

# 第八章 特征提取

模式识别基础，2019春

# 特征变换/提取

- 特征选择：从 $p$ 个特征中选出 $k$ 个
- 特征提取：把 $p$ 个特征变为 $k$ 个新特征

更好分类  
和/或  
减少计算量

变换  $\mathbf{x}' = W(\mathbf{x})$

线性变换  $\mathbf{x}' = \mathbf{W}^T \mathbf{x}$

$\mathbf{W}^T$ :  $k \times p$  维

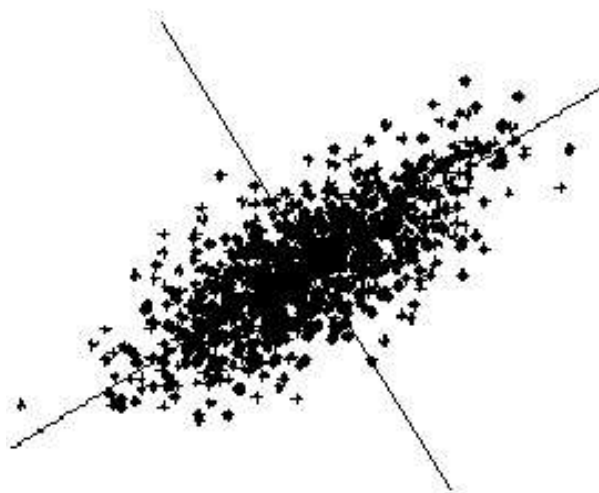
通常,  $p > k$ , “特征压缩”, “特征变换”

# 主成分分析

## (Principal Component Analysis, PCA)



Karl Pearson (1901)



目标：通过线性变换，用一组正交向量来表示原特征。

新的特征向量是原特征向量的线性组合。

# PCA

目标：通过线性变换，用一组正交向量来表示原特征。  
新的特征向量是原特征向量的线性组合。

记原特征向量为  $x_1, \dots, x_p$

$$\xi_i = \sum_{j=1}^p a_{ij} x_j \quad \text{表示原始特征向量的线性组合}$$

写成矩阵：  $\xi = \mathbf{A}^T \mathbf{x}$

目标是寻找正交的转化 $\mathbf{A}$ 产生新的正交向量  $\xi_i$ （线性正交变换）

注意：PCA分析对原始特征的缩放（scaling）是敏感的

# PCA的求解

先考虑第一个主成分  $\xi_1$   $\xi_1 = \sum_{j=1}^p a_{1j} x_j$

选择向量  $\alpha_1 = [\alpha_{11}, \dots, \alpha_{1p}]^T$

使得投影到  $\xi_1$  后的方差最大化，同时满足约束  $\alpha_1^T \alpha_1 = 1$

投影到  $\xi_1$  后的方差

$$\text{var}(\xi_1) = E(\xi_1^2) - E(\xi_1)^2 = E(\alpha_1^T \mathbf{x} \mathbf{x}^T \alpha_1) - E(\alpha_1^T \mathbf{x}) E(\mathbf{x}^T \alpha_1) = \alpha_1^T \Sigma \alpha_1$$

其中  $\Sigma$  表示数据的协方差矩阵，  $\mathbf{X} = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}]_{p \times n}$

带约束的求极值问题，用拉格朗日乘子法求解：

$$f(\alpha_1) = \alpha_1^T \Sigma \alpha_1 - \nu(\alpha_1^T \alpha_1 - 1)$$

对  $\alpha_1$  求偏导，得到：  $\Sigma \alpha_1 - \nu \alpha_1 = 0$

因此  $\alpha_1$  是协方差矩阵的特征向量，  $\nu$  是对应的特征值。

因此  $Var(x_1) = a_1^T S a_1 = n a_1^T a_1 = n$

对协方差矩阵的  $p$  个特征值排序，  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p$

我们希望找使得  $Var(x_1)$  **最大** 的变换（线性组合），因此选择  $\nu$  为最大的特征值  $\lambda_1$ ，该特征值对应的特征向量即为所求的线性组合系数。

得到的  $\xi_1$  被称为第一主成分（**the first principal component**）。

- 求解第二主成分的投影方向  $a_2$ ，使得方差  $\lambda_2$  最大化，满足约束条件：

$$a_2^T a_2 = 1$$

$$a_2^T a_1 = 0 \quad (\text{互相正交})$$

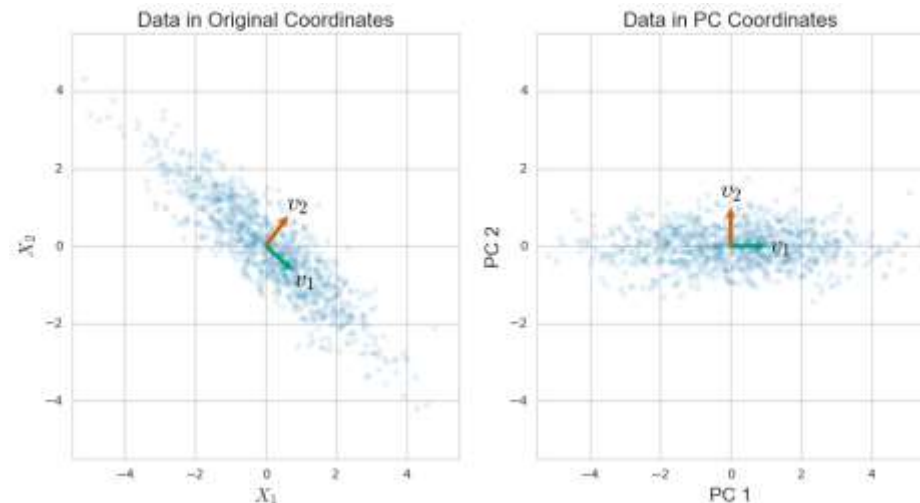
可以求出  $a_2$  是  $\Sigma$  第二大特征值对应的特征向量。

推广：第  $k$  个主成分投影方向就是第  $k$  大的特征值  $\lambda_k$  对应的特征向量

- 投影后的样本方差

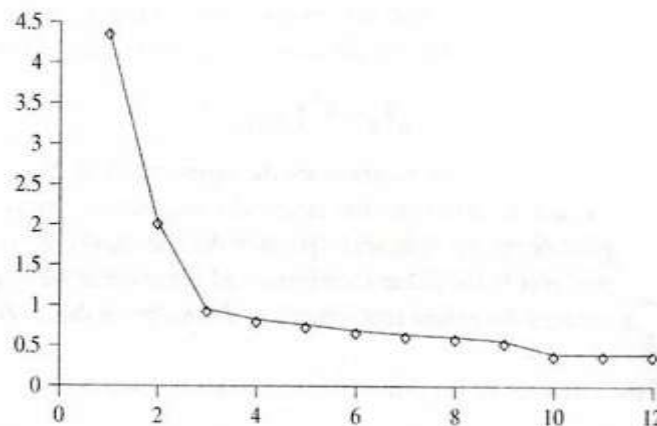
$$\sum_{i=1}^p \text{var}(\xi_i) = \sum_{i=1}^p \lambda_i$$

等于原始数据的样本方差



如果用前k个主成分对样本进行表示，则投影后保留的样本方差占原始数据总方差：

$$\sum_{i=1}^k \lambda_i / \sum_{i=1}^p \lambda_i$$





# PCA的算法流程：

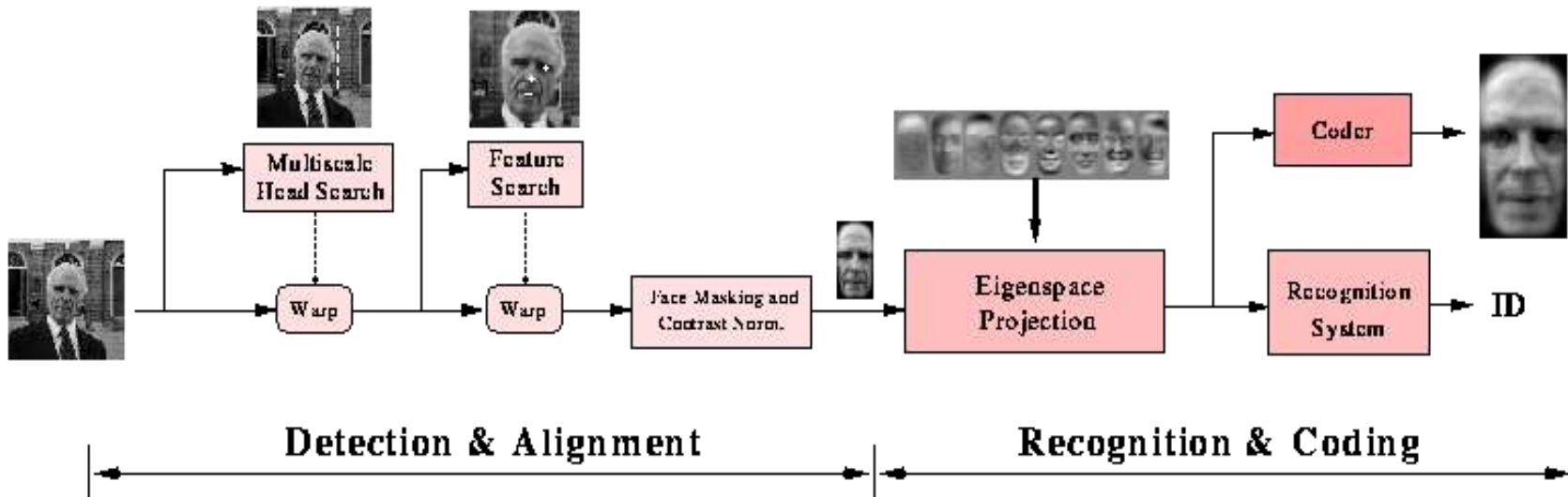
输入数据  $\mathbf{X} = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}]_{p \times n}$ ,  $n$  个  $p$  维的数据样本点

1. 计算样本均值  $\mu = \sum_{i=1}^n \mathbf{x}^{(i)}$
2. 对数据进行中心化  $\mathbf{x}^{(i)} = \mathbf{x}^{(i)} - \mu$
2. 计算协方差矩阵  $\Sigma = \frac{1}{n} \mathbf{X} \mathbf{X}^T$  ( $p \times p$  维)
3. 对协方差矩阵进行特征值分解
4. 取最大的  $k$  个特征值所对应的特征向量，构成投影矩阵  $\mathbf{W} = [\alpha_1, \alpha_2, \dots, \alpha_k]_{p \times k}$
5. 投影到  $\mathbf{X}' = \mathbf{W}^T \mathbf{X}$

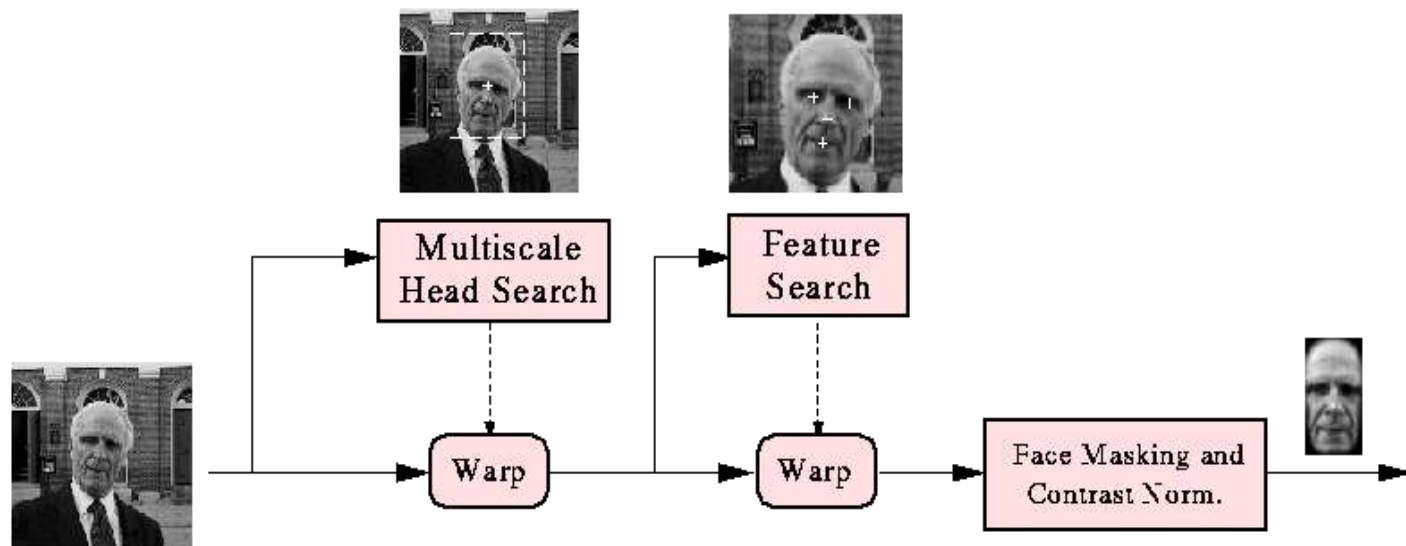
原始数据  $\mathbf{X}$  为 ( $p \times n$  维)，投影后数据  $\mathbf{X}'$  为  $k \times n$  维，其中  $k < p$

# 例子：PCA在人脸识别中的应用举例

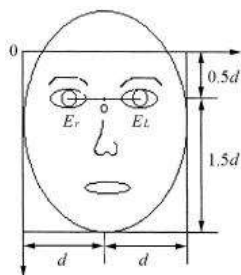
- M. Turk & A. Pentland, Eigenfaces for recognition, *Journal of Cognitive Neuroscience*, vol.3, no.1, pp.71-86, 1991



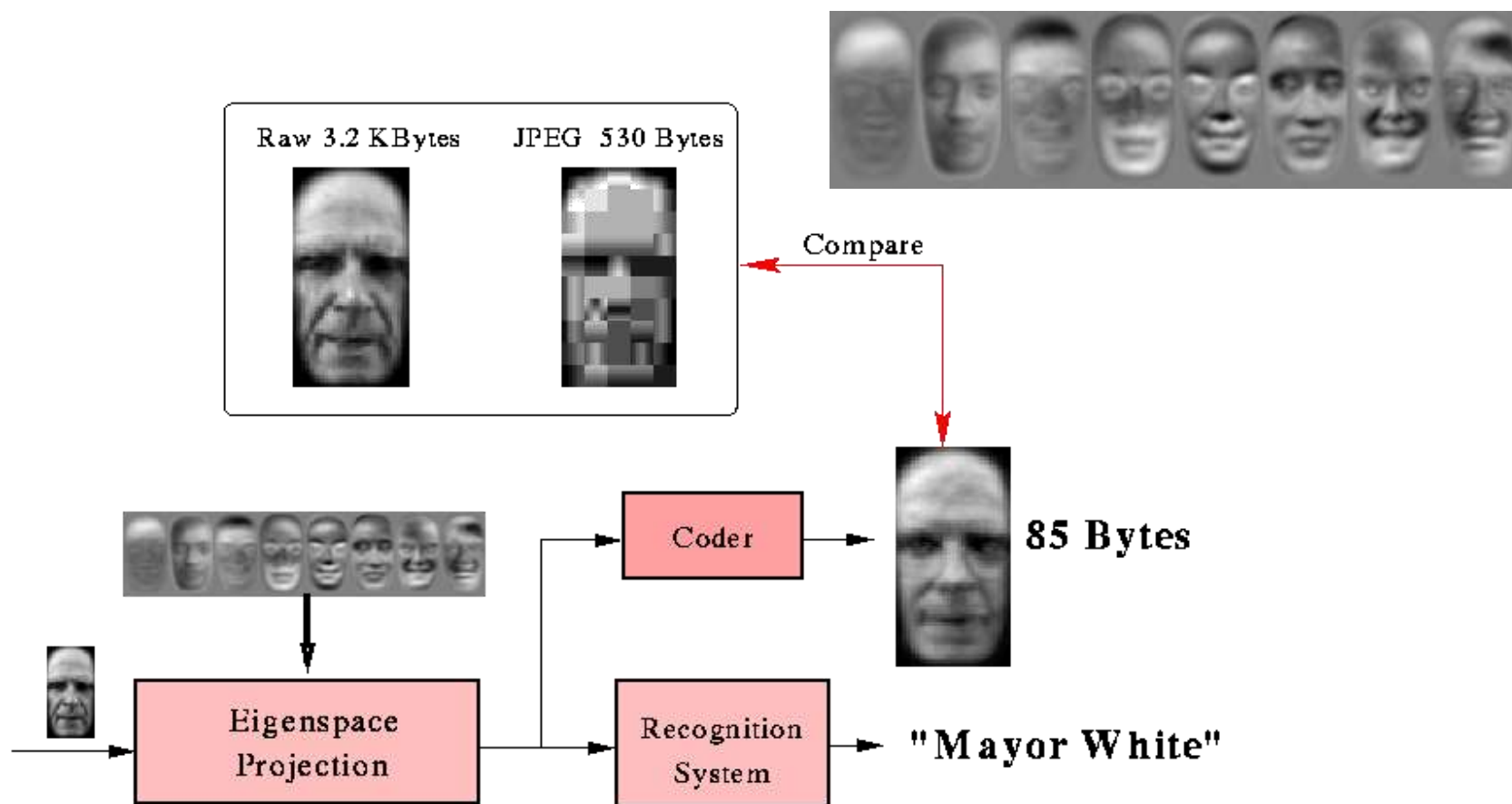
## 预处理



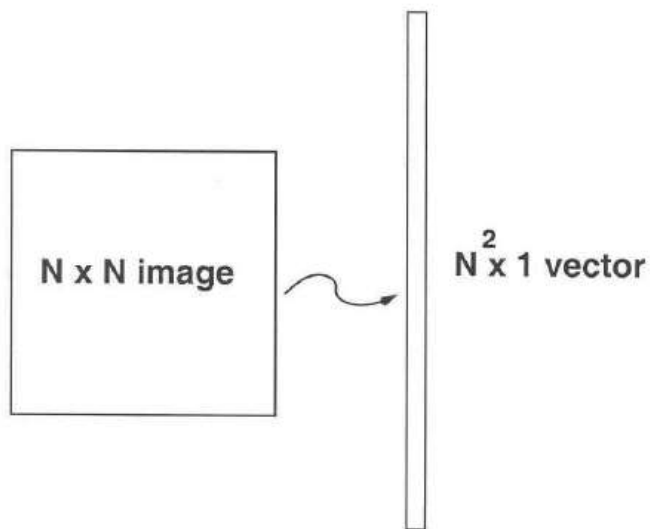
## 图像归一化和裁剪



- 本征脸提取、表示和基于本征脸的分类
- The first 8 normalized eigenfaces:



## 方法



样本集  $\mathbf{x}_i \in R^{N^2}, i = 1, \dots, M$

用PCA进行降维

$$\text{总体散布矩阵 } \Sigma = \frac{1}{M} \sum_{i=0}^{M-1} (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T = \frac{1}{M} \mathbf{X}\mathbf{X}^T$$

$N^2 \times N^2$  维矩阵，求其正交归一的本征向量，但计算困难。

解决办法：

考查  $M \times M$  ( $M$  为样本数,  $M \ll N^2 \times N^2$ ) 维矩阵  $\mathbf{R} = \mathbf{X}^T \mathbf{X}$

其特征方程是：  $\mathbf{X}^T \mathbf{X} \mathbf{v}_i = \lambda_i \mathbf{v}_i$

两边同乘以  $\mathbf{X}$ ：  
 $\mathbf{X} \mathbf{X}^T \mathbf{X} \mathbf{v}_i = \lambda_i \mathbf{X} \mathbf{v}_i$   
 $\Sigma \mathbf{X} \mathbf{v}_i = \lambda_i \mathbf{X} \mathbf{v}_i$

记  $\mathbf{u}_i = \mathbf{X} \mathbf{v}_i$ ，有  $\Sigma \mathbf{u}_i = \lambda_i \mathbf{u}_i$

所以，矩阵  $\mathbf{X}^T \mathbf{X}$  和  $\mathbf{X} \mathbf{X}^T$  具有相同的特征值，而特征向量具有关系

$$\mathbf{u}_i = \mathbf{X} \mathbf{v}_i$$

易求得， $\Sigma$  的归一化的本征向量是

$$\mathbf{u}_i = \frac{1}{\sqrt{\lambda_i}} \mathbf{X} \mathbf{v}_i, \quad i = 1, 2, \dots, M$$

注意，因为矩阵  $\Sigma$  的秩最多为  $M$ ，所以最多只有  $M$  个本征值和本征向量。

每一个本征向量仍然是一个  $N^2$  维向量，即  $N \times N$  维图像，仍然具有类似人脸的样子，因此被称作“本征脸”（eigenfaces）。按照本征值从大到小排列，

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_M$$

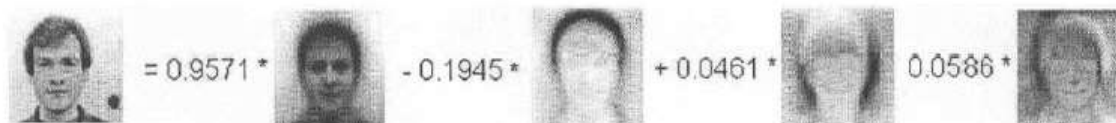
并从前向后取对应的本征脸，即构成对原图像的最佳的降维表示。



原图像可以表示成特征脸的线性组合（在特征脸空间中的点）。

$$\mathbf{y}_i = \mathbf{U}^T \mathbf{x}_i, i = 1, \dots, M$$

$$\hat{\mathbf{x}}_i = \hat{\mathbf{U}} \hat{\mathbf{y}}_i^T \quad \text{其中, } \hat{\mathbf{y}}_i \text{ 为 } d \text{ 维 } (d < p), \hat{\mathbf{U}} \text{ 为 } p * d \text{ 维}$$





比如选取前k个特征向量，使

$$\sum_{i=0}^{k-1} \lambda_i / \sum_{i=0}^{M-1} \lambda_i \geq \alpha$$

比如  $\alpha = 99\%$  即可以保持原样本99%的信息。

对原图像的表达  $\hat{\mathbf{x}}_i - \boldsymbol{\mu} = \sum_{j=1}^k y_{ij} \mathbf{u}_j$



The original face and the recovered face

# PCA的另一种计算方式：奇异值分解

( Singular value decomposition, SVD )

$\Sigma$ 是对角矩阵，对角元素是非负实数，被称为奇异值。U和V是其左奇异向量和右奇异向量

The diagram illustrates the SVD equation  $M = U \Sigma V^*$  and the orthogonality conditions  $U U^* = I_m$  and  $V V^* = I_n$ . Each matrix is represented by a grid of colored squares and its dimensions are specified below.

**SVD Equation:**

$$\begin{matrix} \text{Grid} & & \text{Grid} & & \text{Grid} & & \text{Grid} \\ \mathbf{M} & = & \mathbf{U} & & \mathbf{\Sigma} & & \mathbf{V}^* \\ m \times n & & m \times m & & m \times n & & n \times n \end{matrix}$$

**Orthogonality Conditions:**

$$\begin{matrix} \text{Grid} & & \text{Grid} & & \text{Grid} \\ \mathbf{U} & & \mathbf{U}^* & = & \mathbf{I}_m \\ & & & & m \times m \end{matrix}$$
$$\begin{matrix} \text{Grid} & & \text{Grid} & & \text{Grid} \\ \mathbf{V} & & \mathbf{V}^* & = & \mathbf{I}_n \\ & & & & n \times n \end{matrix}$$

[https://en.wikipedia.org/wiki/Singular\\_value\\_decomposition](https://en.wikipedia.org/wiki/Singular_value_decomposition)

# SVD与PCA

对样本矩阵 $\mathbf{X}^T$ 作矩阵的SVD分解：

$$\mathbf{X}=\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

$$\mathbf{X}^T\mathbf{X}=(\mathbf{U}\mathbf{\Sigma}\mathbf{V})^T(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T)=(\mathbf{V}\mathbf{\Sigma}^T\mathbf{U}^T)(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T)$$

$$\mathbf{U}^T\mathbf{U}=\mathbf{I}_m$$

$$\mathbf{X}^T\mathbf{X}=\mathbf{V}\mathbf{\Sigma}^T\mathbf{\Sigma}\mathbf{V}^T \quad \text{特征值分解!}$$

取 $\mathbf{V}^T$ 中对应于 $k$ 个最大特征值的 $k$ 行： $\mathbf{v}_1, \dots, \mathbf{v}_k$ 作为投影向量

# Karhunen-Loève变换（K-L变换）

- K-L变换的基本原理
- 函数的级数展开：将函数用一组（正交）基函数展开，用展开系数表示原函数。
- K—L展开：把随机向量用一组正交基向量展开，用展开系数代表原向量。
- 基向量所张成的空间：新的特征空间。
- 展开系数组成的向量：新特征空间中的样本向量。

# K-L展开

- 对随机向量 $x$ ，用确定的完备正交归一向量系  $u_j \ j = 1, 2, \dots, \infty$  展开，得

$$x = \sum_{j=1}^{\infty} c_j u_j$$

$$c_j = u_j^T x \quad (\text{两边同乘以 } u_j^T \text{ 即得})$$

$$\text{其中, } u_i^T u_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

如果只用有限项来逼近 $x$ ，即

$$\hat{x} = \sum_{j=1}^d c_j u_j \quad x \text{ 为 } D \text{ 维, } d < D$$

仅用 $d$ 项展开后的均方误差

$$\begin{aligned} e &= E[(x - \hat{x})^T (x - \hat{x})] = E\left[\left(\sum_{j=d+1}^{\infty} c_j u_j\right)^T \left(\sum_{j=d+1}^{\infty} c_j u_j\right)\right] \\ &= E\left[\sum_{j=d+1}^{\infty} c_j^2\right] = E\left[\sum_{j=d+1}^{\infty} u_j^T x x^T u_j\right] = \sum_{j=d+1}^{\infty} u_j^T E[xx^T] u_j \end{aligned}$$

记  $\psi = E[xx^T]$  则  $e = \sum_{j=d+1}^{\infty} u_j^T \psi u_j$

最小化均方误差, 即  $\min(e), \text{ s.t. } u_j^T u_j = 1$

用拉格朗日函数求解

$$g(u) = \sum_{j=d+1}^{\infty} u_j^T \psi u_j - \sum_{j=d+1}^{\infty} \lambda_j [u_j^T u_j - 1]$$

$$\frac{\partial}{\partial u_j} g(u) = 0 \quad j = d+1, \dots, \infty$$

得  $(\psi - \lambda_j I) u_j = 0$

既  $\psi u_j = \lambda_j u_j \quad j = d+1, \dots, \infty$

令  $d = 0$ , 则得  $\psi u_j = \lambda_j u_j \quad j = 1, 2, \dots, \infty$

均方误差：
$$\xi = \sum_{j=d+1}^{\infty} \lambda_j$$

即：
$$\psi = E[xx^T]$$

用矩阵的前 $d$ 个本征值（从大到小排列）对应的本征向量作为基来展开 $x$ 时，截断误差在所有用 $d$ 维正交坐标系展开中是最小的。

$u_j \quad j = 1, 2, \dots, d$  张成了新的特征空间

展开系数  $C_j = u_j^T x$  则组成了新的特征向量。



# K-L展开式的性质

## 1. 数据的最佳（压缩）表达 —— 均方误差最小

与每一维特征  $u_j$  对应的本征值  $\lambda_j$ ，反映了该维特征对表达原空间  $x$  的有效性（贡献）大小。

从大到小排序： $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_D$

取：从前面开始

丢：从后面开始

# K-L展开式的性质

2. 新空间中的特征是互不相关的

$$E[c_i c_j] = E[u_i^T x x^T u_j] = \lambda_i u_i^T u_j = \lambda_i \delta_{ij}$$

即变换后的特征向量  $C = [c_1, c_2, \dots, c_D]^T$  的二阶矩矩阵为

$$E[cc^T] = U^T \psi U = \Lambda = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_d \end{bmatrix}$$

其中  $U = [u_1, u_2, \dots, u_D]$  为变换阵,  $\psi = E[xx^T]$

## 考查样本集协方差阵的特征值

- 本征值大的特征向量代表的是样本集中变化大的方向，即方差大的方向，最能反映样本之间的差异
- 而本征值小的特征向量则对应样本分布集中的方向，这些方向上方差小，均值可较好地代表样本

故把  $\Sigma$  的本征值按从小到大排列：

$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_d \leq \cdots \leq \lambda_D$$

取前 $d$ 个最小本征值对应的 $d$ 个特征向量组成变换阵，可以最好地用均值代表样本集。

# 非监督的K-L特征提取

非监督情况下，没有已知类别的训练样本，只能根据知识和/或假定来进行特征选择和提取。通常用方差作为衡量指标，认为选择或提取总体未知样本方差越大，越有利于将它分开。

用总体协方差矩阵作为K-L产生矩阵：

$$\hat{\mathbf{a}} = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T \quad \mathbf{m} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i$$

特征值从大到小排序，选前 $d$ 个对应的特征向量组成特征提取器。（在均方误差最小意义上用 $d < D$ 维对 $D$ 维样本空间的最佳表示）

----- 等价于PCA

# 用于监督模式识别的K-L变换

- 基本考虑：

- (1) 消除特征各分量之间的相关性（用KL变换）

- (2) 考查变换后各特征的类均值及方差，

选择方差小，类均值与总体均值差别大的特征

三步：1. 计算K-L变换的产生矩阵

$$S_w = \sum_{i=1}^c P_i \Sigma_i$$

（总类内离散度矩阵）

- 用  $S_w$  作KL变换, 得本征值为  $\lambda_i$  的特征向量  $u_i$

新特征  $y_i = u_i^T x \quad i = 1, \dots, D$  方差为  $\lambda_i$

计算  $J(y_i) = \frac{u_j^T S_b u_j}{\lambda_i}$

作为评价新特征的各个分量的分类性能的指标,

其中  $S_b = \sum_{i=1}^c P(\omega_i)(\mu_i - \mu)(\mu_i - \mu)^T$  为类间离散度矩阵。

用  $J(y_i)$  排序

$$J(y_1) \geq J(y_2) \geq \dots \geq J(y_d) \geq \dots \geq J(y_D)$$

选择前d个分量组成新的特征向量, 相应的 $u_j$ 组成变换阵

$$U = [u_1 \ u_2 \ \dots \ u_d]$$

# 非监督K-L变换 (PCA)

假设有样本数量 $N=4$ 的样本集合 $\mathbf{X} = \begin{bmatrix} 3 & 5 & -3 & -5 \\ 1 & 3 & -1 & -3 \end{bmatrix}$ ，目标是将其降到一维。

Step1. 中心化：由于均值向量 $\boldsymbol{\mu} = \mathbf{0}$ ，因此对原始特征进行中心化处理后样本点特征大小不变。

Step2. 计算中心化特征的协方差矩阵 $\Sigma$

$$\Sigma = \frac{1}{N-1} \mathbf{X} \mathbf{X}^T = \begin{bmatrix} 22.67 & 12 \\ 12 & 6.67 \end{bmatrix}$$

Step3. 计算 $\Sigma$ 的特征值和特征向量

$$\lambda = [29.09, 0.24], \quad \mathbf{U} = \begin{bmatrix} 0.88 & -0.47 \\ 0.47 & 0.88 \end{bmatrix}$$

Step4. 选取较大的特征值对应的特征向量，将其作为投影矩阵，计算投影后的特征

$$\hat{\mathbf{X}} = \begin{bmatrix} 3 & 5 & -3 & -5 \\ 1 & 3 & -1 & -3 \end{bmatrix}^T \begin{bmatrix} 0.88 \\ 0.47 \end{bmatrix} = [3.12 \quad 5.82 \quad -3.12 \quad -5.82]$$

# 监督K-L变换：以从类均值中 提取判别信息为例

假设有两类样本  $X_1 = \begin{bmatrix} 3 & 5 \\ 1 & 3 \end{bmatrix}$ ,  $X_2 = \begin{bmatrix} -3 & -5 \\ -1 & -3 \end{bmatrix}$ , 先验概率均为0.5, 样本均值  $\mu = \mathbf{0}$ , 目标仍为将其降到一维。

Step1. 计算类均值向量及协方差矩阵

$$\mu_1 = \begin{bmatrix} 4 \\ 2 \end{bmatrix}, \quad \Sigma_1 = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}, \quad \mu_2 = \begin{bmatrix} -4 \\ -2 \end{bmatrix}, \quad \Sigma_2 = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$$

Step2. 计算总类内散度矩阵  $S_w$  及其特征值和特征向量, 并将原始特征投影到不同特征向量的空间中

$$S_w = \frac{1}{2}\Sigma_1 + \frac{1}{2}\Sigma_2 = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}, \quad \lambda = [4, 0], \quad U = \begin{bmatrix} 0.707 & -0.707 \\ 0.707 & 0.707 \end{bmatrix}$$

Step3. 计算  $S_b$ :  $S_b = \sum_{i=1}^2 0.5 * (\mu_i - \mu)(\mu_i - \mu)^T = \begin{bmatrix} 16 & 8 \\ 8 & 4 \end{bmatrix}$

Step4. 计算性能指标可得  $J(X_2) > J(X_1)$ , 所以选择特征值4对应的特征向量作为投影向量:

$$\hat{X} = \begin{bmatrix} 3 & 5 & -3 & -5 \\ 1 & 3 & -1 & -3 \end{bmatrix}^T \begin{bmatrix} 0.707 \\ 0.707 \end{bmatrix} = [2.83 \quad 5.66 \quad -2.83 \quad -5.66]$$



# 基于类别可分性判据的特征提取

- 定义基于类内类间距离的可分性判据  $J(W)$
- 目标： 求  $W^*$  使得  $J(x) = \max_{\{w\}} J(W^T \mathbf{x})$

准则函数  $J(W)$  : (变换后的可分离性判据)

$$J_1(W) = \text{tr}(W^T (S_w + S_b) W) \quad J_2(W) = \text{tr}[(W^T S_w W)^{-1} (W^T S_b W)]$$

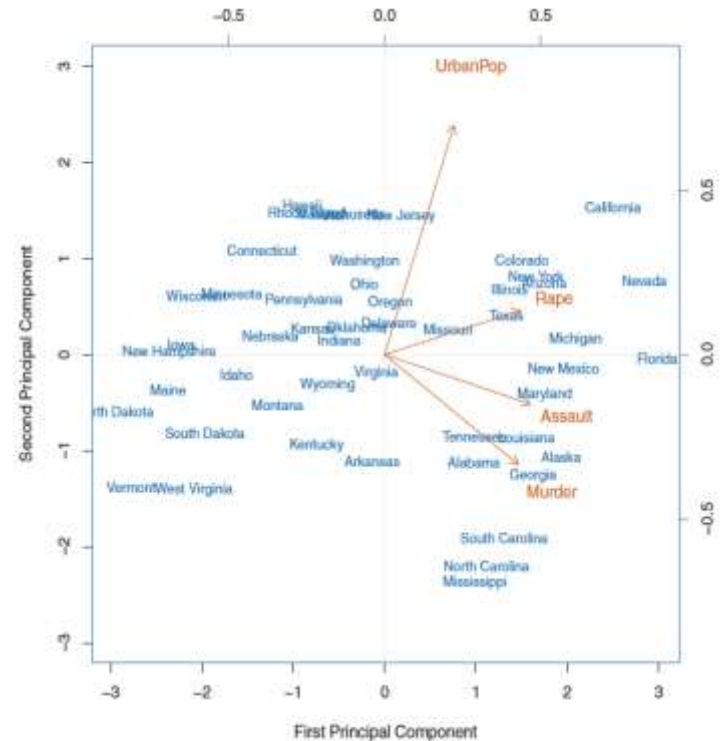
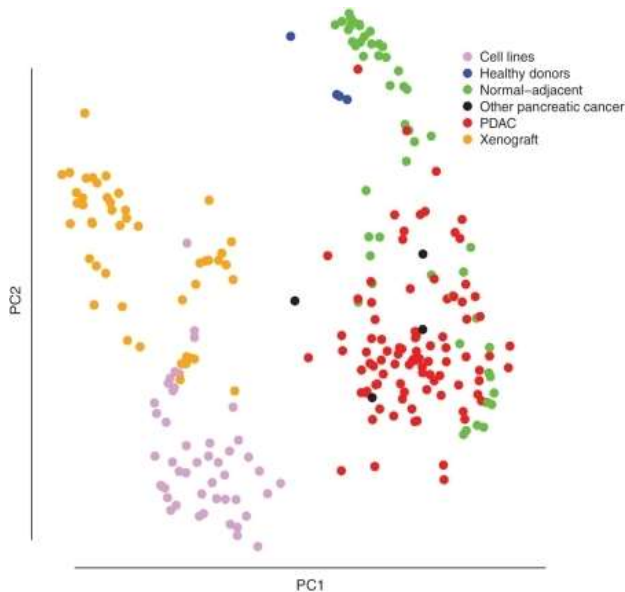
$$J_3(W) = \ln \frac{|W^T S_b W|}{|W^T S_w W|} \quad J_4(W) = \frac{\text{tr}(W^T S_b W)}{\text{tr}(W^T S_w W)}$$

$$J_5(W) = \frac{|W^T \Sigma W|}{|W^T S_w W|} \quad \Sigma = S_w + S_b$$

回顾：Fisher 线性判别函数  $k=1$ 时的特例

# 高维数据的低维显示

- 出发点：我们的大脑通常只能思考三维以下的空间关系
- 将数据投影到低维，有助于我们对样本之间的关系进行观察和理解



**FIGURE 10.1.** The first two principal components for the USArrests data. The blue state names represent the scores for the first two principal components. The orange arrows indicate the first two principal component loading vectors (with axes on the top and right). For example, the loading for Rape on the first component is 0.54, and its loading on the second principal component 0.17 (the word Rape is centered at the point (0.54, 0.17)). This figure is known as a biplot, because it displays both the principal component scores and the principal component loadings.

# 多维尺度法

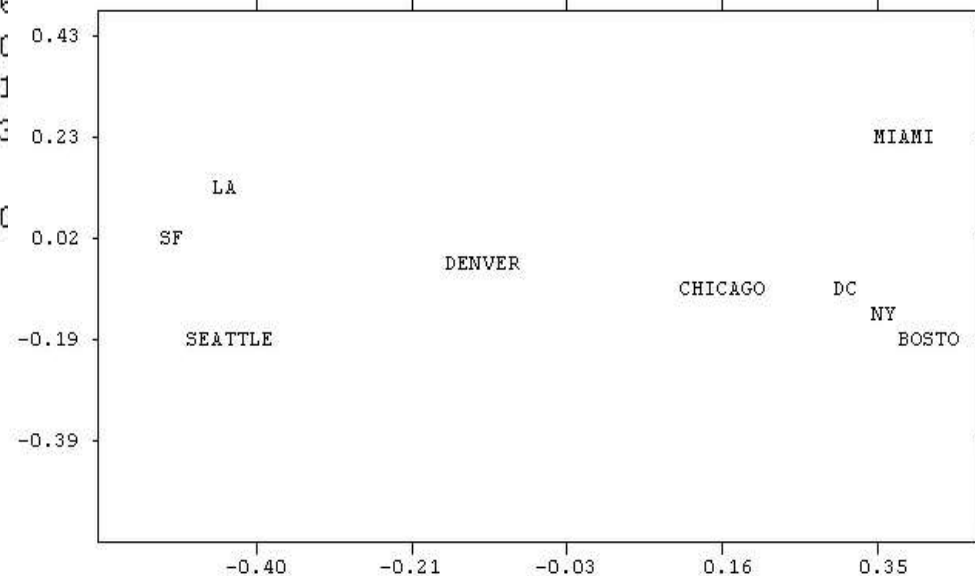
## (Multidimensional Scaling, MDS)

- 中译不统一：“多维尺度分析”、“多维标度分析”、“多维尺度模型”、“多维排列模型”、“多维标度”、“多维尺度”、“多元尺度法”、“多维标度法”等等
- 基本出发点
  - 根据样本之间的距离关系或不相似度关系在低维空间里生成对样本的表示
  - 把样本之间的距离关系或不相似关系在二维或三维空间里表示出来
- 给定距离，求（相对）坐标

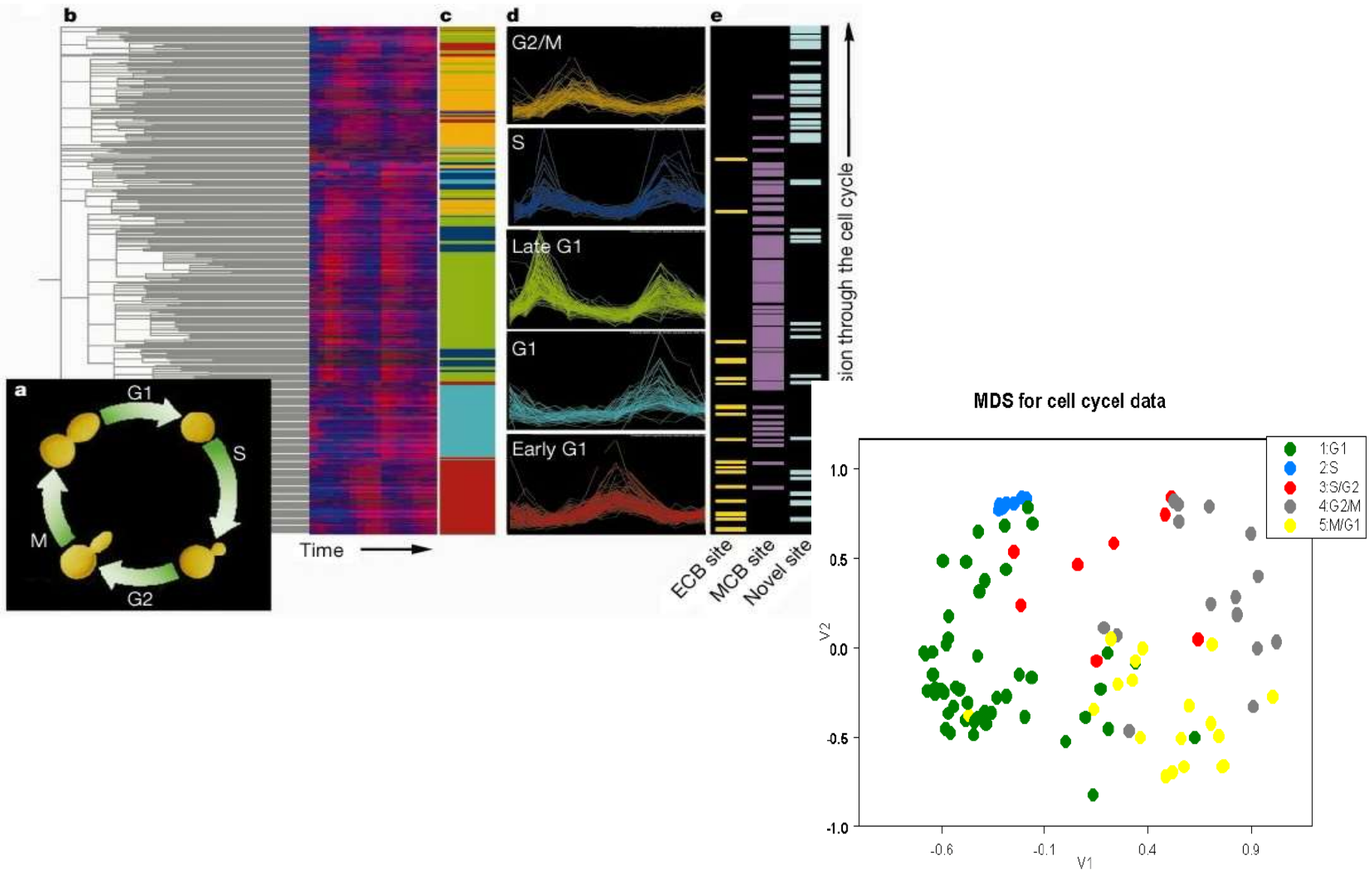
# 地图的例子



		1	2	3	4	5	6	7	8	9
		BOST	NY	DC	MIAM	CHIC	SEAT	SF	LA	DENV
1	BOSTON	0	206	429	1504	963	2976	3095	2979	1949
2	NY	206	0	233	1308	802	2815	2934	2786	1771
3	DC	429	233	0	1075	671	2684	2799	2631	1616
4	MIAMI	1504	1308	1075	0	1329	3273	3053	2687	2037
5	CHICAGO	963	802	671	1329	0	2013	2142	2054	996
6	SEATTLE	2976	2815	2684	3273	2013	0	808	1131	1307
7	SF	3095	2934	2799	3053	2142	808	0	379	1235
8	LA	2979	2786	2631	2687	2054	1131	379	0	1000
9	DENVER	1949	1771	1616	2037	996	1307	1235	1000	0

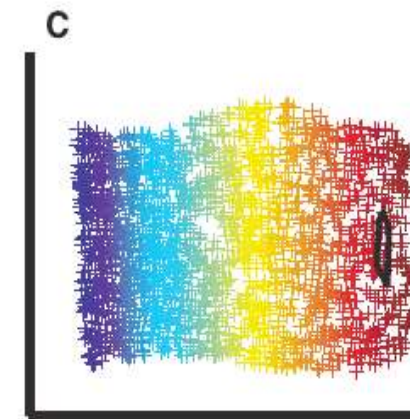
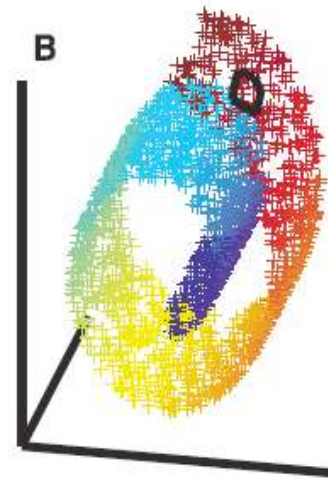
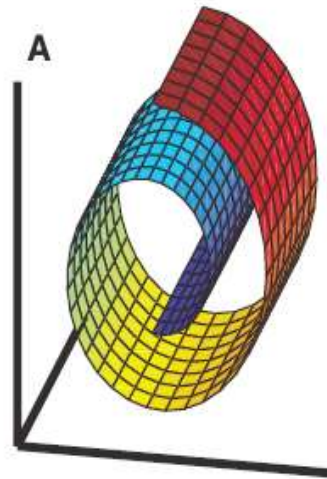
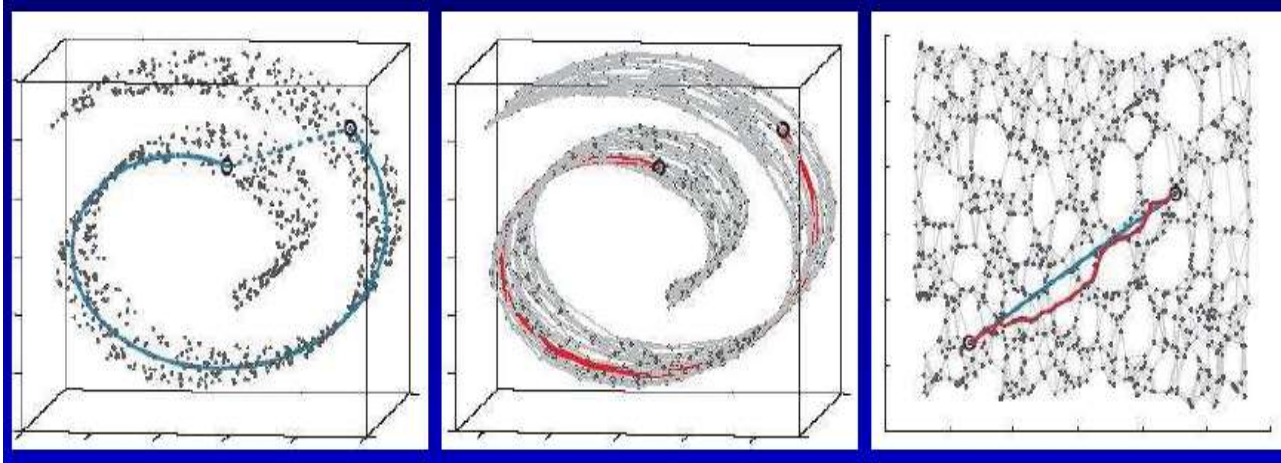


# 细胞周期的数据





# 非线性降维方法



# LLE: locally linear embedding

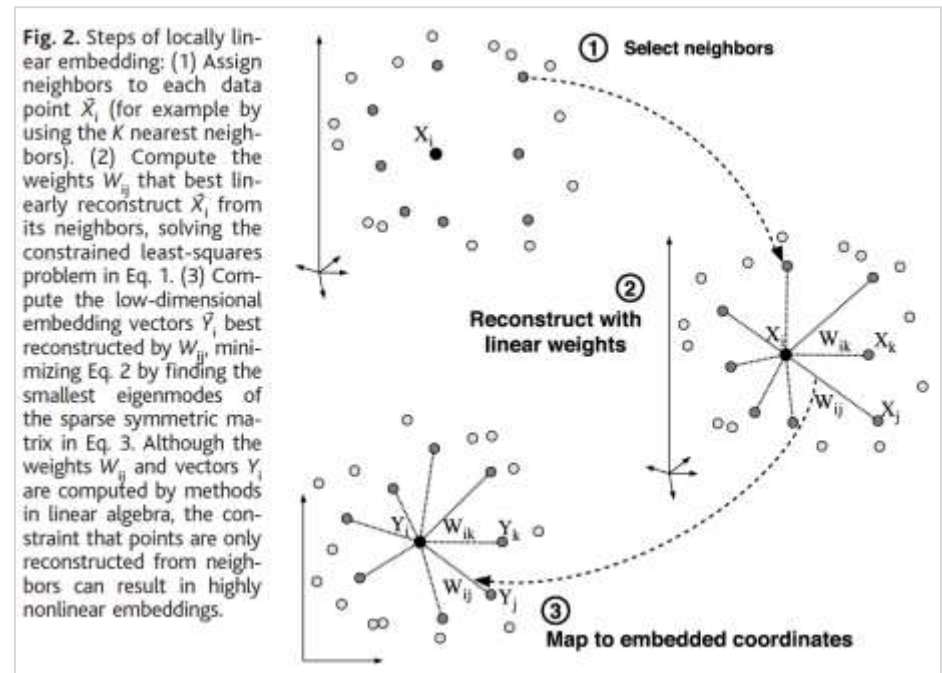
## Reconstruction Error

原空间里选择近邻，得到由近邻重构X误差最小的权重 $W_{ij}$

$$\epsilon(W) = \sum_i \left| \vec{X}_i - \sum_j W_{ij} \vec{X}_j \right|^2$$

利用 $W_{ij}$ ，在低位空间里得到重构误差最小的Y

$$\Phi(Y) = \sum_i \left| \vec{Y}_i - \sum_j W_{ij} \vec{Y}_j \right|^2$$



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

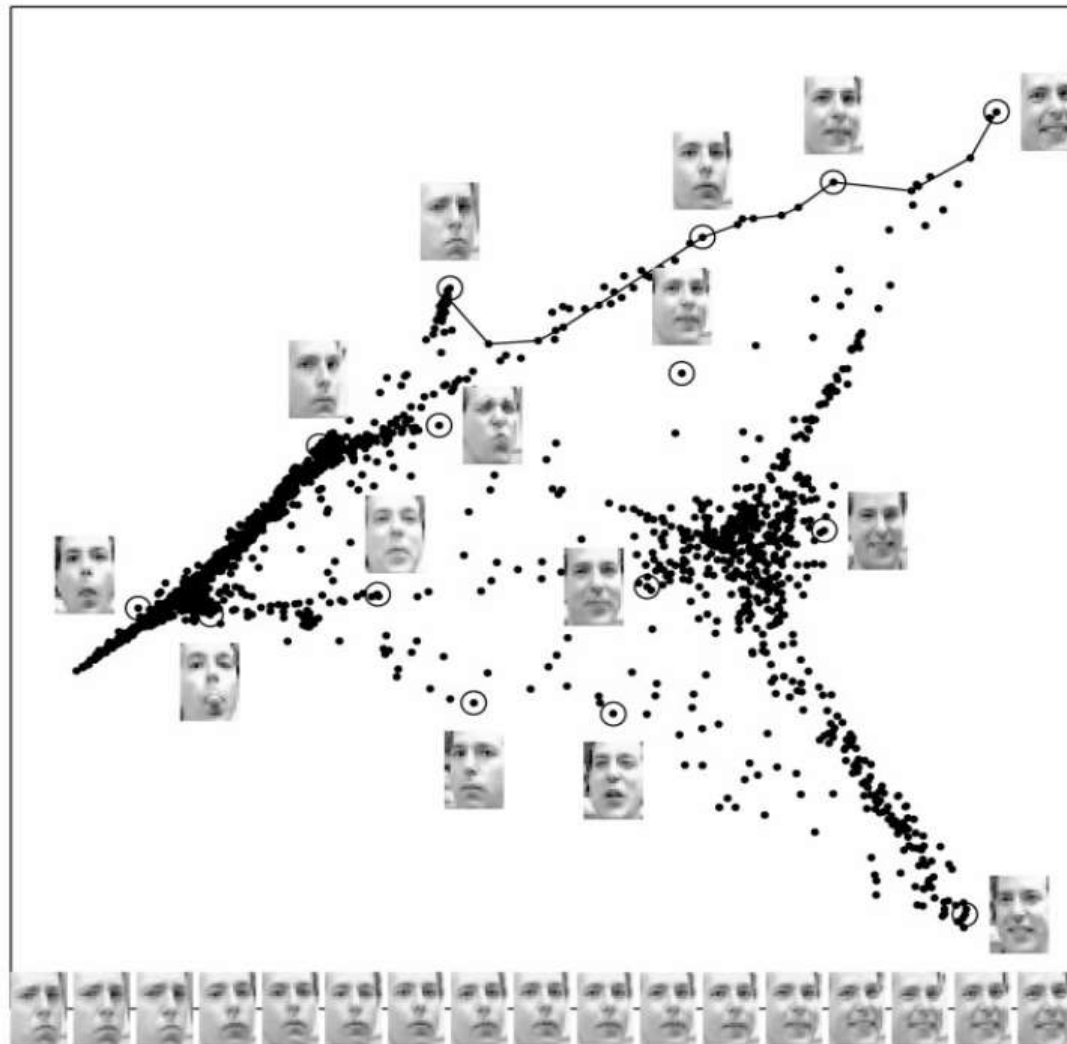
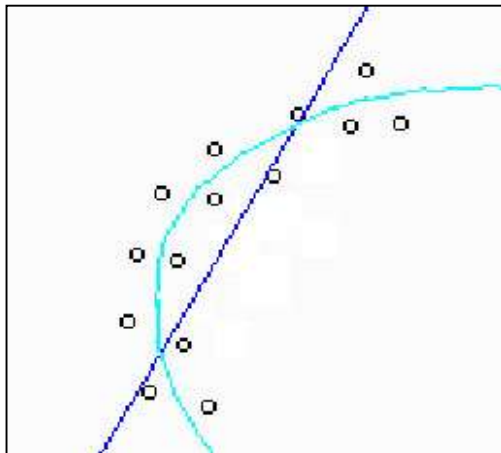
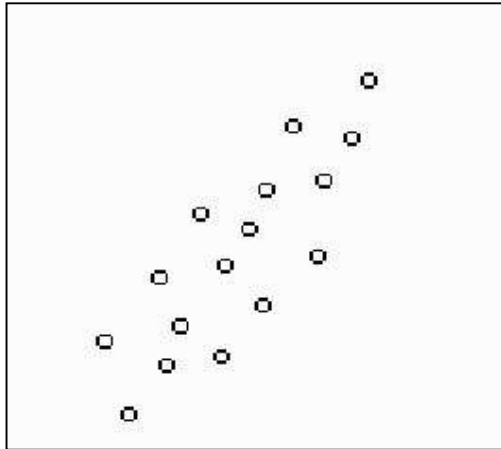


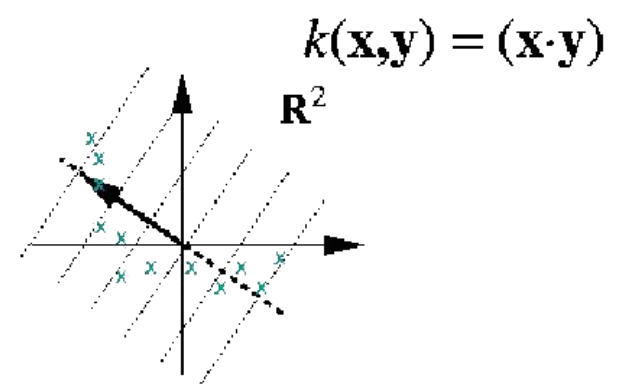
Fig. 3. Images of faces (11) mapped into the embedding space described by the first two coordinates of LLE. Representative faces are shown next to circled points in different parts of the space. The bottom images correspond to points along the top-right path (linked by solid line), illustrating one particular mode of variability in pose and expression.



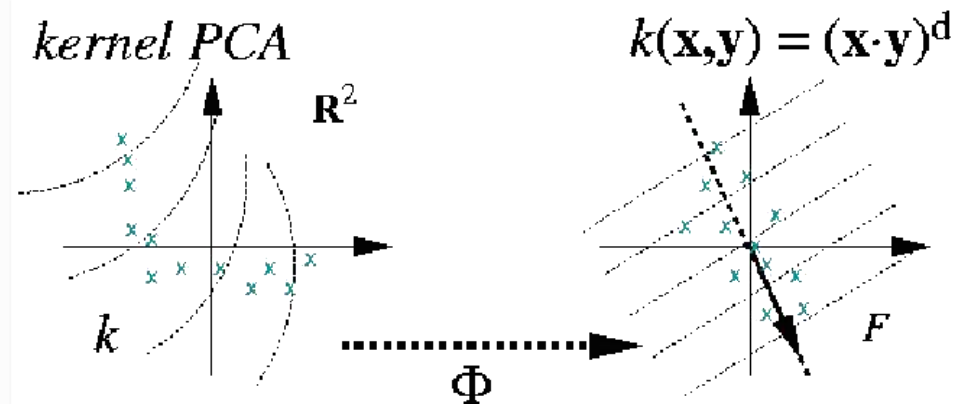
# KPCA (Kernel PCA)



*linear PCA*



*kernel PCA*



# t-SNE 方法

(t-Distributed Stochastic Neighbor Embedding)

- 利用概率分布来度量样本之间的距离（ $j$ 样本作为 $i$ 样本近邻的条件概率）

- 高维空间中的分布：
$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma_i^2)}$$

- 降维后的分布：
$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}}$$

- 分布之间的差异程度（距离）：KL distance

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

- <http://lvdmaaten.github.io/tsne/>
- <https://distill.pub/2016/misread-tsne/>

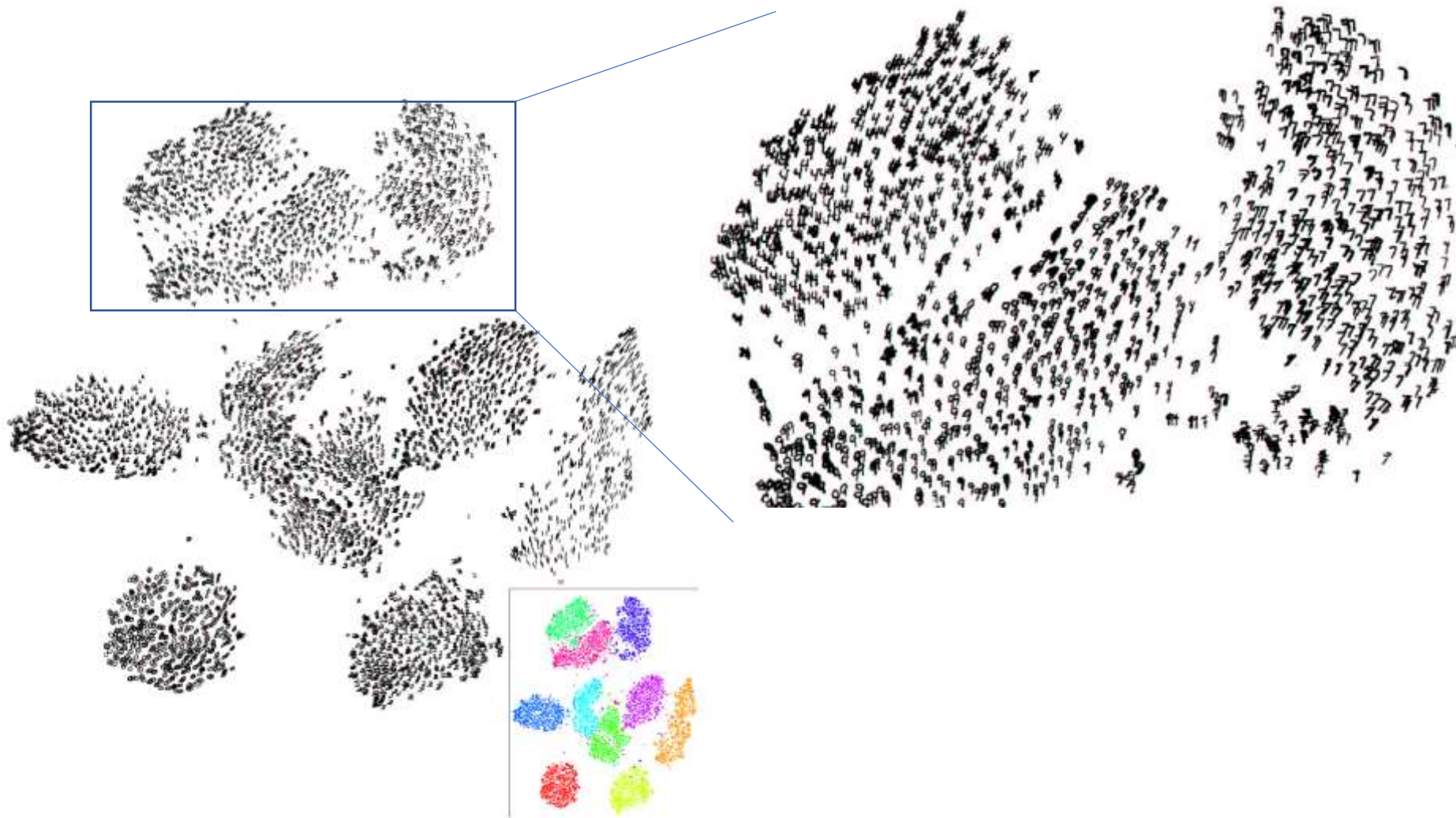
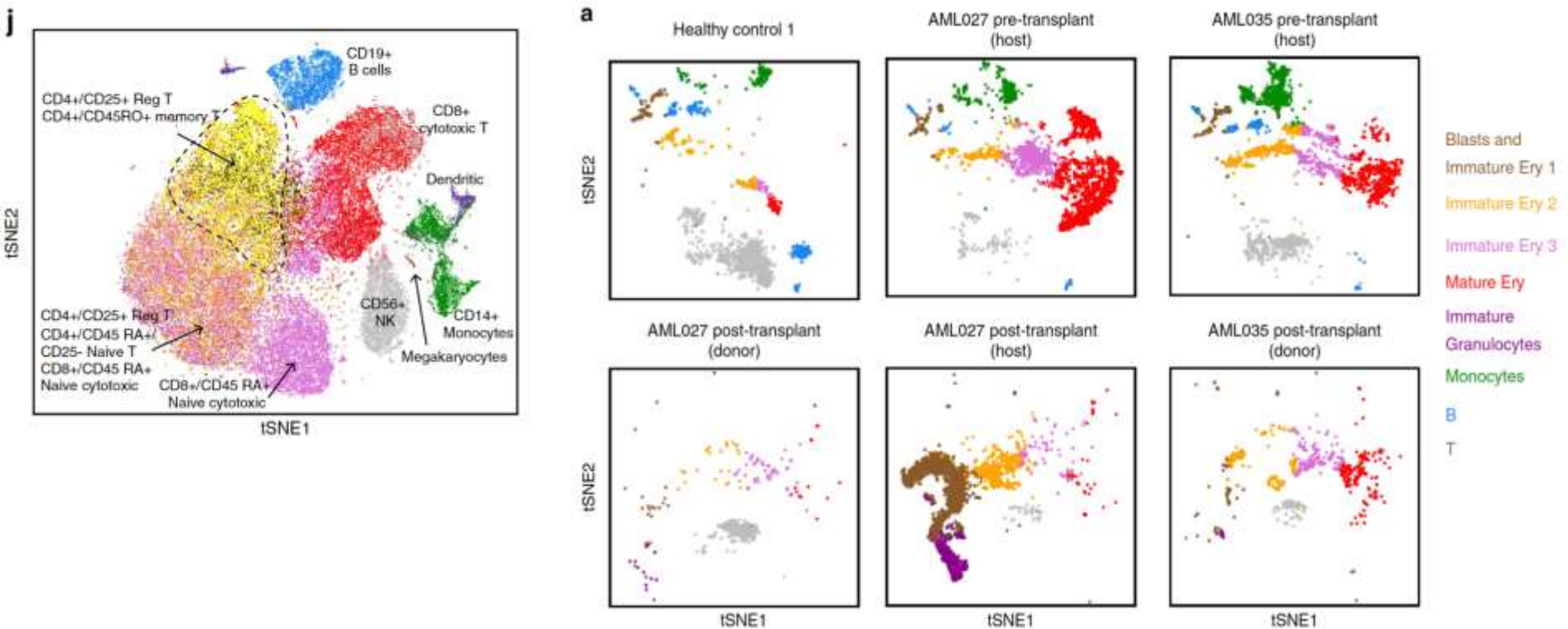


Figure 7: Visualization of 6,000 digits from the MNIST data set produced by the random walk version of t-SNE (employing all 60,000 digit images).

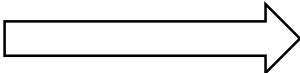
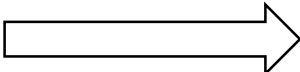
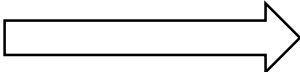
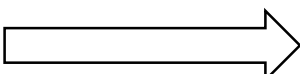
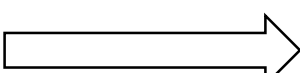
# 单细胞转录组研究



Nat. Commun 2017; 8:14049, Massively parallel digital transcriptional profiling of single cells

# 词语的embedding方法\*

- Embedding: 是指某个对象  $X$  被嵌入到另外一个对象  $Y$  中, 映射  $f: X \rightarrow Y$ ;
- Word embedding: 将一个字或词用向量表示, 称为“词向量”; 最简单的如one-hot embedding;
- 用于神经网络的输入/输出;

Paris		(1,0,0,0,0)
SeaWorld		(0,1,0,0,0)
Dolphin		(0,0,1,0,0)
Porpoise		(0,0,0,1,0)
Camera		(0,0,0,0,1)

- One-hot embedding存在的问题: (1) 没有包含词语的语义信息; (2) 词向量维度过高;

# 利用词语语境embedding

- Cocurrence matrix: 使用词语的语境意对词语进行embedding;
- 取词语前n个以及后n个词语作为“语境”，构成以下矩阵：
- 取词语所在行或列作为词向量；

I like CV.  
I enjor CV.  
I like NLP.  
I enjor deep learning.

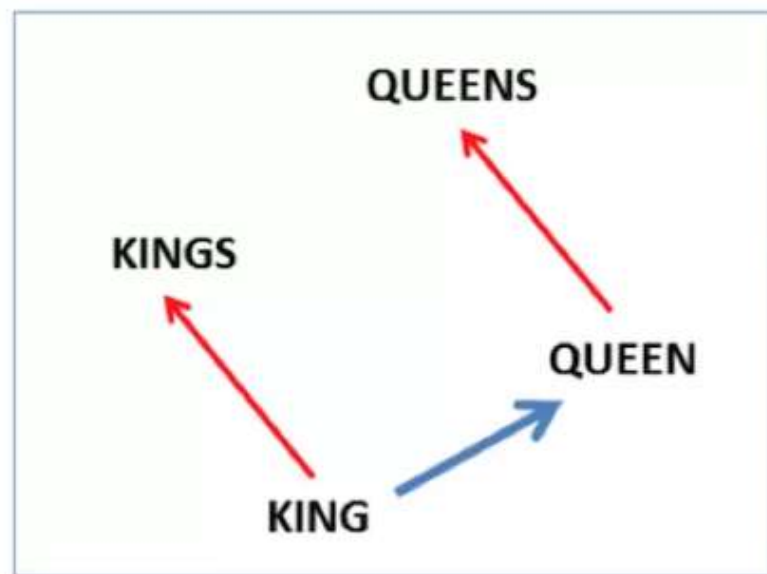
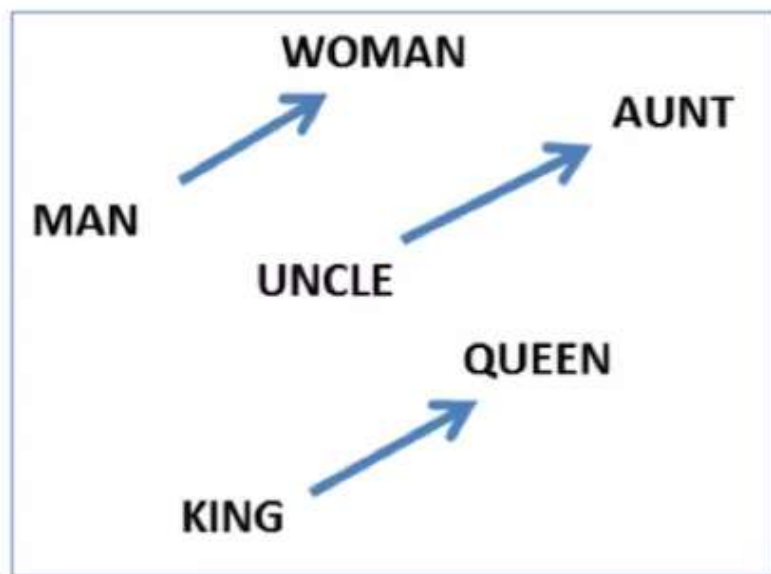


Counts	I	like	enjor	CV	NLP	deep	learning
I	0	2	2	0	0	0	0
like	2	0	0	1	1	0	0
enjor	2	0	0	1	0	1	0
CV	0	1	1	0	0	0	0
NLP	0	1	0	0	0	0	0
deep	0	0	1	0	0	0	1
learning	0	0	0	0	0	1	0

- 若两词向量具有较高的相似度，可以认为它们的语境意相似；
- 缺点：依然维度过高；

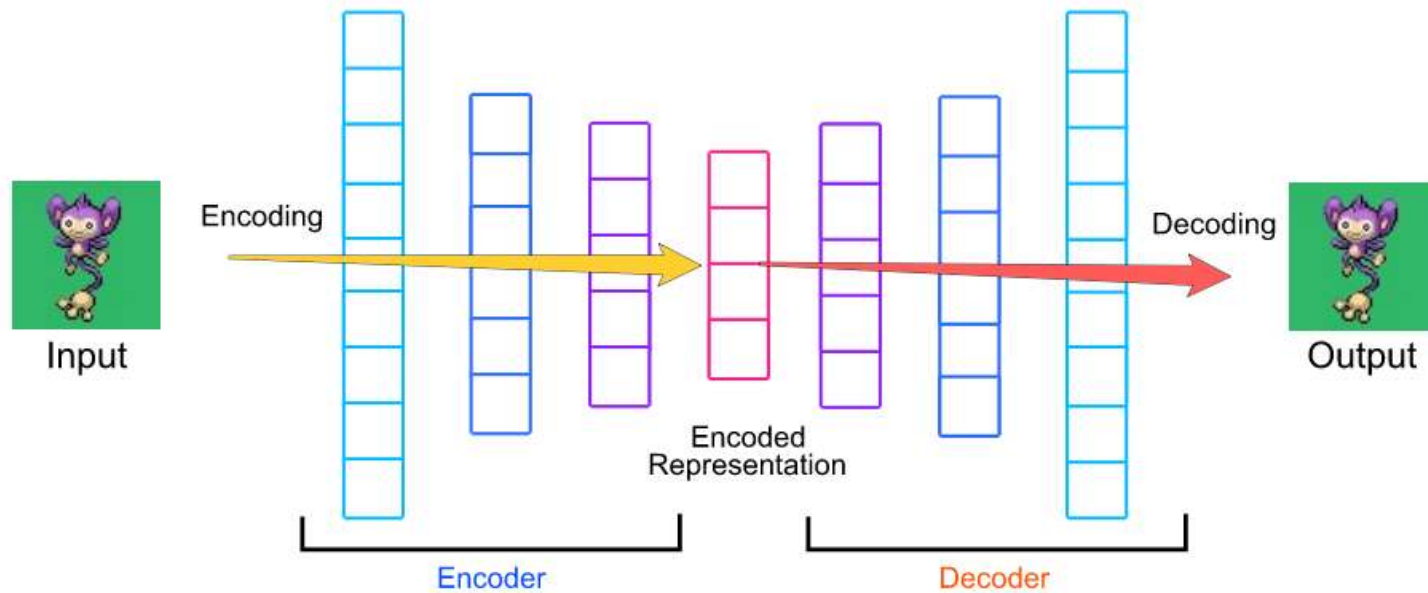
# Distributed representation

- 思路：通过训练，将每个词都映射到一个较短的词向量上来；
- 自动实现：单词语义相似性的度量；
- 如何实现？



# Auto-encoder

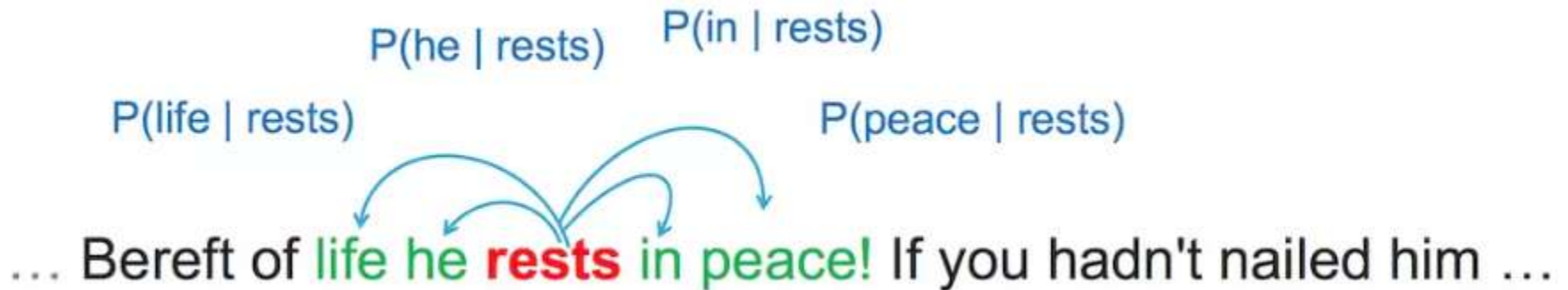
- 目标：学习输入数据的低维特征表示；
- 方法：网络输入、输出保持一致训练，取出低维中间层向量作为数据的特征表示；如下图：



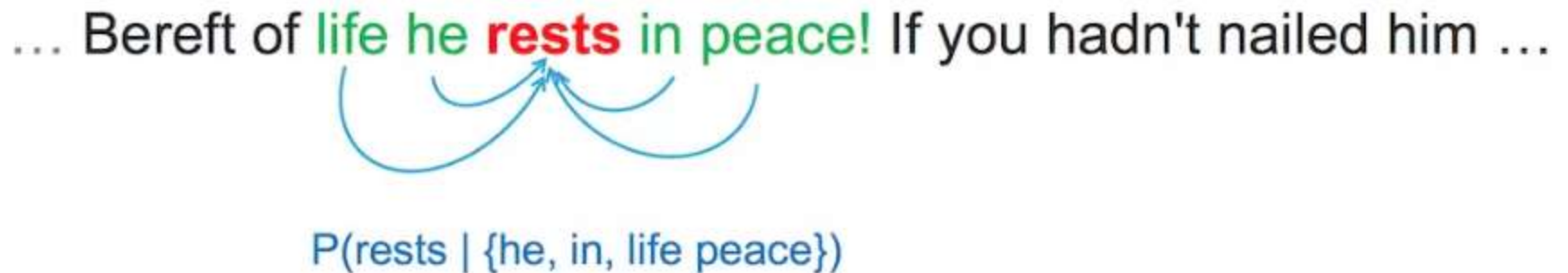


# Word2vec – Skip-gram and Cbow

## ➤ Skip-gram:



## ➤ Cbow:



# Word2vec – skip-gram

➤ Skip-gram矩阵计算:

- 输入:  $X(1 \times V) = [0, 1, 0, \dots, 0, 0]$ : 词语的one-hot编码向量
- Embedding Matrix:  $W(V \times N)$
- Hidden Layer:  $H(1 \times N)$ : 词语的embedding向量
- Context Matrix:  $\widetilde{W}(N \times V)$
- 输出:  $Y(1 \times V)$ : 词语的语境义向量

最终输出

$$\begin{array}{ccccccc} & & V \times N & & & N \times V & \\ & & \boxed{\text{Matrix } W} & & & \boxed{\text{Matrix } \widetilde{W}} & \\ \text{Input: } & \times & & = & \text{Embedding Matrix: } & \times & \\ X(1 \times V) & & & & H(1 \times N) & & \\ & & & & & & \\ & & & & & & \text{Output: } \\ & & & & & & Y(1 \times V) \end{array}$$

# Word2vec – skip-gram

- 由于输入是one-hot编码，故可以直接通过观察权值矩阵，得到该词语的词向量；

$$[0 \quad 0 \quad 0 \quad 1 \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$

输入one-hot编码

权值矩阵

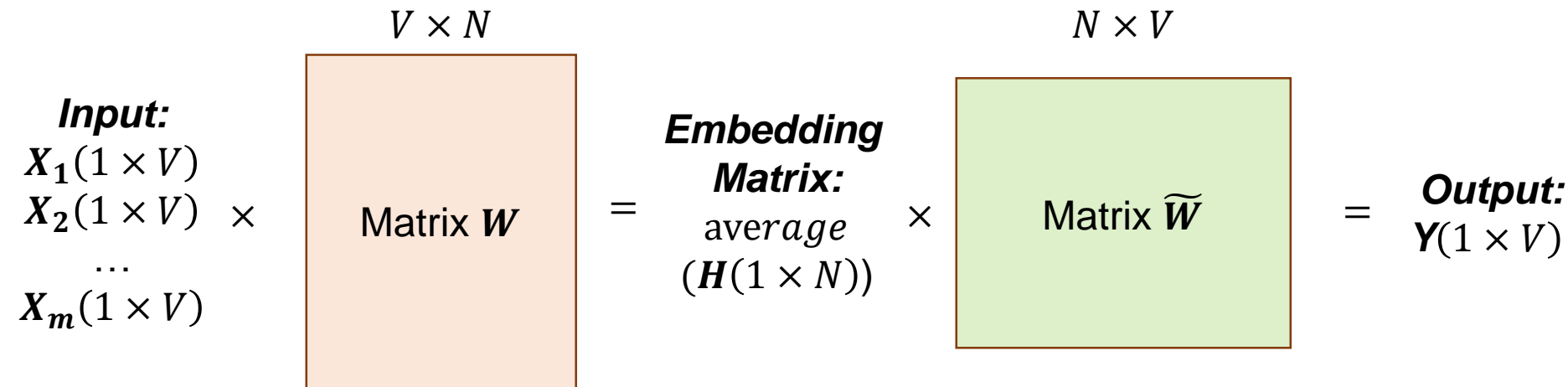
隐藏层输出

# Word2vec – cbow

➤ cbow矩阵计算：

- 输入：  $X_i(1 \times V) = [0, 1, 0, \dots, 0, 0]$  ( $i = 1 \dots m$ )：  $m$ 个语境词语的one-hot编码向量
- $W(V \times N)$ ： 用作Encoding的矩阵
- Hidden Layer：  $\text{average}(H(1 \times N))$ ： 词语的embedding向量
- $\widetilde{W}(N \times V)$ ： 用作Decoding的矩阵
- 输出：  $Y(1 \times V) = [0, 1, 0, \dots, 0, 0]$ ： 词语的one-hot编码向量

最终输出



# Word2vec结果

## ➤ 计算语境意相近的词语

Nearest to 买: 卖, 租, 买房子, 买楼, 买房, 狗狗, 买好, 买到,  
Nearest to 楼: 狗狗, 汇锋接, 房, 万, 炒炒, 楼先, 楼话, 转彬,  
Nearest to 博士: 过噶拉, 笨蛋, 这俩, 先买, 卖车, 天麒译, 名雅问,  
Nearest to 儿子: 女儿, 老婆, 女朋友, 老公, 太太, 妈妈, 家人, 弟弟,  
Nearest to 中意: 钟意, 感兴趣, 喜欢, 岩, 仲意, 满意, 熟悉, 狗狗,  
Nearest to 仲意: 钟意, 中意, 岩, 喜欢, 汇锋接, HX, 狗狗, 赏权,  
Nearest to 钟意: 中意, 喜欢, 岩, 仲意, 感兴趣, 满意, 啱, 得闲,  
Nearest to 睇: 狗狗, 体, 复睇, 体过, T, 搵, 细辉, 汇锋接,  
Nearest to 得闲: 有空, 翻来, 岩, 时间, 没空, 方便, 返来, 不得闲,  
Nearest to 倾成: 14F180, 左电, 推佐福盛, 迟到, 帮区, 90M42, 推掉,  
Nearest to 几好: OK, 不错, 好好, 一般, 一般般, 有意思, 佢, 外部,  
Nearest to 价钱: 价格, 价位, 单价, 总价, 价, 楼价, 楼层, 房价,

## ➤ 词语的加减法

**Obama + Russia - USA = Putin**

**Iraq - Violence = Jordan**

**Library - Books = Hall**

**Bonus) President - Power = Prime Minister**

- 斯坦福大学: GloVe ( Global Vectors for Word Representation )  
<https://nlp.stanford.edu/projects/glove>

# 深度神经网络的特征表示

## Reducing the Dimensionality of Data with Neural Networks

G. E. Hinton\* and R. R. Salakhutdinov

