Pattern Recognition

(EE5907R)

Jiashi FENG

Email: elefjia@nus.edu.sg

Outlines

- Unsupervised Feature Extraction (PCA, NMF,…)
- **Supervised Feature Extraction (LDA, GE, …)**
- Clustering and Applications
- Gaussian Mixture Model
- Feature Coding
- Deep Learning

Area with active research progress in the past decade

Some slides from Shuicheng Yan, Ricardo Gutierrez-Osuna, Dong Xu

Before Start

- What we studied in the last lecture
- How to visualize PCA eigenvectors
- How to select the # of PCs
- More explanation on the reconstruction formulation

$$\overset{\frown}{x}_i = \overline{x} + U_{d,p}(U_{d,p})^T(x_i - \overline{x})$$

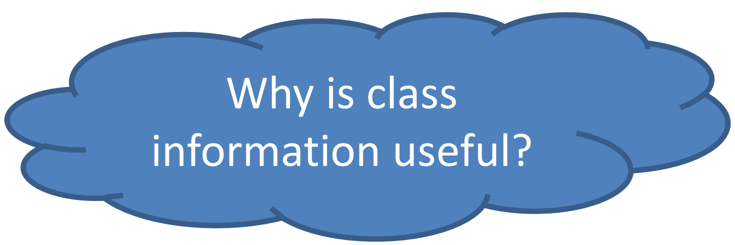
- Matlab license issue

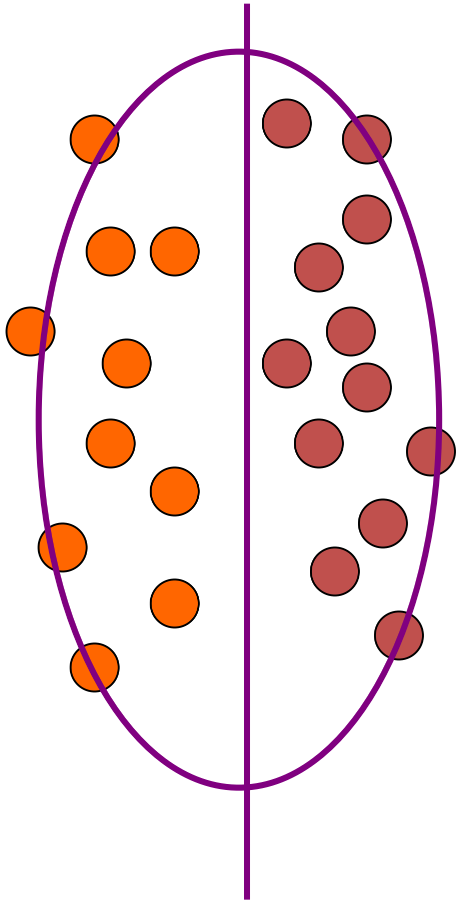
Supervised Feature Extraction:
Linear Discriminant Analysis

Why is class in Pattern Recognition important?

Is PCA a Good Criterion for Classification?

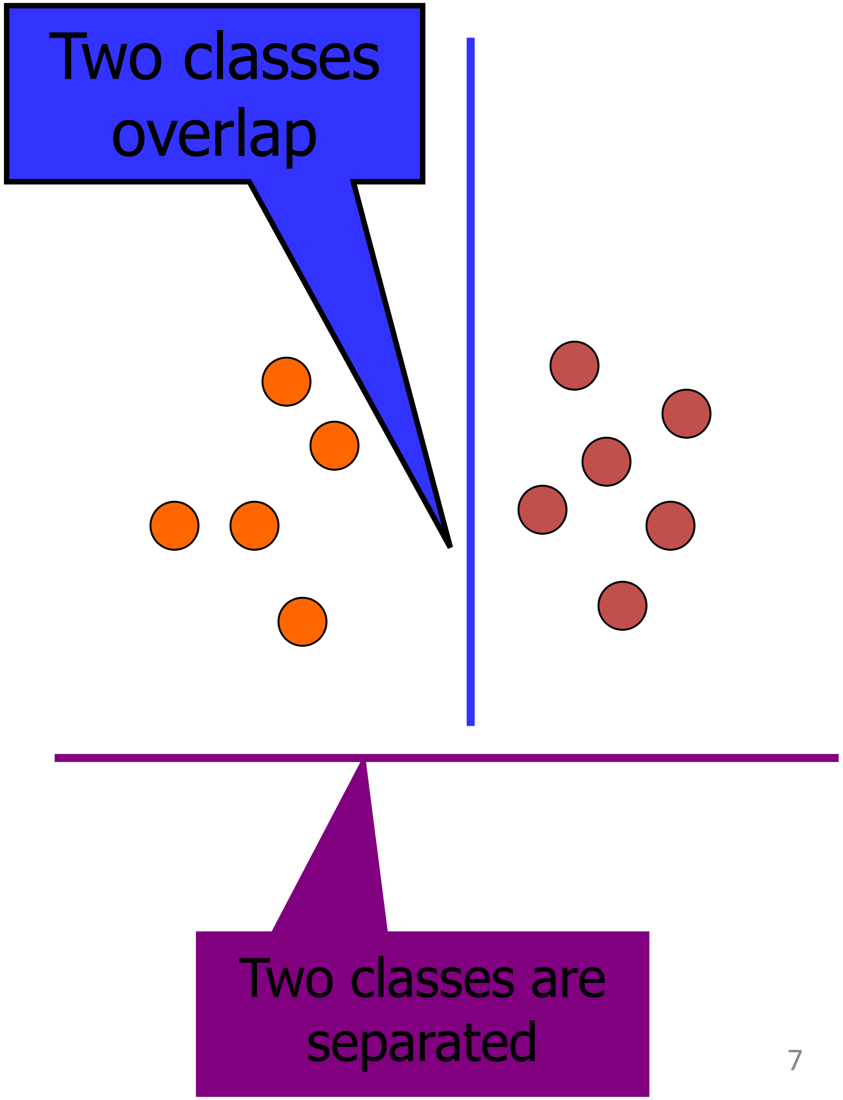
- Data variation determines the projection direction
- What’s missing?
  - Class information





What is a Good Projection?

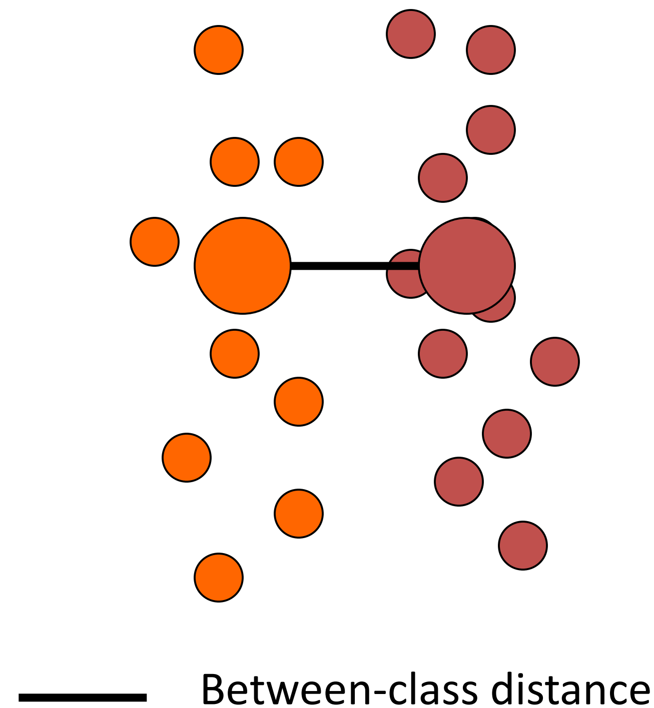
- What is a good criterion?
  - Separating different classes



What Class Information May be Useful?

- Between-class distance
  - Distance between the centroids of different classes

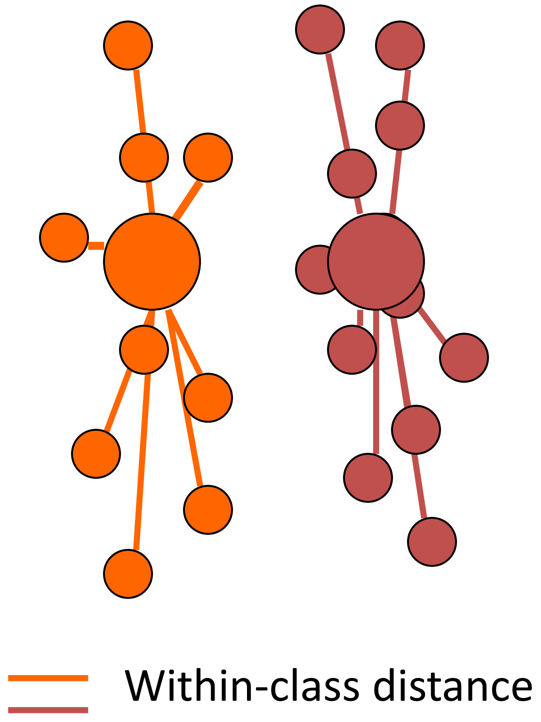
Should be large or small?



What Class Information May be Useful?

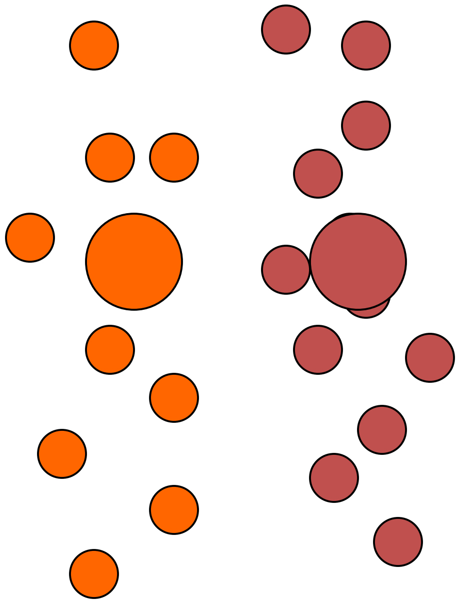
- Between-class distance
  - Distance between the centroids of different classes
- Within-class distance
  - Accumulated distance of an instance to the centroid of its class

Should be large or small?



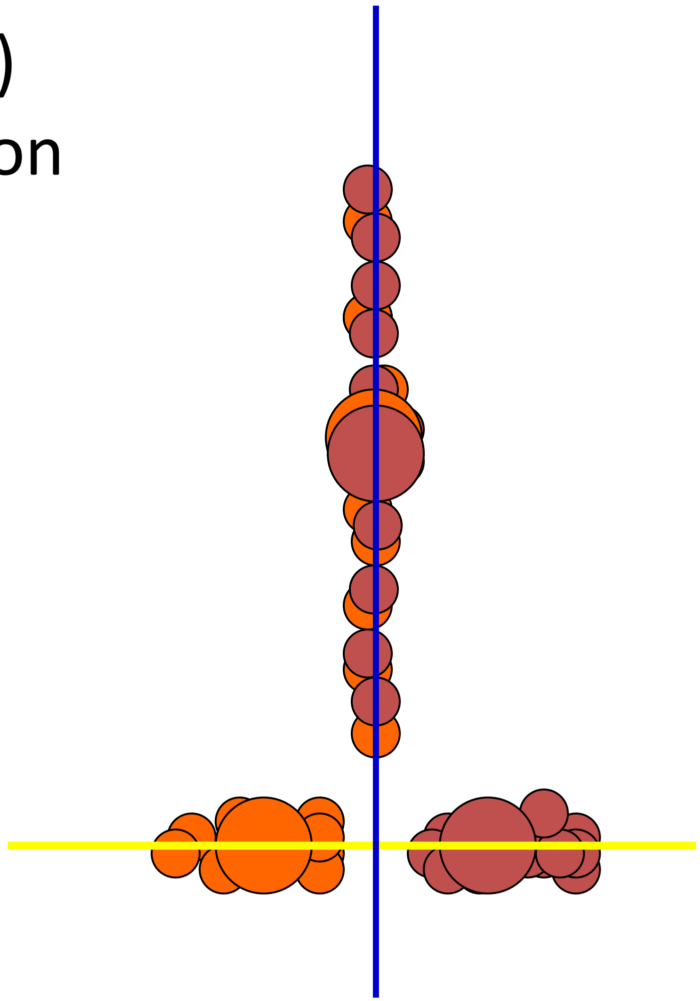
Linear Discriminant Analysis

- **Linear discriminant analysis** (LDA) finds most discriminative projection by maximizing between-class distance and minimizing within-class distance



Linear Discriminant Analysis

- **Linear discriminant analysis** (LDA) finds most discriminative projection by maximizing between-class distance and minimizing within-class distance



Statistical Facts (1)

- Class-specific mean vector (sample):

$$\mu_i = \frac{1}{n_i}\sum_{\mathbf{x}\in C_i}\mathbf{x},\ n_i \text{ is the size of class } C_i.$$

What are x, μ, μ_i, n_i N?

- Class-specific covariance matrix:

$$\mathbf{S}_i = \frac{1}{n_i}\sum_{\mathbf{x}\in C_i}(\mathbf{x}-\mu_i)(\mathbf{x}-\mu_i)^T$$

- Total mean vector (sample):

$$\mu = \frac{1}{N}\sum_{\mathbf{x}}\mathbf{x},\ N \text{ is the number of all samples.}$$

Statistical Facts (2)

- Within-class scatter:

$$\mathbf{S}_W = \sum_{i=1}^{C} \frac{n_i}{N} \mathbf{S}_i = \sum_{i=1}^{C} P_i \mathbf{S}_i, \; C \text{ is the class number.}$$

An estimate of the prior probability for class *i*

- Between-class scatter:

$$\mathbf{S}_B = \sum_{i=1}^{C} P_i (\mu_i - \mu)(\mu_i - \mu)^T$$

Note that in general a covariance matrix is symmetric and positive semi-definite with nonnegative eigenvalues.

Statistical Facts (3)

- Total covariance (sample):

$$\mathbf{S}_T = \frac{1}{N}\sum_{\mathbf{x}}(\mathbf{x}-\mu)(\mathbf{x}-\mu)^T$$
$$= \mathbf{S}_W + \mathbf{S}_B.$$

see next slide

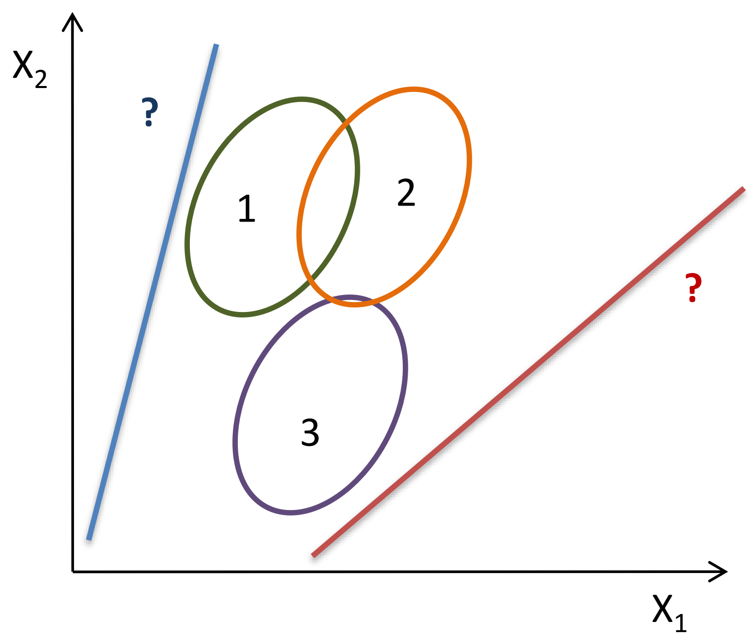
Proof

$$
\begin{aligned}
\mathbf{S}_T &= \frac{1}{N}\sum_{\mathbf{x}}(\mathbf{x}-\mu)(\mathbf{x}-\mu)^T \\
&= \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{n_i}(\mathbf{x}_{ij}-\mu)(\mathbf{x}_{ij}-\mu)^T \ , \ \mathbf{x}_{ij}\in C_i \\
&= \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{n_i}\left[(\mathbf{x}_{ij}-\mu_i)+(\mu_i-\mu)\right]\left[(\mathbf{x}_{ij}-\mu_i)+(\mu_i-\mu)\right]^T \\
&= \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{n_i}\begin{bmatrix}(\mathbf{x}_{ij}-\mu_i)(\mathbf{x}_{ij}-\mu_i)^T+(\mu_i-\mu)(\mu_i-\mu)^T+ \\ (\mathbf{x}_{ij}-\mu_i)(\mu_i-\mu)^T+(\mu_i-\mu)(\mathbf{x}_{ij}-\mu_i)^T\end{bmatrix} \\
&= \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{n_i}\left[(\mathbf{x}_{ij}-\mu_i)(\mathbf{x}_{ij}-\mu_i)^T+(\mu_i-\mu)(\mu_i-\mu)^T\right] \\
&= \frac{1}{N}\sum_{i=1}^{C}n_i\mathbf{S}_i+\frac{1}{N}\sum_{i=1}^{C}n_i(\mu_i-\mu)(\mu_i-\mu)^T = \boxed{\mathbf{S}_W+\mathbf{S}_B}
\end{aligned}
$$

Do it by yourself for 5 minutes!

Linear Discriminant Analysis - Problem (1)

- **PROBLEM: To separate populations**
  - Suppose we have C classes from n-dimensional distributions. We want to project these classes onto a p-dimensional subspace ($p < n$) so that the variation between the classes is **as large as possible**, relative to the variation within the classes.



Linear Discriminant Analysis - Problem (2)

- Practically speaking, after the projection, we want

class means to be as **far apart** from each other as possible	⇒	the **between-class** scatter to be **large**
samples from the same class to be as **close** to their means as possible	⇒	the **within-class** scatter to be **small**

This technique was developed by R. A. Fisher (1936) for **the two-class case** and extended by C. R. Rao (1948) to handle **the multiclass case**.

LDA: Two-class

- LDA seeks to reduce dimensionality while preserving as much of the class discriminatory information as possible
- Assume we have a set of D-dimensional samples $\{x^1, x^2, \dots x^N\}$, N_1 of which belong to class ω_1, and N_2 to class ω_2
- We seek to obtain a scalar y by projecting the samples x onto a line

$$y = w^T x$$

- Of all the possible lines we would like to select the one that maximizes the separability of the scalars

LDA: Two-class

- The mean vector of each class in x-space and y-space is

$$\mu_i = \frac{1}{n_i} \sum_{x \in \omega_i} x \text{ and } \widetilde{\mu_i} = w^T \mu_i$$

- We could then choose the distance between the projected means $\tilde{\mu}_i$ as our objective function

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

LDA: Two-class

- Fisher suggested maximizing the difference between the means, normalized by a measure of the within-class scatter
- For each class we define the scatter, an equivalent of the variance, as

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

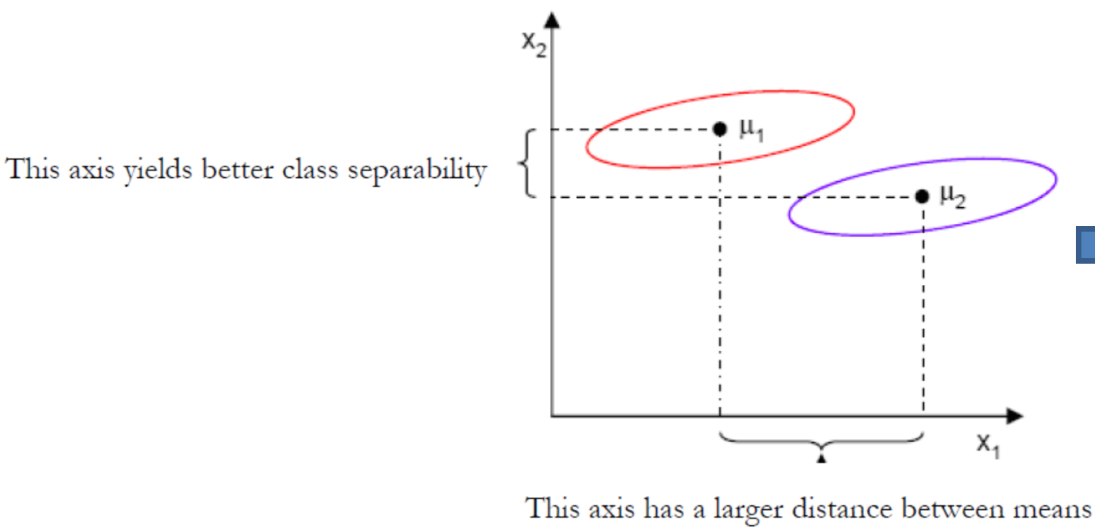
where $(\tilde{S}_1^2 + \tilde{S}_2^2)$ is the within-class scatter of the projected examples.

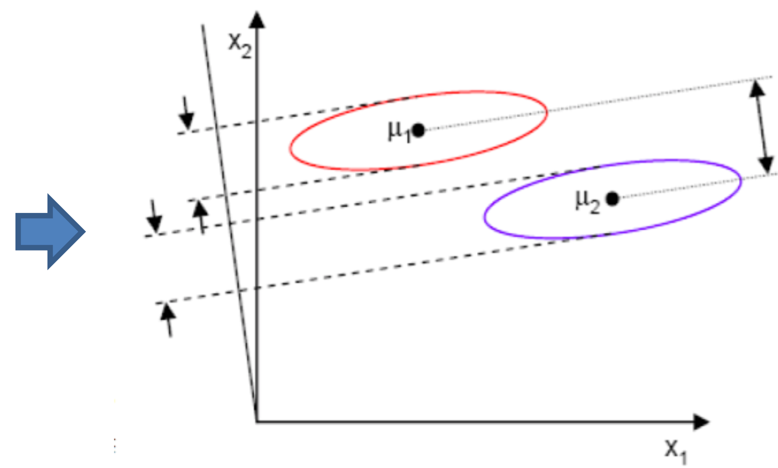
LDA: Two-class

- The Fisher linear discriminant is defined as the linear function $w^T x$ that maximizes the criterion function

$$J(w) = \frac{|\tilde{\mu}_1 - \tilde{\mu}_2|^2}{\tilde{S}_1^2 + \tilde{S}_2^2}$$

- Therefore, we are looking for a projection where examples from the same class are projected very close to each other and, at the same time, the projected means are as farther apart as possible





LDA: Two-class

- To find the optimum w^*, we must express $J(w)$ as a function of w
- As aforementioned, we define S_W as the within-class scatter matrix in feature space x

$$S_1 + S_2 = S_w$$

- The scatter of the projection y can then be expressed as a function of the scatter matrix in feature space x

$$\tilde{S}_i^2 = \sum_{y \in \omega_i} (y - \tilde{\mu}_i)^2 = \sum_{y \in \omega_i} (w^T x - w^T \mu_i)^2$$
$$= \sum_{y \in \omega_i} w^T (x - \mu_i)(x - \mu_i)^T w = w^T S_i w$$
$$\tilde{S}_1^2 + \tilde{S}_2^2 = w^T S_w w$$

LDA: Two-class

- Similarly, the difference between the projected means can be expressed in terms of the means in the original feature space

$$
\begin{aligned}
(\tilde{\mu}_1 - \tilde{\mu}_2)^2 &= (w^T \mu_1 - w^T \mu_2)^2 \\
&= w^T (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T w \\
&= w^T S_B w
\end{aligned}
$$

- The matrix S_B is the between-class scatter. Note that, since S_B is the outer product of two vectors, its rank is at most one

LDA: Two-class

- We can finally express the Fisher criterion in terms of S_W and S_B as

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

- To find the maximum of $J(w)$, we **derive** and **equate to zero**

$$\frac{d}{dw} J(w) = (w^T S_W w) \frac{d(w^T S_B w)}{dw} - (w^T S_B w) \frac{d(w^T S_W w)}{dw}$$

$$= (w^T S_W w) S_B w \; - (w^T S_B w) S_W w = 0$$

- Dividing by $w^T S_W w$

$$\frac{w^T S_W w}{w^T S_W w} S_B w \; - \frac{w^T S_B w}{w^T S_W w} S_W w = 0$$

$$S_B w - J(w) S_W w = 0 \Rightarrow S_W^{-1} S_B w = J(w) w$$

LDA: Two-class

- Solving the generalized eigenvalue problem $S_W^{-1} S_B w = \lambda w, where\ \lambda = J(w) = scalar$ yields w^* is the eigenvector of $\boldsymbol{S_W^{-1} S_B}$

LDA: Multi-class

- Instead of **one projection** y, we will now seek $\mathbf{(C-1)}$ projections $[y_1, y_2, \ldots y_{C-1}]$ by means of $(C-1)$ projection vectors w_i arranged by columns into a projection matrix $W=[w_1|w_2|\ldots|w_{C-1}]$:

$$y_i = w_i^T x \Rightarrow y = W^T x$$

- From our derivation for the two-class problem, we have the scatter matrices <u>for the projected samples</u>

$$\tilde{S}_W = W^T S_W W$$
$$\tilde{S}_B = W^T S_B W$$

LDA: Multi-class

- We look for a projection that maximizes the ratio of between-class to within-class scatter. Since the projection is no longer a scalar (it has $C{-}1$ dimensions), we use the determinant of the scatter matrices to obtain a scalar objective function

$$J(W) = \frac{|\tilde{S}_B|}{|\tilde{S}_W|} = \frac{|W^T S_B W|}{|W^T S_W W|}$$

- We seek the projection matrix $W*$ that maximizes this ratio. The optimal projection matrix $W*$ is the one whose columns are the eigenvectors corresponding to the largest eigenvalues of the following generalized eigenvalue problem.

$$W^* = \arg\max \frac{|W^T S_B W|}{|W^T S_W W|} \Rightarrow (S_B - \lambda_i S_W) w_i^* = 0$$

where $\lambda_i = J(w_i) = scalar$

Some Remarks

- The solving procedure of LDA is **lightweight**.
- We can see that if an optimization problem can be solved as a **Generalized Eigenvalue Decomposition** problem $\mathbf{B}\boldsymbol{\theta}_i = \lambda_i \mathbf{A}\boldsymbol{\theta}_i$ with $\boldsymbol{\theta}_i$ and λ_i being the i-th eigenvector and eigenvalue of $\mathbf{A}^{-1}\mathbf{B}$.
- Unlike principal component analysis (PCA), the linear discriminant transformation W from the original variates $x_1,\ldots,x_n$ to the new variates $y_1,\ldots,y_p$ is **not necessarily orthogonal**.

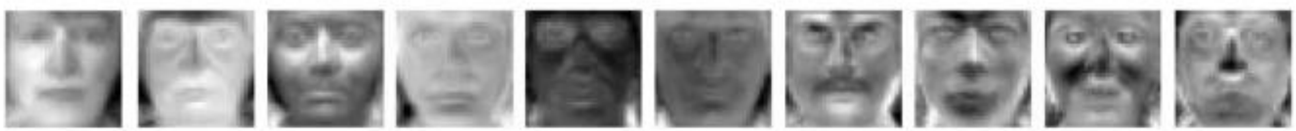
Classification with LDA

- First, select the number of feature dimension for the low-dimensional feature space (may use PCA beforehand).

- Later, use Nearest Neighbor or other classifiers.

Basis Visualization

Sample Images from YALE database



(a)

(b)

The first 10 (a) Eigenfaces, (b) Fisherfaces, calculated from the face images in the YALE database.

(visualization of LDA projection is called Fisherface)

Question

- Shall LDA always be better than PCA for classification task?

Is LDA always better than PCA?

- Case Study: PCA versus LDA
  - A. Martinez, A. Kak, "PCA versus LDA", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228-233, 2001.

- Is LDA always better than PCA?
  - There has been a tendency in the computer vision community to prefer LDA over PCA.
  - This is mainly because LDA deals directly with discrimination between classes while PCA does not pay attention to the underlying class structure.
  - This paper shows that when the training set is small, PCA can outperform LDA.
  - When the number of samples is large and representative for each class, LDA outperforms PCA.

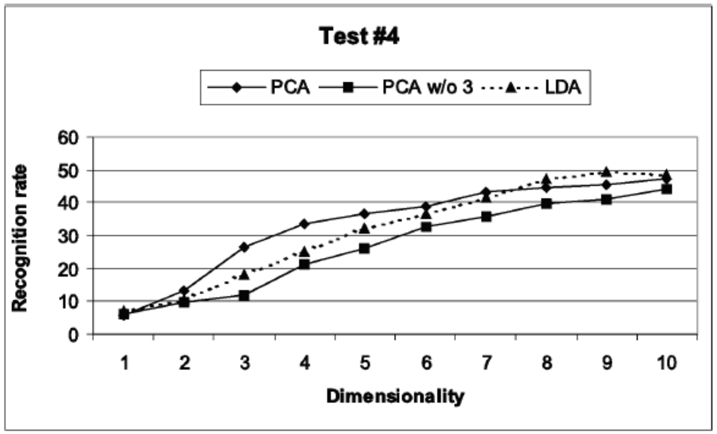
Linear Discriminant Analysis (LDA)

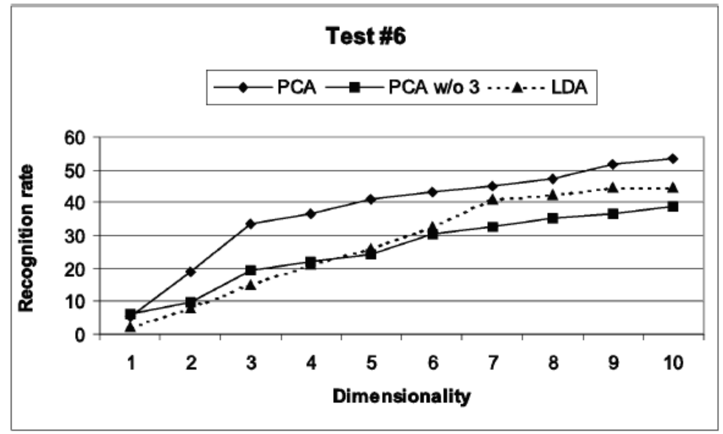
- Is LDA always better than PCA? – cont.

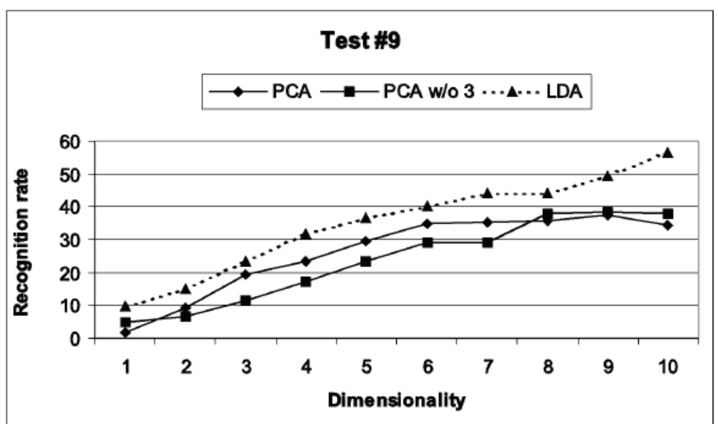
Linear Discriminant Analysis (LDA)

- Is LDA always better than PCA? – cont.

(LDA is not always better when the sample is small)





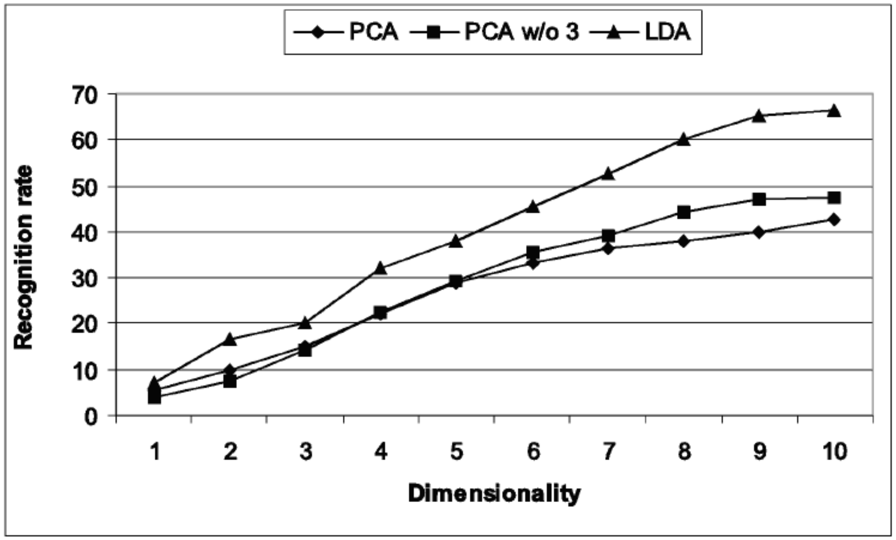


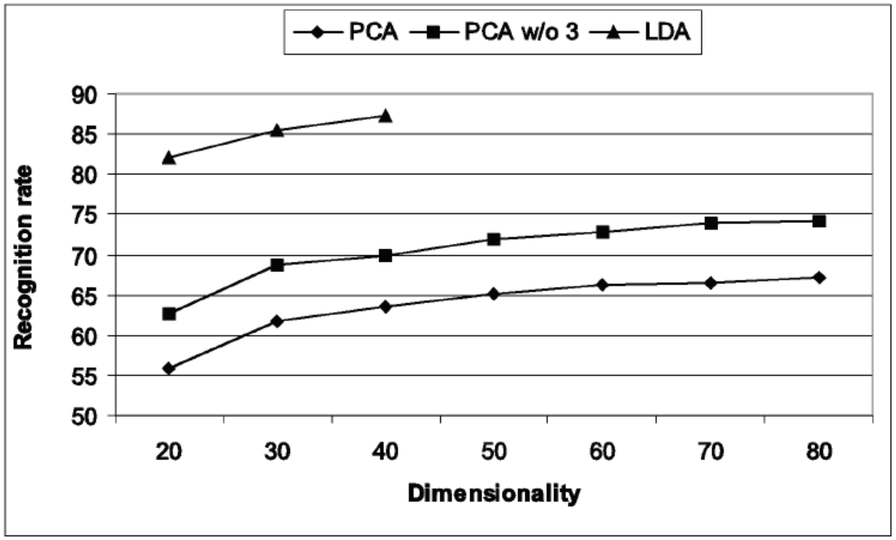
Results obtained for each of the three algorithms using a small dataset for training.

Linear Discriminant Analysis (LDA)

- Is LDA always better than PCA? – cont.

(LDA outperforms PCA when the sample is large)

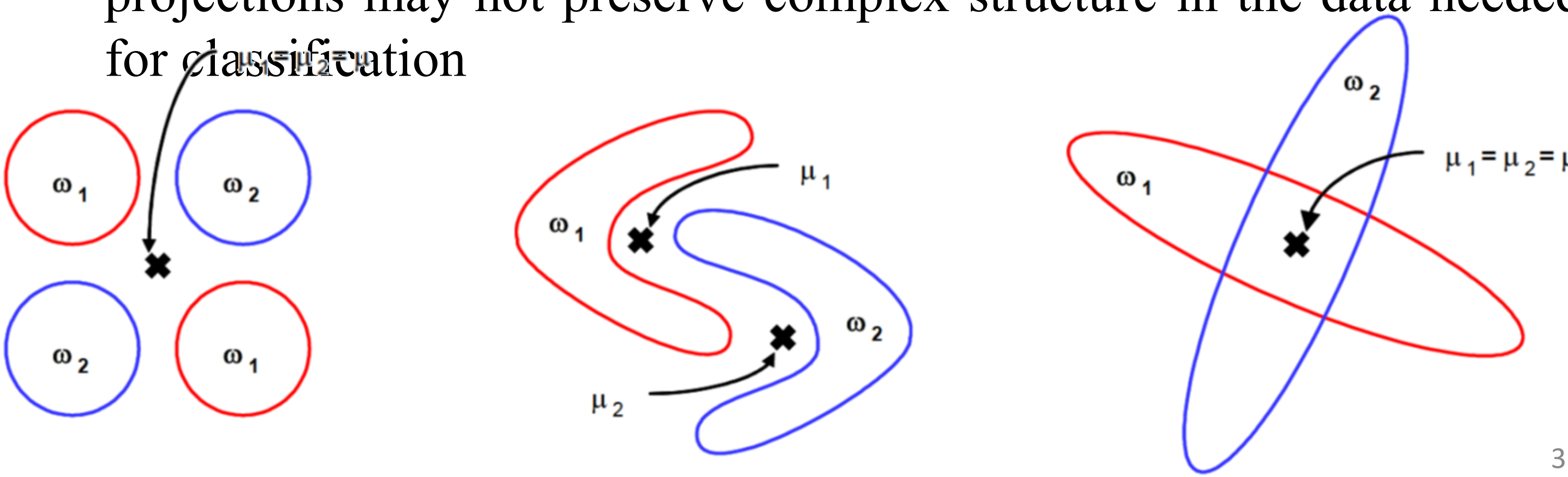




Results obtained for each of the three algorithms using a larger data set for training.

Limitations

- **LDA produces at most $C - 1$ feature projections**
  - If the classification error estimates establish that more features are needed, some other method must be employed to provide those additional features
- **LDA is a parametric method (it assumes unimodal Gaussian likelihoods)**
  - If the distributions are significantly non-Gaussian, the LDA projections may not preserve complex structure in the data needed for classification



Limitations

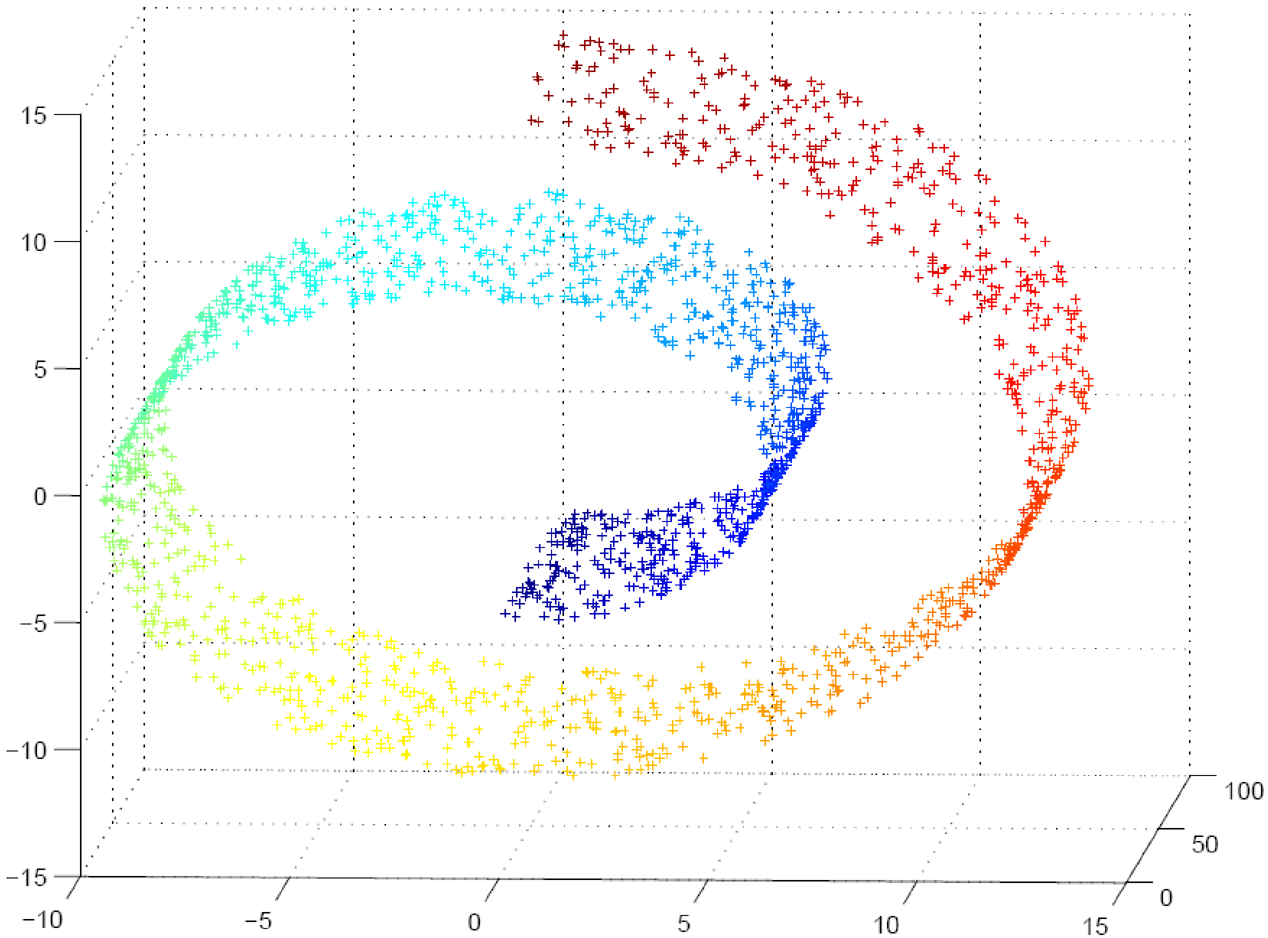
- LDA does not seem to be superior to PCA when the training data set is small.
- If testing sample does not follow the distribution of the training samples, the performance may be worse

Graph Embedding

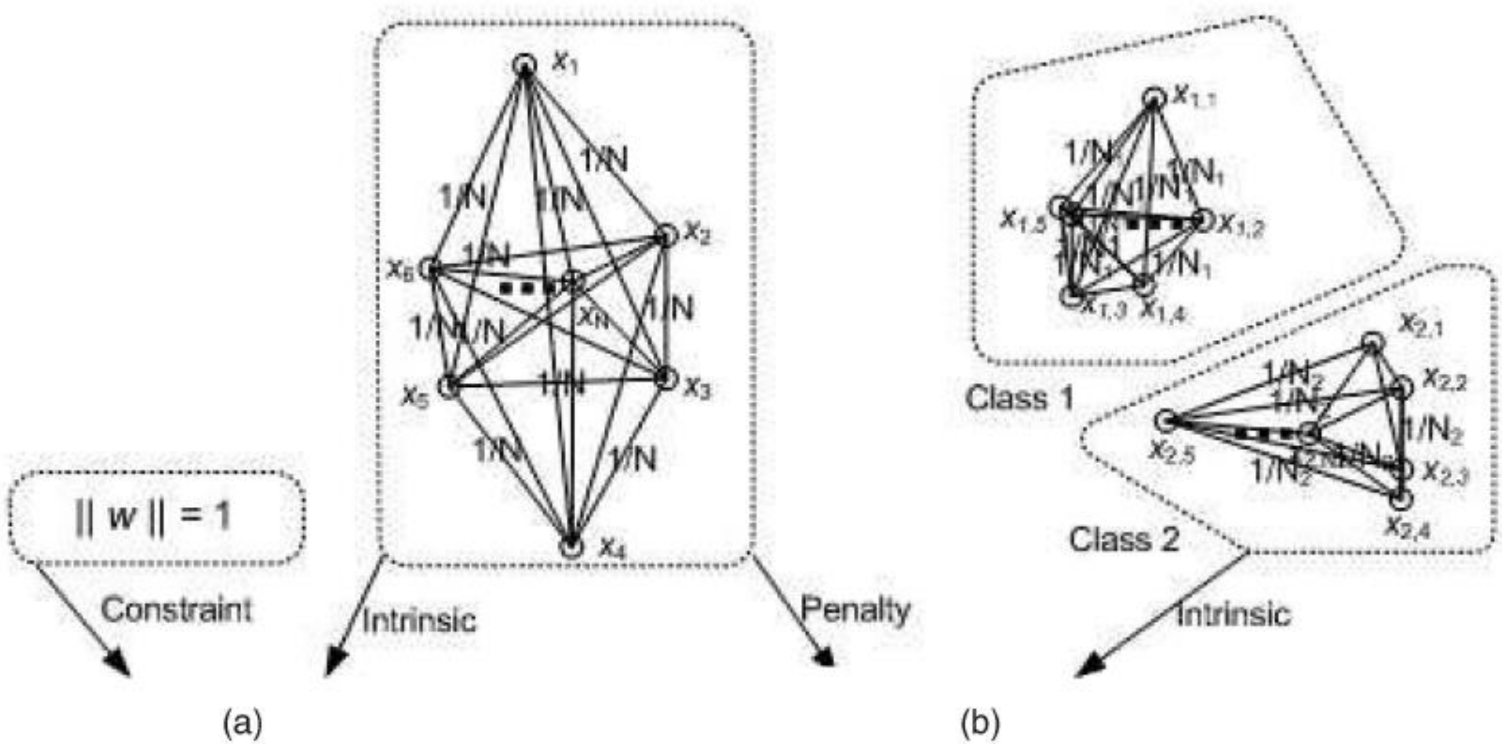
A General Framework for Feature Extraction

Shuicheng Yan, Dong Xu et al.: Graph Embedding and Extensions: A General Framework for Dimensionality Reduction. T-PAMI 2007.

Linear Subspace vs. Manifold

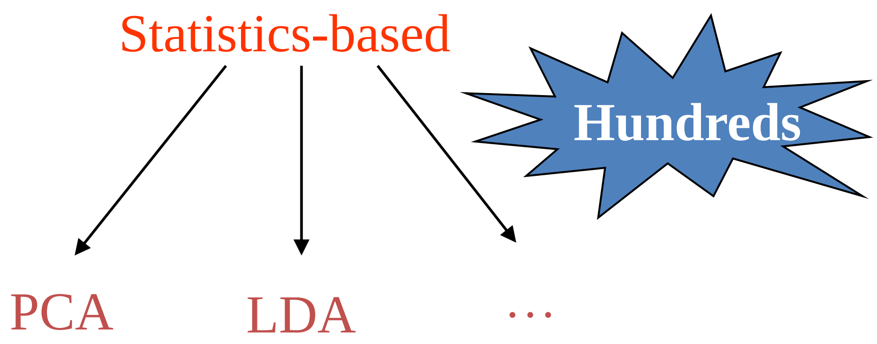


Examples for PCA and LDA



The adjacency graphs for PCA and LDA. (a) Constraint and intrinsic graph in PCA. (b) Penalty and intrinsic graphs in LDA.

Dimensionality Reduction Algorithms



- Any *common perspective* to understand and explain these dimensionality reduction algorithms? Or any *unified formulation* that is shared by them?

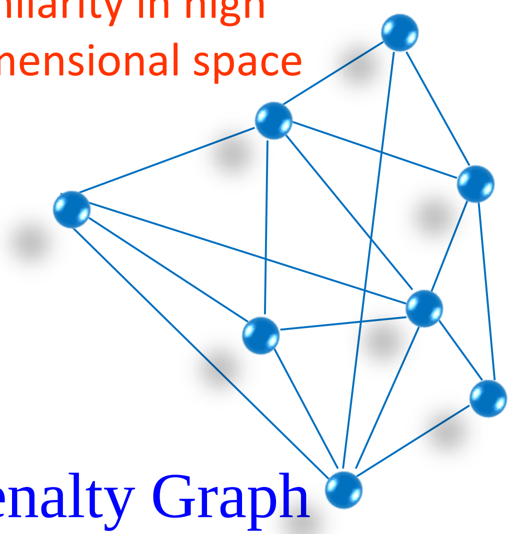
- Any *general tool* to guide developing new algorithms for dimensionality reduction?

Direct Graph Embedding

Intrinsic Graph:

$G = [x_i, S_{ij}]$

Similarity in high dimensional space



Penalty Graph

$G^P = [x_i, S_{ij}^P]$

S, S^P: *Similarity matrix (graph edge)*

L, B: *Laplacian matrix from* S, S^P;

$L = D - S, \quad D_{ii} = \sum_{j \neq i} S_{ij} \quad \forall i$

Data in high-dimensional space and low-dimensional space (assumed as 1D space here):

$X = [x_1, x_2, \ldots, x_N] \quad y = [y_1, y_2, \ldots, y_N]^T$

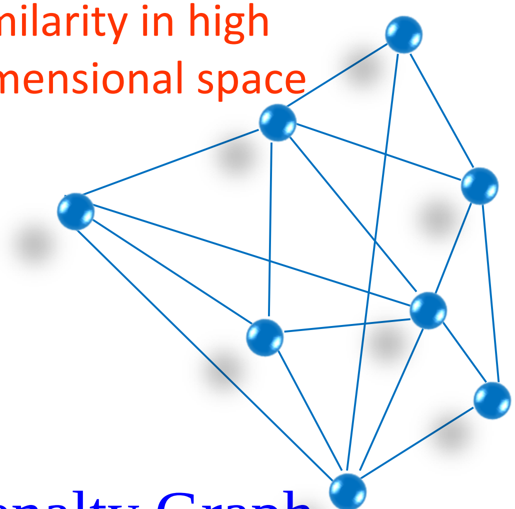
Target: search for a mapping $y_i = f(x_i)$ to Preserve / Avoid these graph similarities

Direct Graph Embedding -- Continued

Intrinsic Graph:

$G = [x_i, S_{ij}]$

Similarity in high dimensional space



Penalty Graph

$G^P = [x_i, S_{ij}^P]$

S, S^P: *Similarity matrix (graph edge)*

L, B: *Laplacian matrix from* S, S^P;

$$L = D - S, \quad D_{ii} = \sum_{j \neq i} S_{ij} \quad \forall i$$

Data in high-dimensional space and low-dimensional space (assumed as 1D space here):

$$X = [x_1, x_2, \ldots, x_N] \quad y = [y_1, y_2, \ldots, y_N]^T$$

Criterion to Preserve Graph Similarity:

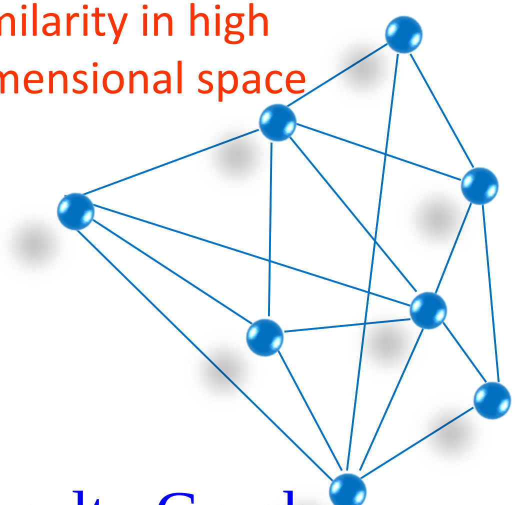
$$y^* = \underset{\substack{y^T y = 1 \text{ or} \\ y^T B y = 1}}{\arg\min} \boxed{\sum_{i \neq j} \| y_i - y_j \|^2 S_{ij}}$$

Direct Graph Embedding -- Continued

Intrinsic Graph:

$G = [x_i, S_{ij}]$

Similarity in high dimensional space



Penalty Graph

$G^P = [x_i, S_{ij}^P]$

S, S^P: Similarity matrix (graph edge)

L, B: Laplacian matrix from S, S^P;

$L = D - S, \quad D_{ii} = \sum_{j \neq i} S_{ij} \quad \forall i$

Data in high-dimensional space and low-dimensional space (assumed as 1D space here):

$X = [x_1, x_2, ..., x_N] \quad y = [y_1, y_2, ..., y_N]^T$

Criterion to Preserve Graph Similarity:

$$y^* = \underset{\boxed{y^T y = 1 \text{ or } y^T B y = 1}}{\arg\min} \boxed{\sum_{i \neq j} \| y_i - y_j \|^2 S_{ij}}$$

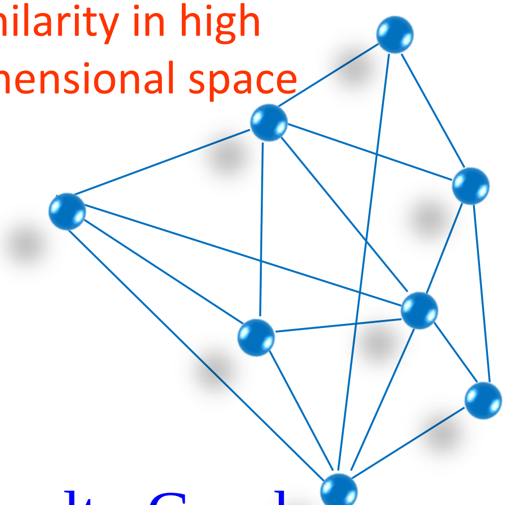
Special case B is Identity matrix (*Scale normalization)*

Direct Graph Embedding -- Continued

Intrinsic Graph:

$G = [x_i, S_{ij}]$

Similarity in high dimensional space



Penalty Graph

$G^P = [x_i, S_{ij}^P]$

S, S^P: *Similarity matrix (graph edge)*

L, B: *Laplacian matrix from* S, S^P;

$$L = D - S, \quad D_{ii} = \sum_{j \neq i} S_{ij} \quad \forall i$$

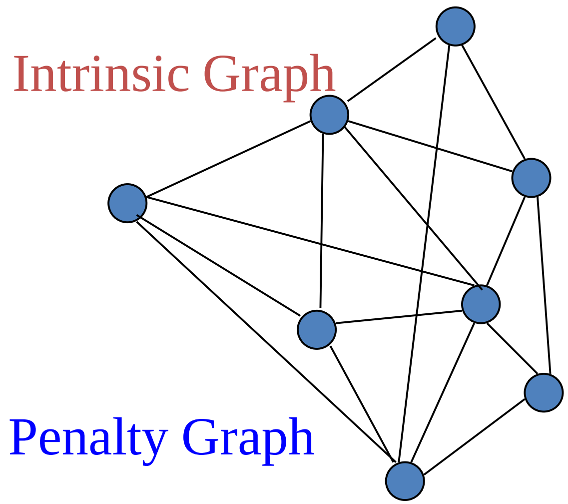
Data in high-dimensional space and low-dimensional space (assumed as 1D space here):

$$X = [x_1, x_2, \ldots, x_N] \quad y = [y_1, y_2, \ldots, y_N]^T$$

Criterion to Preserve Graph Similarity:

$$y^* = \underset{\substack{y^T y=1 \text{ or} \\ y^T By=1}}{\arg\min} \sum_{i \neq j} \| y_i - y_j \|^2 S_{ij} = \underset{\substack{y^T y=1 \text{ or} \\ y^T B y=1}}{\arg\min} y^T L y$$

Linearization



Linear mapping function

$$y = X^{T} w$$

Objective function in Linearization

$$w^{*} = \underset{\substack{w^{T} w=1 \ or \\ w^{T} XBX^{T} w=1}}{\arg\min} \; w^{T} XLX^{T} w$$

Common Formulation

Intrinsic graph

Penalty graph

S, S^P: *Similarity matrix*

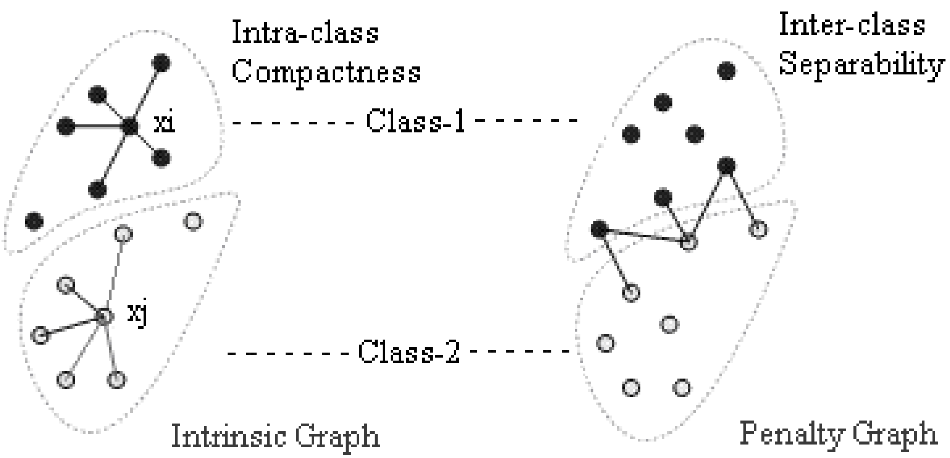
L, B: *Laplacian matrix from S, S^P;*

Linearization

$$w^* = \underset{\substack{w^T w=1 \text{ or} \\ w^T X \boxed{B} X^T w=1}}{\arg\min}\ w^T X \boxed{L} X^T w$$

The solutions are obtained by solving the generalized eigenvalue decomposition problem $\boldsymbol{\tilde{L}v = \lambda \tilde{B}v}$ where $\tilde{L} = XLX^T$ and $\tilde{B} = XBX^T$

New Algorithm: Marginal Fisher Analysis



Important Information for classification:

1) Label information
2) Local manifold structure (neighborhood or margin)

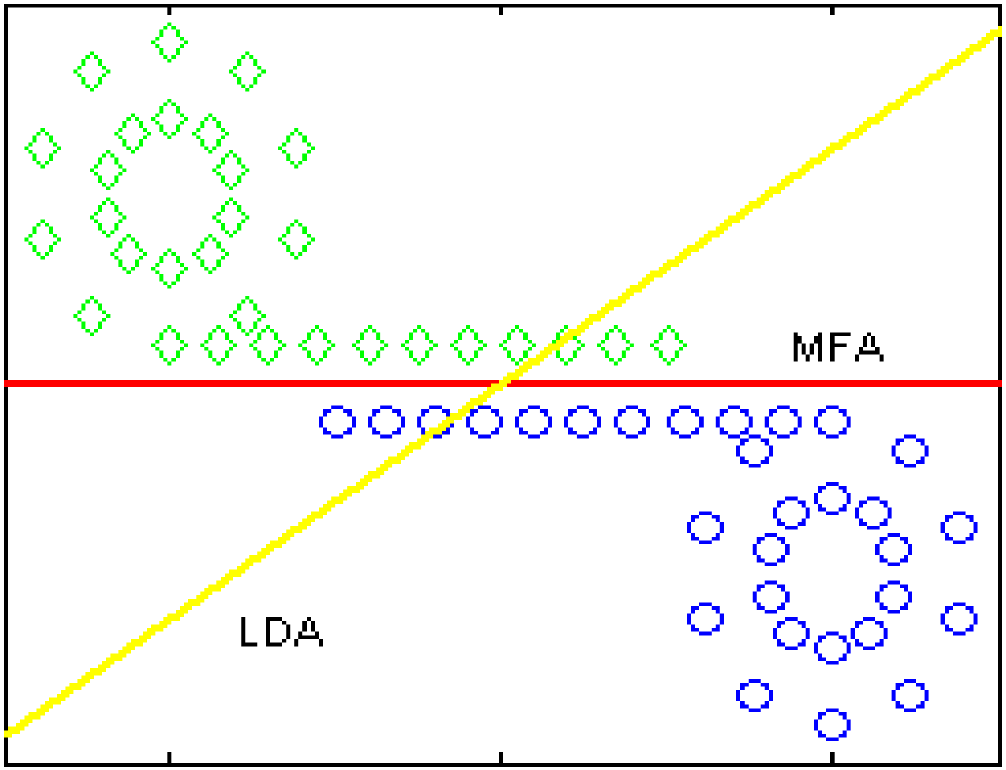
$S_{ij} =$ 1: if x_i is among the k_1-nearest neighbors of x_j in the same class;
0 : otherwise

$S_{ij}^{P} =$ 1: if the pair (i,j) is among the k_2 shortest pairs (from different classes) among the data set;
0: otherwise

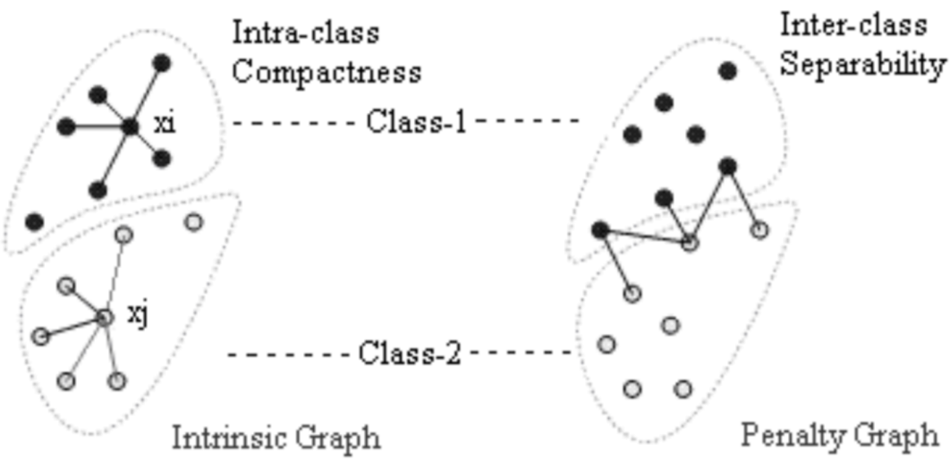
Marginal Fisher Analysis (MFA): Advantage

- MFA advantage: (compare with LDA)
  - The number of available projection directions is much larger
  - No assumption on the data distribution, more general for discriminant analysis
  - The interclass margin can better characterize the separability of different classes

Marginal Fisher Analysis (MFA): Advantage



Marginal Fisher Analysis



- Constructing the intraclass compactness and interclass separability graphs.
- In the **intraclass compactness** graph, for each sample xi, set the adjacency matrix $W_{ij} = W_{ji} = 1$ if x_i is among the k_1-nearest neighbors of x_j in the same class.
- In the **interclass separability** graph, for each class c, set the similarity matrix $W_{ij}^{p} = 1$ if the pair (i, j) if x_i is among the k_2-nearest neighbors of x_j in different class. .

Marginal Fisher Analysis

- Intraclass compactness (intrinsic graph)
- $\tilde{S}_c = \sum_i \sum_{i \in N_{k_1}^+(j) \ or \ j \in N_{k_1}^+(i)} \left\| w^T x_i - w^T x_j \right\|^2$
  $= 2w^T X(D - W)X^T w$

$$W_{ij} = \begin{cases} 1, if \ i \in N_{k_1}^+(j) or \ j \in N_{k_1}^+(i) \\ 0, else \end{cases}$$

Marginal Fisher Analysis

- Interclass separability (penalty graph)
- $\tilde{S}_p = \sum_i \sum_{(i,j)\in P_{k_2}(c_i)\ or\ (i,j)\in P_{k_2}(c_j)} \left\| w^T x_i - w^T x_j \right\|^2$
  $= 2w^T X(D^P - W^P)X^T w$

$$W_{ij}^P = \begin{cases} 1, if\ (i,j) \in P_{k_2}(c_i) or\ (i,j) \in P_{k_2}(c_j) \\ 0, else \end{cases}$$

Marginal Fisher Analysis

- From the linearization of graph embedding, we have Marginal Fisher Criterion

$$w^* = \arg\min_w \frac{w^T X(D - W)X^T w}{w^T X(D^P - W^P)X^T w}$$

, which is a special linearization of graph embedding with

$$L = D - W$$
$$B = D^P - W^P$$

Experiments: Face Recognition

Sample Images from CMU PIE database (after cropping)

ORL	G3/P7	G4/P6
PCA+LDA (Linearization)	87.9%	88.3%
PCA+MFA	89.3%	91.3%
PIE-1	G3/P7	G4/P6
PCA+LDA (Linearization)	65.8%	80.2%
PCA+MFA	71.0%	84.9%

Summary

- Optimization framework that unifies previous dimensionality reduction algorithms as special cases.
- A new dimensionality reduction algorithm: Marginal Fisher Analysis.
- Linear transformation has come to its end!
- Now, deep network is the trend (to study in L6)!

Papers to Read and Study

- H. Wang, S. Yan, D. Xu, X. Tang, and T. Huang. Trace Ratio vs. Ratio Trace for Dimensionality Reduction. In CVPR'07.

- Jieping Ye, Ravi Janardan, Cheonghee Park, and Haesun Park. An optimization criterion for generalized discriminant analysis on undersampled problems. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 26, No. 8, pp. 982—994, 2004. PDF

- A. Martinez, A. Kak, "PCA versus LDA", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 2, pp. 228-233, 2001.