



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 x
- ▶ Better representation makes learning easier
- ▶ **Minimum requirement:**
 - 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 \mathbf{x} to \mathbf{z}
 - The \mathbf{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 \mathbf{z} is **usually** smaller than that of \mathbf{x}
 - This is the name DR comes from.



Linear and Nonlinear

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

Diagram showing the mapping from $\mathcal{F}(\mathbf{x} \in R^p) = \mathbf{z} \in R^d$ to its components:

$$f_1(\mathbf{x}) = z_1 \quad f_i(\mathbf{x}) = z_i \quad f_d(\mathbf{x}) = z_d$$

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

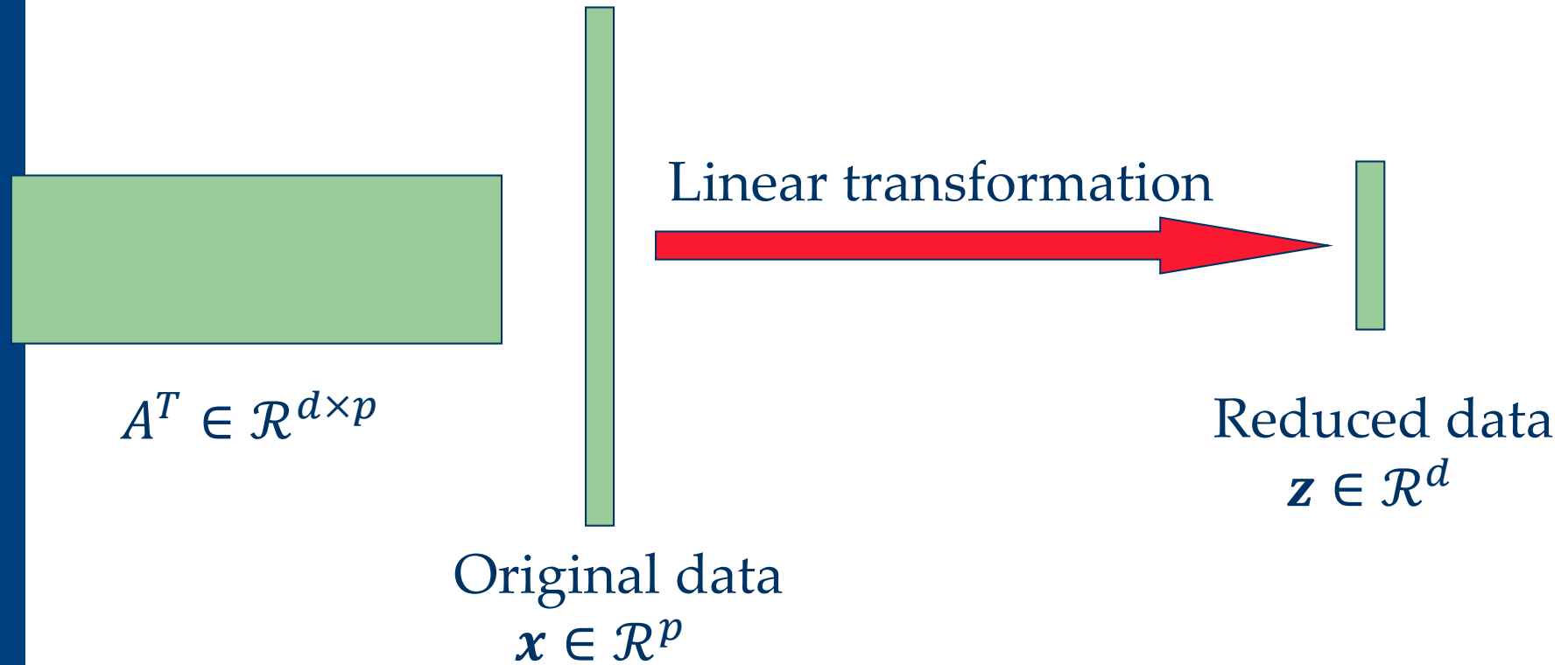
$$\mathbf{a}_1^T \mathbf{x} = z_1 \quad \mathbf{a}_i^T \mathbf{x} = z_i$$

$$A \triangleq [\mathbf{a}_1, \dots, \mathbf{a}_d] \quad A^T \mathbf{x} = \mathbf{z}$$

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



Linear Transformation



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



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

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

- ▶ 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

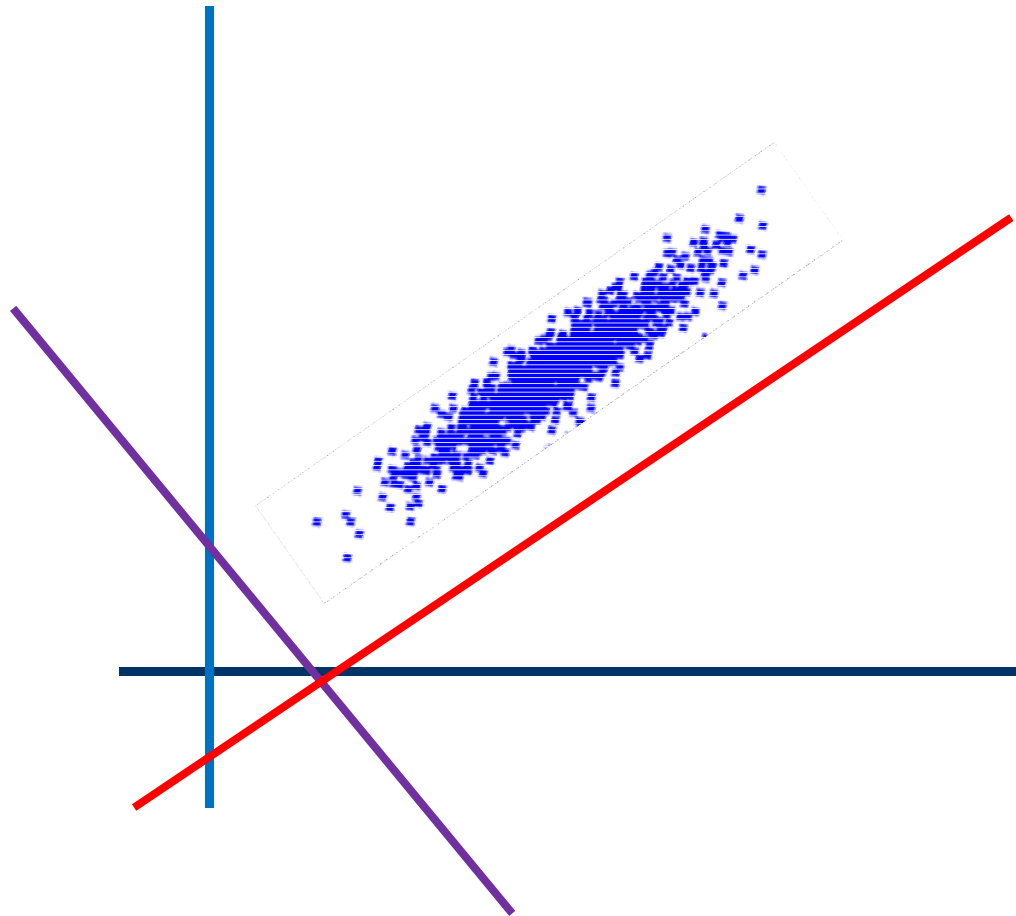


Algorithms

- ▶ 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

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

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

$$z_i^{(1)} = \mathbf{a}_1^T \mathbf{x}_i, \quad i = 1, \dots, n$$

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



Algebraic Derivation of PCs

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

$$= \frac{1}{n} \sum_{i=1}^n (\mathbf{a}_1^T \mathbf{x}_i - \mathbf{a}_1^T \bar{\mathbf{x}})^2$$

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

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

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



Algebraic Derivation of PCs

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

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

Let λ be a Lagrange multiplier

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

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

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

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



Algebraic Derivation of PCs

$$\begin{aligned} & \max_{\mathbf{a}_2} \mathbf{a}_2^T S \mathbf{a}_2 \\ \text{s.t. } & \mathbf{a}_2^T \mathbf{a}_2 = 1, \text{cov}(\mathbf{z}^{(2)}, \mathbf{z}^{(1)}) = 0 \end{aligned}$$

$$\text{cov}(\mathbf{z}^{(2)}, \mathbf{z}^{(1)}) = \mathbf{a}_2^T S \mathbf{a}_1 = \lambda \mathbf{a}_2^T \mathbf{a}_1 = 0$$

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

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



Algebraic Derivation of PCs

- ▶ In general:

$$\text{var}(z^{(k)}) = \mathbf{a}_k^T S \mathbf{a}_k = \lambda_k$$

- ▶ 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

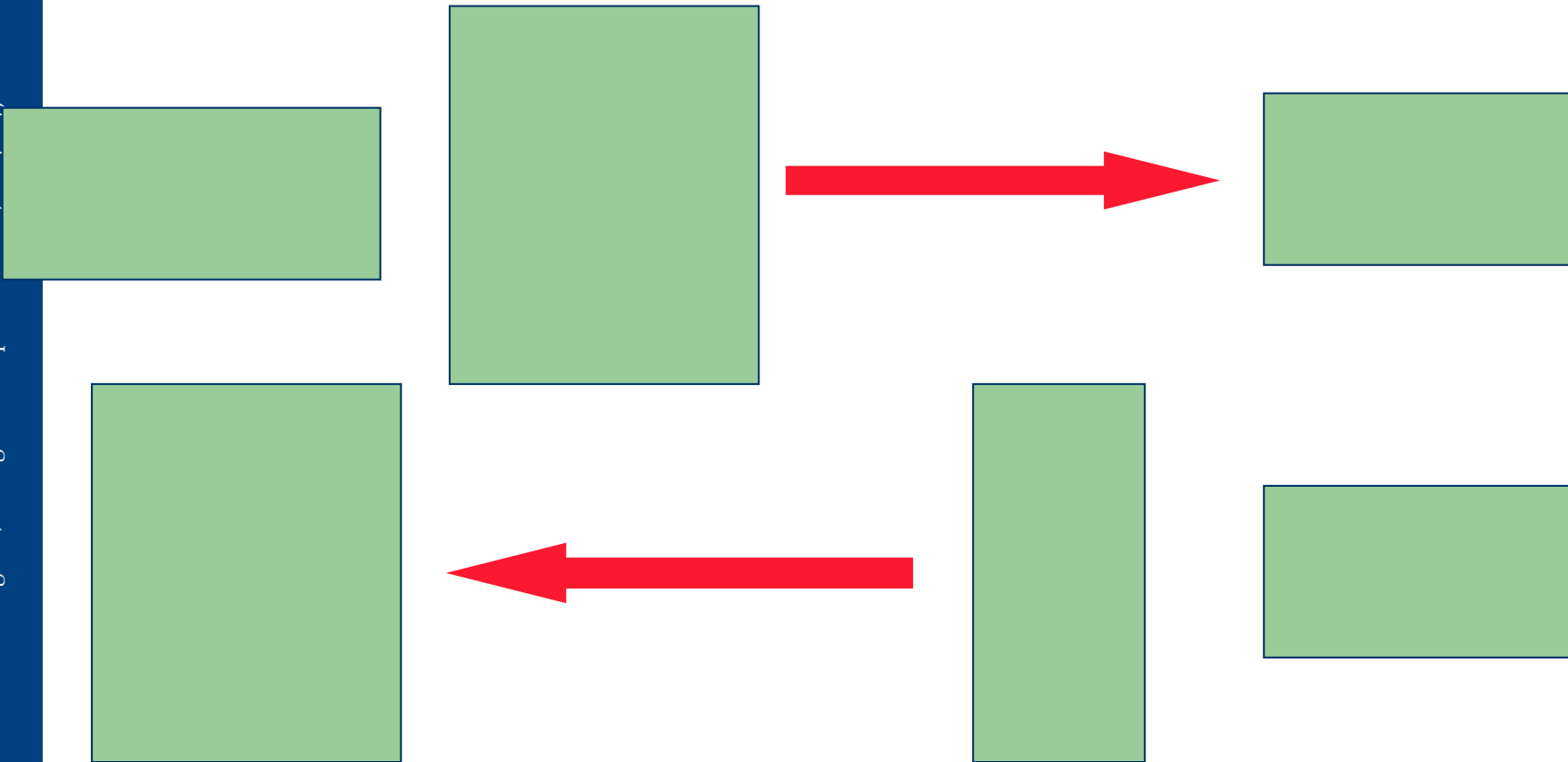
- ▶ Main steps for computing PCs:
 - Form the covariance matrix S .
 - Compute its eigenvectors: $\{\mathbf{a}_i\}_{i=1}^p$
 - Use the first d eigenvectors $\{\mathbf{a}_i\}_{i=1}^d$ to form the d PCs.
 - The transformation A is given by
$$A = [\mathbf{a}_1, \cdots \mathbf{a}_d]$$
- ▶ A test point $\mathbf{x} \in \mathcal{R}^p \rightarrow A^T \mathbf{x} \in \mathcal{R}^d$



Optimality Property of PCA

Reconstruction

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





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:

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

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

Reconstruction error

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



PCA for Image Compression



$d=1$



$d=2$



$d=4$



$d=8$



$d=16$



$d=32$



$d=64$



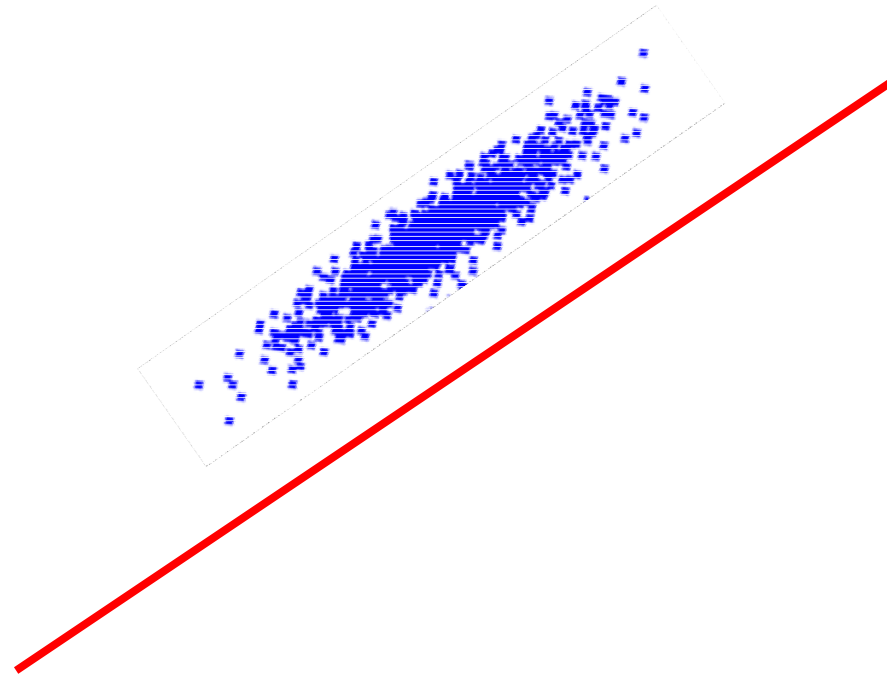
$d=100$

Original
Image



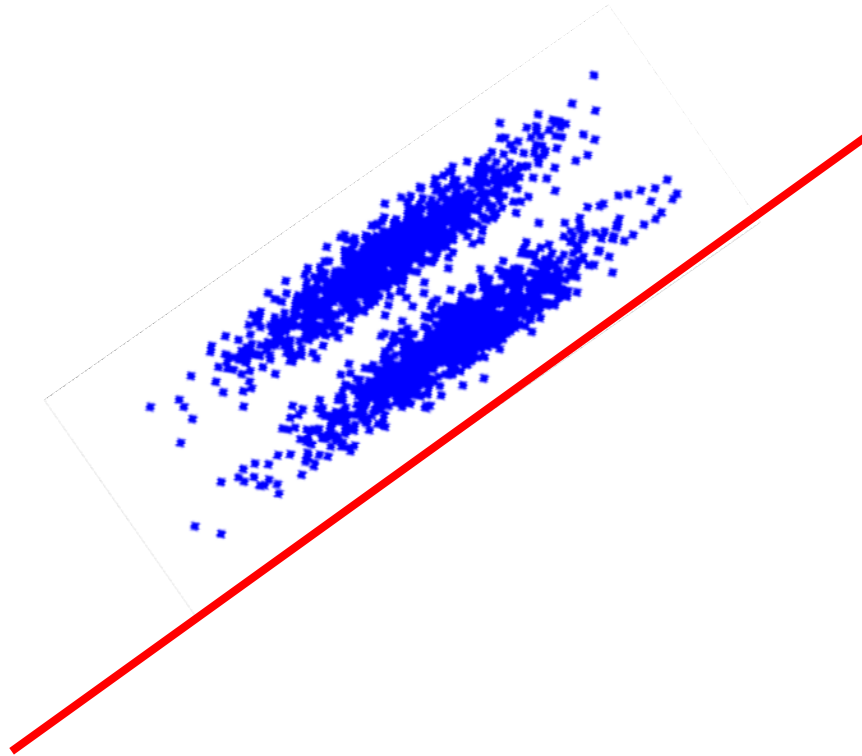


Principal Component Analysis





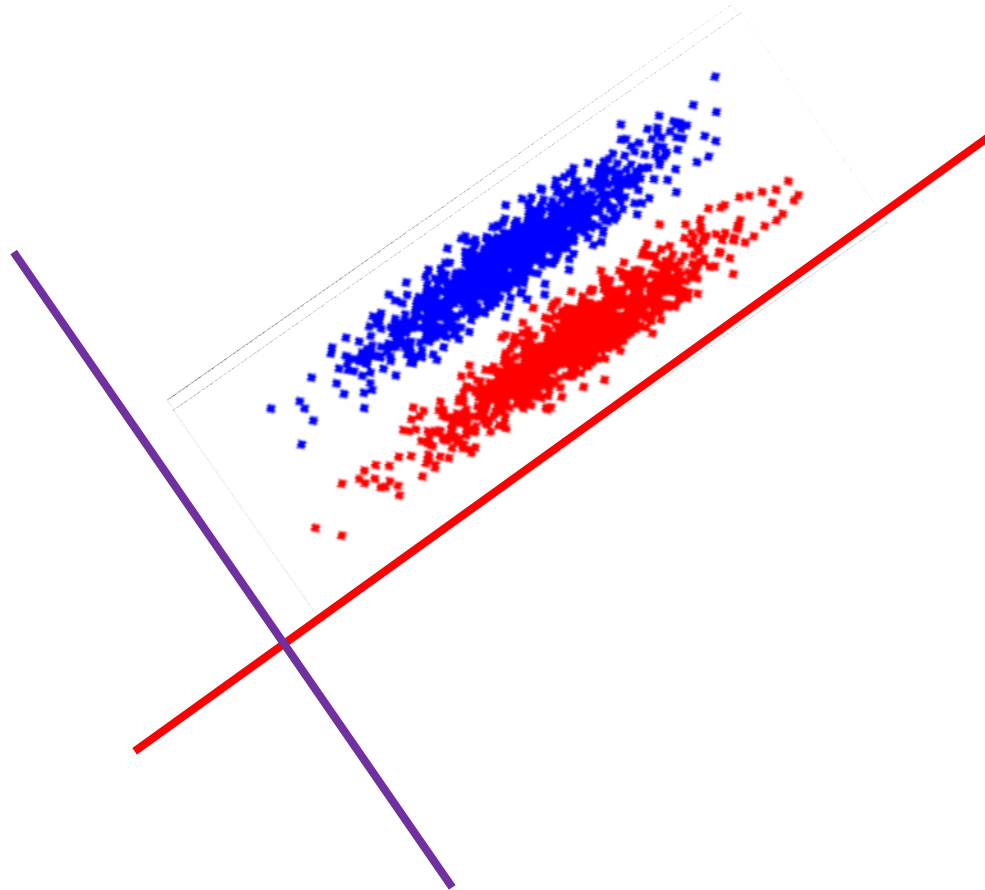
Principal Component Analysis



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

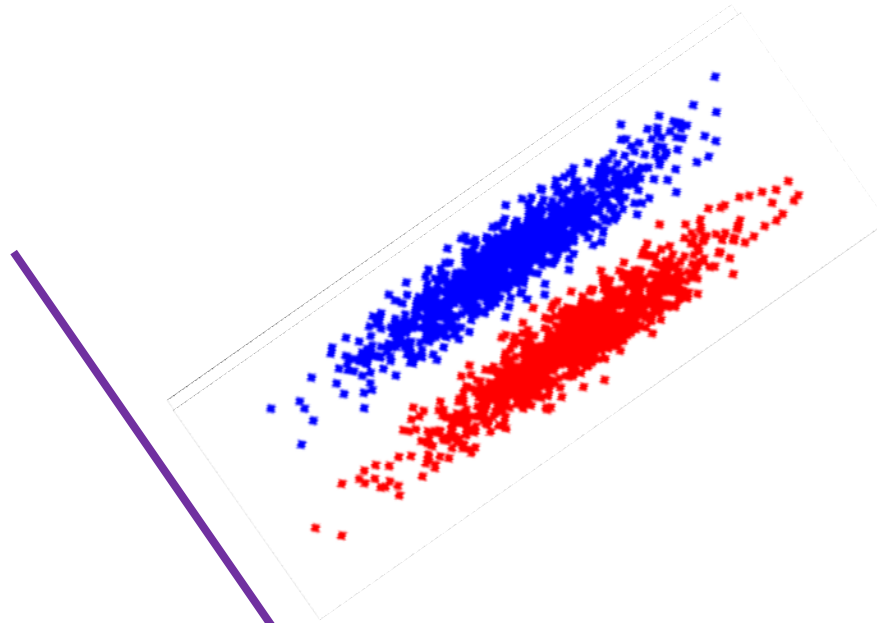


Principal Component Analysis





Linear Discriminant Analysis (Fisher Linear Discriminant)



- Find a transformation \mathbf{a} , such that the $\mathbf{a}^T X_1$ and $\mathbf{a}^T X_2$ 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 ω_1, ω_2

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

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

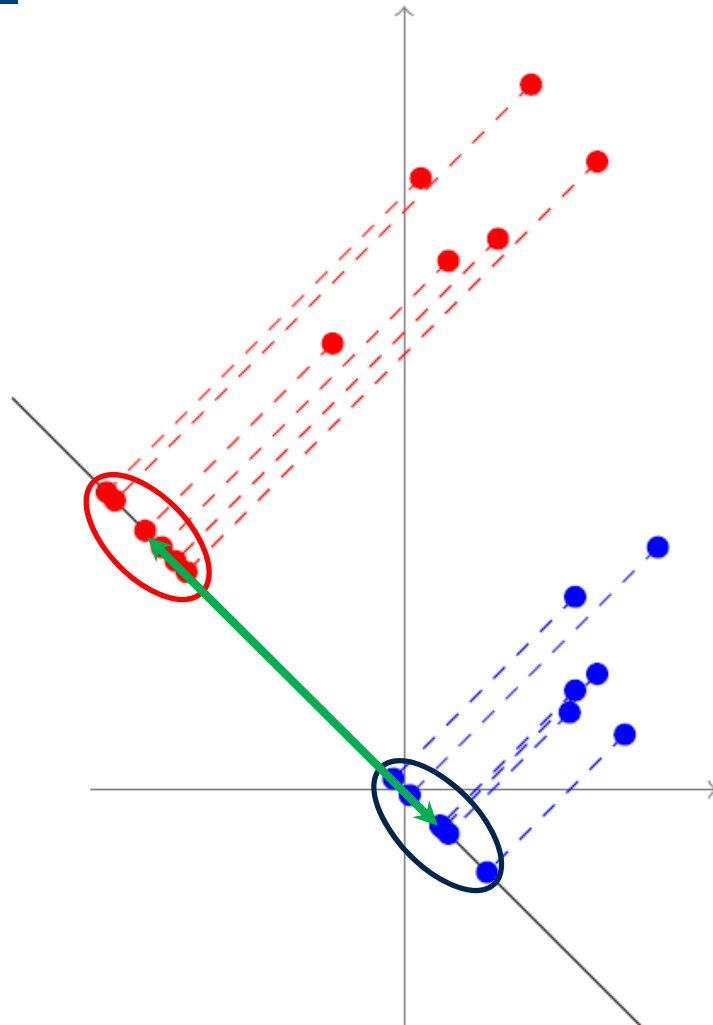
$$\mu_i = \frac{1}{n_i} \sum_{\mathbf{x} \in \omega_i} \mathbf{x}, \tilde{\mu}_i = \mathbf{a}^T \mu_i$$

$$|\tilde{\mu}_1 - \tilde{\mu}_2| = |\mathbf{a}^T (\mu_1 - \mu_2)|$$

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

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

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





Linear Discriminant Analysis

► Two Classes

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

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

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

S_i : scatter matrix

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

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

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



Linear Discriminant Analysis

► Two Classes

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

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

? Homework

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

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

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

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



Linear Discriminant Analysis

► Multi-classes

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

$$\mu = \frac{1}{n} \sum_x \mathbf{x} = \frac{1}{n} \sum_{i=1}^c n_i \mu_i$$

$$S_W \equiv \sum_{i=1}^c S_i \equiv \sum_{i=1}^c \sum_{\mathbf{x} \in \omega_i} (\mathbf{x} - \mu_i)(\mathbf{x} - \mu_i)^T$$

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

$$\begin{aligned} S_T &= \sum_x (\mathbf{x} - \mu)(\mathbf{x} - \mu)^T = \sum_{i=1}^c \sum_{\mathbf{x} \in \omega_i} (\mathbf{x} - \mu_i + \mu_i - \mu)(\mathbf{x} - \mu_i + \mu_i - \mu)^T \\ &= \sum_{i=1}^c \sum_{\mathbf{x} \in \omega_i} (\mathbf{x} - \mu_i)(\mathbf{x} - \mu_i)^T + \sum_{i=1}^c \sum_{\mathbf{x} \in \omega_i} (\mu_i - \mu)(\mu_i - \mu)^T \end{aligned}$$

$$= S_W + \sum_{i=1}^c n_i (\mu_i - \mu)(\mu_i - \mu)^T = S_B$$

$$S_T = S_W + S_B$$



Linear Discriminant Analysis

- Multi-classes

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

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

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

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

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

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



Linear Discriminant Analysis

► Main steps:

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

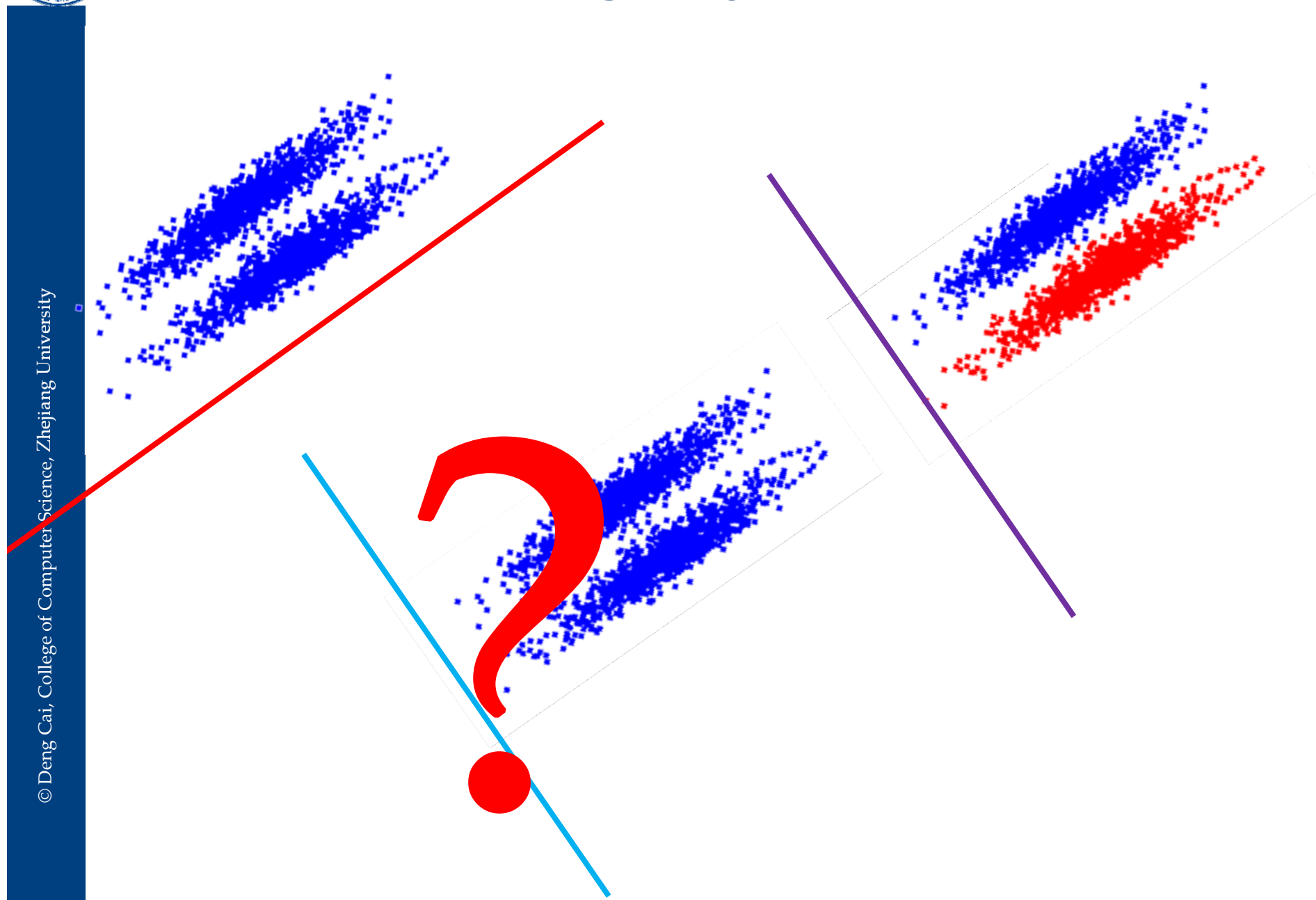
$$S_B \mathbf{a} = \lambda S_W \mathbf{a} \quad \text{or} \quad S_B \mathbf{a} = \lambda S_T \mathbf{a}$$

- The transformation A is given by
$$A = [\mathbf{a}_1, \dots, \mathbf{a}_{c-1}]$$

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

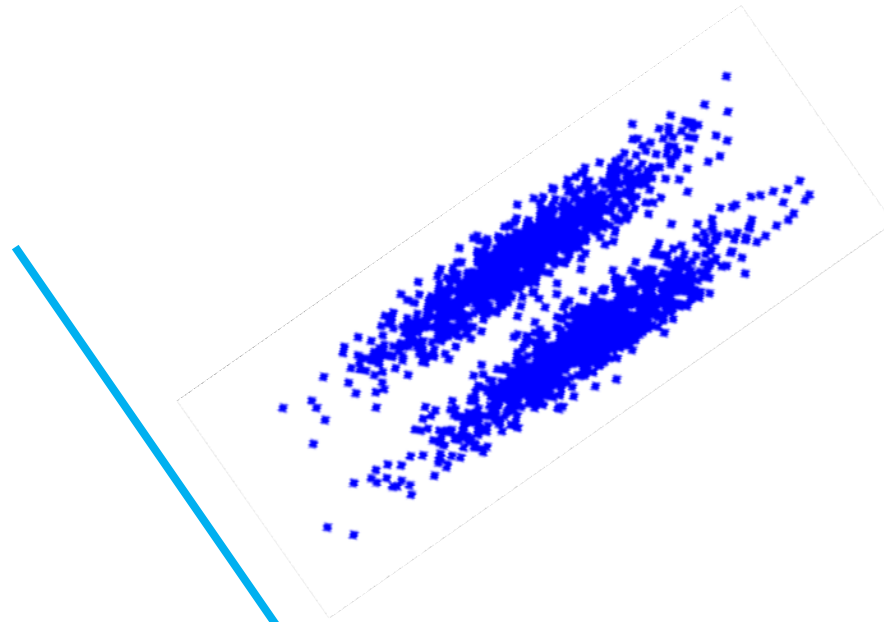


PCA vs LDA





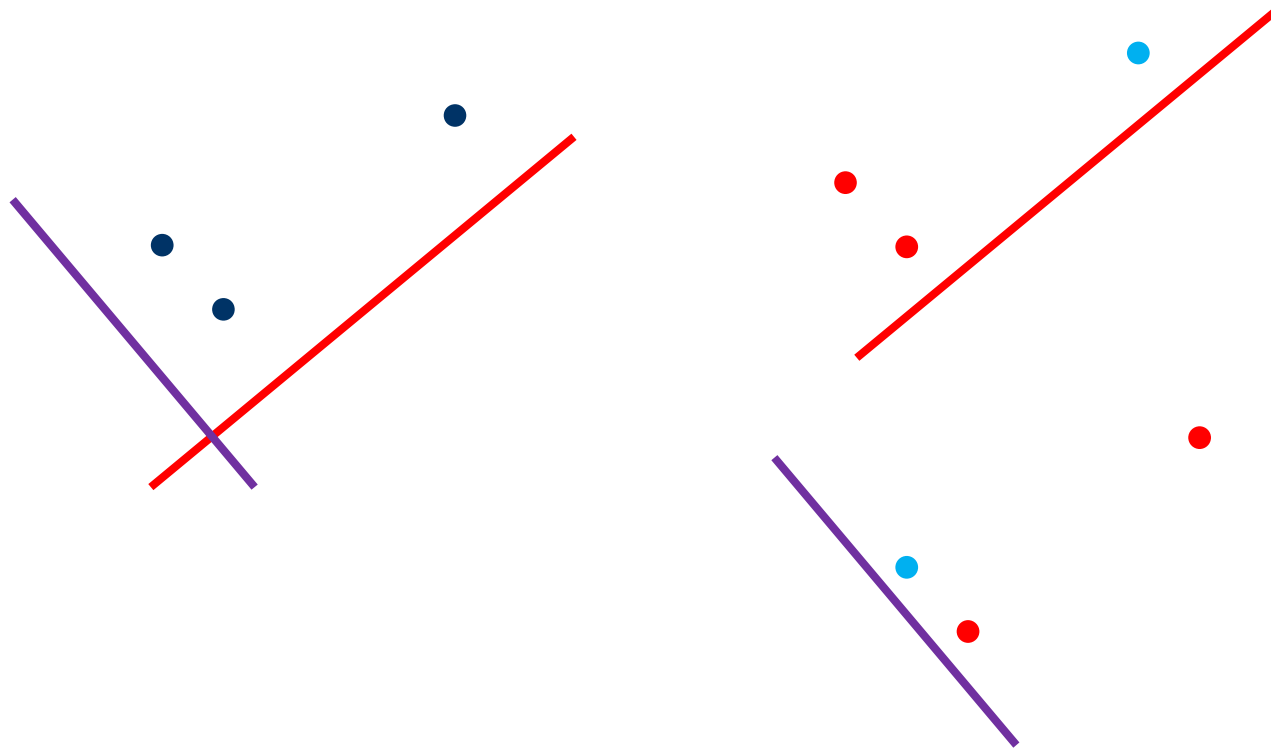
Locality Preserving Projections



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



Locality Preserving Projections (LPP)



- ▶ Basic idea: Locality Preserving



Locality Preserving Projections (LPP)

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

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

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

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

Graph Laplacian

$$L = D - W$$

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

$$= \min \mathbf{a}^T X L X^T \mathbf{a}$$

$$\sum_{ij} w_{ij} (-\mathbf{x}_i \mathbf{x}_j^T - \mathbf{x}_j \mathbf{x}_i^T) = -2 X W X^T \quad \sum_{ij} w_{ij} (\mathbf{x}_i \mathbf{x}_i^T + \mathbf{x}_j \mathbf{x}_j^T) = 2 X D X^T$$

$$X = [\mathbf{x}_1, \dots, \mathbf{x}_n]$$

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



Locality Preserving Projections

$$\begin{aligned} \min \mathbf{a}^T X L X^T \mathbf{a} \\ \text{s.t. } \mathbf{a}^T X D X^T \mathbf{a} = 1 \end{aligned} \quad \Leftrightarrow \quad \min \frac{\mathbf{a}^T X L X^T \mathbf{a}}{\mathbf{a}^T X D X^T \mathbf{a}}$$

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

- Therefore, \mathbf{a} is an eigenvector of the generalized eigen-problem corresponding to the **smallest** eigenvalue.

$$L = D - W$$

$$\begin{aligned} \min \mathbf{a}^T X L X^T \mathbf{a} & \quad \max \mathbf{a}^T X W X^T \mathbf{a} \\ \text{s.t. } \mathbf{a}^T X D X^T \mathbf{a} = 1 & \quad \text{s.t. } \mathbf{a}^T X D X^T \mathbf{a} = 1 \end{aligned}$$

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

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



LPP

vs.

LDA

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

$$\max \frac{\mathbf{a}^T \mathbf{S}_B \mathbf{a}}{\mathbf{a}^T \mathbf{S}_T \mathbf{a}} \quad \max \frac{\mathbf{a}^T \mathbf{S}_B \mathbf{a}}{\mathbf{a}^T \mathbf{S}_W \mathbf{a}}$$

$$=$$

$$=$$

$$W_{ij} = \begin{cases} \frac{1}{n_k} & \text{if } x_i \text{ and } x_j \\ & \text{belong to } k\text{-th class.} \\ 0 & \text{otherwise} \end{cases}$$

$$W = \begin{bmatrix} W_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & W_c \end{bmatrix}$$

$$W_k = \begin{bmatrix} \frac{1}{n_k} & \cdots & \frac{1}{n_k} \\ \vdots & \ddots & \vdots \\ \frac{1}{n_k} & \cdots & \frac{1}{n_k} \end{bmatrix}$$

$$D = I$$

$$\mu = \frac{1}{n} \sum_x \mathbf{x} = \mathbf{0}$$

$$\begin{aligned} S_T &= \sum_x (\mathbf{x} - \mu)(\mathbf{x} - \mu)^T = \sum_x (\mathbf{x})(\mathbf{x})^T \\ &= \mathbf{X} \mathbf{D} \mathbf{X}^T \end{aligned}$$



LPP vs. LDA

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

$$\begin{aligned} S_B &= \sum_{i=1}^c n_i (\mu_i - \mu)(\mu_i - \mu)^T = \sum_{i=1}^c n_i (\mu_i)(\mu_i)^T \\ &= \sum_i \frac{1}{n_i} \sum_{x \in \omega_i} x \sum_{x \in \omega_i} x = \sum_i X_k W_k X_k^T = X W X^T \end{aligned}$$

$$S_B = X W X^T \quad S_T = X D X^T$$

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



LPP vs. PCA

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

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

$$\max \frac{\mathbf{a}^T X W X^T \mathbf{a}}{\mathbf{a}^T X X^T \mathbf{a}}$$

$$X W X^T \mathbf{a} = \lambda X X^T \mathbf{a}$$

$$W = X^T X$$

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

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

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



Graph based Dimensionality Reduction

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

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

- ▶ The transformation A is given by

$$A = [\mathbf{a}_1, \dots, \mathbf{a}_d]$$



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: $XX^T \mathbf{a} = \lambda \mathbf{a}$ $\max \frac{\mathbf{a}^T XX^T \mathbf{a}}{\mathbf{a}^T \mathbf{a}}$
- LDA: $S_B \mathbf{a} = \lambda S_W \mathbf{a}$ or $S_B \mathbf{a} = \lambda S_T \mathbf{a}$
 - $S_B \mathbf{a} = \lambda (S_W + \gamma I) \mathbf{a}$
 - $S_B \mathbf{a} = \lambda (S_T + \gamma I) \mathbf{a}$ $\max \frac{\mathbf{a}^T S_B \mathbf{a}}{\mathbf{a}^T (S_W + \gamma I) \mathbf{a}} \quad \max \frac{\mathbf{a}^T S_B \mathbf{a}}{\mathbf{a}^T (S_T + \gamma I) \mathbf{a}}$
- ▶ LPP: $XLX^T \mathbf{a} = \lambda XDX^T \mathbf{a}$ or $XWX^T \mathbf{a} = \lambda XDX^T \mathbf{a}$
 - $XWX^T \mathbf{a} = \lambda (XDX^T + \gamma I) \mathbf{a}$ $\min \frac{\mathbf{a}^T XLX^T \mathbf{a}}{\mathbf{a}^T (XDX^T + \gamma I) \mathbf{a}} \quad \max \frac{\mathbf{a}^T XWX^T \mathbf{a}}{\mathbf{a}^T (XDX^T + \gamma I) \mathbf{a}}$
- ▶ Regularization from Ridge idea: minimizing $\mathbf{a}^T \mathbf{a}$



Linear vs. Nonlinear Methods

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

$$f(\mathbf{z})$$

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

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



Laplacian Eigenmap

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

$$\begin{aligned} \min \mathbf{a}^T X L X^T \mathbf{a} \\ \text{s.t. } \mathbf{a}^T X D X^T \mathbf{a} = 1 \end{aligned}$$

$$\begin{aligned} \min \mathbf{y}^T L \mathbf{y} \\ \text{s.t. } \mathbf{y}^T D \mathbf{y} = 1 \end{aligned}$$

- ▶ Exactly same as the Normalized Cut.