第5章

降维与度量学习

Dimensionality Reduction and Metric Learning

赫 然
rhe@nlpr.ia.ac.cn
https://rhe-web.github.io/

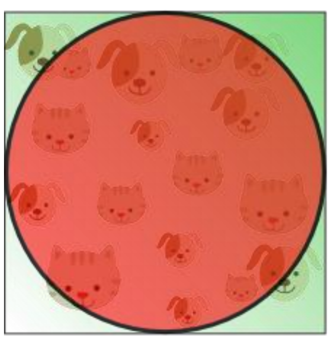
智能感知与计算研究中心(CRIPAC)
中科院自动化研究所 模式识别国家重点实验室

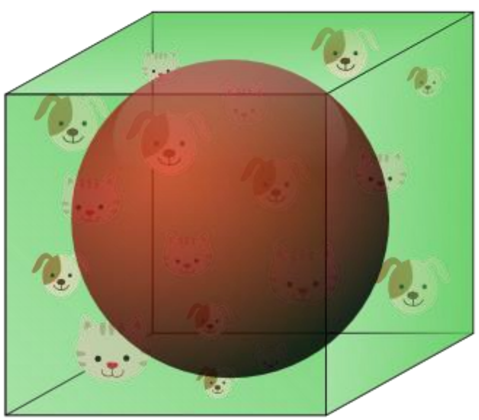
内容提要

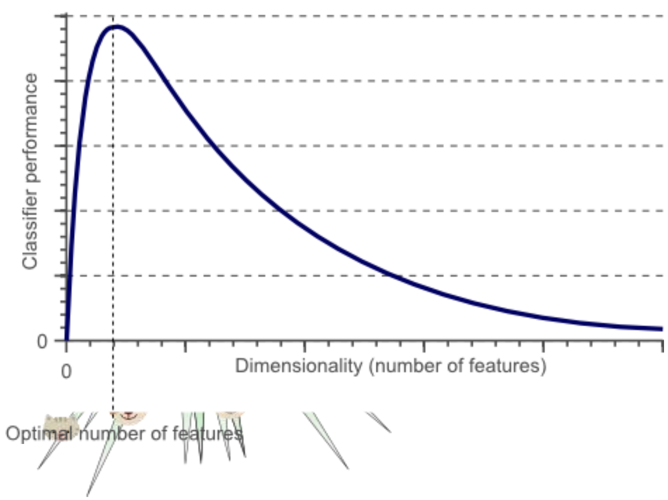
- **<u>引言</u>**
- 主成分分析
- 多维缩放
- 流形学习方法
- 距离度量学习

引言

- 维度灾难（Curse of Dimensionality）
  - 维数灾难最早由理查德·贝尔曼（Richard E. Bellman）在考虑优化问题时提出的，它用来描述当空间维度增加时分析和组织高维空间中的数据会遇到各种问题。







引言

- 维度灾难（Curse of Dimensionality）
  - 随着维数的增加，**计算量**呈指数倍增长。
  - 随着维数的增加，具有相同距离的两个样本其相似程度可以相差很远。
  - 当维度增加时，空间的体积增加得很快，可用数据变得稀疏。
  - 稀疏性对于任何要求“具有统计学意义的方法”而言都是一个问题。但是，为了获得在统计学上正确并且有可靠的结果，用来支撑这一结果所需要的数据量通常随着维数的增加而呈指数级增长。

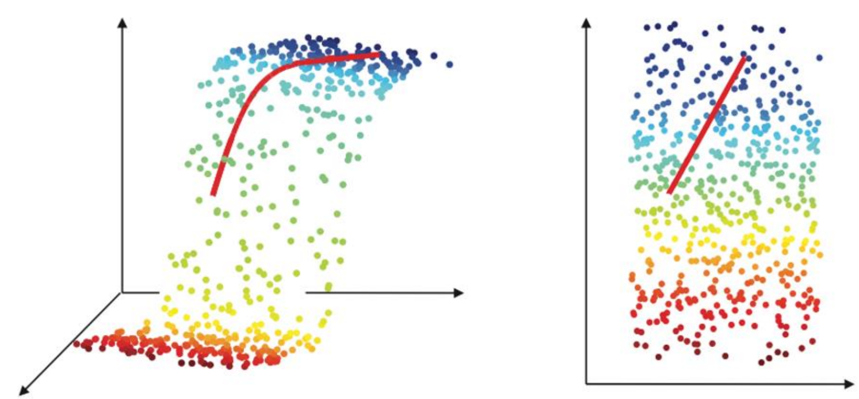
引言

- 维度缩减
  - 缓解维数灾难的一个重要途径是降维，即通过某种数学变换将原始高维特征空间变换至某个低维“子空间”。在该子空间中，样本密度大幅度提高，距离计算也变得更为容易。
  - **为什么能降维**：
    - 在很多时候，人们观测或收集到的数据虽然是高维的，但与学习任务密切相关的特征通常位于某个低维分布上，即高维空间中的一个低维“嵌入”（embedding）。

引言

● 维度缩减

- 缓解维数灾难的一个重要途径是降维，即通过某种数学变换将原始高维特征空间变换至某个低维“子空间”。在该子空间中，样本密度大幅度提高，距离计算也变得更为容易。
- **为什么能降维**：



(a) 三维空间中观察到的样本点　(b) 二维空间中的曲面

内容提要

- 引言
- **<u>主成分分析</u>**
- 多维缩放
- 流形学习方法
- 距离度量学习

主成分分析(Principle Component Analysis)

- 线性降维法
  - 对高维空间中的样本x进行线性变换：

$$y = W^T x \quad x \in R^m, W \in R^{m \times d}, y \in R^d, d < m$$

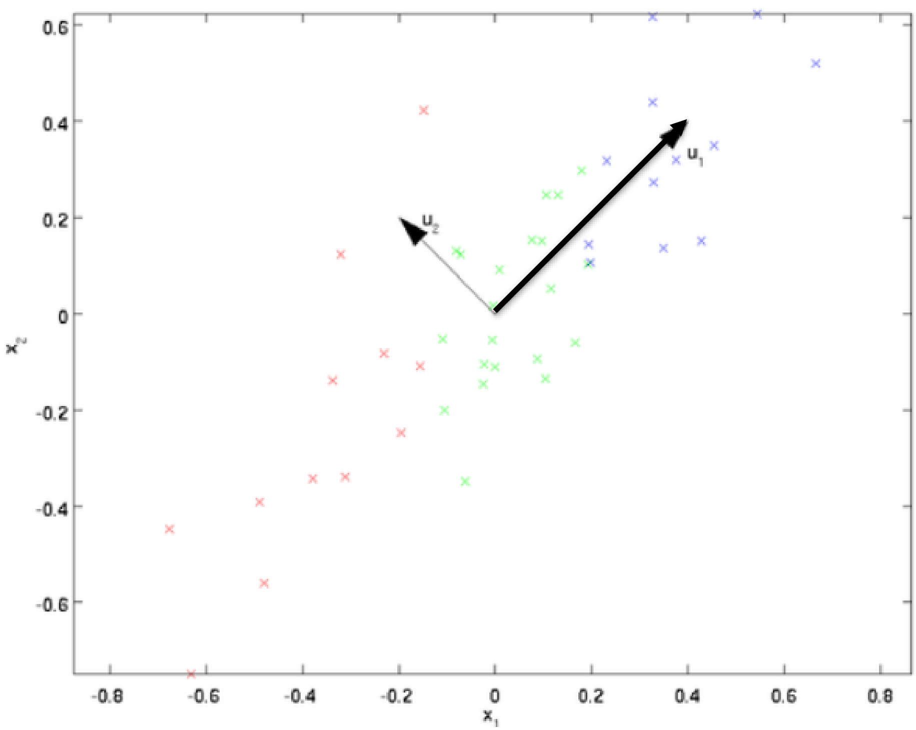
  - 变换矩阵 $W = [w_1, w_2, ..., w_d]$ 可视为m维空间中由d个基向量组成的矩阵。

  - $y = W^T x$ 可视为样本 x 与 d 个基向量分别做内积运算而得到，即 x 在新坐标系下的坐标。显然，新空间中的特征是原空间中特征的线性组合。

  - 不同方法的不同之处在于对低维子空间的性质有不同的要求，即对W施加不同的约束。

主成分分析

- PCA（Principal Component Analysis）基本思想
  - 如何仅用一个超平面从整体上对所有样本 $\{x_1, x_2, \ldots, x_n\} \in R^m$ 进行恰当的表示？



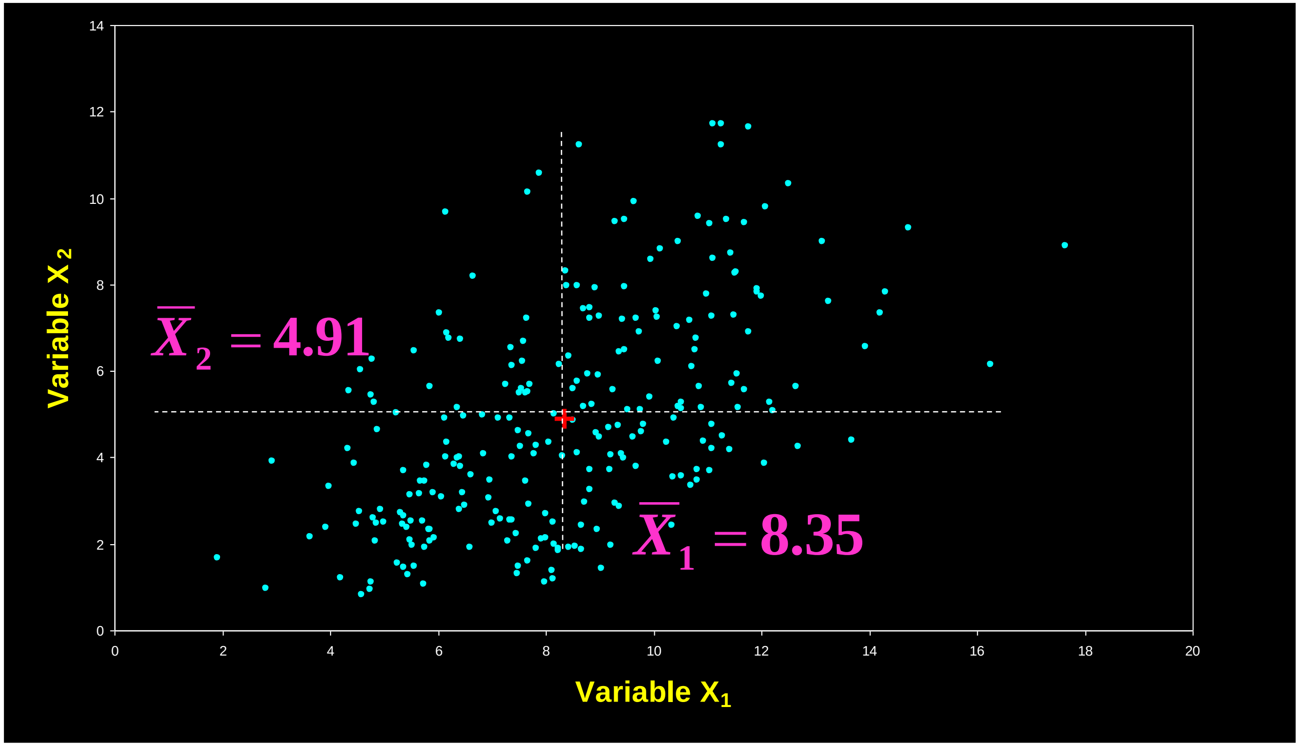
PCA是一种统计方法。通过正交变换将一组可能存在相关性的变量转换为一组线性不相关的变量，转换后的这组变量叫主成分。

主成分分析

- PCA（Principal Component Analysis）基本思想
  - 我们如何仅用一个超平面从整体上对所有样本 $\{x_1, x_2, \ldots, x_n\} \in R^m$ 进行恰当的表示？
  - 通常有如下两种思路：
    - 可重构性：样本到这个超平面的距离都足够近；基于最小投影距离。
    - 可区分性：样本点在这个超平面上的投影能够尽可能地分开。基于最大投影方差。

主成分分析1- Minimum Error Formulation

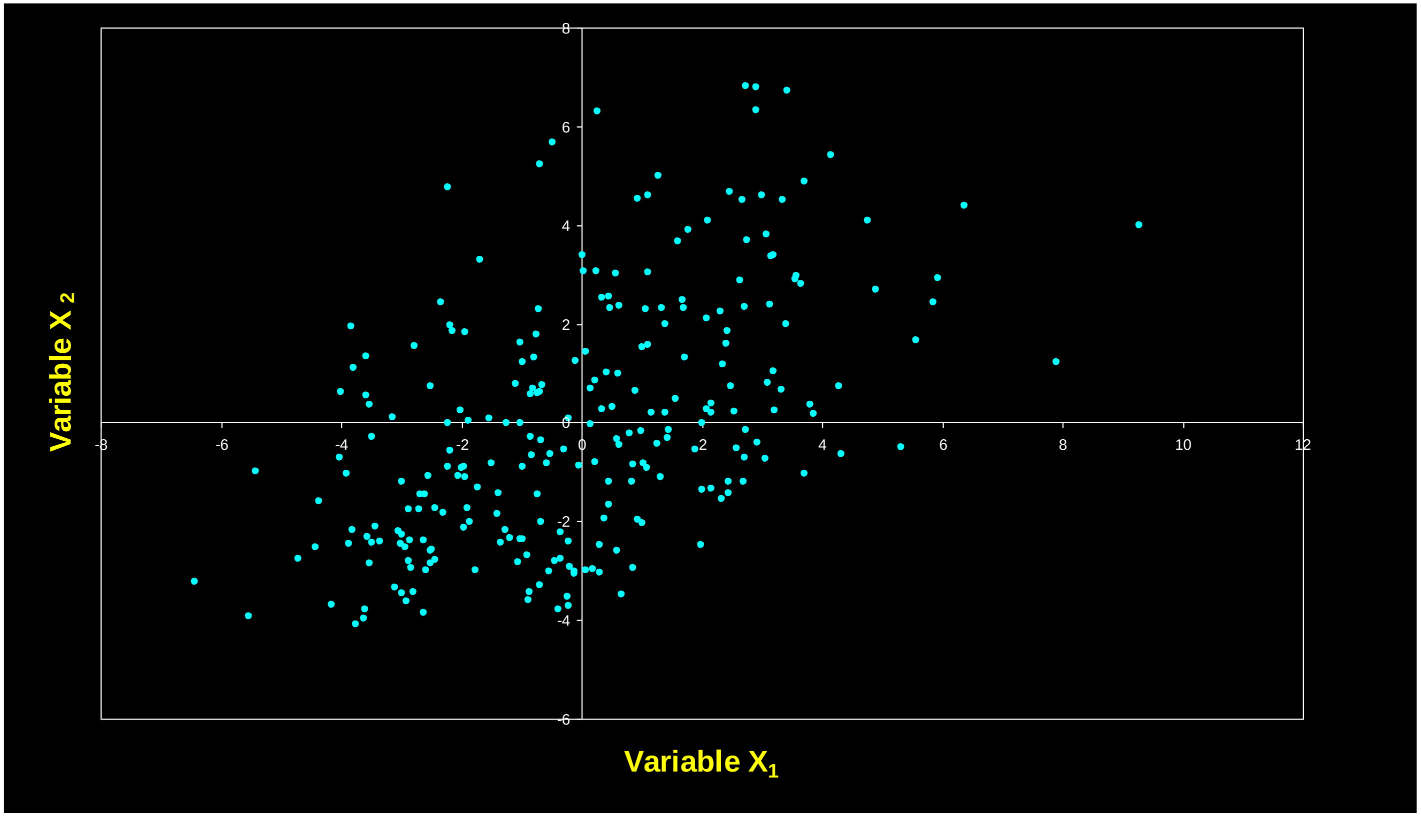
- m维空间的完备正交基组(complete orthogonormal set), 即新坐标系由$W = [w_1, w_2, \dots, w_m]$, 且



$V_1 = 6.67$ $V_2 = 6.24$ $C_{1,2} = 3.42$

主成分分析1- Minimum Error Formulation

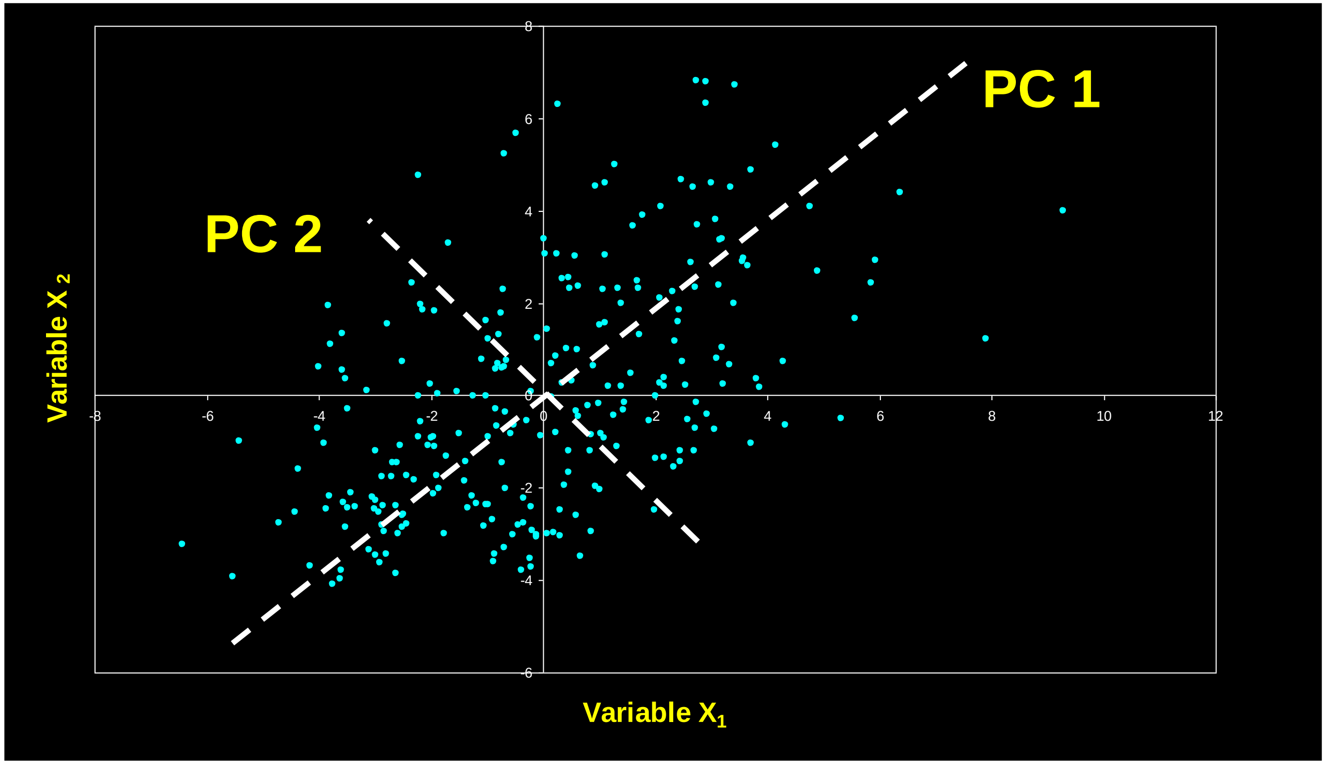
- m 维空间的完备正交基组(complete orthogonormal set), 即新坐标系由$W = [w_1, w_2, \ldots, w_m]$, 且



Reference: https://www.mii.lt/zilinskas/uploads/visualization/lectures/...pca/PCA1.ppt

主成分分析1- Minimum Error Formulation

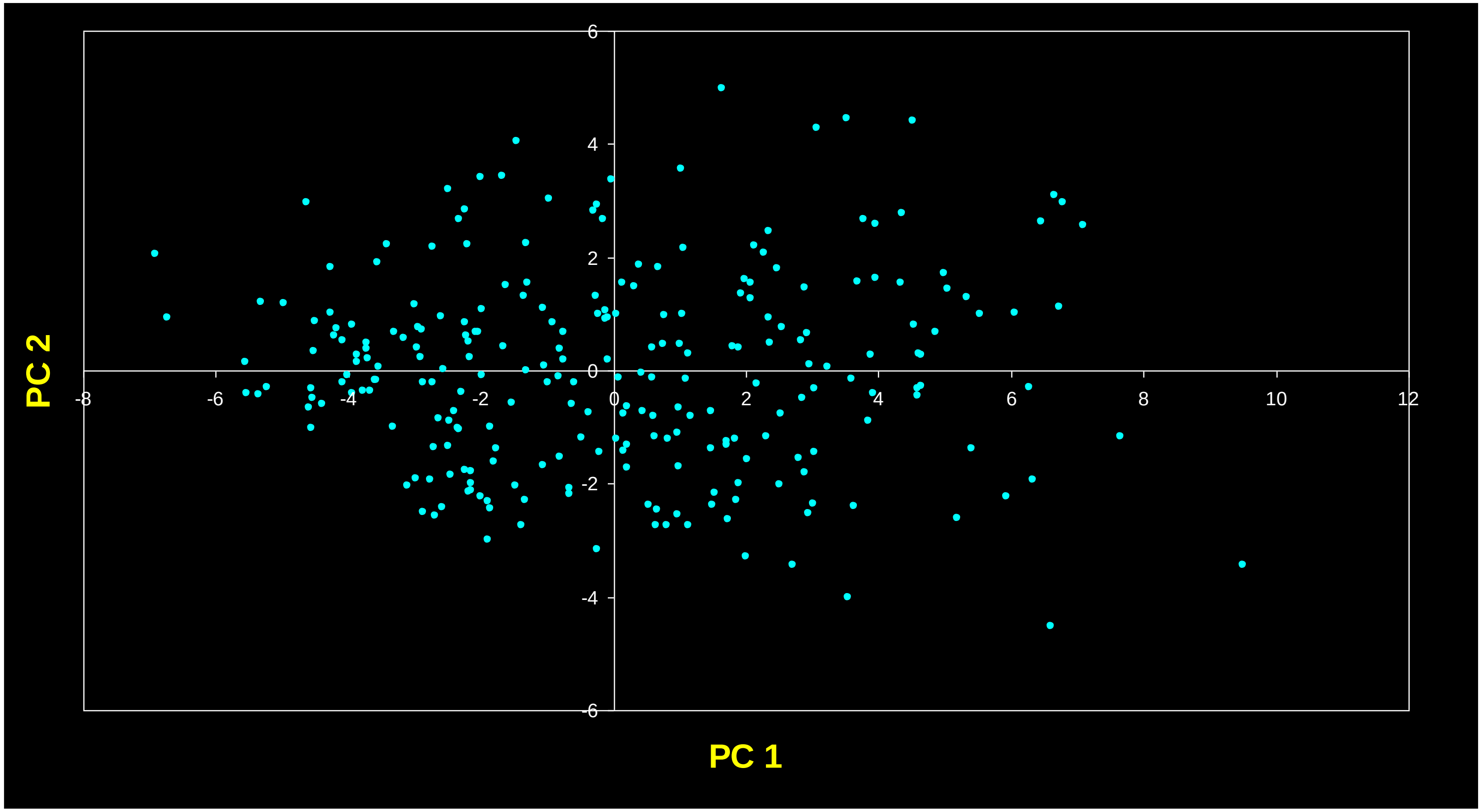
- m维空间的完备正交基组(complete orthogonormal set), 即新坐标系由$W = [w_1, w_2, \ldots, w_m]$, 且



Reference: https://www.mii.lt/zilinskas/uploads/visualization/lectures/...pca/PCA1.ppt

主成分分析1- Minimum Error Formulation

- m维空间的完备正交基组(complete orthogonormal set), 即新坐标系由$W = [w_1, w_2, \dots, w_m]$, 且



Reference: https://www.mii.lt/zilinskas/uploads/visualization/lectures/...pca/PCA1.ppt

主成分分析1- Minimum Error Formulation

–核心思想是通过旋转坐标系以最小化误差(projection error)

–m维空间的完备正交基组(complete orthogonormal set)，即新坐标系由$W = [w_1, w_2, \ldots, wm]$，且

$$w_k^T w_j = \delta_{kj} = \begin{cases} 1, k = j \\ 0, k \neq j \end{cases}$$

$$\Downarrow$$

$$x_i = \sum_{j=1}^{m} y_{ij} w_j$$

主成分分析1- Minimum Error Formulation

$$x_i = \sum_{k=1}^{m} y_{ik} w_k$$

$$w_j^T x_i = \sum_{k=1}^{m} w_j^T y_{ik} w_k \Rightarrow w_j^T x_i = w_j^T y_{ij} w_j \Rightarrow y_{ij} = x_i^T w_j$$

$$x_i = \sum_{j=1}^{m} (x_i^T w_j) w_j$$

主成分分析1- Minimum Error Formulation

- 根据我们对已有数据的理解，我们知道有的冗余信息较多(high redundant information)，有的信息冗余信息较少(low redundant information)。我们希望达到的目的是将两者分开，因此认为将投影后的估计值进行线性拆分

$$\boxed{\tilde{x}_i = \sum_{j=1}^{\underline{d}} z_{ij} w_j + \sum_{j=\underline{d+1}}^{m} b_j w_j}$$

- 其中b_j只跟维度相关

主成分分析1- Minimum Error Formulation

- 目标函数（objective function，loss function）

$$J=\frac{1}{n}\sum_{i=1}^{n}\left\|x_i-\boxed{\tilde{x}_i}\right\|_2^2$$

$$=\frac{1}{n}\sum_{i=1}^{n}\left\|x_i-\boxed{\sum_{j=1}^{\mathrm{d}}z_{ij}w_j-\sum_{j=d+1}^{m}b_jw_j}\right\|_2^2$$

- 对b_j求导并令其为零，得到

$$b_j=\overline{x}^Tw_j\,,\quad j=d+1,d+2,\ldots,m \qquad \overline{x}=\frac{1}{n}\sum_{i=1}^{n}x_i$$

Christopher M Bishop . Pattern Recognition And Machine Learning.

主成分分析1- Minimum Error Formulation

- 对z_{ij}求导得到，

$$\frac{\partial J}{\partial z_{ij}} = \frac{\partial J}{\partial \tilde{x}_i}\frac{\partial \tilde{x}_i}{\partial z_{ij}} = \frac{\partial J}{\partial \tilde{x}_i}\frac{\partial}{\partial z_{ij}}\left(\sum_{k=1}^{d} z_{ik} w_k\right) \qquad \frac{\partial J}{\partial \tilde{x}_i} = -\frac{2}{n}(x_i - \tilde{x}_i)^T$$

$$= -\frac{2}{n}(x_i - \tilde{x}_i)^T w_j$$

$$= -\frac{2}{n}\left(x_i - \sum_{k=1}^{d} z_{ik} w_k\right)^T w_j = -\frac{2}{n}(x_i^T w_j - z_{ij})$$

- 令其为零，我们得到

$$z_{ij} = x_i^T w_j, \quad j = 1, 2, \ldots, d$$

主成分分析1- Minimum Error Formulation

- 将z_{ij}和b_j表达式分别带入$x_i - \tilde{x}_i$，得到，

$$x_i - \tilde{x}_i = \sum_{j=1}^{m} (x_i^T w_j) w_j - (\sum_{j=1}^{\mathrm{d}} z_{ij} w_j + \sum_{j=d+1}^{m} b_j w_j)$$

$$= \sum_{j=1}^{m} (x_i^T w_j) w_j - \sum_{j=1}^{\mathrm{d}} (x_i^T w_j) w_j - \sum_{j=d+1}^{m} (\overline{x}^T w_j) w_j$$

$$= \sum_{j=d+1}^{m} ((x_i - \overline{x})^T w_j) w_j$$

- 带入到目标函数J中

主成分分析1- Minimum Error Formulation

- 目标函数J 可重写为之关于w_j的形式，

$$J=\frac{1}{n}\sum_{i=1}^{n}\{[\sum_{j=d+1}^{m}(\boxed{(x_i-\overline{x})^T w_j})w_j]^T[\sum_{j=d+1}^{m}((x_i-\overline{x})^T w_j)w_j]\}$$

$$=\frac{1}{n}\sum_{i=1}^{n}\{\sum_{j=d+1}^{m}(x_i^T w_j-\overline{x}^T w_j)^2 w_j^T w_j\}$$

$$=\frac{1}{n}\sum_{i=1}^{n}\sum_{j=d+1}^{m}(x_i^T w_j-\overline{x}^T w_j)^2$$

$$=\sum_{j=d+1}^{m} w_j^T S w_j \qquad S=\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})(x_i-\bar{x})^T$$

主成分分析1- Minimum Error Formulation

- 目标函数J 可重写为之关于w_j的形式，

$$\min J = \min \frac{1}{n}\sum_{i=1}^{n}\left\|x_i - \tilde{x}_i\right\|_2^2$$

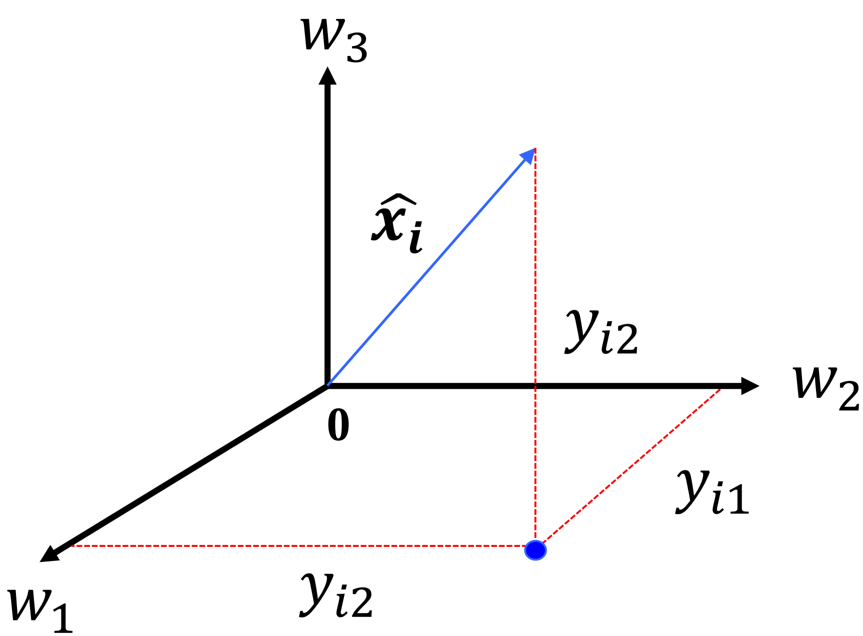
$$= \min_{w_j} \sum_{j=d+1}^{m} w_j^T S w_j = \min_{W} tr(W^T S W)$$

- w_j为协方差矩阵S的最小特征值对应的特征向量

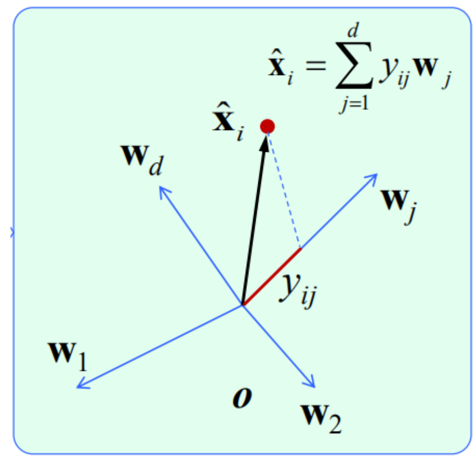
$$S = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^T \qquad \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

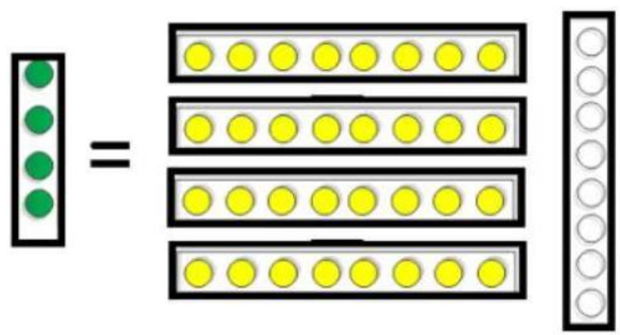
主成分分析2- 重构角度

– 向量空间中一个矢量的表示方法（零均值）



$$\widehat{x}_i = w_1 y_{i1} + w_2 y_{i2} + w_3 y_{i3}$$





$$y = W^T x$$

主成分分析2- 重构角度

– 由W定义新坐标系：假定投影变换是正交变换，即新坐标系由$W=[w_1,w_2,\ldots,w_d]$来表示$(d<m)$，w_i的模等于1，w_i与w_j两两正交。

– 设样本点x_i在新坐标系下的坐标为：$y_i=[y_{i1},y_{i2},\ldots,y_{id}]^T\in R^d$

– 在正交坐标系下，对样本点x_i，有新坐标：

$$y_{ij}=w_j^T x_i, w_j\in R^m \quad j=1,2,\ldots,d$$

– 在新坐标系下，可得x_i的新表示：

$$\hat{x}_i=\sum_{j=1}^{d} y_{ij} w_j \qquad i=1,2,\ldots,n$$

主成分分析

重构误差

$$\sum_{i=1}^{n} ||x_i - \widehat{x_i}\,||_2^2 = \sum_{i=1}^{n} ||x_i - \sum_{j=1}^{d} y_{ij} w_j||_2^2 = \sum_{i=1}^{n} ||x_i - W y_i||_2^2$$

$x \in R^m$

$W \in R^{m*d}, y_i \in R^d$

主成分分析

重构误差

$$\sum_{i=1}^{n}||x_i - \widehat{x}_i||_2^2 = \sum_{i=1}^{n}||x_i - \sum_{j=1}^{d} y_{ij}w_j||_2^2 = \sum_{i=1}^{n}||x_i - Wy_i||_2^2$$

$$= \sum_{i=1}^{n}((Wy_i)^T Wy_i - 2x_i^T Wy_i + x_i^T x_i)$$

主成分分析

重构误差

$$\sum_{i=1}^{n} ||x_i - \widehat{x_i}\,||_2^2 = \sum_{i=1}^{n} ||x_i - \sum_{j=1}^{d} y_{ij} w_j||_2^2 = \sum_{i=1}^{n} ||x_i - W y_i||_2^2$$

$$= \sum_{i=1}^{n} ((W y_i)^T W y_i - 2x_i^T W y_i + x_i^T x_i)$$

$\because W^T W = I$

$$= \sum_{i=1}^{n} (y_i^T y_i - 2y_i^T y_i + x_i^T x_i)$$

主成分分析

重构误差

$$\sum_{i=1}^{n} ||x_i - \hat{x}_i||_2^2 = \sum_{i=1}^{n} ||x_i - \sum_{j=1}^{d} y_{ij} w_j||_2^2 = \sum_{i=1}^{n} ||x_i - W y_i||_2^2$$

$$= \sum_{i=1}^{n} ((W y_i)^T W y_i - 2x_i^T W y_i + x_i^T x_i)$$

$\because W^T W = I$

$$= \sum_{i=1}^{n} y_i^T y_i - 2y_i^T y_i + x_i^T x_i$$

$\because y_i = W^T x_i$

常量

$$= -\sum_{i=1}^{n} y_i^T y_i + \boxed{const} = -\sum_{i=1}^{n} (W^T x_i)^T (W^T x_i) + const$$

主成分分析

重构误差

$$\sum_{i=1}^{n} ||x_i - \hat{x}_i\,||_2^2 = \sum_{i=1}^{n} ||x_i - \sum_{j=1}^{d} y_{ij} w_j||_2^2 = \sum_{i=1}^{n} ||x_i - W y_i||_2^2$$

$$= \sum_{i=1}^{n} ((W y_i)^T W y_i - 2 x_i^T W y_i + x_i^T x_i)$$

$\because W^T W = I$

$$= \sum_{i=1}^{n} (y_i^T y_i - 2 y_i^T y_i + x_i^T x_i)$$

$\because y_i = W^T x_i$

$$= -\sum_{i=1}^{n} y_i^T y_i + const = -\sum_{i=1}^{n} (W^T x_i)^T (W^T x_i) + const$$

$$= -tr\left(\sum_{i=1}^{n} (W^T x_i)^T (W^T x_i)\right) + const$$

主成分分析

重构误差

$$\sum_{i=1}^{n}||x_i-\hat{x}_i||_2^2=\sum_{i=1}^{n}||x_i-\sum_{j=1}^{d}y_{ij}w_j||_2^2=\sum_{i=1}^{n}||x_i-Wy_i||_2^2$$

$$=\sum_{i=1}^{n}((Wy_i)^TWy_i-2x_i^TWy_i+x_i^Tx_i)$$

$\because W^TW=I$

$$=\sum_{i=1}^{n}(y_i^Ty_i-2y_i^Ty_i+x_i^Tx_i)$$

$\because y_i=W^Tx_i$

$$=-\sum_{i=1}^{n}y_i^Ty_i+const=-\sum_{i=1}^{n}(W^Tx_i)^T(W^Tx_i)+const$$

$$=-tr\left(\sum_{i=1}^{n}(W^Tx_i)^T(W^Tx_i)\right)+const$$

$\because tr(AB)=tr(BA)$

$\because tr(A)+tr(B)=tr(A+B)$

$$=-tr\left(W^T\sum_{i=1}^{n}x_ix_i^TW\right)+const$$

主成分分析2- 重构角度

– 进一步，假定数据已经零均值化，$\sum_{i=1}^{n} x_i = 0$ 即

$X = [x_1, x_2, \ldots, x_n] \in R^{m \times n}$

$$\sum_{i=1}^{n} \left\| x_i - \hat{x}_i \right\|_2^2 = -tr\left(W^T \sum_{i=1}^{n} x_i x_i^T W \right) + const$$

$$= \text{-}tr\left(W^T X X^T W \right) + const$$

主成分分析2- 重构角度

– 进一步，假定数据已经零均值化，$\sum_{i=1}^{n} x_i = 0$ 即

$X = [x_1, x_2, \ldots, x_n] \in R^{m \times n}$

$$\sum_{i=1}^{n} \|x_i - \hat{x}_i\|_2^2 = -tr\left(W^T \sum_{i=1}^{n} x_i x_i^T W\right) + const$$

$$= -tr\left(W^T X X^T W\right) + const$$

于是，获得主成分分析的最优化模型:

主成分分析2- 重构角度

– 进一步，假定数据已经零均值化，$\sum_{i=1}^{n} x_i = 0$ 即

$$X = [x_1, x_2, \ldots, x_n] \in R^{m \times n}$$

$$\sum_{i=1}^{n} \|x_i - \hat{x}_i\|_2^2 = -tr\left(W^T \sum_{i=1}^{n} x_i x_i^T W\right) + const$$

$$= -tr\left(W^T XX^T W\right) + const$$

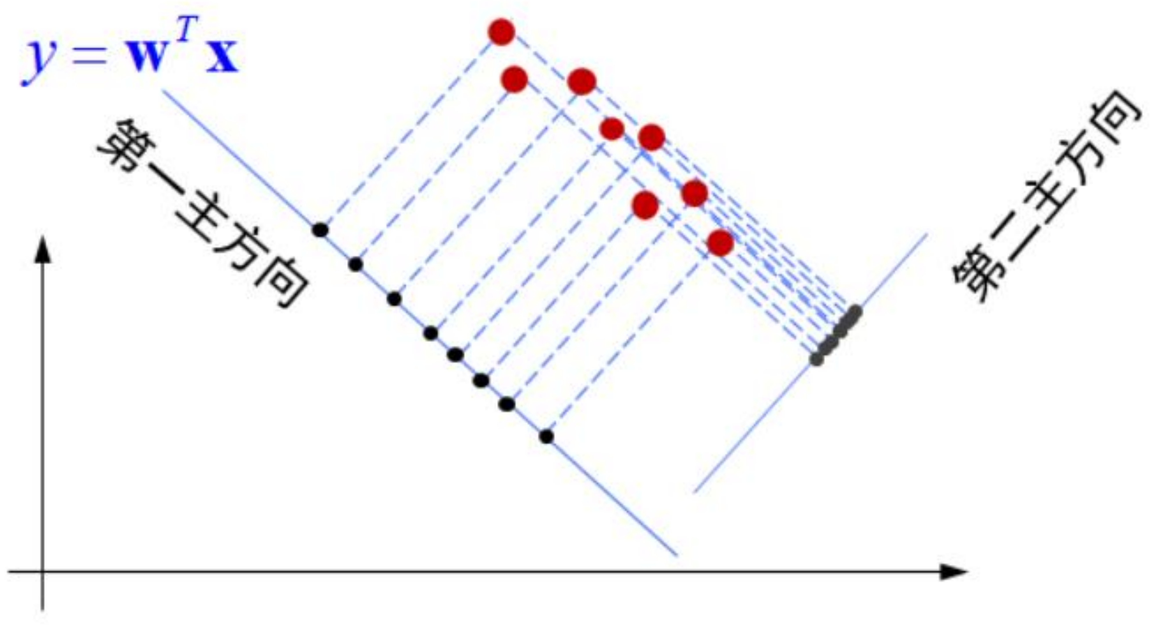
于是，获得主成分分析的最优化模型:

$$\max_{W \in R^{m \times d}} tr\left(W^T XX^T W\right) + const, \quad s.t. W^T W = I$$

主成分分析3- 可区分角度

– 使所有样本点的投影尽可能地分开，则需最大化投影点的方差，投影后获得的样本点为：

$$y_i = \mathrm{w}^T x_i \in R^d, i = 1, 2, \ldots, n$$



主成分分析3- 可区分角度

– 使所有样本点的投影尽可能地分开，则需最大化投影点的方差, 投影后获得的样本点为：

$$y_i = \mathrm{w}^T x_i \in R^d, i = 1, 2, \ldots, n$$

–由于数据点是零均值化的，则:

$$\sum_{i=1}^{n} y_i = \mathrm{w}^T \sum_{i=1}^{n} x_i = 0$$

主成分分析3- 可区分角度

– 使所有样本点的投影尽可能地分开，则需最大化投影点的方差, 投影后获得的样本点为：

$$y_i = \mathrm{w}^T x_i \in R^d, i = 1, 2, \ldots, n$$

–由于数据点是零均值化的，则:

$$\sum_{i=1}^{n} y_i = \mathrm{w}^T \sum_{i=1}^{n} x_i = 0$$

– 因此投影后的样本点的（协）方差为:

$$\sum_{i=1}^{n} w^T x_i x_i^T w = w^T X X^T w$$

主成分分析3- 可区分角度

要使数据具有最大可分性，就应该使数据尽量分散开来，因此应该使其**方差最大**。

$$\max_{W \in R^{m \times d}} \mathrm{w}^T XX^T w \qquad s.t. w^T w = 1$$

主成分分析

- PCA**求解**
  - 采用拉格朗日乘子法，经过简单矩阵运算，我们有：

$$XX^T w = \lambda w$$

$$J(W) = w^T XX^T w + \lambda(1 - w^T w)$$

w求导 $$XX^T w - \lambda w = 0$$

$$\boldsymbol{XX^T w = \lambda w}$$

主成分分析

- PCA**求解**
  - 采用拉格朗日乘子法，经过简单矩阵运算，我们有：

$$XX^T w = \lambda w$$

则主成分分析的解为：

主成分分析

- PCA**求解**
  - 采用拉格朗日乘子法，经过简单矩阵运算，我们有：

$$XX^Tw = \lambda w$$

则主成分分析的解为：

对协方差矩阵 XX^T 进行特征值分解，并对特征值进行排序 $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq ... \geq \lambda_d$ 取前 d 个特征值对应的特征向量构成变换矩阵 W。

主成分分析

- PCA**求解**
  - 采用拉格朗日乘子法，经过简单矩阵运算，我们有：

$$XX^T w = \lambda w$$

则主成分分析的解为：

> 对协方差矩阵 XX^T 进行特征值分解，并对特征值进行排序 $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq ... \geq \lambda_d$ 取前 d 个特征值对应的特征向量构成变换矩阵 W。

λ_1 对应的特征向量称为第一主成分，其他依此类推。

主成分分析

- PCA**求解**

PCA算法步骤

输入样本集 $D=\{x_1,x_2,...,x_m\}$，维数 d'

1 对所有样本进行中心化：$x_i \leftarrow x_i - \frac{1}{m}\sum_{i=1}^{m} x_i$

2 计算样本的**协方差矩阵** XX^T

3 对协方差矩阵 XX^T 做特征分解

4 取最大的 d' 个特征值对应的特征向量 $w_1,w_2,...,w_{d'}$

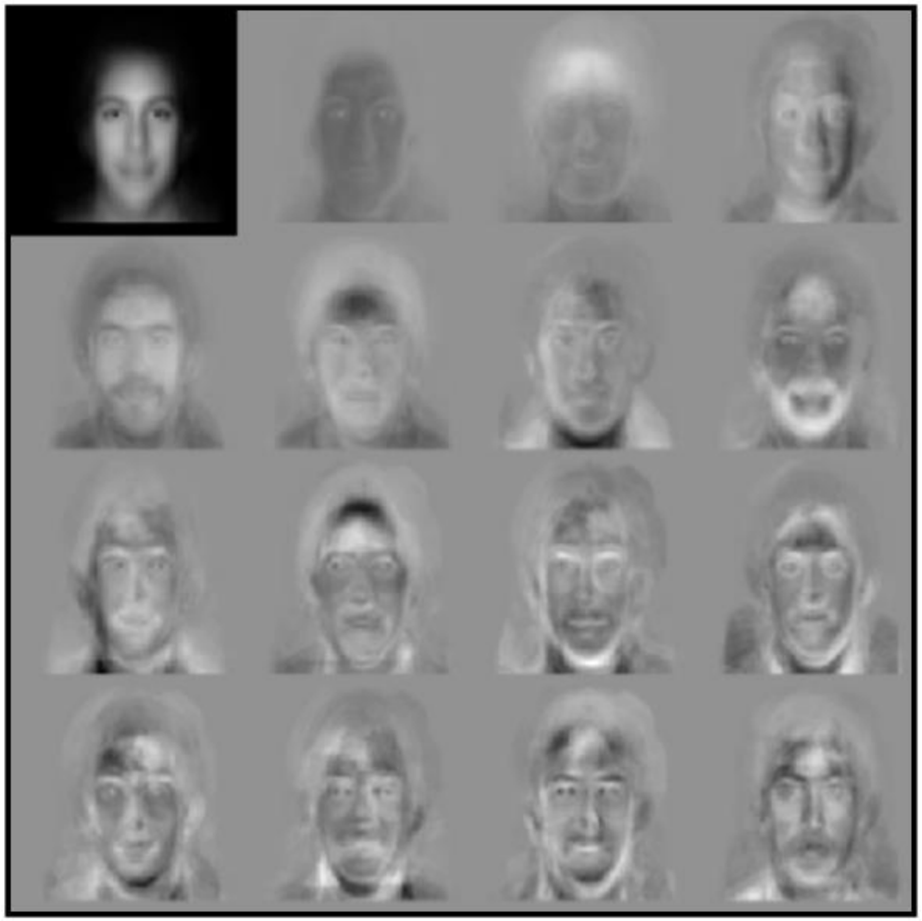
输出投影矩阵 $W=(w_1,w_2,...,w_{d'})$。

主成分分析

- **讨论**
  - 降低至多少维：
    - 采用交叉验证，结合**最近邻分类器**来选择合适的维度 d 。
    - 公式法： $\dfrac{\sum_{i=1}^{d}\lambda_i}{\sum_{i=1}^{m}\lambda_i} \geq t$ （比如， t=95%）
  - 舍弃m-d个特征值对应的特征向量导致了维数缩减。
    - 舍弃这些信息之后能使样本的采样密度增大，这正是降低维的重要动机。
    - 另外，当数据受到噪声影响时，最小的特征值所对应的特征向量往往与噪声有关，将它们舍弃可在一定程度上起到去噪的效果。

主成分分析应用

- **讨论-物理意义（Meanface，EigenFaces）**



Eigenfaces
from 7562 images:

top left image is linear combination of rest.

Sirovich & Kirby (1987)
Turk & Pentland (1991)

Face Recognition using Eigenfaces. CVPR, 1991.
Lawrence Saul. Spectral Methods for Dimensionality Reduction. NPIS2005 Tutorial

主成分分析- 概率PCA

–构建一个概率模型，对于一个预测数据$\boldsymbol{x}$，可以认为它是由一个隐变量z生成并且z及其条件概率$p(x|z)$均服从高斯分布：

$$p(z) = N(z|0, I)$$

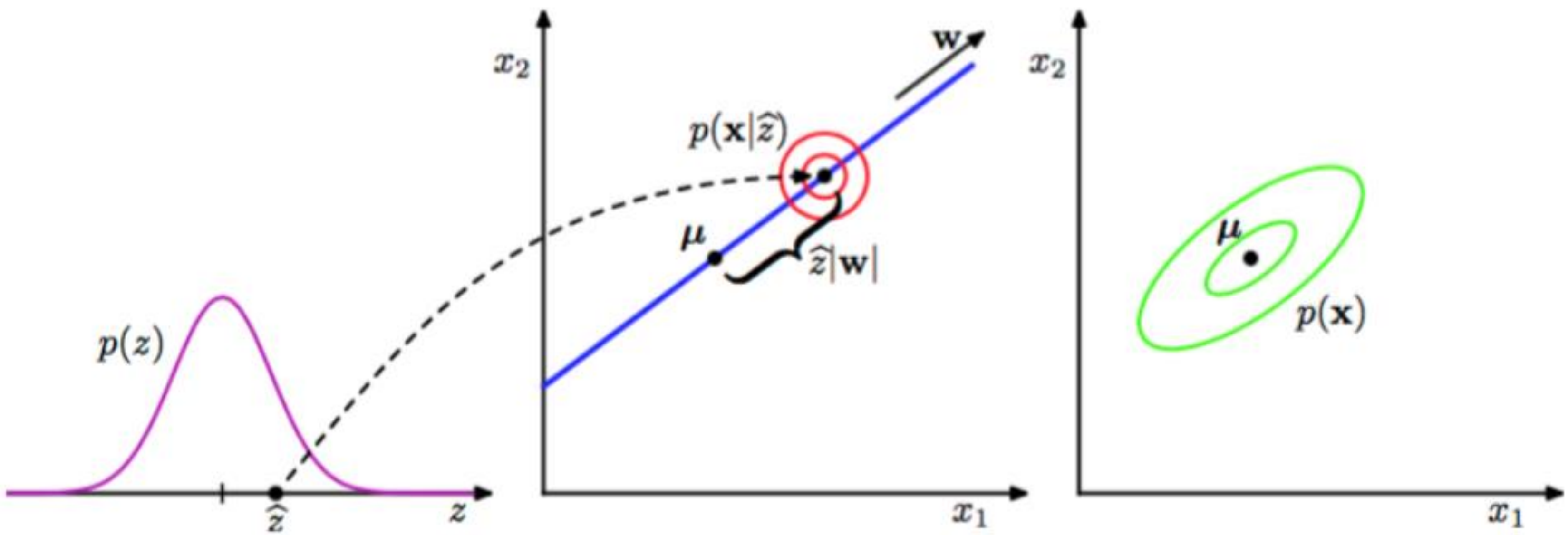
$$p(x|z) = N(x|Wz + \mu, \sigma^2 I)$$

$$x = Wz + \mu + \epsilon$$

其中，x的均值是z的一个一般线性函数，由$m \times d$的矩阵W（列张成数据空间的线性子空间）和m维向量μ控制。而σ^2控制着条件概率分布的方差，ϵ 为m维0均值高斯分布噪声，协方差为$\sigma^2 I$。

主成分分析-概率PCA

构建一个概率模型，对于一个预测数据$\boldsymbol{x}$，可以认为它是由一个隐变量z生成并且z及其条件概率$p(x|z)$均服从高斯分布，生成过程如下：



主成分分析- 概率PCA

线性高斯模型，还是高斯分布

$$p(x)=\int p(x|z)p(z)dz = N(x|\mu, C)$$

$$C = WW^T + \sigma^2 I$$

$$E(x)=E(Wz+\mu+\epsilon)=\mu$$

$$cov(x)=\mathrm{E}([X-E(x)]^2)=E[(Wz+\epsilon)(Wz+\epsilon)^T]$$

$$cov(x)=E(Wzz^TW^T)+E(\epsilon\epsilon^T)=WW^T+\sigma^2 I$$

z, ε相互独立的随机变量, $p(z)=N(z|0,I)$, $p(\epsilon)=N(z|0,\sigma I)$

主成分分析

- 求解
  - 对应的线性高斯模型，它的边缘分布还是高斯模型：

$$p(x) = \mathrm{N}(\mathrm{x}|\mu, C)$$

C是$D \times D$协方差矩阵, $D = \{x_1, x_2, \dots, x_n\}$

$$C = \mathrm{WW}^T + \sigma^2 I$$

  - 极大似然估计函数为：

$$lnp(X|W, u, \sigma^2) = -\frac{n}{2}\{Dln(2\pi) + \ln|C| + tr(C^{-1}S)\}$$

$$S = \frac{1}{n}\sum(x_i - u)(x_i - u)^T$$

中国科学院大学 University of Chinese Academy of Sciences

主成分分析

- 求解
  - 解为：$W = U_M(L_M - \sigma^2 I)^{\frac{1}{2}}R$　L_M是包含特征值的对角矩阵
  - 特殊情况：$\boldsymbol{R = I}$

$$W = U_M(L_M - \sigma^2 I)^{\frac{1}{2}}$$

W中包含特征值$\boldsymbol{L}_{\mathrm{M}}$
起到缩放到标准高斯的作用

  - 考虑特征值的标准PCA的映射关系

$$\boldsymbol{x = Wz = U_M L_M^{\frac{1}{2}} z}$$

  - 伸缩矩阵：$\boldsymbol{L_0 = \left(L_M - \sigma^2 I\right)^{\frac{1}{2}}}$

中国科学院大学 University of Chinese Academy of Sciences

主成分分析

- 求解
  - W的列决定了标准PCA的主子空间，对于σ^2，对应的最大化似然解为：

$$\sigma^2 = \frac{1}{m-d}\sum_{j=d+1}^{m} \lambda_j$$

此时σ^2是与丢弃维度相关的平均方差，如果u是主子空间（principal subspace）中的一个向量：

$$u^T C u = u^T(WW^T + \sigma^2 I)u = \left(u^T U_M (L_M - \sigma^2 I) U_M^T u\right) + \sigma^2$$

$$= (u^T \sum_{\mathrm{j}=1}^{\mathrm{d}} \lambda_{\mathrm{j}} u_j u_j^T u - \sigma^2) + \sigma^2$$

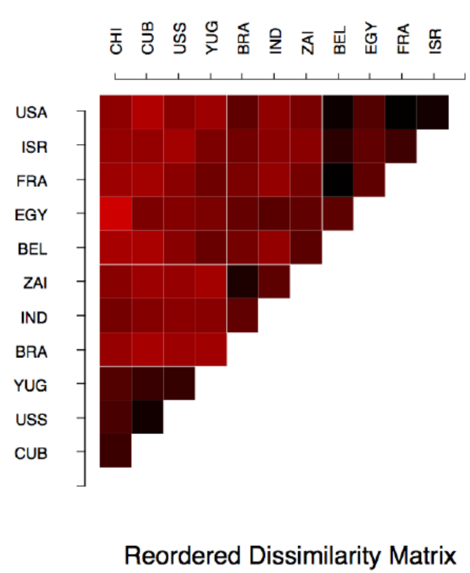
$$= (\lambda - \sigma^2) + \sigma^2 = \lambda$$

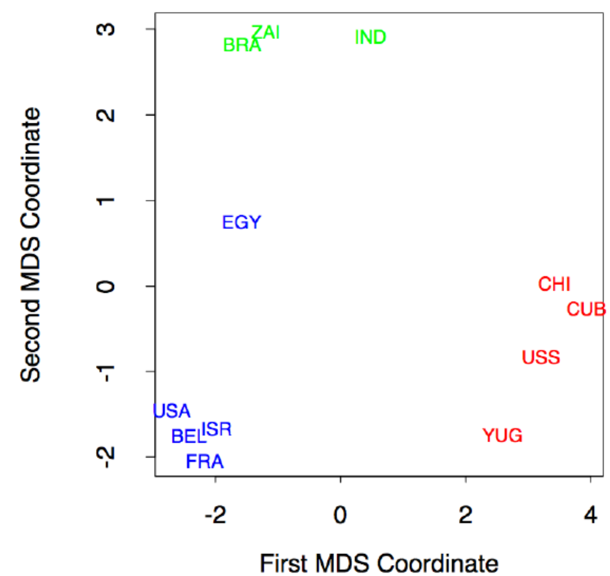
内容提要

- 引言
- 主成分分析
- **<u>多维缩放</u>**
- 流形学习方法
- 距离度量学习

多维缩放(Multiple Dimensional Scaling, MDS)

– 假定m维空间的n个样本 $\{x_1, x_2, \ldots .x_n\} \in R^m$在原始空间的**距离矩阵** $D \in R^{n \times n}$, 其第i行j列的元素为样本x_i到x_j的距离。





已知距离矩阵 → 低维的MDS坐标

多维缩放(Multiple Dimensional Scaling, MDS)

– 假定m维空间的n个样本$\{x_1, x_2, ...x_n\} \in R^m$在原始空间的距离矩阵$D \in R^{n\times n}$，其第$i$行$j$列的元素为样本$x_i$到$x_j$的距离。

– 目标：获得这n个样本$Z = [z_1, z_2, ..., z_n] \in R^{d\times n}$在$d(d<m)$维空间中的表示。

– 准则：假定降维后的样本仍保持两两之间的距离。

MDS不是保留数据的最大可分性，而是**更加关注与高维数据内部的特征**。MDS算法集中于保留高维空间中的"**相似度**"信息，即$B \doteq Z^T Z$。

中国科学院大学
University of Chinese Academy of Sciences

多维缩放(Multiple Dimensional Scaling, MDS)

– 假定m维空间的n个样本$\{x_1, x_2, \ldots, x_n\} \in R^m$在原始空间的距离矩阵$D \in R^{n\times n}$，其第$i$行$j$列的元素为样本$x_i$到$x_j$的距离。

– 目标：获得这n个样本$Z = [z_1, z_2, \ldots, z_n] \in R^{d\times n}$在$d(d<m)$维空间中的表示。

– 准则：假定降维后的样本仍保持两两之间的距离。

$$dist_{ij}^2 = ||z_i||_2^2 + ||z_j||_2^2 - 2z_i^T z_j$$

令$b_{ij} = z_i^T z_j$

$$dist_{ij}^2 = b_{ii} + b_{jj} - 2b_{ij}$$

$$b_{ij} = 1/2(\boxed{b_{ii}} + \boxed{b_{jj}} - dist_{ij}^2)$$

$D \in R^{n\times n}$

多维缩放(Multiple Dimensional Scaling, MDS)

– 如果给定一个距离矩阵D（该矩阵中实际上保存着原始样本数据点之间的距离，是一个实对称矩阵），我们能不能得到矩阵B，进而得到Z?

– 实际上如果不加上其它条件约束的话，这样是做不到的，因为整体平移和旋转样点是不改变样点间距离的，可以得到无数个满足要求的矩阵B。

– 不失一般性，令降维后的样本是**零均值化（零中心化）**的，即 $\sum_{i=1}^{n} z_i = 0 \in R^d$

– 矩阵B的行和列之和均为零，即 $\sum_{i=1}^{n} b_{ij} = \sum_{j=1}^{n} b_{ij} = 0$

$$\sum_{i=1}^{n} b_{ij} = (z_1^T z_j + \ldots + z_n^T z_j) = (z_1^T + \ldots + z_n^T) z_j = 0^T z_j = 0$$

多维缩放(Multiple Dimensional Scaling, MDS)

$$dist_{ij}^2 = \|z_i\|_2^2 + \|z_j\|_2^2 - 2z_i^T z_j$$

– 通过计算可以求得：

$$\sum_{i=1}^{n} b_{ij} = \sum_{j=1}^{n} b_{ij} = 0$$

$$\sum_{i=1}^{n} dist_{ij}^2 = \sum_{i=1}^{n} (b_{ii} + b_{jj} - \boxed{2b_{ij}}) = tr(B) + nb_{jj}$$

$$\sum_{j=1}^{n} dist_{ij}^2 = \sum_{j=1}^{n} (b_{ii} + b_{jj} - \boxed{2b_{ij}}) = tr(B) + nb_{ii}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} dist_{ij}^2 = 2ntr(B)$$

其中$tr(.)$表示矩阵的迹(trace), $tr(B) = \sum\|z_i\|_2^2$

同时tr(B)可计算，

$$tr(B) = \frac{1}{2n} \sum_{i=1}^{n} \sum_{j=1}^{n} dist_{ij}^2$$

多维缩放(Multiple Dimensional Scaling, MDS)

- 引入如下新符号，并记为：

$$dist_{i\cdot}^2 = \frac{1}{n}\sum_{j=1}^{n} dist_{ij}^2$$

$$dist_{\cdot j}^2 = \frac{1}{n}\sum_{i=1}^{n} dist_{ij}^2$$

$$dist_{\cdot\cdot}^2 = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} dist_{ij}^2$$

$$\sum_{j=1}^{n} dist_{ij}^2 = tr(B) + nb_{ii}$$

$$\sum_{i=1}^{n} dist_{ij}^2 = tr(B) + nb_{jj}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} dist_{ij}^2 = 2ntr(B)$$

可以算出，

$$tr(B) = \frac{n}{2} dist_{\cdot\cdot}^2, \quad b_{ii} = dist_{i\cdot}^2 - \frac{1}{2} dist_{\cdot\cdot}^2, \quad b_{jj} = dist_{\cdot j}^2 - \frac{1}{2} dist_{\cdot\cdot}^2$$

多维缩放(Multiple Dimensional Scaling, MDS)

- 可计算出 $b_{ij}=\frac{1}{2}(dist_{i\cdot}^2+dist_{\cdot j}^2-dist_{\cdot\cdot}^2-dist_{ij}^2)$
- 在获得矩阵B之后，则可对矩阵B进行特征值分解。
- 注意到：$B=Z^TZ\in R^{n\times n}$，且B是对称矩阵，于是有：

$$B=U\Lambda U^T \qquad Z=\Lambda_{\mathrm{d}}^{\frac{1}{2}}U_d^T\in R^{d\times n}$$

$$\Lambda_{\mathrm{d}}^{\frac{1}{2}}=diag\left(\sqrt{\lambda_1,\lambda_2,\ldots,\lambda_d}\right)\in R^{d\times d} \quad U_d=[u_1,u_2,\ldots,u_d]\in R^{n\times d}$$

其中，$\Lambda_{\mathrm{d}}^{\frac{1}{2}}$ 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；U_d 由前 d 个最大的特征值对应的特征向量组成。

多维缩放(Multiple Dimensional Scaling, MDS)

MDS算法步骤

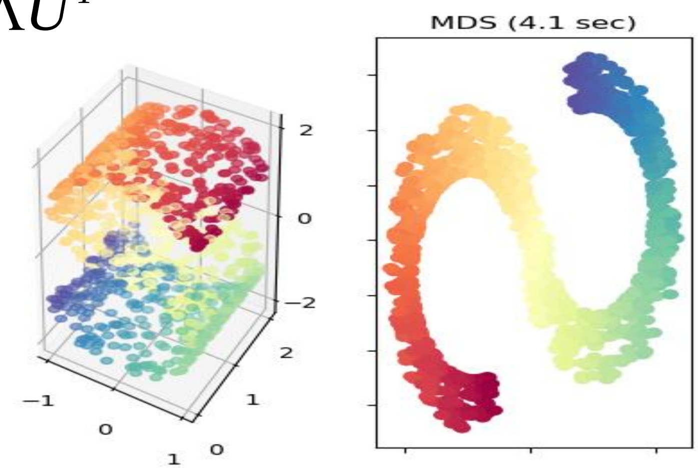
1　给定距离矩阵D;

2　计算$dist^2_{i.}$, $dist^2_{j.}$, $dist^2_{..}$;

3　计算矩阵B: $b_{ij}=-0.5(dist^2_{ij} - dist^2_{i.} - dist^2_{j.} + dist^2_{..})$;

4　对矩阵B进行**特征值分解**：$B = U\Lambda U^T$

5　输出：$Z = \Lambda_d^{\frac{1}{2}} U_d^T \in R^{d\times n}$



MDS举例 -飞行距离可视化

- 18个城市的飞行距离Airline distances (km)

TABLE 13.2. *Airline distances (km) between 18 cities. Source: Atlas of the World, Revised 6th Edition, National Geographic Society, 1995, p. 131.*

	Beijing	Cape Town	Hong Kong	Honolulu	London	Melbourne
Cape Town	12947					
Hong Kong	1972	11867				
Honolulu	8171	18562	8945			
London	8160	9635	9646	11653		
Melbourne	9093	10338	7392	8862	16902	
Mexico	12478	13703	14155	6098	8947	13557
Montreal	10490	12744	12462	7915	5240	16730
Moscow	5809	10101	7158	11342	2506	14418
New Delhi	3788	9284	3770	11930	6724	10192
New York	11012	12551	12984	7996	5586	16671
Paris	8236	9307	9650	11988	341	16793
Rio de Janeiro	17325	6075	17710	13343	9254	13227
Rome	8144	8417	9300	12936	1434	15987
San Francisco	9524	16487	11121	3857	8640	12644
Singapore	4465	9671	2575	10824	10860	6050
Stockholm	6725	10334	8243	11059	1436	15593
Tokyo	2104	14737	2893	6208	9585	8159

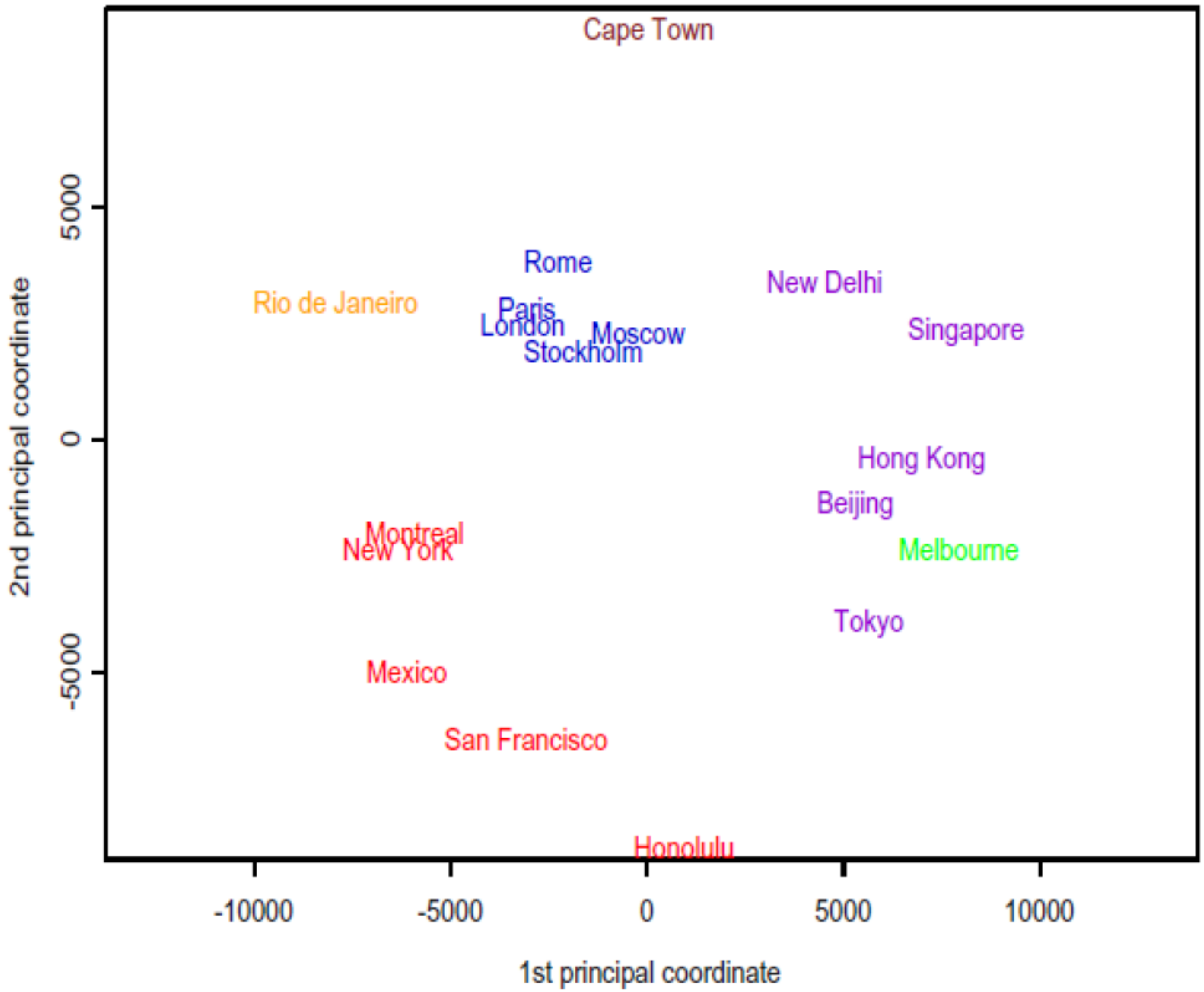
MDS举例 -飞行距离可视化

TABLE 13.6. *Eigenvalues of* B *and the eigenvectors corresponding to the first three largest eigenvalues (in red) for the airline distances example.*

	Eigenvalues	Eigenvectors		
1	471582511	0.245	–0.072	0.183
2	316824787	0.003	0.502	-0.347
3	253943687	0.323	–0.017	0.103
4	–98466163	0.044	–0.487	-0.080
5	–74912121	–0.145	0.144	0.205
6	–47505097	0.366	–0.128	–0.569
7	31736348	–0.281	–0.275	–0.174
8	–7508328	–0.272	–0.115	0.094
9	4338497	–0.010	0.134	0.202
10	1747583	0.209	0.195	0.110
11	–1498641	–0.292	–0.117	0.061
12	145113	–0.141	0.163	0.196
13	–102966	–0.364	0.172	–0.473
14	60477	–0.104	0.220	0.163
15	–6334	–0.140	–0.356	–0.009
16	–1362	0.375	0.139	–0.054
17	100	–0.074	0.112	0.215
18	0	0.260	–0.214	0.173

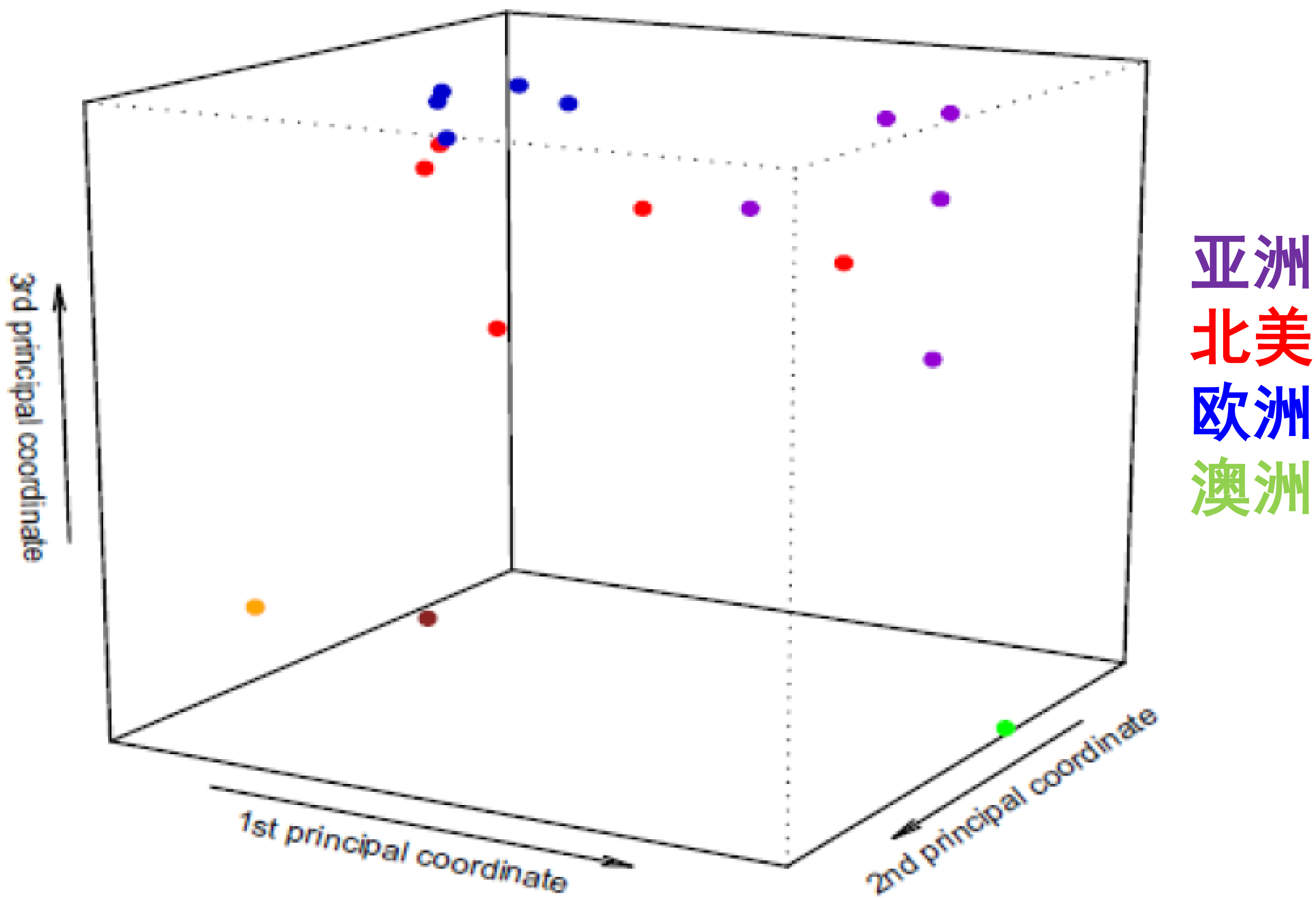
- Airline distance is non-Euclidean
- Take the first 3 largest eigenvalues (inspection of scree plot)

MDS举例 -飞行距离可视化



二维表示（2个特征向量）

MDS举例



三维表示（3个特征向量）

内容提要

- 引言
- 主成分分析
- 多维缩放
- **<u>流形学习方法</u>**
- 距离度量学习

中国科学院大学
University of Chinese Academy of Sciences

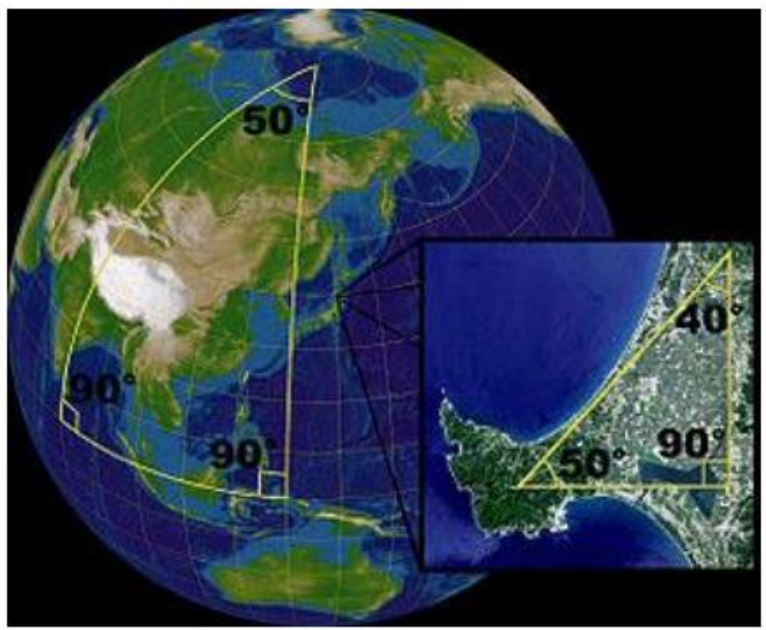
流形学习-流形定义

- **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**：流形是欧氏空间的一种推广，是在低维空间来表达高维空间所难以表达的空间结构。
- **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，但在全局部欧氏距离不成立。

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

流形学习-流形定义

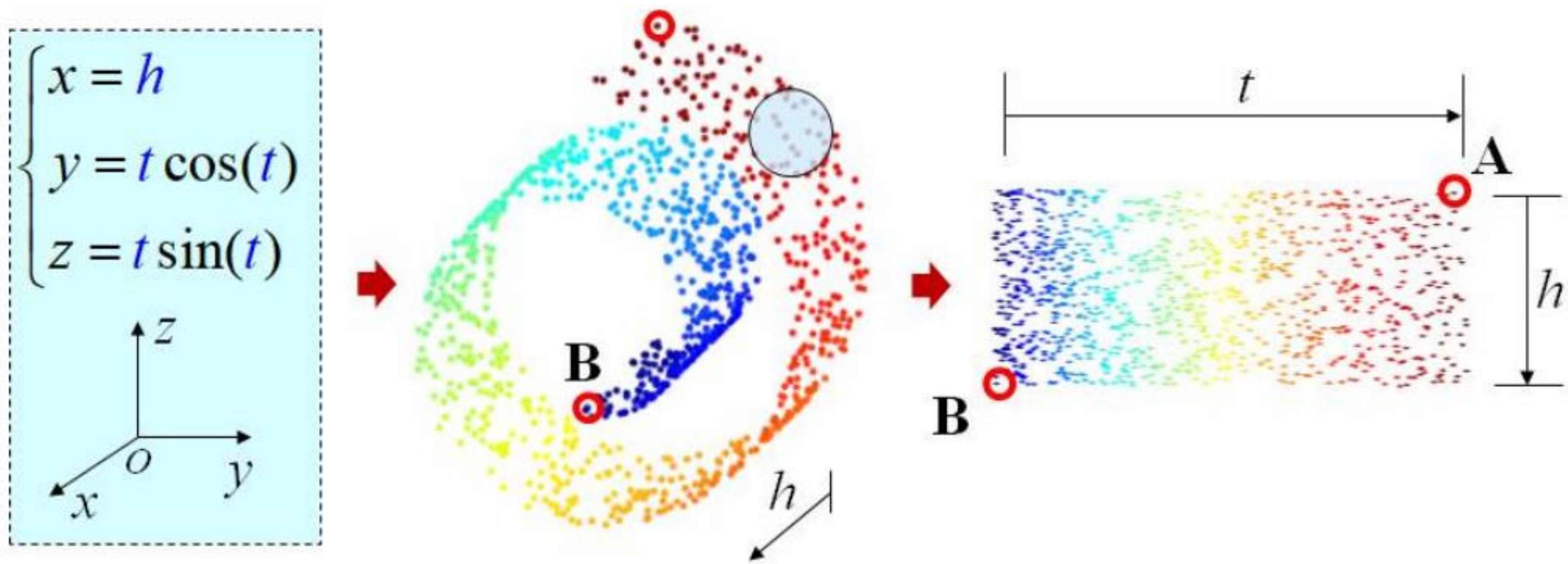
- **定义**：流形上的每一个点的开邻域，与欧氏空间的开集同胚。
- **几何**：流形是一块一块欧氏空间拼装而成的弯曲空间。
- **直观**：流形是欧氏空间的一种推广，是在低维空间来表达高维空间所难以表达的空间结构。
- **在数学上**：流形 … 在局部具有欧氏空间的性质。即可 … 部区域，但在全局部欧氏距离不成立 …



The **surfac** … ifold as it can be repr … nal maps.

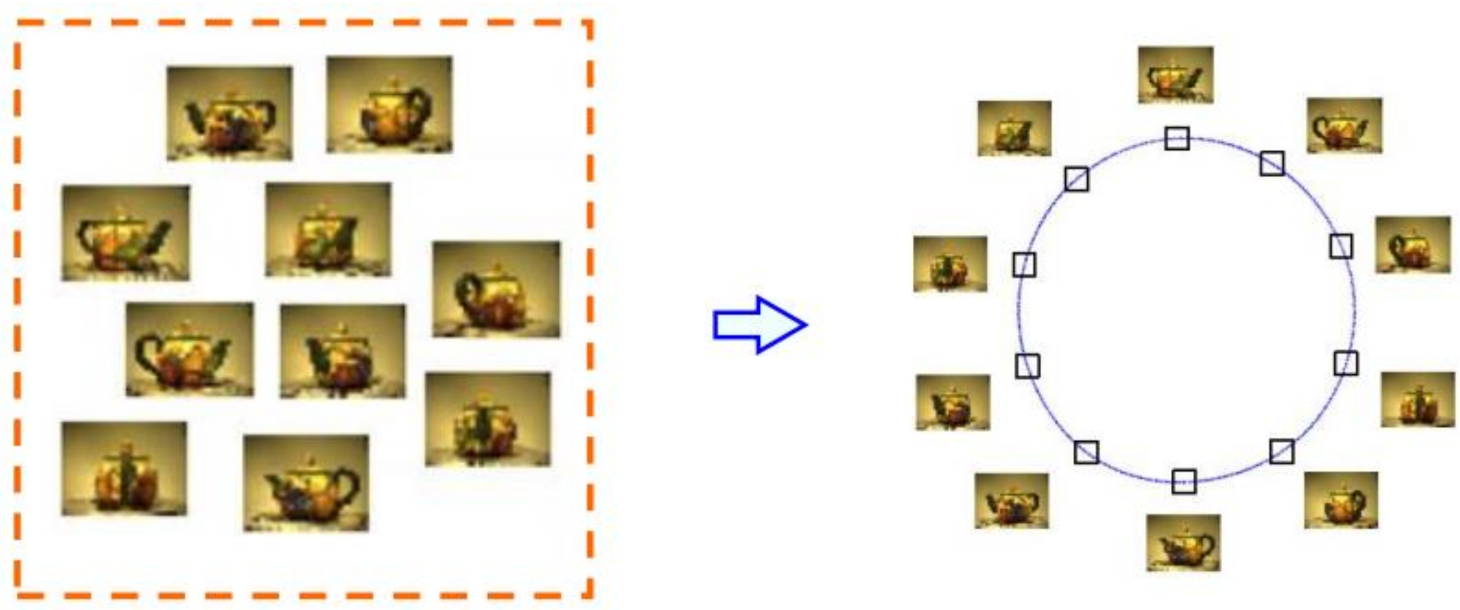
流形学习-流形示例

– **Swiss roll surface**：Swiss roll surface is a **2D** manifold。



流形学习-流形示例

– **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。



流形学习-数学描述

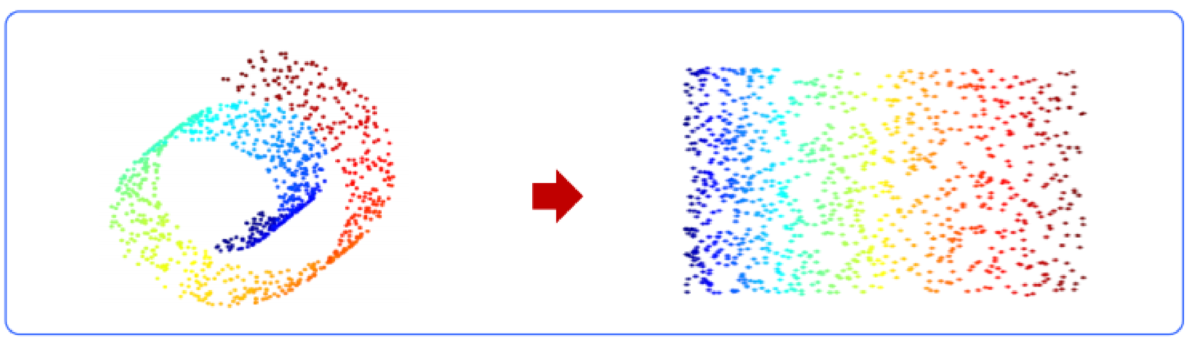
Given ??

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

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

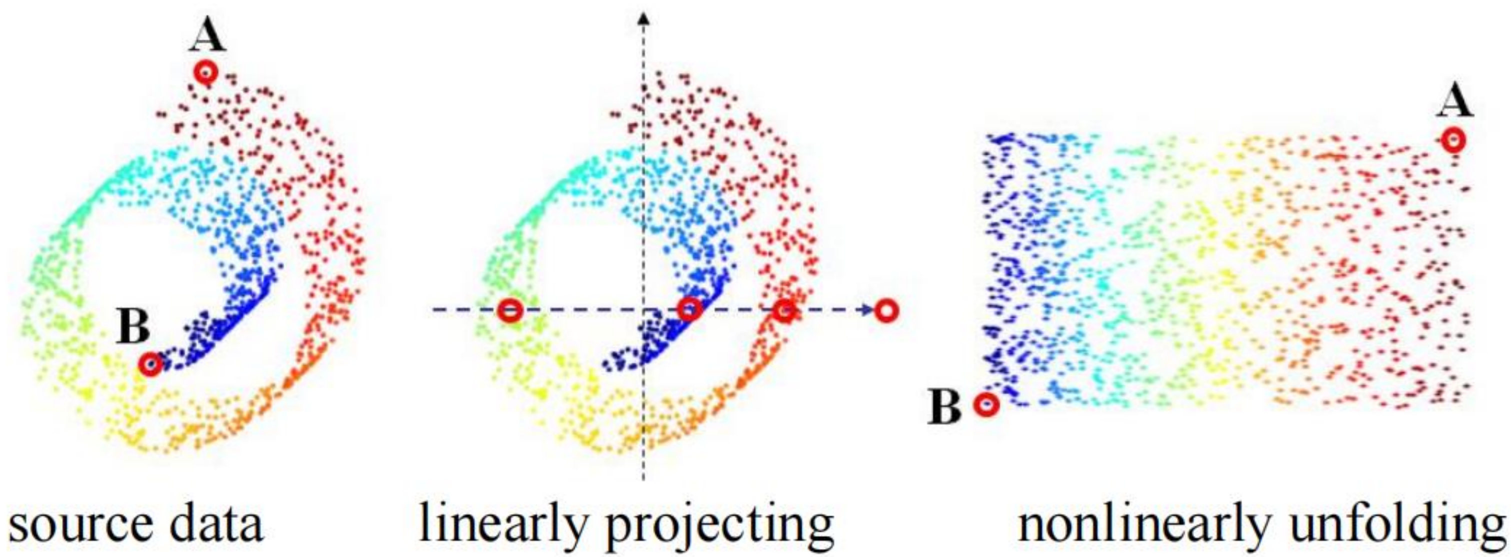
unknown

$$(d < m)$$



流形学习

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



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！

流形学习

- **一些假定**
  - 流形光滑（smooth manifold）：

    $(f: C \subset R^d \rightarrow R^m)$

  - 密集采样（densely sampling）
  - 不可自相交（no self-intersections ）

- **经典算法**
  - LLE， Isomap， Laplacian Eigenmap， HLLE， MVU， LTSA， LSE， etc。

流形学习-LLE(Locally linear embedding)

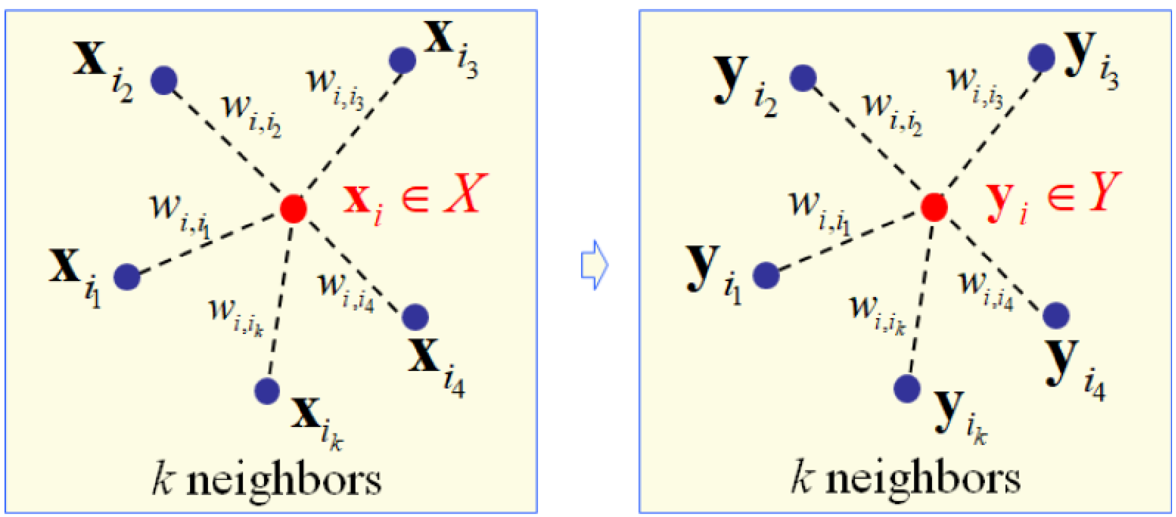
- **基本思想：**给定数据集，通过最近邻等方式构造一个数据图(data graph)。然后在每一个局部区域，高维空间中的样本线性重构关系在低维空间中均得以保持。

Sam Roweis & Lawrence Saul. *Nonlinear dimensionality reduction by locally linear embedding*. ***Science***, v.290 no.5500 , Dec.22, 2000. pp.2323--2326.

流形学习-LLE(Locally linear embedding)

- **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。
- **基本步骤：寻找近邻，线性重构，低维嵌入**

寻找近邻



$$\mathrm{x}_i \approx \sum_{j=1}^{k} w_{i,i_j} \mathrm{x}_{i_j} \qquad y_i \approx \sum_{j=1}^{k} w_{i,i_j} y_{i_j}$$

流形学习-LLE(Locally linear embedding)

– **最优线性表示系数**

$$\boldsymbol{min_{w_i}\sum_{i=1}^{m}||\mathrm{x}_i-\sum_{j=1}^{k}w_{i,i_j}\mathrm{x}_{i_j}||_2^2,\sum_{j=1}^{k}w_{i,i_j}=1}$$

通过拉格朗日乘子法，可得如下有关线性表示系数的解：

$$\boldsymbol{w_i=\frac{Z_i^{-1}e}{e^TZ_i^{-1}e}}$$

流形学习-LLE(Locally linear embedding)

– **最优线性表示系数**

$$\boldsymbol{min}_{\boldsymbol{w}_i}\sum_{\boldsymbol{i}=1}^{\boldsymbol{m}}||\mathrm{x}_{\boldsymbol{i}}-\sum_{\boldsymbol{j}=1}^{\boldsymbol{k}}\boldsymbol{w}_{\boldsymbol{i},\boldsymbol{i}_j}\mathrm{x}_{\boldsymbol{i}_j}||_2^2,\sum_{\boldsymbol{j}=1}^{\boldsymbol{k}}\boldsymbol{w}_{\boldsymbol{i},\boldsymbol{i}_j}=\boldsymbol{1}$$

通过拉格朗日乘子法，可得如下有关线性表示系数的解：

$$\boldsymbol{w}_i=\frac{\boldsymbol{Z}_i^{-1}\boldsymbol{e}}{\boldsymbol{e}^T\boldsymbol{Z}_i^{-1}\boldsymbol{e}}$$

$$w_i=\left[w_{i,i_1},w_{i,i_2},\ldots,w_{i,i_k}\right]^T\subset R^k$$

$$\mathrm{X}_i=[\mathrm{x}_i,\mathrm{x}_i,\ldots,\mathrm{x}_i]^T\subset R^{m\times k}$$

$$N_i=[\mathrm{x}_{i1},\mathrm{x}_{i2},\ldots,\mathrm{x}_{ik}]^T\subset R^{m\times k}$$

$$e=[1,1,\ldots,1]^T\subset R^k$$

$$Z_i=(X_i-N_i)^T(X_i-N_i)$$

中国科学院大学 University of Chinese Academy of Sciences

流形学习-LLE(Locally linear embedding)

– 推导过程

$$J(w)=min_{w_i}\sum_{i=1}^{m}||\mathrm{x}_i-\sum_{j=1}^{k}w_{i,i_j}\mathrm{x}_{i_j}||_2^2$$

$$=\sum_{i=1}^{m}\left\|\sum_{j=1}^{k}w_{i,i_j}\mathrm{x}_i-\sum_{j=1}^{k}w_{i,i_j}\mathrm{x}_{i_j}\right\|_2^2=\sum_{i=1}^{m}\left\|\sum_{j=1}^{k}w_{i,i_j}(\mathrm{x}_i-\mathrm{x}_{i_j})\right\|_2^2$$

$$=\sum_{i=1}^{m}||(X_i-N_i)w_i||_2^2$$

$$=\sum_{i=1}^{m}w_i^T\underbrace{(X_i-N_i)^T(X_i-N_i)}_{Z_i}w_i$$

流形学习-LLE(Locally linear embedding)

– **推导过程**

$$J(w) = \sum_{i=1}^{m} w_i^T Z_i w_i$$

对于限制项，可以写为：

$$\sum_{j=1}^{k} w_{i,i_j} = w_i^T e = 1$$

流形学习-LLE(Locally linear embedding)

– **拉格朗日乘子法**

$$L(w)=\sum_{i=1}^{m} w_i^T Z_i w_i+\lambda\left(w_i^T e-1\right)$$

对w_i求导可得：

$$2Z_i w_i+\lambda e=0$$

$$w_i=\frac{1}{2}\lambda Z_i^{-1} e$$

对w_i归一化：

$$\boldsymbol{w_i=\frac{Z_i^{-1} e}{e^T Z_i^{-1} e}}$$

流形学习-LLE(Locally linear embedding)

– **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：

$$y_i \approx \sum_{j=1}^{k} w_{i,i_j} y_{i_j} \quad i = 1, 2, \dots, n$$

– 考虑所有新样本点的重构误差，得到全局嵌入的目标函数：

$$\sum_{i=1}^{n} \|\boldsymbol{y_i} - \sum_{j=1}^{k} w_{i,i_j} y_{i_j}\|_2^2 = tr\,(\mathrm{Y}(\mathrm{I}-\mathrm{W})(\mathrm{I}-\mathrm{W})^{\mathrm{T}}\mathrm{Y}^{\mathrm{T}})$$

$$Y = \left[y_1, y_2, \dots, y_n\right]^T \in R^{d\times n}$$

W 为权重矩阵，其第 i 行记录对应样本点 x_i 的 k 个权重，只有在对应的邻居位置 $i_1, i_2, \dots, i_k$ 处才有值，其余全为零。

流形学习-LLE(Locally linear embedding)

$$\sum_{i=1}^{n} ||\boldsymbol{y}_i - \sum_{j=1}^{k} w_{i,i_j} y_{i_j} ||_2^2$$

i近邻点 $W_{ij} = w_{i,i_j}$ 否则 $W_{ij} = 0$

$$\sum_{j=1}^{\mathrm{n}} W_{i,j}\, y_{i_j} = \sum_{j=1}^{k} w_{i,i_j} y_{i_j} = YW_i$$

$$\sum_{i=1}^{n} ||\boldsymbol{Y}(\boldsymbol{I}_i - \boldsymbol{W}_i)||_2^2$$

$$= tr\,(\mathrm{Y}(\mathrm{I} - \mathrm{W})(\mathrm{I} - \mathrm{W})^{\mathrm{T}}\mathrm{Y}^{\mathrm{T}})$$

$$\because A = [a_1, a_2, \ldots, a_n],\ \therefore \sum_i (a_i)^2 = \sum_i a_i^T a_i = tr(AA^T)$$

流形学习-LLE(Locally linear embedding)

- LLE（Locally linear embedding）
  - **全局嵌入学习模型**：

$$\min_{Y} tr\,(\mathrm{Y}(\mathrm{I}-\mathrm{W})(\mathrm{I}-\mathrm{W})^{\mathrm{T}}\mathrm{Y}^{\mathrm{T}})\ \ s.t.\, YY^T = I$$

  - 求解：通过求矩阵 $(\mathrm{I}-\mathrm{W})(\mathrm{I}-\mathrm{W})^{\mathrm{T}}$ 的特征值分解来得到。
    - 取出该矩阵最小的 $k+1$ 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。

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

流形学习

LLE算法步骤

Input: 给出数据 $X = [\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n] \in R^{m\times n}$,近邻参数$k$以及低维空间$d$；

for $i = 1,2,\ldots,n$

　　确定 的 k个近邻；

　　对 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W；

求解： $\min_{Y} tr\ (\mathrm{Y}(\mathrm{I}-\mathrm{W})(\mathrm{I}-\mathrm{W})^{\mathrm{T}}\mathrm{Y}^{\mathrm{T}})\ \ s.t. YY^T = I$

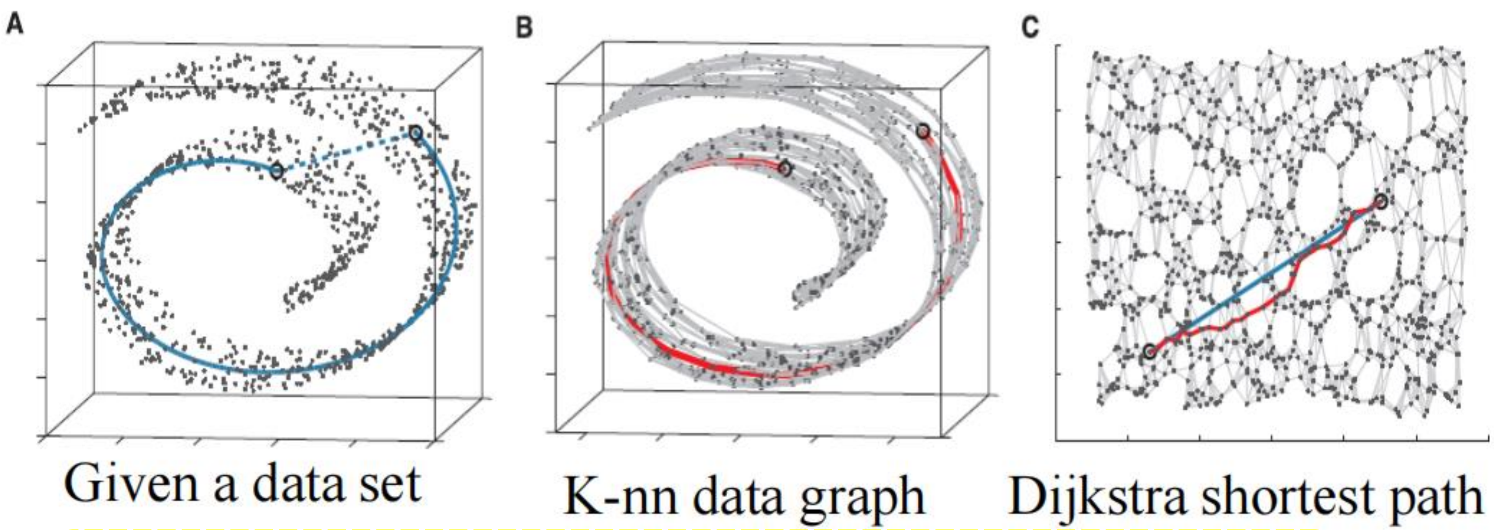
采用第2至第$d+1$个**最小的特征值对应的特征向量**组成新坐标；

输出结果。

University of Chinese Academy of Sciences

流形学习-Isomap(isometric feature mapping)

– 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。

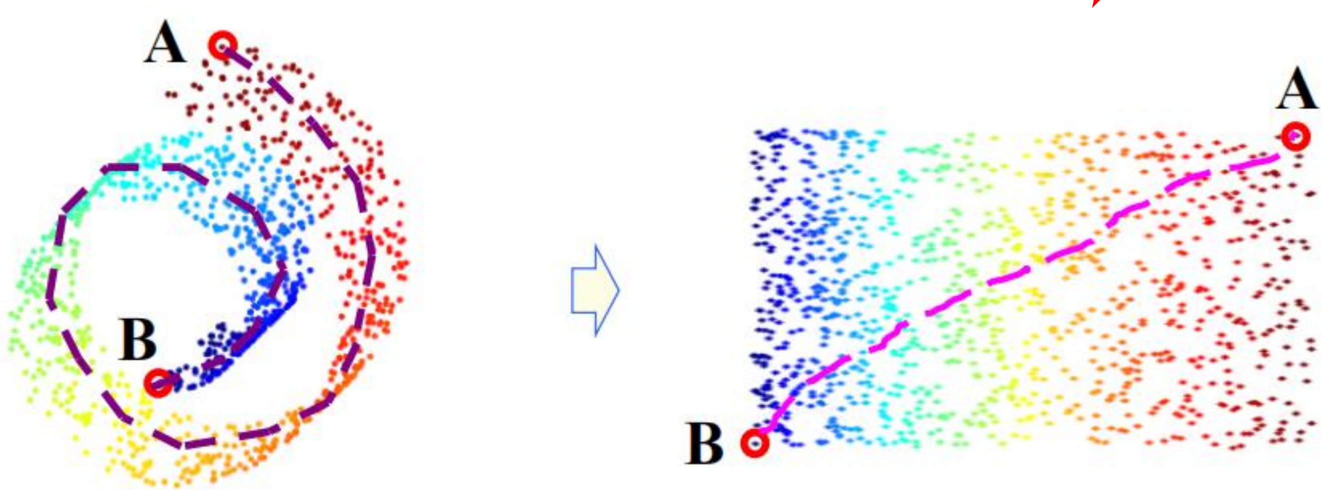


几乎所有的流形学习方法都需要首先构建一个关于数据的图

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.

流形学习-Isomap(isometric feature mapping)

– 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。



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.

流形学习-Isomap(isometric feature mapping)

Isomap算法步骤

1 **Input**: 给出数据 $X=[x_1,x_2,...,x_n]\in R^{m\times n}$ ，近邻参数 k 以及低维空间 d；

2 for $i=1,2,...,n$

3 确定 x_i 的 k 个近邻；

4 x_i 与 k 个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**最短路径法**计算任意两样本点 x_i 与 x_j 之间的距离 d_{ij}。由此可构造距离矩阵。

7 调用**MDS算法**（见MDS算法） ；

8 **MDS算法的计算结果作为低维嵌入结果**

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.

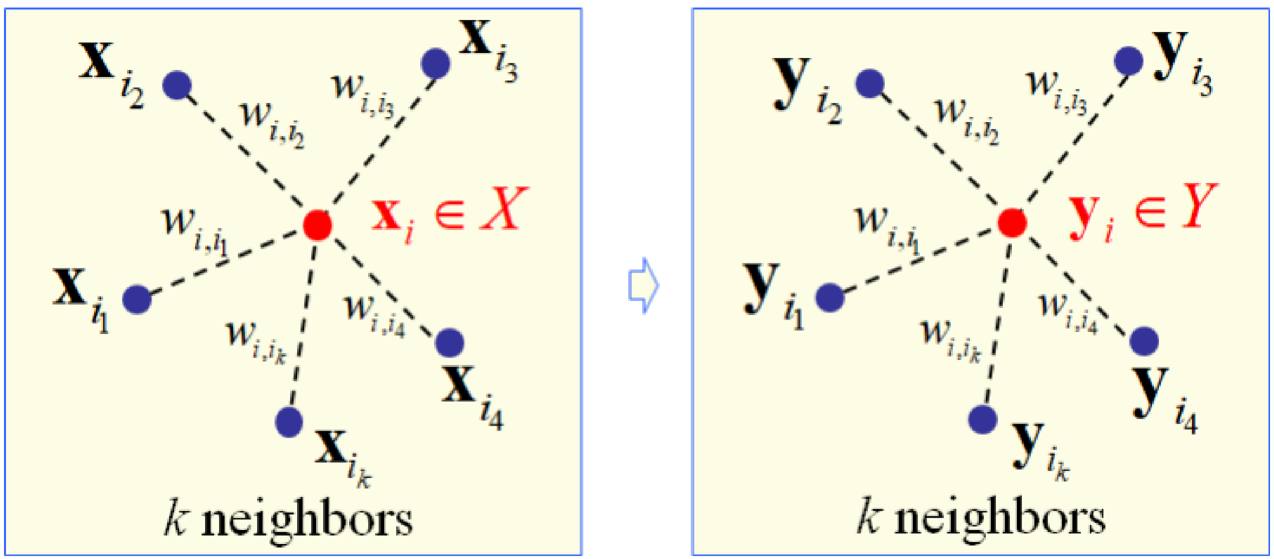
University of Chinese Academy of Sciences

流形学习-LE(Laplacian Eigenmapping)

– 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，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。

流形学习-LE(Laplacian Eigenmapping)

– 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，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。



流形学习-LE(Laplacian Eigenmapping)

– 如何计算点对亲合度？

流形学习-LE(Laplacian Eigenmapping)

- 如何计算**点对亲合度**？

$$w_{i,i_j} = \exp\left(-\frac{\left\|x_i - x_{i_j}\right\|_2^2}{2\sigma^2}\right)$$

流形学习-LE(Laplacian Eigenmapping)

- 如何计算点对亲合度？

$$w_{i,i_j} = \exp\left(-\frac{\left\|x_i - x_{i_j}\right\|_2^2}{2\sigma^2}\right)$$

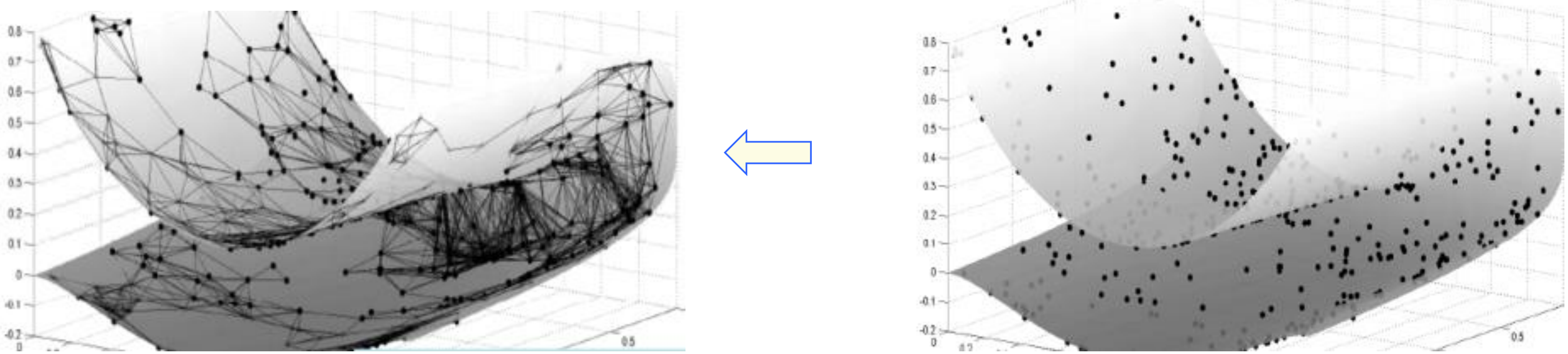
- 图构造 $G(V,E)$（第六章）

流形学习-LE(Laplacian Eigenmapping)

- 如何计算点对亲合度？

$$w_{i,i_j} = \exp\left(-\frac{\left\| x_i - x_{i_j} \right\|_2^2}{2\sigma^2} \right)$$

- 图构造 $G(V,E)$（第六章）



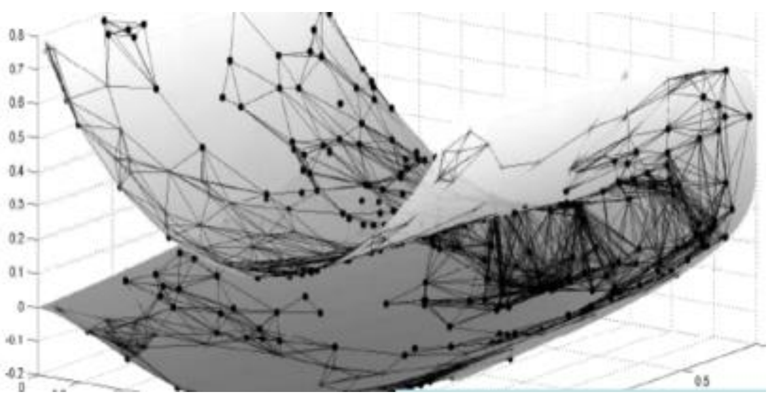
流形学习

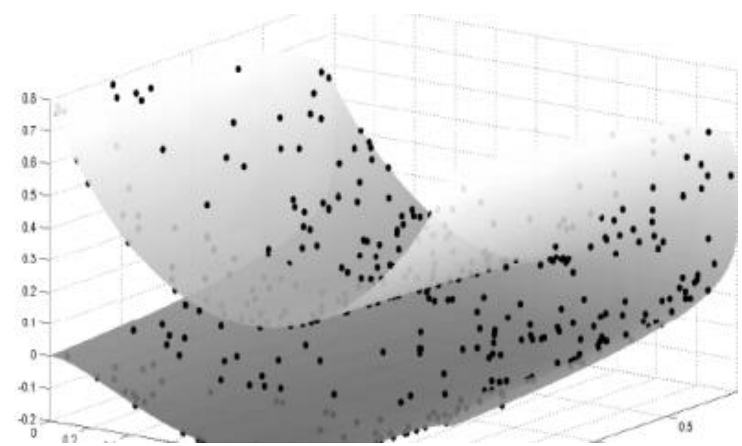
- 如何计算点对亲合度？

$$w_{i,i_j} = \exp\left(-\frac{\left\| x_i - x_{i_j} \right\|_2^2}{2\sigma^2} \right)$$

$$\begin{pmatrix} 0 & w_{12} & w_{12} & \bullet & \bullet & \bullet & w_{1n} \\ w_{21} & 0 & w_{22} & \bullet & \bullet & \bullet & w_{2n} \\ w_{31} & w_{32} & 0 & \bullet & \bullet & \bullet & w_{3n} \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ w_{n1} & w_{n2} & w_{n4} & \bullet & \bullet & \bullet & 0 \end{pmatrix}$$

- 图构造 $G(V,E)$（第六章）





流形学习

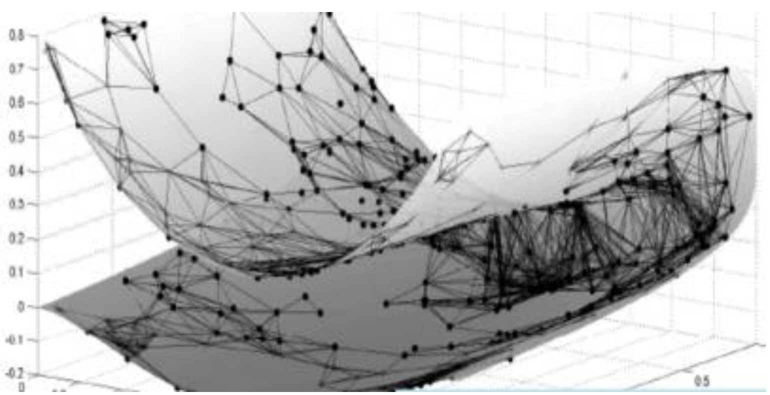
每行 k 个元素为零

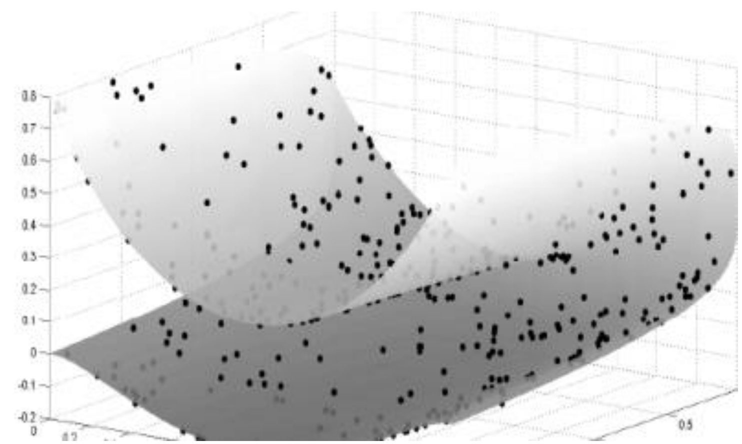
$$\begin{pmatrix} 0 & w_{12} & w_{12} & \bullet & \bullet & \bullet & w_{1n} \\ w_{21} & 0 & w_{22} & \bullet & \bullet & \bullet & w_{2n} \\ w_{31} & w_{32} & 0 & \bullet & \bullet & \bullet & w_{3n} \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ w_{n1} & w_{n2} & w_{n4} & \bullet & \bullet & \bullet & 0 \end{pmatrix}$$

– 如何计算点对亲合度？

$$w_{i,i_j} = \exp\left(- \frac{\left\| x_i - x_{i_j} \right\|_2^2}{2\sigma^2} \right)$$

– 图构造 $G(V,E)$（第六章）





流形学习-LE(Laplacian Eigenmapping)

- 如何在低维空间保持亲合度？

  构造如下目标函数:

流形学习-LE(Laplacian Eigenmapping)

– 如何在低维空间保持亲合度?

构造如下目标函数:

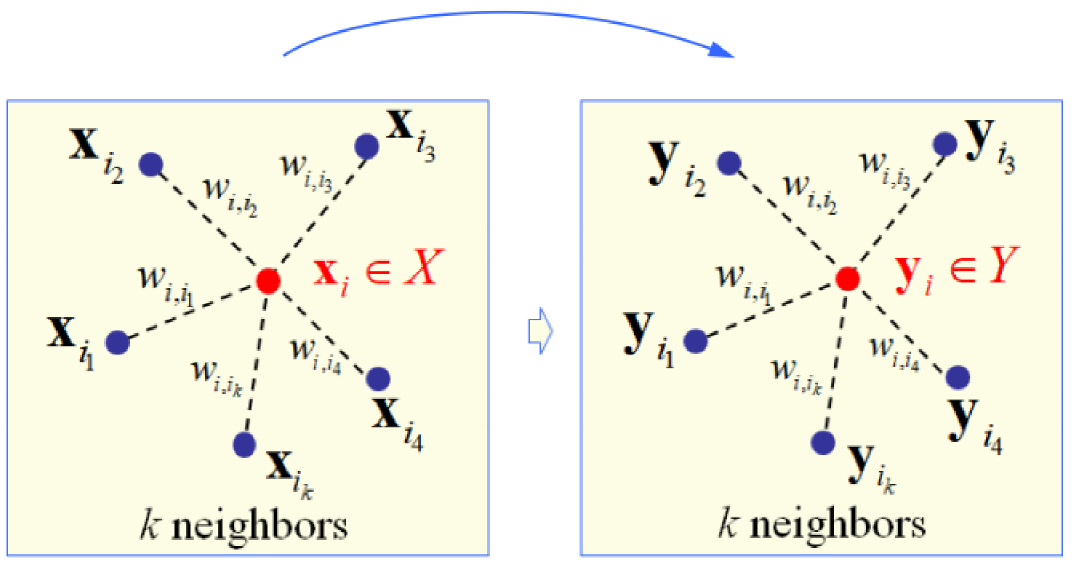
$$E(Y)=\sum_{i,j} w_{i,i_j}\left\|y_i - y_{i_j}\right\|_2^2$$

流形学习-LE(Laplacian Eigenmapping)

– 如何在低维空间保持亲合度？

构造如下目标函数:

$$E(Y)=\sum_{i,j} w_{i,i_j}\left\|y_i - y_{i_j}\right\|_2^2$$



流形学习-LE(Laplacian Eigenmapping)

- 考虑目标函数
  - 对任意向量 $\mathrm{f} = [f_1, \ f_2, \ldots, f_n]^T$ 有如下结论：

令

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

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

$$i = 1,2,\ldots,n$$

流形学习-LE(Laplacian Eigenmapping)

– 考虑目标函数

- 对任意向量 $\mathrm{f} = [f_1, f_2, \ldots, f_n]^T$ 有如下结论：

亲和度矩阵

令

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

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

$$i = 1,2,\ldots,n$$

$$\mathrm{f}^T(D - \boxed{W})\mathrm{f} = \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} f_i f_j w_{ij}$$

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

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

$$\geq 0$$

流形学习-LE(Laplacian Eigenmapping)

– 考虑目标函数

$$\boldsymbol{Y}=[\boldsymbol{y_1},\boldsymbol{y_2},\dots,\boldsymbol{y_n}]=\begin{pmatrix} y_{11} & y_{12} & \bullet & \bullet & \bullet & y_{1n} \\ y_{21} & y_{22} & \bullet & \bullet & \bullet & y_{2n} \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ y_{d1} & y_{d2} & \bullet & \bullet & \bullet & y_{dn} \end{pmatrix}=\begin{pmatrix} \mathrm{f}_1^T \\ \mathrm{f}_2^T \\ \bullet \\ \bullet \\ \bullet \\ \mathrm{f}_n^T \end{pmatrix}\in R^{d\times n}$$

其中，$\mathrm{f}_i=[y_{i1},y_{i2},\dots,y_{in}]^T\in R, i=1,2,\dots,d$

$$E(Y)=\sum_{i,j}w_{i,i_j}\left\|y_i-y_{i_j}\right\|_2^2=\sum_{i,j}w_{i,i_j}[\left(\boldsymbol{y_{1i}}-\boldsymbol{y_{1i_j}}\right)^2+\cdots+\left(\boldsymbol{y_{di}}-\boldsymbol{y_{di_j}}\right)^2]=$$

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

流形学习-LE(Laplacian Eigenmapping)

- 学习模型

$$\min E(\boldsymbol{Y}) = \boldsymbol{tr}\left(\boldsymbol{Y}(\boldsymbol{D}-\boldsymbol{W})\boldsymbol{Y}^{\boldsymbol{T}}\right),\ \ \boldsymbol{s.t.YY^{T}} = \mathbf{1}$$

- 令$\mathrm{L} = \boldsymbol{D}-\boldsymbol{W}$, L有一个特征值为0，对应的特征向量全为1：

$$Le = (D-W)e = (D-W)(1,1,\dots,1)^{T} = \begin{pmatrix} d_1 - \sum_{j=1}^{k} w_{1,1j} \\ d_2 - \sum_{j=1}^{k} w_{2,2j} \\ \bullet \\ \bullet \\ \bullet \\ d_n - \sum_{j=1}^{k} w_{n,nj} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \bullet \\ \bullet \\ \bullet \\ 0 \end{pmatrix} \in R^{n}$$

$$\Downarrow$$

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

流形学习

LE算法步骤

1 Input: 给出数据 $X=[x_1,x_2,...,x_n]\in R^{m\times n}$ ，近邻参数$\boldsymbol{k}$以及低维空间$\boldsymbol{d}$ ；

2 确定x_i的k个近邻，确定**亲和度矩阵W**，计算**度矩阵D**；

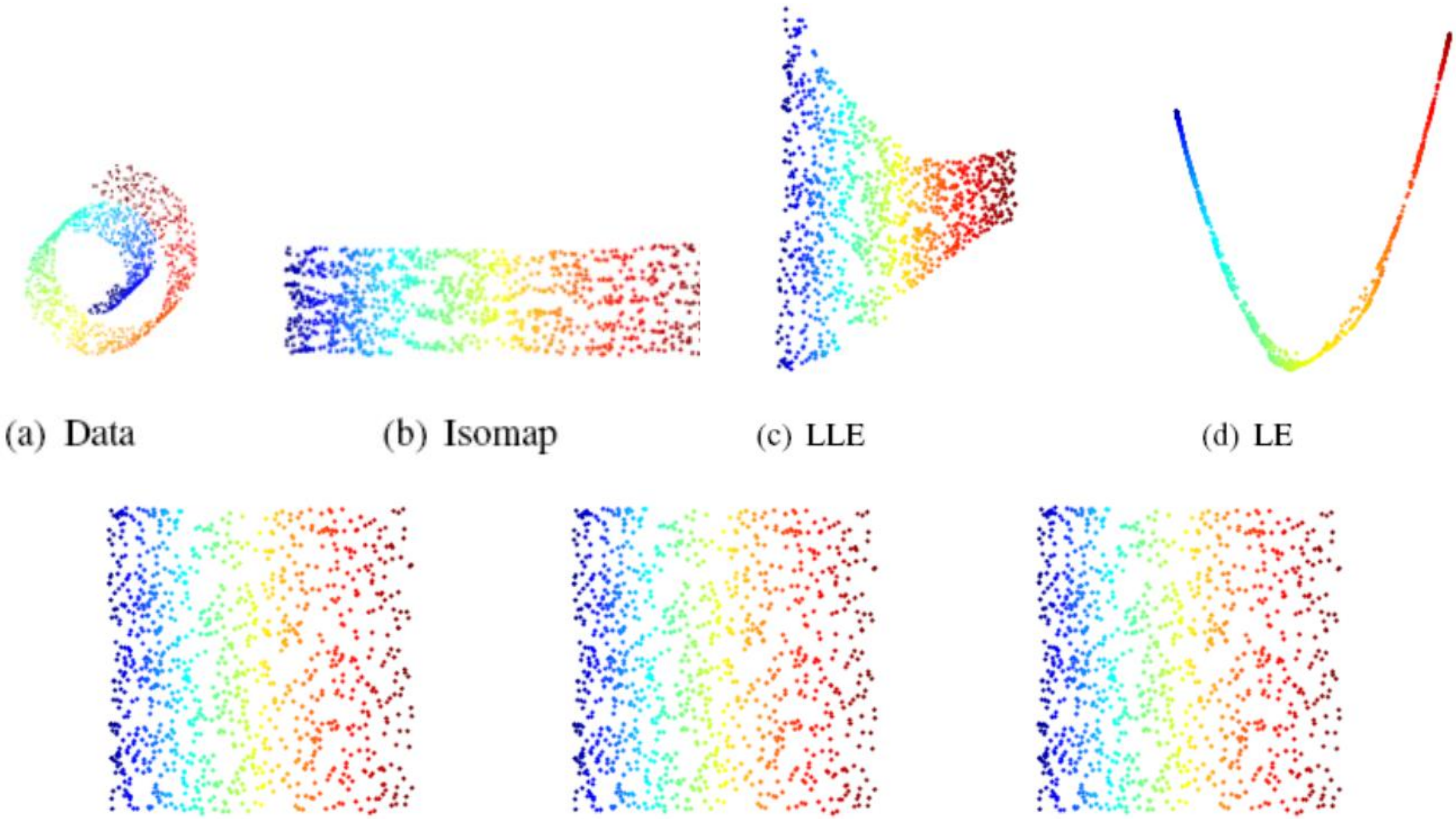
3 求解模型 $\mathbf{\min E(Y), s.t. YY^T = I}$；

4 采用第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；

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

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

流形学习-性能对比



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

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

流形学习-总结

– 统一的学习模型

- 目标：给定高维数据 $\{x_i\}_{i=1}^{n} \subset R^m$ 寻找其低维表示。

- 学习模型： $\{y_i\}_{i=1}^{n} \subset R^d, d < m$。

对L 进行特征值分解

$$\min_{Y} trace(YMY^T)$$
$$s.t. YY^T = I, Y = [y_1, y_2, \dots, y_n] \in R^{d \times n}$$

任务：构造 M 矩阵--与数据图构造和局部描述紧密相关！

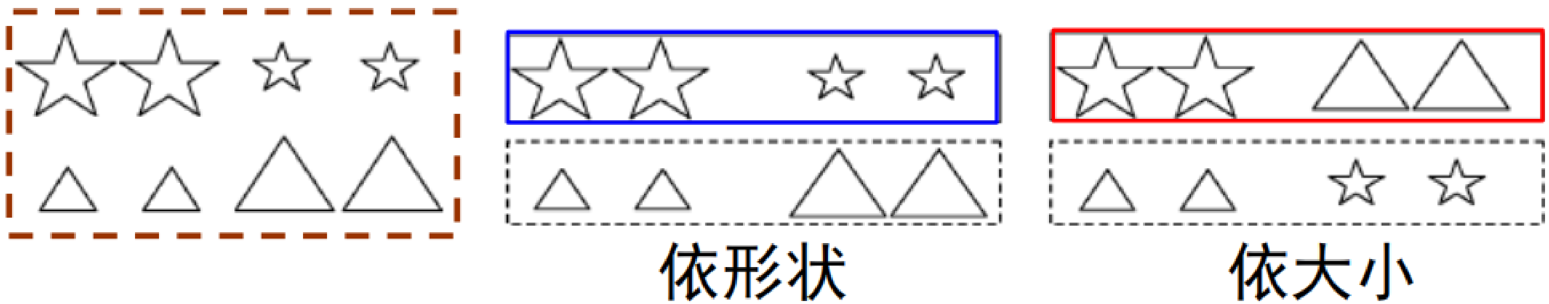
内容提要

- 引言
- 主成分分析
- 多维缩放
- 流形学习方法
- **距离度量学习**

度量学习

- **学习任务——距离度量是模式分类的基础！**
  - 能否：让距离度量反映用户偏好或某些先验知识？

形状检索



图像检索

度量学习

- **学习任务——距离度量是模式分类的基础！**
  - 给定一些相似点对和不相似点对，学习一个距离度量来反映用户的某种偏好。

$$d=\sqrt{\sum_{i=1}^{m}(x_i-y_i)^2}=\left((x-y)^T(x-y)\right)^{1/2}$$

欧氏距离等同地对待每一个特征分量，不能反映出偏好信息！

度量学习

- **学习任务——距离度量是模式分类的基础！**

- **考虑马氏距离**

单位矩阵

$$d_E(x,y)=\sqrt{(x-y)^T(x-y)}=\sqrt{(x-y)^T\boldsymbol{I}(x-y)}$$

$$d_A(x,y)=\sqrt{(x-y)^T\boldsymbol{A}(x-y)}=\sqrt{(x-y)^T\boldsymbol{W}\boldsymbol{W}^{\boldsymbol{T}}(x-y)}$$

度量学习

- **学习任务——距离度量是模式分类的基础！**
- **考虑马氏距离**

单位矩阵

$$d_E(x,y)=\sqrt{(x-y)^T(x-y)}=\sqrt{(x-y)^T\boldsymbol{I}(x-y)}$$

$$d_A(x,y)=\sqrt{(x-y)^T\boldsymbol{A}(x-y)}=\sqrt{(x-y)^T\boldsymbol{WW^T}(x-y)}$$

其中A为一个半正定矩阵。令 $A=WW^T$，则马氏矩阵度量等价于将数据通过 $y=W^Tx$ 变换后在新空间做欧氏距离计算。

度量学习

- **经典学习模型**

$$\max_{A} \sum_{x_i, x_j \in D} d_A(x_i, x_j)$$

$$s.t. \sum_{x_i, x_j \in S} \left[d_A(x_i, x_j)\right]^2 \leq 1$$

$$\boldsymbol{A \succ= 0}$$

E.P. Xing, A.Y. Ng, M.I. Jordan, S. Russell, Distance metric learning, with application to clustering with side-information, NIPS, 2003

度量学习

- **经典学习模型**

$$\max_{A} \sum_{x_i, x_j \in D} d_A(x_i, x_j)$$

不相似点对马氏距离尽可能小

$$s.t. \sum_{x_i, x_j \in S} \left[d_A(x_i, x_j)\right]^2 \le 1$$

相似点对马氏距离尽可能小

$$\boldsymbol{A \succeq 0}$$

半正定

E.P. Xing, A.Y. Ng, M.I. Jordan, S. Russell, Distance metric learning, with application to clustering with side-information, NIPS, 2003

度量学习

- 距离准则—**迹比值最大化**

$$\max_{W^TW=I} g(W) = \frac{tr(W^T S_b W)}{tr(W^T S_w W)}$$

所有**不相似**点对距离平方和

所有**相似**点对距离平方和

矩阵的迹

度量学习

- 距离准则—迹比值最大化

$$\max_{W^TW=I} g(W) = \frac{tr(W^T S_b W)}{tr(W^T S_w W)}$$

所有不相似点对距离平方和

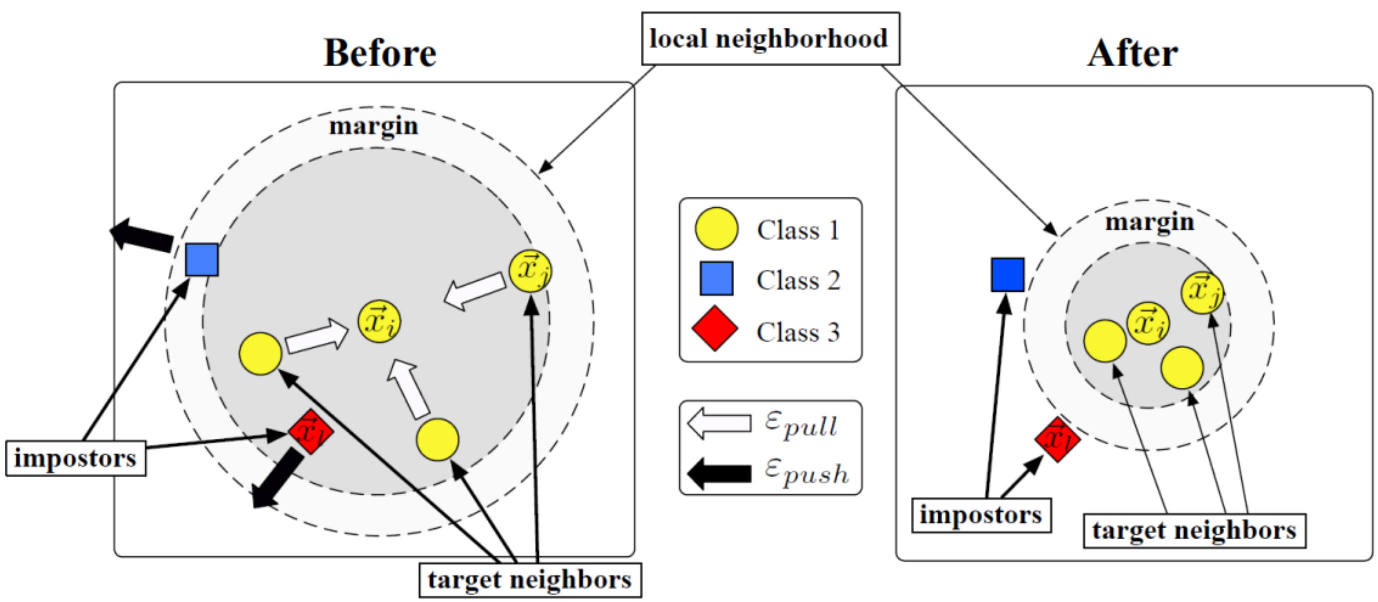
所有相似点对距离平方和

矩阵的迹

$$S_w = \sum_{x_i, x_j \in S} (x_i - x_j)(x_i - x_j)^T, S_b = \sum_{x_i, x_j \in D} (x_i - x_j)(x_i - x_j)^T$$

度量学习

- **最大间隔近邻分类（LMNN）**
  - **基本思想：**大间隔最近邻居算法的主要想法就是通过学习一种距离度量使得在一个新的转换空间中，对于一个输入x_i的k个近邻都属于同一类别，而不同类别的样本与x_i 保持一定大的距离。



度量学习

- **最大间隔近邻分类（LMNN）**
  - **学习模型**

$\xi_{ijl} \geq 0$
$M \succ= 0$

$$\min \sum_{ij} \boldsymbol{\eta_{ij}} (x_i - x_j)^T M (x_i - x_j) + \sum_{ijl} \eta_{ij} (1 - \boldsymbol{y_{il}}) \xi_{ijl}$$

$$s.t. (x_i - x_l)^T M (x_i - x_l) - (x_i - x_j)^T M (x_i - x_j) \geq 1 - \xi_{ijl}$$

$$\eta_{ij} = \begin{cases} 1, & \text{if } (\mathbf{x}_i, \mathbf{x}_j) \in \text{the same class, and } \mathbf{x}_j \in N_k(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{cases}$$

$$y_{il} = \begin{cases} 1, & \text{if } (\mathbf{x}_i, \mathbf{x}_l) \in \text{the same class} \\ 0, & \text{otherwise} \end{cases}$$

度量学习

● 最大间隔近邻分类（LMNN）

– **学习模型**

属于同一类的邻近点之间的距离要小

同一邻域内，不属于同一类的点对

$\xi_{ijl} \geq 0$

$M \succ= 0$

Relax variables

$$\min \sum_{ij} \boxed{\boldsymbol{\eta_{ij}} (x_i - x_j)^T M (x_i - x_j)} + \sum_{ijl} \boxed{\eta_{ij}(1 - \boldsymbol{y_{il}})\xi_{ijl}}$$

$$s.t.\ (x_i - x_l)^T M (x_i - x_l) - (x_i - x_j)^T M (x_i - x_j) \geq 1 - \xi_{ijl}$$

同一邻域内，不属于同一类的点之间的距离要大于“1”

度量学习

- **当前的一些主要的工作**
  - 基于弱信息（侧信息）的学习模型构建
  - 基于信息论的距离度量学习
  - 相关/相似关系学习
  - 核矩阵学习

参考文献

- 周志华.《机器学习》
- 李航. 《统计学习方法》

致 谢

- 感谢**向世明**老师的20版PPT作为原始材料
- 感谢**王锐**与**段俊贤**对本PPT的制作与修改

致谢

- 感谢向世明老师的20版PPT作为原始材料
- 感谢王锐与段俊贤对本PPT的制作与修改

Thank All of You!

(Questions?)

赫然
rhe@nlpr.ia.ac.cn

https://rhe-web.github.io/

智能感知与计算研究中心(CRIPAC)
中科院自动化研究所　模式识别国家重点实验室