（请大家预习）

第7章第2讲
特征变换与特征选择
Feature Extraction and Feature Selection

张 燕 明
ymzhang@nlpr.ia.ac.cn
people.ucas.ac.cn/~ymzhang

模式分析与学习课题组(PAL)
多模态人工智能系统实验室 中科院自动化所

助教： 杨 奇 (yangqi2021@ia.ac.cn)
张 涛 (zhangtao2021@ia.ac.cn)

内容回顾

End-to-end Deep learning

- AE, RBM, CNN, RNN,
- GNN,
- Transformer

Raw Data → Feature Space → Classification

1. Domain specific feature extraction: MFCC, BOW, TF-IDF, word2vec, SIFT, LBP
2. Feature transformation / feature selection: PCA, LDA, Isomap

SVM, MLP, Boosting

Traditional Pattern Recognition

内容回顾

- **特征提取的目的**
  - 提取观测数据的内在特性
  - 减少噪声影响
  - 提高稳定性

- **特征变换的目的**
  - 降低特征空间的维度，增加数据密度、降低过拟合风险
  - 便于分析和减少后续步骤的计算量
  - 有利于分类
  - 线性特征变换：PCA, LDA

7.7 多维缩放

- **多维缩放(Multiple Dimensional Scaling, MDS)**
  - 假设 n 个样本$\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}\subset R^m$在原始空间的距离矩阵为$\mathbf{D}\in R^{n\times n}$，其第 i 行 j 列的元素为样本 $\mathbf{x}_i$ 到 $\mathbf{x}_j$ 的距离，记为 $\mathbf{D}_{ij}$
  - 目标：获得这 n 个样本在 d ($d<m$) 维空间中的向量表示 $\mathbf{Z}=[\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n]\in R^{d\times n}$，使得降维后的样本仍保持两两之间的距离：

$$\mathbf{D}_{ij} \approx \left\| \mathbf{z}_i - \mathbf{z}_j \right\|_2^2, \quad i, j = 1, 2 \cdots, n$$

7.7 多维缩放

- **MDS**
  - 基于低维空间的样本表示$\mathbf{Z}\in R^{d\times n}$，计算样本间距离$\tilde{\mathbf{D}}\in R^{n\times n}$：

$$\tilde{\mathbf{D}}=\begin{pmatrix} d_{11}^2 & d_{12}^2 & \cdots & d_{1n}^2 \\ d_{21}^2 & d_{22}^2 & \cdots & d_{2n}^2 \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1}^2 & d_{n2}^2 & \cdots & d_{nn}^2 \end{pmatrix}=\begin{pmatrix} \|\mathbf{z}_1-\mathbf{z}_1\|_2^2 & \|\mathbf{z}_1-\mathbf{z}_2\|_2^2 & \cdots & \|\mathbf{z}_1-\mathbf{z}_n\|_2^2 \\ \|\mathbf{z}_2-\mathbf{z}_1\|_2^2 & \|\mathbf{z}_2-\mathbf{z}_2\|_2^2 & \cdots & \|\mathbf{z}_2-\mathbf{z}_n\|_2^2 \\ \vdots & \vdots & \ddots & \vdots \\ \|\mathbf{z}_n-\mathbf{z}_1\|_2^2 & \|\mathbf{z}_n-\mathbf{z}_2\|_2^2 & \cdots & \|\mathbf{z}_n-\mathbf{z}_n\|_2^2 \end{pmatrix}$$

$$=\begin{pmatrix} 0 & \|\mathbf{z}_1\|_2^2+\|\mathbf{z}_2\|_2^2-2\mathbf{z}_1^T\mathbf{z}_2 & \cdots & \|\mathbf{z}_1\|_2^2+\|\mathbf{z}_n\|_2^2-2\mathbf{z}_1^T\mathbf{z}_n \\ \|\mathbf{z}_1\|_2^2+\|\mathbf{z}_2\|_2^2-2\mathbf{z}_1^T\mathbf{z}_2 & 0 & \cdots & \|\mathbf{z}_2\|_2^2+\|\mathbf{z}_n\|_2^2-2\mathbf{z}_2^T\mathbf{z}_n \\ \vdots & \vdots & \ddots & \vdots \\ \|\mathbf{z}_1\|_2^2+\|\mathbf{z}_n\|_2^2-2\mathbf{z}_1^T\mathbf{z}_n & \|\mathbf{z}_2\|_2^2+\|\mathbf{z}_n\|_2^2-2\mathbf{z}_2^T\mathbf{z}_n & \cdots & 0 \end{pmatrix}$$

- **MDS（接上页）**
  - 基于低维空间的样本表示$\mathbf{Z} \in R^{d\times n}$，计算样本间距离$\tilde{\mathbf{D}} \in R^{n\times n}$：

$$\tilde{\mathbf{D}} = \begin{pmatrix} \|\mathbf{z}_1\|_2^2 & \|\mathbf{z}_1\|_2^2 & \cdots & \|\mathbf{z}_1\|_2^2 \\ \|\mathbf{z}_2\|_2^2 & \|\mathbf{z}_2\|_2^2 & \cdots & \|\mathbf{z}_2\|_2^2 \\ \vdots & \vdots & \ddots & \vdots \\ \|\mathbf{z}_n\|_2^2 & \|\mathbf{z}_n\|_2^2 & \cdots & \|\mathbf{z}_n\|_2^2 \end{pmatrix} + \begin{pmatrix} \|\mathbf{z}_1\|_2^2 & \|\mathbf{z}_2\|_2^2 & \cdots & \|\mathbf{z}_n\|_2^2 \\ \|\mathbf{z}_1\|_2^2 & \|\mathbf{z}_2\|_2^2 & \cdots & \|\mathbf{z}_n\|_2^2 \\ \vdots & \vdots & \ddots & \vdots \\ \|\mathbf{z}_1\|_2^2 & \|\mathbf{z}_2\|_2^2 & \cdots & \|\mathbf{z}_n\|_2^2 \end{pmatrix}$$

$$-2\begin{pmatrix} \mathbf{z}_1^T\mathbf{z}_1 & \mathbf{z}_1^T\mathbf{z}_2 & \cdots & \mathbf{z}_1^T\mathbf{z}_n \\ \mathbf{z}_2^T\mathbf{z}_1 & \mathbf{z}_2^T\mathbf{z}_2 & \cdots & \mathbf{z}_2^T\mathbf{z}_n \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{z}_n^T\mathbf{z}_1 & \mathbf{z}_n^T\mathbf{z}_2 & \cdots & \mathbf{z}_n^T\mathbf{z}_n \end{pmatrix}$$

- **MDS**
  - 令矩阵B为样本点在新表示下的内积：

$$\mathbf{B} = \mathbf{Z}^T \mathbf{Z} \in R^{n \times n}$$

  - 令向量c为样本点在新表示下模长的平方：

$$\mathbf{c} = \left[\|\mathbf{z}_1\|_2^2, \|\mathbf{z}_2\|_2^2, \ldots, \|\mathbf{z}_n\|_2^2 \right]^T \in R^n$$

  - 我们有：

$$\tilde{\mathbf{D}} = \mathbf{c}\mathbf{e}^T + \mathbf{e}\mathbf{c}^T - 2\mathbf{B} \qquad \mathbf{e}=[1,1,\ldots,1]^T \in R^n$$

$$\tilde{\mathbf{D}} = \begin{pmatrix} \|\mathbf{z}_1\|_2^2 & \|\mathbf{z}_1\|_2^2 & \cdots & \|\mathbf{z}_1\|_2^2 \\ \|\mathbf{z}_2\|_2^2 & \|\mathbf{z}_2\|_2^2 & \cdots & \|\mathbf{z}_2\|_2^2 \\ \vdots & \vdots & \ddots & \vdots \\ \|\mathbf{z}_n\|_2^2 & \|\mathbf{z}_n\|_2^2 & \cdots & \|\mathbf{z}_n\|_2^2 \end{pmatrix} + \begin{pmatrix} \|\mathbf{z}_1\|_2^2 & \|\mathbf{z}_2\|_2^2 & \cdots & \|\mathbf{z}_n\|_2^2 \\ \|\mathbf{z}_1\|_2^2 & \|\mathbf{z}_2\|_2^2 & \cdots & \|\mathbf{z}_n\|_2^2 \\ \vdots & \vdots & \ddots & \vdots \\ \|\mathbf{z}_1\|_2^2 & \|\mathbf{z}_2\|_2^2 & \cdots & \|\mathbf{z}_n\|_2^2 \end{pmatrix} - 2 \begin{pmatrix} \mathbf{z}_1^T\mathbf{z}_1 & \mathbf{z}_1^T\mathbf{z}_2 & \cdots & \mathbf{z}_1^T\mathbf{z}_n \\ \mathbf{z}_2^T\mathbf{z}_1 & \mathbf{z}_2^T\mathbf{z}_2 & \cdots & \mathbf{z}_2^T\mathbf{z}_n \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{z}_n^T\mathbf{z}_1 & \mathbf{z}_n^T\mathbf{z}_2 & \cdots & \mathbf{z}_n^T\mathbf{z}_n \end{pmatrix}$$

7.7 多维缩放

- **MDS**
  - 不失一般性，令降维后的样本是零均值化的，即：

$$\sum_{i=1}^{n} \mathbf{z}_i = \mathbf{0} \in R^d$$

  - 引入数据零均值化矩阵：$\mathbf{H}=\mathbf{I}-\frac{1}{n}\mathbf{e}\mathbf{e}^T \in R^{n\times n}$（$\mathbf{I}$为单位矩阵）

$\mathbf{e}=[1,1,\ldots,1]^T \in R^n$

我们有：$\mathbf{ZH}=\mathbf{Z}(\mathbf{I}-\frac{1}{n}\mathbf{e}\mathbf{e}^T)=\mathbf{Z}-\frac{1}{n}\mathbf{Z}\mathbf{e}\mathbf{e}^T=\mathbf{Z} \quad \in R^{d\times n}$

$$\mathbf{H}^T\mathbf{B}\mathbf{H}=\mathbf{H}^T\mathbf{Z}^T\mathbf{Z}\mathbf{H}=\mathbf{Z}^T\mathbf{Z}=\mathbf{B}$$

- **MDS**
  - 进一步，我们有：

$$\mathbf{H}=\mathbf{I}-\frac{1}{n}\mathbf{e}\mathbf{e}^T\in R^{n\times n}$$

$$\tilde{\mathbf{D}}=\mathbf{c}\mathbf{e}^T+\mathbf{e}\mathbf{c}^T-2\mathbf{B}$$

$$\begin{aligned}
\mathbf{H}^T\tilde{\mathbf{D}}\mathbf{H} &= \mathbf{H}^T(\mathbf{c}\mathbf{e}^T+\mathbf{e}\mathbf{c}^T-2\mathbf{B})\mathbf{H}\\
&= \mathbf{H}^T\mathbf{c}\mathbf{e}^T(\mathbf{I}-\frac{1}{n}\mathbf{e}\mathbf{e}^T)+(\mathbf{I}-\frac{1}{n}\mathbf{e}\mathbf{e}^T)\mathbf{e}\mathbf{c}^T\mathbf{H}-2\mathbf{H}^T\mathbf{B}\mathbf{H}\\
&= \mathbf{H}^T\mathbf{c}\mathbf{e}^T-\frac{1}{n}\mathbf{H}^T\mathbf{c}\mathbf{e}^T\mathbf{e}\mathbf{e}^T+\mathbf{e}\mathbf{c}^T\mathbf{H}-\frac{1}{n}\mathbf{e}\mathbf{e}^T\mathbf{e}\mathbf{c}^T\mathbf{H}-2\mathbf{H}^T\mathbf{B}\mathbf{H}\\
\left(\because\ \mathbf{e}^T\mathbf{e}=n\right)\quad &= \mathbf{H}^T\mathbf{c}\mathbf{e}^T-\mathbf{H}^T\mathbf{c}\mathbf{e}^T+\mathbf{e}\mathbf{c}^T\mathbf{H}-\mathbf{e}\mathbf{c}^T\mathbf{H}-2\mathbf{H}^T\mathbf{B}\mathbf{H}\\
&= -2\mathbf{H}^T\mathbf{B}\mathbf{H}\\
&= -2\mathbf{B}
\end{aligned}$$

7.7 多维缩放

- **MDS**
  - 于是有： $\mathbf{B}=-\frac{1}{2}\mathbf{H}^T\tilde{\mathbf{D}}\mathbf{H}$
  - 基于MDS的目标，可令 $\tilde{\mathbf{D}}=\mathbf{D}$ ，从而计算**B**
  - 获得矩阵**B**之后，因为**B**= $\mathbf{Z}^T\mathbf{Z}\in R^{n\times n}$，且**B**是对称矩阵，对其特征值分解有：$\mathbf{B}=\mathbf{U}\Lambda\mathbf{U}^T$，

$$\mathbf{Z}=\mathbf{\Lambda}_d^{1/2}\mathbf{U}_d^T\in R^{d\times n}$$

$$\mathbf{\Lambda}_d^{1/2}=diag(\sqrt{\lambda_1},\sqrt{\lambda_2},\ldots,\sqrt{\lambda_d})\in R^{d\times d},\ \ \mathbf{U}_d=[\mathbf{u}_1,\mathbf{u}_2,\ldots,\mathbf{u}_d]\in R^{n\times d}$$

其中，$\mathbf{\Lambda}_d^{1/2}$表示由矩阵**B**的前d个最大的特征值开根号后对应的对角矩阵；$\mathbf{U}_d$由前d个最大的特征值对应的特征向量组成。

7.7 多维缩放

MDS算法步骤：

1 计算数据的距离矩阵 $\mathbf{D} \in R^{n \times n}$

2 计算内积矩阵 $\mathbf{B} = -\frac{1}{2}\mathbf{H}^T\mathbf{D}\mathbf{H}$；

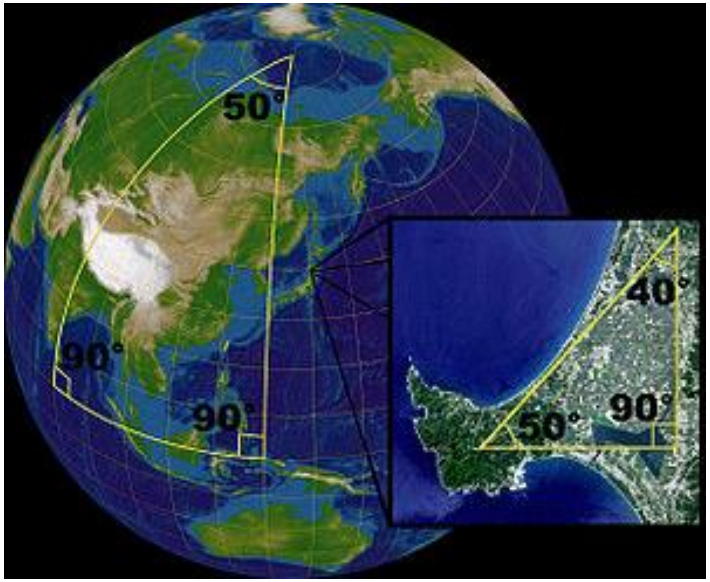
3 对内积矩阵B进行特征值分解：$\mathbf{B} = \mathbf{U}\Lambda\mathbf{U}^T$ ；

4 $\mathbf{Z} = \mathbf{\Lambda}_d^{1/2}\mathbf{U}_d^T \in R^{d \times n}$ ；

输出：$\mathbf{Z}$

7.8 流形学习

- **What are manifolds**



✓ 定义：流形上的每一个点的开邻域，与欧氏空间的开集同胚。

✓ 几何：流形是一块一块欧氏空间拼装而成的弯曲空间。

✓ 直观：流形是高维空间中的几何对象，其局部特性与欧式空间相似，但在全局范围可能具有复杂的拓扑结构。

The surface of a sphere is a two-dimensional manifold as it can be represented by a collection of two-dimensional maps.

(image from: http://en.wikipedia.org/wiki/Manifold)

7.8 流形学习

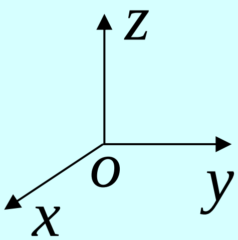
- **Mathematical definition**
  - **A manifold is a mathematical space that on a small enough scale resembles the Euclidean space of a specific dimension.**
  - **Local region can be coordinatized!**

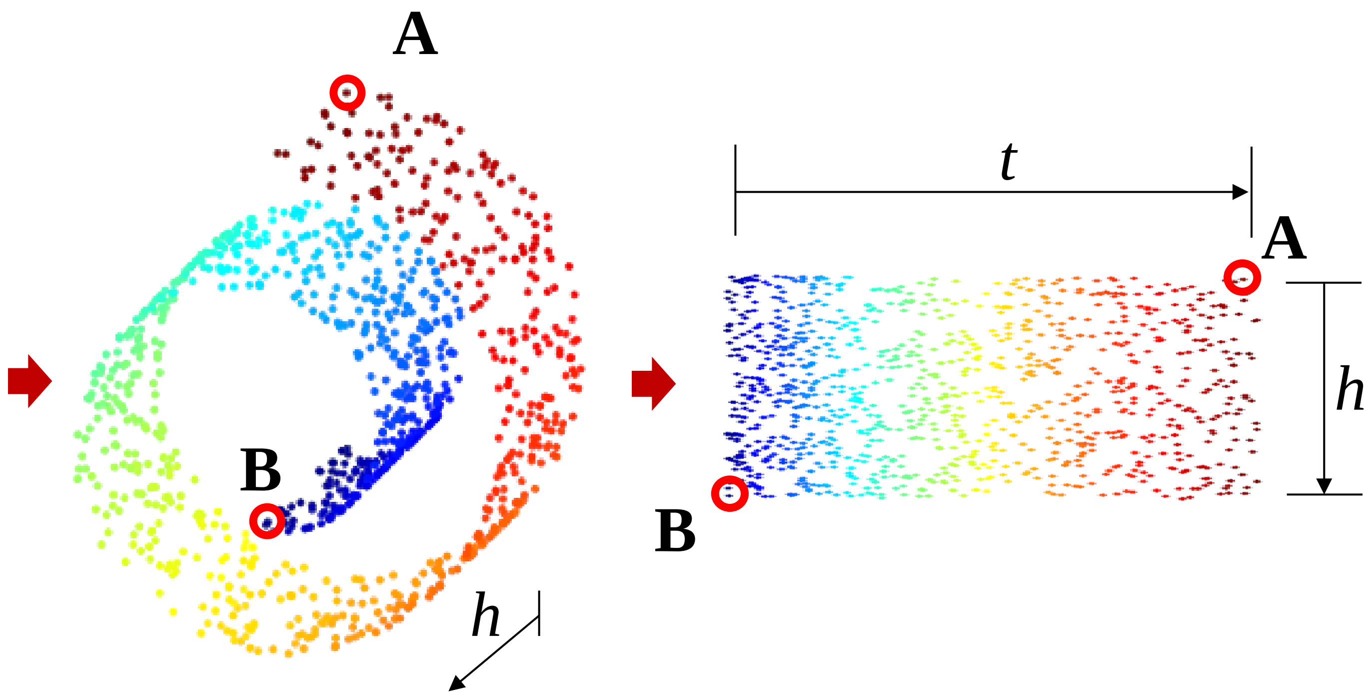
在数学上，流形用于描述一个几何形体，它在局部具有欧氏空间的性质。即可以应用欧氏距离来描述局部区域，但在全局欧氏距离不成立。

7.8 流形学习

- **What are manifolds**
  - **Swiss roll surface**

$$\begin{cases} x = h \\ y = t\cos(t) \\ z = t\sin(t) \end{cases}$$

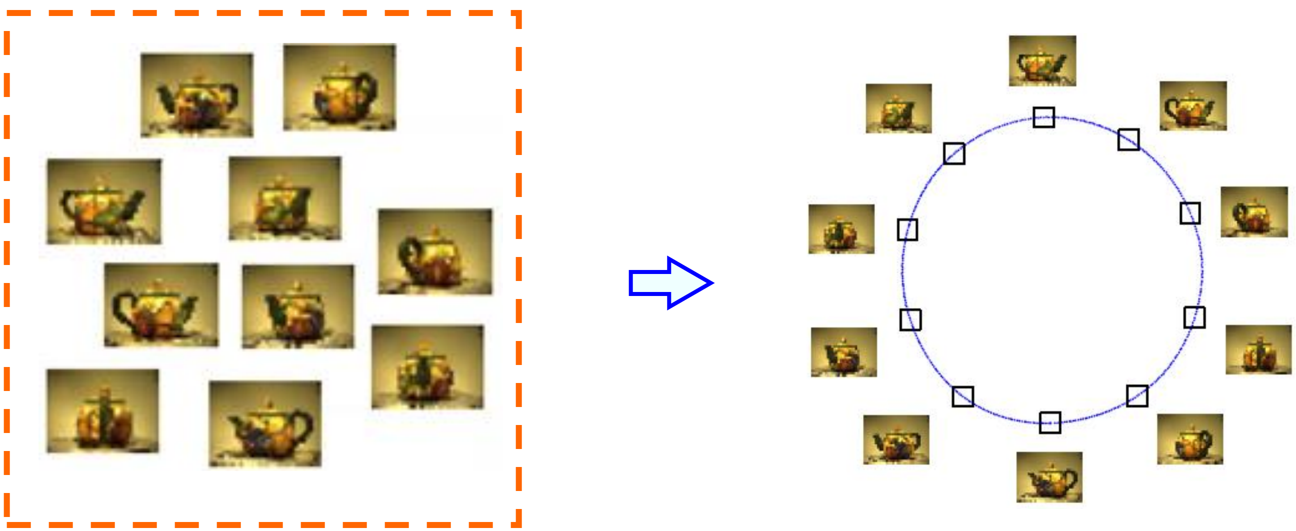




Swiss roll surface is a 2D manifold

7.8 流形学习

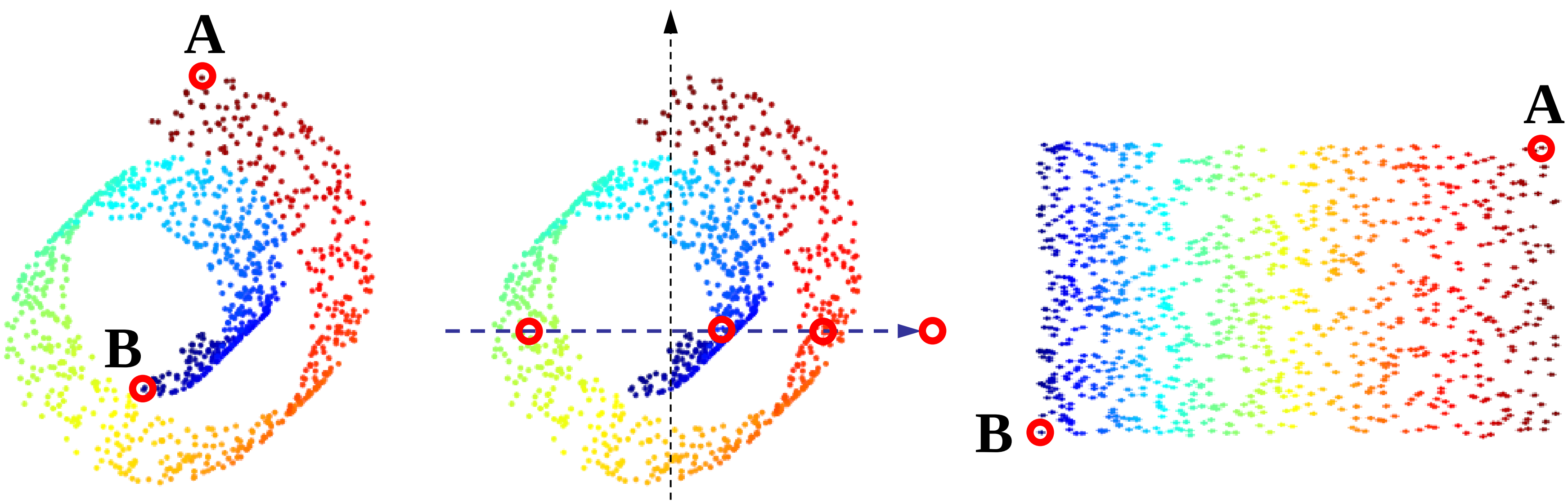
- **What are manifolds**
  - **一个直观的例子**



400张360度全角度拍摄的图片会排列成一个圆！

7.8 流形学习

- **非线性维数缩减**
  - Problem formulation in machine learning：in view of dimensionality reduction



source data　　linearly projecting　　nonlinearly unfolding

通过线性投影将高维数据降到低维将难以展开非线性结构！

7.8 流形学习

- **Problem Formulation**

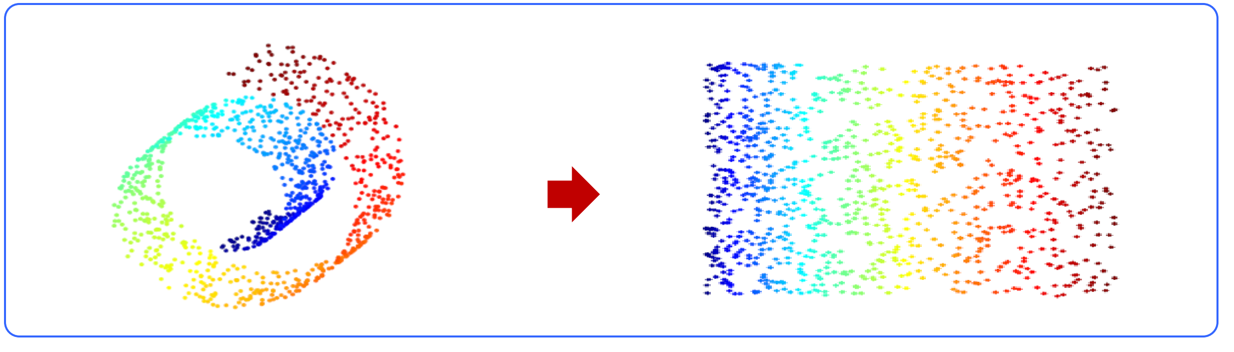
??

Given $\{\mathbf{x}_i\}_{i=1}^{n} \subset R^m$ $\Rightarrow$ $\{\mathbf{y}_i\}_{i=1}^{n} \subset R^d$

$\mathbf{x}_i = f(\mathbf{y}_i)$

unknown

$(d < m)$



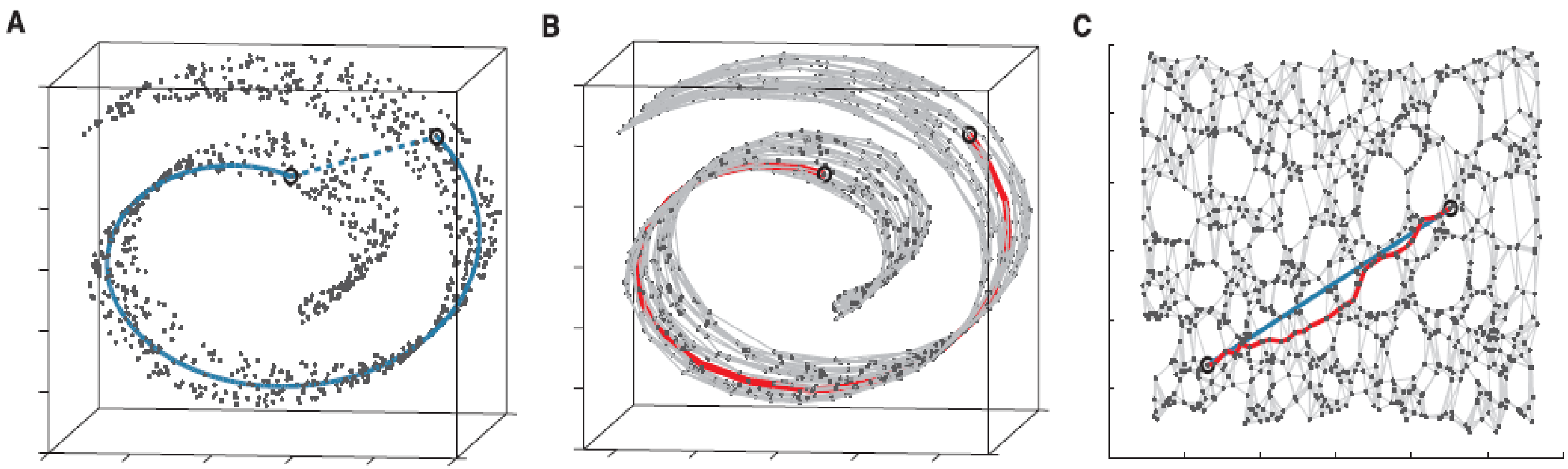
7.8 流形学习

- **一些假定**
  - **smooth manifold:** $(f : \mathbf{C} \subset \mathbf{R}^d \rightarrow \mathbf{R}^m)$
  - **densely sampling:**
  - **no self-intersections:**

- **基本思想：**
  - 高维空间相似的数据点，映射到低维空间距离也是相似的
- **经典算法**
  - Isomap, LLE, Laplacian Eigenmaps, HLLE, MVU, LTSA, LSE, *t*-SNE(stochastic neighbor embedding), etc

7.8 流形学习--Isomap

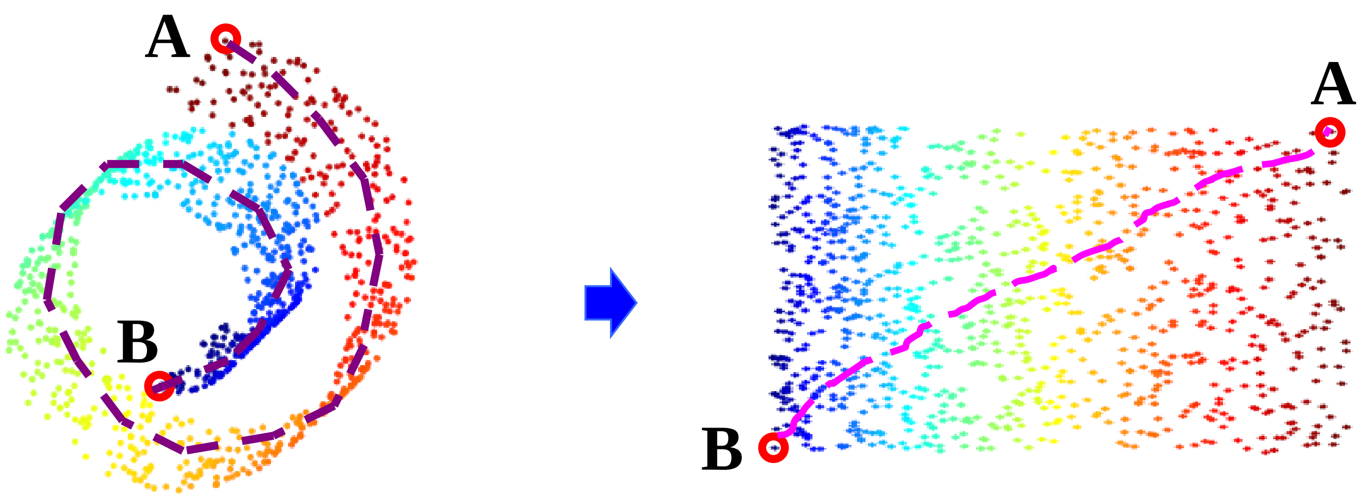
- **Isometric feature mapping (Isomap)**
  - **基本思想：**给定数据集，通过最近邻等方式构造一个数据图(data graph)。然后，计算任意两个点之间的最短路径（即测地距离）。对于所有的任意两个点对，期望在低维空间中保持其测地距离。



data set　　*k*NN data graph　　Dijkstra shortest path

7.8 流形学习--Isomap

- **Isometric feature mapping (Isomap)**
  - **基本思想：**给定数据集，通过最近邻等方式构造一个数据图(data graph)。然后，计算任意两个点之间的最短路径（即测地距离）。对于所有的任意两个点对，期望在低维空间中保持其测地距离。



Isomap算法步骤：

1 Given data $\mathbf{X}=[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]\subset R^{m\times n}$，近邻参数$k$，低维空间$d$

2 for i =1, 2,…, n

3 确定$\mathbf{x}_i$的k个近邻；

4 $\mathbf{x}_i$与近邻点之间的距离设定为欧氏距离，与非近邻点的距离设置为无穷大

5 end for

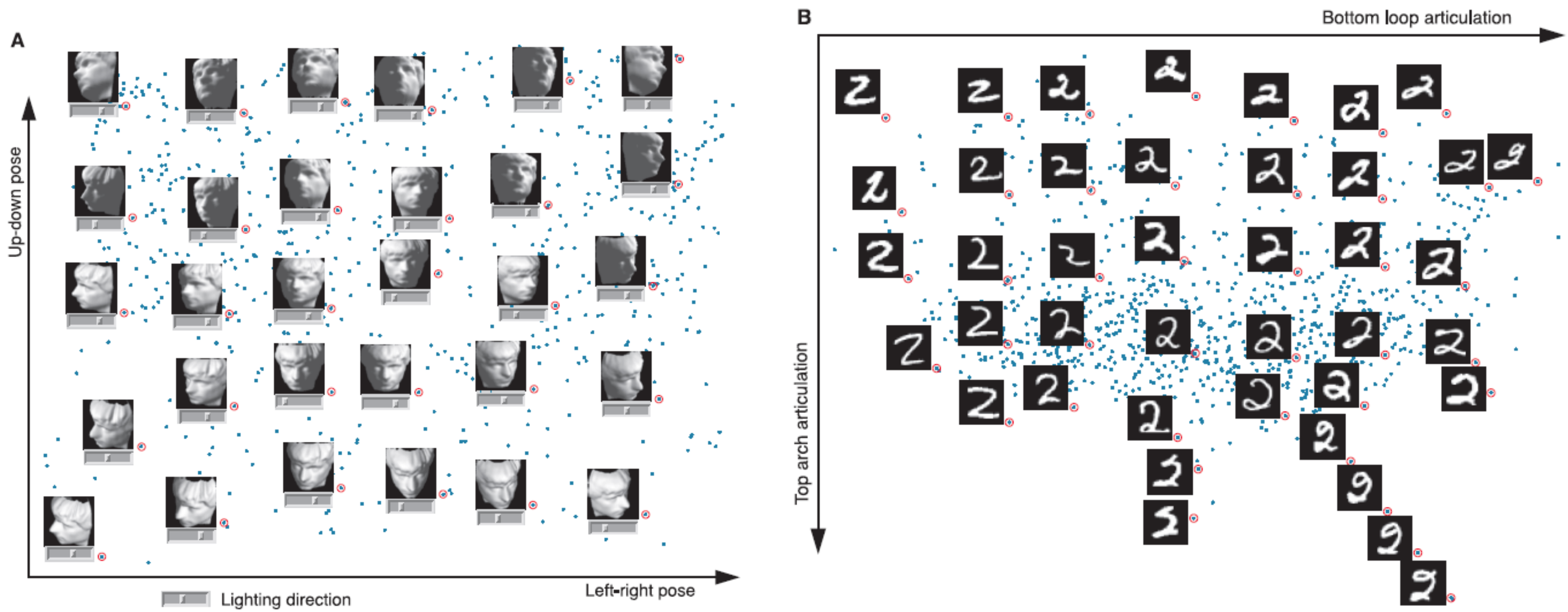
6 调用最短路径算法计算任意两样本点$\mathbf{x}_i$与$\mathbf{x}_j$之间的距离d_{ij}，由此构造距离矩阵。

7 调用MDS算法

输出：MDS算法的计算结果作为低维嵌入

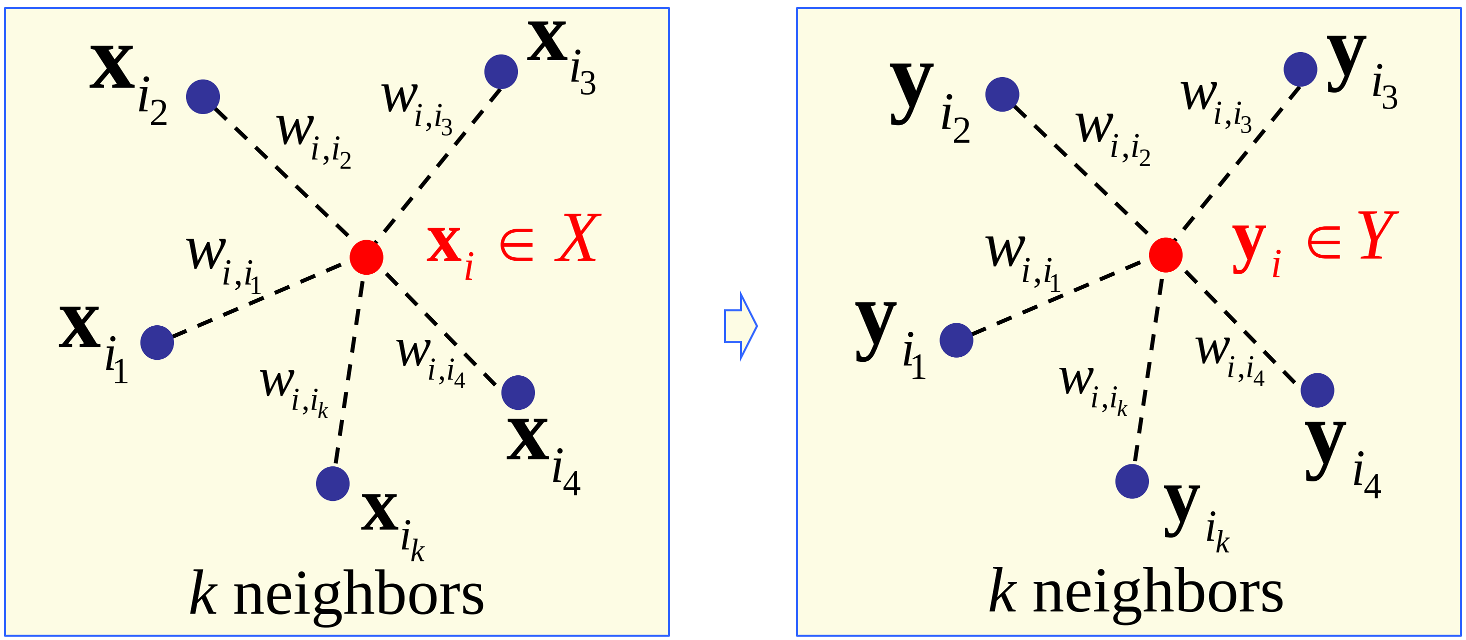
J. B. Tenenbaum, V. de Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” Science, vol. 290, pp. 2319–2323, 2000.

7.8 流形学习--Isomap



7.8 流形学习--LLE

- **Locally linear embedding (LLE)**
  - **基本思想：**给定数据集，通过最近邻等方式构造一个数据图(data graph)。在每一个局部区域，高维空间中的样本线性重构关系在低维空间中均得以保持



$$\mathbf{x}_i \approx \sum_{j=1}^{k} w_{i,i_j} \mathbf{x}_{i_j} \qquad \mathbf{y}_i \approx \sum_{j=1}^{k} w_{i,i_j} \mathbf{y}_{i_j}$$

- **LLE**
  - 最优线性表示系数

$$\min_{\mathbf{w}_i} \left\| \mathbf{x}_i - \sum_{j=1}^{k} w_{i,i_j} \mathbf{x}_{i_j} \right\|_2^2, \quad \text{s.t.} \quad \sum_{j=1}^{k} w_{i,i_j} = 1$$

通过拉格朗日乘子法，可得线性表示系数的解：

$$\mathbf{w}_i = \frac{(\mathbf{X}_i^T \mathbf{X}_i)^{-1} \mathbf{e}}{\mathbf{e}^T (\mathbf{X}_i^T \mathbf{X}_i)^{-1} \mathbf{e}}$$

(自己推导)

$$\mathbf{w}_i = [w_{i,i_1}, w_{i,i_2}, \ldots, w_{i,i_k}]^T \in R^k,$$

$$\mathbf{X}_i = [\mathbf{x}_{i_1} - \mathbf{x}_i, \mathbf{x}_{i_2} - \mathbf{x}_i, \ldots, \mathbf{x}_{i_k} - \mathbf{x}_i] \in R^{m \times k},$$

$$\mathbf{e} = [1, 1, \ldots, 1]^T \in R^k$$

防止矩阵奇异

$$\mathbf{w}_i = \frac{(\mathbf{X}_i^T \mathbf{X}_i + \lambda \mathbf{I})^{-1} \mathbf{e}}{\mathbf{e}^T (\mathbf{X}_i^T \mathbf{X}_i + \lambda \mathbf{I})^{-1} \mathbf{e}}$$

- **LLE**
  - 全局嵌入：利用在原始空间中获得的局部线性重构关系，在低维空间中重构对应的样本点：

$$\mathbf{y}_i \approx \sum_{j=1}^{k} w_{i,i_j} \mathbf{y}_{i_j}, \quad i = 1, 2, \cdots, n$$

  - 考虑所有新样本点的重构误差，得到全局嵌入的目标函数：

$$\min_{\mathbf{Y}} \sum_{i=1}^{n} \left\| \mathbf{y}_i - \sum_{j=1}^{k} w_{i,i_j} \mathbf{y}_{i_j} \right\|_2^2 = \left\| \mathbf{Y} - \mathbf{Y}\mathbf{W}^T \right\|_2^2 = tr\left(\mathbf{Y}(\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W}) \mathbf{Y}^T \right)$$

where $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n] \in R^{d \times n}$.

$\mathbf{W} \in R^{n \times n}$为权重矩阵，其第$i$行记录样本$\mathbf{x}_i$的$k$个权重，只有在对应的邻居位置$i_1, i_2, \ldots, i_k$处才有值，其余全为零。

7.8 流形学习--LLE

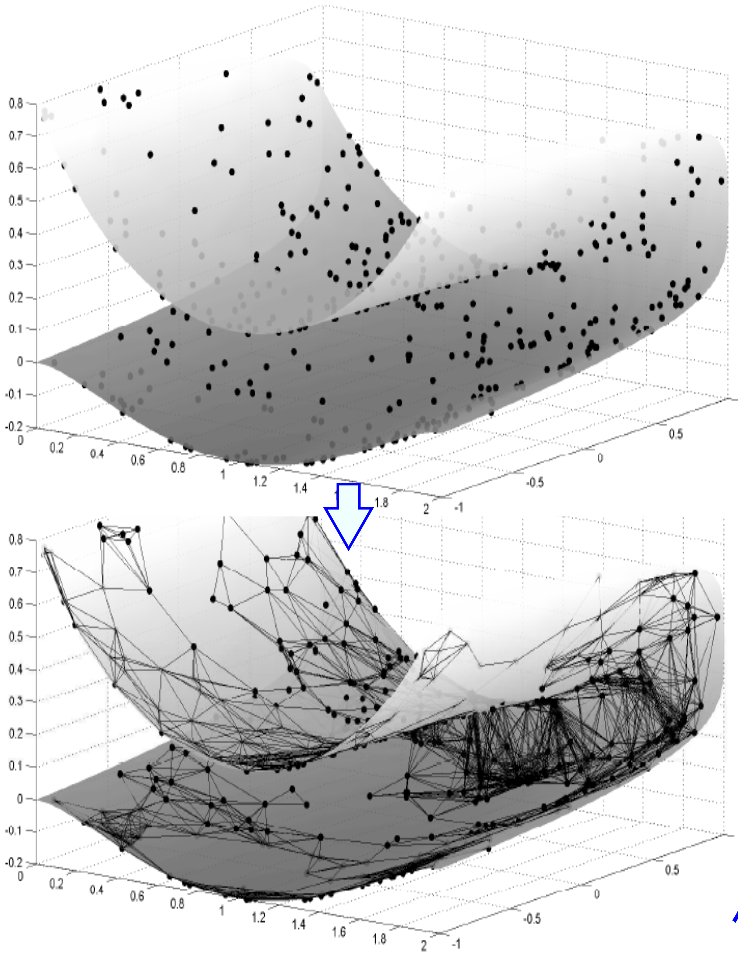
- **LLE**
  - 全局嵌入学习模型：

$$\min_{\mathbf{Y}} \quad tr\left(\mathbf{Y}(\mathbf{I}-\mathbf{W})^T(\mathbf{I}-\mathbf{W})\mathbf{Y}^T\right), \quad \text{s.t.} \quad \mathbf{YY}^T = \mathbf{I}$$

  - 求解：通过求矩阵$(\mathbf{I}\text{-}\mathbf{W})^T(\mathbf{I}\text{-}\mathbf{W})$的特征值分解得到
    - 取出该矩阵最小的d+1个特征值对应的特征向量；
    - 丢弃特征值零对应的分量全相等的特征向量；
    - 即采用第2至第d+1个最小的特征值对应的特征向量组成样本的新的坐标。

S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," Science, vol. 290, pp. 2323–2326, 2000.

- **LLE计算流程：**



对每个样本点，找出其k个近邻

线性表示

$$\mathbf{x}_i \approx \sum_{j=1}^{k} w_{i,i_j}\mathbf{x}_{i_j}$$

$$\text{s.t.} \quad \sum_{j=1}^{k} w_{i,i_j} = 1$$

保持表示

$$\mathbf{y}_i \approx \sum_{j=1}^{k} w_{i,i_j}\mathbf{y}_{i_j}$$

全局误差

$$E(\mathbf{Y}) = tr(\mathbf{Y}(\mathbf{I}-\mathbf{W})^T(\mathbf{I}-\mathbf{W})\mathbf{Y}^T)$$

低维嵌入

$$\min E(\mathbf{Y}), \quad \text{s.t.} \quad \mathbf{Y}\mathbf{Y}^T = \mathbf{I}$$

几乎所有的流形学习方法都需要首先构建一个关于数据的图

LLE算法步骤：

Given Data $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n]\subset R^{m\times n}$，近邻参数$k$，低维空间$d$

for i=1,2,…,n

    确定$\mathbf{x}_i$的k个近邻；

    对$\mathbf{x}_i$进行线性最优表示，获取近邻重构权重

end for

构造权重矩阵$\mathbf{W}$；

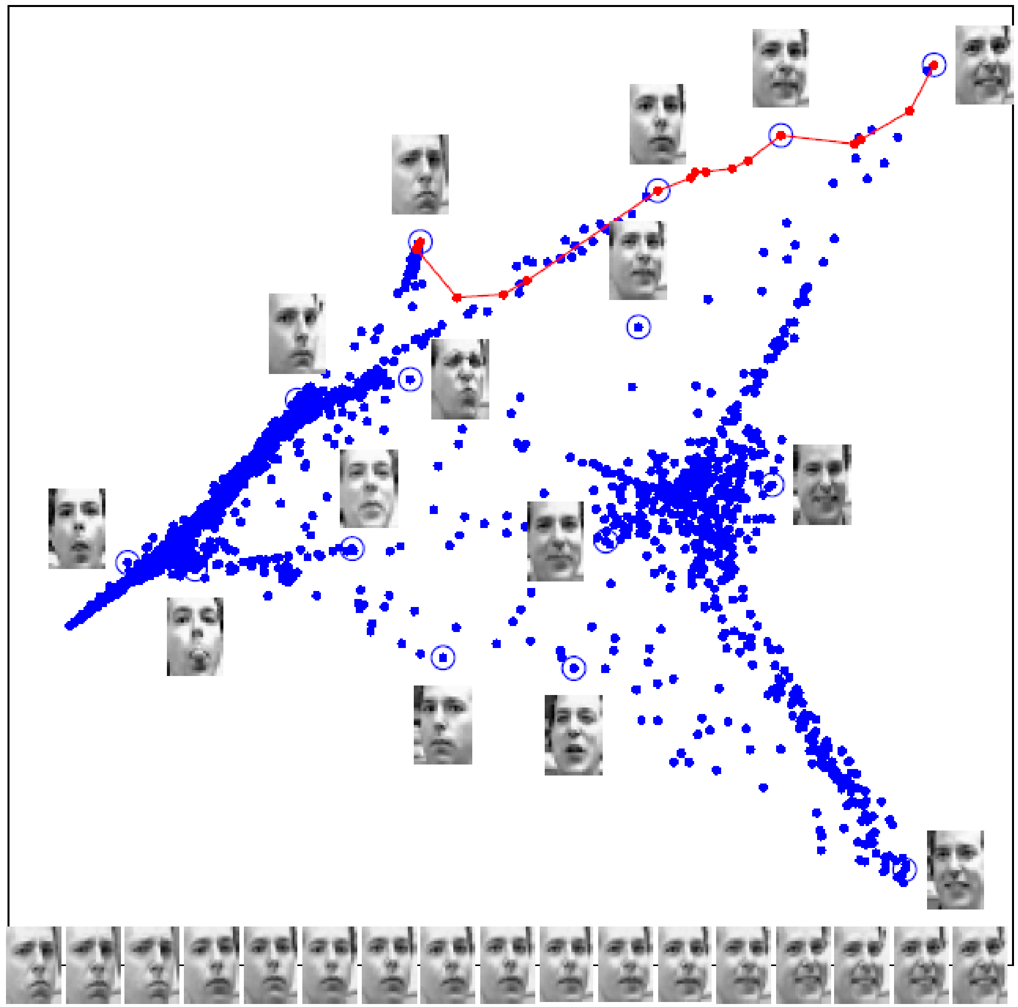
求解：$\min\limits_{\mathbf{Y}} \; tr\left(\mathbf{Y}(\mathbf{I}-\mathbf{W})^T(\mathbf{I}-\mathbf{W})\mathbf{Y}^T\right), \quad s.t. \quad \mathbf{Y}\mathbf{Y}^T=\mathbf{I}$

采用第2至第d +1个最小的特征值对应的特征向量组成新坐标

输出嵌入结果

S. Roweis and L. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” Science, vol. 290, pp. 2323–2326, 2000.

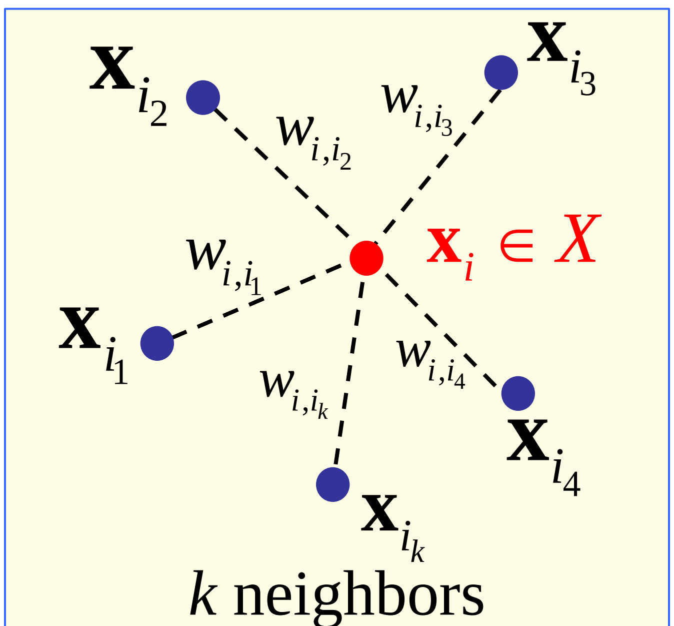
- **LLE的一些结果：**

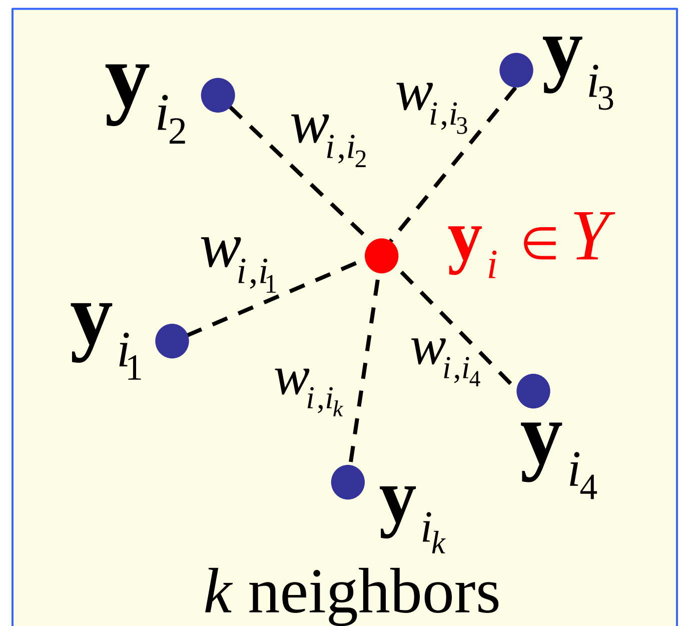


n=1965, m=560, d=2, k=12

7.8 流形学习--LE

- **Laplacian Eigenmaps (LE)**
  - **基本思想：**给定数据集，通过最近邻等方式构造一个数据图(data graph)。在每一个局部区域，计算点与点之间的亲合度（相似度），期望点对亲合度在低维空间中也得到保持。

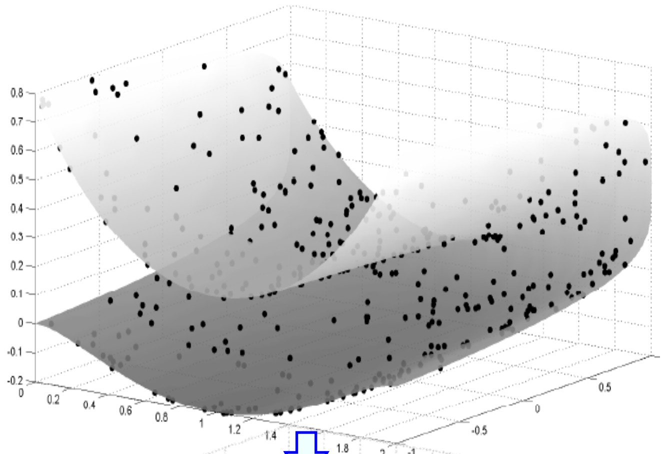


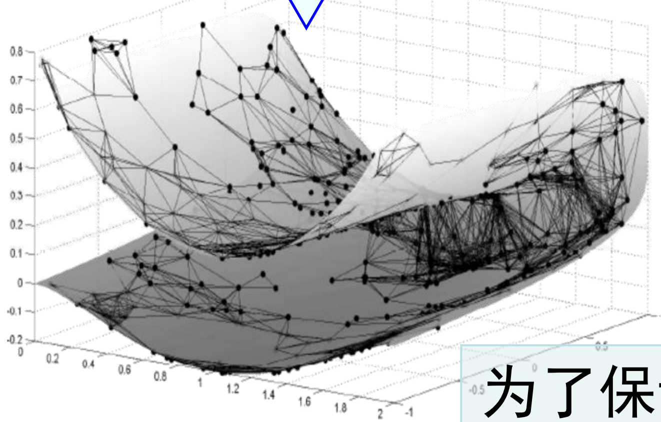


7.8 流形学习--LE

- **LE**
  - 如何计算点对亲合度？
  - 图构造 G(V, E)

$$w_{i,i_j} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_{i_j}\|_2^2}{2\sigma^2}\right)$$





$$\mathbf{W} = \begin{pmatrix} 0 & w_{12} & w_{13} & \cdots & w_{1n} \\ w_{21} & 0 & w_{23} & \cdots & w_{2n} \\ w_{31} & w_{32} & 0 & \cdots & w_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & w_{n3} & \cdots & 0 \end{pmatrix}_{n\times n}$$

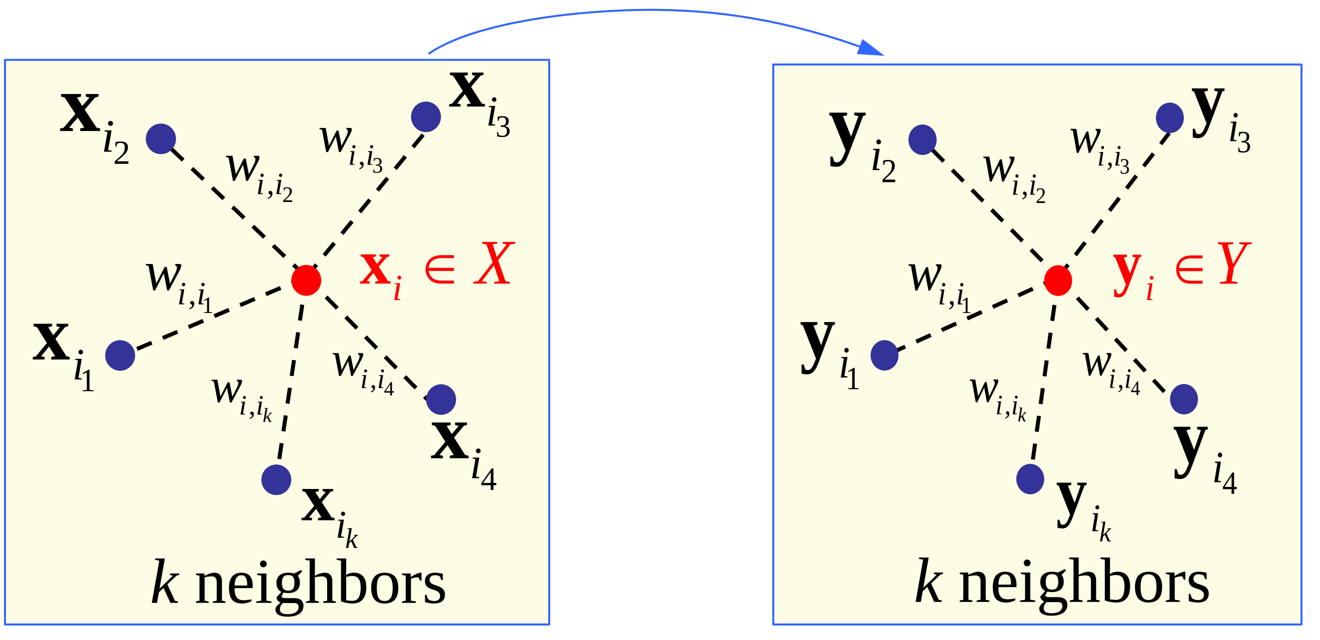
$\mathbf{W} \in R^{n\times n}$为权重矩阵，其第$i$行记录样本$\mathbf{x}_i$的$k$个权重，只有在对应的邻居位置$i_1$, i_2…, i_k处才有值，其余全为零。

为了保证**W**矩阵的对称性，可以令$\mathbf{W}=(\mathbf{W}^T+\mathbf{W})/2$

7.8 流形学习--LE

- **LE**
  - 如何在低维空间保持亲合度？构造如下目标函数:

$$E(\mathbf{Y})=\sum_{i,j} w_{ij}\left\|\mathbf{y}_i-\mathbf{y}_j\right\|_2^2$$



7.8 流形学习--LE

- **LE**：考虑目标函数 $E(\mathbf{Y})=\sum_{i,j} w_{ij}\left\|\mathbf{y}_i-\mathbf{y}_j\right\|_2^2$
  - 对任意向量 $\mathbf{f}=[f_1, f_2, \ldots, f_n]^T \in R^n$，有如下结论：

令

$$\mathbf{D}=\begin{pmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{pmatrix}$$

$$d_i=\sum_{j=1}^{k} w_{i,i_j}, \quad i=1,\ldots,n$$

⇨

$$\sum_{i,j=1}^{n} w_{ij}(f_i-f_j)^2$$

$$=\sum_{i=1}^{n} d_i f_i^2-2\sum_{i,j=1}^{n} f_i f_j w_{ij}+\sum_{j=1}^{n} d_j f_j^2$$

$$=2\left(\sum_{i=1}^{n} d_i f_i^2-\sum_{i,j=1}^{n} f_i f_j w_{ij}\right)$$

$$=2\mathbf{f}^T(\mathbf{D}-\mathbf{W})\mathbf{f}$$

D： 度矩阵；**W**：亲和度矩阵；**D-W**：图拉普拉斯矩阵

7.8 流形学习--LE

- **LE：**考虑目标函数 $E(\mathbf{Y})=\sum_{i,j} w_{ij}\left\|\mathbf{y}_i-\mathbf{y}_j\right\|_2^2$

$$\text{Let}\quad \mathbf{Y}=[\mathbf{y}_1,\mathbf{y}_2,\ldots,\mathbf{y}_n]=\begin{pmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ y_{d1} & y_{d2} & \cdots & y_{dn} \end{pmatrix}=\begin{pmatrix} \mathbf{f}_1^T \\ \mathbf{f}_2^T \\ \vdots \\ \mathbf{f}_d^T \end{pmatrix}\in R^{d\times n}$$

$$\text{where，}\ \mathbf{f}_i=[y_{i1},y_{i2},\ldots,y_{in}]^T\in R^n,\quad i=1,2,\ldots,d$$

$$E(\mathbf{Y})=\sum_{i,j} w_{ij}\|\mathbf{y}_i-\mathbf{y}_j\|_2^2=\sum_{i,j} w_{ij}\left((y_{1i}-y_{1j})^2+\cdots+(y_{di}-y_{dj})^2\right)$$

$$=2\mathbf{f}_1^T(\mathbf{D}-\mathbf{W})\mathbf{f}_1+2\mathbf{f}_2^T(\mathbf{D}-\mathbf{W})\mathbf{f}_2+\cdots+2\mathbf{f}_d^T(\mathbf{D}-\mathbf{W})\mathbf{f}_d$$

$$=2tr(\mathbf{Y}(\mathbf{D}-\mathbf{W})\mathbf{Y}^T)$$

7.8 流形学习--LE

- **LE**
  - **学习模型**

$$\min E(\mathbf{Y}) = tr\left(\mathbf{Y}(\mathbf{D}-\mathbf{W})\mathbf{Y}^T\right), \quad s.t.\ \mathbf{Y}\mathbf{Y}^T = \mathbf{I}$$

  - **令L=D - W**
    - **L有一个特征值为零，对应的特征向量全为1：**

$$\mathbf{Le} = (\mathbf{D}-\mathbf{W})\mathbf{e} = (\mathbf{D}-\mathbf{W})\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} d_1 - \sum_{j=1}^{k} w_{1,1_j} \\ d_2 - \sum_{j=1}^{k} w_{1,1_j} \\ \vdots \\ d_n - \sum_{j=1}^{k} w_{n,n_j} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in R^n$$

$$\Downarrow$$

$$\mathbf{Le} = 0\mathbf{e}$$

7.8 流形学习--LE

- **LE算法步骤**

LE算法步骤：

1 Given data $\mathbf{X}=[\mathbf{x}_1, \mathbf{x}_2,\ldots, \mathbf{x}_n]\subset R^{m\times n}$，近邻参数$k$，低维空间$d$

2 确定$\mathbf{x}_i$的k个近邻；确定亲合度矩阵**W**，计算度矩阵**D**

3 求解模型 $\min E(\mathbf{Y}), \quad s.t.\ \mathbf{Y}\mathbf{Y}^T=\mathbf{I}$

4 采用**D-W**第2至第d+1个最小的特征值对应的特征向量组成低维嵌入**Y**

输出：$\mathbf{Y}\in \boldsymbol{R}^{d\times n}$

M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," Neural Computation, vol. 15, no. 6, pp. 1373–1396, 2003

7.8 流形学习--LPP

- **Locality preserving projections (LPP)**
  - 是LE的线性近似，但同时具备流形学习方法和线性降维方法的优点。
  - 由浙江大学何晓飞教授(2002年于芝加哥大学）提出，一种著名的线性降维方法，在机器学习、模式识别、数据挖掘中得到广泛应用。

Xiaofei He and Partha Niyogi. Locality Preserving Projections. NIPS, 2003.

7.8 流形学习--LPP

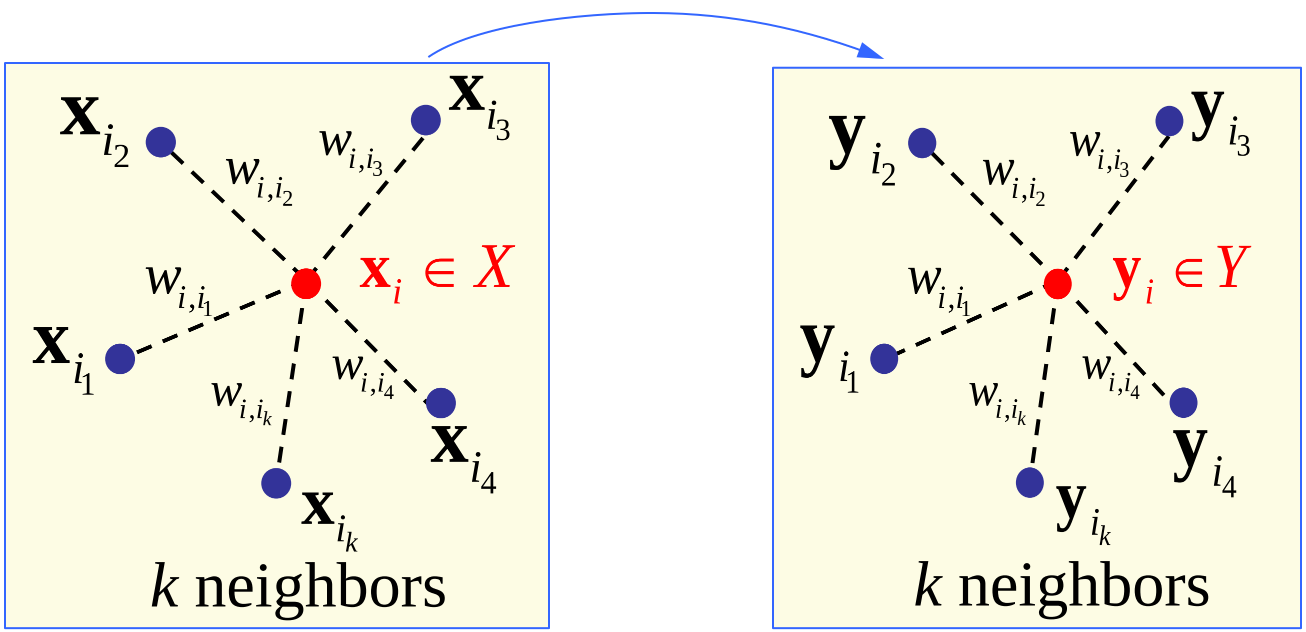
- **LPP算法思想**
  - 构建原空间中各样本点对之间的亲和度关系，并在线性投影中保持这种亲和度。
  - 在降维的同时保留原空间中样本的局部结构，即尽量避免样本集在投影空间中发散，保持原来的近邻结构。
  - 在低维空间中最小化近邻样本间的距离加权平方和。

7.8 流形学习--LPP

- **算法描述**
  - 对样本集$\{\mathbf{x}_1, \mathbf{x}_2,\ldots,\mathbf{x}_n\}\subset R^m$，引入一个线性变换：

$$\mathbf{y} = \mathbf{V}^T\mathbf{x}, \quad \text{where } \mathbf{x} \in R^m, \mathbf{V} \in R^{m\times d}, \mathbf{y} \in R^d, d < m$$

  - 在该线性变换下，希望保持样本的原来的近邻关系：



7.8 流形学习--LPP

- **算法描述**
  - 回顾LE： $\min_{\mathbf{Y}} \quad tr(\mathbf{Y}(\mathbf{D}-\mathbf{W})\mathbf{Y}^T), \quad s.t. \quad \mathbf{Y}\mathbf{Y}^T = \mathbf{I}$
  - 引入线性变换V∈ $R^{m\times d}$，对n个数据点我们有：

$$\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n] = \mathbf{V}^T\mathbf{X} = \mathbf{V}^T[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in R^{d\times n}$$

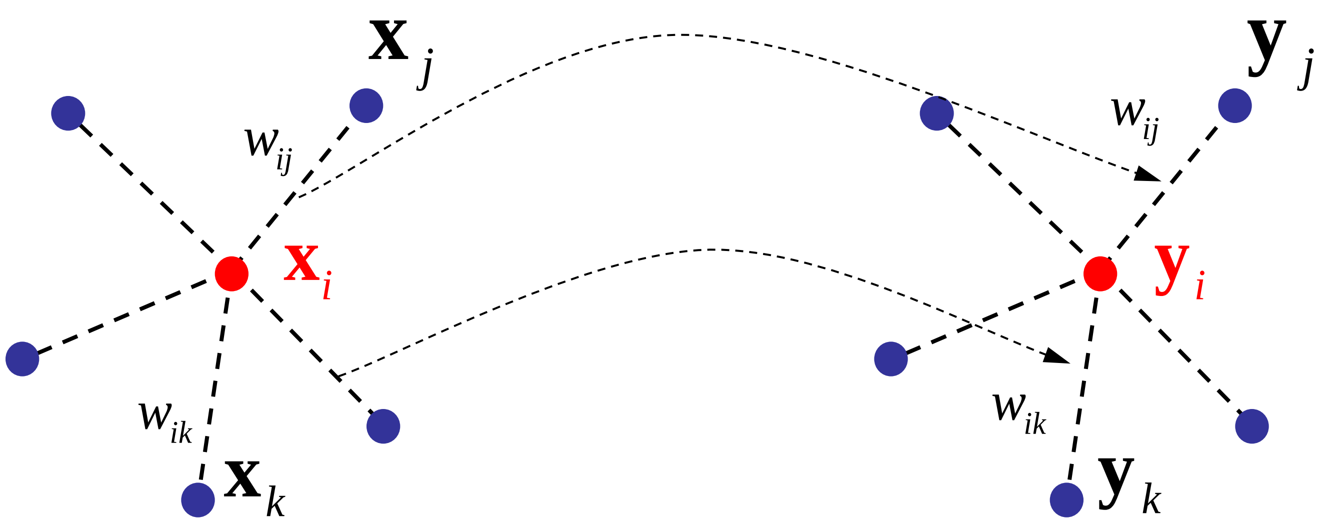
  - 于是，得到LPP的学习模型：

$$\min_{\mathbf{V}} \quad tr\left(\mathbf{V}^T\mathbf{X}(\mathbf{D}-\mathbf{W})\mathbf{X}^T\mathbf{V}\right)$$

$$s.t. \quad \mathbf{V}^T\mathbf{V} = \mathbf{I}$$

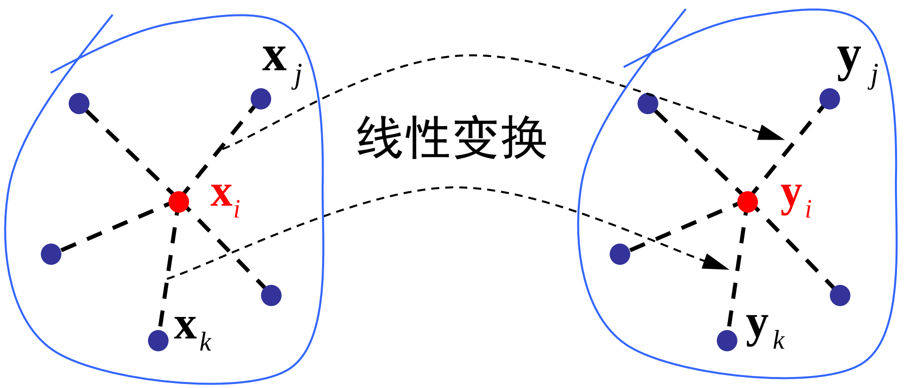
7.8 流形学习

- **其它流形学习思想**
  - 处理好局部关系是构造新的流形学习算法的关键



7.8 流形学习[以下内容略]

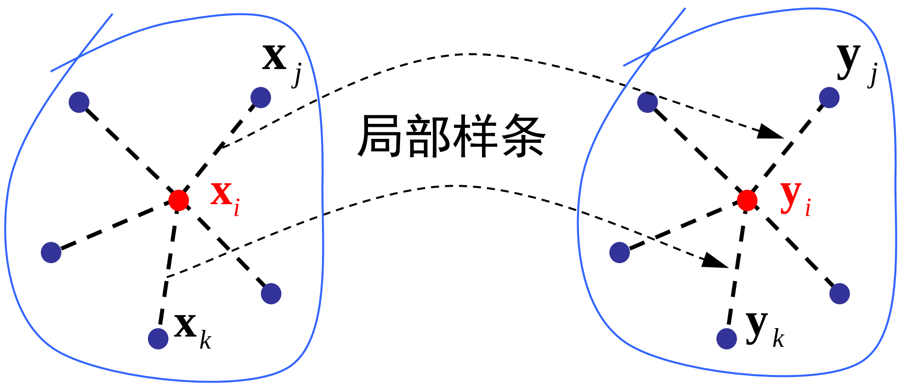
- **局部切空间对齐（LTSA）**
  - **基本思想：**对每一个数据，在局部引入一个线性变换，将其近邻点映射到低维坐标系中的对应近邻点
    - 在最优局部线性变换下，可以计算映射误差。然后将所有局部邻域中计算的误差进行累加，获得全局购入的目标函数，
    - 这就是LTSA算法（浙江大学数学系张振跃老师提出）



Z. Zhang and H. Zha. "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," SIAM Journal on Scientific Computing, vol. 26, no. 1, pp. 313–338, 2004.

7.8 流形学习

- **局部样条嵌入**
  - 对每一个数据，局部引入一个非线性变换，将其近邻点映射到低维坐标系中的对应近邻点
    - 在最优局部样条映射下，可以计算映射误差。然后将所有局部邻域中计算的误差进行累加，获得全局购入的目标函数，
    - 这就是局部样条嵌入(local spline embedding, LSE)



Shiming Xiang, et al.. Nonlinear Dimensionality Reduction with Local Spline Embedding. IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1285-1298, 2009

7.8 流形学习

- **其它流形学习方法**
  - Hessian LLE
  - Maximum variance unfolding
  - Relative distance comparison preserving
  - Stochastic neighbor embedding

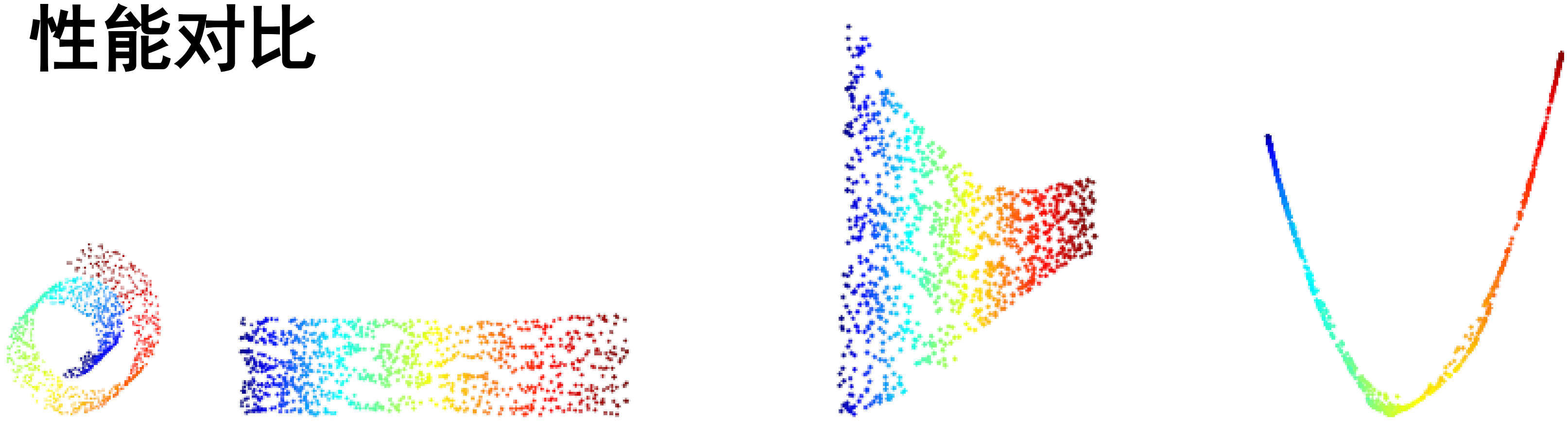
✓ K. Q. Weinberger, F. Sha, and L. K. Saul, "Learning a kernel matrix for nonlinear dimensionality reduction," in International Conference on Machine learning, Banff, Canada, 2004, pp. 888–905.

✓ D. L. Donoho and C. Grimes, "Hessian eigenmaps: locally linear embedding techniques for highdimensional data," Proceedings of the National Academy of Arts and Sciences, vol. 100, no. 10, pp. 5591–5596, 2003.

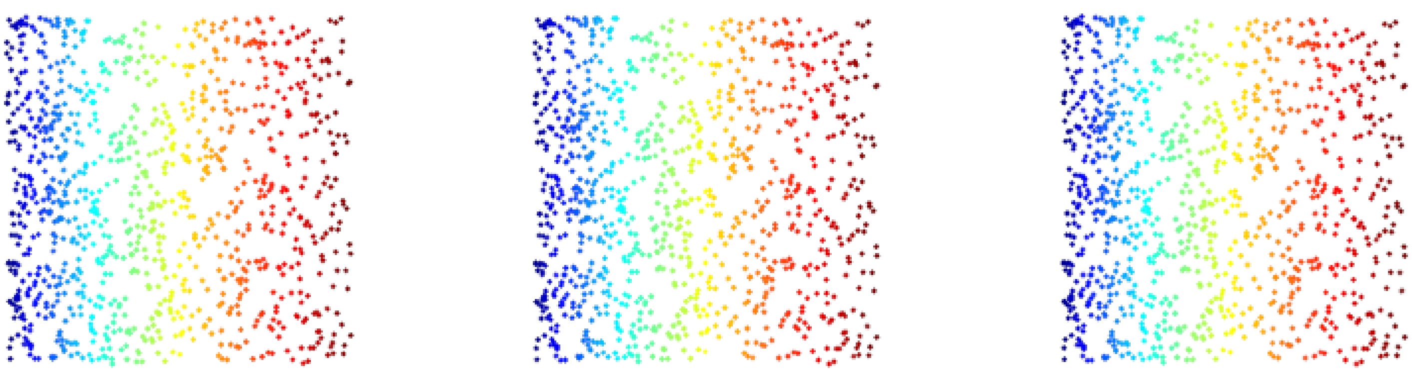
✓ Van der Maaten L, Hinton G. Visualizing data using t-SNE. Journal of Machine Learning Research, 9(2579-2605): 85，2008.

7.8 流形学习

- **性能对比**



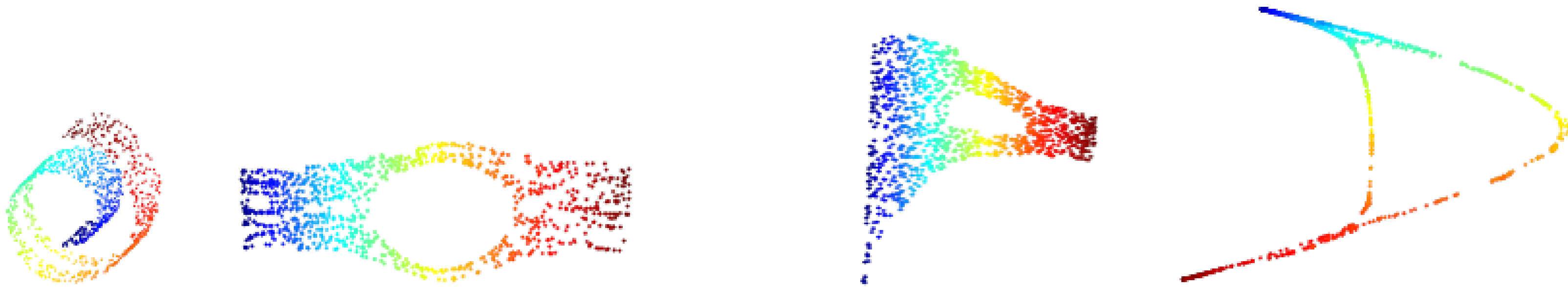
(a) Data (b) Isomap (c) LLE (d) LE



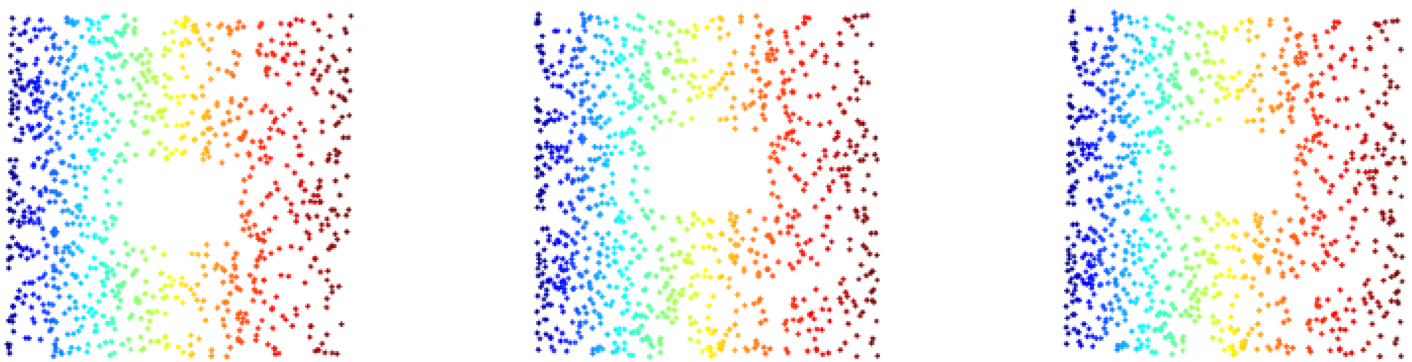
(e) HLLE (f) LTSA (g) LSE

7.8 流形学习

- **性能对比**



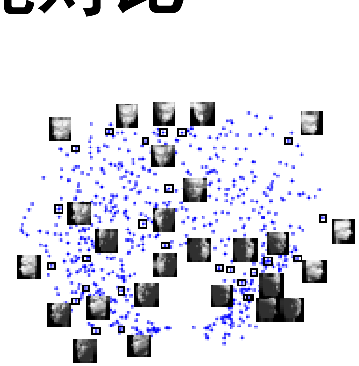
(a) Data (b) Isomap (c) LLE (d) LE



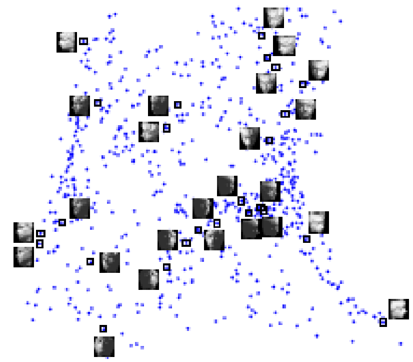
(e) HLLE (f) LTSA (g) LSE

7.8 流形学习

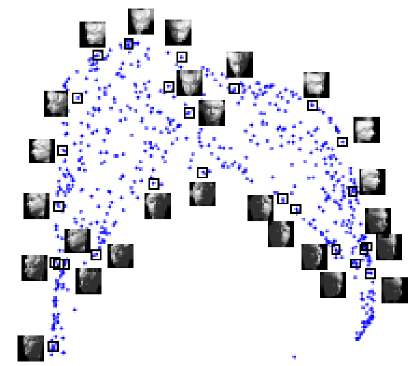
- **性能对比**



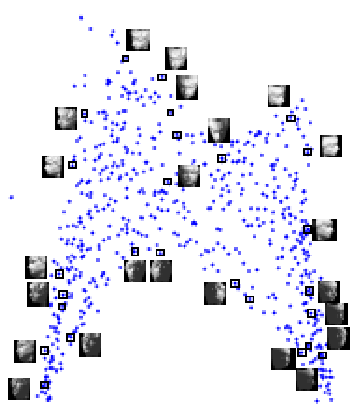
(a) Isomap



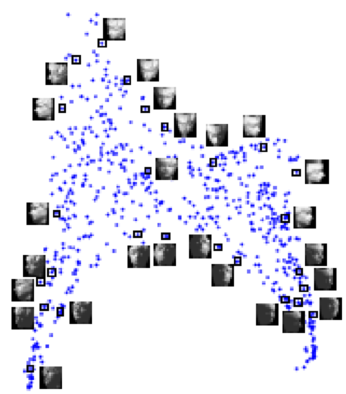
(b) LLE



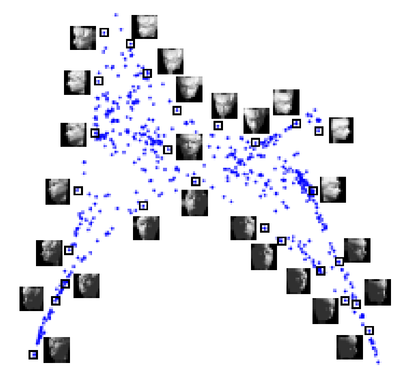
(c) LE



(d) HLLE



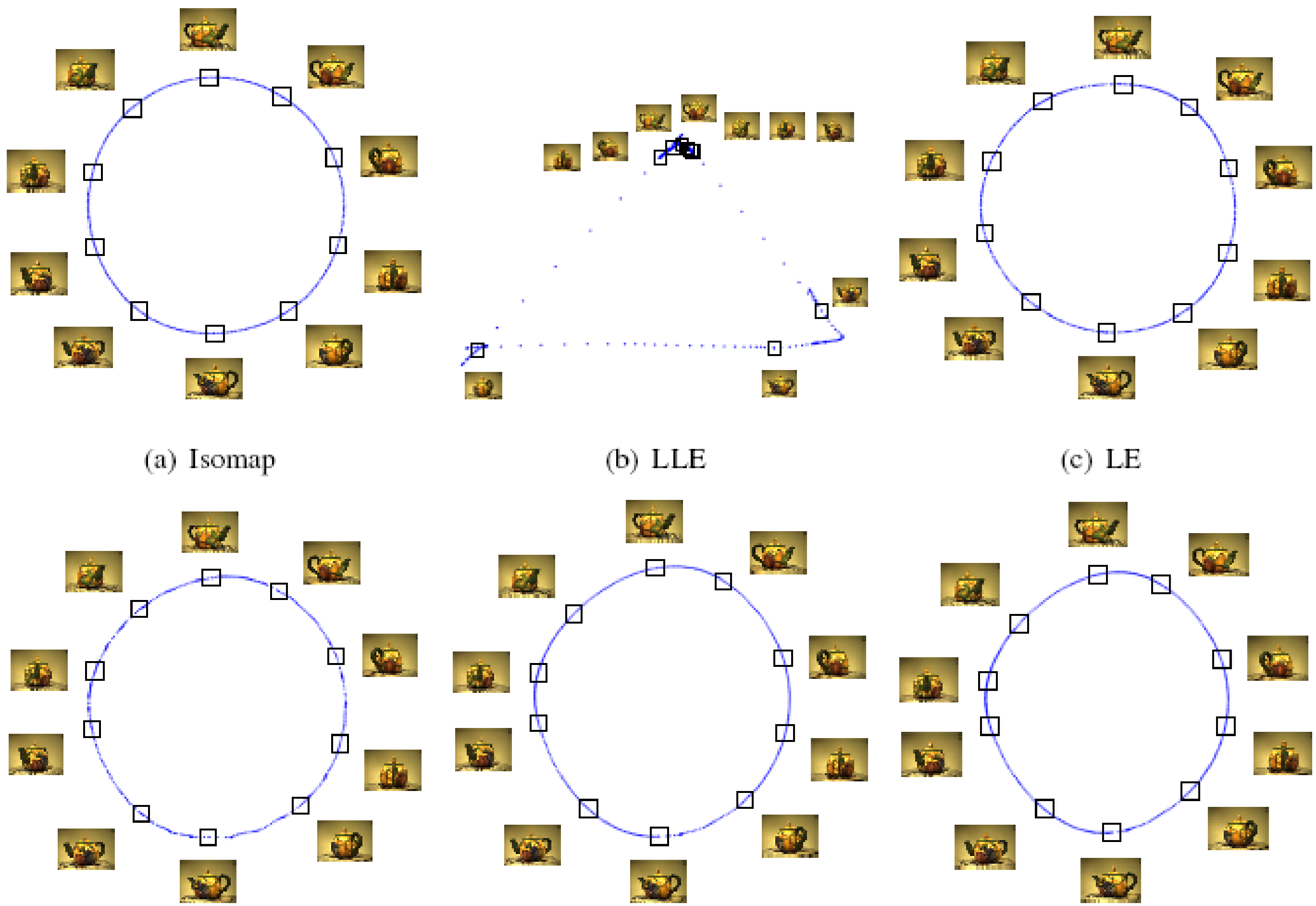
(e) LTSA



(f) LSE

7.8 流形学习

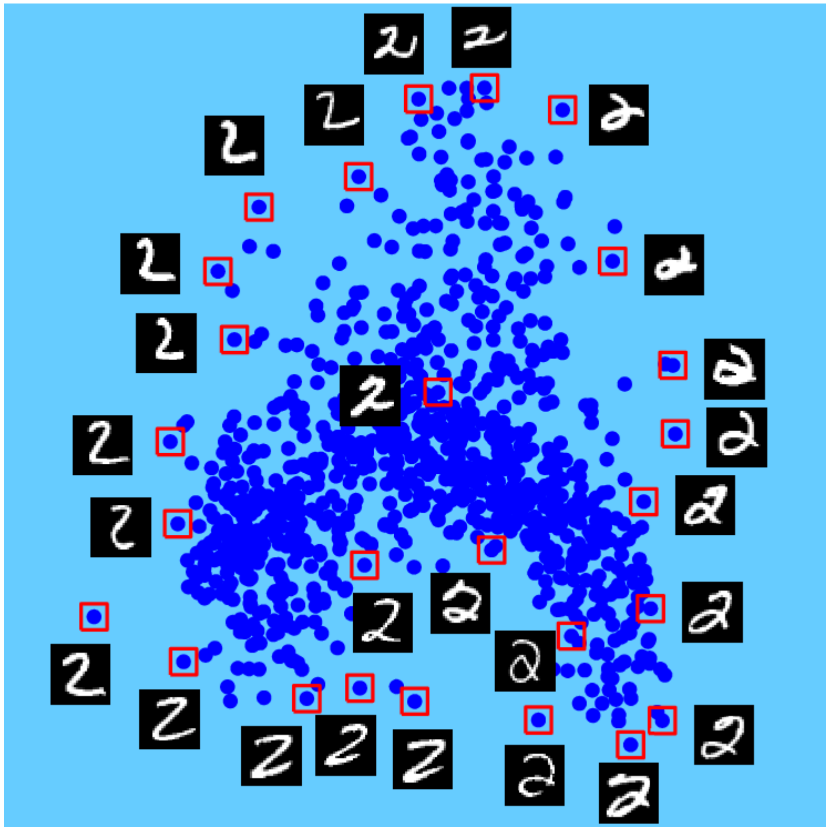
- **性能对比**



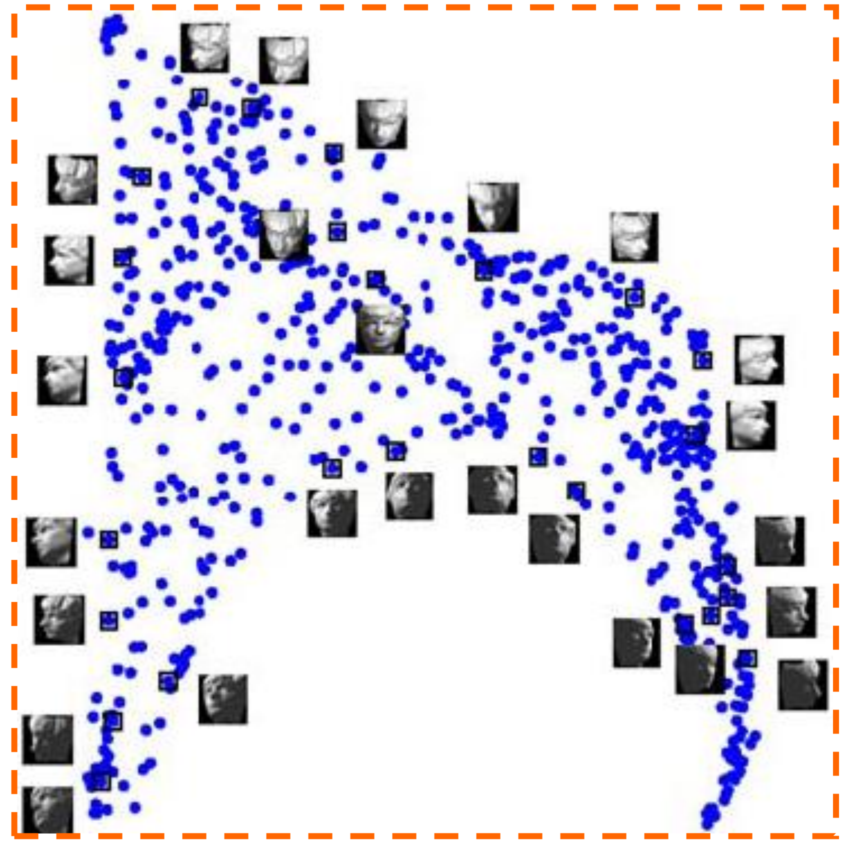
(a) Isomap (b) LLE (c) LE

(d) HLLE (e) LTSA (f) LSE

7.8 流形学习



风格



姿态+光照

Shiming Xiang, et al.. Nonlinear Dimensionality Reduction with Local Spline Embedding. IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1285-1298, 2009

7.8 流形学习

- **统一的学习模型**
  - ✓ 目标：给定高维数据 $\{\mathbf{x}_i\}_{i=1}^{n} \subset R^m$ 寻找其低维表示
  - ✓ 学习模型：$\{\mathbf{y}_i\}_{i=1}^{n} \subset R^d \quad (d < m)$

图拉普拉斯矩阵

对**M**进行特征值分解

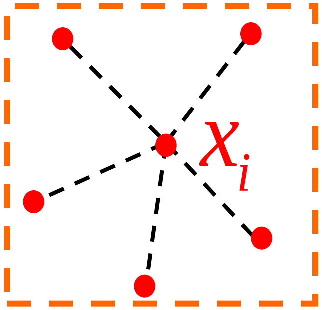
$$\min_{\mathbf{Y}} \quad \mathrm{trace}(\mathbf{YMY}^T)$$

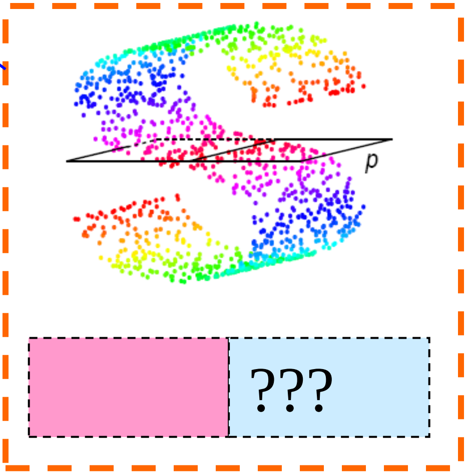
$$s.t. \quad \mathbf{YY}^T = \mathbf{I} \quad (\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n] \in R^{d\times n})$$

任务：构造 M 矩阵--与数据图构造和局部描述紧密相关！

7.8 流形学习

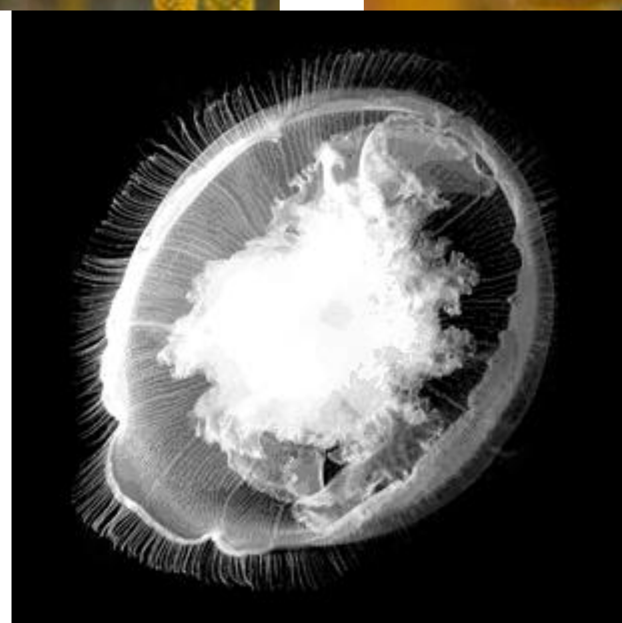
- **流形学习中的一些挑战性问题**
  - 低维本质维数的确定
  - 如何构建一个好的数据图
  - 如何将新样本嵌入到已有的低维结构中去，即所谓的out-of-sample problem
  - 超大规模计算





比如：1千万个数据，意味着需要对1千万x1千万大小的矩阵进行特征值分解！

流形学习应用



基于图的半监督分类

Shiming Xiang, et al., Semi-Supervised Classification via Local Spline Regression, T-PAMI, 2010.

第二部分：特征选择

7.9 特征选择--引言

回顾特征变换：

从一组已有特征进行变换，得到新特征的过程：

- 降低特征空间的维度，缓解“维数灾难”，减少计算量
- 减少特征之间可能存在的相关性，降低分类器学习的难度
- 处理高维数据的两大主流技术之一

线性特征变换（子空间分析）：采用线性变换将原特征变换至一个新的空间（通常维度更低），PCA、LDA

非线性特征变换：采用非线性变换将原特征变换至一个新的空间（通常性能更好），KPCA、KLDA

7.9 特征选择--引言

- **特征选择任务：**
  - 给定一个学习任务，对于给定的数据特征集，从中选出与任务相关（对学习任务有利）的特征子集。
- **特征选择目的：**
  - 处理高维数据的两大主流技术之一；
  - 减少数据维度，缓解“维数灾难”，减少计算量；
  - 通过去除与任务不相关特征、冗余特征、或者关联性较小的特征，降低学习任务的难度；
  - 通过选择与任务相关的特征，提高分类器性能。

特征变换和特征选择是处理高维数据的两大主流技术

7.9 特征选择--引言

- **特征选择的总体技术路线：**子集搜索+子集评价
  - 子集搜索(subset search)： 从特征集合$\{x_1, x_2,\ldots, x_d\}$中搜索最优的特征子集。
  - 子集评价(subset evaluation)： 对给定的特征子集，依据某种评价准则，对其优劣进行评价。
    - 通常基于类别可分性来进行特征子集评价。
    - 常用的判定准则包括：信息增益、信息熵等。

7.9 特征选择--引言

- **特征子集评价判据：**评价一组特征性能好坏的标准
  - 直接判据：分类器的分类错误率
  - 间接判据：与分类器的分类性能存在一定关系的判据
    - 不同类别数据的可分程度
    - 不同类别的概率分布的差异性
    - 特征对于分类的不确定性程度
  - 评价准则
    - 基于距离的准则（Distance-based criterion）
    - 基于分布的准则（Distribution-based criterion）
    - 基于熵的准则（Entropy-based criterion）

- **什么是“理想的”评价准则？**
  - 对分类任务，评价准则 J_{ij} 反映在一组特征下，第i和第j类的可分程度
  - **理想的评价准则应满足：**
    - 与分类错误率具有正相关性，以反映特征的分类性能
    - 对于独立特征，评价标准应具有可加性

$$J_{ij}(\mathbf{x}) = J_{ij}(x_1, x_2, \cdots, x_d) = \sum_{k=1}^{d} J_{ij}(x_k)$$

    - 是特征数目的单调函数，即新加入特征不会减少可分度：

$$\begin{cases} J_{ij} > 0, & \text{for } i \neq j \\ J_{ij} = 0, & \text{for } i = j \\ J_{ij} = J_{ji} & \\ J_{ij}(x_1, x_2, \cdots, x_d) \leq J_{ij}(x_1, x_2, \cdots, x_d, x_{d+1}) & \end{cases}$$

7.9 特征选择--引言

- **基于距离的评价准则**
  - 记 $\mathbf{x}_k^{(i)} \in R^d$和 $\mathbf{x}_l^{(j)} \in R^d$ 分别为类别 ω_i 和 ω_j 的两个样本
  - 记两者之间的距离为：$d(\mathbf{x}_k^{(i)}, \mathbf{x}_l^{(j)})$
  - 定义所有类别上的总距离为：

$$J_d(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{c} P_i \sum_{j=1}^{c} P_j \frac{1}{n_i n_j} \sum_{k=1}^{n_i} \sum_{l=1}^{n_j} d\left(\mathbf{x}_k^{(i)}, \mathbf{x}_l^{(j)}\right)$$

P_i: 第i类的先验概率

n_i: 第i类的样本数

  - 若采用平方欧氏距离：$d\left(\mathbf{x}_k^{(i)}, \mathbf{x}_l^{(j)}\right) = \left(\mathbf{x}_k^{(i)} - \mathbf{x}_l^{(j)}\right)^T \left(\mathbf{x}_k^{(i)} - \mathbf{x}_l^{(j)}\right)$

$$\Rightarrow \quad J_d(\mathbf{x}) = \sum_{i=1}^{c} P_i \left[\frac{1}{n_i} \sum_{k=1}^{n_i} \left(\mathbf{x}_k^{(i)} - \mathbf{m}_i\right)^T \left(\mathbf{x}_k^{(i)} - \mathbf{m}_i\right) + \left(\mathbf{m} - \mathbf{m}_i\right)^T \left(\mathbf{m} - \mathbf{m}_i\right) \right]$$

$\mathbf{m}_i$: 第i类的中心　　$\mathbf{m}$: 所有数据点的中心

7.9 特征选择--引言

- **基于距离的评价准则**
  - 利用散度矩阵，可将上式整理成更简单的形式
  - 定义类间散度如下：

$$\mathbf{S}_b = \sum_{i=1}^{c} P_i (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T$$

  - 定义类内散度如下：

$$\mathbf{S}_w = \sum_{i=1}^{c} P_i \frac{1}{n_i} \sum_{k=1}^{n_i} \left(\mathbf{x}_k^{(i)} - \mathbf{m}_i\right)\left(\mathbf{x}_k^{(i)} - \mathbf{m}_i\right)^T$$

  - 则有：

$$J_d(\mathbf{x}) = tr\left(\mathbf{S}_b + \mathbf{S}_w\right)$$

特别地，上述计算可以定义在任何特征子集上！

7.9 特征选择--引言

- **基于距离的评价准则**
  - 类似于线性判别准则，可定义如下的评价准则，其核心思想是使类内散度尽可能小，类间散度尽可能大。
  - 常用的5个判据：

$$J_1(\mathbf{x}) = tr\left(\mathbf{S}_b + \mathbf{S}_w\right), \qquad J_2(\mathbf{x}) = tr\left(\mathbf{S}_w^{-1}\mathbf{S}_b\right),$$

$$J_3(\mathbf{x}) = \frac{tr\left(\mathbf{S}_b\right)}{tr\left(\mathbf{S}_w\right)}, \qquad J_4(\mathbf{x}) = \frac{\left|\mathbf{S}_b\right|}{\left|\mathbf{S}_w\right|}, \qquad J_5(\mathbf{x}) = \frac{\left|\mathbf{S}_b + \mathbf{S}_w\right|}{\left|\mathbf{S}_w\right|}$$

7.9 特征选择--引言

- **基于分布的评价准则：**基于类条件概率密度函数
  - 假设定义了有关类条件概率密度函数 $p(\mathbf{x}|\omega_i)$ 和 $p(\mathbf{x}|\omega_j)$ 之间的一个“距离”函数：
    - 该“距离”函数应该是非负的；
    - 该距离应反映这两个条件分布之间的重合程度；
      - 当这两个条件分布不重叠时，该距离函数取得最大值
      - 当这两个条件分布一样时，该距离函数应该取零值

7.9 特征选择--引言

- **基于分布的评价准则：**基于类条件概率密度函数
  - 定义x点处的对数似然比：$L_{ij}(\mathbf{x}) = \ln \dfrac{p(\mathbf{x}|\omega_i)}{p(\mathbf{x}|\omega_j)}$
  - KL散度：也称为相对熵(relative entropy)，定义为对数似然比的数学期望：

$$KL_{ij} \triangleq E\left[L_{ij}(\mathbf{x})\right] = \int p(\mathbf{x}|\omega_i) \ln \frac{p(\mathbf{x}|\omega_i)}{p(\mathbf{x}|\omega_j)} d\mathbf{x}$$

  - 注意，KL散度不是一个度量：$KL_{ij} \neq KL_{ji}$
  - 可将其变成一个度量：

$$J_{ij} = KL_{ij} + KL_{ji} = \int \left[p(\mathbf{x}\,|\,\omega_i) - p(\mathbf{x}\,|\,\omega_j)\right] \ln \frac{p(\mathbf{x}|\omega_i)}{p(\mathbf{x}|\omega_j)} d\mathbf{x}$$

7.9 特征选择--引言

- **基于熵的评价准则：**基于类后验概率分布
  - 后验概率 $p(\omega_i|\mathbf{x})$ 反映特征 $\mathbf{x}$ 刻画类别 i 的有效性
  - 两个极端例子：
    - 如果后验概率对于所有的类别都一样，即 $p(\omega_i|\mathbf{x})=1/c$，则说明该特征对类别没有任何鉴别性；
    - 如果后验概率对于类别i为1，对其他类别均为0，即 $p(\omega_i|\mathbf{x})=1$，则说明特征非常有效；
  - 对于某个给定特征，样本属于各类的后验概率越平均，越不利于分类；如果越集中于某一类，则越有利于分类。
  - 因此，可利用后验概率的信息熵来度量特征对类别的可分性。

7.9 特征选择--引言

- **基于熵的评价准则：**基于类后验概率分布
  - 信息熵
    - 可用来衡量一个随机事件发生的不确定性，不确定越大，信息熵越大
    - 香农熵：

$$H(\mathbf{x}) = -\sum_{i=1}^{c} P(w_i|\mathbf{x}) \log_2 P(w_i|\mathbf{x})$$

    - 平方熵：

$$H(\mathbf{x}) = 2\left[1 - \sum_{i=1}^{c} \left(P(w_i|\mathbf{x})\right)^2\right]$$

  - 使用时要求期望

$$E\left[H(\mathbf{x})\right] = \int p(\mathbf{x}) H(\mathbf{x}) d\mathbf{x}$$

7.9 特征选择--引言

- **子集搜索**
  - 任务：从给定的特征集合中选择最优的特征子集
  - 子集搜索是典型的组合问题 (组合爆炸)。若从 d 个特征中选择 m 个，则特征的组合数目为：

$$\frac{d!}{(d-m)!m!}$$

7.9 特征选择--引言

- **子集搜索**
  - 根据子集搜索策略不同：
    - 穷举法：搜索所有的特征组合，可保证找到最优解。
    - 前向搜索策略：在特征选择的迭代过程中，**每次只加入一个新特征**，并对得到的特征子集进行评价，**直到不会显著增加特征子集性能为止。**
    - 后向搜索策略：从完整特征集合开始，**每次迭代去掉一个无关特征**，并对得到的特征子集进行评价，**直到会显著降低特征子集性能为止。**
    - 双向搜索策略：将前向特征选择和后向特征选择相结合。
    - 随机搜索策略：使用随机策略进行子集搜索，然后对得到的特征子集进行评价。

7.10 最优特征选择方法

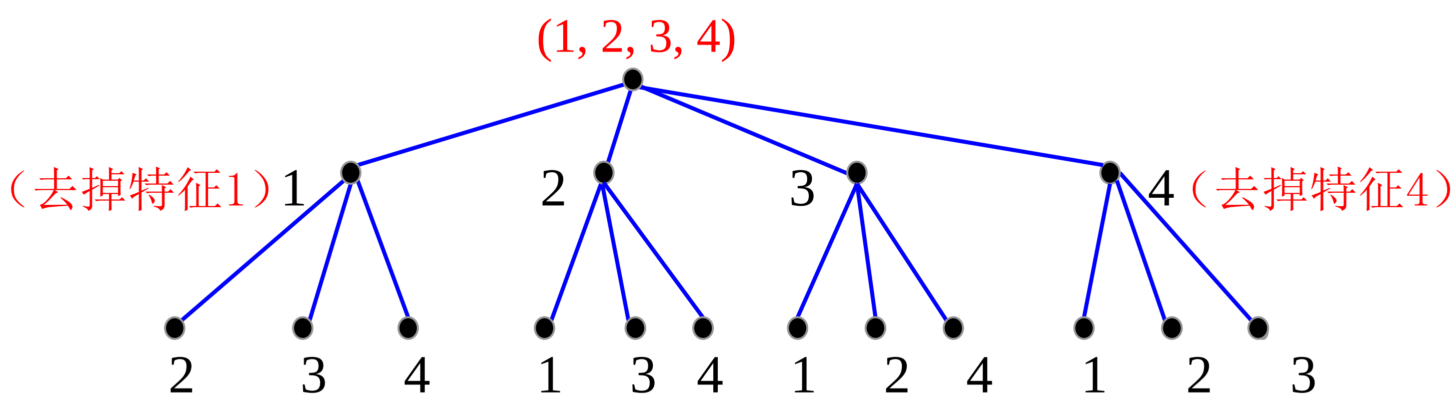
- 最优方法之一：穷举法
  - 从给定的 d 个特征中，挑选出 m 个特征，若采用穷举法，需要遍历$\frac{d!}{(d-m)!m!}$个子集。当 d 很大时，该方法计算量巨大。能否有更聪明的搜索方法？
- 最优方法之二：分支定界法(Branch and Bound)
  - 基本思想：将所有可能的特征组合以树的形式进行表示，采用分枝定界方法对树进行搜索，使得搜索过程尽早达到最优解，而不必搜索整个树。
  - 基本前提：特征评价准则对特征具有单调性，即特征增多时，判据值不会减少：

$$X_1 \subset X_2 \subset \cdots \subset X_m \Rightarrow J(X_1) \le J(X_2) \le \cdots \le J(X_m)$$

基于KL散度和基于信息熵的评价准则满足上述要求

7.10 最优特征选择方法

- **分支定界法：特征子集的树表示**
  - 根节点包含全部特征；
  - 每一个节点，在父节点基础上去掉一个特征，并将去掉的特征序号写在节点的旁边；
  - 对 d 维特征，若选择 m 个特征，每一级去掉一个特征，则需要d-m层达到所需特征数量，即树的深度为d-m 。
    - 比如，可能形成如下一棵树：



d=4, m=2

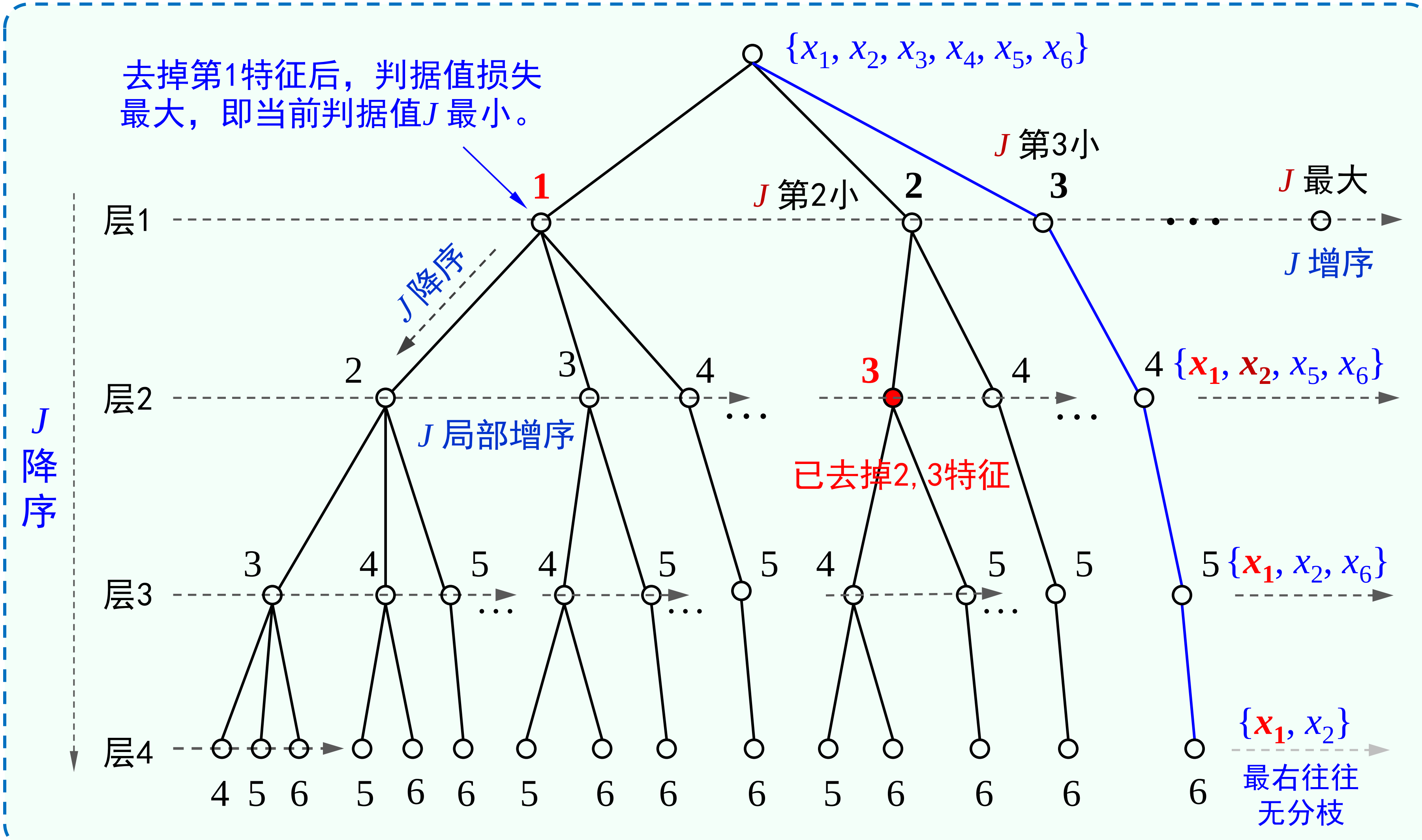
7.10 最优特征选择方法

- **树的生长过程（树的构造）**
  - 1. 将所有特征组成根节点，根节点记为第0层；
  - 2. 对于第1层，分别计算“去掉上一层节点单个特征后”剩余特征的评价判据值，按判据值从小到大进行排序，按从左到右的顺序生成第1层的节点；
  - 3. 对于第2层，针对上一层最右侧的节点开始，重复第2步；
  - 4. 依次类推，直到第 d-m 层，到达叶子结点，记录对应的判据值 J，同时记录对应的特征选择集合
- **回溯：**从第一个叶子结点开始，对树进行回溯

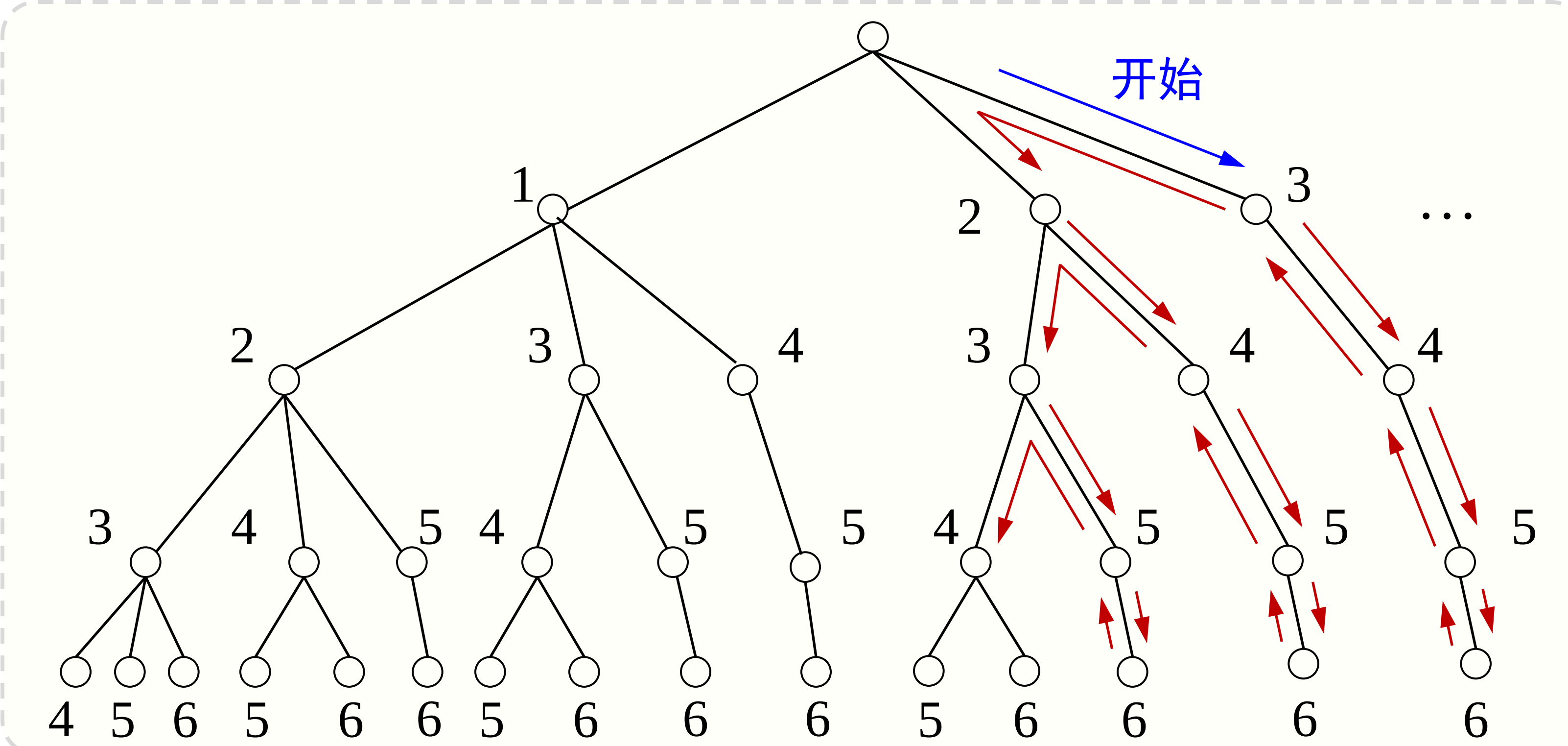
7.10 最优特征选择方法

- **树的生长过程—细节解释**
  - 树的生长：同一层左侧节点对应的特征集合的判据值J小于右侧节点的判据值。
  - 如果去掉某个特征后，准则函数的损失（即J的减少量）最大，则该特征最不可能被去掉，因此将其放在该层的最左侧。
  - 根据评价判据的单调性：下层节点对应的特征集合判据值小于上层节点。
  - 在同一节点D_i下：至多生成 $|D_i|-m+1$个子节点。
  - 在建立下一层时：从最右侧节点开始生长，已在左侧的节点上的特征在本节点之下不再舍弃。
  - 当到达叶节点时：计算当前到达的准则函数值，记作界限B。

树的构造



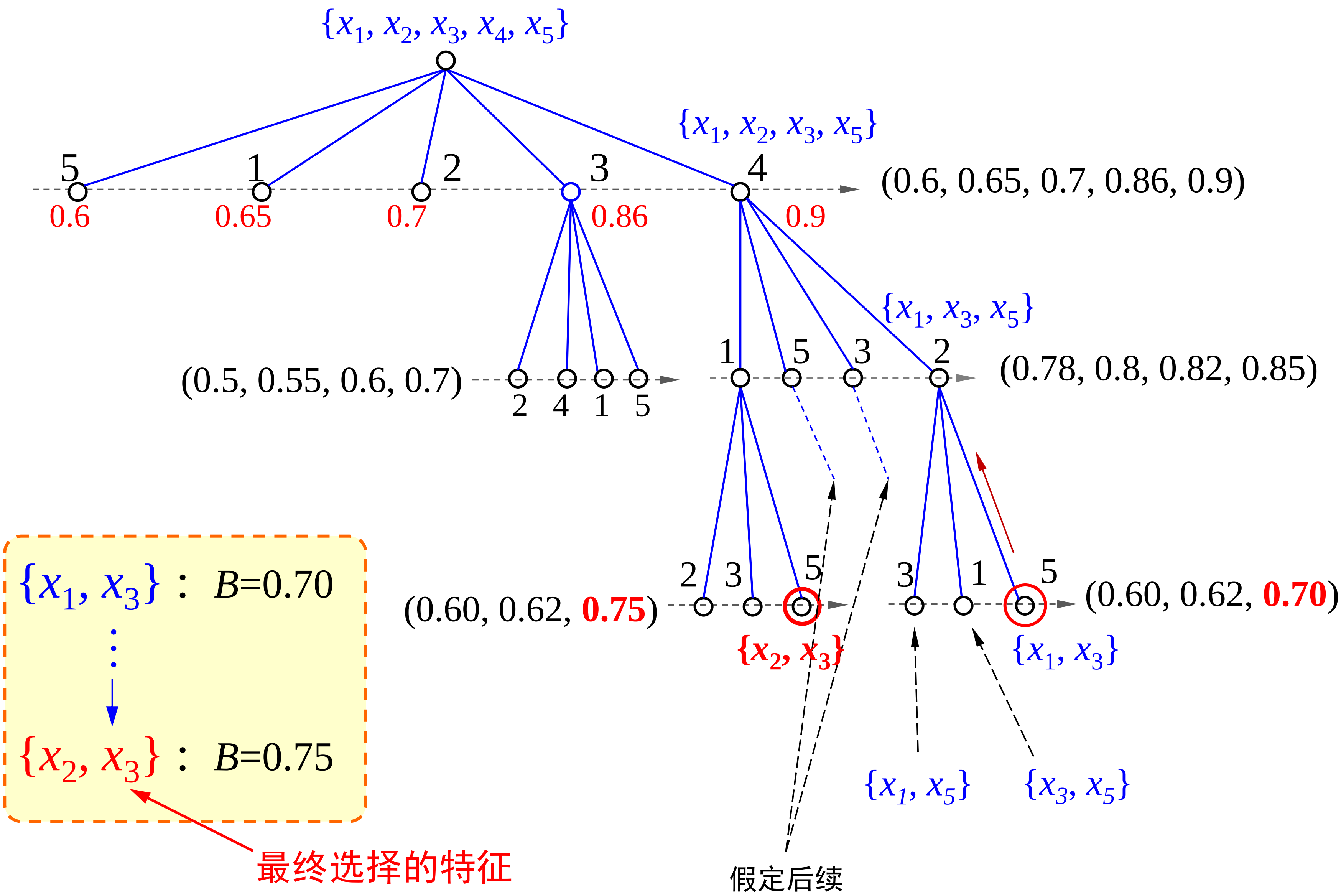
去掉第1特征后，判据值损失最大，即当前判据值J最小。
$\{x_1, x_2, x_3, x_4, x_5, x_6\}$
层1
层2
层3
层4
J降序
J 第2小
J 第3小
J 最大
J 增序
J 降序
J 局部增序
已去掉2, 3特征
$\{x_1, x_2, x_5, x_6\}$
$\{x_1, x_2, x_6\}$
$\{x_1, x_2\}$
最右往往无分枝
1 2 3
2 3 4 3 4 4
3 4 5 4 5 5 4 5 5 5
4 5 6 5 6 6 5 6 6 6 5 6 6 6 6



- 树的回溯：从树的最右侧开始，当到达叶子节点时向上回溯。每回溯一步将相应节点上舍弃的特征回收回来。
- 遇到最近的分枝节点时停止回溯，从该节点向下搜索左侧最近的一个分枝（从该分枝的右侧子枝开始）
- 在搜索到某一个节点时，准则函数值若小于界限B，则说明最优解不可能在本节点之下，停止沿本树枝的搜索，从此节点重新向上回溯。
- 若搜索到一个新的叶子节点，且准则函数值更大，则更新B的值，向上回溯。
- 最后一次更新B时，取得的特征组合则为特征选择的最后结果。

- 树的回溯的总体思路：对树搜索时进行分枝限界，从右到左、从上到下
- 算法步骤：
  - 1. 从某个节点开始向面对树进行回溯，直到遇见分枝节点，搜索分枝节点最右侧未处理的一个分枝
  - 2. 对于该分枝下每一层节点，计算对应特征集合的判据值V
  - 3. 如果$V < B$，根据判据的单调性，该节点以下的判据值都小于V，无需往下搜索，往上回溯，转到第1步；否则继续往下搜索，转第2步；若遇见叶节点，转第4步
  - 4. 计算叶节点对应特征集合的判据值B'，如果$B' > B$，更新B和相应的特征选择集合。转第1步；否则，算法终止（如果回溯过程中遇到根节点，且根据界限B不能再向下搜索其它树枝；或J值不能大于当前值为止）

示例：共有5个特征，从中选择2个



7.11 特征选择的次优方法

- 主要方法
  - 过滤式特征选择方法(Filter methods)
    - 单独特征选择法
    - 顺序前进特征选择法
    - 顺序后退特征选择法
    - 增 l 减 r 特征选择法
  - 包裹式特征选择法(Wrapper methods)
  - 嵌入式特征选择方法(Embedded methods)

次优算法：贪心策略

7.11.1 过滤式选择方法

- **基本思想：**
  - 过滤式方法先对数据集进行特征选择，再训练分类器。
- **核心任务：**
  - 如何定义特征的评价准则和搜索策略
- **算法特点：**
  - 特征选择过程与后续学习器无关；
  - 启发式特征选择方法，无法获得最优子集；
  - 与包裹式选择方法相比，计算量降低了很多。

7.11.1.1 单独特征选择法

- 基本假设
  - 特征相互独立，子集的性能等于其所包含的各个特征的性能之和。因此，单独作用时性能最优的特征，它们组合起来性能也是最优的

单独特征选择法

输入：数据集、特征集合、待选择特征个数 m

输出：选择的特征集合Φ

1. 计算每个特征的性能评价准则；
2. 根据特征的性能评价准则，对所有特征排序
3. 取前 m 个特征加入特征选择集合Φ

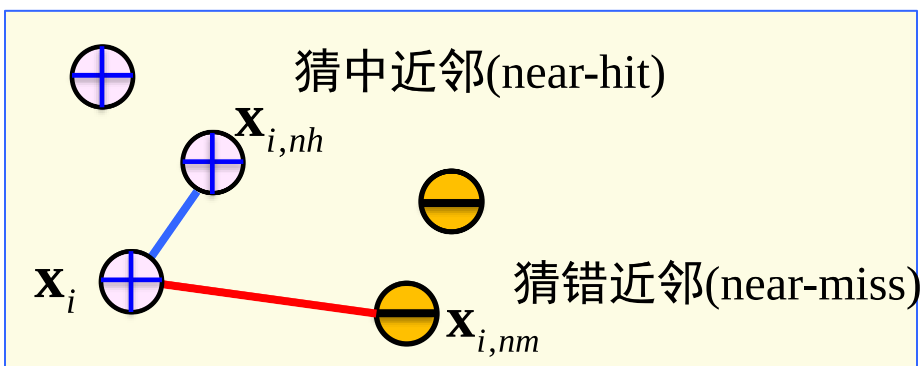
7.11.1.1 单独特征选择法：Relief 方法

- 设计了“相关统计量”来度量特征的重要性。考虑二类分类问题：
  - 对于样本$\mathbf{x}_i$，定义它的“猜中近邻”(nearest-hit)为其同类样本中的最近邻 $\mathbf{x}_{i,nh}$；定义它的“猜错近邻”(nearest-miss)为其不同类样本中的最近邻$\mathbf{x}_{i,nm}$。
  - Relief采用如下的特征性能判据：

$$\delta^j = \sum_{i=1}^{n} \operatorname{dis}\left(x_i^j, x_{i,nm}^j\right)^2 - \operatorname{dis}\left(x_i^j, x_{i,nh}^j\right)^2$$

表示第 i 个样本的第 j 个属性

度量样本属性差异：$\operatorname{dis}(x_a^j, x_b^j) = \left|x_a^j - x_b^j\right|$



✓ 若样本与其猜中(同类）近邻在属性 j 上的距离小于其猜错(类间)近邻的距离，则说明属性 j 对区分同类和异类样本有益。

7.11.1.1 单独特征选择法：Relief 方法

考虑多类问题：

- 对于样本$\mathbf{x}_i$，记“猜中近邻”(nearest-hit)为它在同类样本中的最近邻$\mathbf{x}_{i,nh}$；记“猜错近邻” (nearest-miss)为它在每个不同类样本中的最近邻$\mathbf{x}_{i,k,nm}$。多类Relief（Relief-F）采用如下的特征性能判据：

$$\delta^j = \sum_{i=1}^{n} \left\{ \left[\sum_{k \neq c(x_i)} P_k \times \operatorname{dis}\left(x_i^j, x_{i,k,nm}^j\right)^2 \right] - \operatorname{dis}\left(x_i^j, x_{i,nh}^j\right)^2 \right\}$$

第 k 类样本在数据集中所占的比例

- 优点：
  - 为加快计算速度，Relief只需在数据集的采样上而不是整个数据集上进行
  - 因此，Relief的时间开销随采样次数以及原始特征数线性增长，运行效率很高。

7.11.1.1 单独特征选择法：Relief 方法

Algorithm 1 Pseudo-code for the original Relief algorithm

n: number of training instances
d: number of features
m: number of random training instances out of n used to update δ

initialize all feature weights $\delta[j]:=0$
for $i:=1$ **to** m **do**
  randomly select a 'target' instance $\mathbf{x}_i$
  find a nearest hit $\mathbf{x}_{i,\,nh}$ and nearest miss $\mathbf{x}_{i,\,nm}$
  for $j := 1$ **to** d **do**
    $\delta[j] := \delta[j] - dis(\mathbf{x}_i^j, \mathbf{x}_{i,nh}^j) + dis(\mathbf{x}_i^j, \mathbf{x}_{i,nm}^j)$
  end for
end for
return the vector δ of feature scores that estimate the quality of features

7.11.1.2 顺序前进特征选择法

顺序前进特征选择法

输入：数据集、特征集合、待选择特征个数 m

输出：已选择特征集合 Φ

1. 计算每个特征的性能评价准则，选择最优的特征加入特征选择集合 Φ
2. 对于每个剩余特征，分别计算它**与已选择特征组合在一起的性能评价准则**
3. 根据评价准则，**选择最优的特征**加入特征选择集合 Φ
4. 重复2-3步，直到已选择特征数量达到预定数量

✓ 优点：相比单独特征选择法更鲁棒一些，考虑了一定的特征组合因素；计算速度依然很快。

✓ 缺点：特征一旦入选，就无法被剔除。

7.11.1.3 顺序后退特征选择法

顺序后退特征选择法

输入：数据集、特征集合、待选择特征个数 m

输出：已选择特征集合 Φ

1. 将所有特征加入特征选择集合 Φ
2. 对于已选择特征集合 Φ 中的每一个特征，计算去掉该特征后**剩余特征的性能评价准则**
3. 根据评价准则，**选择使得准则最优所对应的特征**，将其从特征选择集合 Φ 中去除
4. 重复2-3步，直到已选择特征数量达到预定数量

缺点：

- ✓ 自顶向下的方法，相对顺序前进法（自底向上），**计算量更大**，因为大部分计算都在高维空间（特征个数从最大逐渐较少）；
- ✓ 特征一旦剔除，就无法再加入。

7.11.1.4 前向-后向特征选择法

增 l 减 r 特征选择法（$l > r$）：

- 基本思想：
  - 为了使已选择或者剔除的特征有机会重新被考虑，将顺序前进特征选择法和顺序后退特征选择法相结合。
- 基本步骤：
  - 采用 l 次顺序前进特征选择，选择 l 个特征；
  - 对已选择特征集合，采用 r 次顺序后退特征选择；
  - 重复上述特征选择和剔除过程，直到选择到所需数目的特征。

注意：顺序前进步骤和顺序后退步骤可以对调，此时，$r > l$，对应减 r 增 l 法。

7.11.2 包裹式特征选择方法

- 过滤式特征选择方法：
  - 先对数据集进行特征选择，然后再训练分类器；**特征选择过程与分类单独进行**，特征选择评价判据间接反应分类性能。
- 包裹式特征选择方法：
  - 特征选择过程与分类性能相结合，特征评价判据为分类器性能。对给定分类方法，选择最有利于提升分类性能的特征子集。
  - 通常采用交叉验证来评价选取的特征子集的好坏
    - *K*折交叉验证(*k*-fold cross validation)， 留一法(Leave-one-out)
  - 包裹式特征选择方法对分类器的基本要求：
    - 分类器能够处理高维特征向量；
    - 特征维度很高、样本个数较少时，分类器依然可以取得较好的效果。

7.11.2 包裹式特征选择方法

- 主要方法：
  - 直观方法：给定特征子集，训练分类器模型，以分类器错误率为特征性能判据，进行特征选择。
    - 需要大量尝试不同的特征组合，计算量大。
  - 替代方法(递归策略)：首先利用所有的特征进行分类器训练，然后考查各个特征在分类器中的贡献，逐步剔除贡献小的特征。
    - 递归支持向量机（R-SVM：Recursive SVM）
    - 支持向量机递归特征剔除（SVM-RFE）
    - Adaboost

7.11.2 包裹式特征选择方法

- 直观方法 (General framework)：
  - 1. Initialize $F = \emptyset$.
  - 2. Repeat:
    - (a) For $i = 1, \ldots, d$ if $i \notin F$, let $F_i = F \cup \{i\}$, and use some version of cross validation to evaluate features F_i :
      - train your learning algorithm using only the features in F_i , and estimate its generalization error.
    - (b) Set F to be the best features (subset) found on step (a).
  - 3. Select and output the best feature subset that was evaluated during the entire search procedure.

可见：
- 启发式方法，无法保证得到最优子集
- 需频繁调用学习算法进行候选特征子集的评价
- 通常特征选择效果很好，但计算量很大

7.11.2 包裹式特征选择方法

- **替代方法的技术路线（以支持向量机为例）：**
  1. 用当前所有特征训练线性支持向量机。
  2. 评估每个特征在支持向量机中的相对贡献，按照相对贡献大小进行排序。
  3. 根据事先确定的递归选择特征的数目，选择出排序靠前的特征，用这组特征构成新特征。
  4. 重复1-3步，直到达到规定的特征选择数目。

- 特征选择的递归策略：
  - 每次选择固定比例的特征
  - 人为给定一个逐级减少的特征数目序列

7.11.2 包裹式特征选择方法

线性SVM的决策函数：

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b$$

N: 支撑向量的个数
$y_i \in \{-1,1\}$: $\mathbf{x}_i$的类别
α_i: SVM在对偶空间的参数

- ✓ R-SVM根据每个特征在数据上的分离程度定义特征的相对贡献。
- ✓ SVM-RFE根据每个特征对SVM预测误差的贡献定义特征的相对贡献。

7.11.2 包裹式特征选择方法

方法1：利用线性SVM的类分离度 (Recursive SVM)

$$\mathbf{m}^{+}=\frac{1}{n_1}\sum_{\mathbf{x}^{+}\in\omega_1}\mathbf{x}^{+},\quad \mathbf{m}^{-}=\frac{1}{n_2}\sum_{\mathbf{x}^{-}\in\omega_2}\mathbf{x}^{-},$$

$$S=\frac{1}{n_1}\sum_{\mathbf{x}^{+}\in\omega_1}f(\mathbf{x}^{+})-\frac{1}{n_2}\sum_{\mathbf{x}^{-}\in\omega_2}f(\mathbf{x}^{-})=\sum_{j=1}^{d}\boxed{w_j\left(m_j^{+}-m_j^{-}\right)}+\left(\frac{1}{n_1}-\frac{1}{n_2}\right)\times b$$

特征 j 的贡献：$s_j=w_j(m_j^{+}-m_j^{-})$

方法2：利用线性SVM的预测误差 (SVM-RFE)

$$r^{+}=\sum_{\mathbf{x}_i^{+}:\ \text{SVs in }\omega_1}\alpha_i\mathbf{x}_i^{+},\quad r^{-}=\sum_{\mathbf{x}_i^{-}:\ \text{SVs in }\omega_2}\alpha_i\mathbf{x}_i^{-}$$

$$S=f(\mathbf{r}^{+})-f(\mathbf{r}^{-})=\mathbf{w}^{T}\Big(\sum_{\mathbf{x}_i:\ \text{SVs in both classes}}\alpha_i y_i\mathbf{x}_i\Big)=\mathbf{w}^{T}\mathbf{w}$$

特征 j 的贡献：$s_j=w_j^2$

7.11.2 包裹式特征选择方法

方法3：利用核SVM的目标函数

核SVM对偶问题的目标函数值：

$$Q=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i\cdot\mathbf{x}_j)$$

去掉第k维特征，目标函数值：

$$Q(k)=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i^{(-k)}\cdot\mathbf{x}_j^{(-k)})$$

$\mathbf{x}_i^{(-k)}$ 表示第i个数据去掉第k维特征后的新数据。

$$DQ(k)=Q-Q(k)=\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j\left(K(\mathbf{x}_i\cdot\mathbf{x}_j)-K(\mathbf{x}_i^{(-k)}\cdot\mathbf{x}_j^{(-k)})\right)$$

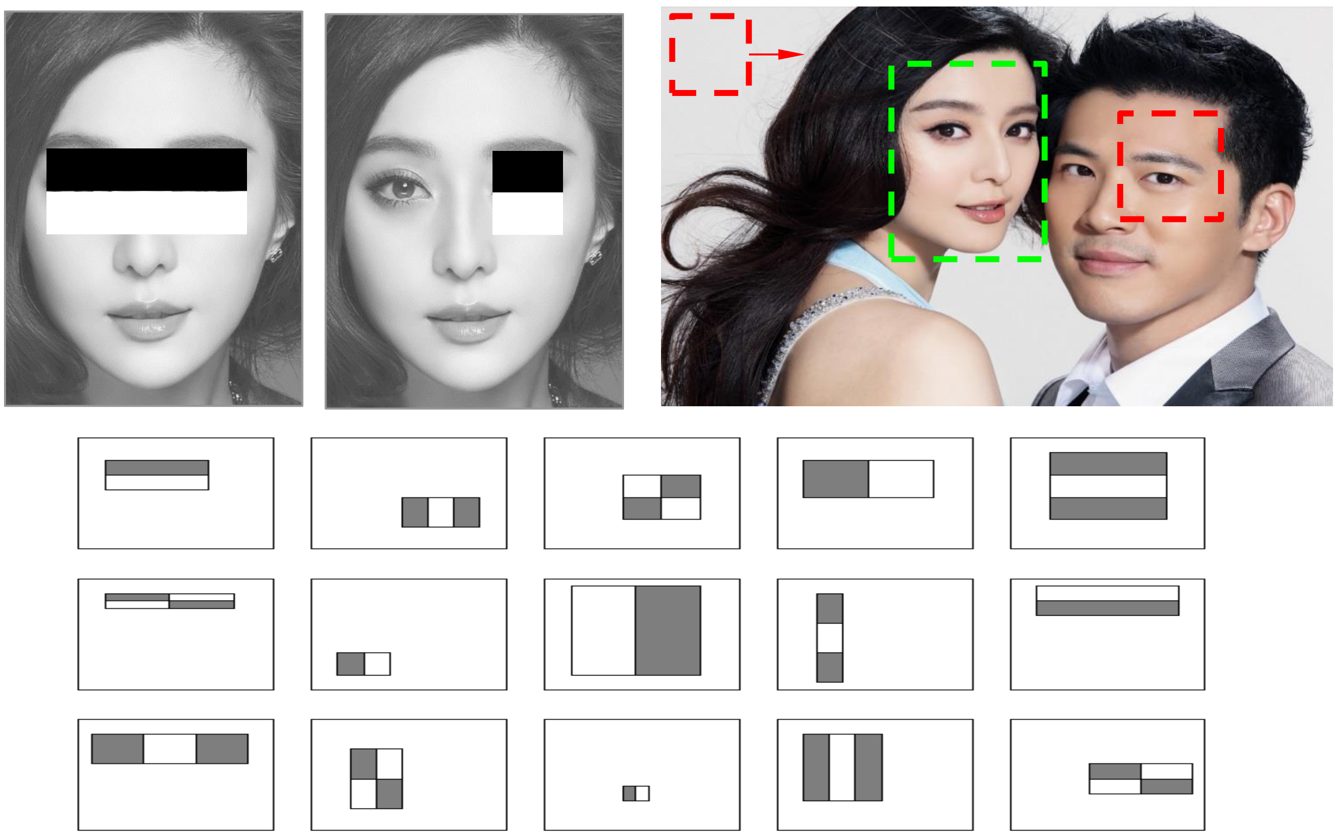
- **方法4：Adaboost for feature selection**：
  - Input: given $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_1), \ldots, (\mathbf{x}_n, y_n)\}, y_i \in\{1,-1\}$.
  - Output: The features ranked by their weights
    - Initialize: $\mathbf{W}_1(i) = 1/n$
    - for $t = 1, 2,\ldots,T$
      - Normalize the weights $\mathbf{W}_t$
      - $\forall$ **feature *j*** (or those in the remainder features), train weak classifier $h_j(\mathbf{x})$, and obtain its error e_j
      - Choose the classifier h_t with the lowest e_t
      - Set $\beta_t = e_t / (1-e_t)$, and $\alpha_t = -\log \beta_t$
      - Update the data distribution:

$$\mathbf{W}_{t+1}(i) = \begin{cases} \mathbf{W}_t(i)\beta_t, & \text{if } h_t(\mathbf{x}_i) = y_i \\ \mathbf{W}_t(i), & \text{otherwise} \end{cases}$$

    - end
    - Return: a strong classifier: $H(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{T} \alpha_t h_t(\mathbf{x})\right)$

7.11.2 包裹式特征选择方法

- **基于Haar特征的人脸检测**



Haar特征可很好地描述人脸眼部区域的灰度分布情况

7.12 嵌入式特征选择--基于L_1范数的特征选择

考虑线性分类方法：

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$$

- $w_i = 0$，第 i 个特征对分类没有影响
- $w_i \neq 0$，第 i 个特征属于有用特征

基本思路：在学习 $\mathbf{w}$ 的时候，对 $\mathbf{w}$ 进行限制，使其不仅能满足训练样本的误差要求，同时使得 $\mathbf{w}$ **中非零元素尽可能少**（只使用少数特征）。

7.12 嵌入式特征选择--基于L_1范数的特征选择

- **模型扩展：稀疏学习**
  - 向量稀疏性度量：
    - L_0：$\|\mathbf{w}\|_0$ =w中非零元素个数

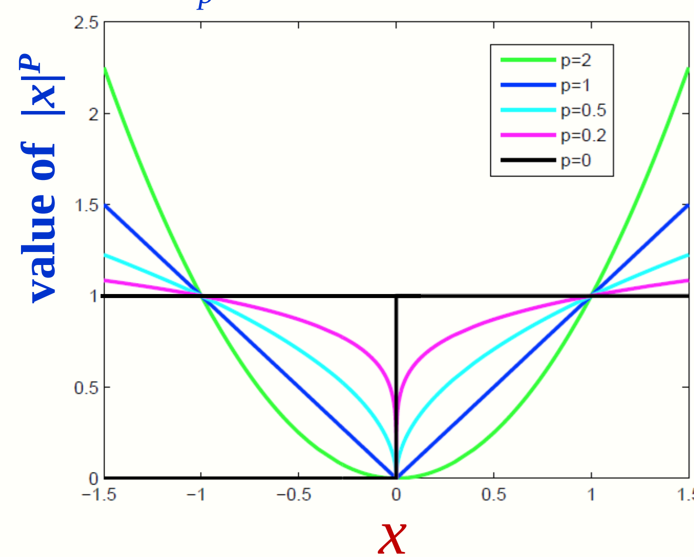
      例：w=[1.2, 0, -3, 0, 0]，$\|\mathbf{w}\|_0=2$
    - L_1：$\|\mathbf{w}\|_1 = \sum_i |w_i|$
    - 采用 L_1 来近似 L_0
  - 矩阵稀疏性度量：

    ⇨ $$\|\mathbf{W}\|_1 = \sum_{i,j} |W_{ij}|$$

向量的p范数：

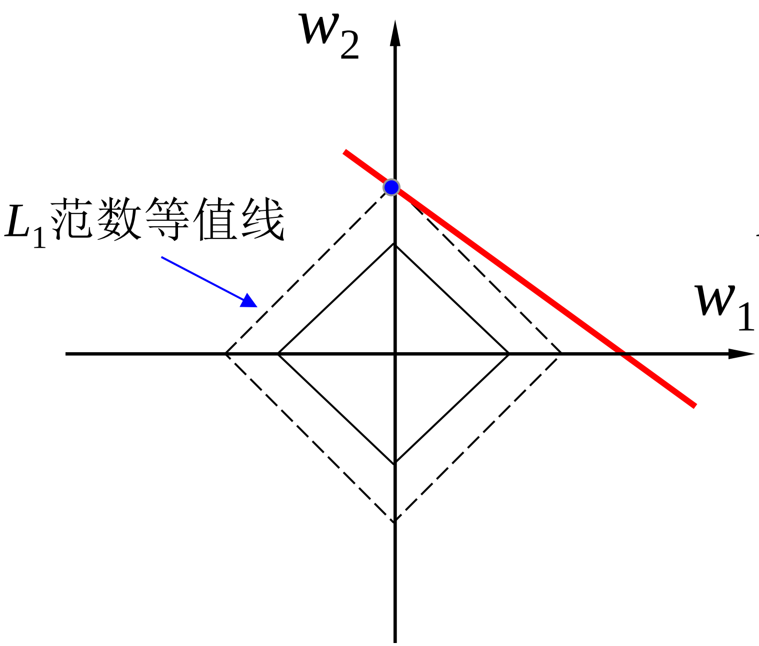
$$\|\mathbf{w}\|_p = \left(\sum_{i=1}^{m} |w_i|^p\right)^{\frac{1}{p}}$$

L_p范数示意图（1维）

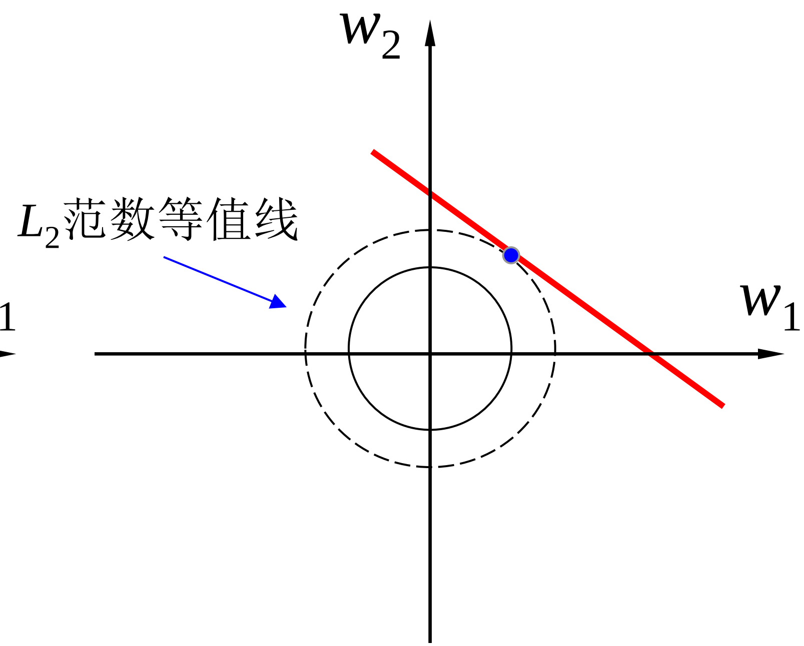


7.12 嵌入式特征选择--基于L_1范数的特征选择

- 什么是稀疏性？
  - 解向量的大部分位置值为零，只有少数部分位置的值不为零
- 最小化L_1范数可得到稀疏解



$$\min \ \|\mathbf{w}\|_1, \quad \text{s.t.} \quad \mathbf{w}^T\mathbf{x} = b$$



$$\min \ \|\mathbf{w}\|_2, \quad \text{s.t.} \quad \mathbf{w}^T\mathbf{x} = b$$

7.12 嵌入式特征选择--基于L_1范数的特征选择

- **LASSO的基本形式：**

$$\min_{\mathbf{w},b} \quad \frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{w}^T\mathbf{x}_i + b - y_i\right)^2, \quad \text{s.t.} \quad \|\mathbf{w}\|_1 \le t$$

  - 样本$\mathbf{x}_i$为 d 维向量
  - n : 样本个数
  - t : 指定的自由参数，用于控制正则化的程度

LASSO (Least Absolute Shrinkage and Selection Operation)

7.12 嵌入式特征选择--基于L_1范数的特征选择

- **将上式可进一步写成对应的向量形式**

$$\min_{\mathbf{w},b} \quad \frac{1}{n}\|\mathbf{X}\mathbf{w}+\mathbf{b}-\mathbf{y}\|_2^2, \quad \text{s.t.} \quad \|\mathbf{w}\|_1 \le t$$

分量全为b

其中，$X_{ij}=[\mathbf{x}_i]_j$

- 注意到给定 w 后，b 的优化有闭合解:

$$b=\bar{y}-\mathbf{w}^T\bar{\mathbf{x}}$$

- 将其带入原目标函数，进一步简化目标函数的形式。

7.12 嵌入式特征选择--基于L_1范数的特征选择

- **于是有：**

$$\min_{\mathbf{w}} \quad \frac{1}{n}\|\mathbf{X}\mathbf{w}-\mathbf{y}\|_2^2, \quad \text{s.t.} \quad \|\mathbf{w}\|_1 \le t$$

规范化：分别减去均值

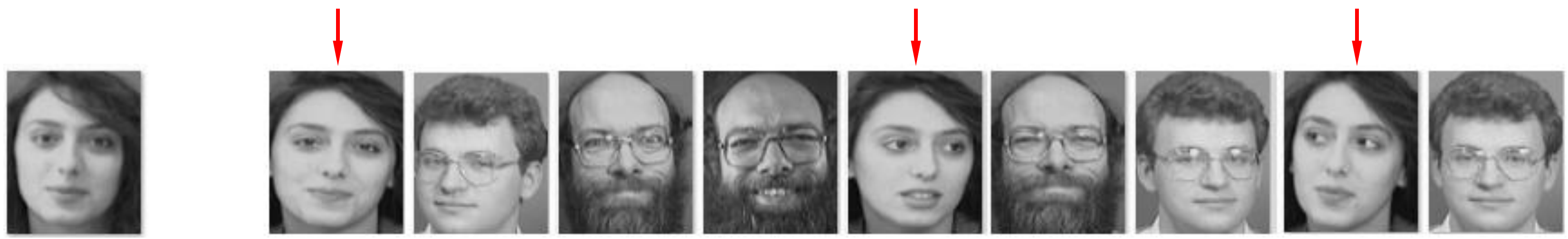
- 能够证明，上述最优化问题，可通过以下问题近似求解:

$$\min_{\mathbf{w}} \quad \frac{1}{n}\|\mathbf{X}\mathbf{w}-\mathbf{y}\|_2^2 + \lambda\|\mathbf{w}\|_1$$

LASSO (Least Absolute Shrinkage and Selection Operation)

7.12 嵌入式特征选择--基于L_1范数的特征选择

- **模型扩展: 从稀疏表示的角度来理解**



$$\mathbf{y} = 0.7\mathbf{x}_1 + 0.0\mathbf{x}_2 + 0.0\mathbf{x}_3 + 0.0\mathbf{x}_4 + 0.2\mathbf{x}_5 + 0.0\mathbf{x}_6 + 0.0\mathbf{x}_7 + 0.1\mathbf{x}_8 + 0.0\mathbf{x}_9$$

$$(\mathrm{p}_0) \quad \min_{\mathbf{w}} \|\mathbf{w}\|_0, \quad \text{subject to } \mathbf{Xw} = \mathbf{y}$$

$$(\mathrm{p}_0) \quad \min_{\mathbf{w}} \|\mathbf{Xw} - \mathbf{y}\|_2^2, \quad \text{subject to } \|\mathbf{w}\|_0 < t$$

L_1-范数松弛

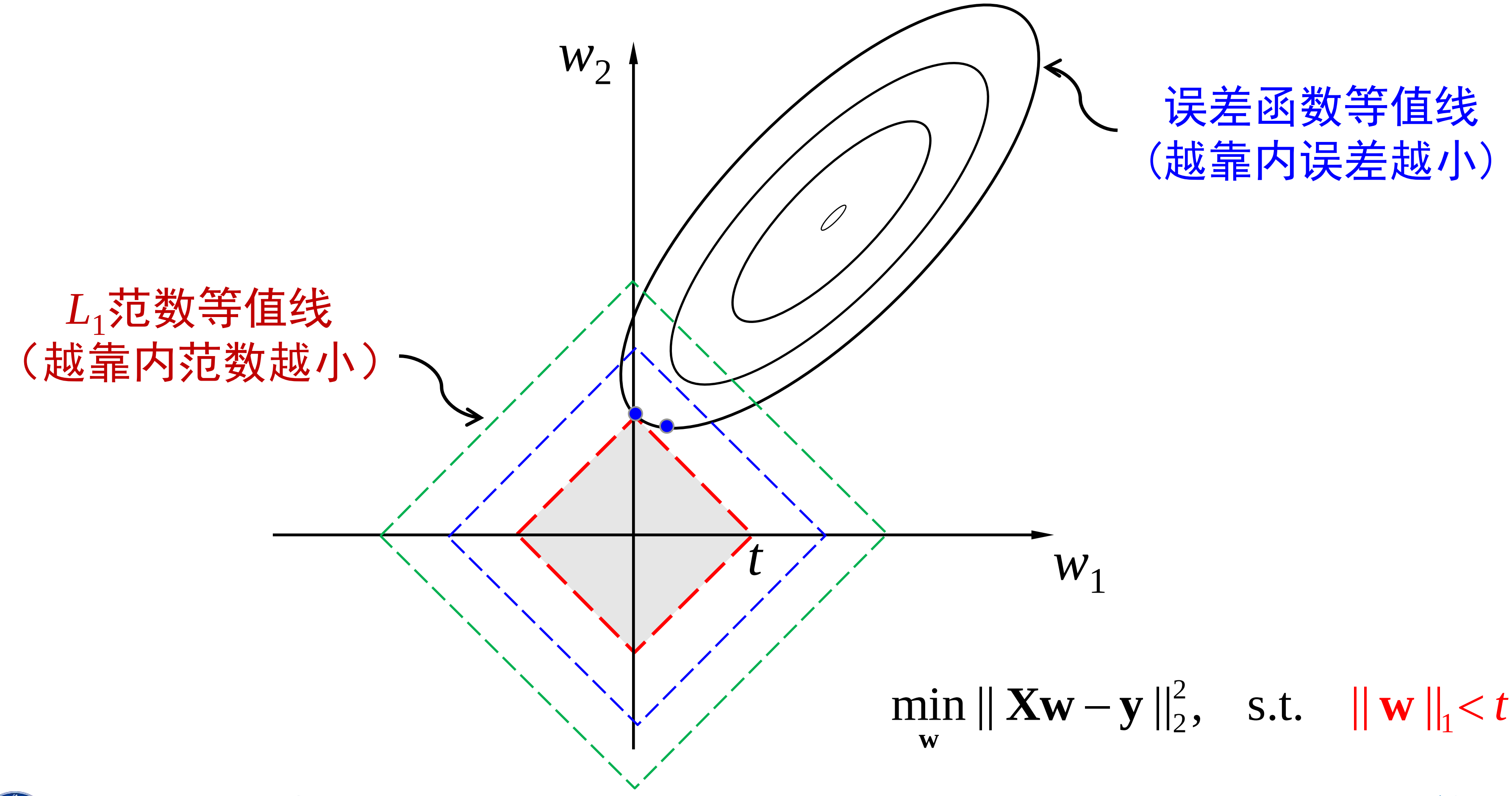
$$(\mathrm{p}_1) \quad \min_{\mathbf{w}} \|\mathbf{Xw} - \mathbf{y}\|_2^2, \quad \text{subject to } \|\mathbf{w}\|_1 < t$$

$$(\mathrm{p}_1) \quad \min_{\mathbf{w}} \|\mathbf{Xw} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{w}\|_1$$

LASSO算法!

7.12 嵌入式特征选择--基于L_1范数的特征选择

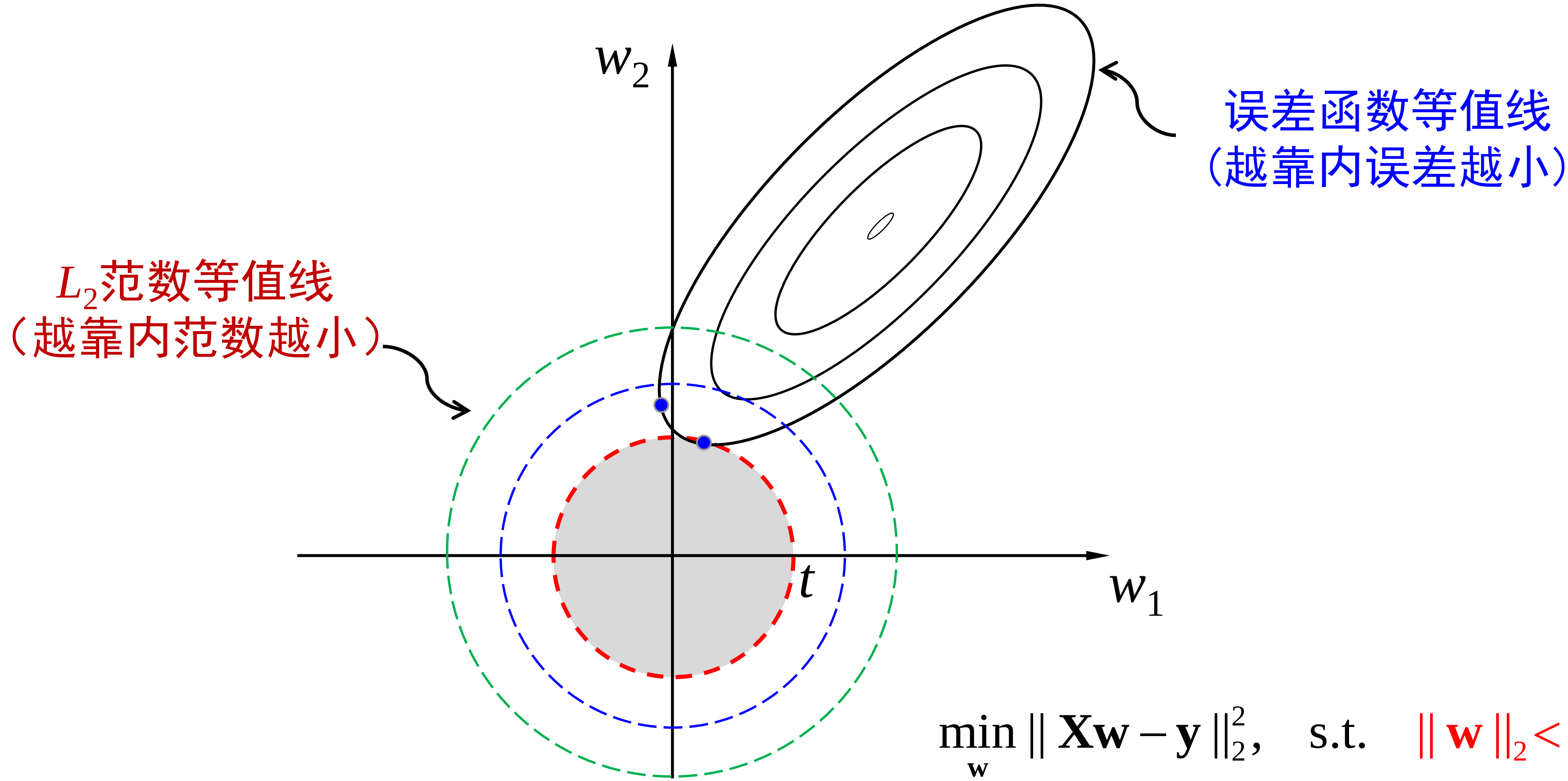
L_1范数 VS L_2范数：



$$\min_{\mathbf{w}} \|\mathbf{X}\mathbf{w}-\mathbf{y}\|_2^2, \quad \text{s.t.} \quad \|\mathbf{w}\|_1 < t$$

7.12 嵌入式特征选择--基于L_1范数的特征选择

L_1范数 VS L_2范数：



$$\min_{\mathbf{w}} \| \mathbf{X}\mathbf{w} - \mathbf{y} \|_2^2, \quad \text{s.t.} \quad \| \mathbf{w} \|_2 < t$$

7.12 嵌入式特征选择--基于L_1范数的特征选择

- 求解LASSO的几种常用方法：
  - 次梯度下降法(subgradient descent methods)
    - 梯度下降法的推广
  - 最小角度回归(least-angle regression, LARS)
    - 与LASSO模型密切相关
  - 近端梯度下降法(proximal gradient descent methods)
    - 目前非常流行，效果也是最好的
  - 半二次切分(half-quadratic splitting)

7.12 嵌入式特征选择--基于L_1范数的特征选择

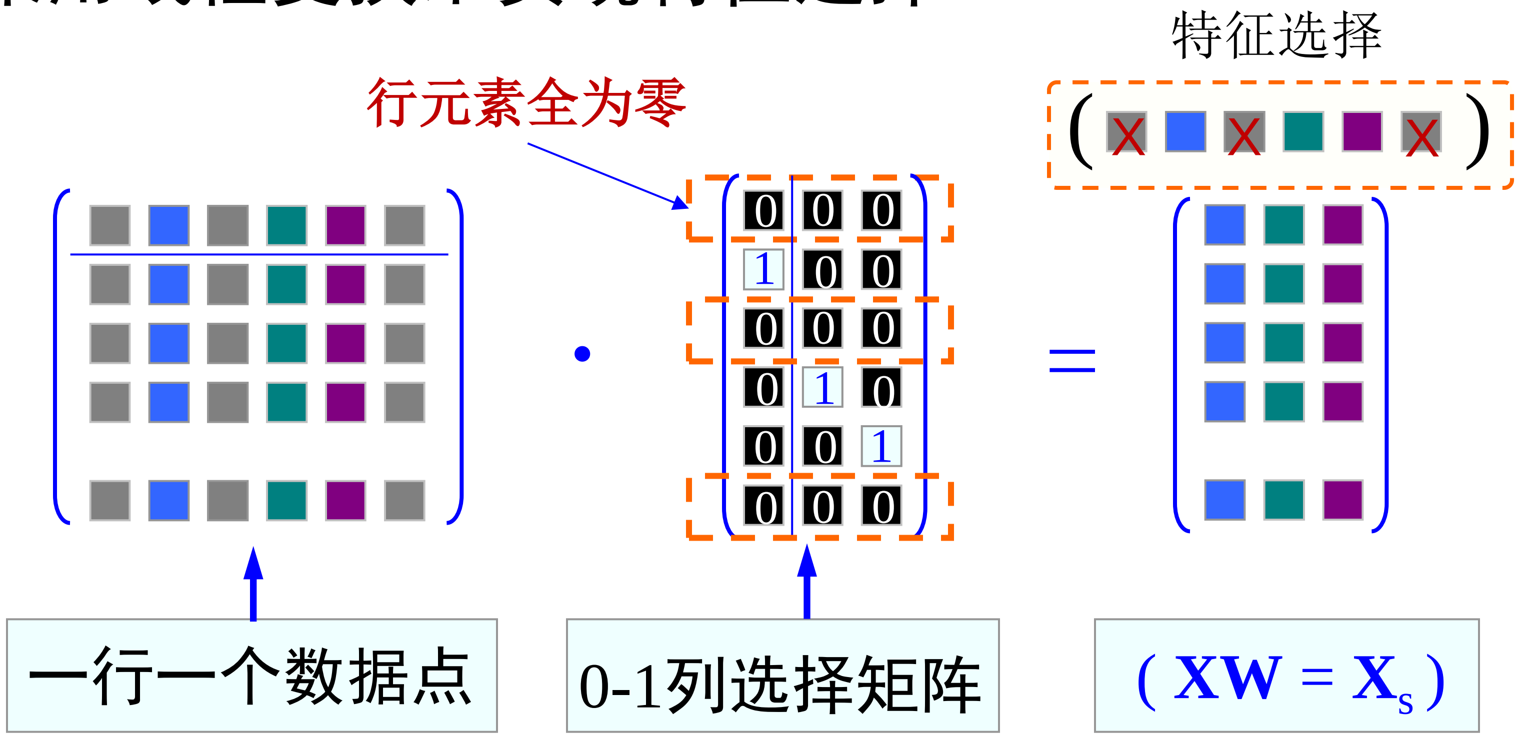
- 算法总体特征
  - 不能直接设置最终选择特征的个数 m；
  - 通过设置正则化系数 λ 来隐式控制 m；
  - λ 值越大，模型越关注稀疏性，得到的非零系数个数越少；反之，非零稀疏个数越多；
  - 可以设置一个选择特征个数的上限，通过设置不同 λ 值，得到满足要求的特征。
  - 是一种嵌入式特征选择方法：将分类器学习与特征选择融为一体，分类器训练过程自动完成了特征选择。

7.13 基于稀疏学习的特征选择

- 稀疏学习
  - 针对具体的学习问题，可在线性模型中引入恰当的稀疏约束条件或稀疏性度量。
    - 稀疏是一种先验（比如：服从拉普拉斯分布）。
    - 稀疏是对某种已知知识的描述。
    - 从结构化风险最小化的角度，引入稀疏约束条件是增加所学函数在假设空间的简单性，所学系数向量越稀疏，则函数越简单。
    - 从正则化的角度看，就是为了防止过拟合，提高线性最小二乘法所学模型的泛化能力。
    - ……

7.13 基于稀疏学习的特征选择

- 采用线性变换来实现特征选择

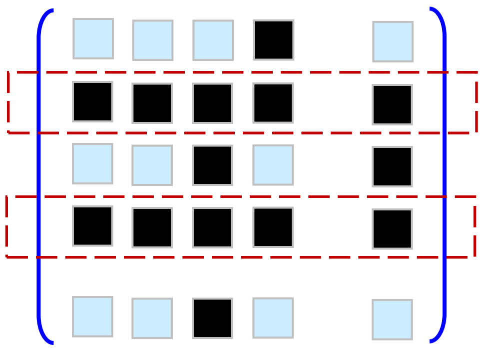


若列选择矩阵第i行全为零，则第i个特征分量不起作用！

7.13 基于稀疏学习的特征选择

$$\mathbf{W} = \begin{pmatrix} W_{11} & \cdots & W_{1m} \\ \vdots & \ddots & \vdots \\ W_{d1} & \cdots & W_{1m} \end{pmatrix}$$

- 矩阵行稀疏性度量：结构化稀疏

W= 

选择矩阵

$$w_1 = \sqrt{\sum_j |W_{1j}|^2}$$

$$w_2 = \sqrt{\sum_j |W_{2j}|^2}$$

$$\vdots$$

$$w_d = \sqrt{\sum_j |W_{dj}|^2}$$

$$\mathbf{w} = [w_1, w_2, \cdots, w_d]^T \in R^d$$

要求W的某行为零，只需要该行元素的平方和为零。因此，可以将行平方和开根号收集为一个向量，再考虑其零范数

$\|\mathbf{w}\|_0$ **is NP hard！ So we soft it as its L_1 norm** $\|\mathbf{w}\|_1 \Rightarrow \|\mathbf{W}\|_{2,1}$

7.13 基于稀疏学习的特征选择

- **矩阵的$L_{2,1}$范数：**

$$\|\mathbf{W}\|_{2,1} = \|\mathbf{w}\|_1 = \sum_{i=1}^{d} \sqrt{\sum_j |W_{ij}|^2}$$

$$\mathbf{W} = \begin{pmatrix} W_{11} & \cdots & W_{1m} \\ \vdots & \ddots & \vdots \\ W_{d1} & \cdots & W_{1m} \end{pmatrix}$$

  – The $L_{2,1}$ norm of matrix is a true norm

$$d(\mathbf{X}, \mathbf{Y}) = \|\mathbf{X} - \mathbf{Y}\|_{2,1}$$

满足：自反性、非负性、对称性和三角不等式关系

- 矩阵的$L_{p,r}$范数(伪范数)：

$$\|W\|_{p,r} = \left(\sum_{i=1}^{n} \left(\sum_{j=1}^{m} |w_{ij}|^p \right)^{\frac{r}{p}} \right)^{\frac{1}{r}}$$

7.13 基于稀疏学习的特征选择

- **回顾：正则化线性回归**
  - 线性模型： $\mathbf{y} = \mathbf{W}^T\mathbf{x} + \mathbf{b}$, where $\mathbf{y} \in R^c$, $\mathbf{W} \in R^{d\times c}$, $\mathbf{b} \in R^c$
  - 任务：对n个样本$\{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\}$，希望：

$$\mathbf{X}\mathbf{W} + \mathbf{e}_n\mathbf{b}^T \approx \mathbf{Y}, \quad \text{where } \mathbf{X} \in R^{n\times d}, \ \mathbf{Y} \in R^{n\times c}, \ \mathbf{e}_n \in [1,\ldots,1] \in R^n$$

  - 参数学习：最小化“正则化的回归误差”

回归误差

$$\min_{\mathbf{W},\mathbf{b}} \sum_{i=1}^{n} \left\| \mathbf{W}^T\mathbf{x}_i + \mathbf{b} - \mathbf{y}_i \right\|_F^2 + \lambda \|\mathbf{W}\|_F^2 = \boxed{\left\| \mathbf{X}\mathbf{W} + \mathbf{e}_n\mathbf{b}^T - \mathbf{Y} \right\|_F^2} + \boxed{\lambda \|\mathbf{W}\|_F^2}$$

正则项

$$\text{where } \mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n]^T = \begin{pmatrix} x_{11} & \cdots & x_{1d} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nd} \end{pmatrix} \in R^{n\times d}$$

（每一行为一个样本）

7.13 基于稀疏学习的特征选择

线性回归模型

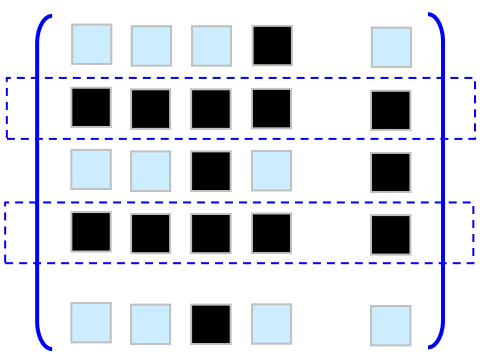
$$\min_{\mathbf{W},\mathbf{b}} \left\| \mathbf{XW} + \mathbf{e}_n\mathbf{b}^T - \mathbf{Y} \right\|_F^2 + \lambda \| \mathbf{W} \|_F^2$$

- **学习模型:**

$$\min_{\mathbf{W},\mathbf{b}} \left\| \mathbf{XW} + \mathbf{e}_n\mathbf{b}^T - \mathbf{Y} \right\|_F^2 + \lambda \| \mathbf{W} \|_{2,1}$$

- 最后如何实现特征选择的目标？— 排序

W=



选择矩阵

$$w_1 = \sqrt{\sum_j |W_{1j}|^2}$$

$$w_2 = \sqrt{\sum_j |W_{2j}|^2}$$

$$w_d = \sqrt{\sum_j |W_{dj}|^2}$$

$$\mathbf{w} = [w_1, w_2, \cdots, w_d]^T \in R^d$$

Shiming Xiang, et al., Discriminative Least Squares Regression for Multiclass Classification and Feature Selection. IEEE Transactions on Neural Network and Learning System,2012

7.14 小结

特征选择的一般技术路线：

1. 确定特征子集
2. 评价特征子集性能

评价特征子集性能常用的可分性判据：

- ✓ 基于类内类间距离的可分性判据
- ✓ 基于熵的可分性判据
- ✓ 基于SVM模型的可分性判据

7.14 小结

确定特征选择子集的方法：

- 基于树的方法（**最优算法**）：基于分枝限界技术对特征子集的树表示进行遍历，只需要查找一小部分特征组合，即可找到全局最优的特征组合
- 遍历法（**次优算法**）：顺序前向法、顺序后退法、增l减r法
- 稀疏约束

根据特征选择与分类器的结合程度：

- 过滤式特征选择方法：“选择”与“学习”独立
- 包裹式特征选择方法：“选择”依赖“学习”
- 嵌入式特征选择方法：“选择”与“学习”同时进行

致谢

- PPT由向世明老师提供

Thank All of You!

(Questions?)

张燕明

ymzhang@nlpr.ia.ac.cn

people.ucas.ac.cn/~ymzhang

模式分析与学习课题组(PAL)

中科院自动化研究所· 模式识别国家重点实验室