Linear Discriminant Analysis (Fisher Linear Discriminant)

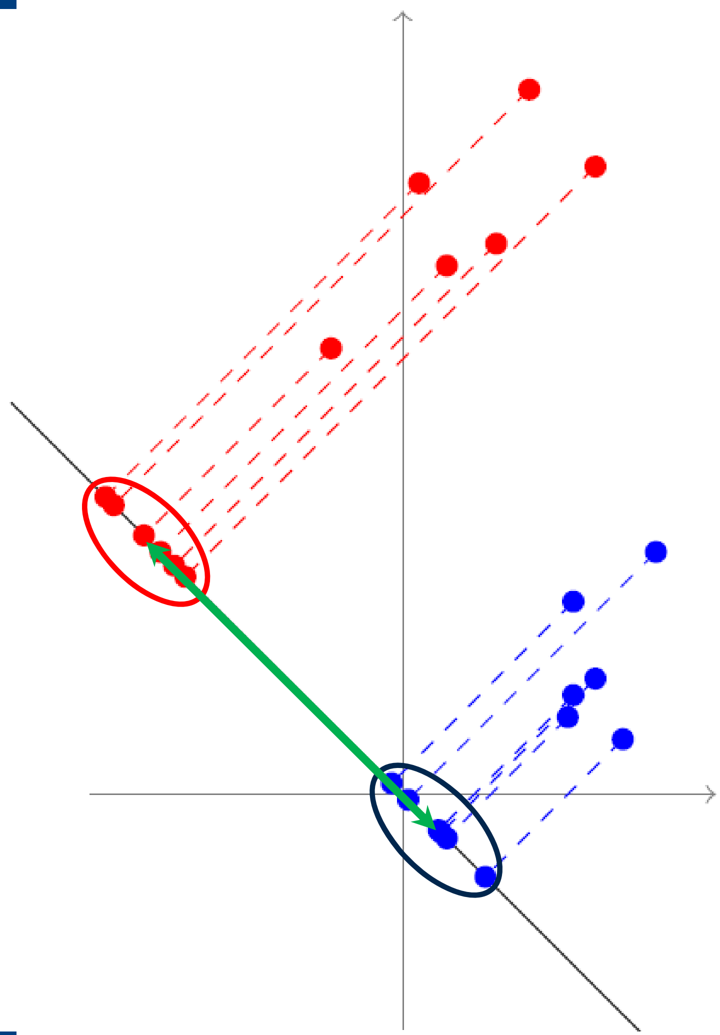


- Find a transformation $\boldsymbol{a}$, such that the $\boldsymbol{a}^T X_1$ and $\boldsymbol{a}^T X_2$ are maximally separated & each class is minimally dispersed (maximum separation)

Linear Discriminant Analysis

- Perform dimensionality reduction “while preserving as much of the class discriminatory information as possible”.
- Seeks to find directions along which the classes are best separated.
- Takes into consideration the scatter within-classes but also the scatter between-classes.

Linear Discriminant Analysis



- **Two Classes $\boldsymbol{\omega_1}, \boldsymbol{\omega_2}$**

$$z = \boldsymbol{a}^T \boldsymbol{x}$$

$$\tilde{\mu}_i = \frac{1}{n_i} \sum_{z \in \omega_i} z$$

$$\boldsymbol{\mu}_i = \frac{1}{n_i} \sum_{\boldsymbol{x} \in \omega_i} \boldsymbol{x}, \tilde{\mu}_i = \boldsymbol{a}^T \boldsymbol{\mu}_i$$

$$|\tilde{\mu}_1 - \tilde{\mu}_2| = \left| \boldsymbol{a}^T (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \right|$$

$$\tilde{s}_i^2 = \sum_{z \in \omega_i} (z - \tilde{\mu}_i)^2$$

$$\frac{1}{n} \left(\tilde{s}_1^2 + \tilde{s}_2^2 \right)$$

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

Linear Discriminant Analysis

- **Two Classes**

$$J(\boldsymbol{a}) = \frac{(\tilde{\mu}_1 - \tilde{\mu}_2)^2}{\tilde{s}_1^2 + \tilde{s}_2^2} = \frac{\boldsymbol{a}^T S_B \boldsymbol{a}}{\boldsymbol{a}^T S_W \boldsymbol{a}}$$

$$\tilde{s}_i^2 = \sum_{y \in \omega_i} (y - \tilde{\mu}_i)^2 = \sum_{\boldsymbol{x} \in \omega_i} (\boldsymbol{a}^T \boldsymbol{x} - \boldsymbol{a}^T \boldsymbol{\mu}_i)^2 = \sum_{\boldsymbol{x} \in \omega_i} (\boldsymbol{a}^T \boldsymbol{x} - \boldsymbol{a}^T \boldsymbol{\mu}_i)(\boldsymbol{a}^T \boldsymbol{x} - \boldsymbol{a}^T \boldsymbol{\mu}_i)^T$$

$$= \boldsymbol{a}^T \sum_{\boldsymbol{x} \in \omega_i} (\boldsymbol{x} - \boldsymbol{\mu}_i)(\boldsymbol{x} - \boldsymbol{\mu}_i)^T \boldsymbol{a} \quad = \boldsymbol{a}^T S_i \boldsymbol{a}$$

S_i: *scatter matrix*

within-class scatter matrix: $S_W = S_1 + S_2 \qquad \tilde{s}_1^2 + \tilde{s}_2^2 = \boldsymbol{a}^T S_W \boldsymbol{a}$

$$(\tilde{\mu}_1 - \tilde{\mu}_2)^2 = (\boldsymbol{a}^T \boldsymbol{\mu}_1 - \boldsymbol{a}^T \boldsymbol{\mu}_2)^2 = \boldsymbol{a}^T (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \boldsymbol{a} \quad = \boldsymbol{a}^T S_B \boldsymbol{a}$$

between-class scatter matrix: $S_B = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T$

Linear Discriminant Analysis

- **Two Classes**

$$J(\boldsymbol{a}) = \frac{\boldsymbol{a}^T S_B \boldsymbol{a}}{\boldsymbol{a}^T S_W \boldsymbol{a}}$$

$$S_B \boldsymbol{a} = \lambda S_W \boldsymbol{a}$$

? Homework

$$\boldsymbol{a}^* = S_W^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

$$S_W = \sum_i^2 S_i = \sum_i^2 \sum_{x \in \omega_i} (\boldsymbol{x} - \boldsymbol{\mu}_i)(\boldsymbol{x} - \boldsymbol{\mu}_i)^T$$

- In case $P(\boldsymbol{x}|\omega_i)$: multivariate normal densities
- Case $\Sigma_i = \Sigma$
- $g_i(\boldsymbol{x}) = \boldsymbol{\mu}_i^T \Sigma^{-1} \boldsymbol{x} - \frac{1}{2} \boldsymbol{\mu}_i^T \Sigma^{-1} \boldsymbol{\mu}_i + \ln P(\omega_i)$
- $g_i(\boldsymbol{x}) = \boldsymbol{w}_i^T \boldsymbol{x} + w_{i0}$
- Two classes: $\boldsymbol{w} = \boldsymbol{w}_1 - \boldsymbol{w}_2 = \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$

$$\Sigma = \sum_{i=1}^2 \sum_{j \in \omega_i} \frac{(\boldsymbol{x}_j - \boldsymbol{\mu}_i)(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^T}{N}$$

Linear Discriminant Analysis

- Multi-classes

$$J(\boldsymbol{a}) = \frac{\boldsymbol{a}^T S_B \boldsymbol{a}}{\boldsymbol{a}^T S_W \boldsymbol{a}} \qquad \boldsymbol{\mu} = \frac{1}{n}\sum_{\boldsymbol{x}} \boldsymbol{x} = \frac{1}{n}\sum_{i=1}^{c} n_i \boldsymbol{\mu}_i$$

$$\boxed{S_W = \sum_{i=1}^{c} S_i = \sum_{i=1}^{c} \sum_{\boldsymbol{x}\in\omega_i} (\boldsymbol{x}-\boldsymbol{\mu}_i)(\boldsymbol{x}-\boldsymbol{\mu}_i)^T}$$

$$S_B = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T$$

$$S_T = \sum_{\boldsymbol{x}} (\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^T = \sum_{i=1}^{c} \sum_{x\in\omega_i} (\boldsymbol{x}-\boldsymbol{\mu}_i+\boldsymbol{\mu}_i-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu}_i+\boldsymbol{\mu}_i-\boldsymbol{\mu})^T$$

$$= \sum_{i=1}^{c} \sum_{x\in\omega_i} (\boldsymbol{x}-\boldsymbol{\mu}_i)(\boldsymbol{x}-\boldsymbol{\mu}_i)^T + \sum_{i=1}^{c} \sum_{x\in\omega_i} (\boldsymbol{\mu}_i-\boldsymbol{\mu})(\boldsymbol{\mu}_i-\boldsymbol{\mu})^T$$

$$= S_W + \boxed{\sum_{i=1}^{c} n_i (\boldsymbol{\mu}_i-\boldsymbol{\mu})(\boldsymbol{\mu}_i-\boldsymbol{\mu})^T = S_B} \qquad S_T = S_W + S_B$$

Linear Discriminant Analysis

- Multi-classes

$$J(\boldsymbol{a}) = \frac{\boldsymbol{a}^T S_B \boldsymbol{a}}{\boldsymbol{a}^T S_W \boldsymbol{a}}$$

$$S_W = \sum_{i=1}^{c} S_i = \sum_{i=1}^{c} \sum_{\boldsymbol{x} \in \omega_i} (\boldsymbol{x} - \boldsymbol{\mu}_i)(\boldsymbol{x} - \boldsymbol{\mu}_i)^T$$

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

$$S_B \boldsymbol{a} = \lambda S_W \boldsymbol{a}$$

Homework: Prove the rank of S_B is at most c-1

$$S_B \boldsymbol{a} = \lambda S_T \boldsymbol{a}$$

Linear Discriminant Analysis

- Main steps:
  - Form the scatter matrices S_B and S_W.
  - Compute the eigenvectors $\{\boldsymbol{a}_i\}_{i=1}^{c-1}$ corresponding to the non-zero eigenvalue of the generalized eigen-problem:

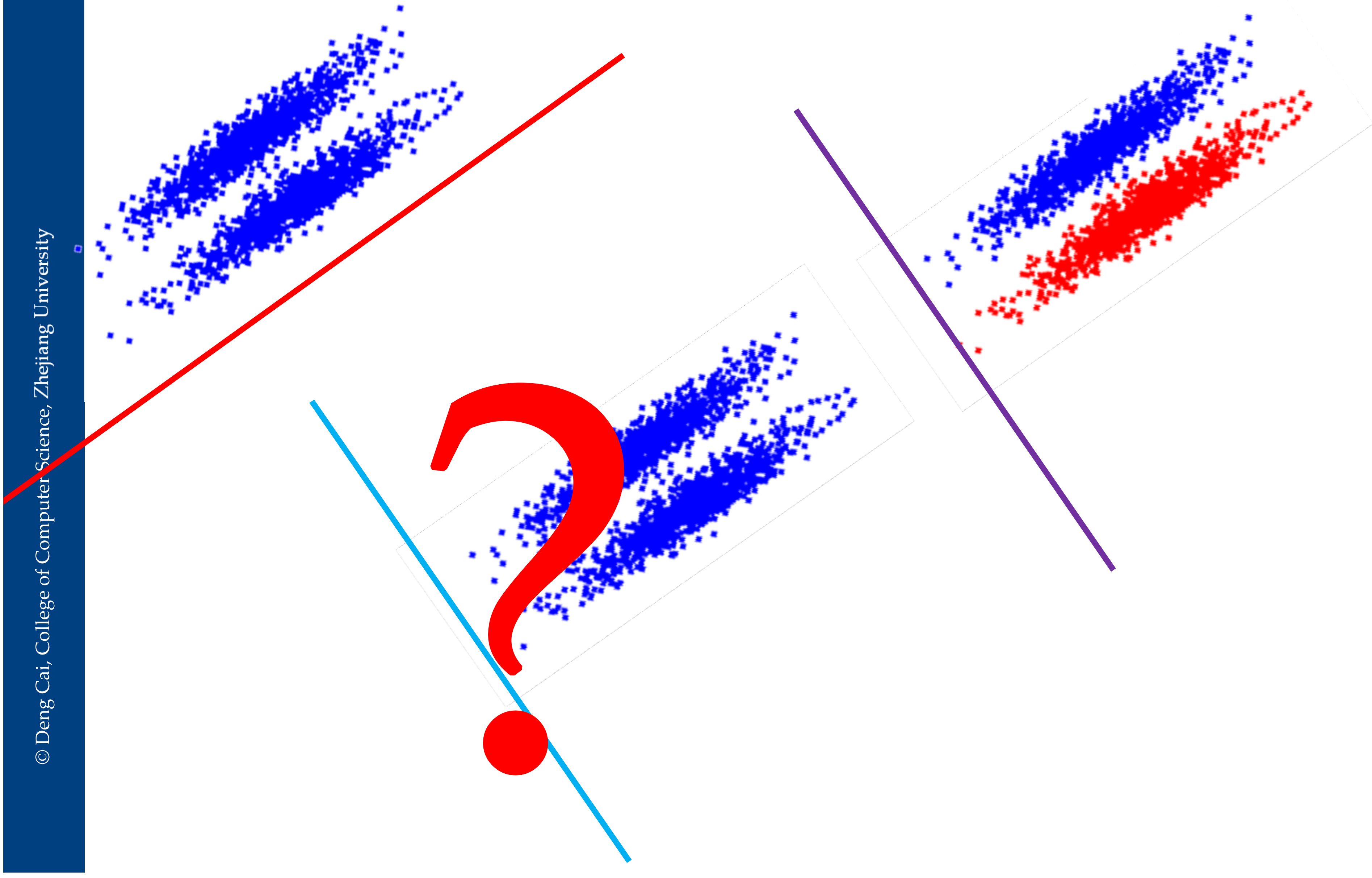
$$S_B\boldsymbol{a} = \lambda S_W\boldsymbol{a} \quad \text{or} \quad S_B\boldsymbol{a} = \lambda S_T\boldsymbol{a}$$

  - The transformation A is given by

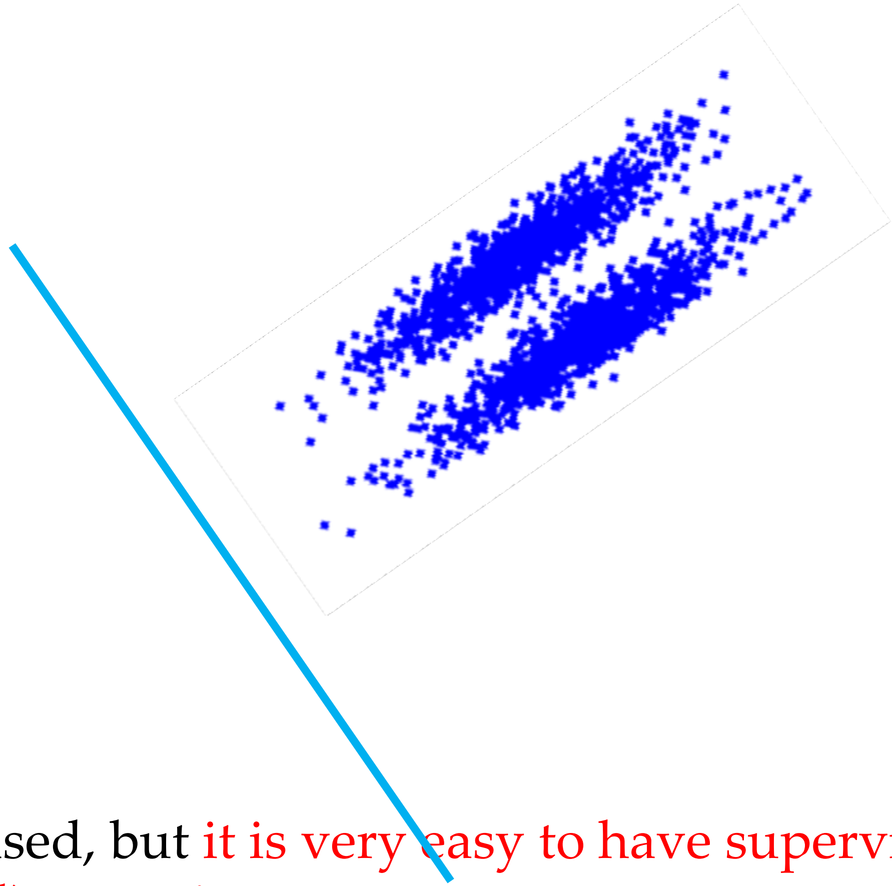
$$A = [\boldsymbol{a}_1, \cdots \boldsymbol{a}_{c-1}]$$

- A test point $\boldsymbol{x} \in \mathcal{R}^p \rightarrow A^T\boldsymbol{x} \in \mathcal{R}^{(c-1)}$

PCA vs LDA

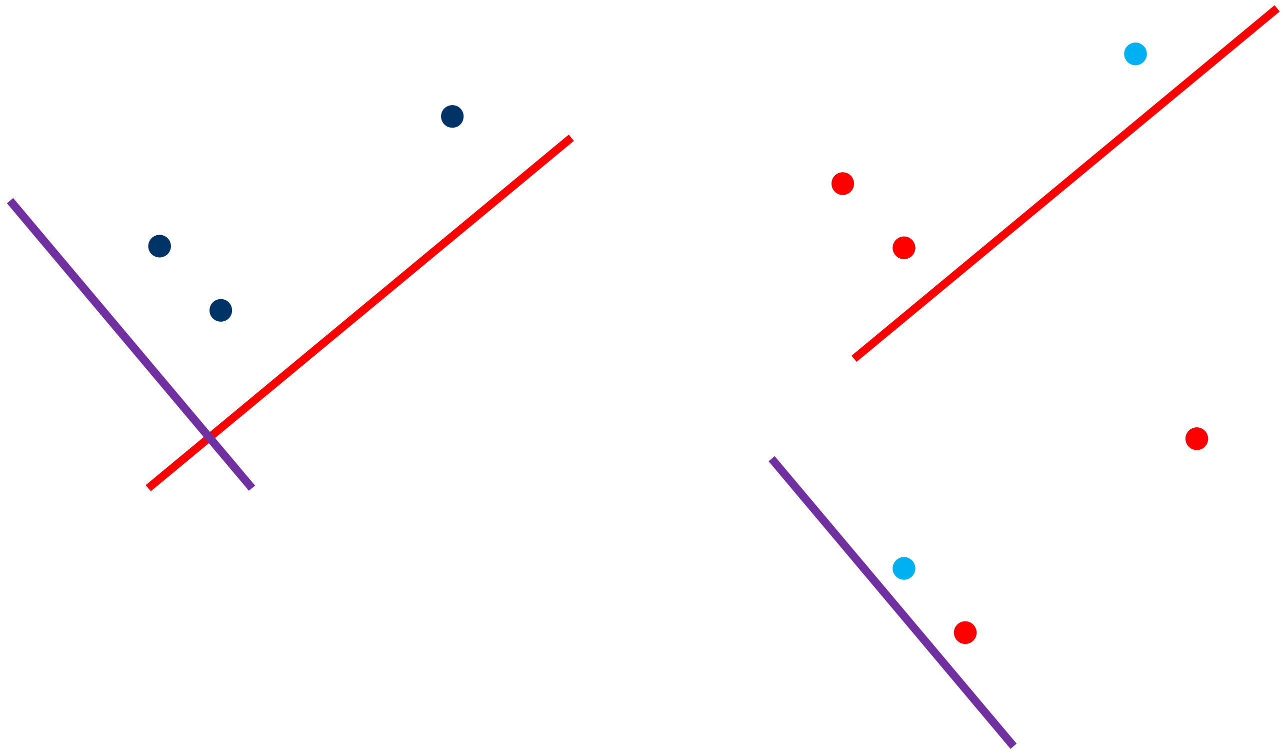


Locality Preserving Projections



- Unsupervised, but it is very easy to have supervised (semi-supervised) extensions.

Locality Preserving Projections (LPP)



- Basic idea: Locality Preserving

Locality Preserving Projections (LPP)

$$\boldsymbol{x} \in \mathcal{R}^p \ \rightarrow \boldsymbol{a}^T\boldsymbol{x} = z$$

$$W \in \mathcal{R}^{n\times n}, w_{ij} = \begin{cases} 1 & \text{if } x_i \text{ and } x_j \text{ are neighbors.} \\ 0 & \text{otherwise} \end{cases}$$

$$\min \sum_{ij} w_{ij}(z_i - z_j)^2 = \min \sum_{ij} w_{ij}(\boldsymbol{a}^T\boldsymbol{x}_i - \boldsymbol{a}^T\boldsymbol{x}_j)^2$$

$$= \min \sum_{ij} w_{ij}\boldsymbol{a}^T(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T\boldsymbol{a}$$

Graph Laplacian

$$L = D - W$$

$$= \min \boldsymbol{a}^T \sum_{ij} w_{ij}(\boldsymbol{x}_i\boldsymbol{x}_i^T - \boldsymbol{x}_i\boldsymbol{x}_j^T - \boldsymbol{x}_j\boldsymbol{x}_i^T + \boldsymbol{x}_j\boldsymbol{x}_j^T)\,\boldsymbol{a} = \min 2\boldsymbol{a}^T X(D - W)X^T\boldsymbol{a}$$

$$= \min \boldsymbol{a}^T XLX^T\boldsymbol{a}$$

$$\sum_{ij} w_{ij}(-\boldsymbol{x}_i\boldsymbol{x}_j^T - \boldsymbol{x}_j\boldsymbol{x}_i^T) = -2XWX^T \qquad \sum_{ij} w_{ij}\left(\boldsymbol{x}_i\boldsymbol{x}_i^T + \boldsymbol{x}_j\boldsymbol{x}_j^T\right) = 2XDX^T$$

$$X = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n]$$

$$D = \begin{bmatrix} d_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_{nn} \end{bmatrix}, d_{ii} = \sum_j w_{ij}$$

Locality Preserving Projections

$$\min \boldsymbol{a}^T X L X^T \boldsymbol{a} \qquad \Leftrightarrow \min \frac{\boldsymbol{a}^T X L X^T \boldsymbol{a}}{\boldsymbol{a}^T X D X^T \boldsymbol{a}}$$

$$s.t.\ \boldsymbol{a}^T X D X^T \boldsymbol{a} = 1$$

$$X L X^T \boldsymbol{a} = \lambda X D X^T \boldsymbol{a}$$

- Therefor, $\boldsymbol{a}$ is an eigenvector of the generalized eigen-problem corresponding to the smallest eigenvalue.

$$L = D - W$$

$$\min \boldsymbol{a}^T X L X^T \boldsymbol{a} \qquad \max \boldsymbol{a}^T X W X^T \boldsymbol{a}$$

$$s.t.\ \boldsymbol{a}^T X D X^T \boldsymbol{a} = 1 \qquad s.t.\ \boldsymbol{a}^T X D X^T \boldsymbol{a} = 1$$

$$X W X^T \boldsymbol{a} = \lambda X D X^T \boldsymbol{a}$$

- $\boldsymbol{a}$ is an eigenvector of the generalized eigen-problem corresponding to the largest eigenvalue.

LPP vs. LDA

$$\max \frac{\boldsymbol{a}^T X W X^T \boldsymbol{a}}{\boldsymbol{a}^T X D X^T \boldsymbol{a}}$$

$$\max \frac{\boldsymbol{a}^T S_B \boldsymbol{a}}{\boldsymbol{a}^T S_T \boldsymbol{a}} \quad \max \frac{\boldsymbol{a}^T S_B \boldsymbol{a}}{\boldsymbol{a}^T S_W \boldsymbol{a}}$$

$=$

$=$

$$W,\ w_{ij} = \begin{cases} \dfrac{1}{n_k} & \text{if } x_i \text{ and } x_j \text{ belong to } k\text{–th class.} \\ 0 & \text{otherwise} \end{cases}$$

$$W = \begin{bmatrix} W_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & W_c \end{bmatrix} \qquad W_k = \begin{bmatrix} \dfrac{1}{n_k} & \cdots & \dfrac{1}{n_k} \\ \vdots & \ddots & \vdots \\ \dfrac{1}{n_k} & \cdots & \dfrac{1}{n_k} \end{bmatrix}$$

$$D = I$$

$$S_T = \sum_{\boldsymbol{x}} (\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T = \sum_{\boldsymbol{x}} (\boldsymbol{x})(\boldsymbol{x})^T$$

$$= XDX^T$$

$$\boldsymbol{\mu} = \frac{1}{n} \sum_{\boldsymbol{x}} \boldsymbol{x} = \boldsymbol{0}$$

LPP vs. LDA

$$W = \begin{bmatrix} W_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & W_c \end{bmatrix} \quad W_k = \begin{bmatrix} \frac{1}{n_k} & \cdots & \frac{1}{n_k} \\ \vdots & \ddots & \vdots \\ \frac{1}{n_k} & \cdots & \frac{1}{n_k} \end{bmatrix}$$

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

$$= \sum_{i}^{c} \frac{1}{n_i} \sum_{\boldsymbol{x}\in\omega_i} \boldsymbol{x} \sum_{\boldsymbol{x}\in\omega_i} \boldsymbol{x} = \sum_{i}^{c} X_k W_k X_k^T = XWX^T$$

$$S_B = XWX^T \qquad S_T = XDX^T$$

- LPP is equivalent to LDA when a specifically designed supervised graph is used

LPP vs. PCA

$$\max \frac{\boldsymbol{a}^T X W X^T \boldsymbol{a}}{\boldsymbol{a}^T X D X^T \boldsymbol{a}} \qquad X W X^T \boldsymbol{a} = \lambda X D X^T \boldsymbol{a}$$

$$\max \frac{\boldsymbol{a}^T X W X^T \boldsymbol{a}}{\boldsymbol{a}^T X X^T \boldsymbol{a}} \qquad X W X^T \boldsymbol{a} = \lambda X X^T \boldsymbol{a}$$

$$W = X^T X$$

$$X X^T X X^T \boldsymbol{a} = \lambda X X^T \boldsymbol{a}$$

$$X X^T \boldsymbol{a} = \lambda \boldsymbol{a}$$

- LPP is similar to PCA when an inner product graph is used.

Graph based Dimensionality Reduction

- Given examples $\boldsymbol{x}_i \in \mathcal{R}^p$, $i = 1, \cdots, n$
- Construct a graph with weight matrix $W \in \mathcal{R}^{n \times n}$
- Solve the optimization problem:

$$\max \frac{\boldsymbol{a}^T X W X^T \boldsymbol{a}}{\boldsymbol{a}^T X D X^T \boldsymbol{a}}$$

$$X W X^T \boldsymbol{a} = \lambda X D X^T \boldsymbol{a}$$

- The transformation A is given by

$$A = [\boldsymbol{a}_1, \cdots \boldsymbol{a}_d]$$

Graph based Dimensionality Reduction Method

- Locality Preserving Projection [He & Niyogi, 2003]
- Linear Discriminant Analysis [Fisher, 1936]
- Neighborhood Preserving Embedding [He et. al, 2005]
- Marginal Fisher Analysis [Yan et. al, 2005]
- Local discriminant embedding [Chen et. al, 2005]
- Augmented Relation Embedding [Lin et. al, 2005]
- Isometric Projection [Cai et. al, 2006]
- Semantic Subspace Projection [Yu & Tian, 2006]
- Locally Sensitive Discriminant Analysis [Cai et. al, 2007]
- Semi-supervised Discriminant Analysis [Cai et. al, 2007]
- ……

Regularization on Graph based Dimensionality Reduction Methods

- PCA: $XX^T\boldsymbol{a} = \lambda\boldsymbol{a}$

$$\max \frac{\boldsymbol{a}^T X X^T \boldsymbol{a}}{\boldsymbol{a}^T \boldsymbol{a}}$$

- LDA: $S_B\boldsymbol{a} = \lambda S_W\boldsymbol{a}$ or $S_B\boldsymbol{a} = \lambda S_T\boldsymbol{a}$
  - $S_B\boldsymbol{a} = \lambda(S_W + \gamma I)\boldsymbol{a}$
  - $S_B\boldsymbol{a} = \lambda(S_T + \gamma I)\boldsymbol{a}$

$$\max \frac{\boldsymbol{a}^T S_B \boldsymbol{a}}{\boldsymbol{a}^T (S_W + \gamma I)\boldsymbol{a}} \qquad \max \frac{\boldsymbol{a}^T S_B \boldsymbol{a}}{\boldsymbol{a}^T (S_T + \gamma I)\boldsymbol{a}}$$

- LPP: $XLX^T\boldsymbol{a} = \lambda XDX^T\boldsymbol{a}$ or $XWX^T\boldsymbol{a} = \lambda XDX^T\boldsymbol{a}$
  - $XWX^T\boldsymbol{a} = \lambda\left(XDX^T + \gamma I\right)\boldsymbol{a}$

$$\min \cancel{\frac{\boldsymbol{a}^T X L X^T \boldsymbol{a}}{\boldsymbol{a}^T (X D X^T + \gamma I)\boldsymbol{a}}} \qquad \max \frac{\boldsymbol{a}^T X W X^T \boldsymbol{a}}{\boldsymbol{a}^T (X D X^T + \gamma I)\boldsymbol{a}}$$

- Regularization from Ridge idea: minimizing $\boldsymbol{a}^T\boldsymbol{a}$

Linear vs. Nonlinear Methods

$\boldsymbol{x} \in \mathcal{R}^p$ $\boldsymbol{z} \in \mathcal{R}^k$ z

$f(z)$

Linear: $\boldsymbol{a}^T\boldsymbol{x} = z$

Nonlinear: $g(\boldsymbol{x}) = z$

Laplacian Eigenmap

- Nonlinear dimensionality reduction
- LPP is a linear version of Laplacian Eigenmap
- LPP objective function:

$$\min \boldsymbol{a}^T X L X^T \boldsymbol{a}$$
$$s.t.\ \boldsymbol{a}^T X D X^T \boldsymbol{a} = 1$$

$$\min \boldsymbol{y}^T L \boldsymbol{y}$$
$$s.t.\ \boldsymbol{y}^T D \boldsymbol{y} = 1$$

- Exactly same as the Normalized Cut.