



计算机视觉中的机器学习方法

董秋雷

中国科学院自动化研究所
模式识别国家重点实验室



1

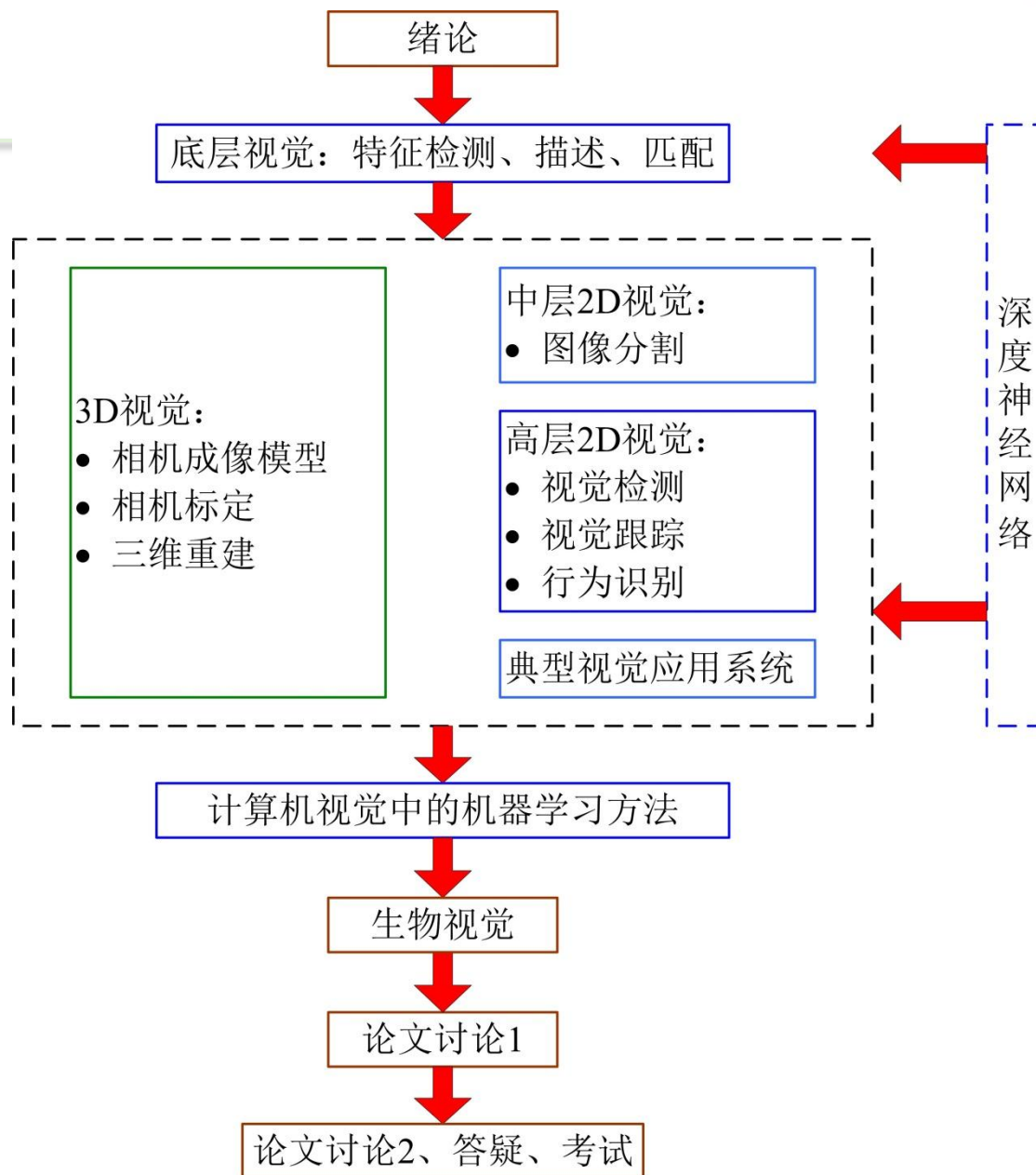
背景内容

2

计算机视觉中的机器学习方法

3

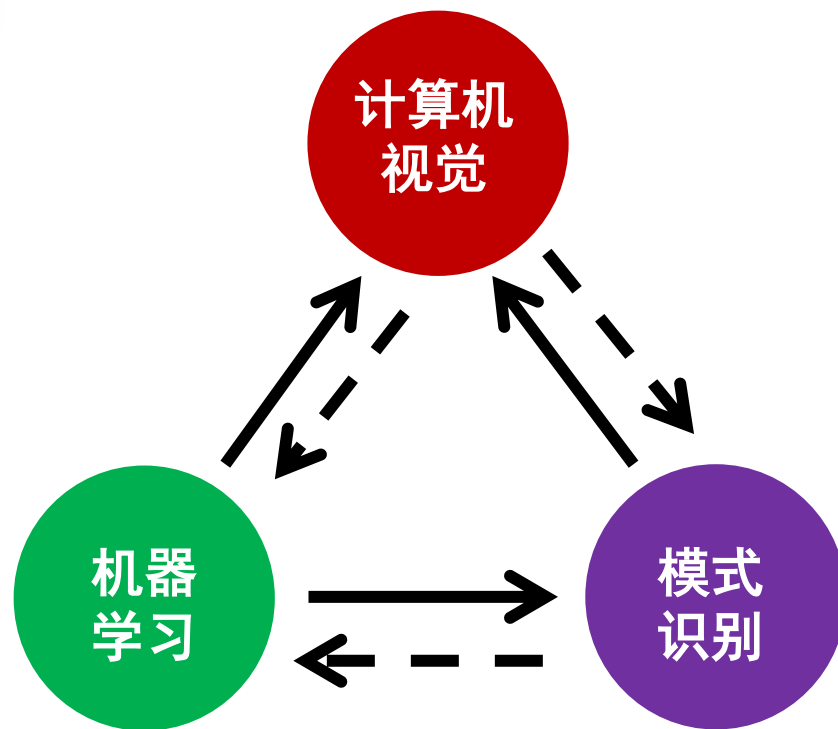
小节



计算机视觉与其他学科的关系

深度
学习

大数据



1

背景内容

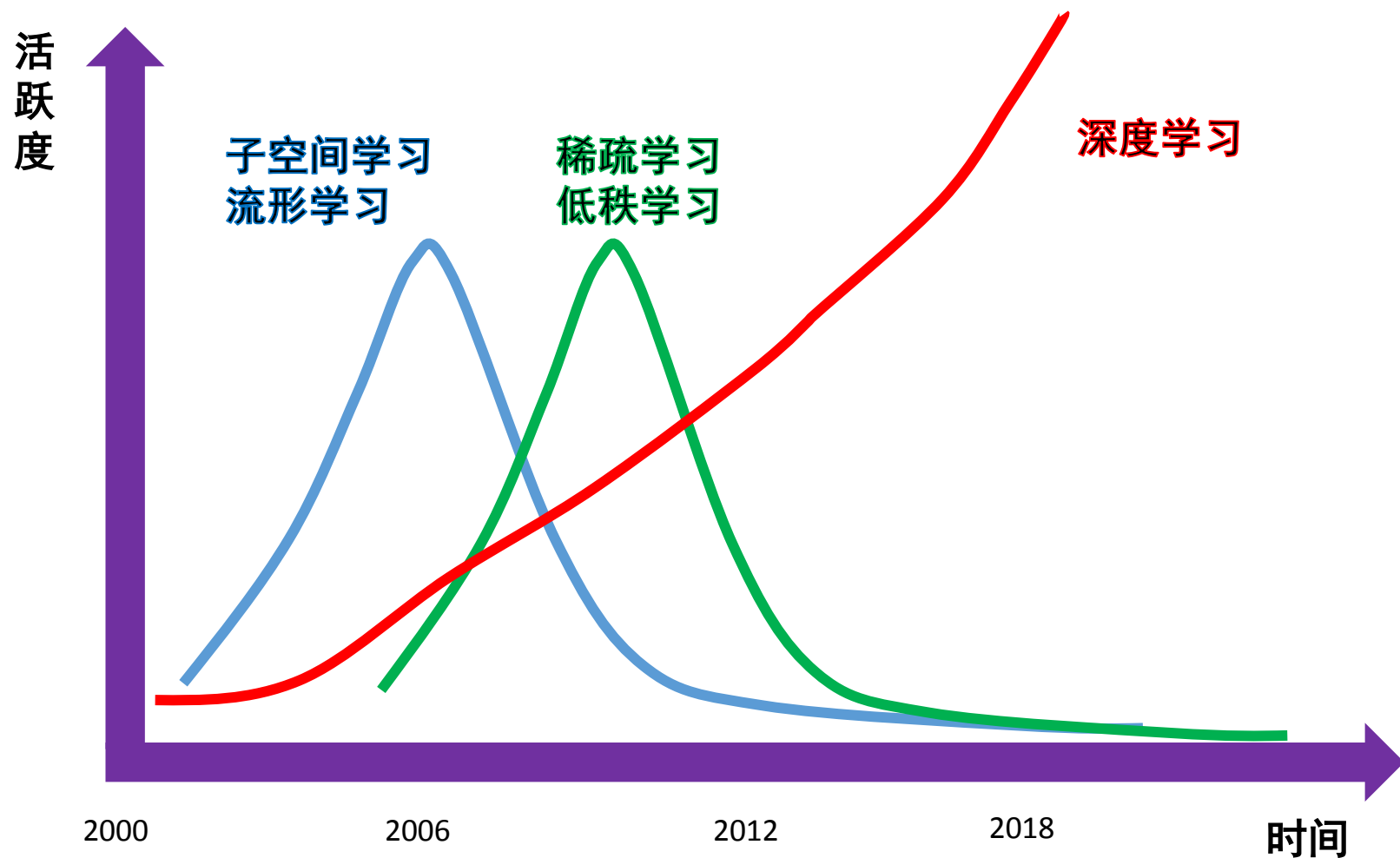
2

计算机视觉中的机器学习方法

3

小节

计算机视觉中机器学习算法活跃度



子空间分析

问题

下面的数字都是多少？

3	2	7	3	8	6	9	0	5	6	0	7	6	1	8	7	9
7	4	9	8	0	9	4	1	4	4	6	0	4	5	6	7	0
1	1	7	8	0	2	6	7	8	3	9	0	4	6	7	4	6
1	7	1	1	6	3	0	2	9	3	1	1	0	4	9	2	0
4	1	6	3	4	3	9	1	3	3	8	5	4	7	7	4	2
1	9	9	6	0	3	7	2	8	2	9	4	4	6	4	9	7
1	0	3	2	3	5	9	1	7	6	2	8	2	2	5	0	7
8	3	6	1	0	3	1	0	0	1	1	2	7	3	0	4	6
9	3	0	7	1	0	2	0	3	5	4	6	5	8	6	3	7
2	2	3	3	6	4	7	5	0	6	2	7	9	8	5	9	2
2	5	3	9	3	9	0	5	9	6	5	7	4	1	3	4	0

进一步的问题

怎样处理大数据量、高维数、非结构化的数据呢？

- 直接在高维数据上处理；
- 降维后再对低维数据进行处理；
- 升到更高维度上再进行处理；

进一步的问题

- 直接在高维数据上处理：
 - 维数灾难 (Curse of Dimensionality) : 满足一定统计指标(期望与方差)的模型(精度), 需要的样本数量将随着维数的增加, 指数增长(或模型复杂程度, 或模型表示长度指数增长)。
 - 特征和特征之间是冗余的、信息量是不一样的。

子空间分析

- 降维后再对低维数据进行处理：
- 子空间分析
 - 把高维空间中松散分布的样本，通过线性或非线性变换压缩到一个低维的子空间中，在低维的子空间中使样本的分布更紧凑、更有利于分类，同时使计算复杂度减少。

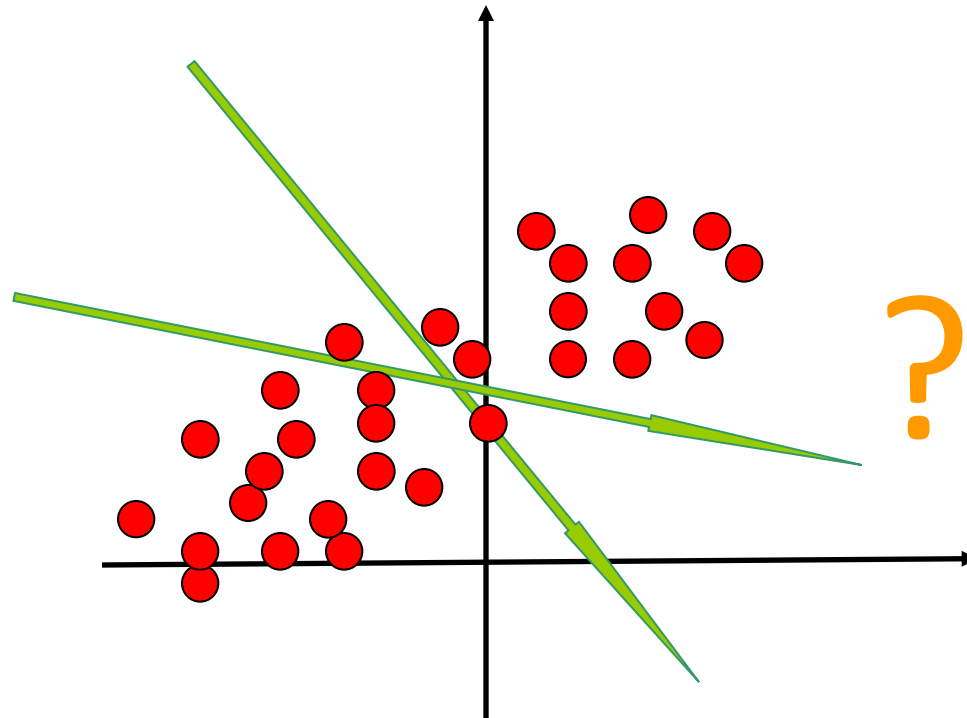
经典子空间分析算法

- 主成分分析方法 (Principal Component Analysis, PCA)
- 独立成分分析方法 (Independent Component Analysis, ICA)
- 线性判别分析方法 (Linear Discriminant Analysis, LDA)

PCA

- PCA: Principal Component Analysis, 主成分分析。
- 又名: Karhunen-Loève (K-L) 变换、Hottelling变换
- 基本思想:
 - 将多个变量通过线性变换以选出较少个重要变量, 这些新变量尽可能保持原有的信息。
 - 换成数学表述为: 寻找投影映射 P , 使得样本从 N 维降到 m 维 ($m < N$), 同时最小化平方误差。

PCA



PCA的发展历史

- Karl Pearson于1901提出；
- Harold Hotelling于1933年加以发展；
- Kari Karhunen和Michel Loève提出Karhunen-Loève theorem；
- Turk和Pentland于1991年正式提出eigenface的概念。



PCA

- 变量定义：
- 训练样本集： $X = [x_1, x_2, \dots, x_n]$ ，其中 x_i 是 N 维向量
- 样本均值： $\mu = \frac{1}{n} \sum_{i=1}^n x_i$
- 离散度矩阵： $S = \sum_{i=1}^n (x_i - \mu)(x_i - \mu)^T$
- 投影后的低维数据： $y_i = P^T (x_i - \mu)$ ，其中 P 是 $N \times m$ 维投影矩阵

PCA



怎样使新变量尽可能地保持原有的信息呢？

重构误差最小 (least squares reconstruction)

样本: $X = [x_1, x_2, \dots, x_n]$

投影: $y_i = P^T (x_i - \mu)$, 其中 P 是 $N \times m$ 维矩阵

样本重构: $\hat{x}_i = P y_i$

重构误差: $e = \sum_{i=1}^n \|(x_i - \mu) - \hat{x}_i\|^2$

PCA

- 最小化重构误差，等价于如下优化问题：

$$P = \arg \max_P |P^T S P|$$

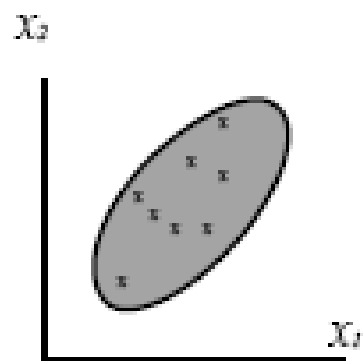
- 特征值分解： $S v_i = \lambda_i v_i, i = 1, 2, \dots, n$

其中 $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ ， λ_i 与 v_i 分别是特征值和对应的特征向量

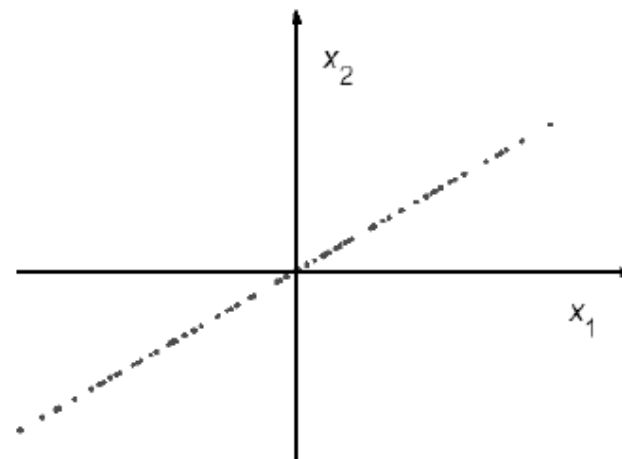
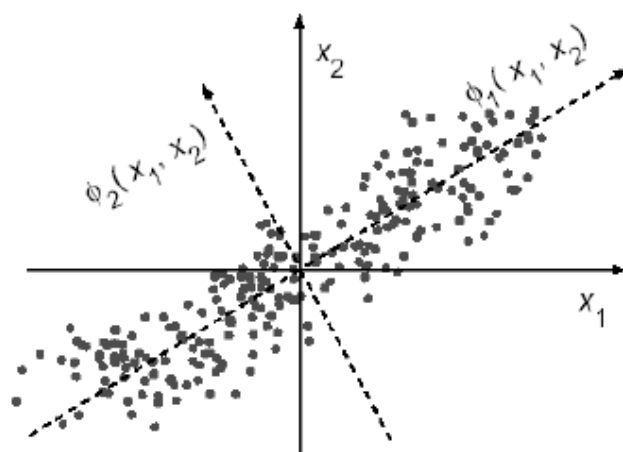
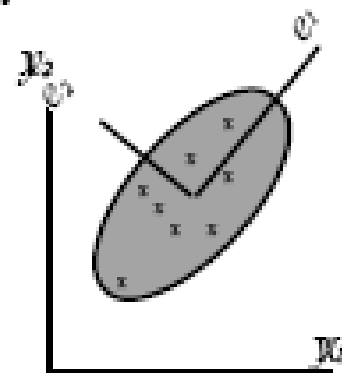
- 最大的一组特征值对应的特征向量组成的子空间即为所求：

$$P = [v_1, v_2, \dots, v_m]$$

二维情况的例子



PCA



计算步骤

- 对于给定的样本矩阵 $X = [x_1, x_2, \dots, x_n]$, 其中 x_i 是 N 维向量
 1. 计算样本均值 $\mu = \frac{1}{n} \sum_{i=1}^n x_i$
 2. 计算离散度矩阵 $S = \sum_{i=1}^n (x_i - \mu)(x_i - \mu)^T$
 3. 将离散度矩阵进行特征值分解, 取最大的 m 个特征值和相应的特征向量 $P = [v_1, v_2, \dots, v_m]$, 其中 P 是 $N \times m$ 维矩阵。
 4. 实现降维 $y_i = P^T (x_i - \mu)$

PCA的应用

- 特征脸 (Eigenfaces)

- Sirovich & Kirby 1987

- “Low dimensional Procedure for the characterization of human faces” Journal of the Optical Society of America

- Turk & Pentland 1991

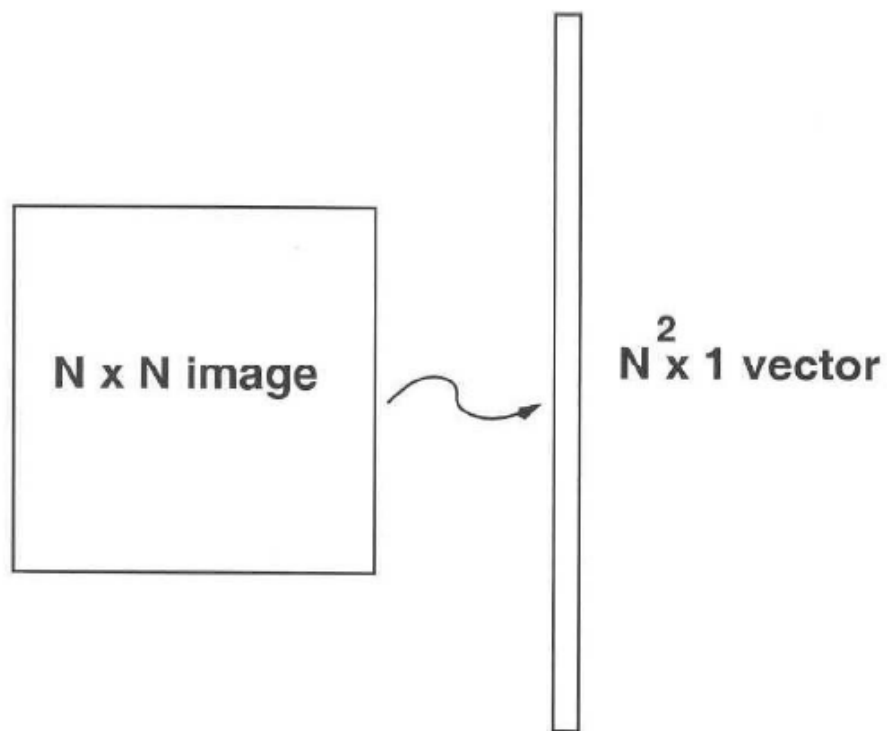
- “Face recognition using Eigenfaces” IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

Eigenfaces



Eigenfaces

人脸图像可以表示为一个向量



Eigenfaces

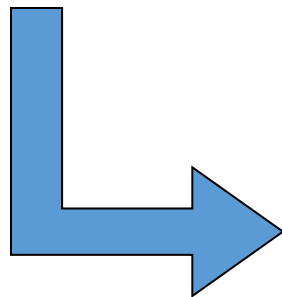
- 1. 给定训练图像矩阵 $X = [x_1, x_2, \dots, x_n]$, 其中 x_i 表示一副训练图像。
- 2. 计算样本均值 $\mu = \frac{1}{n} \sum_{i=1}^n x_i$
- 3. 计算离散度矩阵 $S = \sum_{i=1}^n (x_i - \mu)(x_i - \mu)^T$
- 4. 将离散度矩阵进行特征值分解, 取最大的 m 个特征值和相应的特征向量 $P = [v_1, v_2, \dots, v_m]$ 。

Eigenfaces

- 1. 给定训练图像矩阵 $X = [x_1, x_2, \dots, x_n]$, 其中 x_i 表示一副训练图像。
- 2. 计算样本均值 $\mu = \frac{1}{n} \sum_{i=1}^n x_i$
- 3. 计算离散度矩阵 $S = \sum_{i=1}^n (x_i - \mu)(x_i - \mu)^T$
- 4. 将离散度矩阵进行特征值分解, 取最大的 m 个特征值和相应的特征向量 $P = [v_1, v_2, \dots, v_m]$ 。



原始人脸图像



特征脸 (Eigenface)

人脸表述与识别

- 人脸表述：基于得到的投影矩阵 P ，每张人脸可以由一个 m 维向量表示

$$y_i = P^T (x_i - \mu)$$

- 人脸识别：

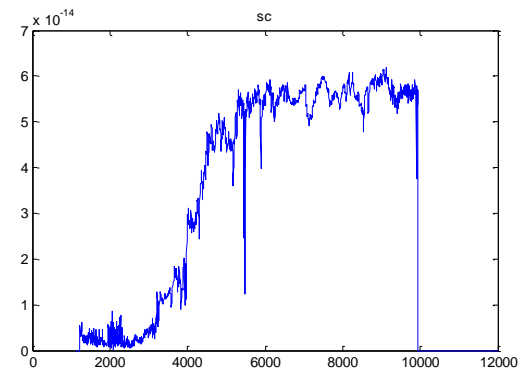
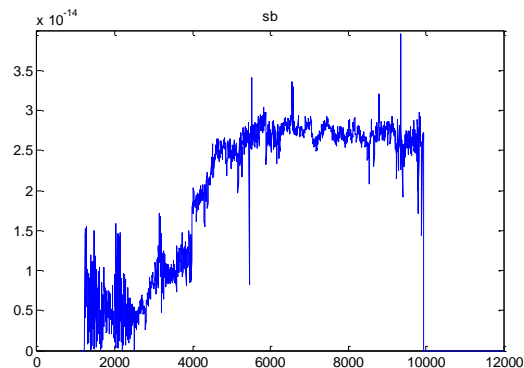
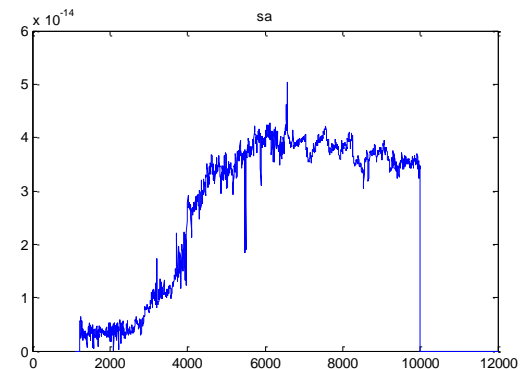
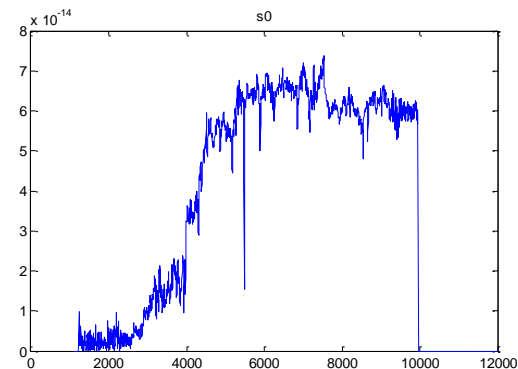
- 对于一张输入的人脸图像，计算它的低维表述

$$y_{input} = P^T (x_{input} - \mu)$$

- 计算 $k = \arg \min_i \|y_{input} - y_i\|$

PCA在光谱方面的应用

- 四个正常星系的光谱维数为2726，横轴为波长，纵轴为流量



PCA在光谱方面的应用

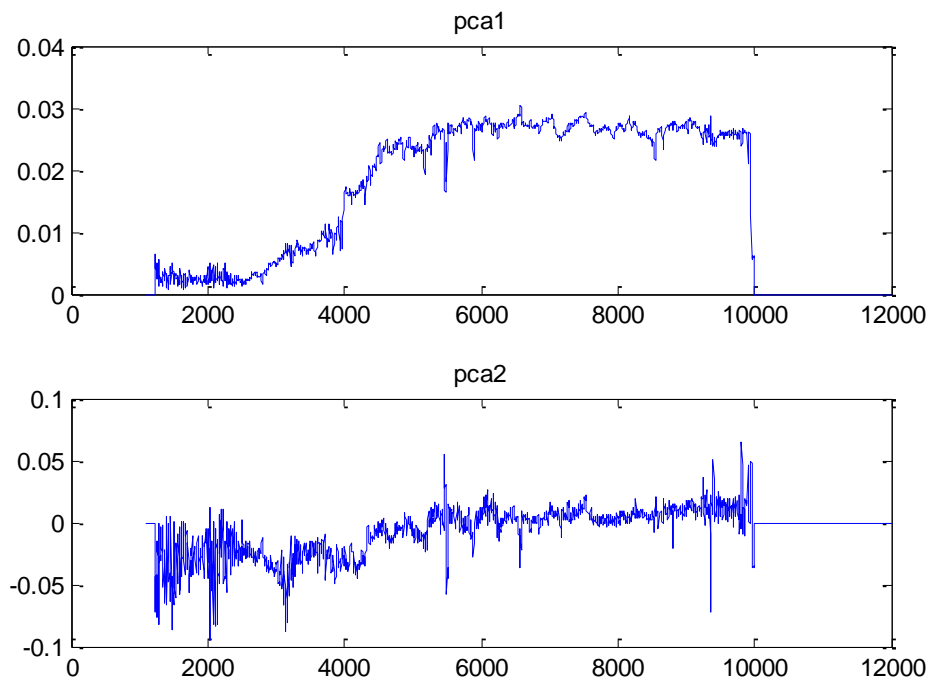
$$DD = \begin{bmatrix} 3.9737 & 0 & 0 & 0 \\ 0 & 0.0139 & 0 & 0 \\ 0 & 0 & 0.0092 & 0 \\ 0 & 0 & 0 & 0.0032 \end{bmatrix} \quad D = \begin{bmatrix} 3.9737 & 0 \\ 0 & 0.0139 \end{bmatrix}$$

特征值矩阵

方差贡献率大于0.995的
两个特征值

PCA在光谱方面的应用

- 投影基向量



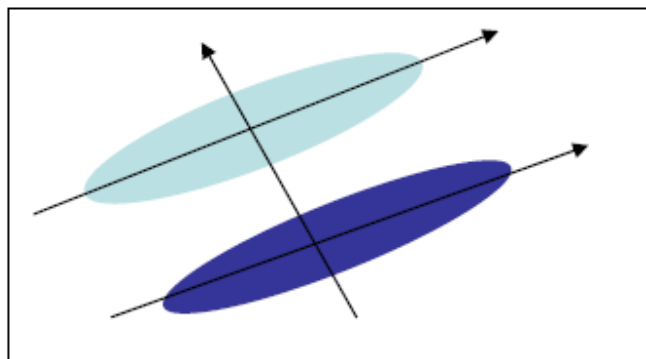
PCA在光谱方面的应用

- 4个光谱向量的二维pca表示

$$spec = \begin{bmatrix} 0.9979 & 0.9968 & 0.9953 & 0.9968 \\ 0.0465 & -0.0206 & -0.0871 & 0.0611 \end{bmatrix}$$

PCA

- PCA对于椭球状分布的样本集有很好的效果，学习所得的主方向就是椭圆的主轴。
- PCA 是一种非监督的算法，能找到很好代表所有样本的方向，但这个方向对于分类未必是最有利的。



Robust PCA

PCA: 样本: $X = [x_1, x_2, \dots, x_n]$

投影: $y_i = P^T (x_i - \mu)$, 其中 P 是 $N \times m$ 维矩阵

样本重构: $\hat{x}_i = P y_i$

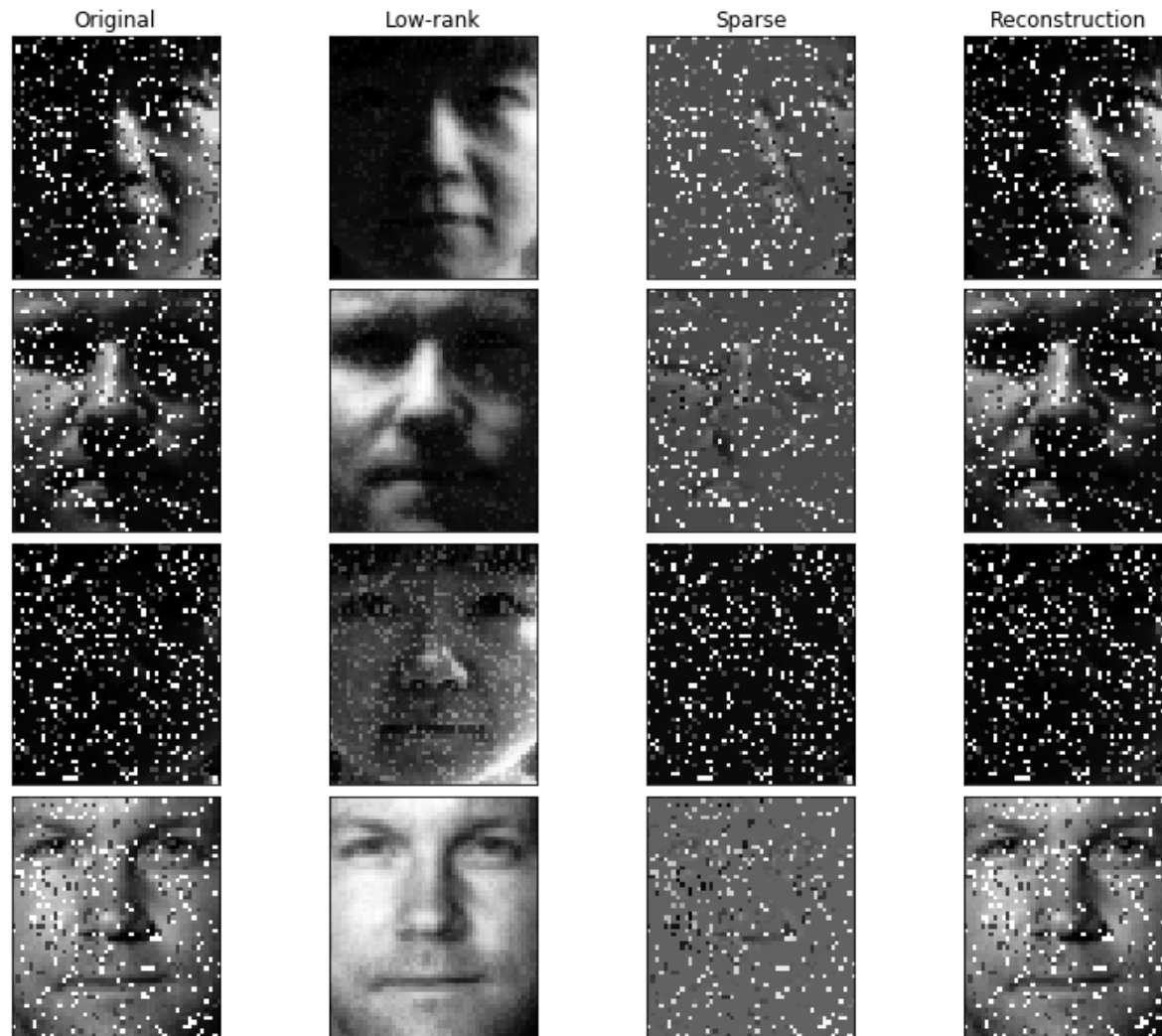
重构误差: $e = \sum_{i=1}^n \|(x_i - \mu) - \hat{x}_i\|^2$

$$\begin{array}{ccc}
 \min_{\hat{X}} \|E\|_F^2 & \min_{\hat{X}, E} \|\hat{X}\|_{rank} + \lambda \|E\|_0 & \min_{\hat{X}, E} \|\hat{X}\|_* + \lambda \|E\|_1 \\
 \text{s.t. } rank(\hat{X}) \leq k & \text{s.t. } X = \hat{X} + E & \text{s.t. } X = \hat{X} + E \\
 X = \hat{X} + E & &
 \end{array}$$

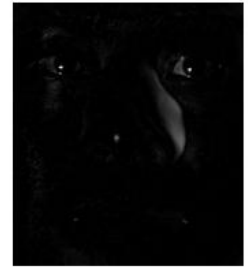
Robust PCA

Candès, Emmanuel J, Li X, Ma Y, et al. Robust principal component analysis?.
Journal of the ACM, 2011, 58(3):1-37.

Robust PCA

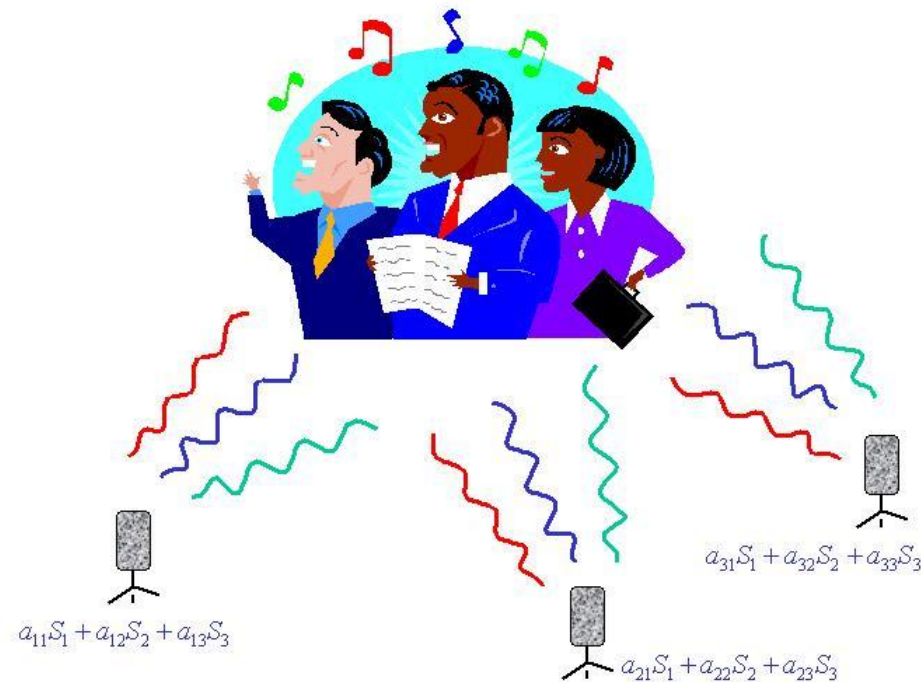


Robust PCA

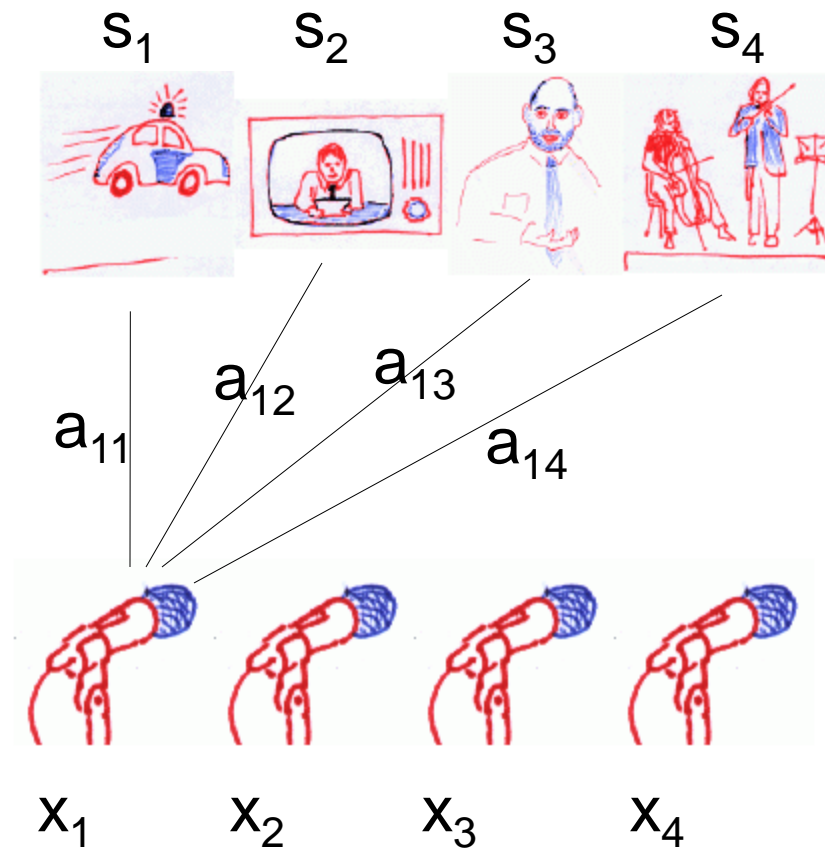


ICA


- 20世纪80年代，盲信号分离（Blind Source Separation）：
鸡尾酒会问题



ICA



ICA

- ICA，独立成分分析：指从多个源信号的线性混合信号中分离出源信号的技术。
- ICA假设：源信号统计独立。
- 模型： $x = A * s$ 或 $x = \sum_{i=1}^n a_i s_i$

- 求解： $u = W * x = W * A * s$

ICA发展历史

- 起源：20世纪80年代，盲信号分离（Blind Source Separation）

例子：鸡尾酒会问题

- 正式提出：1994年，P. Comon

“Independent Component Analysis --- a new concept?”
Signal Processing

- 应用：1998年，Bartlett, et al 应用到人脸分析。

算法（课后练习）

❖ InfoMax 算法（信息极大化算法）：1995年, Bell & Sejnowski

An information maximization approach to blind separation and blind deconvolution,

Neural Computation

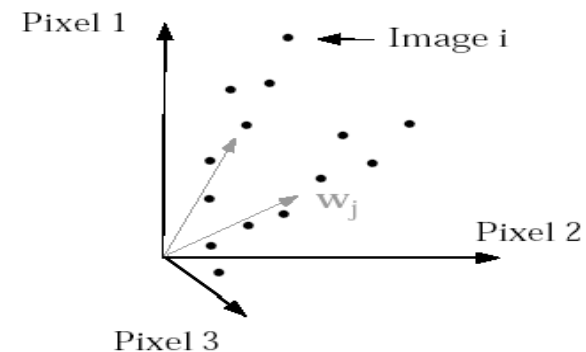
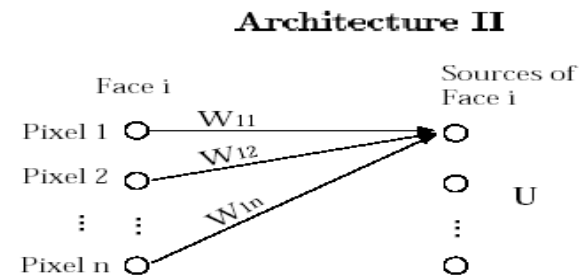
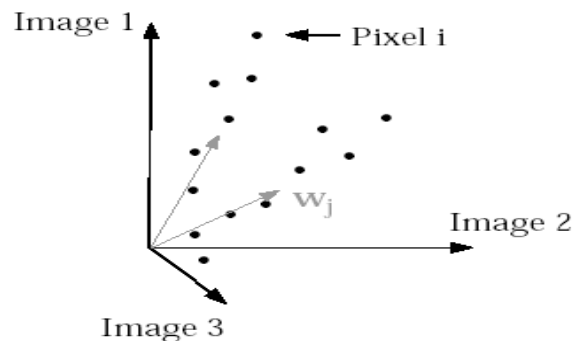
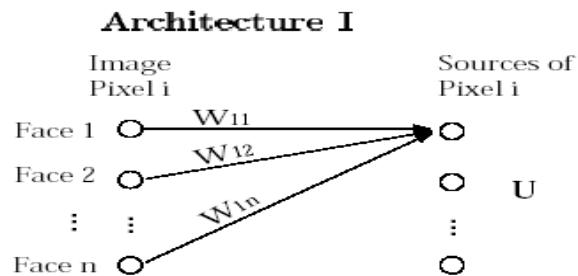
❖ FastICA 算法（固定点算法）：1997年, A. Hyvärinen & E. Oja

A fast fixed-point algorithm for independent component analysis.

Neural Computation

人脸的ICA表示方法

- basis image (独立基图像)
- factorial code (因子表示)



ICA在人脸识别中的应用

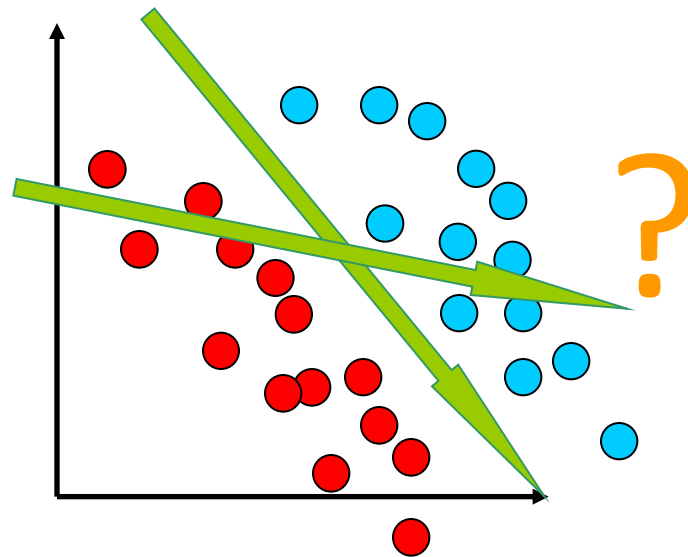
• 结构1:

$$\begin{array}{c} \text{Image of a man's face} \end{array} = b_1 * \begin{array}{c} \mathbf{u}_1 \\ \text{Image of eyes} \end{array} + b_2 * \begin{array}{c} \mathbf{u}_2 \\ \text{Image of mouth} \end{array} + \dots + b_n * \begin{array}{c} \mathbf{u}_n \\ \text{Image of nose} \end{array}$$

• 结构2:

$$\begin{array}{c} \text{Image of a man's face} \end{array} = u_1 * \begin{array}{c} \mathbf{a}_1 \\ \text{Image of a face with different expression} \end{array} + u_2 * \begin{array}{c} \mathbf{a}_2 \\ \text{Image of a face with different expression} \end{array} + \dots + u_n * \begin{array}{c} \mathbf{a}_n \\ \text{Image of a face with different expression} \end{array}$$

LDA



LDA

- LDA (Linear Discriminant Analysis), 又称 Fisher Discriminant Analysis, 是一种监督的维数约简方法.
- Fisher判别原则: 寻找投影 W , 使得投影后的样本类内散度最小, 而类间散度最大。

LDA

- 假设有C类样本，第*i*类样本个数是 N_i ， μ_i 是第*i*类样本的均值， x_j^i 是第*i*类中第*j*个样本。
- 类内散度 $S_W = \sum_{i=1}^C \sum_j (x_j^i - \mu_i)(x_j^i - \mu_i)^T$
- 类间散度 $S_B = \sum_{i=1}^C N_i (\mu_i - \mu)(\mu_i - \mu)^T$
- 总体散度 $S_t = \sum_i (x_i - \mu)(x_i - \mu)^T$
- 容易证明 $S_t = S_W + S_B$

LDA

- Fisher 准则可以转化为最优化问题：寻找投影映射 w 使得目标函数最大

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

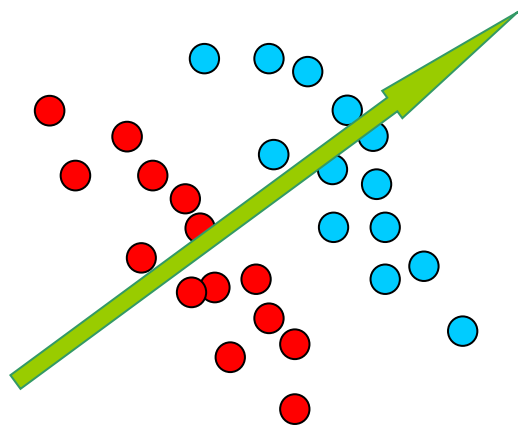
求解

- 可以转化成广义特征值问题：

$$S_B w = S_W w \Lambda$$

- 当 S_W 非奇异时，上述问题等价于 $S_W^{-1} S_B$ 的特征值求解问题：

$$S_W^{-1} S_B w = w \Lambda$$



奇异问题

- 当样本个数小于特征维数时（小样本问题）， S_W 是奇异的，无法求解。
- 解决方案：
 1. 用PCA对样本进行降维，使得 S_W 非奇异。
 2. 直接在 S_W 的零空间求解最优投影。
 3. 扰动法：在 S_W 的对角上加小的扰动，使其非奇异。

线性方法的不足

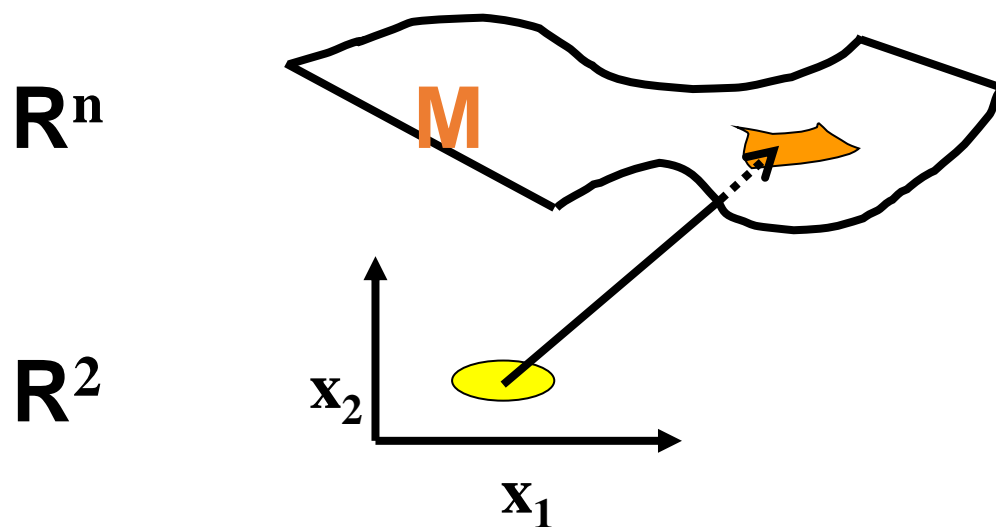
- 现实中数据的有效特性往往不是特征的线性组合。



Manifold Learning 流形学习

什么是流形？

- **定义1**：如果一个 N 维的拓扑空间 M 内的任意一点都存在一个邻域 $U \in M$ 使得该邻域是 N 维欧氏空间的同胚，则这个拓扑空间 M 被称为流形。



流形学习的数学基础

- 参考文献:

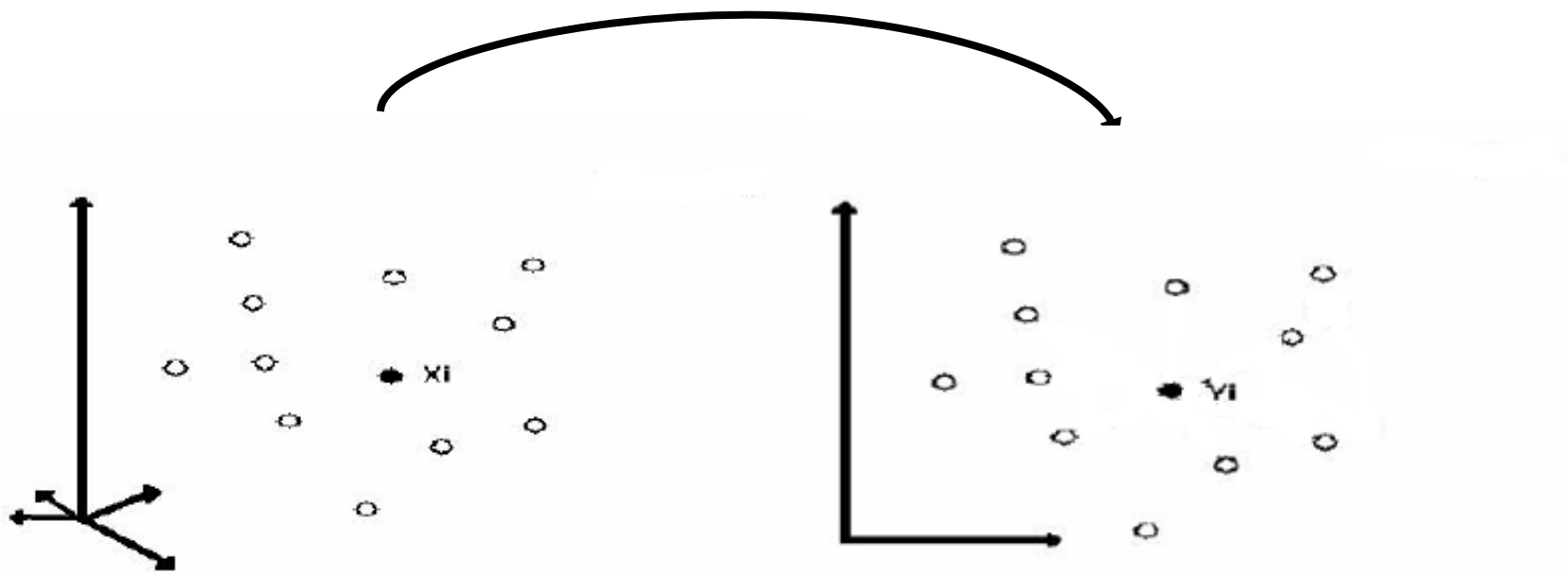
- 陈省身, 陈维桓, 微分几何讲义. 北京大学出版社, 1983
- 陈维桓, 微分流形初步(第二版). 高等教育出版社, 2001

什么是流形学习?

- **定义2**: 令 Y 是包含在欧氏空间 R^d 的 d 维域, $f: Y \rightarrow R^N$ 为光滑嵌入, 其中 $N > d$ 。数据点 $\{y_i\} \subset Y$ 是随机生成的, 经 f 映射, 形成观察空间的数据 $\{x_i = f(y_i)\} \subset R^N$ 。一般称 Y 为隐空间, $\{y_i\}$ 为隐数据。流形学习的目标是要从观察数据 $\{x_i\}$ 中重构 f 和 $\{y_i\}$ 。

- 流形是线性子空间的一种非线性推广.
- 流形是一个局部可坐标化的拓扑空间.

什么是流形学习?



流形学习的可行性

- 1 许多高维采样数据都是由少数几个隐含变量所决定的，如人脸采样由光线亮度，人离相机的距离，人的头部姿势，人的脸部肌肉等因素决定.
- 2 从认知心理学的角度，心理学家认为人的认知过程是基于认知流形和拓扑连续性的.

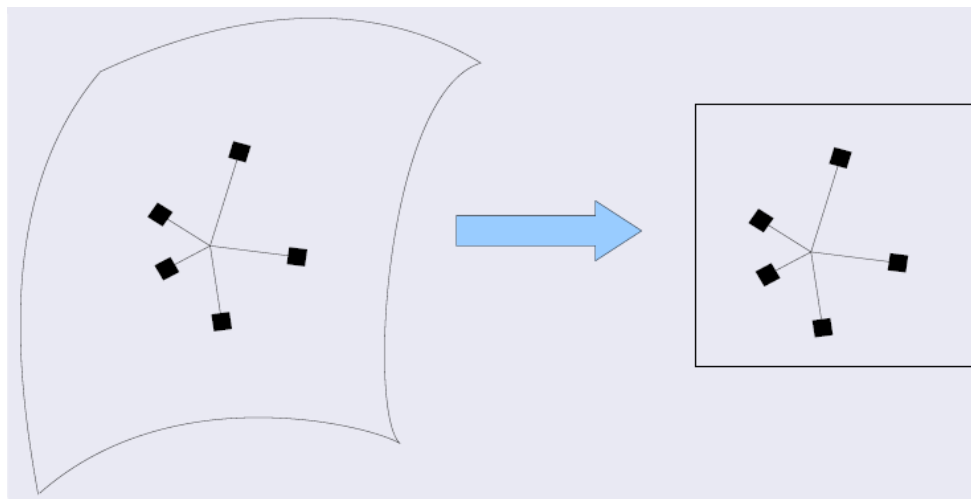


算法简介

- Sciences 2000年:
 - Tenenbaum等人: Isomap
 - Roweis和Saul: LLE
- NIPS, 2001年:
 - M.Belkin和P.Niyogi: Laplacian Eigenmaps
- NIPS & ICCV 2003:
 - Xiaofei He等人: LPP
- PAMI 2007:
 - Graph Embedding and Extensions: A General Framework for Dimension Reduction

LLE

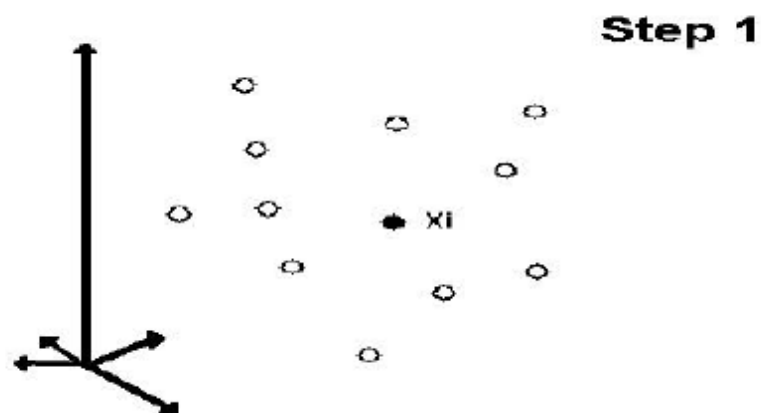
- LLE (Locally Linear Embedding) : 强调在样本集结构不满足全局线性结构时, 样本空间与内在低维子空间之间在局部意义下的结构可以用线性空间近似。



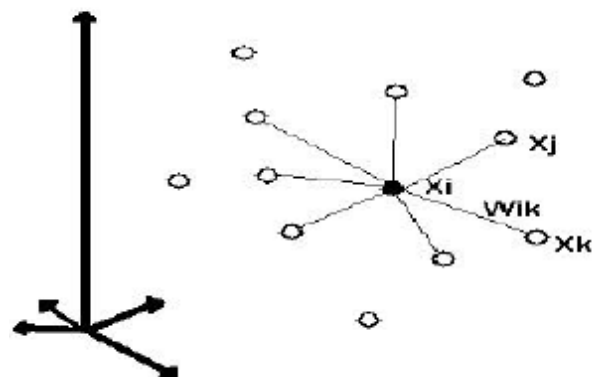
$$x_i \approx \sum W_{ij} x_j$$

LLE

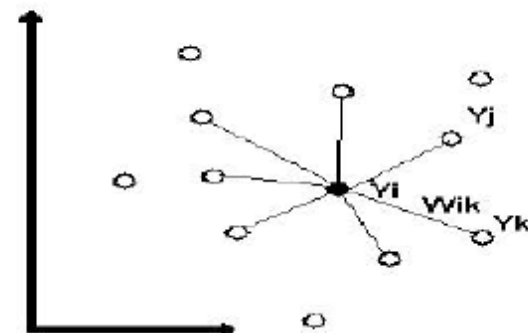
●邻接图



Step 2



Step 3



LLE

权值计算：

$$\xi(W) = \sum_i |x_i - \sum_j W_{ij} x_j|^2$$

学习目标：

在低维空间中保持每个邻域中的权值不变，即假设嵌入映射在局部是线性的条件下，最小化重构误差。

$$\xi(y) = \sum_i |y_i - \sum_j W_{ij} y_j|^2$$

LLE

• 流程图:

Step 1: 构建邻域。对于原始空间任一给定样本点，用 **K** 近邻法得到它的一组邻域点。

Step 2: 计算权值。在第二步用权值 W_{ij} 描述原始空间任一点与其邻域的关系。权值 W_{ij} 是使得样本点 x_i 用它的相邻点 x_j 重构误差最小的解：

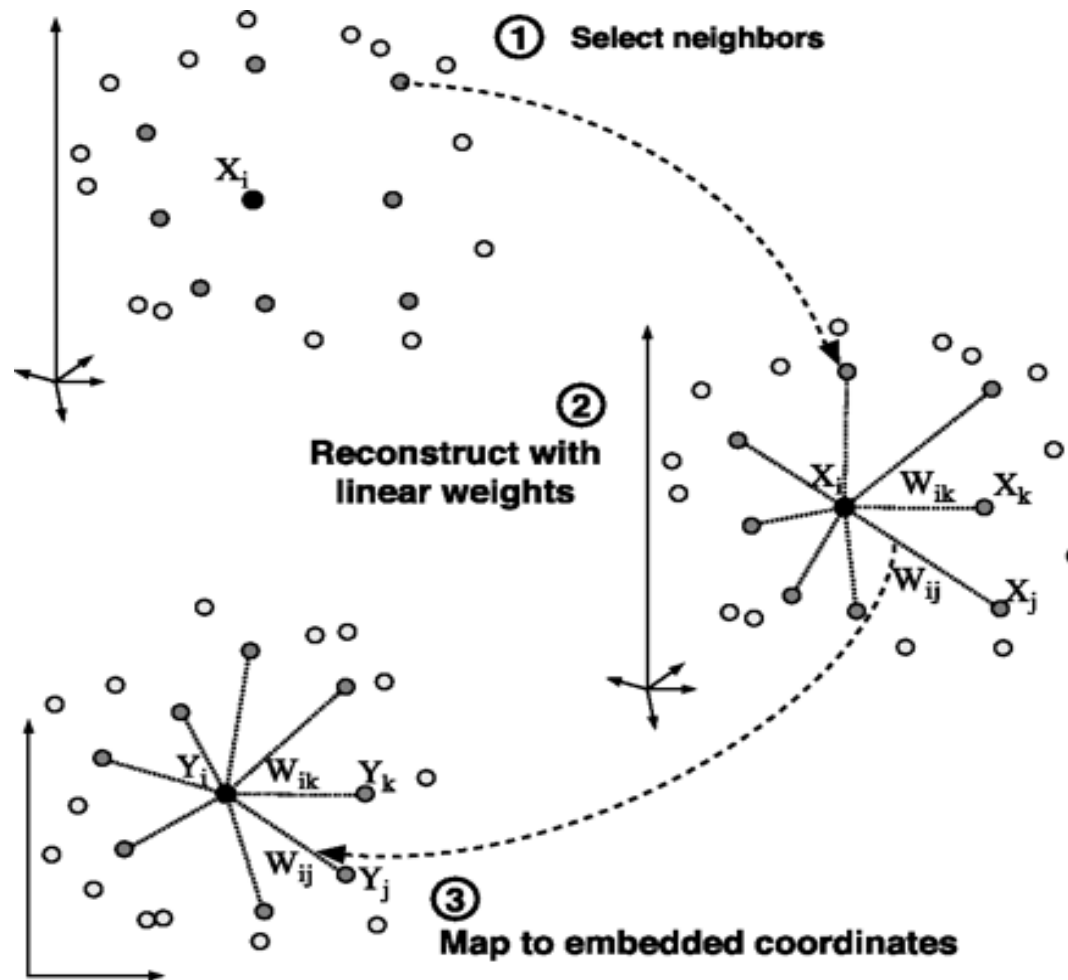
$$\xi(W) = \sum_i |x_i - \sum_j W_{ij} x_j|^2$$

Step 3: 嵌入。最后的嵌入通过最小化误差来保留尽可能多的原空间几何性质：

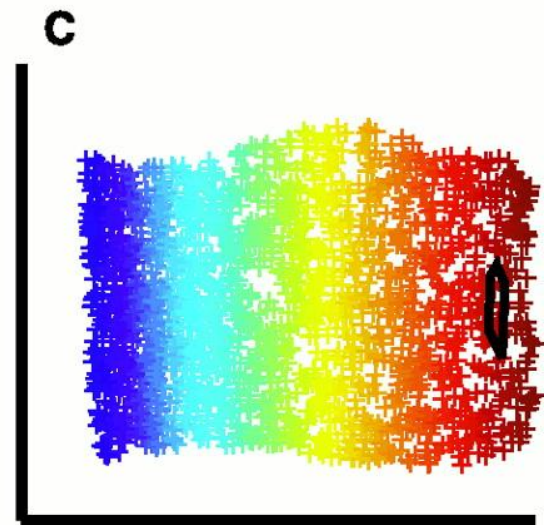
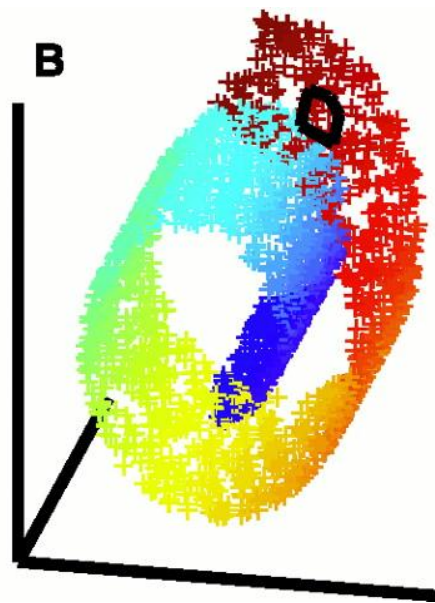
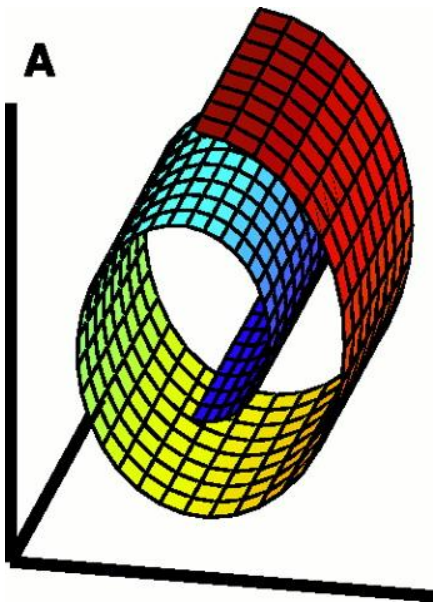
$$\xi(y) = \sum_i |y_i - \sum_j W_{ij} y_j|^2$$

这里 W 是第二步计算的权值， y_i 和 y_j 是样本点在嵌入空间的投影

LLE

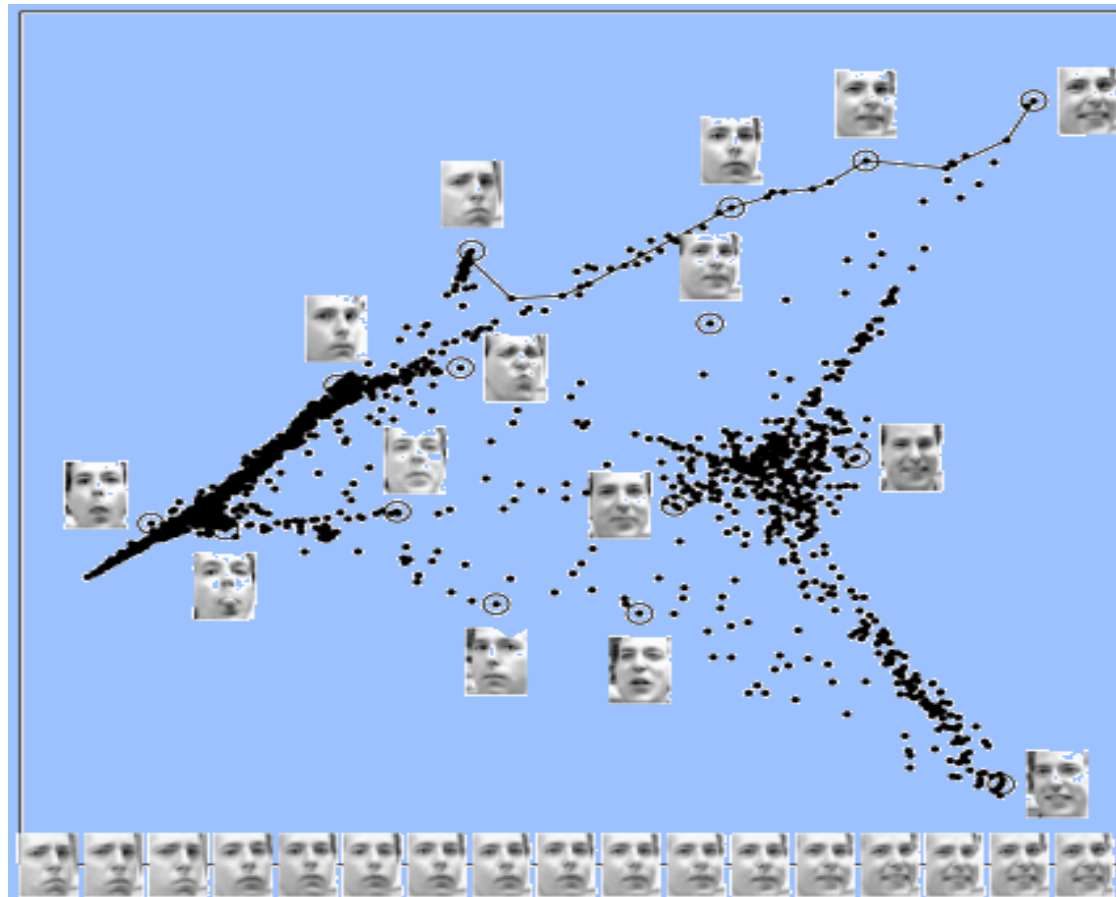


LLE的应用



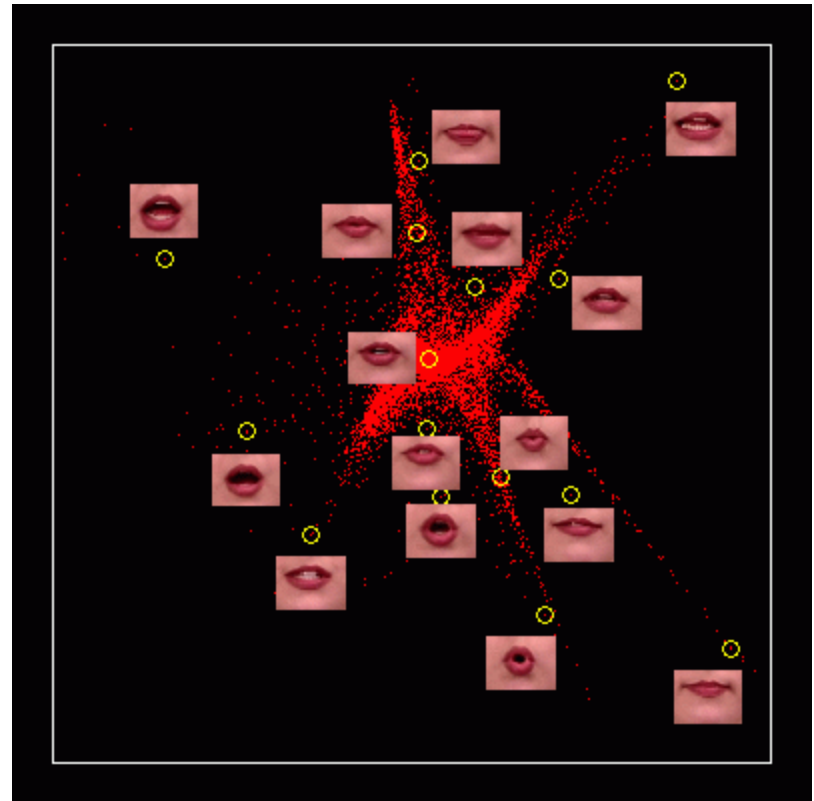
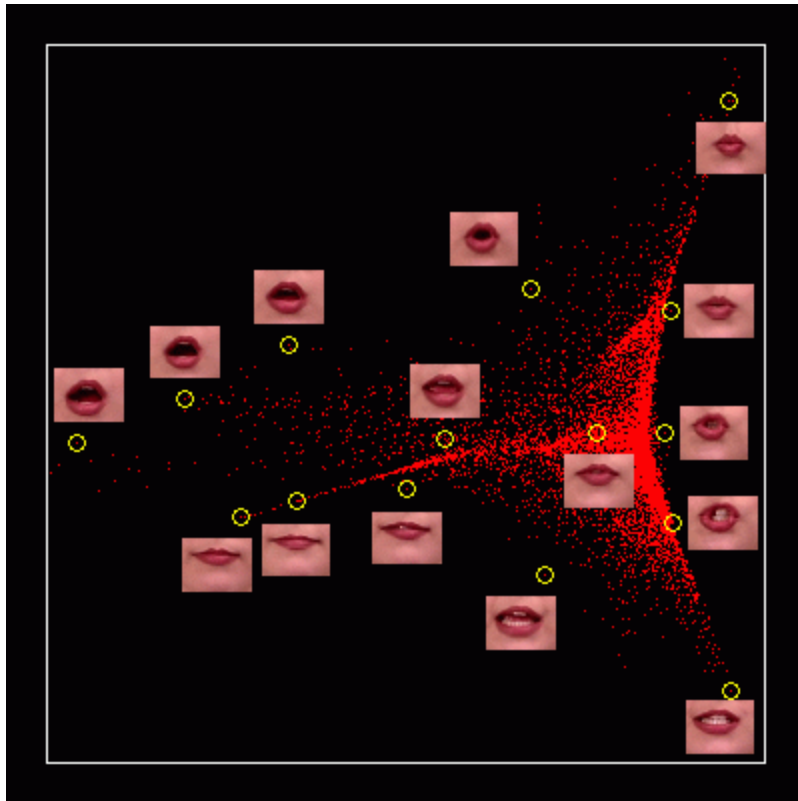
LLE的应用

- 人脸图像在2D流形空间的投影



LLE的应用

● 嘴唇图像在2D流形空间的投影



LLE的优点

- LLE算法可以学习任意维数的低维流形.
- LLE算法中的待定参数很少, K 和 d .
- LLE算法中每个点的近邻权值在平移, 旋转, 伸缩变换下是保持不变的.
- LLE算法有解析的整体最优解, 不需迭代.
- LLE算法归结为稀疏矩阵特征值计算, 计算复杂度相对较小, 容易执行.

LLE的不足

- LLE算法要求所学习的流形只能是不闭合的且在局部是线性的.
- LLE算法要求样本在流形上是稠密采样的.
- LLE算法中的参数 K , d 有过多的选择.
- LLE算法对样本中的噪音很敏感.

Isomap (等距映射)

- J. Tenenbaum, V. Silva and K. Langford

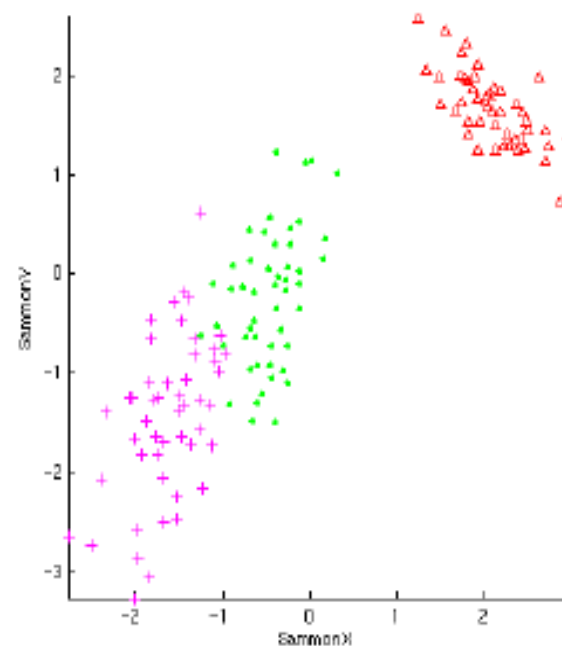
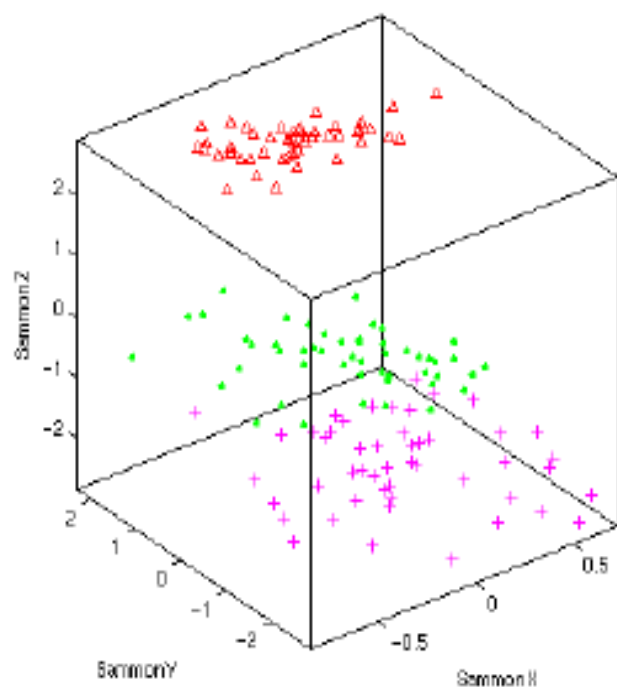
A global geometric framework for nonlinear dimensionality reduction

Science 2000

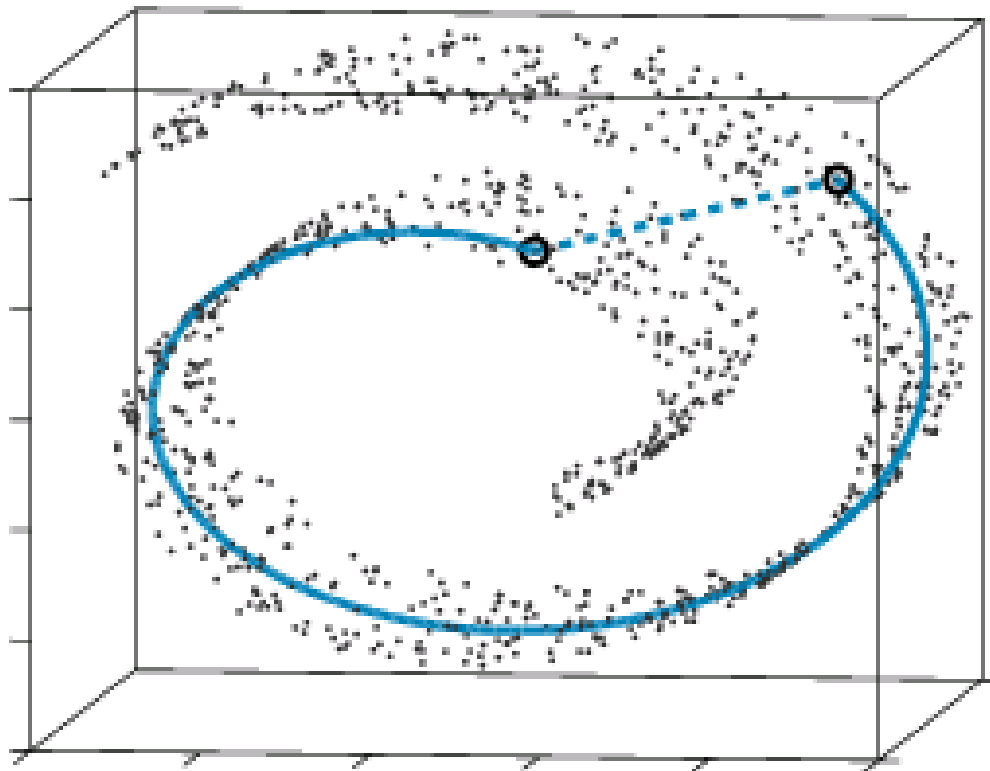
多维尺度变换 (MDS)

- MDS 是一种非监督的维数约简方法.
- MDS的基本思想: 约简后低维空间中任意两点间的距离应该与它们在原始空间中的距离相同.
- MDS的求解: 通过适当定义准则函数来体现在低维空间中对高维距离的重建误差, 对准则函数用梯度下降法求解, 对于某些特殊的距离可以推导出解析解法.

MDS的示意图



MDS的失效



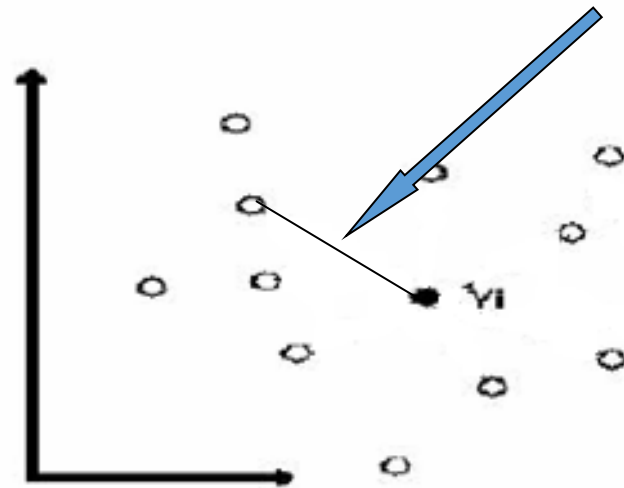
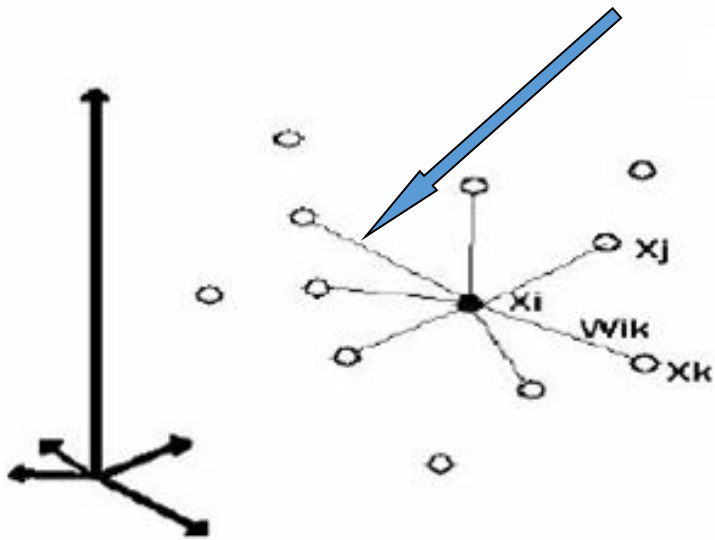
Isomap

- 主要思想：

建立在多维尺度变换(MDS)的基础上，力求保持数据点的内在几何性质，即保持两点间的测地距离，不是欧氏距离。

- $\text{Isomap} = \text{MDS} + \text{测地距离}$

Isomap



Isomap

流程图:

Step 1: 在样本集上构建近邻图 G 。如果样本 i 和 j 之间距离小于某个阈值,或者他们为 k -近邻, 则连接 i 和 j

Step 2: 计算样本两两之间测地距离 (用**Dijkstra**算法), 建立测地距离矩阵 $D_G = d_G(x_i, x_j)$

Step 3: 利用**MDS**算法构造内在 d 维子空间, 最小化下式

$$E = \| \tau(D_G) - \tau(D_Y) \|_{L^2}$$

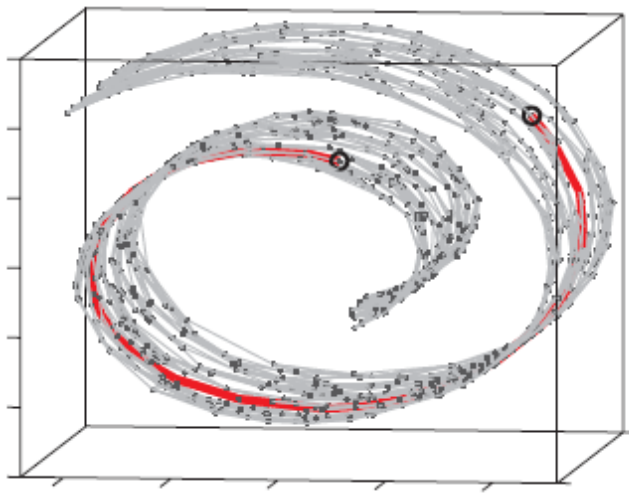
矩阵变换算子 $\tau(D) = -HSH/2$ 将距离转换成**MDS**所需内积形式, 其中 S 是平方距离矩阵 $\{S_{x_i x_j} = D_{x_i x_j}^2\}$ 是集中矩阵

$$\{H_{x_i x_j} = \delta_{x_i x_j} - 1/N\}$$

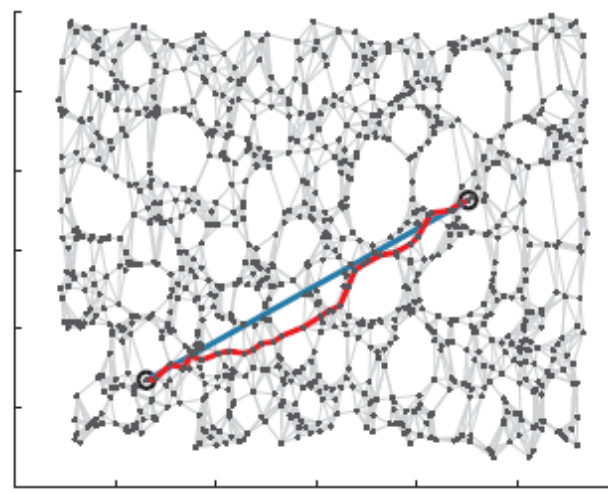
上式的最小值可以通过求矩阵 $\tau(D_G)$ 的 d 个最大特征值对应的特征向量来实现

Isomap的应用

- Swiss Roll在2D流形空间的投影



3维数据集

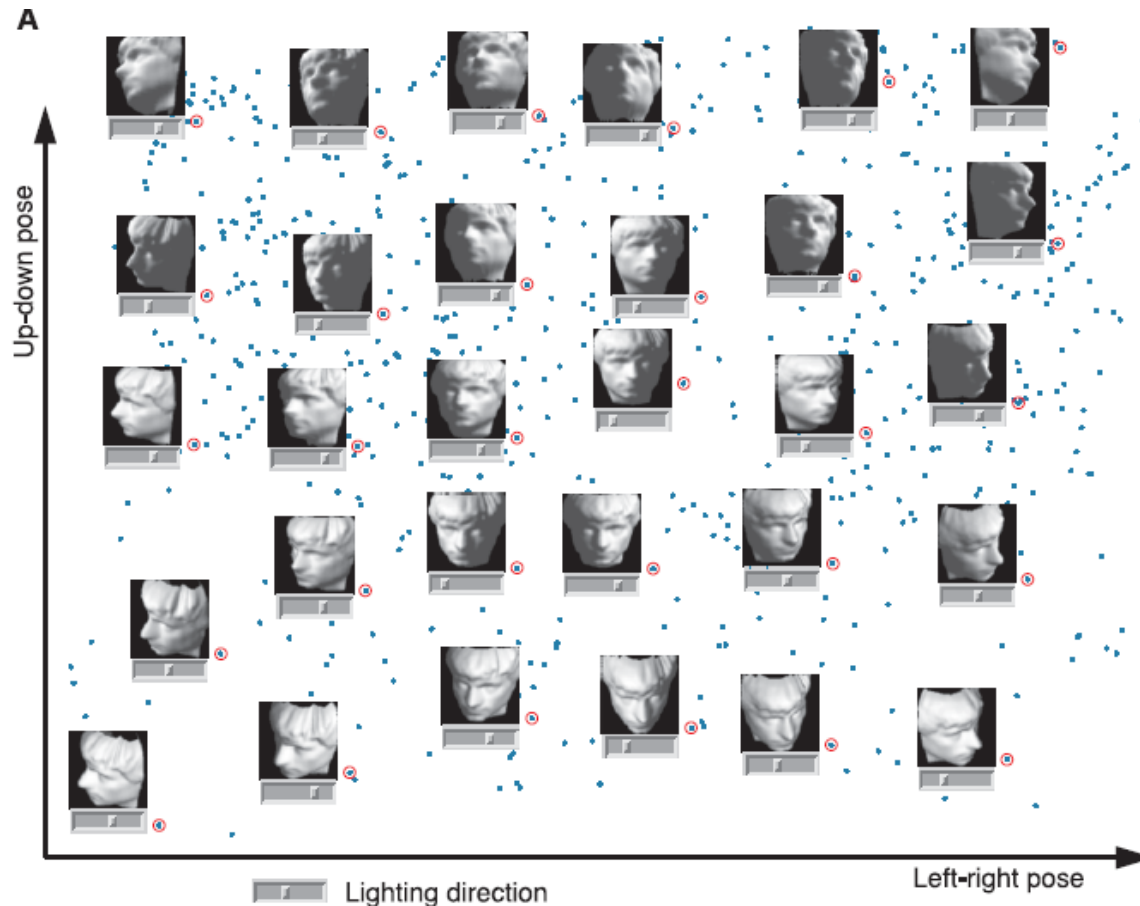


2维投影

Isomap的应用

● 人脸图像在2D流形空间的投影

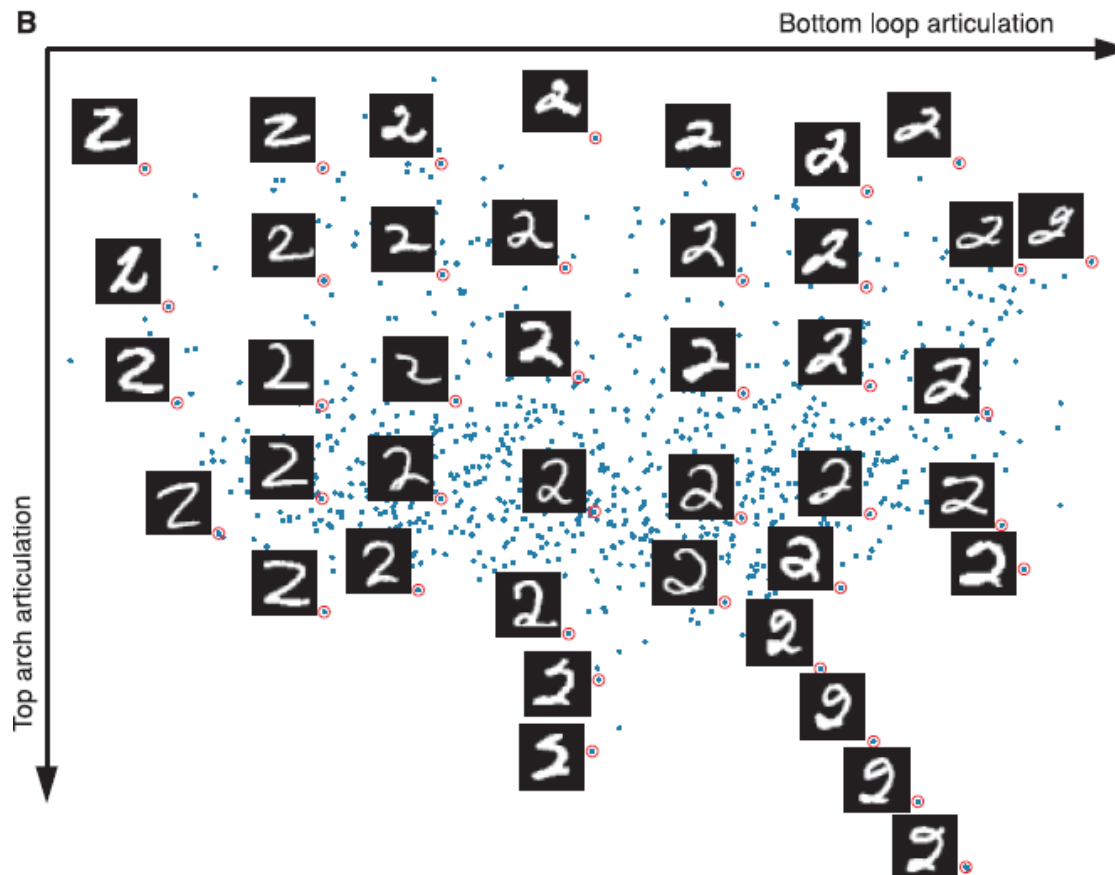
横坐标反映了光照变化，纵坐标反映姿态变化



Isomap的应用

- 手写数字（2）在2D流形空间的投影

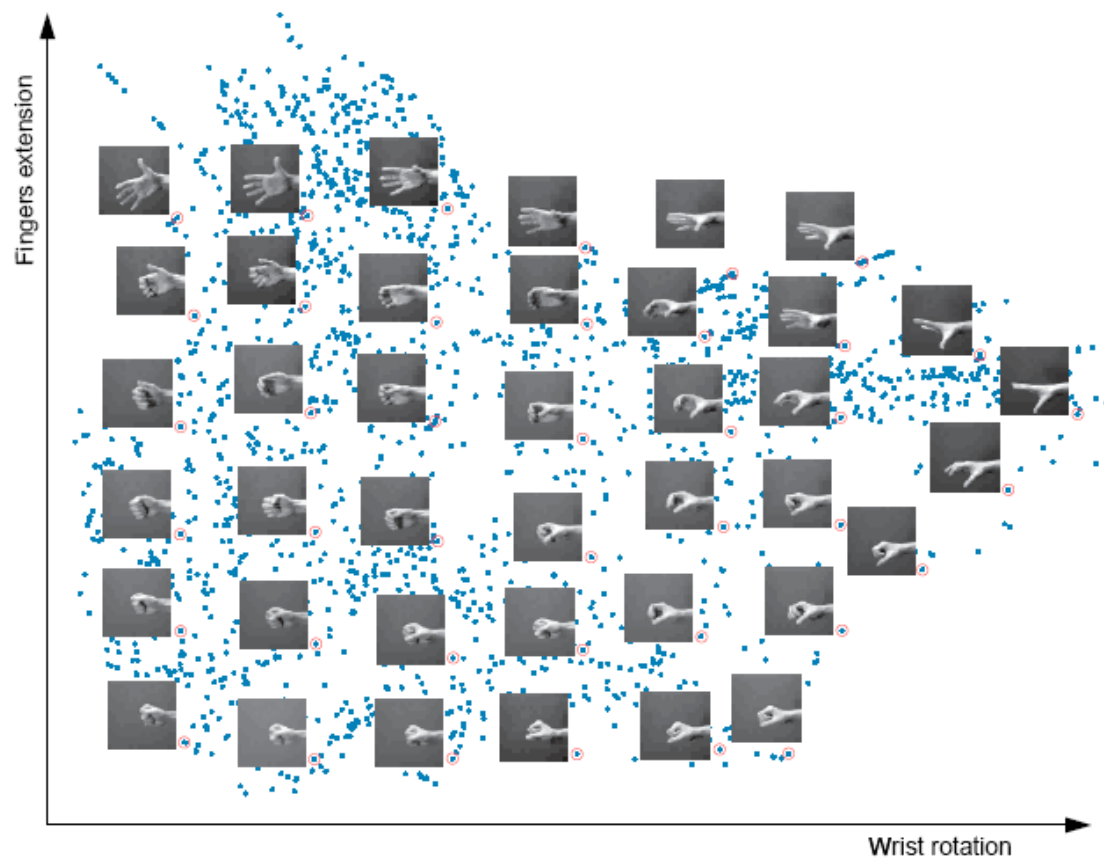
横坐标反映底部环型变化，纵坐标反映顶上穹型变化



Isomap的应用

● 手势在2D流形空间的投影

横坐标反映手腕旋转变化的，纵坐标反映手指的伸展变化



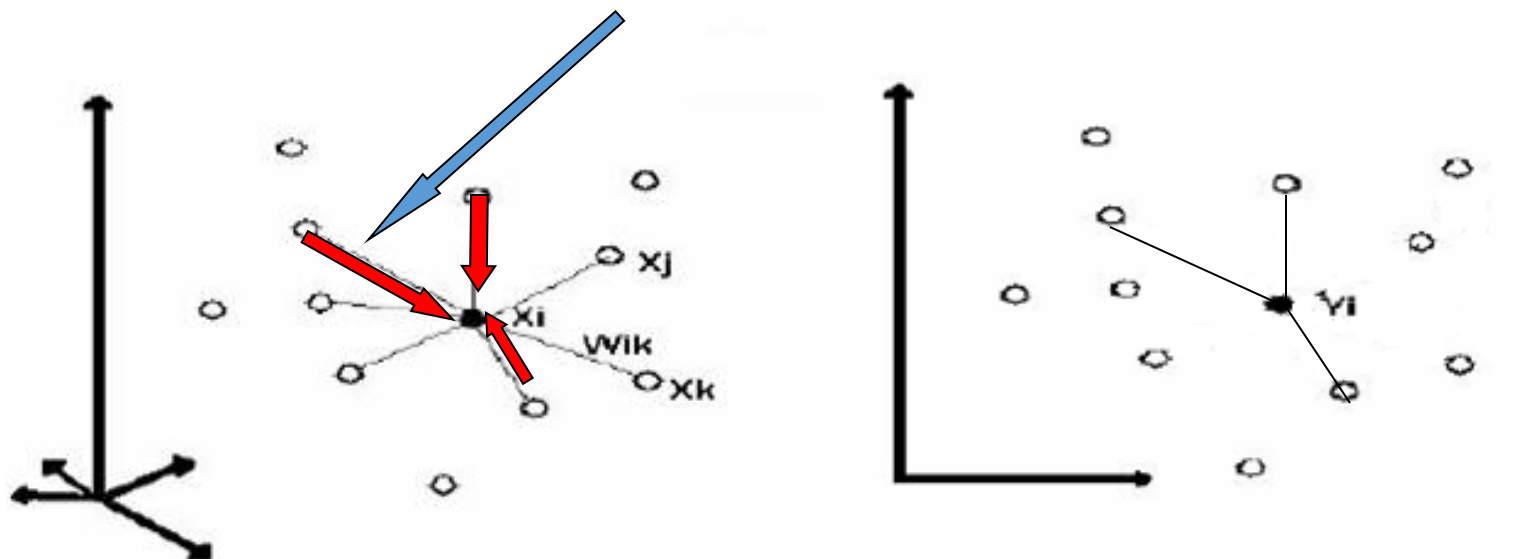
Isomap算法的特点

- Isomap是非线性的，适用于学习内部平坦的低维流形，不适于学习有较大内在曲率的流形。
- Isomap算法中有两个待定参数 K ， d 。

Laplacian Eigenmap

主要思想：

在高维空间中离得很近的点投影到低维空间中的像也应该离得很近。



M. Belkin and P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, NIPS 2001

Laplacian Eigenmap

- 主要思想的数学表达:

令样本集: $X = (x_1, x_2, \dots, x_n)$ 投影后样本: $Y = (y_1, y_2, \dots, y_n)$

LE的目标是最小化目标函数:

$$\sum_{i,j=1}^n \|y_i - y_j\|^2 W_{ij}$$

Laplacian Eigenmap

- 这里权值反映样本之间的关系，一般用热核表示：

$$W_{ij} = e^{-\frac{\|x_i - x_j\|^2}{t}}$$

也可以简单定义成1（节点i和j相邻）或0（不相邻）

- 为了使最小化问题解唯一，必须加上尺度归一的限制条件，目标函数变为：

$$\arg \min_{\mathbf{y}^T D \mathbf{y} = 1} \mathbf{y}^T L \mathbf{y}$$

这里 $L=D-W$ 被称为Laplacian矩阵

$D_{ii} = \sum_j W_{ji}$ 是对角矩阵

- 可以转化成广义特征值问题求解：

$$L \mathbf{y} = \lambda D \mathbf{y}$$

Laplacian Eigenmap算法的特点

- 算法是局部的非线性方法。
- 算法与谱图理论有很紧密的联系。
- 算法中有两个参数 k 、 d 。
- 算法通过求解稀疏矩阵的特征值问题解析地求出整体最优解。
- 算法使原空间中离得很近的点在低维空间也离得很近，可以用于聚类。
- 没有给出显式的投影映射，即，对于新样本（out-of-sample）无法直接得到其在低维子流形上的投影。

LLE, Isomap, Laplacian Eigenmap 有效的原因

- 1 它们都是非参数的方法，不需要对流形的很多的参数假设。
- 2 它们是非线性的方法，都基于流形的内在几何结构，更能体现现实中数据的本质。
- 3 它们的求解简单，都转化为求解特征值问题，而不需要用迭代算法。

流形学习研究的常规模式：

- 1 对嵌入映射或者低维流形作出某种特定的假设，或者以保持高维数据的某种性质不变为目标。
- 2 将问题转化为求解优化问题。
- 3 提供有效的算法。

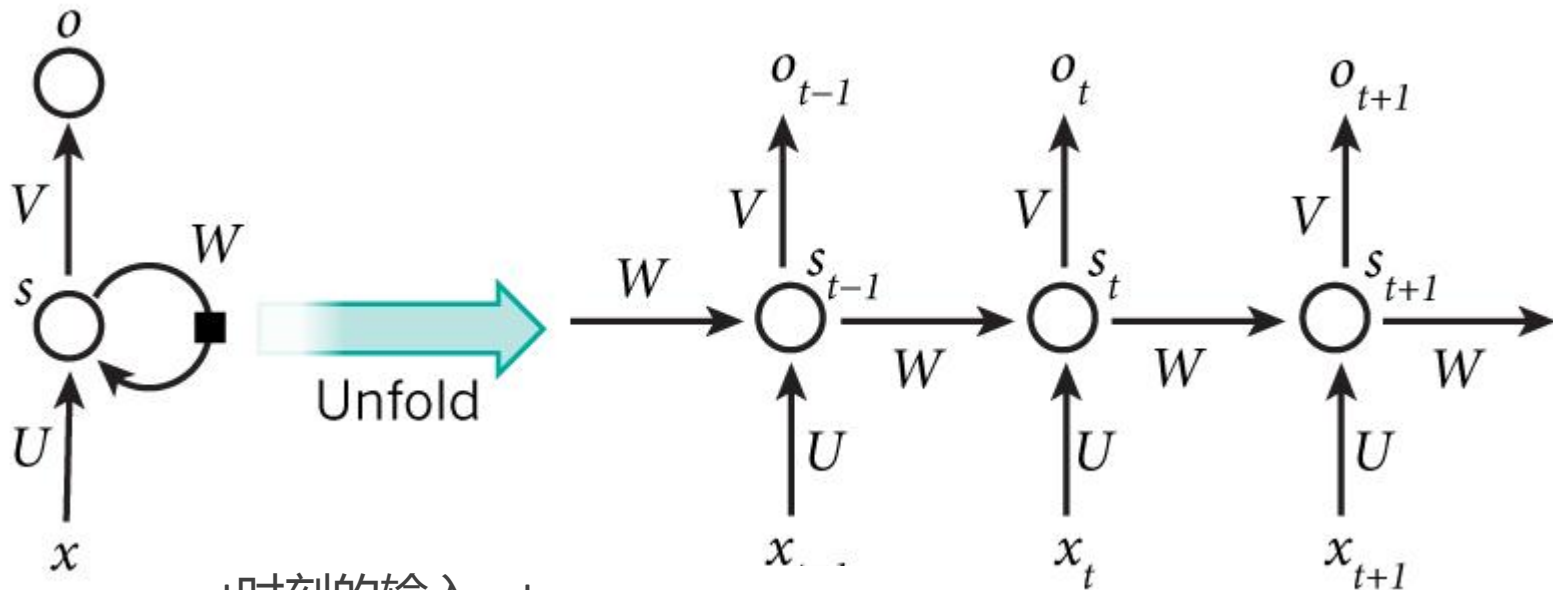
流形学习中存在的问题：

- 1 如何确定低维目标空间的维数？
- 2 当采样数据很稀疏时，怎样进行有效的学习？
- 3 将统计学习理论引入流形学习对其泛化性能进行研究。

RNN

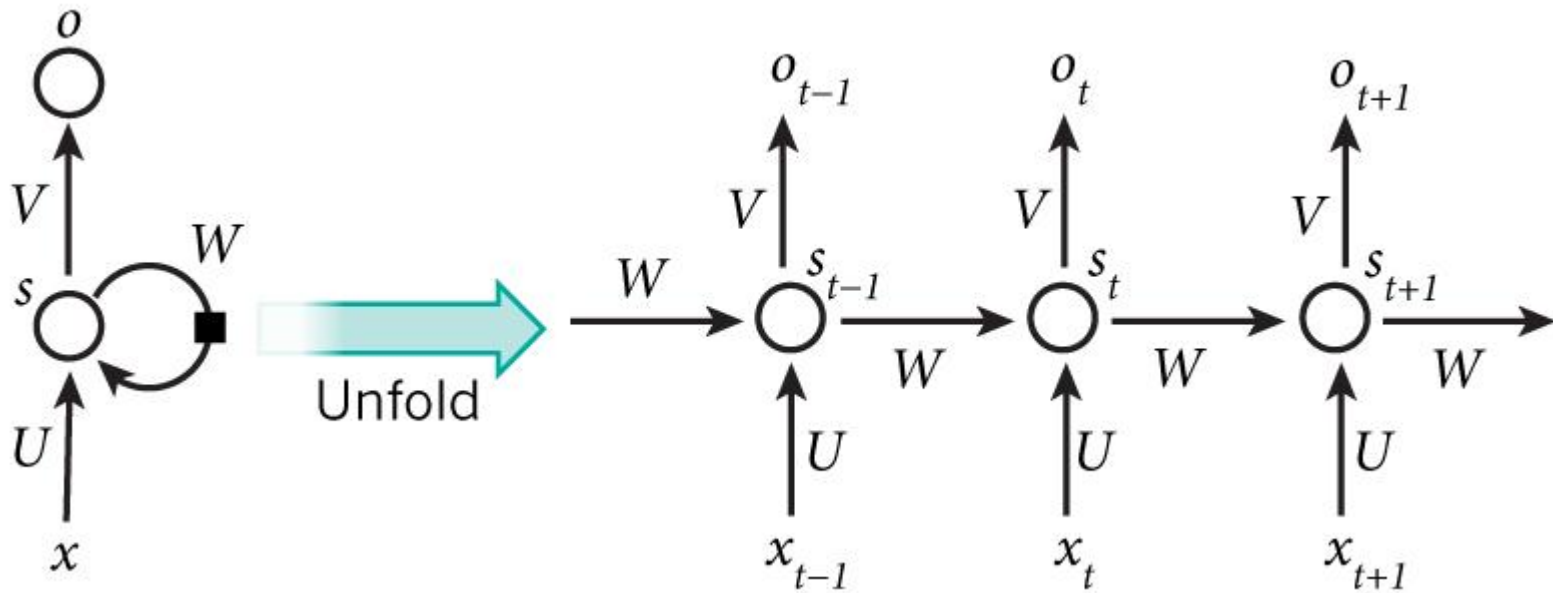
RNN (Recurrent Neural Networks)

- RNN is a class of artificial neural network where connections between units form a directed cycle. (developed in the 1980s)



- t 时刻的输入 x_t
- s_t 代表时刻 t 的隐藏状态
- o_t 代表时刻 t 的输出
- 输入层到隐藏层直接的权重由 U 表示
- 隐藏层到输出层的权重 V
- 隐藏层到隐藏层的权重 W

RNN (Recurrent Neural Networks)

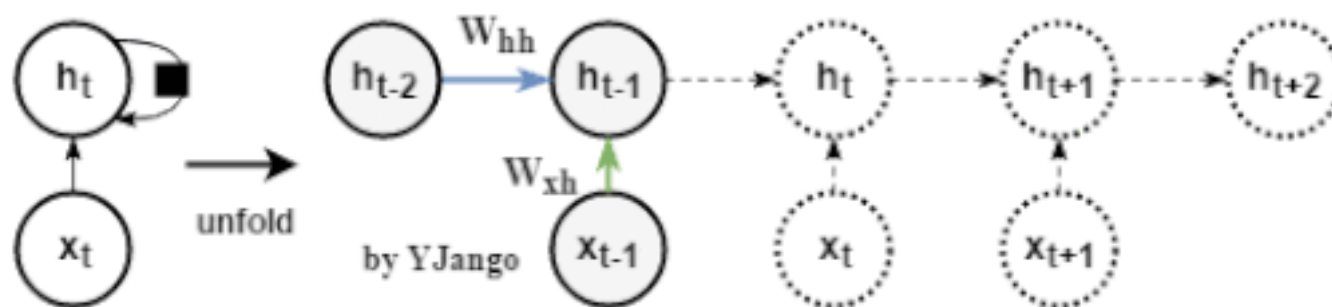


$$\begin{aligned} s_1 &= Ux_1 + Wh_0 \\ h_1 &= f(s_1) \\ o_1 &= g(Vh_1) \end{aligned}$$

$$\begin{aligned} s_2 &= Ux_2 + Wh_1 \\ h_2 &= f(s_2) \\ o_2 &= g(Vh_2) \end{aligned}$$

$$\begin{aligned} s_t &= Ux_t + Wh_{t-1} \\ h_t &= f(Ux_t + Wh_{t-1}) \\ o_t &= g(Vh_t) \end{aligned}$$

Forward



$$s_1 = Ux_1 + Wh_0$$

$$h_1 = f(s_1)$$

$$o_1 = g(Vh_1)$$

$$s_2 = Ux_2 + Wh_1$$

$$h_2 = f(s_2)$$

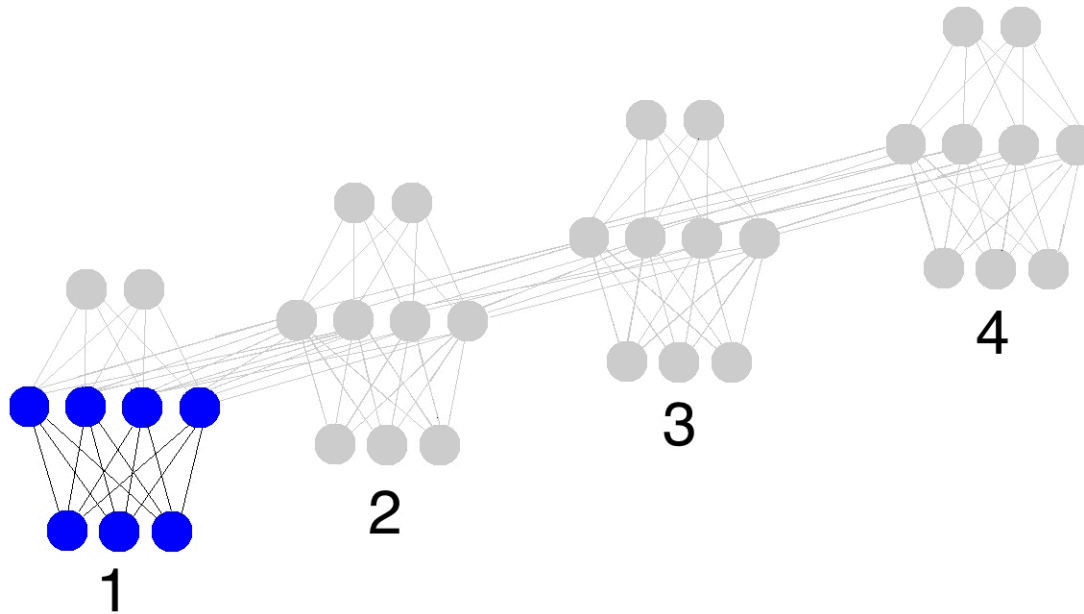
$$o_2 = g(Vh_2)$$

$$s_t = Ux_t + Wh_{t-1}$$

$$h_t = f(Ux_t + Wh_{t-1})$$

$$o_t = g(Vh_t)$$

Forward



MakeAGIF.com

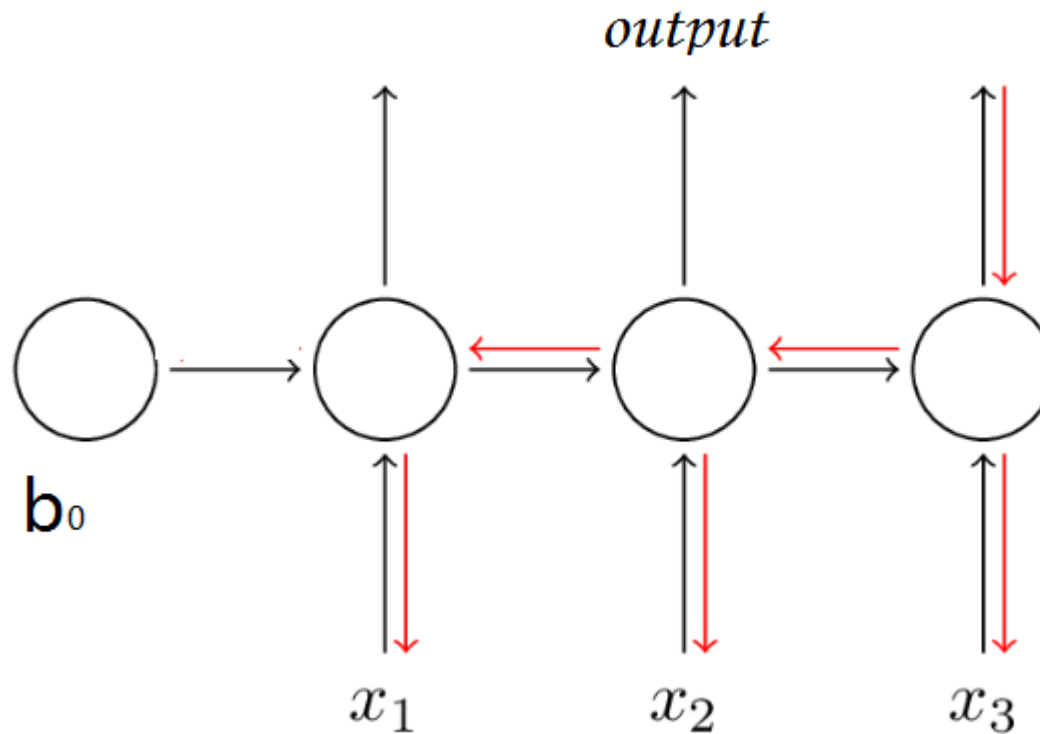
$$\begin{aligned} s_1 &= Ux_1 + Wh_0 \\ h_1 &= f(s_1) \\ o_1 &= g(Vh_1) \end{aligned}$$

$$\begin{aligned} s_2 &= Ux_2 + Wh_1 \\ h_2 &= f(s_2) \\ o_2 &= g(Vh_2) \end{aligned}$$

$$\begin{aligned} s_t &= Ux_t + Wh_{t-1} \\ h_t &= f(Ux_t + Wh_{t-1}) \\ o_t &= g(Vh_t) \end{aligned}$$

Training

- Back Propagation Through Time, BPTT

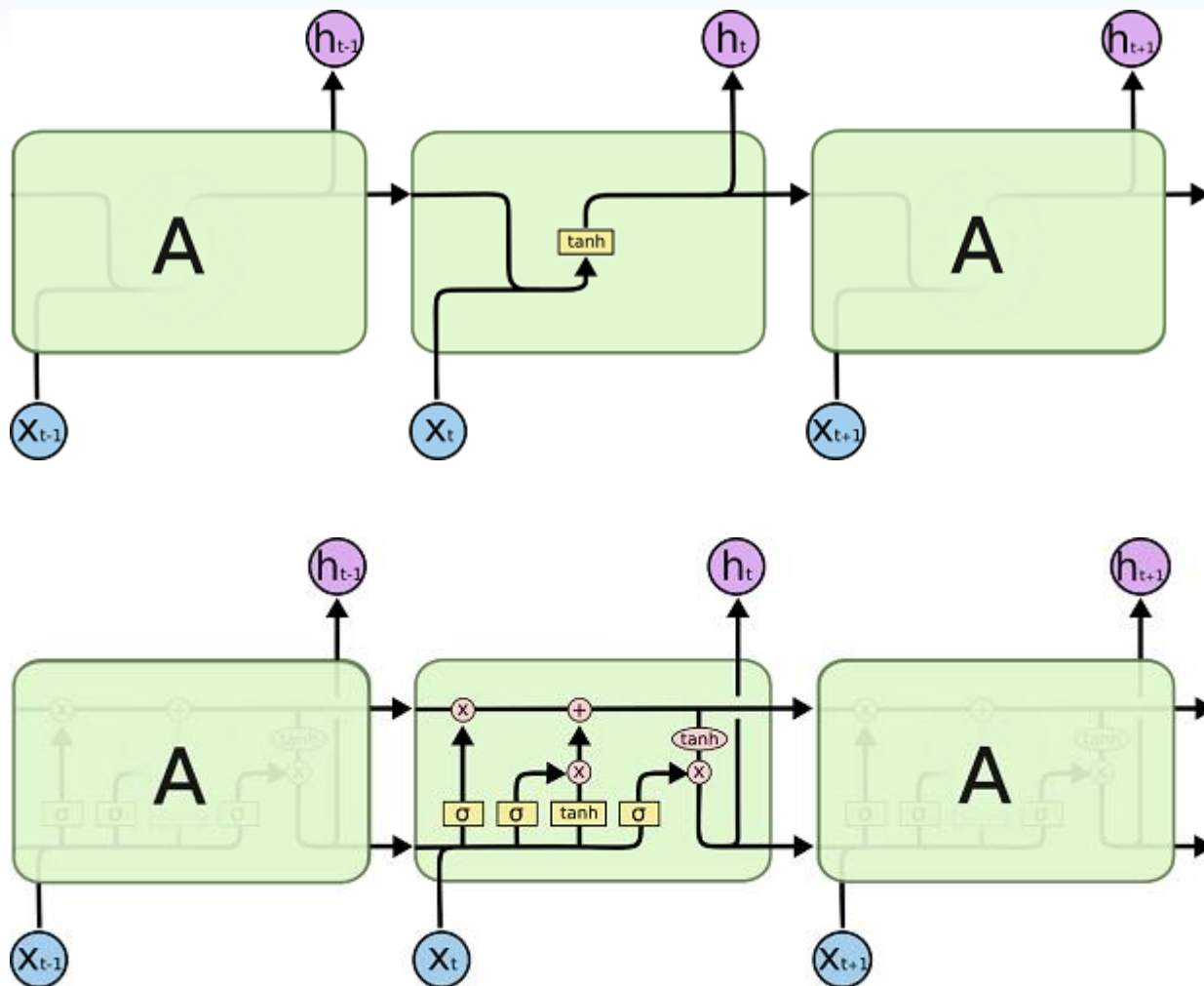


Problem: gradient explode and gradient vanish

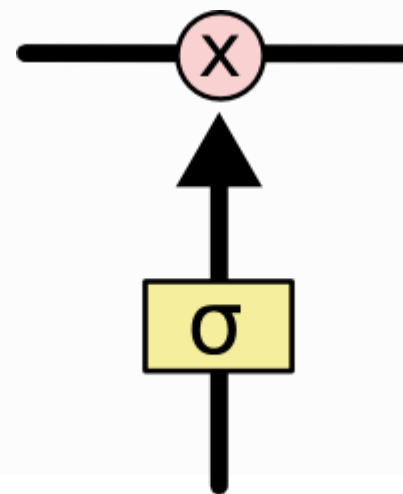
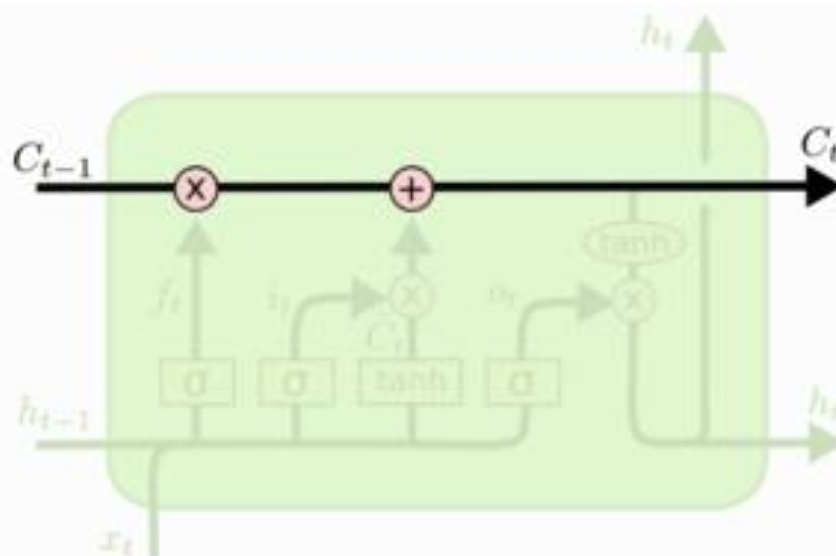
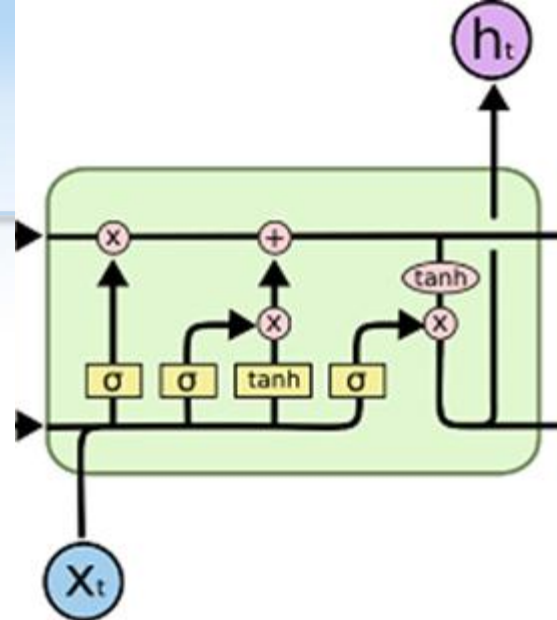
Gradient explode and gradient vanish

- Gradient explode:
 - gradient clipping
- Gradient vanish:
 - LSTM (Long Short-Term Memory, Hochreiter and Schmidhuber, 1997)

LSTM -- RNNs



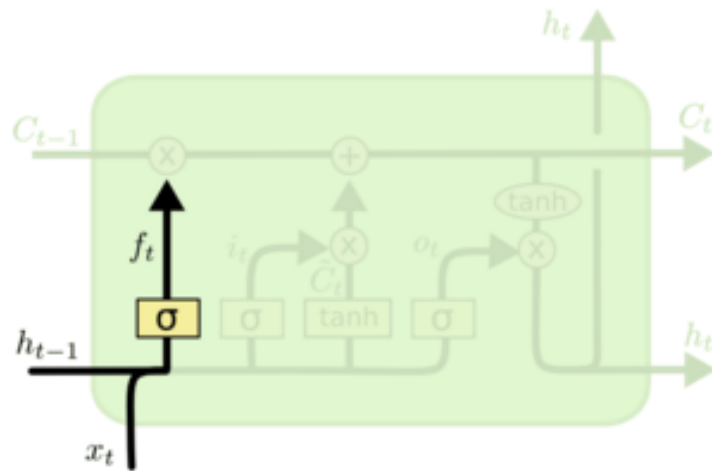
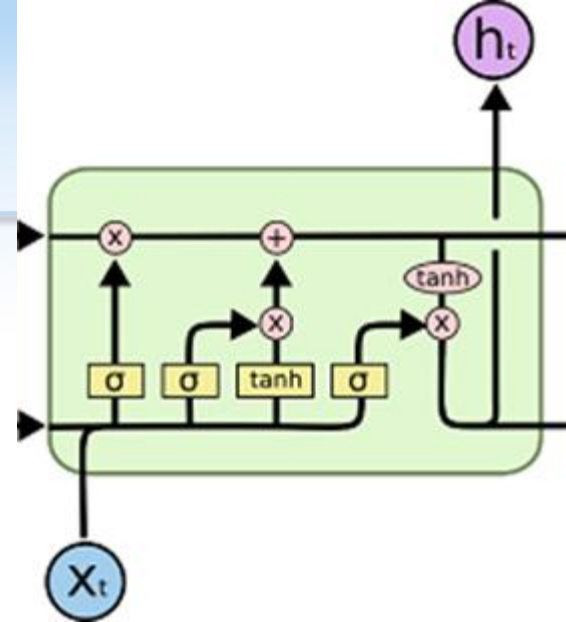
Cell state and gate



Cell state

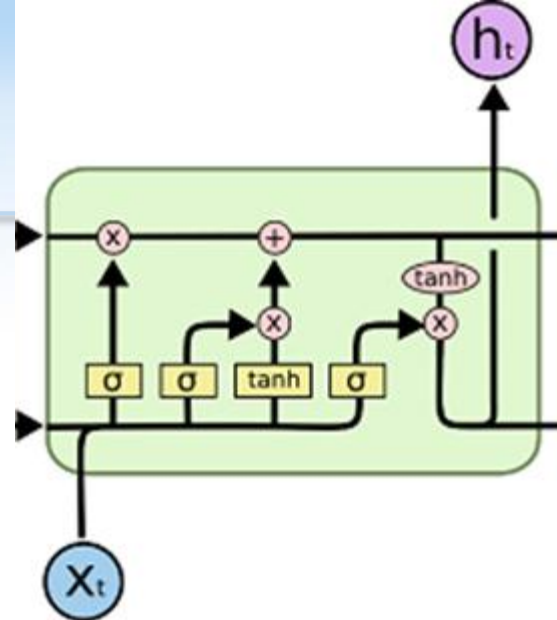
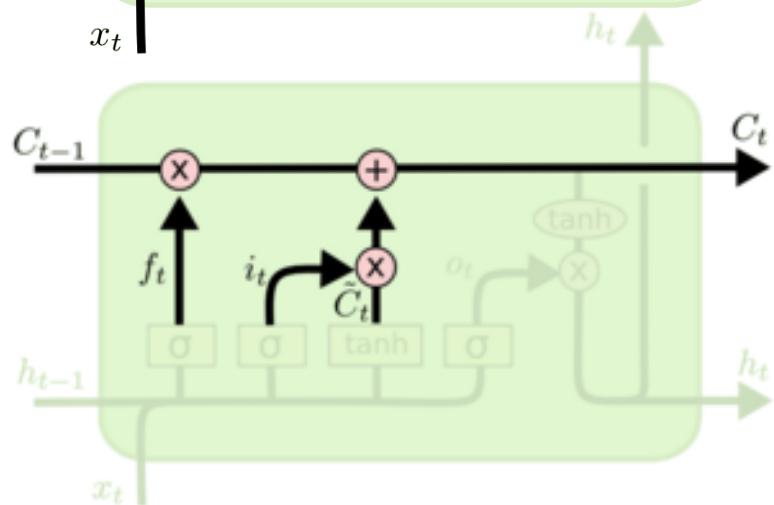
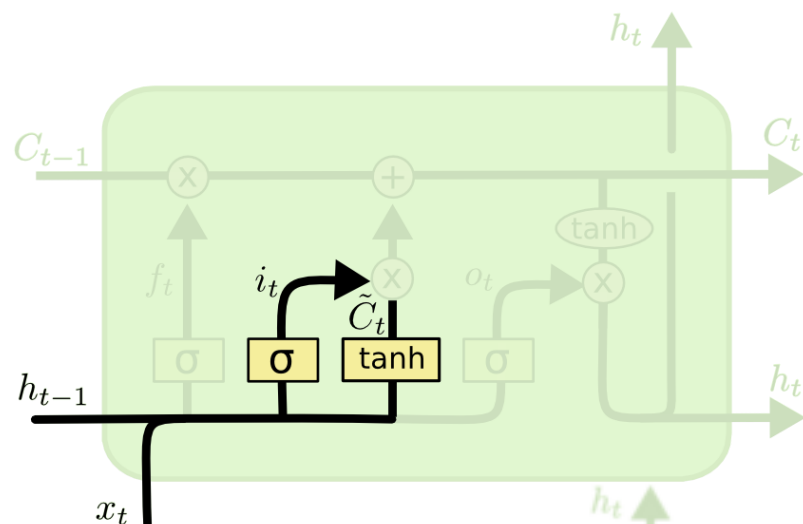
Gate: sigmoid

forget gate



$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

input gate



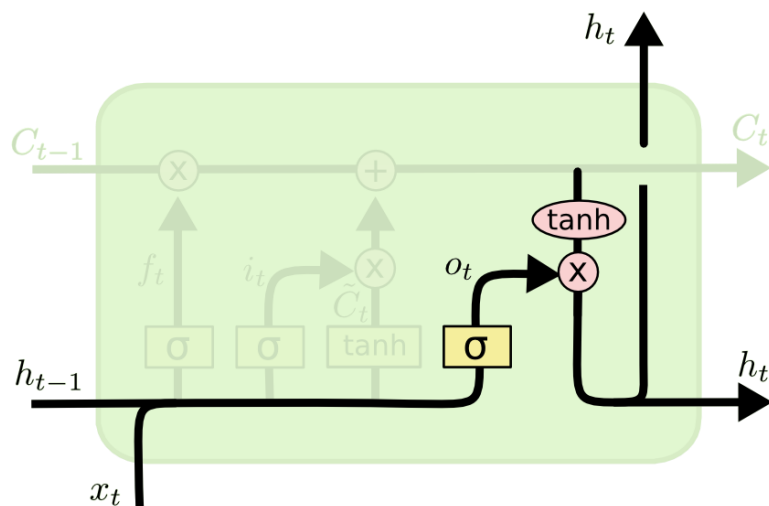
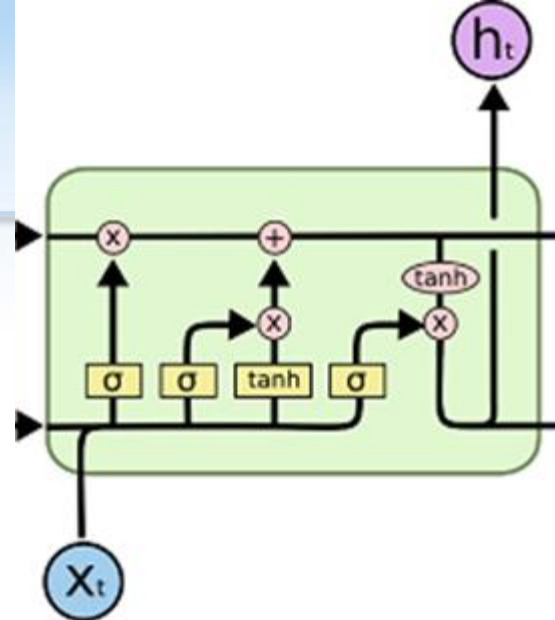
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

<http://blog.csdn.net/menc15>

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

output gate



$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

小节

- 计算机视觉中的机器学习方法
 - 子空间分析
 - 流形学习
 - RNN

课后练习

1试编程实现基于PCA的人脸识别。

2试编程实现LLE算法。

谢谢！