Ch 08. Feature Reduction and Selection

Part 1 Feature Reduction

Error and Dimensionality

- For example

$$P(\mathbf{x}|\omega_j) \sim N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}) \;\; j = 1, 2$$

$$P(\omega_1) = P(\omega_2)$$

  - Bayes error probability

$$P(e) = \frac{1}{\alpha} \int_{r/2}^{\infty} e^{-u^2/2} du$$

Mahalanobis distance from $\boldsymbol{\mu}_1$ to $\boldsymbol{\mu}_2$

$$r^2 = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^t \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

  - As r goes up, the probability of error $P(e)$ goes down
  - $r \to \infty, \;\; P(e) \to 0$
  - Suppose each feature is independent：

The introduction of new features can increase r and thus reduce the error probability $P(e)$

$$\boldsymbol{\Sigma} = diag(\sigma_1, \sigma_2, \ldots, \sigma_d) \qquad r^2 = \sum_{i=1}^{d} \left(\frac{\mu_{i1} - \mu_{i2}}{\sigma_i} \right)^2$$

Curse of Dimensionality

- In practical application
  - When the number of features increases to a certain critical point, the continuous increase will lead to the performance of the classifier becomes worse — “Curse of dimensionality” (维度灾难)
  - Reason
    - The hypothetical probabilistic model does not match the real model
    - Due to the limited number of training samples, the estimate of probability distribution is inaccurate
    - ……
- For high-dimensional data, "Curse of dimensionality" makes it very difficult to solve the problem of pattern recognition. At this time, it is often required to reduce the dimension of feature vectors first

Dimensionality Reduction

- Feasibility of reducing eigenvector dimensionality
  - Eigenvectors often contain redundant information!
    - Some features may be **irrelevant** to the classification problem
    - There is a strong **correlation** between the features
- The way to Dimensionality Reduction
  - **Feature combination**
    - Combine several features to form a new feature
  - **Feature selection**
    - Select a subset of the existing feature set

Dimensionality Reduction

- The problem of dimensionality reduction
  - Linear transformation vs. Non-linear transformation
  - Use category label (supervised) vs. no category label (unsupervised)
  - Different training objectives
    - Minimize reconstruction errors （ **principal component analysis**, **PCA**, 主成分分析）
    - Maximize category separability （ **linear discriminant analysis**，**LDA**，线性判别分析）
    - Minimize classification error （**discriminative training**, 判别训练）
    - Projection that retains the most detail （**projection pursuit**, 投影寻踪）
    - Maximize independence between features （ **Independent Component Analysis**, **ICA**, 独立成分分析）

Principal Component Analysis (PCA)

- Represent d-dimensional samples with one-dimensional vectors
  - The sample is represented by points on a line (with a unit vector of e) passing through the sample mean **m**

$$\hat{\mathbf{x}}_k = \mathbf{m} + \boxed{a_k}\,\mathbf{e} \Longrightarrow \mathbf{x}_k$$

a_k alone determines $\hat{\mathbf{x}}_k$

  - Minimize square reconstruction errors

$$J_1(a_1,\ldots,a_n,\mathbf{e}) = \sum_{k=1}^{n} \left\| (\mathbf{m} + a_k\mathbf{e} - \mathbf{x}_k) \right\|^2 = \sum_{k=1}^{n} \left\| (a_k\mathbf{e} - (\mathbf{x}_k - \mathbf{m})) \right\|^2$$

$$= \sum_{k=1}^{n} a_k^2 \left\|\mathbf{e}\right\|^2 - 2\sum_{k=1}^{n} a_k \mathbf{e}^t(\mathbf{x}_k - \mathbf{m}) + \sum_{k=1}^{n} \left\|\mathbf{x}_k - \mathbf{m}\right\|^2$$

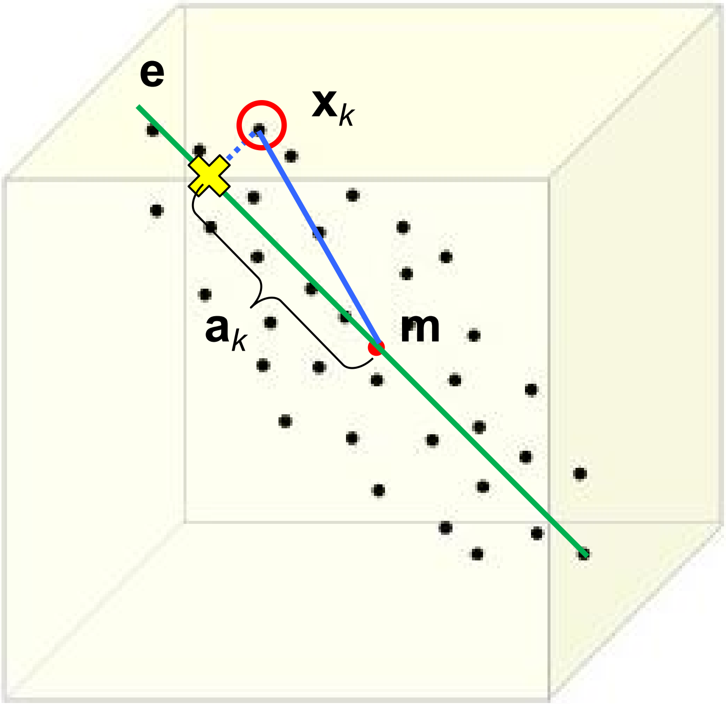
$$\frac{\partial J_1(a_1,\ldots,a_n,\mathbf{e})}{\partial a_k} = 2a_k - 2\mathbf{e}^t(\mathbf{x}_k - \mathbf{m}) = 0$$

$$a_k = \mathbf{e}^t(\mathbf{x}_k - \mathbf{m})$$

The projection of (x_k-m) onto e

Principal Component Analysis (PCA)

- Represent d-dimensional samples with one-dimensional vectors



Principal Component Analysis (PCA)

- Find the optimal direction of $\mathbf{e}$

$$a_k = \mathbf{e}^t(\mathbf{x}_k - \mathbf{m}) \qquad J_1(a_1,\ldots,a_n,\mathbf{e}) = \sum_{k=1}^{n} a_k^2 \|\mathbf{e}\|^2 - 2\sum_{k=1}^{n} a_k \mathbf{e}^t(\mathbf{x}_k - \mathbf{m}) + \sum_{k=1}^{n} \|\mathbf{x}_k - \mathbf{m}\|^2$$

$$J_1(\mathbf{e}) = \sum_{k=1}^{n} a_k^2 - 2\sum_{k=1}^{n} a_k^2 + \sum_{k=1}^{n} \|\mathbf{x}_k - \mathbf{m}\|^2$$

$$= -\sum_{k=1}^{n} [\mathbf{e}^t(\mathbf{x}_k - \mathbf{m})]^2 + \sum_{k=1}^{n} \|\mathbf{x}_k - \mathbf{m}\|^2$$

$$= -\sum_{k=1}^{n} \mathbf{e}^t(\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^t \mathbf{e} + \sum_{k=1}^{n} \|\mathbf{x}_k - \mathbf{m}\|^2$$

$$= -\mathbf{e}^t \mathbf{S} \mathbf{e} + \sum_{k=1}^{n} \|\mathbf{x}_k - \mathbf{m}\|^2$$

$$\mathbf{S} = \sum_{k=1}^{n} (\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^t$$

scatter matrix （散布矩阵）

Principal Component Analysis (PCA)

- Maximize $\mathbf{e}^t\mathbf{S}\mathbf{e}$ by **e** that minimize $J_1(\mathbf{e})$
  - Lagrangian multiplier method (s.t. $\mathbf{e}^t\mathbf{e}=1$)

$$u=\mathbf{e}^t\mathbf{S}\mathbf{e}-\lambda(\mathbf{e}^t\mathbf{e}-1)$$

$$\frac{\partial u}{\partial \mathbf{e}}=2\mathbf{S}\mathbf{e}-2\lambda\mathbf{e}=0$$

$$\boxed{\mathbf{S}\mathbf{e}=\lambda\mathbf{e}} \longrightarrow$$

λ is the eigenvalue of S (本征值)

e is the eigenvector of S (本征向量)

$$\mathbf{e}^t\mathbf{S}\mathbf{e}=\lambda\mathbf{e}^t\mathbf{e}=\lambda \longrightarrow$$

The maximum eigenvalue λ corresponds to the maximum value of $\mathbf{e}^t\mathbf{S}\mathbf{e}$

  - Conclusion: **e** is the eigenvector corresponding to the maximum eigenvalue of the scatter matrix

Principal Component Analysis (PCA)

- Extending one-dimensional a_k into $d'\,(d' \le d)$-dimensional space
  - Represent $\mathbf{x}_k$ by $\mathbf{y}_k = \begin{bmatrix} a_{k1} \\ a_{k2} \\ \vdots \\ a_{kd'} \end{bmatrix}$

$$\hat{\mathbf{x}}_k = \mathbf{m} + \sum_{i=1}^{d'} a_{ki}\mathbf{e}_i$$

  - Minimize squared error

$$J_{d'}(\mathbf{e}) = \sum_{k=1}^{n} \left\| \left(\mathbf{m} + \sum_{i=1}^{d'} a_{ki}\mathbf{e}_i \right) - \mathbf{x}_k \right\|^2$$

Principal Component Analysis (PCA)

- Extending one-dimensional a_k into $d'(d' \leq d)$-dimensional space
  - **Conclusion:**
    - The vector that minimizes the squared error $\mathbf{e}_1, \mathbf{e}_2, \ldots \mathbf{e}_{d'}$ is the eigenvector corresponding to the d' largest eigenvalues of the scatter matrix **S** respectively
    - **S** is a real symmetric matrix, so $\mathbf{e}_1, \mathbf{e}_2, \ldots \mathbf{e}_{d'}$ are orthogonal to each other
    - $\mathbf{e}_1, \mathbf{e}_2, \ldots \mathbf{e}_{d'}$ can be considered as unit vector bases of a subspace in the feature space
    - a_{ki} is the coefficient on $\mathbf{x}_k$ that corresponds to the base $\mathbf{e}_i$, or the projection on the $\mathbf{e}_i$
    - a_{ki} is called **principal component** （主成分）
    - Geometric meaning

    $\mathbf{e}_1, \mathbf{e}_2, \ldots \mathbf{e}_{d'}$ are straight lines along the direction of maximum variance of the data cloud

  - Using PCA, the d-dimensional data can be reduced to $d'(d' \leq d)$ dimensions, while minimizing the squared error between the downscaled data and the source data

Principal Component Analysis (PCA)

- Principal component analysis steps (d-dimensions is reduced to $d' \, (d' \le d)$-dimensions)

1. Calculate the scatter matrix **S**

$$\mathbf{S} = \sum_{k=1}^{n} (\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^t$$

2. Calculate the eigenvalue of **S** and the eigenvector

$$\mathbf{S}\mathbf{e} = \lambda \mathbf{e}$$

3. Sort the eigenvectors by their corresponding eigenvalues from the maximum to the minimum

4. The maximum d' eigenvectors are selected as projection vectors $\mathbf{e}_1, \mathbf{e}_2, \ldots \mathbf{e}_{d'}$, to form projection $d \times d'$ matrix **W**, in which the i-th column is $\mathbf{e}_i$

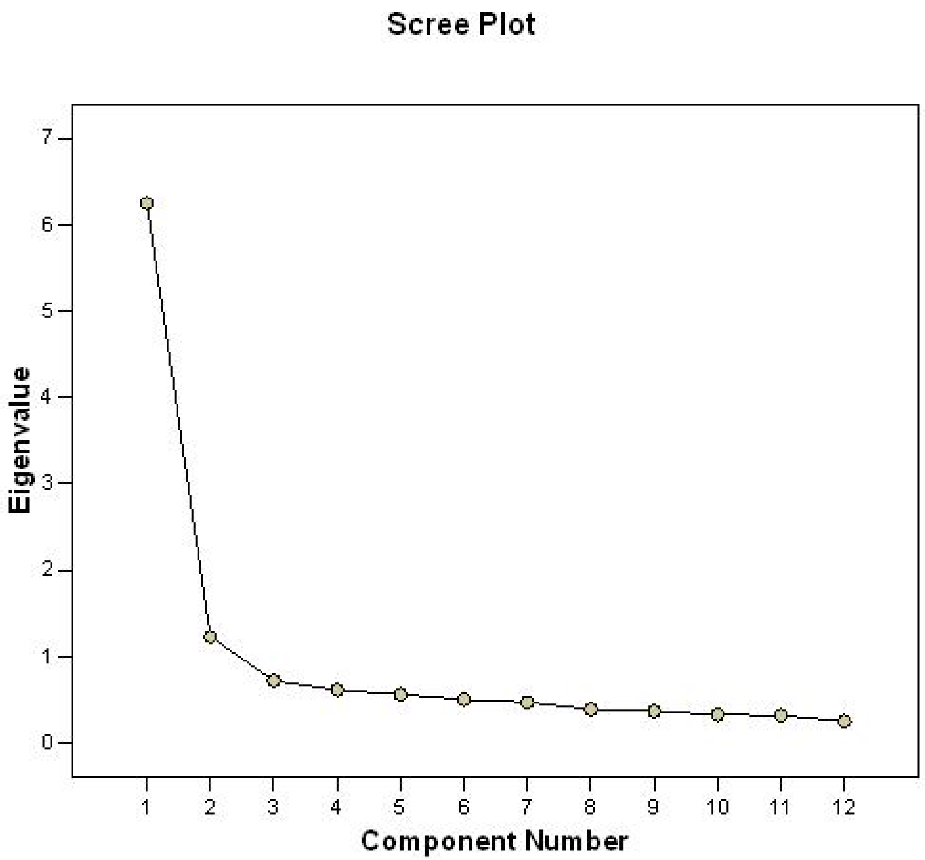
5. For an arbitrary d-dimensional sample **x**, its d' dimensional vector after dimensionality reduction by PCA is

$$\mathbf{y} = \mathbf{W}^t (\mathbf{x} - m)$$

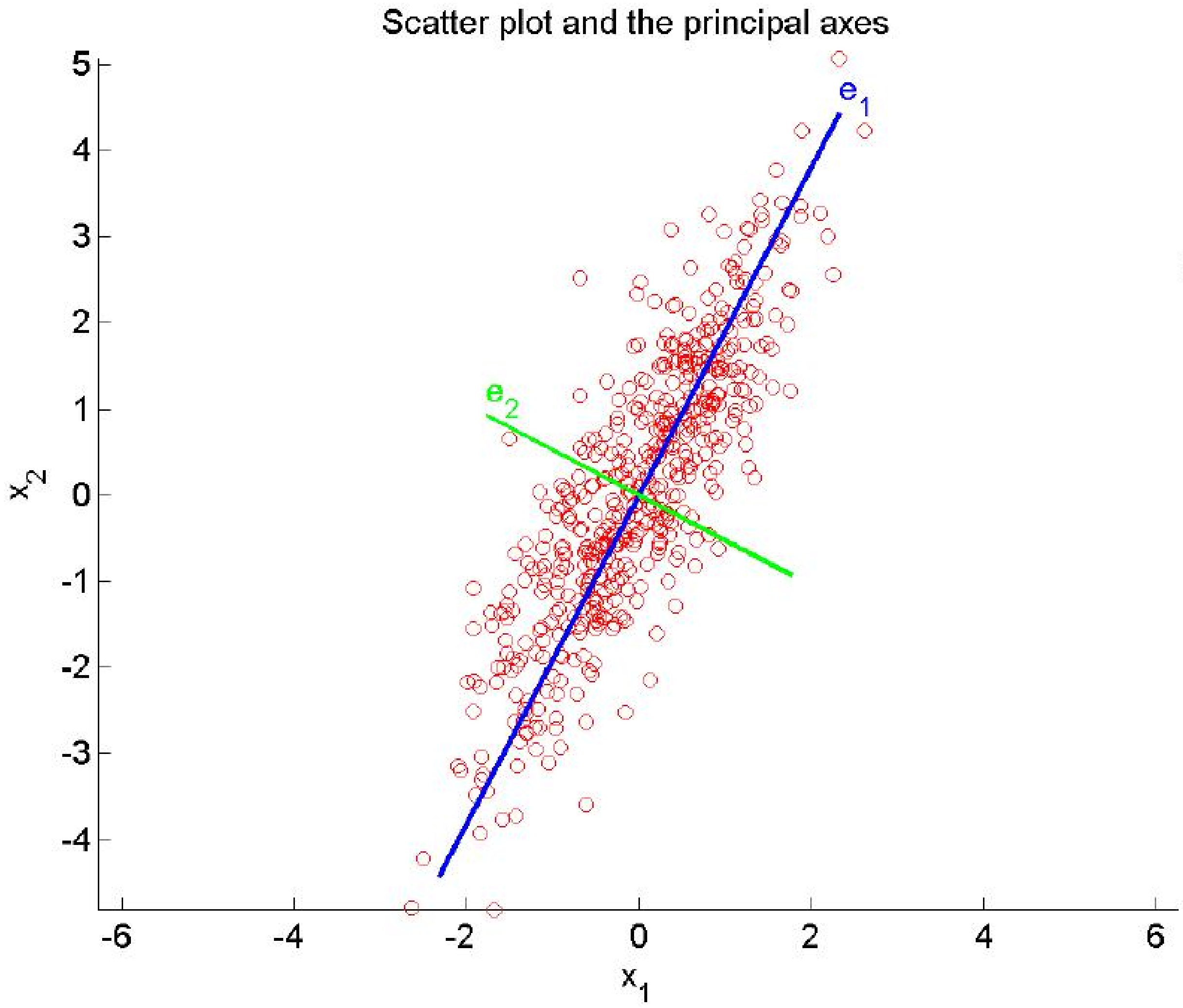
Principal Component Analysis (PCA)

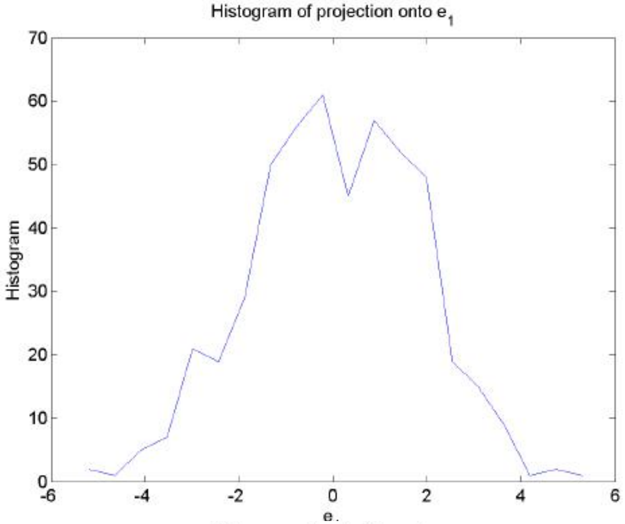
- In general, **the several maximum eigenvalues account for most of the sum of all eigenvalues**

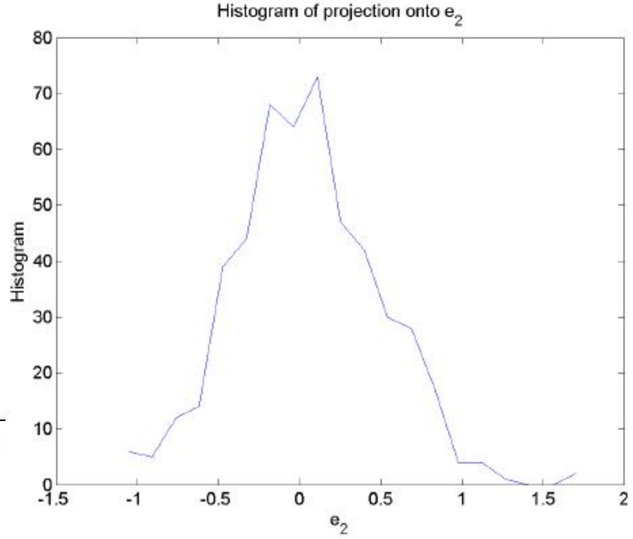
- **A few eigenvectors corresponding to the largest eigenvalues can represent the vast majority of the information in the original data, while the remaining small part (i.e. the information represented by the eigenvectors corresponding to the smaller eigenvalues) can generally be considered as data noise and lost**



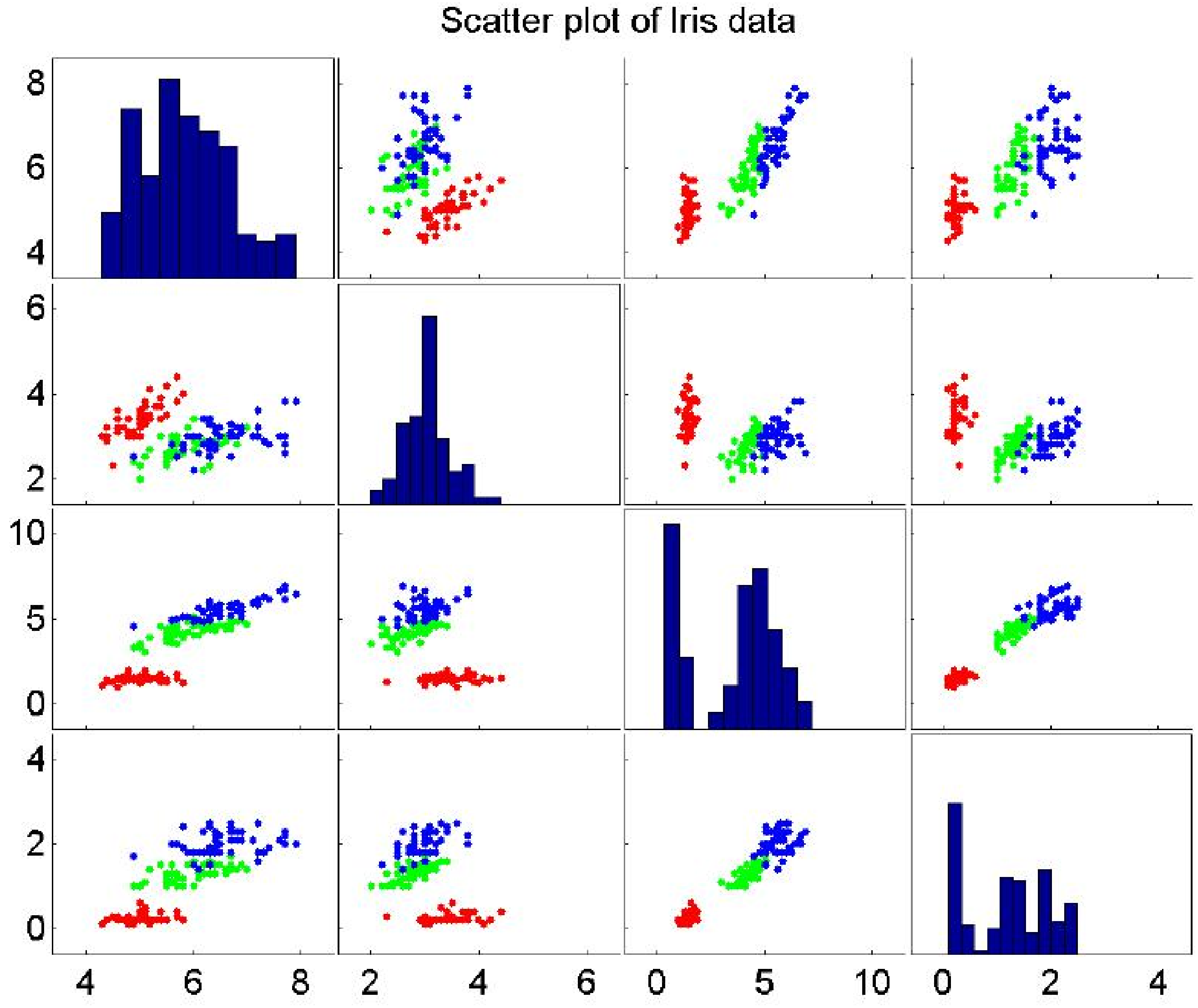
Principal Component Analysis (PCA)





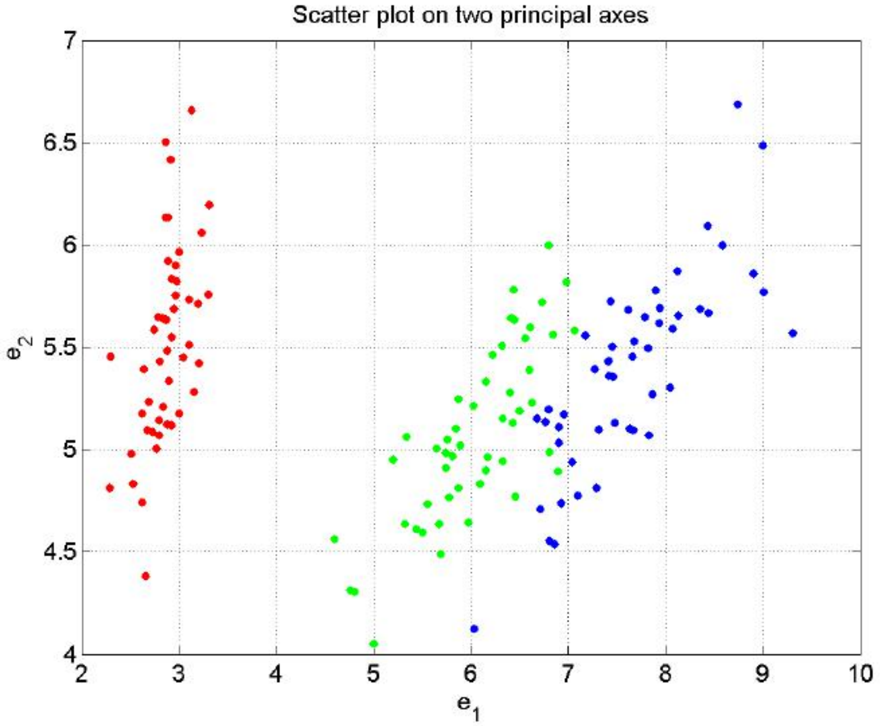


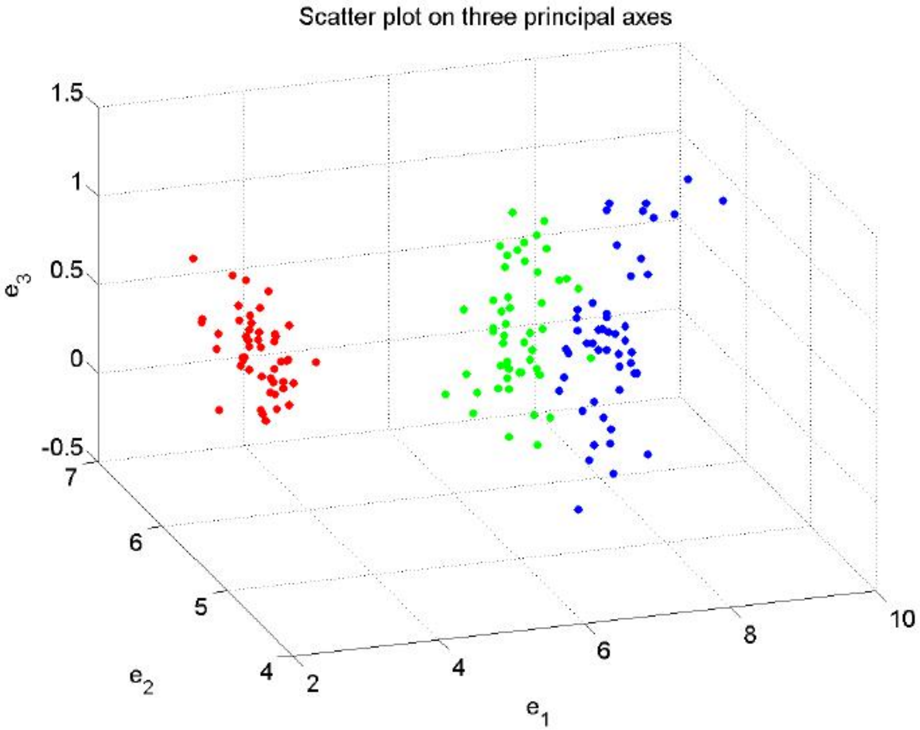
Principal Component Analysis (PCA)



- Dataset：**Iris**
- Original dimensionality：**4**

Principal Component Analysis (PCA)





Reduce to 2-dimensions by PCA

Reduce to 3-dimensions by PCA

Singular Value Decomposition (SVD)

- The eigenvalue decomposition of the scatter matrix $\mathbf{S}$ in PCA is computationally large, and it is very difficult to directly decompose the eigenvalue of $\mathbf{S}$ if the eigenvector dimension is high
- For example, PCA analysis of images:
  - Image：100×100
  - Scatter matrix：10000×10000

$$\mathbf{S} = \sum_{k=1}^{n} (\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^t$$

  - 10000×10000 matrix eigenvalue decomposition ?

$$\mathbf{Se} = \lambda \mathbf{e}$$

Space complexity and time complexity are unacceptable!

Singular Value Decomposition (SVD)

- Instead of performing an eigenvalue decomposition on S directly, use SVD to perform an eigenvalue decomposition on a smaller matrix
- **SVD theorem**
  - Let **A** be a $d \times n$ matrix of rank n, then there are two orthogonal matrices

    $$\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_d] \in \mathbb{R}^{d \times n} \qquad \mathbf{U}^T\mathbf{U} = \mathbf{I}$$

    $$\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n] \in \mathbb{R}^{n \times n} \qquad \mathbf{V}^T\mathbf{V} = \mathbf{I}$$

    and diagonal matrix $\mathbf{\Lambda} = \mathrm{diag}[\lambda_1, \lambda_2, \dots, \lambda_n] \in \mathbb{R}^{n \times n}$ $\qquad \lambda_1 \geqslant \lambda_2 \geqslant \dots \geqslant \lambda_n$

    satisfy $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{V}^T$

where $\lambda_i (i = 1, 2, \dots, n)$ is the non-zero eigenvalue of matrix $\mathbf{A}\mathbf{A}^T$ and $\mathbf{A}^T\mathbf{A}$, $\mathbf{u}_i$ and $\mathbf{v}_i$ are respectively the eigenvectors of $\mathbf{A}\mathbf{A}^T$ and $\mathbf{A}^T\mathbf{A}$ corresponding to λ_i. This decomposition is called the **singular value decomposition** (**SVD**) of matrix **A**, $\sqrt{\lambda_i}$ is the singular value of **A**

Singular Value Decomposition (SVD)

- Inference

$$\mathbf{A} = \mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{V}^T \Rightarrow \mathbf{U} = \mathbf{A}\mathbf{V}\mathbf{\Lambda}^{-\frac{1}{2}}$$

- **Use SVD to simplify the eigenvalue decomposition of S**

Scatter matrix $\mathbf{S} = \sum_{k=1}^{n} (\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^T = \mathbf{A}\mathbf{A}^T \in \mathbb{R}^{d \times d}$

where $\mathbf{A} = [\mathbf{x}_1 - \mathbf{m}, \mathbf{x}_2 - \mathbf{m}, \ldots, \mathbf{x}_n - \mathbf{m}] \in \mathbb{R}^{d \times n}$

Let $\mathbf{R} = \mathbf{A}^T \mathbf{A} \in \mathbb{R}^{n \times n}$

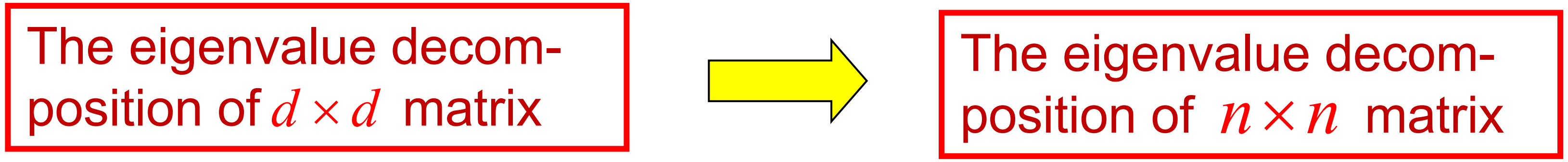
If $d > n$, then the eigenvalue decomposition for R is faster than the eigenvalue decomposition for **S**

For example, for most image training sets, the number of pixels in the image is much larger than the number of samples in the training set, i.e. $d \gg n$

Singular Value Decomposition (SVD)

- Perform the eigenvalue decomposition for R
  - eigenvalue ： $\lambda_i (i=1,2,\ldots,n)$
  - eigenvector： $\mathbf{v}_i$
  - According to $\mathbf{U}=\mathbf{A}\mathbf{V}\mathbf{\Lambda}^{-\frac{1}{2}}$, the eigenvector of $\mathbf{S}=\mathbf{A}\mathbf{A}^T$ is

$$\mathbf{u}_i=\frac{1}{\sqrt{\lambda_i}}\mathbf{A}\mathbf{v}_i$$

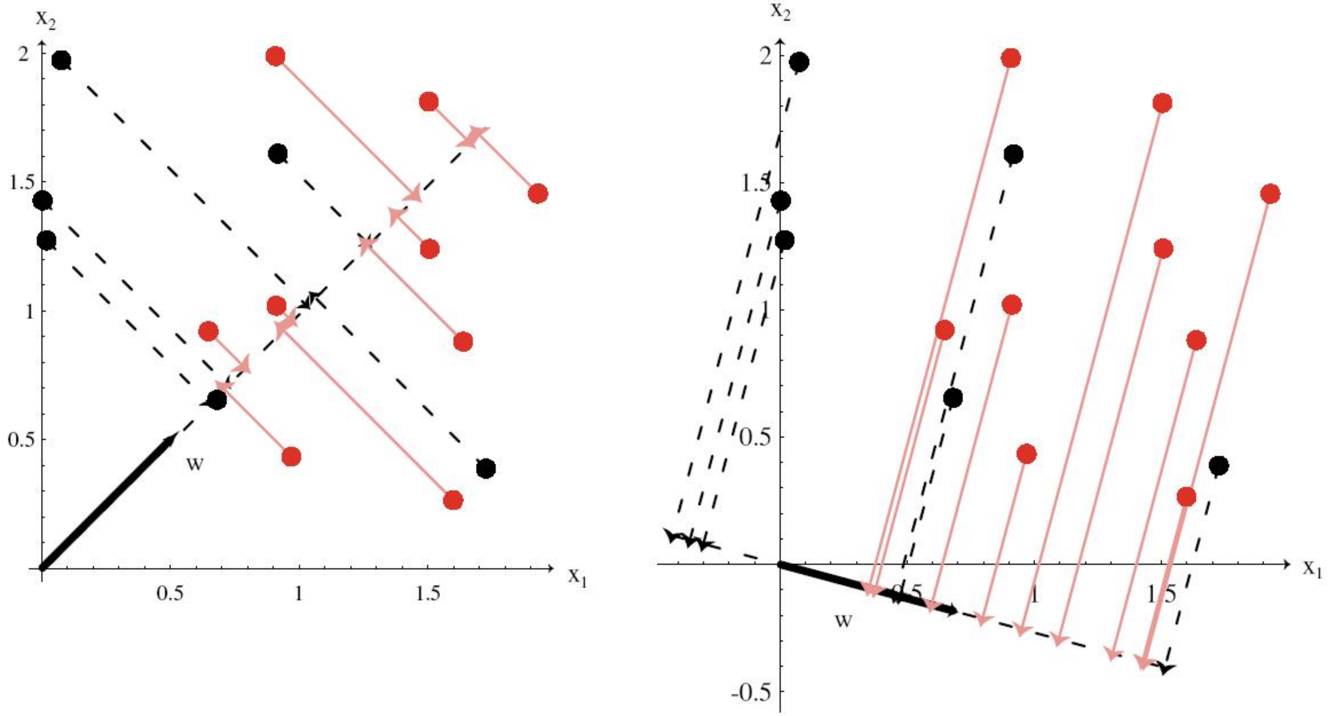


The eigenvalue decomposition of $d \times d$ matrix

The eigenvalue decomposition of $n \times n$ matrix

Fisher Linear Discriminant Analysis

- The PCA method looks for the principal axises used to represent the data efficiently (in the sense of least square error)
- Linear discriminant analysis (LDA) looks for directions that can be used to effectively classify



Fisher Linear Discriminant Analysis

- Suppose
  - n d-dimensional samples $\mathrm{x}_1, \ldots, \mathrm{x}_n$, which belong to category ω_1 and ω_2
  - where n_1 samples belonging to category ω_1 make up sample subset $\mathcal{D}_1$, n_2 samples belonging to category ω_2 make up sample subset $\mathcal{D}_2$
  - Projection in the direction of the unit vector $\mathbf{w}$: $y = \mathbf{w}^T \mathbf{x}$
  - Projection point $y_1, \ldots, y_n$ is also divided into two subsets $\mathcal{Y}_1$ and $\mathcal{Y}_2$ based on the category of source data
- Goal: projection points are easier to classify after projected onto w
  - Projection points of different categories should be separated as far as possible
  - Projection points of same categories should be as close as possible

Fisher Linear Discriminant Analysis

- Projection points of different categories should be separated as far as possible
  - Let m_i be the sample mean of i-th category

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in \mathcal{D}_i} \mathbf{x}$$

  - The sample mean value after projection

$$\widetilde{\mathrm{m}}_i = \frac{1}{n_i} \sum_{y \in \mathcal{Y}_i} y = \frac{1}{n_i} \sum_{x \in D_i} \mathbf{w}^t \mathbf{x} = \mathbf{w}^t \mathbf{m}_i$$

  - The distance between the means of the two categories of samples after projection

$$|\widetilde{m}_1 - \widetilde{m}_2| = |\mathbf{w}^t(\mathbf{m}_1 - \mathbf{m}_2)|$$

The greater this distance, the more separated the two categories of projection points are

Fisher Linear Discriminant Analysis

- Projection points of same categories should be as close as possible
  - Within-class scatter of projection

$$\tilde{s}_i^2 = \sum_{y \in \mathcal{Y}_i} (y - \tilde{m}_i)^2$$

  - The sum of the within-class scatters of projection for each class

$$\tilde{s}_1^2 + \tilde{s}_2^2$$

This total within-class scatter reflects the "tightness" of the classes behind the projection, the smaller it is, the closer the projection points are to each other within the same class

Fisher Linear Discriminant Analysis

- Fisher criterion function

$$J(\mathbf{w}) = \frac{|\tilde{m}_1 - \tilde{m}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

The distance between the two categories of sample means

Total within-class scatter

Maximize $J(w)$ by maximizing the between-class gap (numerator) while minimizing the within-class gap (denominator)

Fisher Linear Discriminant Analysis

- Represent $J(\mathbf{w})$ as the expressions for $\mathbf{w}$
  - Within-class scatter matrix of the original data space

$$\mathbf{S}_i = \sum_{\mathbf{x} \in \mathcal{D}_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^t$$

  - Total within-class scatter matrix

$$\mathbf{S}_W = \mathbf{S}_1 + \mathbf{S}_2$$

  - Deduce

$$\tilde{s}_i^2 = \sum_{\mathbf{x} \in \mathcal{D}_i} (\mathbf{w}^t\mathbf{x} - \mathbf{w}^t\mathbf{m}_i)^2$$

$$= \sum_{\mathbf{x} \in \mathcal{D}_i} \mathbf{w}^t(\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^t\mathbf{w}$$

$$= \mathbf{w}^t\mathbf{S}_i\mathbf{w};$$

$$\tilde{s}_1^2 + \tilde{s}_2^2 = \mathbf{w}^t\mathbf{S}_W\mathbf{W}$$

Fisher Linear Discriminant Analysis

- Represent $J(\mathbf{w})$ as the expressions for **w**
  - Total between-class scatter matrix

    $$\mathbf{S}_B = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^t$$

  - Deduce

    $$\begin{aligned} |\widetilde{m}_1 - \widetilde{m}_2|^2 &= |\mathbf{w}^t\mathbf{m}_1 - \mathbf{w}^t\mathbf{m}_2|^2 \\ &= \mathbf{w}^t(\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^t\mathbf{w} \\ &= \mathbf{w}^t\mathbf{S}_B\mathbf{w}, \end{aligned}$$

Fisher Linear Discriminant Analysis

- Fisher Criterion Function

$$J(\mathbf{w}) = \frac{|\tilde{m}_1 - \tilde{m}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

⇓

$$J(\mathbf{w}) = \frac{\mathbf{w}^t \mathbf{S}_B \mathbf{w}}{\mathbf{w}^t \mathbf{S}_W \mathbf{w}}$$

- The Fisher criterion function is maximized when $\mathbf{w}$ satisfies

$\mathbf{S}_\mathbf{w}$ is non-singular

$$\mathbf{S}_B \mathbf{w} = \lambda \mathbf{S}_W \mathbf{w} \quad \Rightarrow \quad \mathbf{S}_W^{-1} \mathbf{S}_B \mathbf{w} = \lambda \mathbf{w}$$

Generalized eigenvalue problems

Conventional eigenvalue problems

Fisher Linear Discriminant Analysis

- Extension of two categories to c categories - Multiple Discriminant Analysis
  - Total within-class scatter matrix

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in \mathcal{D}_i} \mathbf{x}$$

$$\mathbf{S}_i = \sum_{\mathbf{x} \in \mathcal{D}_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^t$$

$$\mathbf{S}_W = \sum_{i=1}^{c} \mathbf{S}_i$$

Fisher Linear Discriminant Analysis

- Extension of two categories to c categories - Multiple Discriminant Analysis
  - Total mean vector

$$\mathbf{m} = \frac{1}{n}\sum_{\mathbf{x}} \mathbf{x} = \frac{1}{n}\sum_{i=1}^{c} n_i \mathbf{m}_i$$

  - Total scatter matrix

$$\mathbf{S}_T = \sum_{\mathbf{x}} (\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^t$$

Fisher Linear Discriminant Analysis

- Extension of two categories to c categories - Multiple Discriminant Analysis
  - Derivation

$$\mathbf{S}_T = \sum_{i=1}^{c} \sum_{\mathbf{x}\in\mathcal{D}_i} (\mathbf{x}-\mathbf{m}_i+\mathbf{m}_i-\mathbf{m})(\mathbf{x}-\mathbf{m}_i+\mathbf{m}_i-\mathbf{m})^t$$

$$= \sum_{i=1}^{c} \sum_{\mathbf{x}\in\mathcal{D}_i} (\mathbf{x}-\mathbf{m}_i)(\mathbf{x}-\mathbf{m}_i)^t + \sum_{i=1}^{c} \sum_{\mathbf{x}\in\mathcal{D}_i} (\mathbf{m}_i-\mathbf{m})(\mathbf{m}_i-\mathbf{m})^t$$

$$\boxed{= \mathbf{S}_W + \sum_{i=1}^{c} n_i(\mathbf{m}_i-\mathbf{m})(\mathbf{m}_i-\mathbf{m})^t}$$

between-class scatter matrix

Fisher Linear Discriminant Analysis

- Extension of two categories to c categories - Multiple Discriminant Analysis
  - Between-class scatter matrix

$$\mathbf{S}_B = \sum_{i=1}^{c} n_i(\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^t$$

$$\mathbf{S}_T = \mathbf{S}_W + \mathbf{S}_B$$

  - Projection

Transformation matrix

$$\mathbf{y} = \mathbf{W}^t\mathbf{x}$$

Projection point

Original sample point

Fisher Linear Discriminant Analysis

- Extension of two categories to c categories - Multiple Discriminant Analysis
  - In the projection subspace made by $\mathbf{W}$

$$\tilde{\mathbf{m}}_i = \frac{1}{n_i} \sum_{\mathbf{y} \in \mathcal{Y}_i} \mathbf{y}$$

$$\tilde{\mathbf{m}} = \frac{1}{n} \sum_{i=1}^{c} n_i \tilde{\mathbf{m}}_i$$

$$\tilde{\mathbf{S}}_W = \sum_{i=1}^{c} \sum_{\mathbf{y} \in \mathcal{Y}_i} (\mathbf{y} - \tilde{\mathbf{m}}_i)(\mathbf{y} - \tilde{\mathbf{m}}_i)^t$$

$$\tilde{\mathbf{S}}_B = \sum_{i=1}^{c} n_i (\tilde{\mathbf{m}}_i - \tilde{\mathbf{m}})(\tilde{\mathbf{m}}_i - \tilde{\mathbf{m}})^t$$

Fisher Linear Discriminant Analysis

- Extension of two categories to c categories - Multiple Discriminant Analysis
  - Substituting $y = \mathbf{W}^t \mathrm{x}$ in, we get

$$\tilde{\mathbf{S}}_W = \mathbf{W}^t \mathbf{S}_W \mathbf{W}$$
$$\tilde{\mathbf{S}}_B = \mathbf{W}^t \mathbf{S}_B \mathbf{W}$$

  - **To seek the most efficient classification of W**： maximizing the ratio of the between-class scatter to the within-class scatter
    - Dispersion metrics ： the determinant of the scatter matrix

Fisher Linear Discriminant Analysis

- Extension of two categories to c categories - Multiple Discriminant Analysis
  - Criterion Function

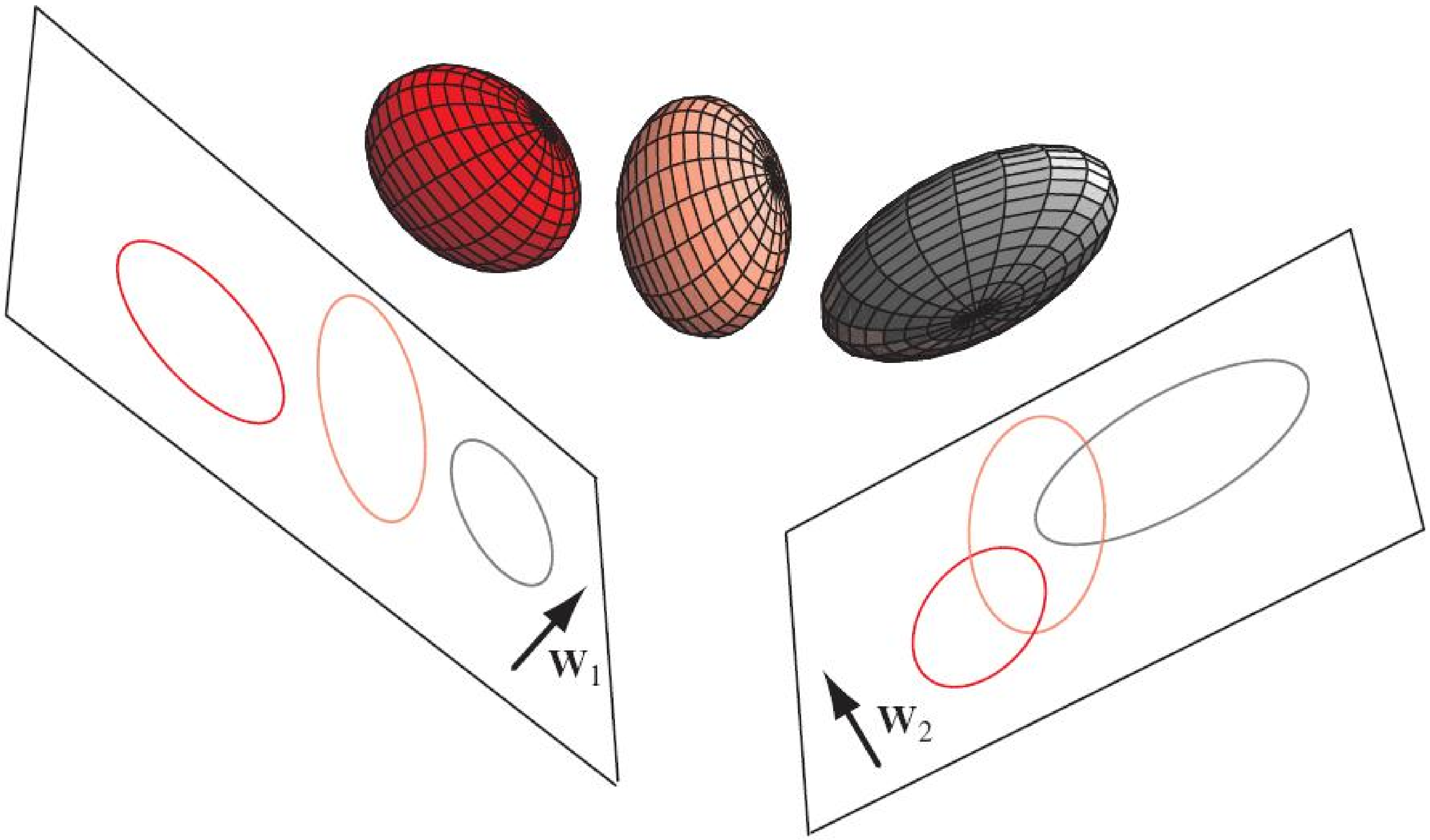
$$J(\mathbf{W}) = \frac{|\tilde{\mathbf{S}}_B|}{|\tilde{\mathbf{S}}_W|} = \frac{|\mathbf{W}^t\mathbf{S}_B\mathbf{W}|}{|\mathbf{W}^t\mathbf{S}_W\mathbf{W}|}$$

  - The column vector of **W** that maximizes *J*(**W**) consists of the eigenvector corresponding to the largest eigenvalue in the generalized eigenvalue problem as follows

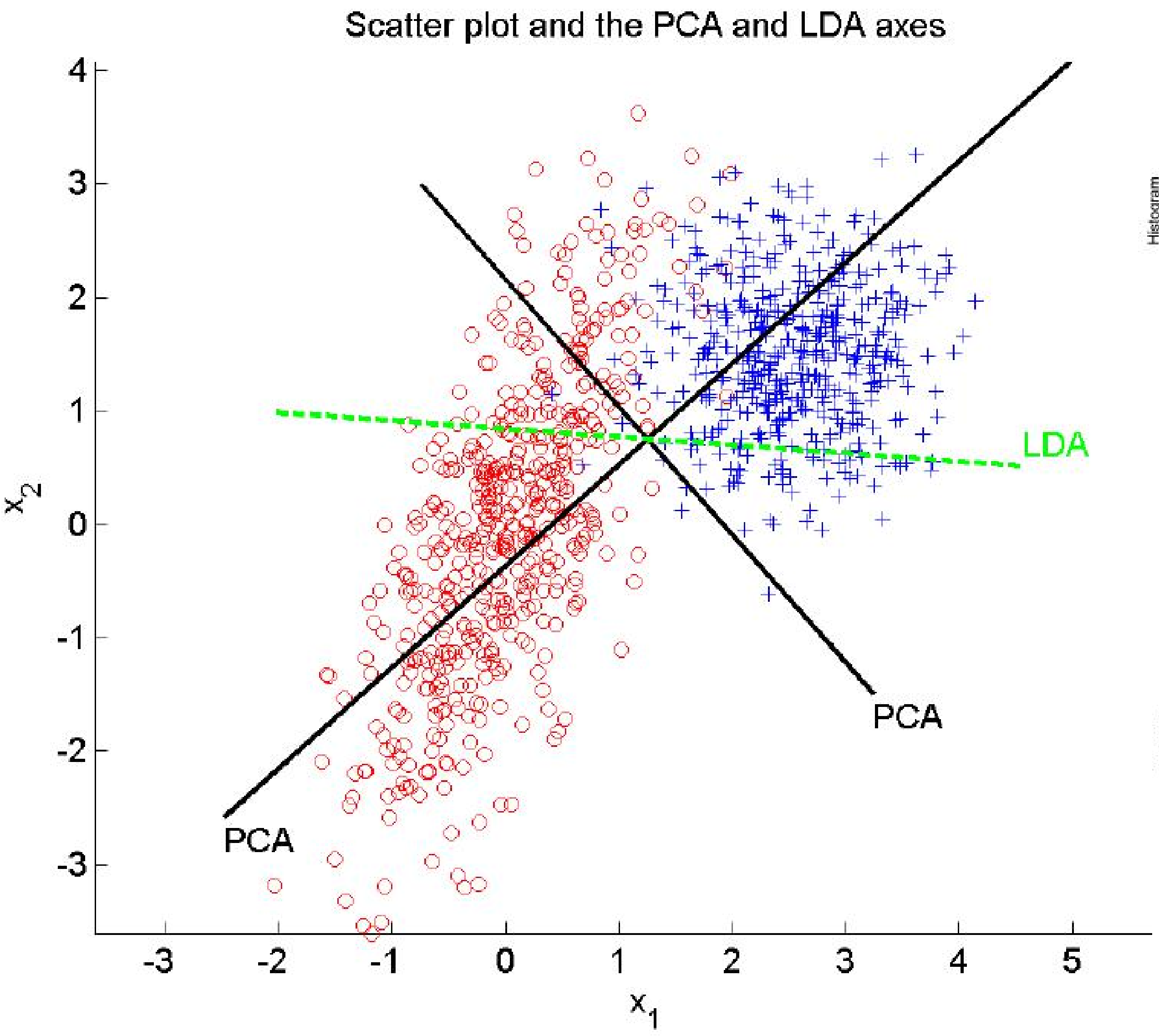
$$\mathbf{S}_B\mathbf{w}_i = \lambda_i\mathbf{S}_W\mathbf{w}_i$$

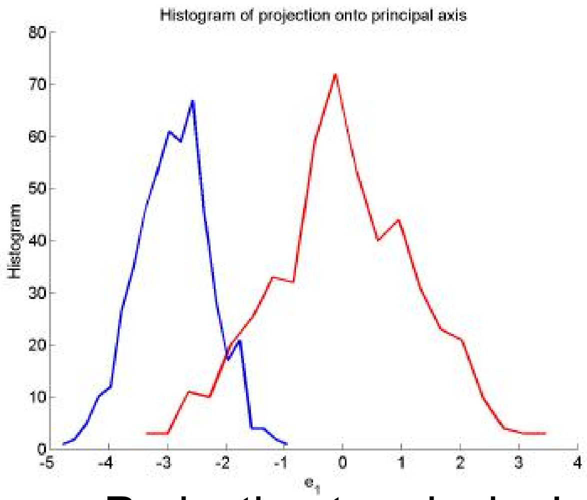
$\mathbf{S}_\mathbf{B}$ is the sum of c matrices of rank one or zero, and because only c−1 of these are independent, **$\mathbf{S}_\mathbf{B}$** is of rank c−1 or less
Thus, no more than c−1 of the eigenvalues are nonzero, which correspond to c−1 eigenvectors, and the matrix **W** has at most c-1 columns

Fisher Linear Discriminant Analysis

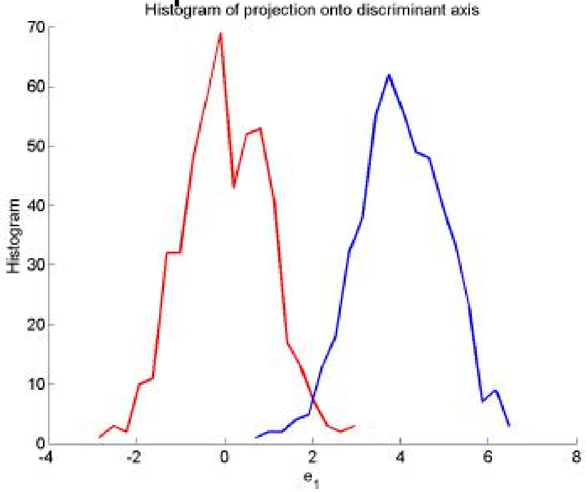


Fisher Linear Discriminant Analysis



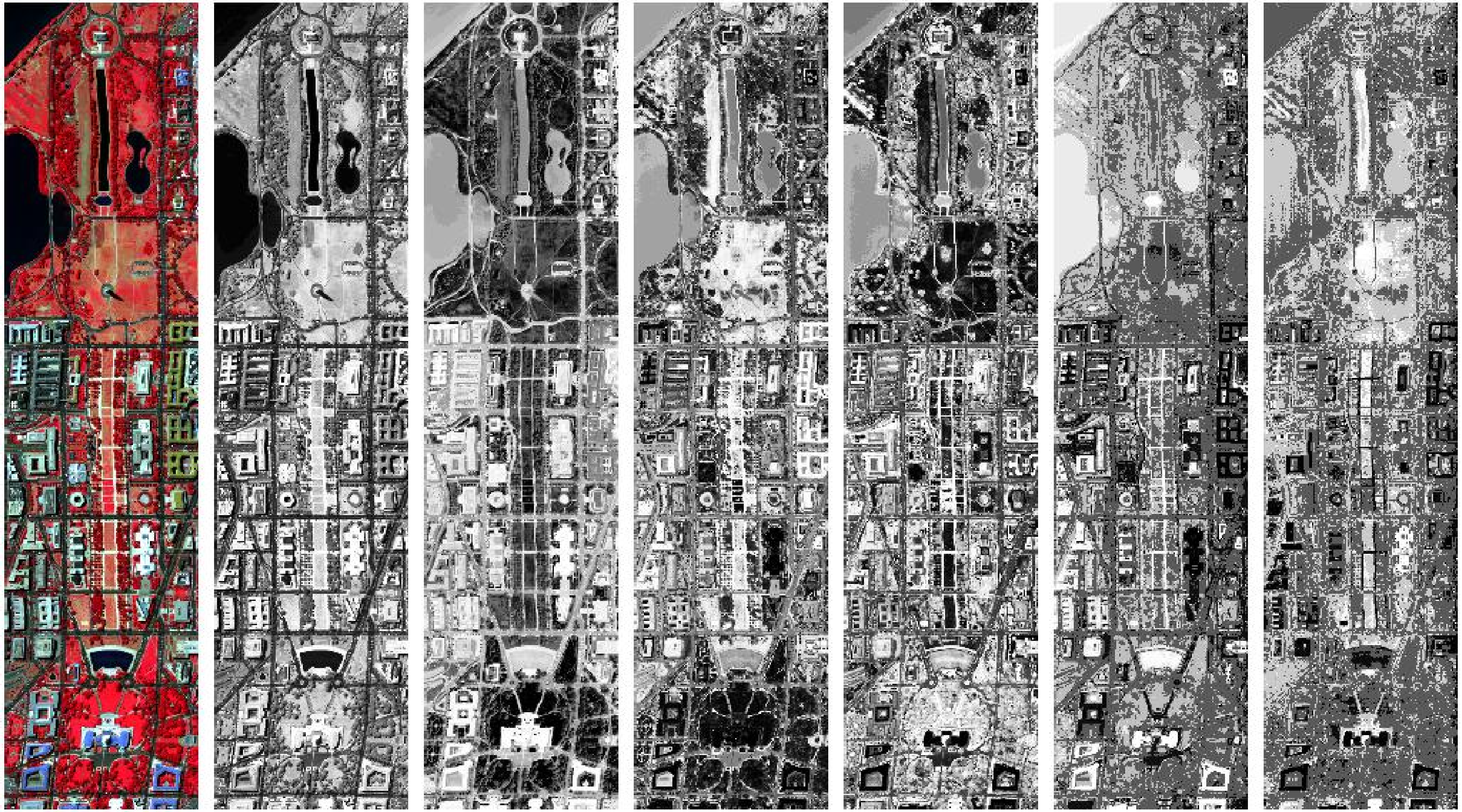


Projection to principal component orientation



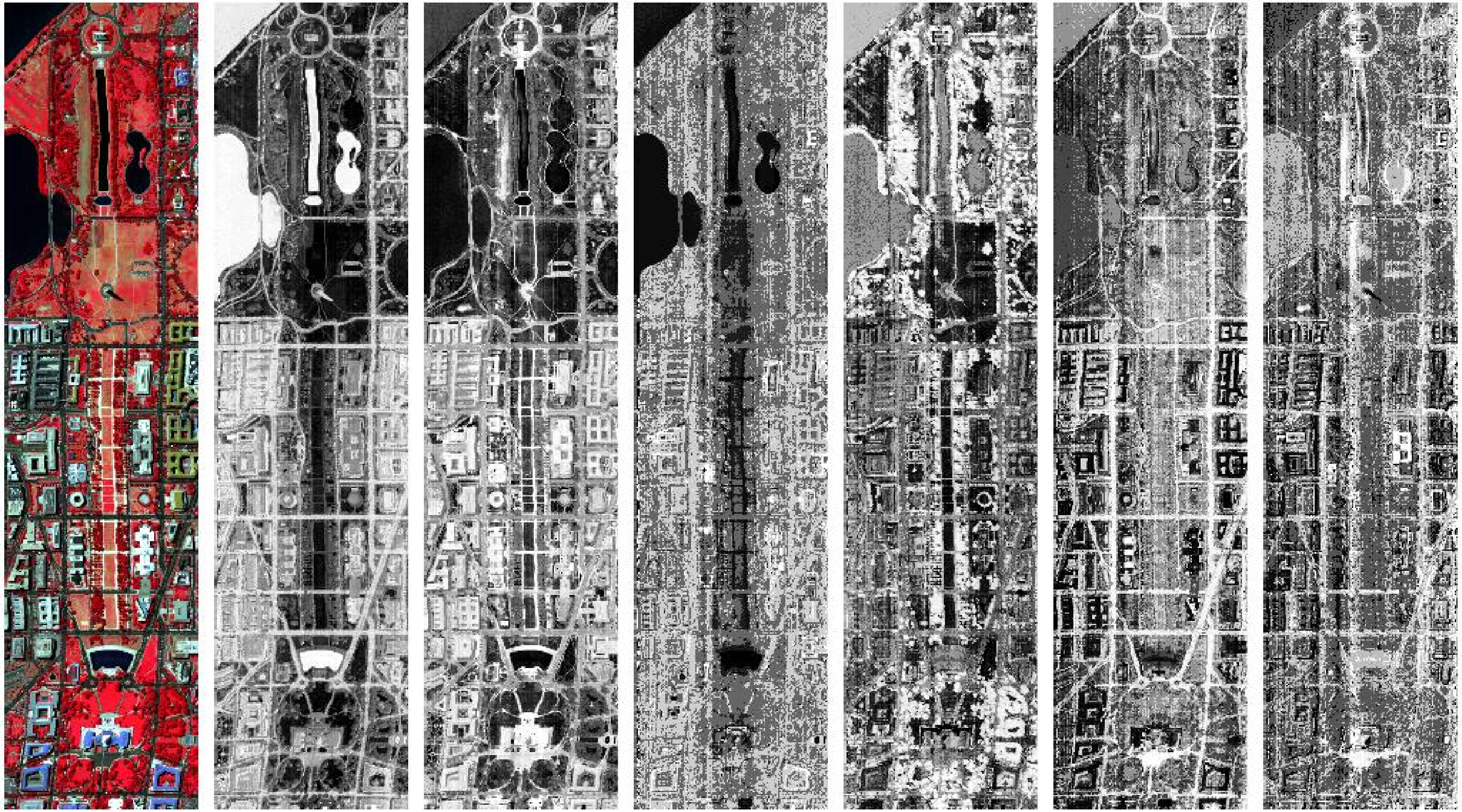
Projection to LDA orientation

Dimensionality Reduction Example: Satellite Image Analysis



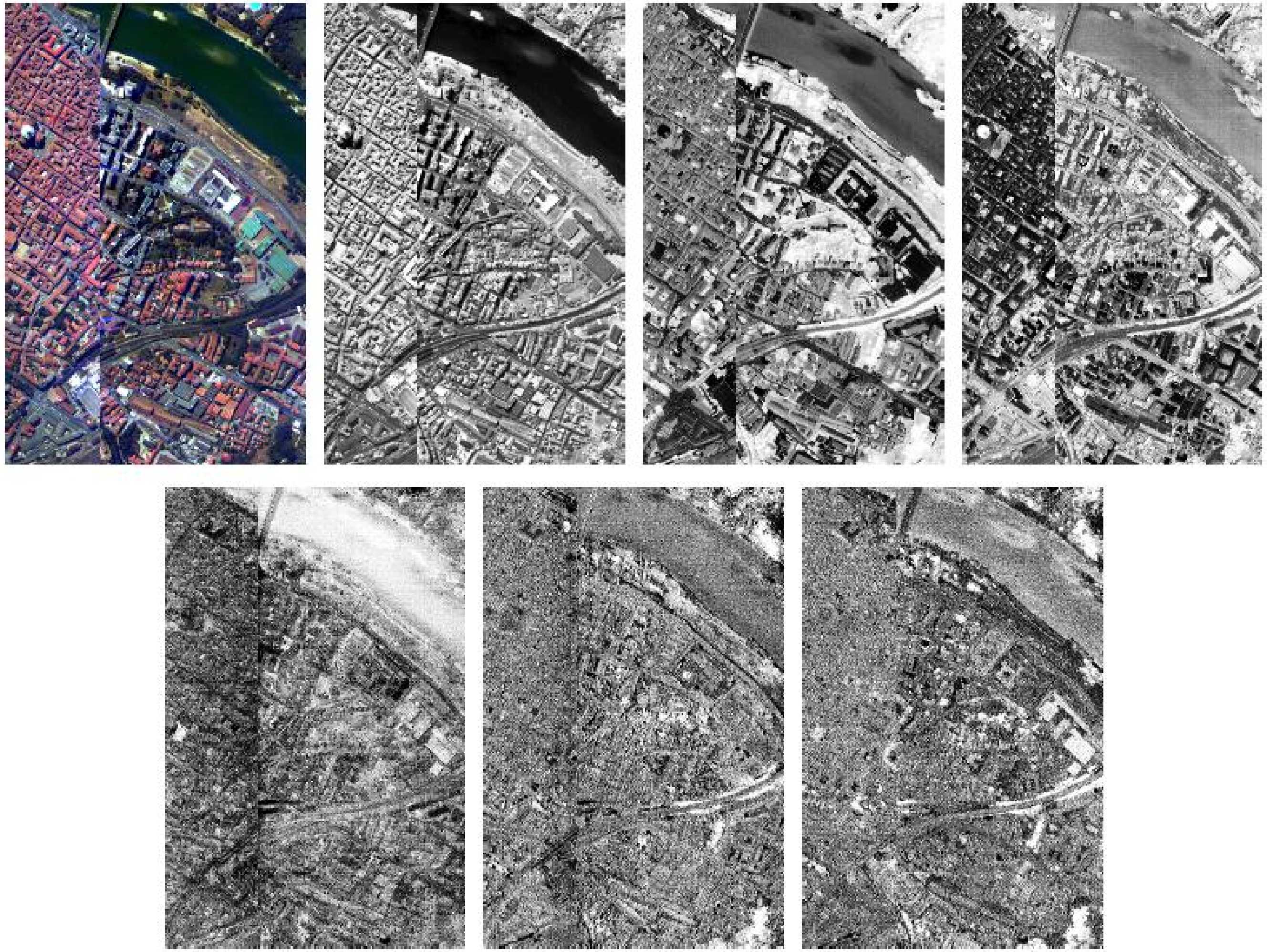
The original satellite image and the first six PCA principal component projection directions

Dimensionality Reduction Example: Satellite Image Analysis



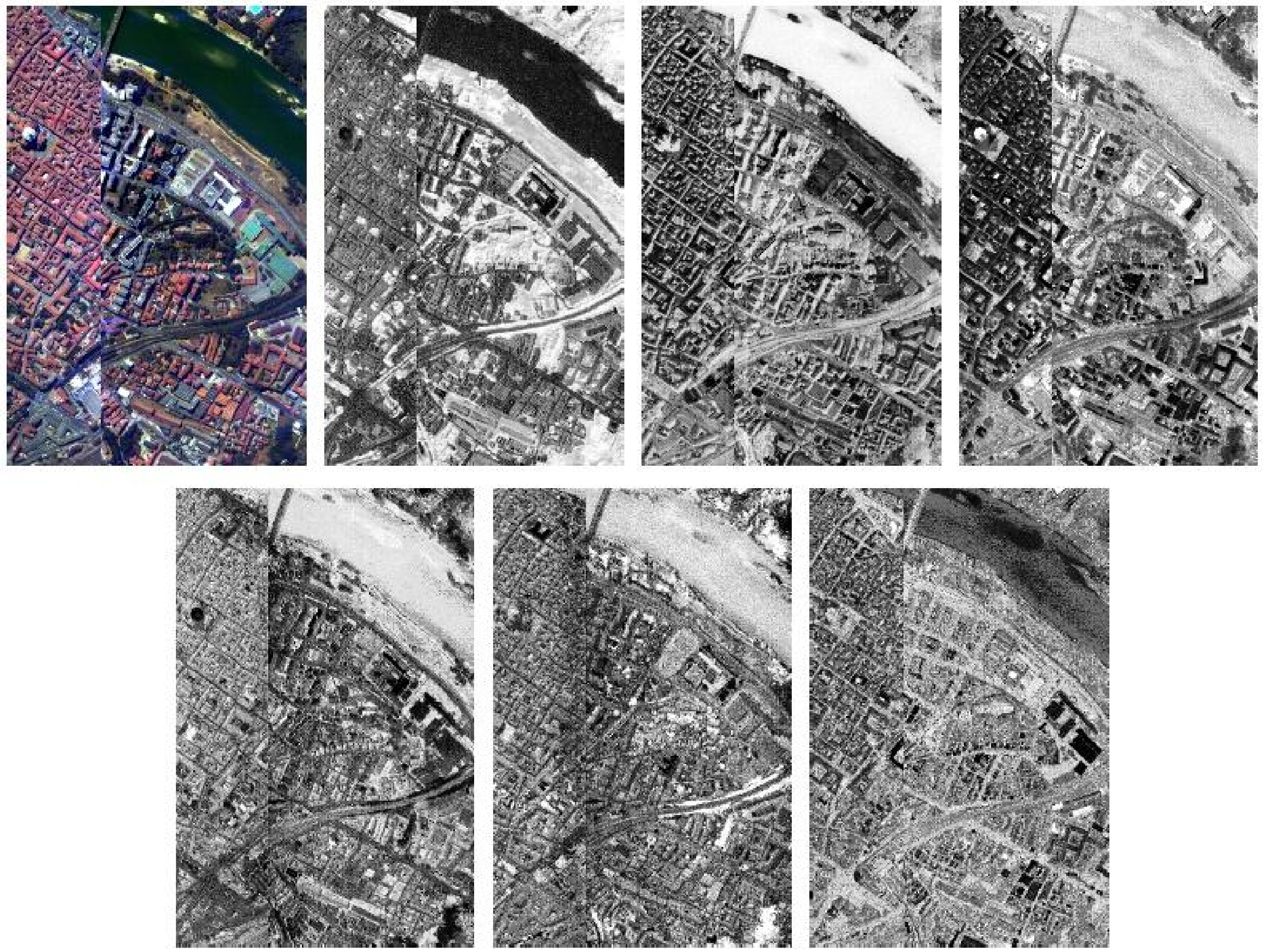
The original satellite image and the first six LDA projection directions

Dimensionality Reduction Example: Satellite Image Analysis



The original satellite image and the first six PCA principal component projection directions

Dimensionality Reduction Example: Satellite Image Analysis

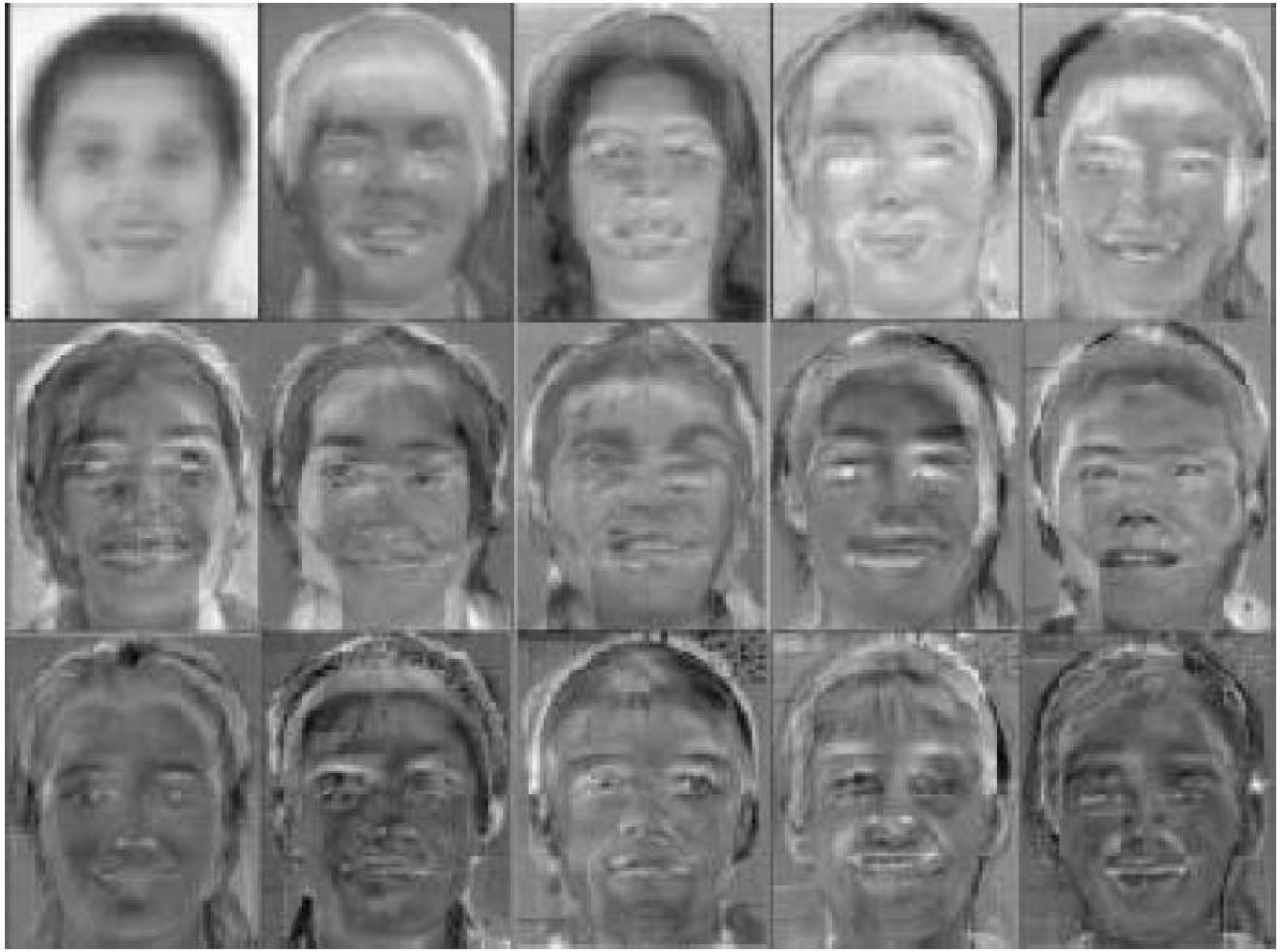


The original satellite image and the first six LDA projection directions

Dimension reduction example: face recognition

Typical face image collection

Dimension reduction example: face recognition



The first 15 PCA principal component projection directions of the face image, also known as “Eigenface“(本征脸)

Summary

- Feature combination to reduce dimensionality
  - Principal component analysis (PCA)
    - Look for projections to **represent data effectively**
    - Unsupervised
  - Linear discriminant analysis (LDA)
    - Look for projections to **classify data effectively**
    - Supervised

Ch 06. Feature Reduction and Selection

Part 2 Feature Selection

Dimensionality Reduction

- The way to Dimensionality Reduction
  - **Feature combination**

    Combine several features to form a new feature
  - **Feature selection**

    Select a subset of the existing feature set

Feature Selection

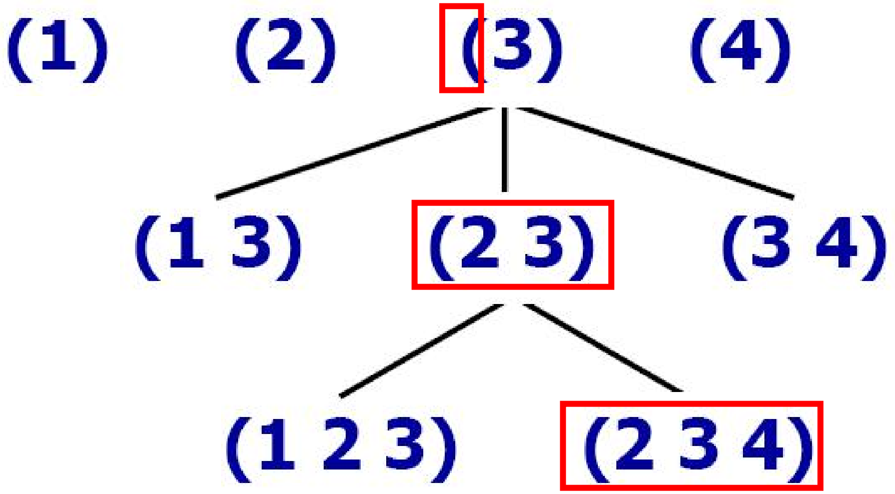
- The feature selection method consists of two main components
  - **Search process**
  - **Selection criteria**
- **Search process**
  - The process of searching the system in all candidate feature subsets
  - In principle, exhaustive search （穷尽搜索） can find optimal child sets, In practice，a more efficient non-exhaustive search algorithm is often used to find the suboptimal solution
- **Selection criteria**
  - Criteria used to determine whether a subset of features is superior to another subset of features
  - In principle, the selection criterion is the evaluation criterion of system performance, such as classification error rate, etc. In practice, simplified selection criteria are often used

Search Process

- **Sequential Forward Selection**
  （**循序向前选择法，SFS**）
  - First, the optimal individual features are selected
  - Then, combine all the other features with the first selected feature to form a candidate feature pair to find the optimal pair
  - The remaining features are then paired with the best features selected in the previous step to form candidate feature triples and find the optimal triples
  - The process stops until enough features are selected

Search Process

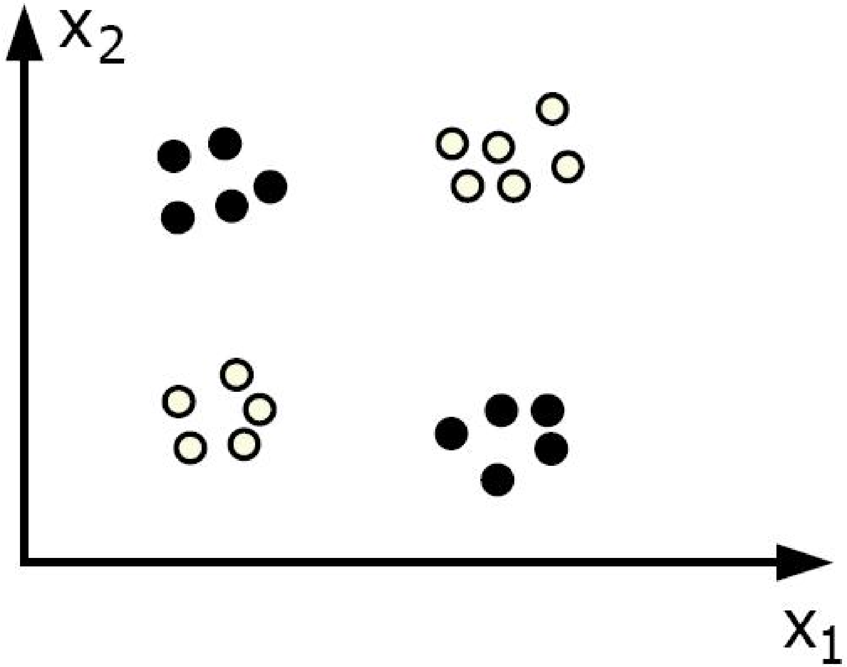
- **Sequential Forward Selection**
  （循序向前选择法，**SFS**）



Search Process

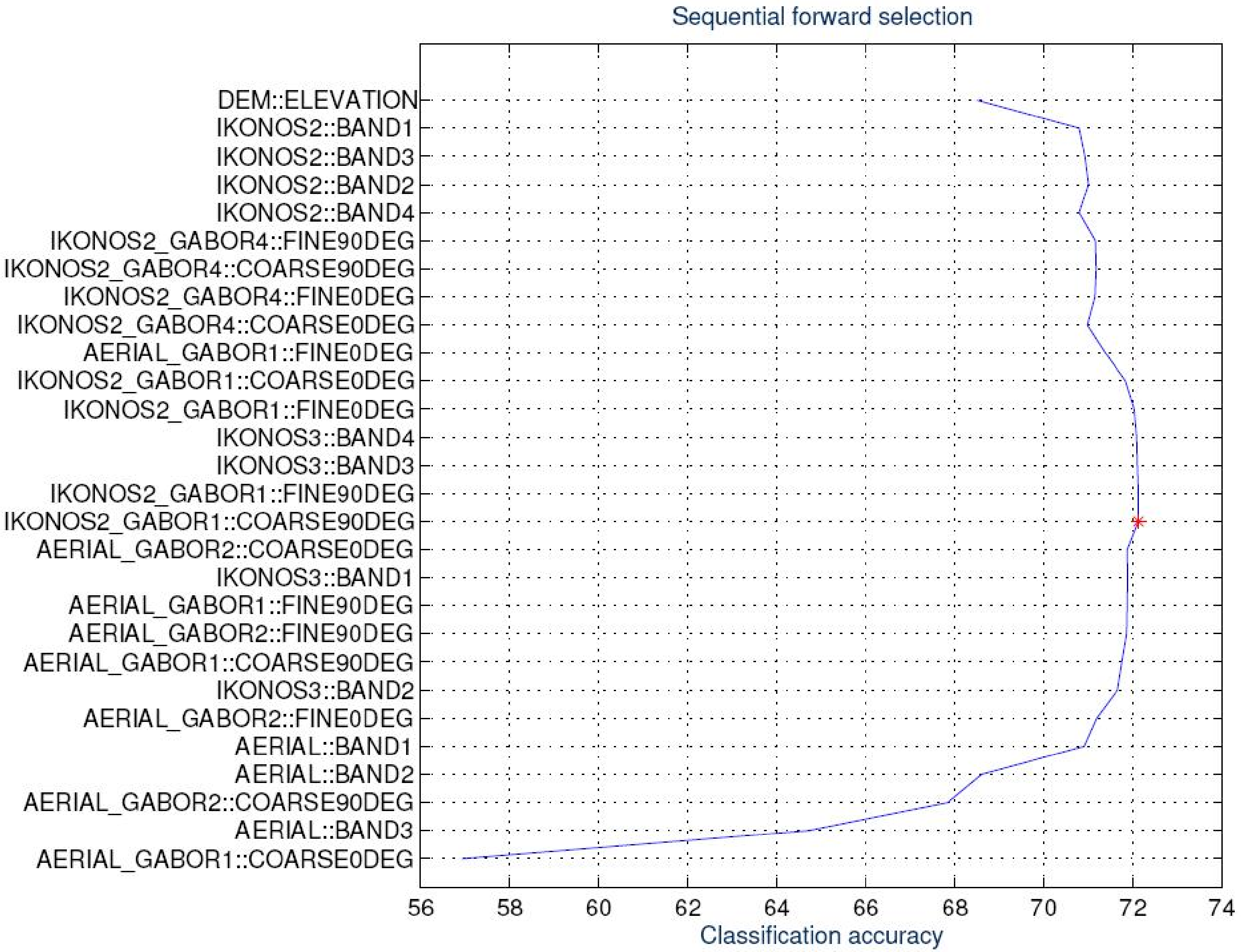
- **Sequential Forward Selection**
  （循序向前选择法，SFS）
  - Disadvantage
    - The distinguishing force of single feature is poor, but the combined distinguishing force of two features is strong. In this case, SFS fails

When each feature of the optimal subset is considered separately, it is not always optimal



Search Process

- **SFS：example——**Satellite image analysis

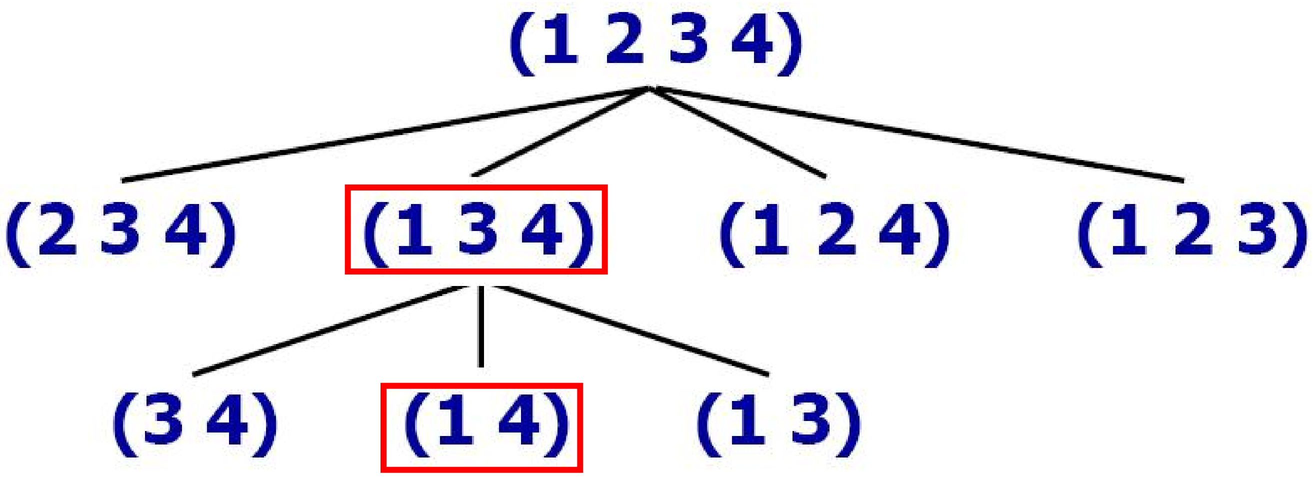


Search Process

- **Sequential Backward Selection （循序向后选择法，SBS）**
  - First, select all d features
  - Then, an arbitrary one of the features is removed to form the d-1 feature set with d candidates, and the best one is selected
  - Then remove an arbitrary feature from the D-1 feature set obtained in the previous step to form d-1 d-2 feature sets, and select the best one
  - The process stops when the number of features in the feature set reaches a predetermined value

Search Process

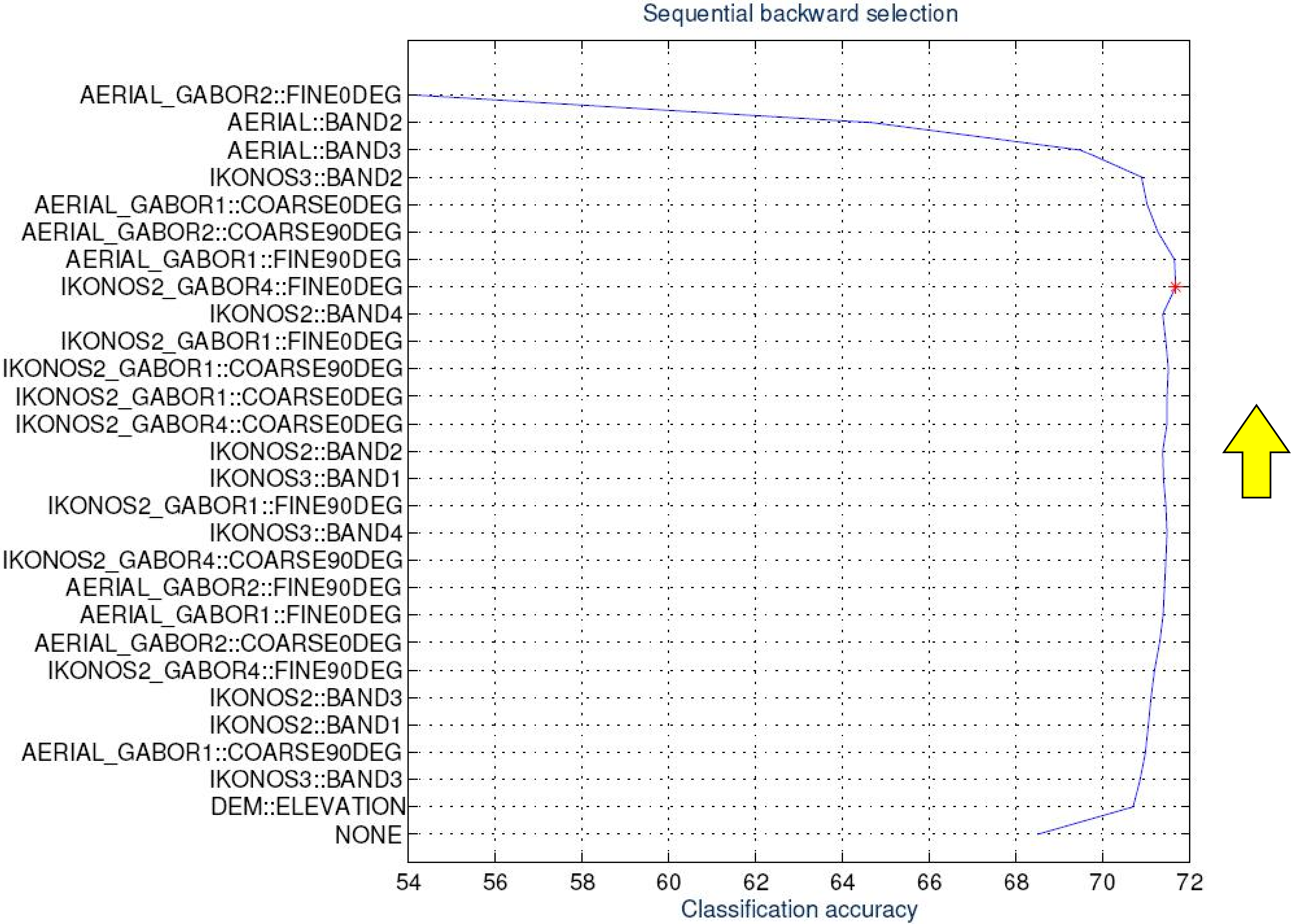
- **Sequential Backward Selection**
  （循序向后选择法，**SBS**）



Because the number of features considered by SBS is greater than or equal to the expected number of features, SBS usually requires more selection criteria calculation than SFS

Search Process

- **SBS：example**——Satellite image analysis



Search Process

- Other search processes
  - Single optimal subset of features
    - Search directly for the optimal individual features (one feature at a time, compute the selection criteria) and use the set of them as the result of the feature selection
    - Simple, but often unreliable
    - The optimal feature subset can be found only if the features are completely independent of each other
  - …

Selection Criteria

- Ideal method
  - Represent the training sample with the selected feature subset, train the classifier, and then test the generalization error of the classifier (e.g., cross-validation).
  - for each feature subset, we need to train a classifier, so the computation is very large
- Simplified method
  - Define a within-class distance metric to describe class separability when a subset of features is adopted
  - There is no need to train a classifier for each feature subset, so the computation is small

Selection Criteria

- **Within-class distance**
  - **within-class scatter**

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in \mathcal{D}_i} \mathbf{x}$$

$$\mathbf{S}_i = \sum_{\mathbf{x} \in \mathcal{D}_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^t$$

$$\mathbf{S}_W = \sum_{i=1}^{c} \mathbf{S}_i$$

Selection Criteria

- **Within-class distance**
  - **mean square distance**

$$D_i = \frac{2}{n_i(n_i - 1)} \sum_{\substack{a,b \in \omega_i \\ a \neq b}} \|a - b\|^2$$

$$D_W = \sum_{i=1}^{c} D_i$$