吉林大学
JILIN UNIVERSITY

数据降维与流形学习

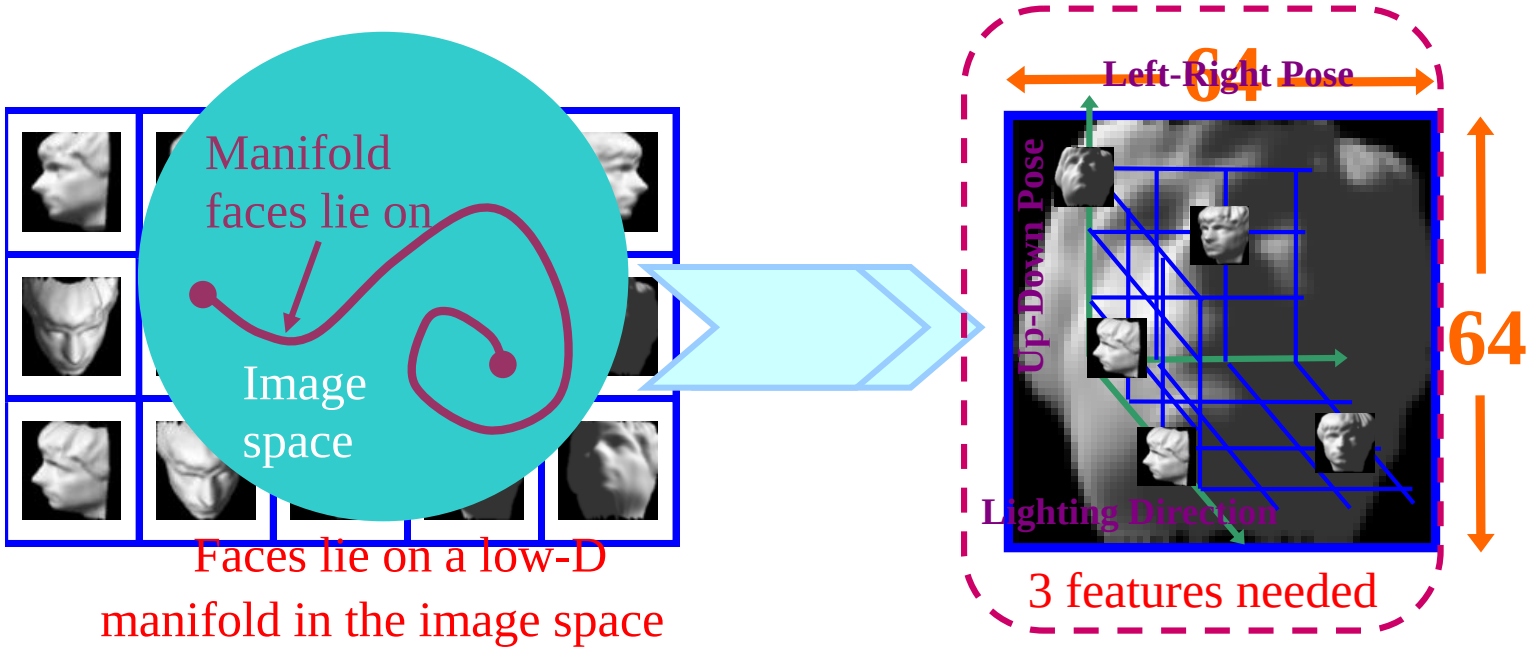
机器学习研究室

目录

- 概述
- 线性降维方法
- 非线性降维方法
- 流形学习

从降维问题说起

- 降维的动机
  - 原始观察空间中的样本具有极大的信息冗余
  - 样本的高维数引发分类器设计的“维数灾难”
  - 数据可视化、特征提取、分类与聚类等任务需求



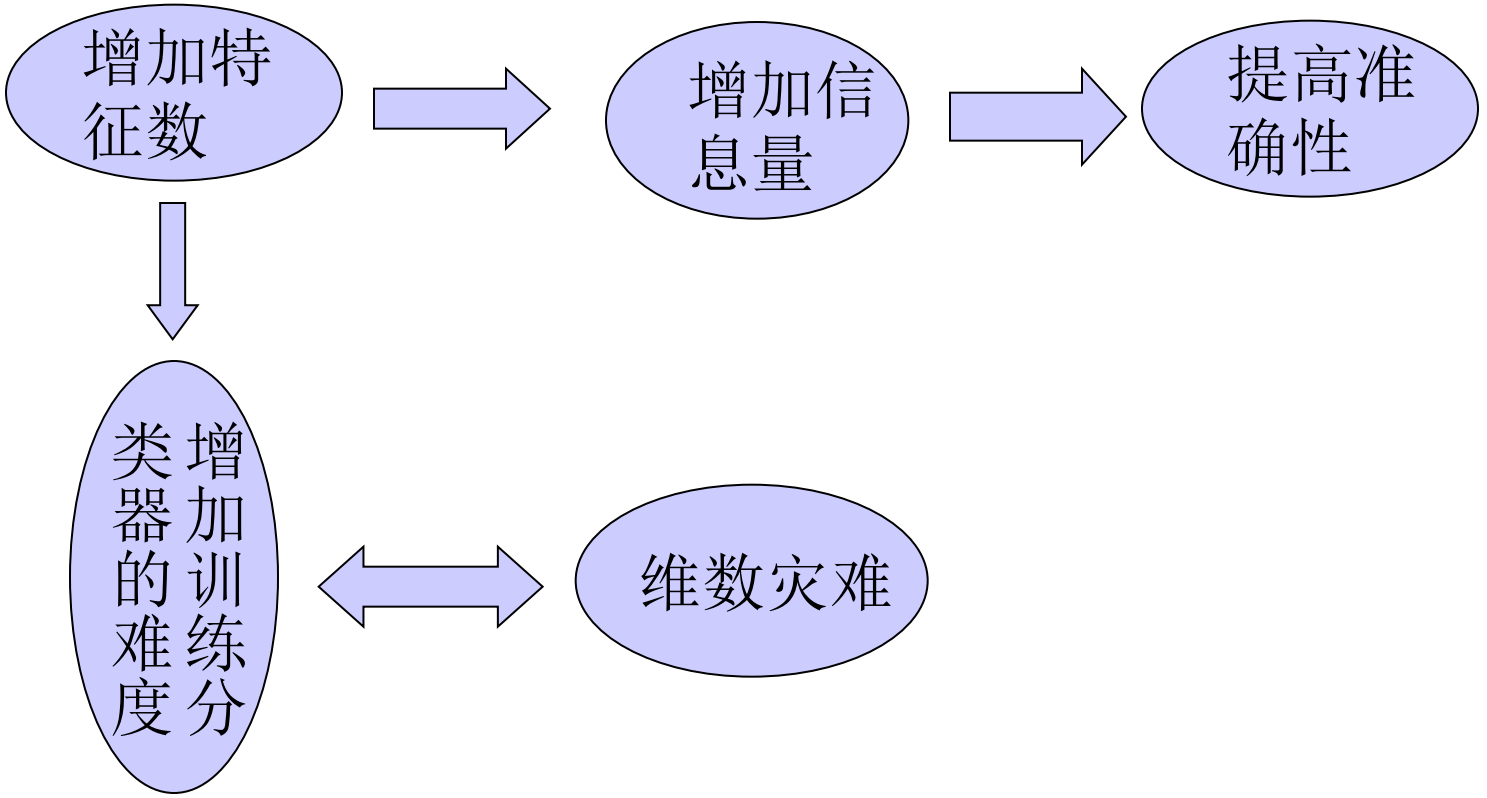
概 述

从降维问题说起

- 降维的动机
  - 减少数据存储所需空间和计算时间
  - 原始观察空间中的样本具有极大的信息冗余
  - 去除噪声，提高模型性能
  - 样本的高维数引发分类器设计的“维数灾难”
  - 数据可视化、特征提取、分类与聚类等任务需求

从降维问题说起

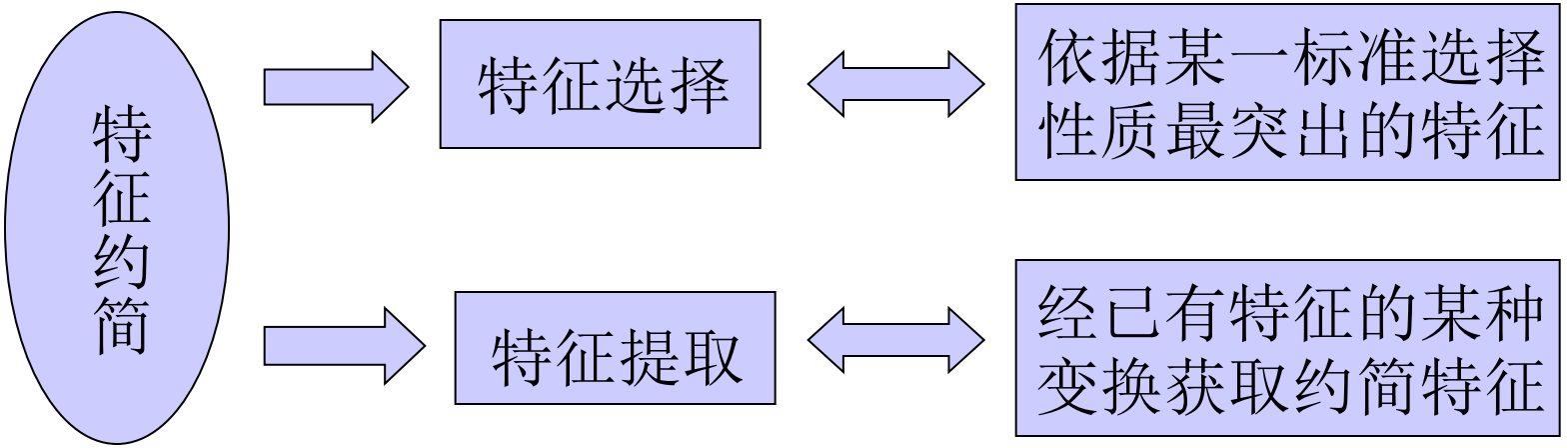
- 降维的动机



解决办法：选取尽可能多的，可能有用的特征，然后根据需要进行特征/维数约简.

从降维问题说起

- 降维的动机



实验数据分析，数据可视化（通常为2维或3维）等也需要维数约简

降维方法分类

- 依据样本信息是否利用
  - 无监督降维方法
  - 有监督降维方法
  - 半监督降维方法
- 根据所要处理的数据属性类型的不同
  - 线性降维：PCA, LDA,…
  - 非线性降维
    - 传统非线性降维：通过非线性变换将高维数据映射到低维空间，不一定假设数据嵌入在低维流形上。
    - 流形学习：明确假设数据分布在某个流形上，通常要求数据在流形的某个局部区域内具有良好的线性结构。

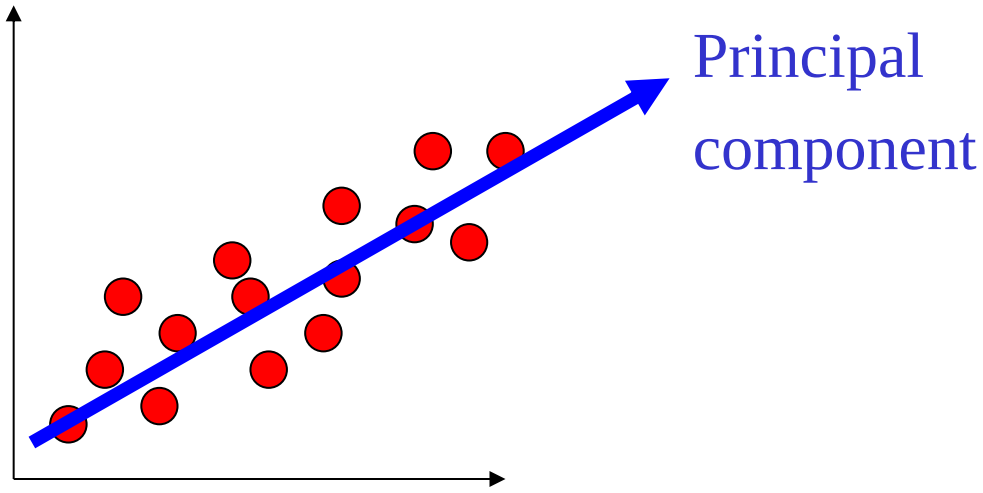
线性降维方法

线性降维方法

- 线性降维
  - 通过特征的线性组合来降维
  - 本质上是把数据投影到低维线性子空间
  - 线性方法相对比较简单且容易计算
  - 代表方法
    - 主成分分析(PCA)
    - 线性判别分析(LDA)
    - 奇异值分解(SVD)
    - 多维尺度变换(MDS)

线性降维方法

- 主成分分析(PCA) [Jolliffe, 1986]
  - 降维目的：寻找能够保持采样数据方差的最佳投影子空间
  - 求解方法：对样本的散度矩阵进行特征值分解, 所求子空间为经过样本均值, 以最大特征值所对应的特征向量为方向的子空间



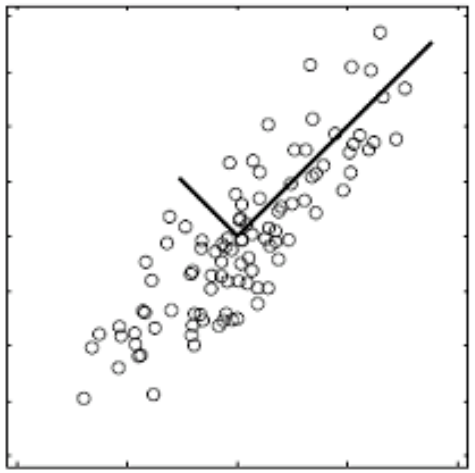
PCA: An Intuitive Approach

Let us say we have $\mathbf{x}_i$, i=1…N data points in p dimensions (p is large)

If we want to represent the data set by a single point $\mathbf{x}_0$, then

Sample mean

$$\mathbf{x}_0 = \mathbf{m} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$$



Can we justify this choice mathematically?

$$J_0(\mathbf{x}_0) = \sum_{i=1}^{N}\left\|\mathbf{x}_i - \mathbf{x}_0\right\|^2$$

It turns out that if you minimize J_0,
you get the above solution, namely, sample mean

PCA: An Intuitive Approach…

Representing the data set $\mathbf{x}_i$, i=1…N by its mean is quite uninformative

So let's try to represent the data by a straight line of the form:

$$\mathbf{x} = \mathbf{m} + a\mathbf{e}$$

This is equation of a straight line that says that it passes through **m**

e is a unit vector along the straight line

And the signed distance of a point **x** from **m** is a

The training points projected on this straight line would be

$$\mathbf{x}_i = \mathbf{m} + a_i\mathbf{e}, \quad i = 1...N$$

PCA: An Intuitive Approach…

Let's now determine a_i's

$$J_1(a_1, a_2, \ldots, a_N, \mathbf{e}) = \sum_{i=1}^{N} \left\| \mathbf{m} + a_i \mathbf{e} - \mathbf{x}_i \right\|^2$$

$$= \sum_{i=1}^{N} a_i^2 \| \mathbf{e} \|^2 - 2 \sum_{i=1}^{N} a_i \mathbf{e}^T (\mathbf{x}_i - \mathbf{m}) + \sum_{i=1}^{N} \| \mathbf{x}_i - \mathbf{m} \|^2$$

$$= \sum_{i=1}^{N} a_i^2 - 2 \sum_{i=1}^{N} a_i \mathbf{e}^T (\mathbf{x}_i - \mathbf{m}) + \sum_{i=1}^{N} \| \mathbf{x}_i - \mathbf{m} \|^2$$

Partially differentiating with respect to a_i we get: $a_i = \mathbf{e}^T (\mathbf{x}_i - \mathbf{m})$

Plugging in this expression for a_i in J_1 we get:

$$J_1(\mathbf{e}) = -\sum_{i=1}^{N} \mathbf{e}^T (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T \mathbf{e} + \sum_{i=1}^{N} \| \mathbf{x}_i - \mathbf{m} \|^2 = -\mathbf{e}^T S \mathbf{e} + \sum_{i=1}^{N} \| \mathbf{x}_i - \mathbf{m} \|^2$$

where $S = \sum_{i=1}^{N} (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T$ is called the **scatter matrix**

PCA: An Intuitive Approach…

So minimizing J_1 is equivalent to maximizing: $\mathbf{e}^T S\mathbf{e}$

Subject to the constraint that **e** is a unit vector: $\mathbf{e}^T\mathbf{e} = 1$

Use Lagrange multiplier method to form the objective function:

$$\mathbf{e}^T S\mathbf{e} - \lambda(\mathbf{e}^T\mathbf{e} - 1)$$

Differentiate to obtain the equation: $2S\mathbf{e} - 2\lambda\mathbf{e} = \mathbf{0}$ *or* $\mathbf{Se} = \lambda\mathbf{e}$

Solution is that **e** is the eigenvector of S corresponding to the largest eigenvalue

PCA: An Intuitive Approach…

The preceding analysis can be extended in the following way.

Instead of projecting the data points on to a straight line, we may now want to project them on a d-dimensional plane of the form:

$$\mathbf{x} = \mathbf{m} + a_1\mathbf{e}_1 + \cdots + a_d\mathbf{e}_d$$

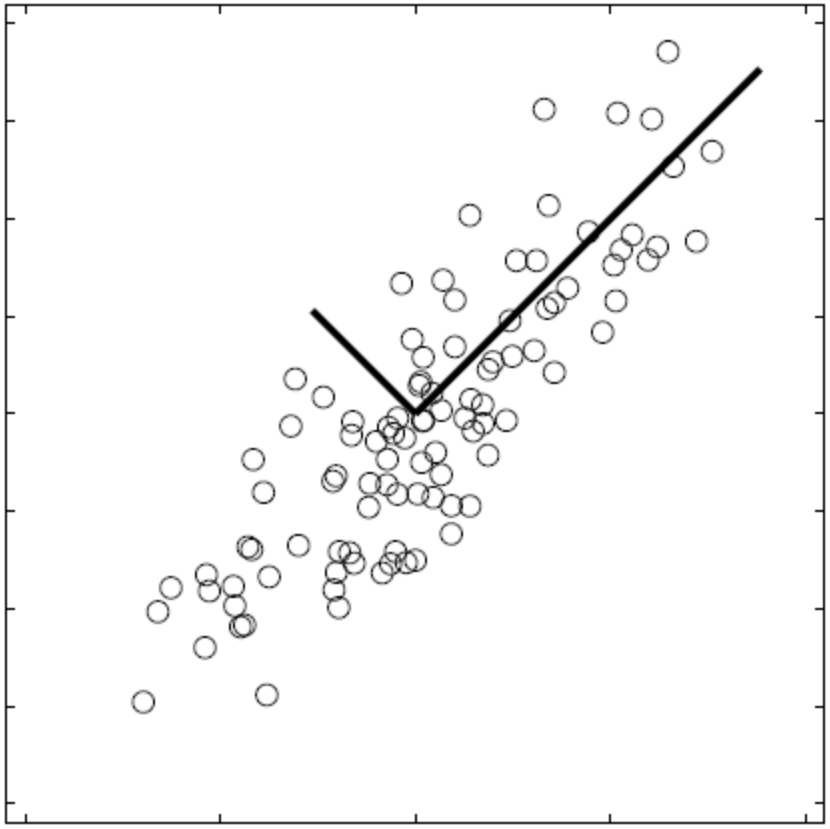
d is much smaller than the original dimension *p*

In this case one can form the objective function: $J_d = \sum_{i=1}^{N} \| (\mathbf{m} + \sum_{k=1}^{d} a_{ik}\mathbf{e}_k) - \mathbf{x}_i \|^2$

It can also be shown that the vectors $\mathbf{e}_1$, $\mathbf{e}_2$, …, $\mathbf{e}_d$ are *d* eigenvectors corresponding to *d* largest eigen values of the scatter matrix

$$S = \sum_{i=1}^{N} (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T$$

PCA: Visually



Data points are represented in a rotated orthogonal coordinate system: the origin is the mean of the data points and the axes are provided by the eigenvectors.

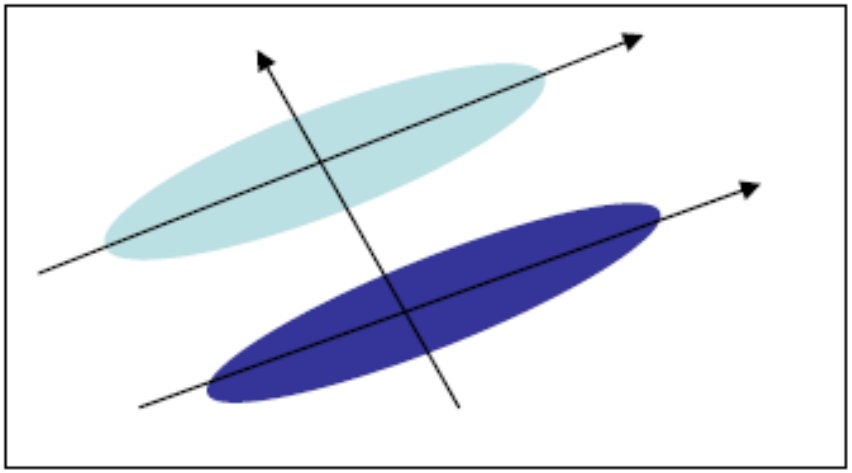
PCA Steps

- 设$x = (x_1, x_2, \cdots, x_n)^T$为n 维随机矢量
  - (1) 将原始观察数据进行0均值化；
  - (2) 之后组成0均值化后的观察矩阵X，每一列为一个观察样本，每一行为一维；
  - (3) 计算样本X的协方差矩阵 covX=COV(X)
  - (4) 计算covX的特征值和特征向量，并将特征值按从大到小排列
  - (5) 选取前m个最大特征值对应的特征向量组成矩阵V
  - (6) $Y=V^TX$，则Y为降维后的矩阵

降维后如何重构？

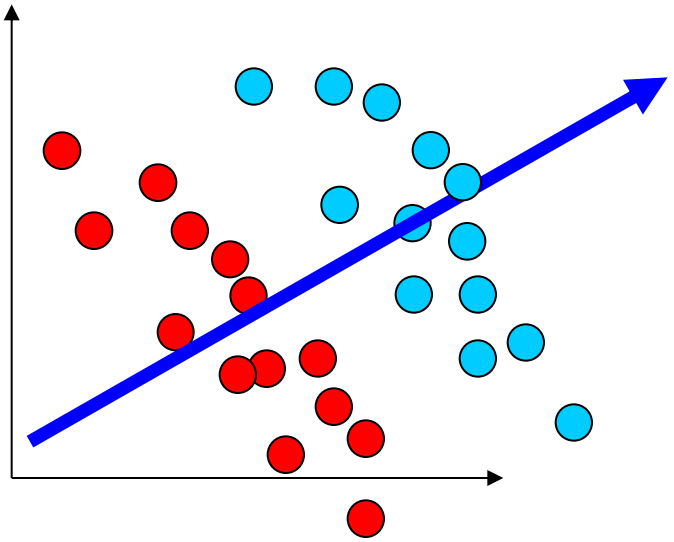
线性降维方法

- 主成分分析(PCA) [Jolliffe, 1986]
  - 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球的主轴方向.
  - PCA 是一种非监督的算法, 能找到很好地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



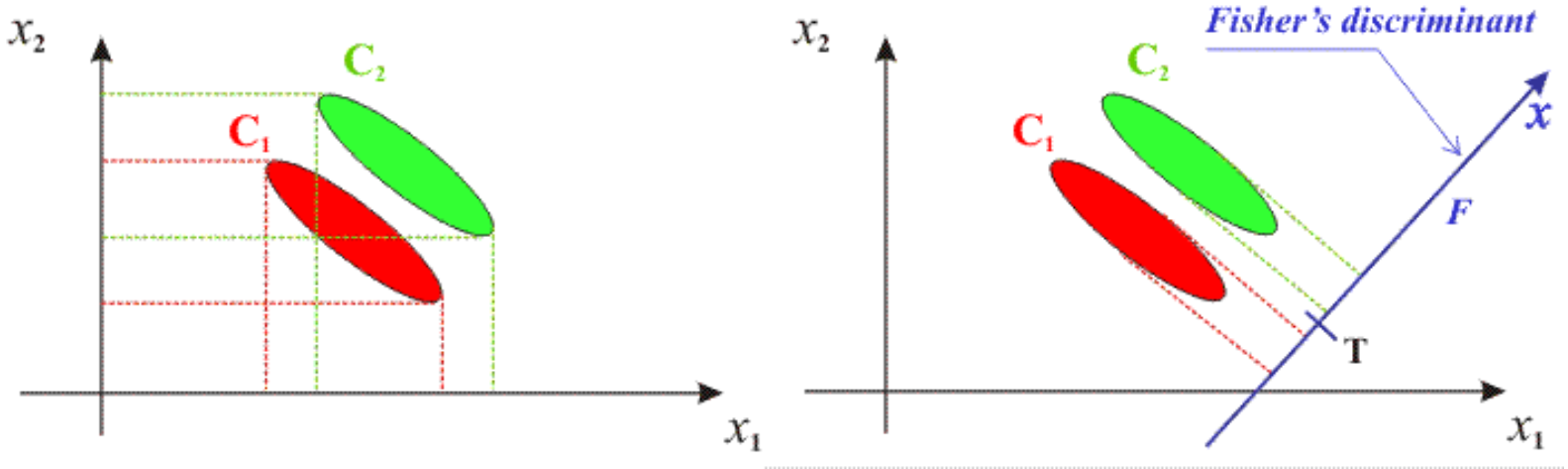
线性降维方法

- 线性判别分析(LDA) [Fukunaga, 1991]
  - 降维目的：寻找最能把两类样本分开的投影直线，使投影后两类样本的均值之差与投影样本的总类散度的比值最大
  - 求解方法：经过推导把原问题转化为关于样本集总类内散度矩阵和总类间散度矩阵的广义特征值问题

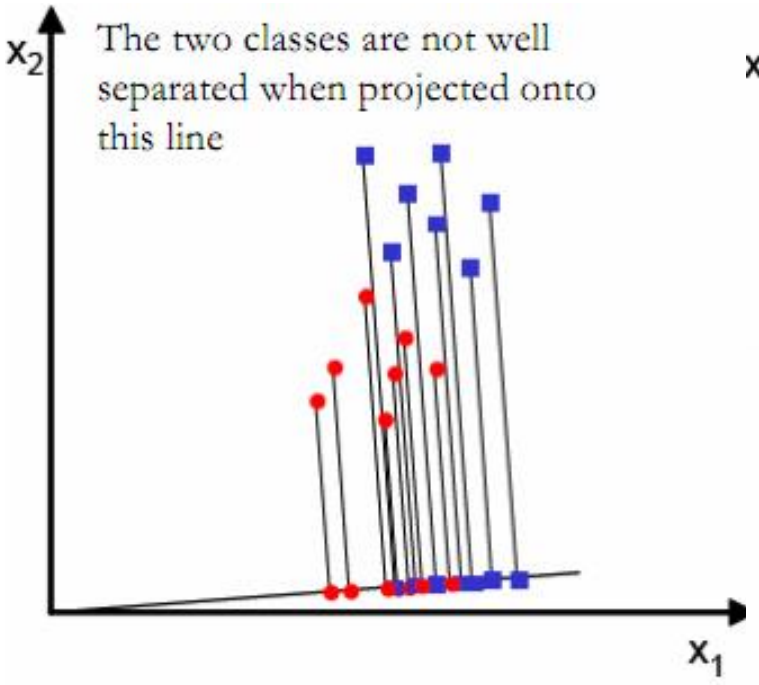


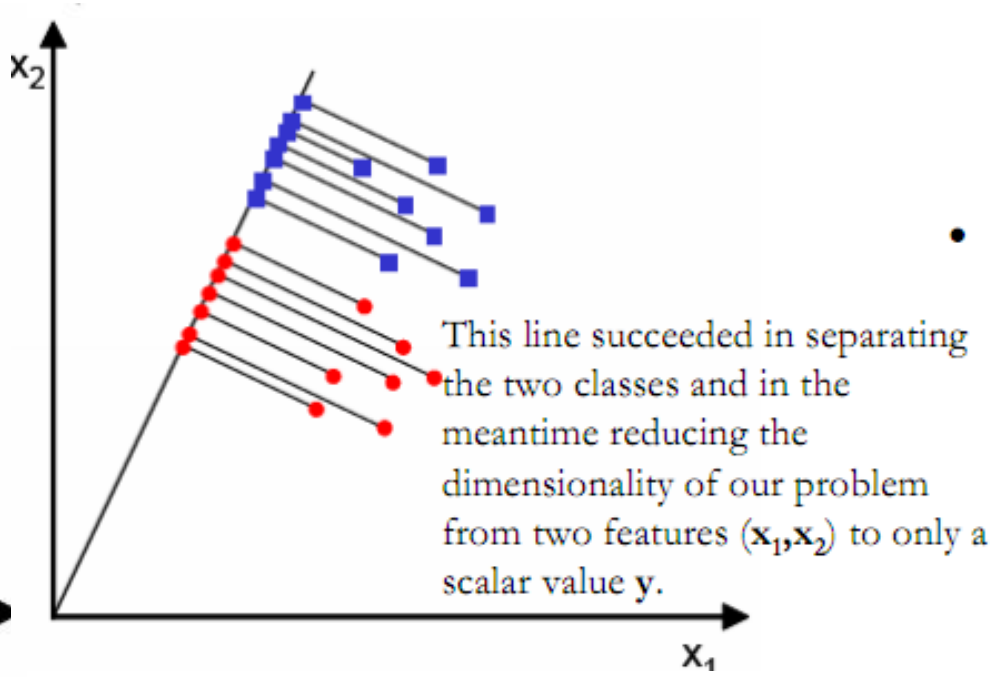
Best projection direction for classification

下面给出一个例子，说明LDA的目标：



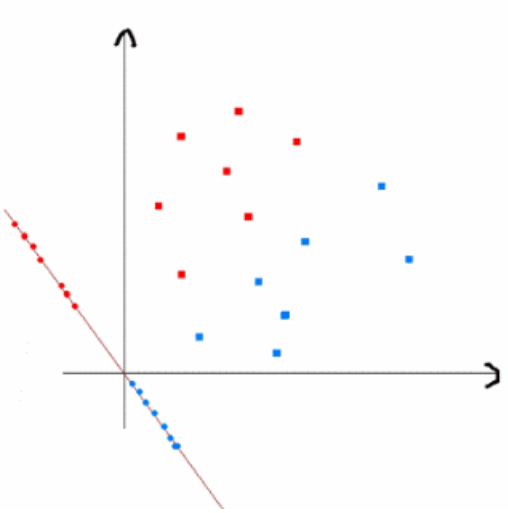
可以看到两个类别，一个绿色类别，一个红色类别。左图是两个类别的原始数据，现在要求将数据从二维降维到一维。直接投影到x1轴或者x2轴，不同类别之间 会有重复，导致分类效果下降。右图映射到的直线就是用LDA方法计算得到的，可以看到，红色类别和绿色类别在映射之后之间的距离是最大的，而且每个类别内 部点的离散程度是最小的（或者说聚集程度是最大的）。


x_2
The two classes are not well separated when projected onto this line
x_1


x_2
This line succeeded in separating the two classes and in the meantime reducing the dimensionality of our problem from two features ($\mathbf{x_1}$,$\mathbf{x_2}$) to only a scalar value **y**.
x_1

LDA

下面推导二分类LDA问题的公式：



假设用来区分二分类的直线（投影函数)为：

$$y = w^T x$$

LDA分类的一个目标是使得不同类别之间的距离越远越好，同一类别之中的距离越近越好，所以我们需要定义几个关键的值:

LDA

类别i的原始中心点(均值)为：（Di表示属于类别i的点):

$$m_i = \frac{1}{n_i} \sum_{x \in D_i} x$$

类别i投影后的中心点为：

$$\tilde{m}_i = w^T m_i$$

衡量类别i投影后，类别点之间的分散程度（方差）为：

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

最终我们可以得到一个下面的公式，表示LDA投影到w后的目标优化函数：

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

最大化该部分

LDA

分类的目标是，使得类别内的点距离越近越好（集中），类别间的点越远越好。

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

分母表示每一个类别内的方差之和，方差越大表示一个类别内的点越分散，分子为两个类别各自的中心点的距离的平方，我们最大化J(w)就可以求出最优的w

LDA

定义一个投影前的各类别分散程度的矩阵，如果某一个分类的输入点集Di里面的点距离这个分类的中心店mi越近，则Si里面元素的值就越小，如果分类的点都紧紧地围绕着mi，则Si里面的元素值越更接近0.

$$S_i = \sum_{x \in D_i} (x - m_i)(x - m_i)^T$$

带入Si，将J(w)分母化为：

$$\tilde{s}_i = \sum_{x \in D_i} (w^T x - w^T m_i)^2 = \sum_{x \in D_i} w^T (x - m_i)(x - m_i)^T w = w^T S_i w$$

$$\tilde{s}_1^2 + \tilde{s}_2^2 = w^T (S_1 + S_2) w = w^T S_w w$$

LDA

同样的将J(w)分子化为：

$$|\widetilde{m_1}-\widetilde{m_2}|^2 = w^T(m_1-m_2)(m_1-m_2)^T w = w^T S_B w$$

这样目标优化函数可以化成下面的形式：

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

推广到多类情况：

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

$$S_W=\sum_{i=1}^{c} S_i=\sum_{i=1}^{c}\sum_{x\in D_i}(x-m_i)(x-m_i)^T$$

$$S_B=\sum_{i=1}^{c} n_i(m_i-m)(m_i-m)^T$$

LDA

目标最大化函数：

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

在求导之前，需要对分母进行归一化，因为不做归一的话，w扩大任何倍，都成立，我们就无法确定w。因此令

$$w^T S_w w = 1$$

然后加入拉格朗日乘子，求导

$$c(w) = w^t S_B w - \lambda(w^t S_W w - 1)$$

$$\Rightarrow \frac{dc}{dw} = 2S_B w - 2\lambda S_W w = 0$$

$$\Rightarrow S_B w = \lambda S_W w$$

其中用到了矩阵微积分，求导时可以简单地把$w^T S_w w$当作$S_w w^2$看待。

LDA

若可逆，上式左右两端都乘以S_w^{-1}，化为

$$S_w^{-1}S_B w = \lambda w$$

也就是说， λ为$S_w^{-1}S_B$的特征值，w为相应的特征向量。这个公式称为Fisher linear discrimination。

LDA

$$S_w^{-1} S_B w = \lambda w$$

因为$S_B = (m_1 - m_2)(m_1 - m_2)^T$，于是

$$S_B w = (m_1 - m_2)(m_1 - m_2)^T w =: (m_1 - m_2)\lambda_w$$

因此，（3.4.18）式化为

$$S_w^{-1}(m_1 - m_2)\lambda_w = \lambda w$$

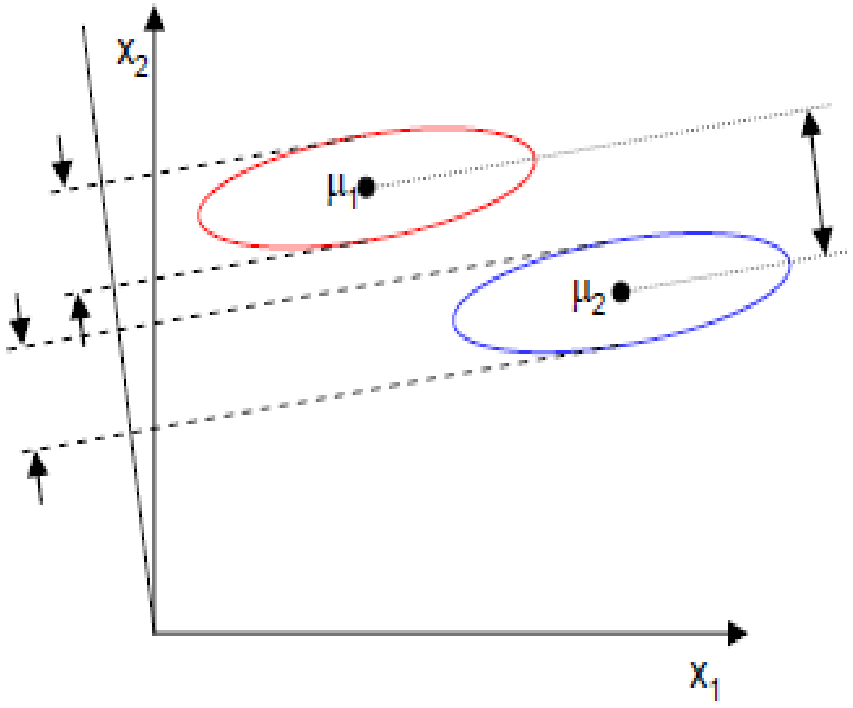
因为我们只关心w的方向而不关心其大小，所以可以令

$$w = S_w^{-1}(m_1 - m_2)$$

至此，我们只需要求出原始样本的均值和方差就可以求出最佳的方向w，这就是Fisher于1936年提出的线性判别分析。

LDA

上面二维样本的投影结果图：



奇异值分解（SVD）

- SVD：奇异值分解
- SVD可以看做是特征值分解的一种推广，或者说特征值分解可以看作是SVD的一种特例。当矩阵不是方阵时同样适用，应用很广。

SVD分解(1)

对任意矩阵$A \in F^{m\times n}$，$rank(A)=r$,总可以取A的如下分解

左奇异向量

右奇异向量

$$A = U_m \Sigma_A V_n^T$$

其中U、V为正交矩阵

非零奇异值，**从大到小**依次排序

$$\Sigma_A = \begin{pmatrix} \sigma_1 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \sigma_r & 0 \\ 0 & \cdots & 0 & 0 \end{pmatrix}, \qquad \left(A^T A\right) v_i = \lambda_i v_i, \quad \sigma_i = \sqrt{\lambda_i}$$

SVD分解(2)

$A = U\Sigma_A V^T$

$$\begin{cases} AV = U\Sigma_A V^T V = U\Sigma_A, \\ U^T A = U^T U\Sigma_A V^T = \Sigma_A V^T \end{cases}, \quad \begin{cases} U = (u_1, u_2, \cdots, u_m), & u_i: m \times 1\text{向量} \\ V = (v_1, v_2, \cdots, v_n), & v_1: n \times 1\text{向量} \end{cases}$$

$$AV = U\Sigma_A \Rightarrow (Av_1, \cdots, Av_n) = (u_1, \cdots, u_m)\Sigma_A$$

$$\Rightarrow (Av_1, \cdots, Av_n) = (u_1, \cdots, u_m)\begin{pmatrix} \sigma_1 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \sigma_r & 0 \\ 0 & \cdots & 0 & 0 \end{pmatrix} = (u_1\sigma_1, \cdots, u_2\sigma_r, 0)$$

$$\begin{cases} Av_1 = u_1\sigma_1 \\ \cdots \\ Av_r = u_r\sigma_r \end{cases}$$

奇异值**从大到小**
依次排序

SVD分解(3)

$$U^T A = \Sigma_A V^T$$

$$\begin{cases} U^T A = \begin{pmatrix} u_1^T \\ \cdots \\ u_m^T \end{pmatrix} A = \begin{pmatrix} u_1^T A \\ \cdots \\ u_m^T A \end{pmatrix} \\ \Sigma_A V^T = \begin{pmatrix} \sigma_1 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \sigma_r & 0 \\ 0 & \cdots & 0 & 0 \end{pmatrix} \begin{pmatrix} v_1^T \\ \vdots \\ v_n^T \end{pmatrix} = \begin{pmatrix} \sigma_1 v_1^T \\ \vdots \\ \sigma_r v_r^T \\ 0 \end{pmatrix} \end{cases}$$

奇异值**从大到小**依次排序

$$\Rightarrow \begin{pmatrix} u_1^T A \\ \cdots \\ u_m^T A \end{pmatrix} = \begin{pmatrix} \sigma_1 v_1^T \\ \vdots \\ \sigma_r v_r^T \\ 0 \end{pmatrix} \Rightarrow \begin{cases} u_1^T A = \sigma_1 v_1^T \\ \cdots \\ u_r^T A = \sigma_r v_r^T \end{cases}$$

SVD算法解析

$$\Sigma_A = \begin{pmatrix} \sigma_1 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \sigma_r & 0 \\ 0 & \cdots & 0 & 0 \end{pmatrix},$$

奇异值**从大到小**依次排序

$$\left(A^T A\right) v_i = \lambda_i v_i, \quad \sigma_i = \sqrt{\lambda_i}$$

可以看作是矩阵A的“平方”，而奇异值又是A“平方”阵的特征根的开方，因此**奇异值**可以看作是矩阵A的**“伪特征值”**，**左奇异向量**可以看作矩阵A的**“行特征向量”**，**右奇异向量**可以看作是矩阵A的**“列特征向量”**。

SVD矩阵近似（1）

$$A = U\Sigma_A V^T,$$

$U = (u_1, u_2, \cdots, u_m), u_i: n \times 1$向量（代表矩阵A的行特征）

$V = (v_1, v_2, \cdots, v_n), v_i: m \times 1$向量（代表矩阵A的列特征）

$$\Sigma_A = \begin{pmatrix} \sigma_1 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \sigma_r & 0 \\ 0 & \cdots & 0 & 0 \end{pmatrix},$$

奇异值从大到小依次排序

$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_r u_r v_r^T$$

SVD矩阵近似（2）

$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_r u_r v_r^T$$

σ_1，σ_2,…，σ_r为矩阵A的非零奇异值，从大到小依次排列，且下降的非常快。奇异值代表的对应的“行列特征”的重要程度。

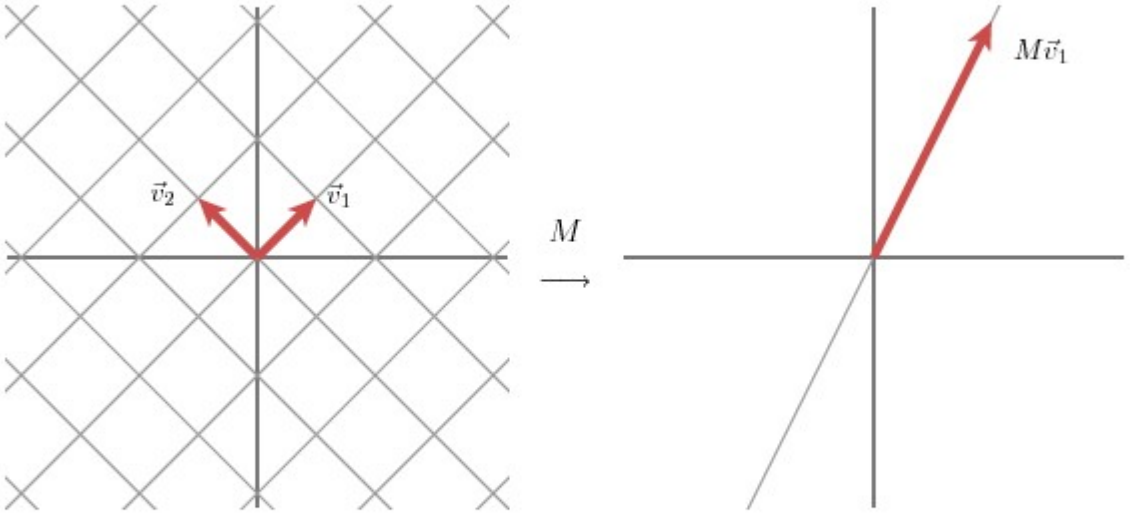
因此可以提取矩阵的前t个特征来近似矩阵。

$$A \approx A\sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_t u_t v_t^T$$

SVD实例1

$$M = \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix}$$

经过这个矩阵变换后的效果如下图所示



在这个例子中，第二个奇异值为 0，因此经过变换后只有一个方向上有表达。

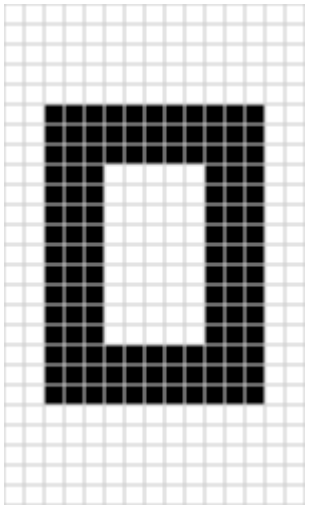
$$M = \mathbf{u}_1 \sigma_1 \mathbf{v}_1^T.$$

换句话说，如果某些奇异值非常小的话，其相对应的几项就可以不同出现在矩阵 M 的分解式中。因此，我们可以看到矩阵 M 的秩的大小等于非零奇异值的个数。

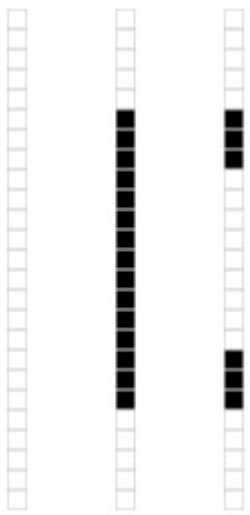
```
>> A=[1,1;2,2]

A =

     1     1
     2     2

>> [U,SIGMA,V]=svd(A)

U =

   -0.44721   -0.89443
   -0.89443    0.44721

SIGMA =

Diagonal Matrix

   3.1623e+000             0
             0   1.5701e-016

V =

   -0.70711   -0.70711
   -0.70711    0.70711
```

SVD实例2

一张 15 x 25 的图像数据　将图像表示成 15 x 25 （= 375） 的矩阵



该图像主要由下面三部分构成



$$M = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}$$

SVD实例2

如果我们对矩阵M进行奇异值分解以后，得到奇异值分别是

$$\sigma_1 = 14.72$$

$$\sigma_2 = 5.22$$

$$\sigma_3 = 3.31$$

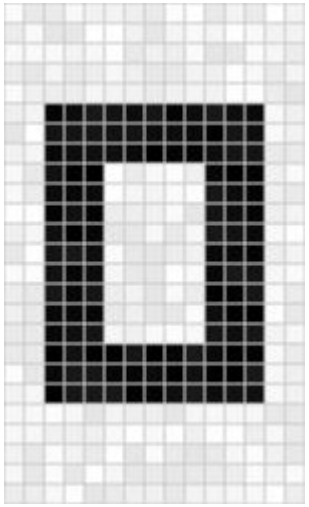
矩阵M就可以表示成

$$M=\mathbf{u}_1\sigma_1 \mathbf{v}_1^T + \mathbf{u}_2\sigma_2 \mathbf{v}_2^T + \mathbf{u}_3\sigma_3 \mathbf{v}_3^T$$

$\mathbf{v}_i$具有15个元素，$\mathbf{u}_i$ 具有25个元素，σ_i 对应不同的奇异值。如上图所示，我们就可以用123个元素来表示具有375个元素的图像数据了。

减噪(noise reduction)

前面的例子的奇异值都不为零，或者都还算比较大，下面我们来探索一下拥有零或者非常小的奇异值的情况。通常来讲，大的奇异值对应的部分会包含更多的信息。比如，我们有一张扫描的，带有噪声的图像，如下图所示

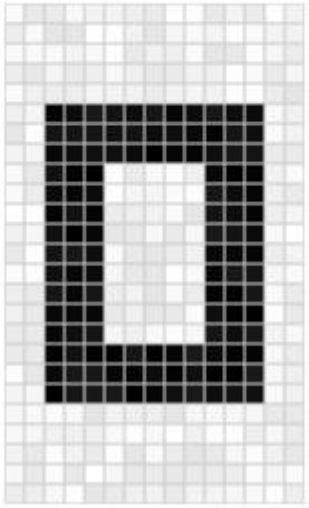


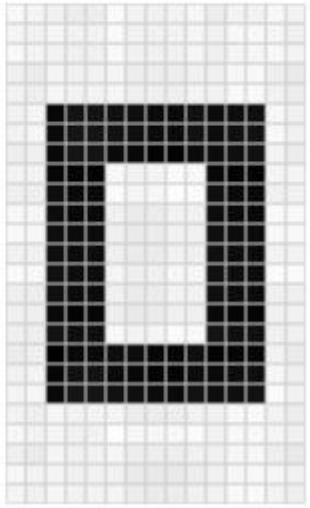
很明显，前面三个奇异值远远比后面的奇异值要大，这样矩阵 M 的分解方式就可以如下：

$$M \approx \mathbf{u}_1\sigma_1\,\mathbf{v}_1^T + \mathbf{u}_2\sigma_2\,\mathbf{v}_2^T + \mathbf{u}_3\sigma_3\,\mathbf{v}_3^T$$

经过奇异值分解后，我们得到了一张降噪后的图像。

Noisy image　　Improved image





$\sigma_1 = 14.15$

$\sigma_2 = 4.67$

$\sigma_3 = 3.00$

$\sigma_4 = 0.21$

$\sigma_5 = 0.19$

...

$\sigma_{15} = 0.05$

SVD应用—推荐算法

会员		电影							
		喜剧				恐怖			
偏好	ID	宿醉	东成西就	大话西游	八星报喜	午夜凶铃	咒怨	林中小屋	寂静岭
喜剧	至尊宝	4	4	5	5	2	3	2	3.75
	小小宝	5	5	5	4	2	2	3	1
	流氓兔	5	4	4	5	2	3	1	2
	霹*雳	5	4	5	5	3	2	1	2
	中原不败	4	5	5	4	2	1	3	2
恐怖	魂飞魄散	1	2	3	2	5	3.875	5	5
	荒村少年	3	1	2	2	4	5	4	4
	憨豆豆	2	1	3	2	4	5	4	5
	怪大叔	2	2	3	1	5	5	5	4
	美味僵尸	1	3	2	1	4	5	4	5

SVD——矩阵变换

矩阵$A(10\times8)$

4	4	5	5	2	3	2	3.75
5	5	5	4	2	2	3	1
5	4	4	5	2	3	1	2
5	4	5	5	3	2	1	2
4	5	5	4	2	1	3	2
1	2	3	2	5	3.875	5	5
3	1	2	2	4	5	4	4
2	1	3	2	4	5	4	5
2	2	3	1	5	5	5	4
1	3	2	1	4	5	4	5

矩阵$A^T(8\times10)$

4	5	5	5	4	1	3	2	2	1
4	5	4	4	5	2	1	1	2	3
5	5	4	5	5	3	2	3	3	2
5	4	5	5	4	2	2	2	1	1
2	2	2	3	2	5	4	4	5	4
3	2	3	2	1	3.875	5	5	5	5
2	3	1	1	3	5	4	4	5	4
3.75	1	2	2	2	5	4	5	4	5

$$A^TA=\left(A^TA\right)=$$

8×8矩阵

126	115	133	121	90	95	84	88
115	117	129	113	88	90	86	88
133	129	151	131	111	114	107	112
121	113	131	121	86	90	79	88
90	88	111	86	123	128	119	125
95	90	114	90	128	142	124	135
84	86	107	79	119	124	122	122
88	88	112	88	125	135	122	134

SVD——求奇异值

矩阵$\left(A^T A\right)$的非零特征值为{879.7,130.79,12.17,6.4,3.9},对应矩阵A的非零奇异值为{σ_1=29.7, σ_2=11.4, σ_3=3.5, σ_4=2.5},对应右奇异向量为：

$$v_1 = \begin{pmatrix} 0.34 \\ 0.33 \\ 0.40 \\ 0.33 \\ 0.35 \\ 0.37 \\ 0.34 \\ 0.36 \end{pmatrix}, \quad v_2 = \begin{pmatrix} 0.39 \\ 0.34 \\ 0.28 \\ 0.40 \\ -0.31 \\ -0.36 \\ -0.34 \\ -0.36 \end{pmatrix}$$

由于奇异值（特征的权重）下降的速度非常快，表明矩阵的信息量集中分布在前几个较大的特征值中，本例中提取前2个特征。

SVD——右奇异向量解析

影片类型	片名	特征1（29.7）	特征2（11.4）	得分均值
喜剧	宿醉	0.34	0.39	3.20
	东成西就	0.33	0.34	3.10
	大话西游	0.40	0.29	3.70
	八星报喜	0.33	0.40	3.10
恐怖	午夜凶铃	0.35	-0.31	3.30
	咒怨	0.37	-0.37	3.49
	林中小屋	0.34	-0.34	3.20
	寂静岭	0.36	-0.37	3.38

可以看作电影的本身的**精彩程度**的特征

可以看做有关电影**影片类型**的特征

SVD——左奇异向量解析

偏好	ID	特征1（29.7）	特征2（11.4）	打分平均值
喜剧	至尊宝	0.34	0.23	3.59
	小小宝	0.32	0.34	3.38
	流氓兔	0.31	0.32	3.25
	霹*雳	0.32	0.35	3.38
	中原不败	0.31	0.31	3.25
恐怖	魂飞魄散	0.32	-0.33	3.36
	荒村少年	0.30	-0.27	3.13
	憨豆豆	0.31	-0.31	3.25
	怪大叔	0.32	-0.34	3.38
	美味僵尸	0.30	-0.34	3.13

可以看做是会员的**打分习惯**特征

可看做是会员对影片**类型偏好**的特征

SVD——模型打分（1）

至尊宝	0.34	0.23
小小宝	0.32	0.34
流氓兔	0.31	0.32
霹*雳	0.32	0.35
中原不败	0.31	0.31
魂飞魄散	0.32	-0.33
荒村少年	0.30	-0.27
憨豆豆	0.31	-0.31
怪大叔	0.32	-0.34
美味僵尸	0.30	-0.34

$$\cdot \begin{pmatrix} 29.7 \\ 11.4 \end{pmatrix} \cdot$$

宿醉	东成西就	大话西游	八星报喜	午夜凶铃	咒怨	林中小屋	寂静岭
0.34	0.33	0.40	0.33	0.35	0.37	0.34	0.36
0.39	0.34	0.29	0.40	-0.31	-0.37	-0.34	-0.37

$\approx A$

各部分得分权重

$$29.7 \times (0.34 \times 0.36) + 11.4 \times (0.23 \times -0.37) = 2.6$$

计算至尊宝对《寂静岭》的评分

影片相对客观分数

影片类型适应度得分得分

SVD——模型打分（2）

至尊宝	0.34	0.23
小小宝	0.32	0.34
流氓兔	0.31	0.32
霹*雳	0.32	0.35
中原不败	0.31	0.31
魂飞魄散	0.32	-0.33
荒村少年	0.30	-0.27
憨豆豆	0.31	-0.31
怪大叔	0.32	-0.34
美味僵尸	0.30	-0.34

$$\cdot \begin{pmatrix} 29.7 \\ & 11.4 \end{pmatrix} \cdot$$

宿醉	东成西就	大话西游	八星报喜	午夜凶铃	咒怨	林中小屋	寂静岭
0.34	0.33	0.40	0.33	0.35	0.37	0.34	0.36
0.39	0.34	0.29	0.40	-0.31	-0.37	-0.34	-0.37

$\approx A$

各部分得分权重

$$29.7 \times (0.32 \times 0.37) + 11.4 \times (-0.33 \times -0.37) = 4.9$$

计算魂飞魄散对《咒怨》的评分

影片相对客观分数

影片类型适应度得分得分

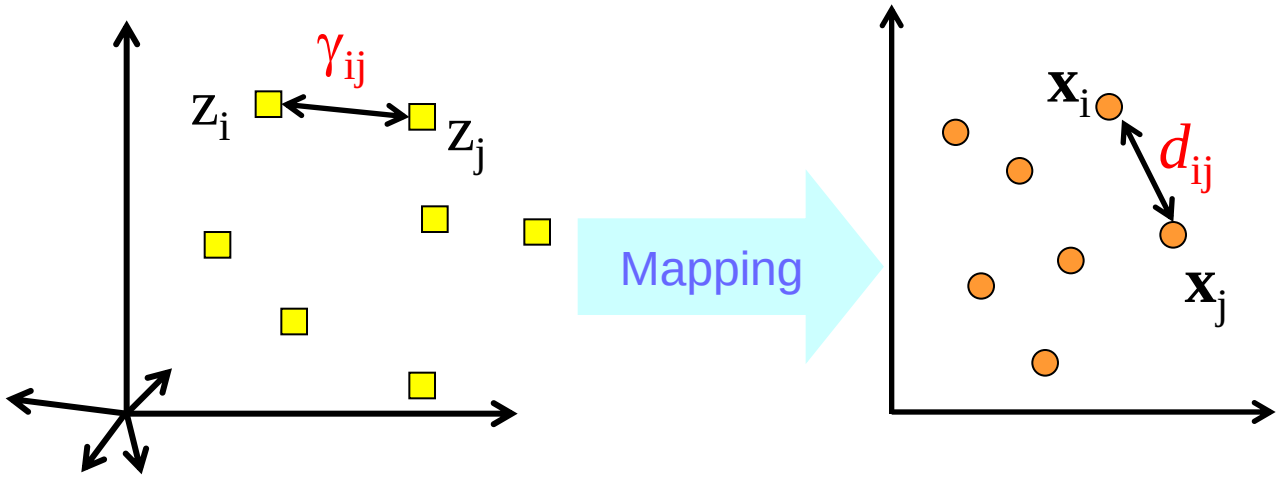
SVD结果简要测评

- 至尊宝的观影历史显示其对喜剧类的电影评分较高，对恐怖电影普遍评分较低，因此可以推测他应该是**不喜欢**看《寂静岭》的，模型给出的打分为**2.6**，与实际情况是相符的。
- 魂飞魄散的观影历史显示其对恐怖类的电影评分较高，对喜剧电影普遍评分较低，因此可以推测他应该是很**喜欢**看《咒怨》的，模型给出的打分为**4.9**，与实际情况是相符的。

偏好	ID	宿醉	东成西就	大话西游	八星报喜	午夜凶铃	咒怨	林中小屋	寂静岭
喜剧	至尊宝	4	4	5	5	2	3	2	2.6
恐怖	魂飞魄散	1	2	3	2	5	4.9	5	5

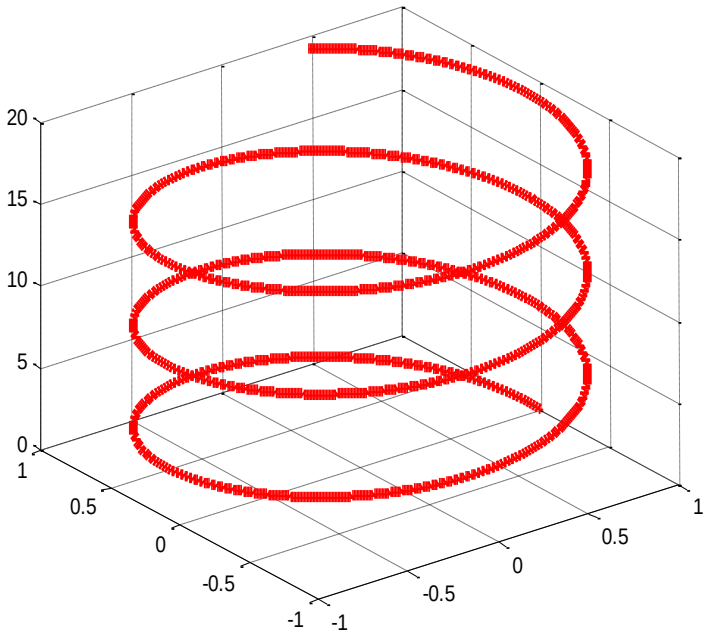
线性降维方法

- 线性降维
  - 主成分分析 (PCA) [Jolliffe, 1986]
  - 线性判别分析 (LDA) [Fukunaga, 1991]
  - 奇异值分解(SVD)
  - 多维尺度变换 (MDS) [Cox, 1994]



线性降维方法的不足

- 原始数据无法表示为特征的简单线性组合
  - 比如：PCA无法表达Helix曲线流形



1-D Helix曲线流形

线性降维方法的不足

- 真实数据中的有用信息不能由线性特征表示
  - 比如：如何获取并表示多姿态人脸的姿态信息

#1

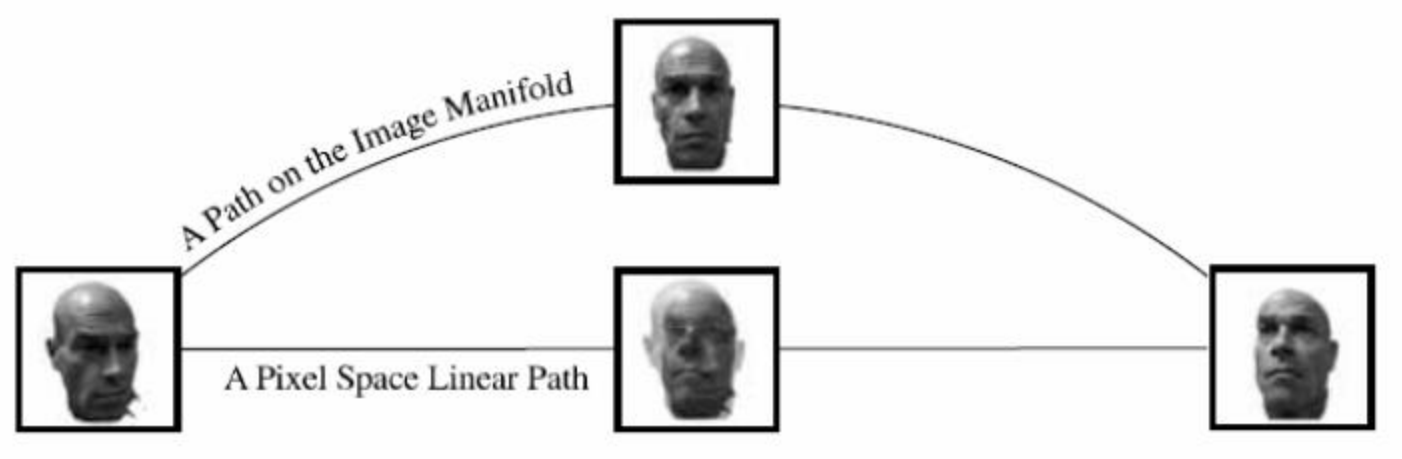
  - 比如：如何获取运动视频序列中某个动作的对应帧

#2

#1 引自J.B. Tenenbaum et al. 2000

#2 引自Jenkins et. al, IROS 2002

线性降维方法的不足



A line path and a path on Manifold

非线性降维方法

非线性降维方法

- 传统非线性降维
  - 核主成分分析 (KPCA) [Scholkopf, 1998]
  - 主曲线 (Principal Curves) [Hastie, 1989] [Tibshirani, 1992]
  - 自组织映射 (SOM) [Kohonen, 1995]
  - 产生式拓扑映射 (GTM) [Bishop, 1998]
  - …
- 流形学习
  - 保距特征映射 (ISOMAP) [Tenenbaum, 2000]
  - 局部线性嵌入 (LLE) [Roweis, 2000]
  - 拉普拉斯特征映射 (LE, Laplacian Eigenmap) [Belkin, 2001]
  - …

传统非线性降维

Kernel PCA (KPCA): Basic Idea

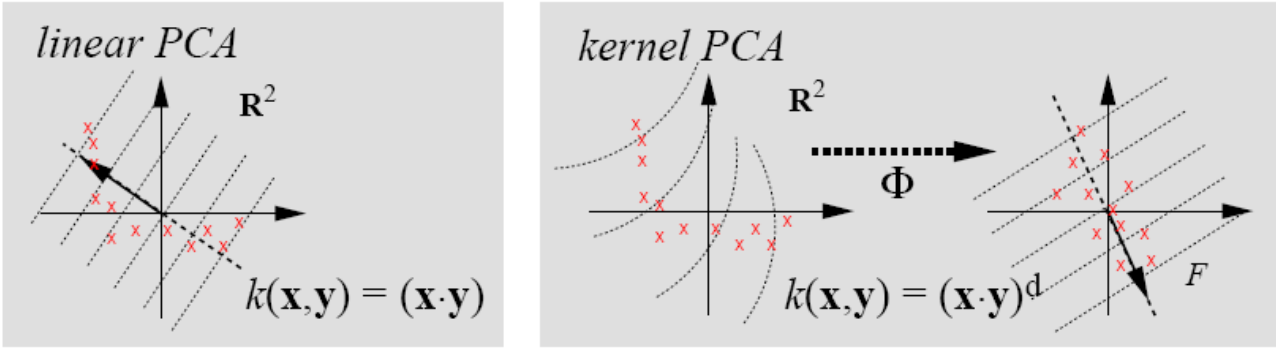


Fig. 1. Basic idea of kernel PCA: by using a nonlinear kernel function k instead of the standard dot product, we implicitly perform PCA in a possibly high–dimensional space F which is nonlinearly related to input space. The dotted lines are contour lines of constant feature value.

Kernel PCA Formulation

- We need the following fact:
- Let **v** be an eigenvector of the scatter matrix:

$$S = \sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^T$$

- Then **v** belongs to the linear space spanned by the data points $\mathbf{x}_i$ i=1, 2, …N.
- Proof:

$$S\mathbf{v} = \lambda\mathbf{v} \Rightarrow \mathbf{v} = \frac{1}{\lambda}\sum_{i=1}^{N} \mathbf{x}_i (\mathbf{x}_i^T \mathbf{v}) = \sum_{i=1}^{N} \alpha_i \mathbf{x}_i$$

Kernel PCA Formulation…

- Let C be the scatter matrix of the centered mapping $\phi(\mathbf{x})$:

$$C = \sum_{i=1}^{N} \phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^T$$

- Let $\mathbf{w}$ be an eigenvector of C, then $\mathbf{w}$ can be written as a linear combination:

$$\mathbf{w} = \sum_{k=1}^{N} \alpha_k \phi(\mathbf{x}_k)$$

- Also, we have: $C\mathbf{w} = \lambda\mathbf{w}$

- Combining, we get: $(\sum_{i=1}^{N} \phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^T)(\sum_{k=1}^{N} \alpha_k \phi(\mathbf{x}_k)) = \lambda \sum_{k=1}^{N} \alpha_k \phi(\mathbf{x}_k)$

Kernel PCA Formulation…

$$(\sum_{i=1}^{N}\phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^T)(\sum_{k=1}^{N}\alpha_k\phi(\mathbf{x}_k)) = \lambda\sum_{k=1}^{N}\alpha_k\phi(\mathbf{x}_k) \Rightarrow$$

$$\sum_{i=1}^{N}\sum_{k=1}^{N}\phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^T\phi(\mathbf{x}_k)\alpha_k = \lambda\sum_{k=1}^{N}\alpha_k\phi(\mathbf{x}_k) \Rightarrow$$

$$S\nu = \lambda\nu$$

$$\sum_{i=1}^{N}\sum_{k=1}^{N}\phi(\mathbf{x}_l)^T\phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^T\phi(\mathbf{x}_k)\alpha_k = \lambda\sum_{k=1}^{N}\alpha_k\phi(\mathbf{x}_l)^T\phi(\mathbf{x}_k),\ l = 1,2,\ldots,N \Rightarrow$$

$$K^2\boldsymbol{\alpha} = \lambda K\boldsymbol{\alpha} \Rightarrow$$

$$K\boldsymbol{\alpha} = \lambda\boldsymbol{\alpha},\ \text{where}\ K_{ij} = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j).$$

Kernel or Gram matrix

Kernel PCA Formulation…

From the eigen equation $K\boldsymbol{\alpha} = \lambda\boldsymbol{\alpha}$

And the fact that the eigenvector **w** is normalized to 1, we obtain:

$$\| \mathbf{w} \|^2 = (\sum_{i=1}^{N} \alpha_i \phi(\mathbf{x}_i))^T (\sum_{i=1}^{N} \alpha_i \phi(\mathbf{x}_i)) = \boldsymbol{\alpha}^T K \boldsymbol{\alpha} = 1 \Rightarrow$$

$$\boldsymbol{\alpha}^T \boldsymbol{\alpha} = \frac{1}{\lambda}$$

KPCA Algorithm

Step 1: Compute the Gram matrix: $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j),\ i, j = 1, \ldots, N$

Step 2: Compute (eigenvalue, eigenvector) pairs of *K:*

$$(\boldsymbol{\alpha}^l, \lambda_l), l = 1, \ldots, M$$

Step 3: Normalize the eigenvectors: $\boldsymbol{\alpha}^l \leftarrow \frac{\boldsymbol{\alpha}^l}{\lambda_l}$

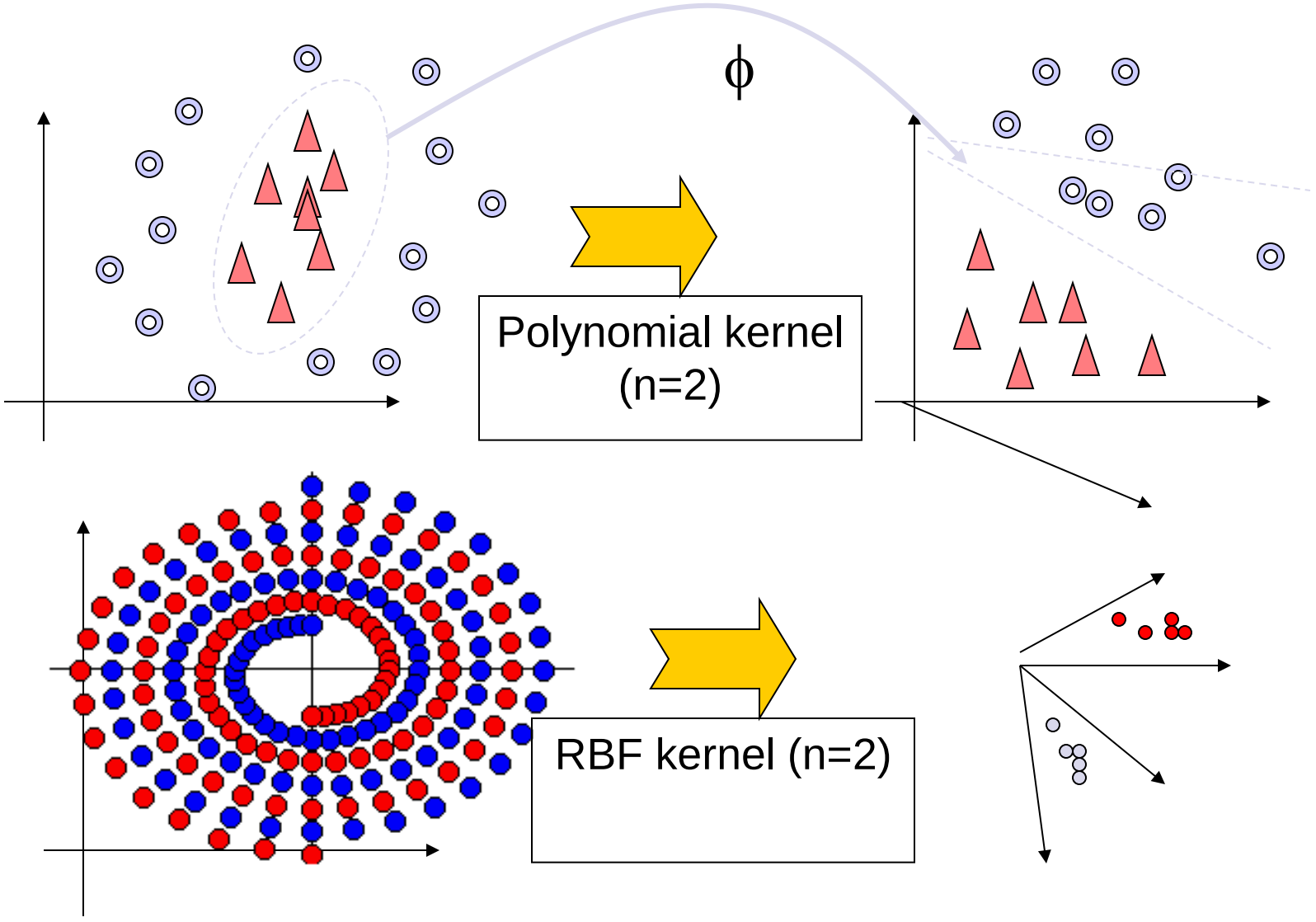
Thus, an eigenvector $\mathbf{w}^l$ of *C* is now represented as: $\mathbf{w}^l = \sum_{k=1}^{N} \alpha_k^l \phi(\mathbf{x}_k)$

To project a test feature $\phi(\mathbf{x})$ onto $\mathbf{w}^l$ we need to compute:

$$\phi(\mathbf{x})^T \mathbf{w}^l = \phi(\mathbf{x})^T \left(\sum_{k=1}^{N} \alpha_k^l \phi(\mathbf{x}_k)\right) = \sum_{k=1}^{N} \alpha_k^l k(\mathbf{x}_k, \mathbf{x})$$

So, we never need ϕ explicitly

Examples of Kernels



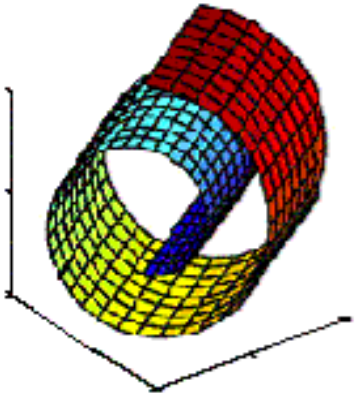
流形学习

流形学习

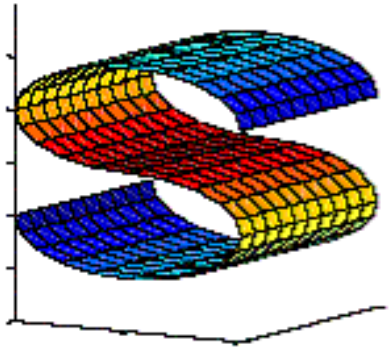
- 基于流形学习的非线性降维
  - 保距特征映射 (ISOMAP) [Tenenbaum, 2000]
  - 局部线性嵌入 (LLE) [Roweis, 2000]
  - 拉普拉斯特征映射 (LE, Laplacian Eigenmap) [Belkin, 2001]
  - Hessian LLE (HLLE) [Donoho, 2003]
  - 局部切空间对齐 (LTSA, Local Tangent Space Alignment) [Zhang, 2004]
  - 最大方差展开 (MVU/SDE, Maximum Variance Unfolding) [Weinberger, 2004]
  - 局部保持映射 (Locality Preserving Projections) [He, 2003]
  - …

流形学习框架

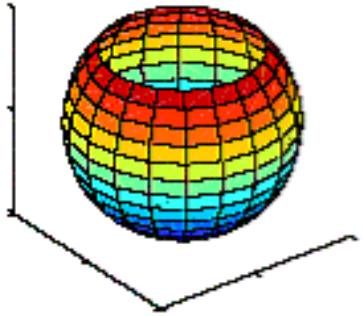
- 什么是流形？
  - 流形是线性子空间的一种非线性推广
  - 拓扑学角度：局部区域线性，与低维欧式空间拓扑同胚
  - 微分几何角度：有重叠chart的光滑过渡
  - 黎曼流形就是以光滑的方式在每一点的切空间上指定了欧氏内积的微分流形



Swiss-roll



S-curve



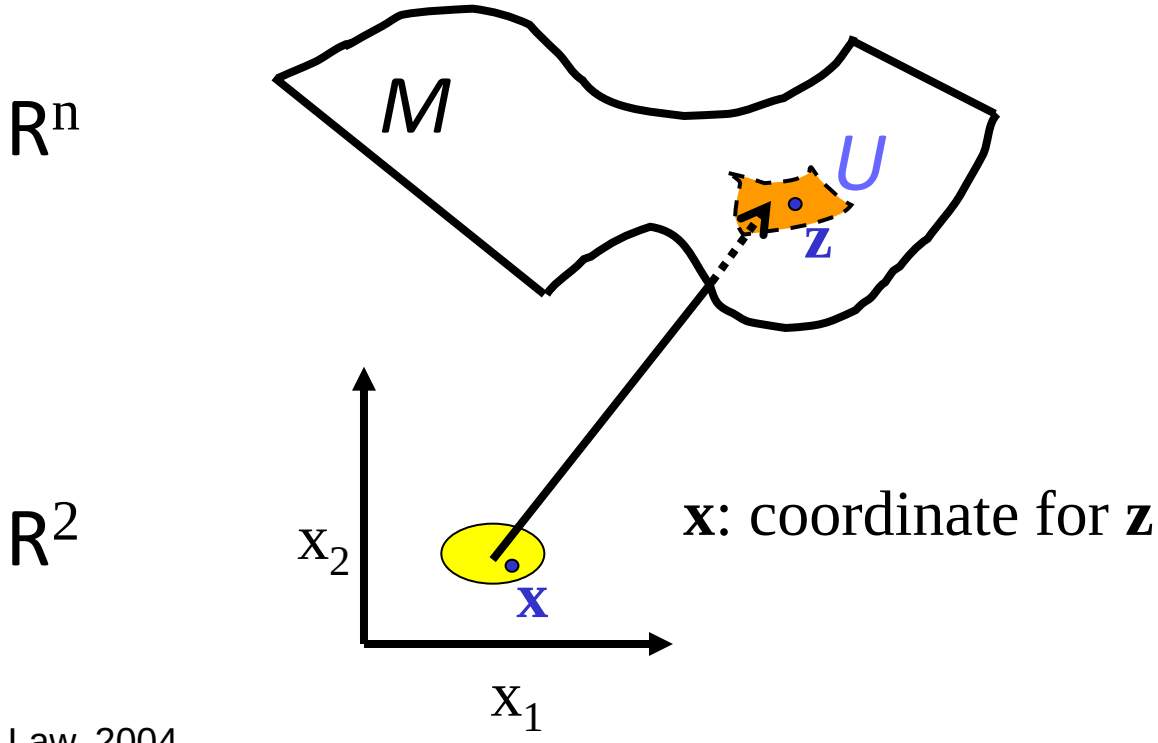
Fishbow

#1

#1 引自S.T. Roweis et al. 2000

流形学习框架

- 流形的数学定义
  - 设 M 是一个Hausdorff拓扑空间，若对每一点 $p \in M$ 都有 p 的一个开邻域 U 和 $\square^d$ 的一个开子集同胚，则称 M 为 d 维拓扑流形，简称为 d 维流形.



#1

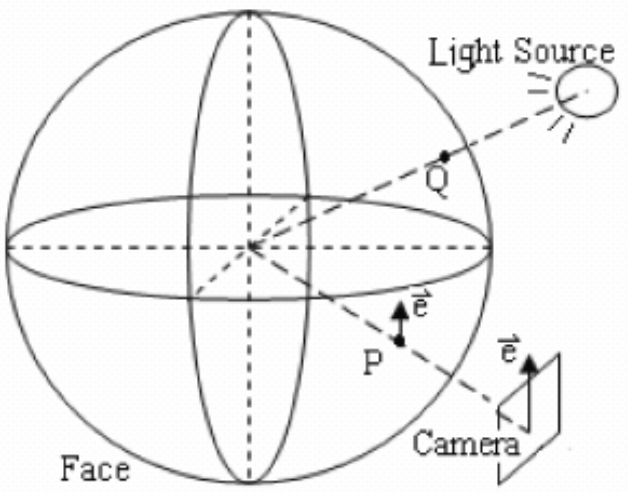
#1 引自M. H. Law, 2004

流形学习框架

- 一些基本数学概念
  - 拓扑，Hausdorff 空间，坐标卡，微分结构
  - 光滑函数，光滑映射，切向量，切空间
  - …
- 参考文献
  - 陈省身, 陈维桓, 微分几何讲义. 北京大学出版社, 1983
  - M Berger, B Gostiaux. Differential Geometry: Manifolds, Curves and Surfaces, GTM115. Springer-Verlag, 1974
  - 陈维桓, 微分流形初步(第二版). 高等教育出版社, 2001

流形学习框架

- 流形学习的目的
  - 流形学习是一种非线性的维数约简方法
  - 高维观察数据的变化模式本质是由少数几个隐含变量所决定的
    - 如：人脸采样由光线亮度、人与相机的距离、人的头部姿势、人的面部表情等因素决定
  - 从认知心理学的 ，的认知过程是基于认知流



#1

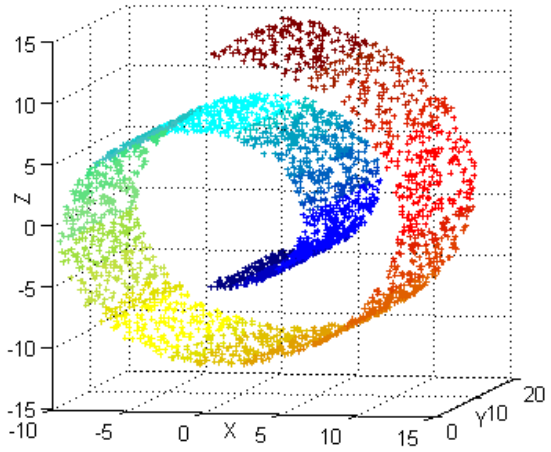
#1 引自Lin *et al*. PAMI 2008

流形学习的数学定义

设 $Y \subset R^d$是一个低维流形，$f: Y \to R^D$是一个光滑嵌入，其中 $D>d$. 数据集 $\{y_i\}$是随机生成的，且经过 f 映射为观察空间的数据 $\{x_i = f(y_i)\}$. 流形学习就是在给定观察样本集$\{x_i\}$的条件下重构 f 和 $\{y_i\}$.

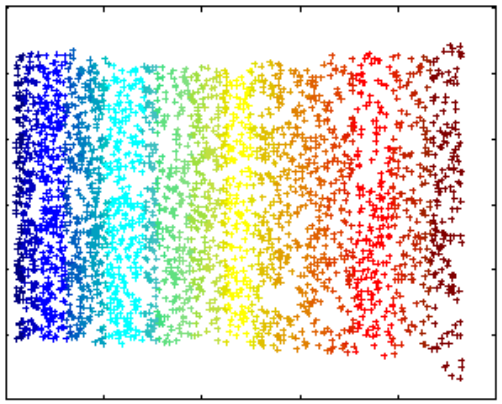
V. de Silva and J. B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction . ***Neural Information Processing Systems 15 (NIPS'2002)*****, pp. 705-712, 2003.**

流形学习示例



非线性降维

保持一定几何拓扑关系，如测地距离/邻域线性重构关系



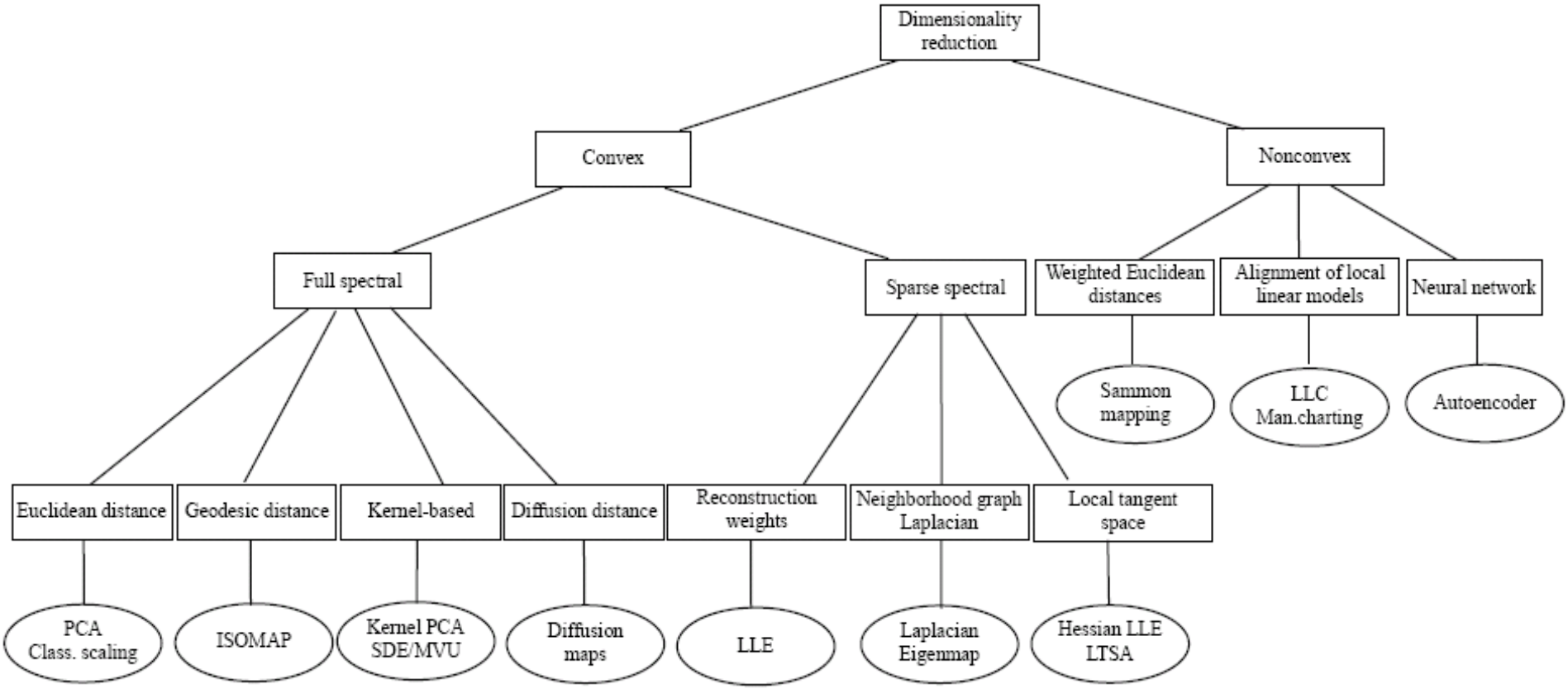
高维数据空间
data / observation space

低维嵌入空间
embedding / coordinate space

经典流形学习方法一览

方法简称	所保持的几何属性	全局/局部关系	计算复杂度
ISOMAP	点对测地距离	全局	非常高
LLE	局部线性重构关系	局部	低
LE	局部邻域相似度	局部	低
HLLE	局部等距性	局部	高
LTSA	局部坐标表示	全局+局部	低
MVU	局部距离	全局+局部	非常高
Logmap	测地距离与方向	局部	非常低
Diffusion Maps	diffusion距离	全局	中等

经典方法分类结构图


Dimensionality reduction
Convex
Nonconvex
Full spectral
Sparse spectral
Weighted Euclidean distances
Alignment of local linear models
Neural network
Sammon mapping
LLC Man.charting
Autoencoder
Euclidean distance
Geodesic distance
Kernel-based
Diffusion distance
Reconstruction weights
Neighborhood graph Laplacian
Local tangent space
PCA Class. scaling
ISOMAP
Kernel PCA SDE/MVU
Diffusion maps
LLE
Laplacian Eigenmap
Hessian LLE LTSA

重点介绍的几个方法

- 等距映射(**ISOMAP**)

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.

- 局部线性嵌入(**LLE**)

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, vol. 290, pp. 2323--2326, 2000.

- 拉普拉斯特征映射(**Laplacian Eigenmap**)

M. Belkin, P. Niyogi, Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, Vol. 15, Issue 6, pp. 1373 –1396, 2003 .

重点介绍的几个方法

- **等距映射(ISOMAP)**

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.

- **局部线性嵌入(LLE)**

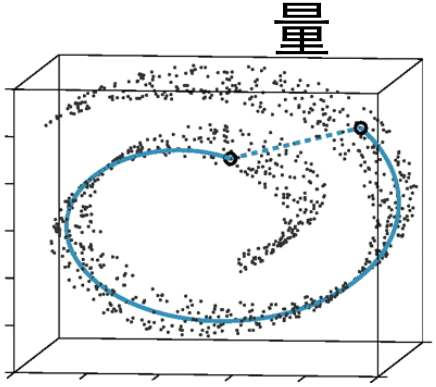
S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, vol. 290, pp. 2323--2326, 2000.

- **拉普拉斯特征映射(Laplacian Eigenmap)**

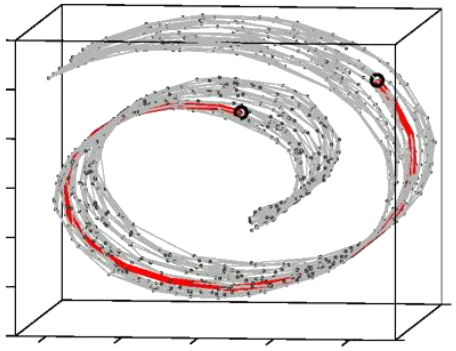
M. Belkin, P. Niyogi, Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, Vol. 15, Issue 6, pp. 1373 –1396, 2003 .

代表性算法-1

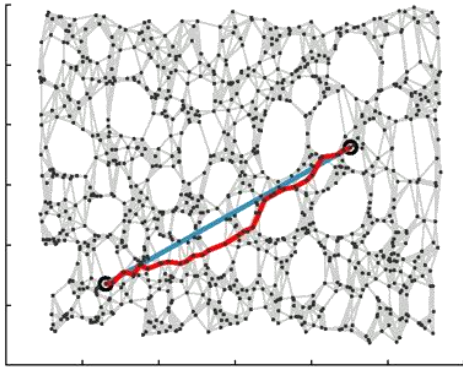
- ISOMAP (Isometric feature mapping)
  - 保持全局测地距离
    - 测地距离反映数据在流形上的真实距离差异
  - 等距映射
    - 基于线性算法MDS，采用“测地距离”作为数据差异度量



欧式距离 vs.
测地距离



最短路径近
似测地距离



降维嵌入空间

#1

#1 引自J.B. Tenenbaum et al. 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高维空间中的距离相同.**
- **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的准则函数

$$J_{ee} = \frac{\sum_{i<j} (d_{ij} - \delta_{ij})^2}{\sum_{i<j} \delta_{ij}^{\ 2}},$$

$$J_{ff} = \sum_{i<j} \left(\frac{d_{ij} - \delta_{ij}}{\delta_{ij}} \right)^2$$

$$J_{ef} = \frac{1}{\sum_{i<j} \delta_{ij}} \sum_{i<j} \frac{(d_{ij} - \delta_{ij})^2}{\delta_{ij}}$$

CMDS的数学模型[22]：

假设 $D=[d_{ij}]_{n\times n}$ 是 $n\times n$ 维的距离矩阵，在 d 维空间 R^d 中求 n 个点 $x_1,x_2,...x_n$，使得 n 个点的距离与矩阵 D 中的距离在某种意义下尽量接近。设求得的 n 个点为 $x_1,x_2,...x_n$，表示为：

$$X=(x_1,x_2,...x_n)^T$$

则称 X 为距离矩阵 D 的一个低维嵌入，由这 n 个点之间的距离构成的距离阵称为 D 的拟合距离阵 $\hat{D}$。为证明方便，先引入中心矩阵的概念，其定义如下：

设中心矩阵 $H=I_n-\frac{1}{n}J_n$，其中 I_n 是一个 $n\times n$ 维的单位矩阵，J_n 是一个所有元素均为 1 的 $n\times n$ 维矩阵，于是

$$H=\begin{bmatrix} 1-\frac{1}{n} & -\frac{1}{n} & \cdots & -\frac{1}{n} \\ -\frac{1}{n} & 1-\frac{1}{n} & \cdots & -\frac{1}{n} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{n} & -\frac{1}{n} & \cdots & 1-\frac{1}{n} \end{bmatrix}_{n\times n}$$

并引入如下两个简记符号：

$A=[a_{ij}]_{n\times n}$，其中 $a_{ij}=-\frac{1}{2}d_{ij}$

$B=HAH$

则 B 中的一个元素由下式给出：

$$b_{ij}=a_{ij}-\frac{1}{n}\sum_{k=1}^{n}a_{ik}-\frac{1}{n}\sum_{k=1}^{n}a_{kj}+\frac{1}{n^2}\sum_{k=1}^{n}\sum_{l=1}^{n}a_{kl}$$

定理：假设矩阵 B 的 p 个非零特征值 $\lambda_1\geq\lambda_2\geq\ldots\geq\lambda_p>0(\lambda_{p+1}=\ldots\lambda_n=0)$ 对应的特征向量为 $X_1,X_2,\ldots,X_p$，即有

$$BX_i=\lambda_i X_i \qquad (2\text{-}1)$$

若 $X_i(i=1,2,\ldots,p)$ 满足，$X_i^T X_i=\lambda_i$，令 $X^T=(X_1,X_2,\ldots,X_p)$，则 X^T 是原始距离矩阵 D 的一个低维嵌入。

证明：

已知 $BX^T-X^T\Lambda=0$ 以及 $XX^T=\Lambda$（Λ 为由 B 的 p 个非零特征值构成的对角矩阵），

令

$$M=\begin{pmatrix}\Lambda & O_{p,n-p}\\ O_{n-p,p} & O_{n-p}\end{pmatrix},\quad N=\begin{pmatrix}\Lambda & O_{p,n-p}\\ O_{n-p,p} & I_{n-p}\end{pmatrix}$$

以及

$$Y^T=(X_1,X_2,\ldots,X_p,Y_{p+1},\ldots,Y_n)$$

其中，$Y'=(Y_{p+1},\ldots,Y_n)$ 中列向量标准正交，且正交于 $X_i(i=1,2,\ldots,p)$，于是有 $Y'^TY'=1$，也就是说，Y' 是对应于 B 的 $n-p$ 个零特征值的特征向量。

于是（2-1）式转换为：

$$BY^T-Y^TM=0 \qquad (2\text{-}2)$$

令

$$\Gamma=Y^TN^{-\frac{1}{2}}$$

其中

$$N^{-\frac{1}{2}}=\begin{pmatrix}\lambda_1^{-\frac{1}{2}} & & & & & \\ & \ddots & & & & \\ & & \lambda_p^{-\frac{1}{2}} & & & \\ & & & 1 & & \\ & & & & \ddots & \\ & & & & & 1\end{pmatrix}_{n\times n}$$

则

$$\Gamma\Gamma^T = Y^T N^{-\frac{1}{2}} \cdot N^{-\frac{1}{2}} Y = I_n, \quad \Gamma^T\Gamma = N^{-\frac{1}{2}} Y \cdot Y^T N^{-\frac{1}{2}} = I_n$$

（2-2）式两边同乘以 $N^{-\frac{1}{2}}$，即可得

$$B\Gamma - Y^T M N^{-\frac{1}{2}} = 0$$

对上式进行数学变换，可以得到：

$$B = Y^T N^{-\frac{1}{2}} M N^{-\frac{1}{2}} Y = Y^T M N^{-1} Y = Y^T \begin{pmatrix} I_p & \\ & O_{n-p} \end{pmatrix}_{n\times n} Y$$

$$= \left(X_1, X_2, \ldots, X_p, 0, \ldots, 0\right)_{n\times n} \begin{pmatrix} X_1^T \\ \vdots \\ X_p^T \\ 0^T \\ \vdots \\ 0^T \end{pmatrix}_{n\times n}$$

$$= \left(X_1, X_2, \ldots, X_p\right)_{n\times p} \begin{pmatrix} X_1^T \\ \vdots \\ X_p^T \end{pmatrix}_{p\times n}$$

$$= X^T X$$

其中 $X^T = (X_1, X_2, \ldots, X_p)$

在此认为 X^T 是嵌入空间坐标矩阵， X^T 是一个 $1\times p$ 的向量，即

$$X^T=\begin{pmatrix}x_1\\ \vdots\\ x_n\end{pmatrix}$$

下面证明 X^T 中的 n 个点构造的距离矩阵与原始距离矩阵 D 相同。以 $d_X(i,j)$ 表示嵌入空间中第 i 个点和第 j 个点之间的距离，有

$$\begin{aligned}d_X(i,j)&=(x_i-x_j)^T(x_i-x_j)\\&=x_i^Tx_i-2x_i^Tx_j+x_j^Tx_j\\&=b_{ii}-2b_{ij}+b_{jj}\\&=a_{ii}-2a_{ij}+a_{jj}\\&=-2a_{ij}\\&=d_{ij}\end{aligned}$$

用 $d(x_i, x_j)$ 表示样本点 x_i 与 x_j 的距离，即

$$d(x_i, x_j)^2 = \|x_i - x_j\|^2 = x_i^T x_i - 2x_i^T x_j + x_j^T x_j.$$

记 N 维向量 ψ 为 $\psi = [x_1^T x_1, \ldots, x_N^T x_N]^T$，

则距离矩阵 $D = (d^2(x_i, x_j))_{i,j=1}^N$ 能重新写成

$$D = \psi \mathbf{1}_N^T - 2X^T X + \mathbf{1}_N \psi^T.$$

不失一般性，假设样本点被中心化，即 $\sum_{i=1}^N x_i = 0$

则有

$$H \equiv -(I - \mathbf{1}_N \mathbf{1}_N^T/N) D (I - \mathbf{1}_N \mathbf{1}_N^T/N)/2 = X^T X.$$

记 H 的特征值分解为

$$H = U \operatorname{diag}(\lambda_1, \ldots, \lambda_N) U^T,$$

其中 $U \in R^{N \times m}$ 为正交阵以及特征值 $\lambda_1 \geq \cdots \geq \lambda_N$ 降序排列，则 $T = \operatorname{diag}(\lambda_1^{1/2}, \ldots, \lambda_d^{1/2}) U_d$

其中 U_d 为最大的 d 个特征值所对应的特征向量所构成的矩阵。

假设数据集为 $X=\{x_1,x_2,...,x_n\}$，其中 $x_i \in R^D$，并且我们有任意两数据点之间相似程度组成的矩阵 $D=\{d_{ij}\}$，数据点相应得低维坐标为 $Y=\{y_1,y_2,...,y_n\}$，其中 $y_i \in R^d$

STEP 1.计算 $S=\{d_{ij}^2\}$；

STEP 2.取矩阵 $H=\{h_{ij}\}$，满足 $h_{ij}=\delta_{ij}-\frac{1}{n}$，其中 $\delta_{ij}=\begin{cases}1 & i=j\\ 0 & i\neq j\end{cases}$；

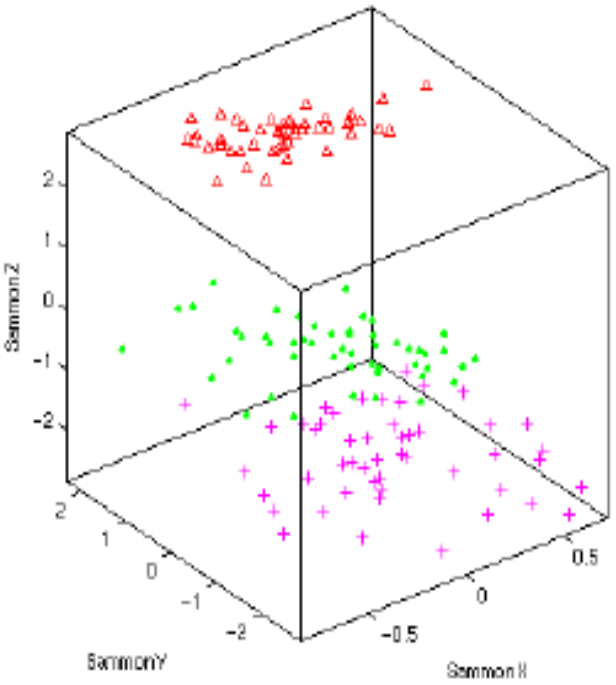
STEP 3.（双中心化）计算 $\tau(D)=-\frac{HSH}{2}$；

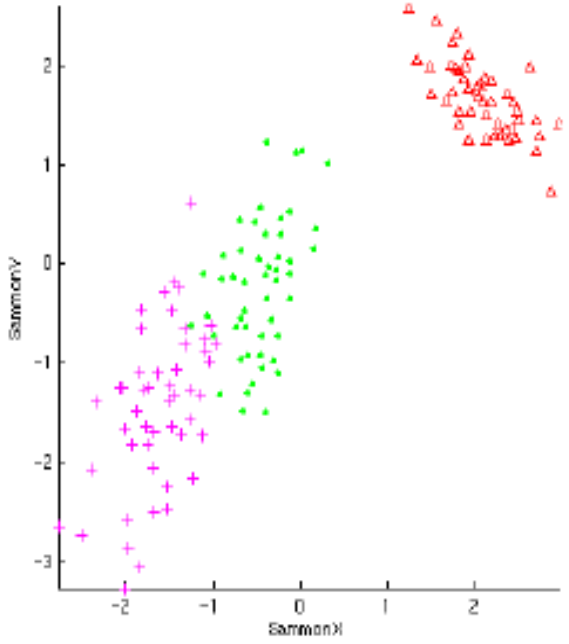
STEP 4.求矩阵 $\tau(D)$ 的 d 个最大的特征值和其对应得特征向量，矩阵 Λ_d 是对角阵，对角元素是从大到小排列的特征值，矩阵 U_d 的列为相应的特征向量。

STEP 5.计算 Y，对 Λ_d 对角线元素依次取算术平方根，得 $\sqrt{\Lambda_d}$，那么 $Y=U_d\sqrt{\Lambda_d}$。

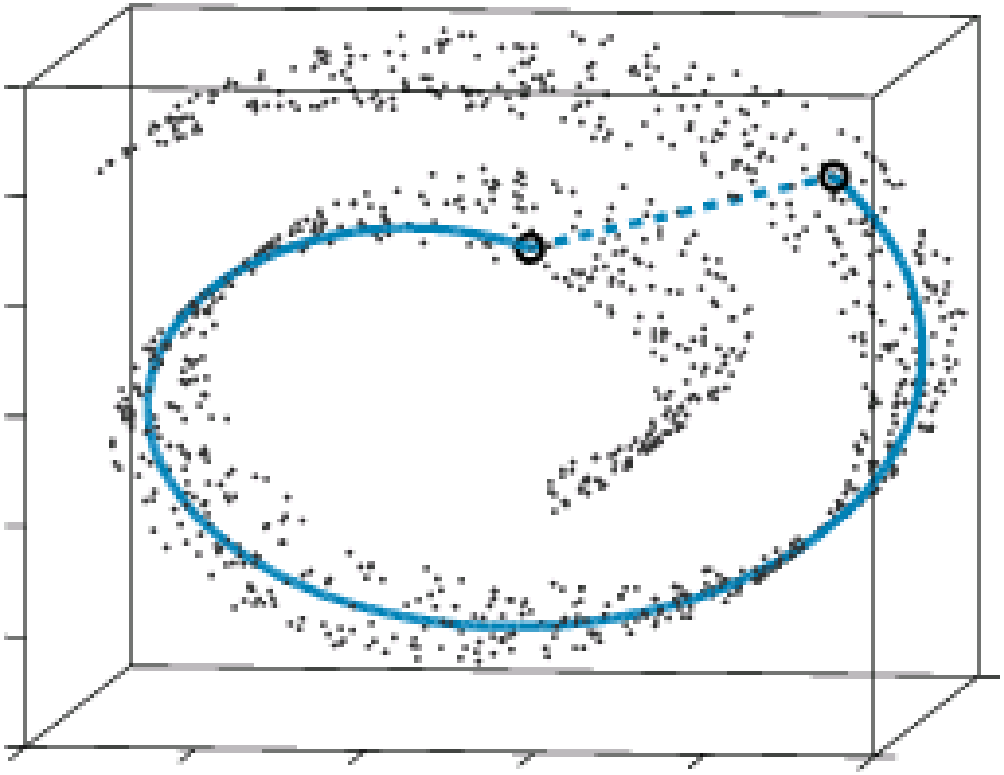
通常我们取相似程度为数据点之间的欧式距离，即 $d_{ij}=\left\|x_i-x_j\right\|_2$，这时 MDS 和 PCA 是等价的。

MDS的示意图





MDS的失效



测地距离

- 测地线：流形上连接两个点的最短曲线
  - 例如：球面上的测地线就是球面上的大圆弧
- 测地距离：测地线的长度

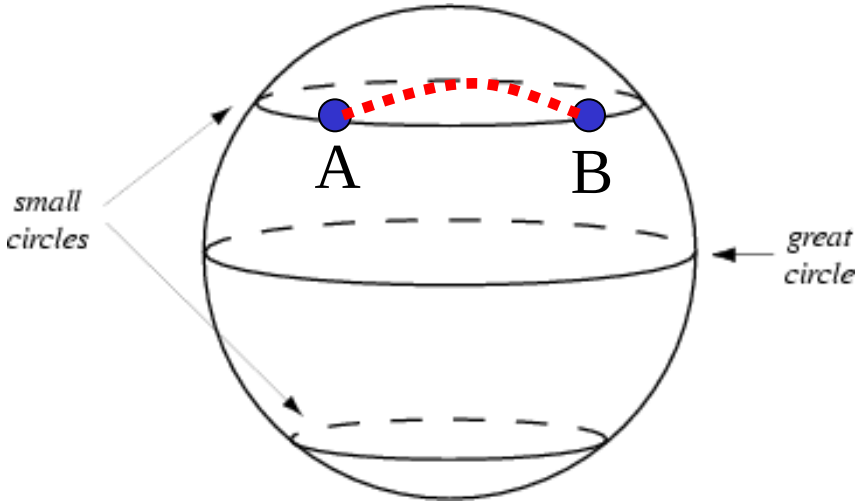


Figure from
http://mathworld.wolfram.com/GreatCircle.html

ISOMAP算法流程

1 计算每个点的近邻点 **(**用**K**近邻或 ε 邻域**).**

2 在样本集上定义一个赋权无向图 如果 X_i 和 X_j 互为近邻点**,** 则边的权值为 $d_{\mathrm{X}}(i, j)$.

3 计算图中两点间的最短距离**,** 记所得的距离矩阵为

$D_G = \{d_G(i, j)\}$ **.**

4 用**MDS**求低维嵌入坐标 **,**

令 $S = (S_{ij}) = (D_{ij}^2),\ \ H = (H_{ij}) = (\delta_{ij} - 1/N),\ \ \tau(D) = -HSH/2,$

低维嵌入是 $\tau(D)$ 的第**1**大到第 **d**大的特征值所对应的特征向量**.**

图距离逼近测地距离

M. Bernstein, V. Silva, J.C. Langford, J.B. Tenenbaum 证明了如下的渐进收敛定理.

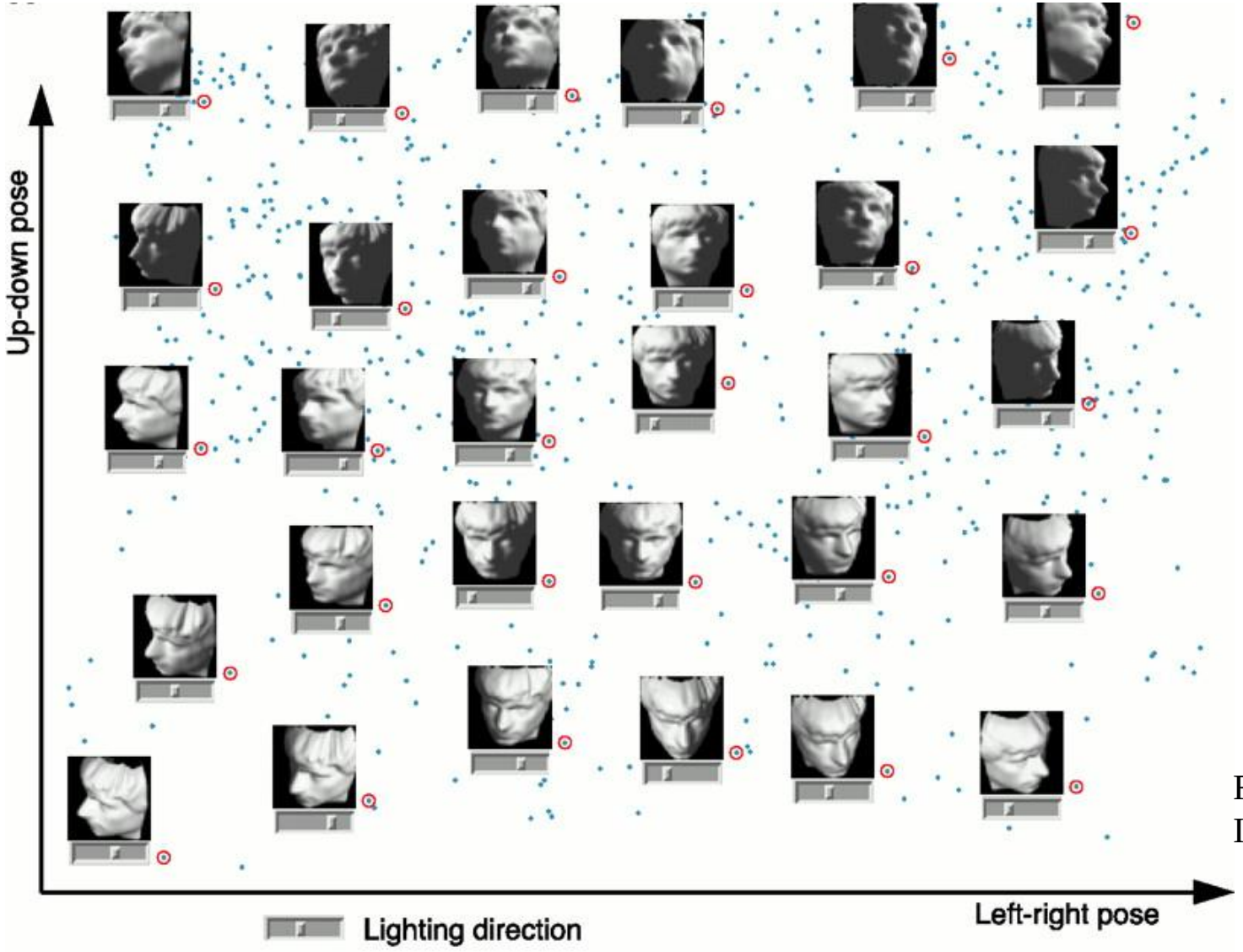
假设采样点是随机均匀抽取的, 则

渐进收敛定理 给定 $\lambda_1, \lambda_2, \mu > 0,$ 则只要样本集充分大且适当选择**K** , 不等式

$$1 - \lambda_1 \leq \frac{\text{graph distance}}{\text{geodesic distance}} \leq 1 + \lambda_2$$

至少以概率 $1 - \mu$ 成立.

ISOMAP实验结果



Figures from ISOMAP paper

ISOMAP实验结果

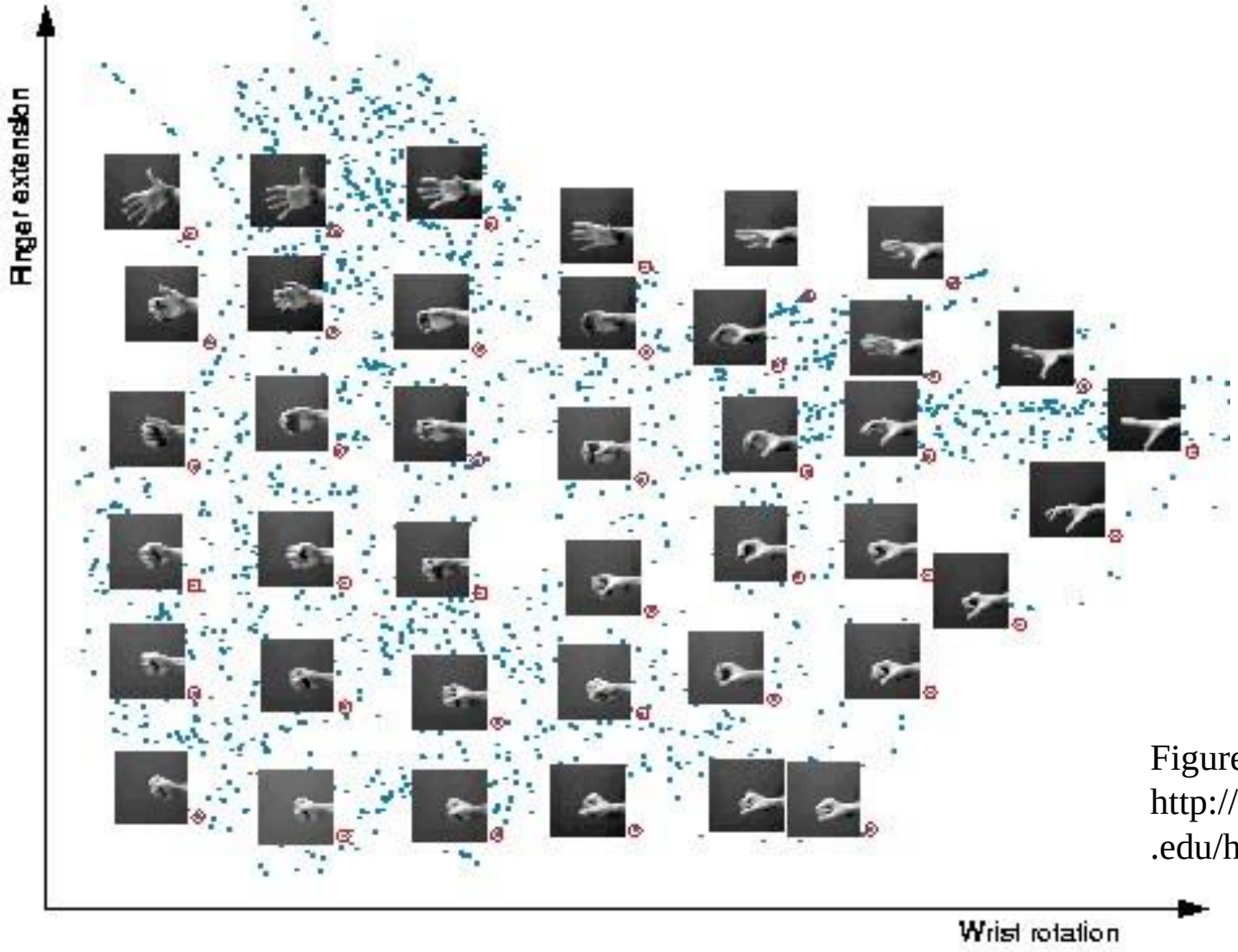
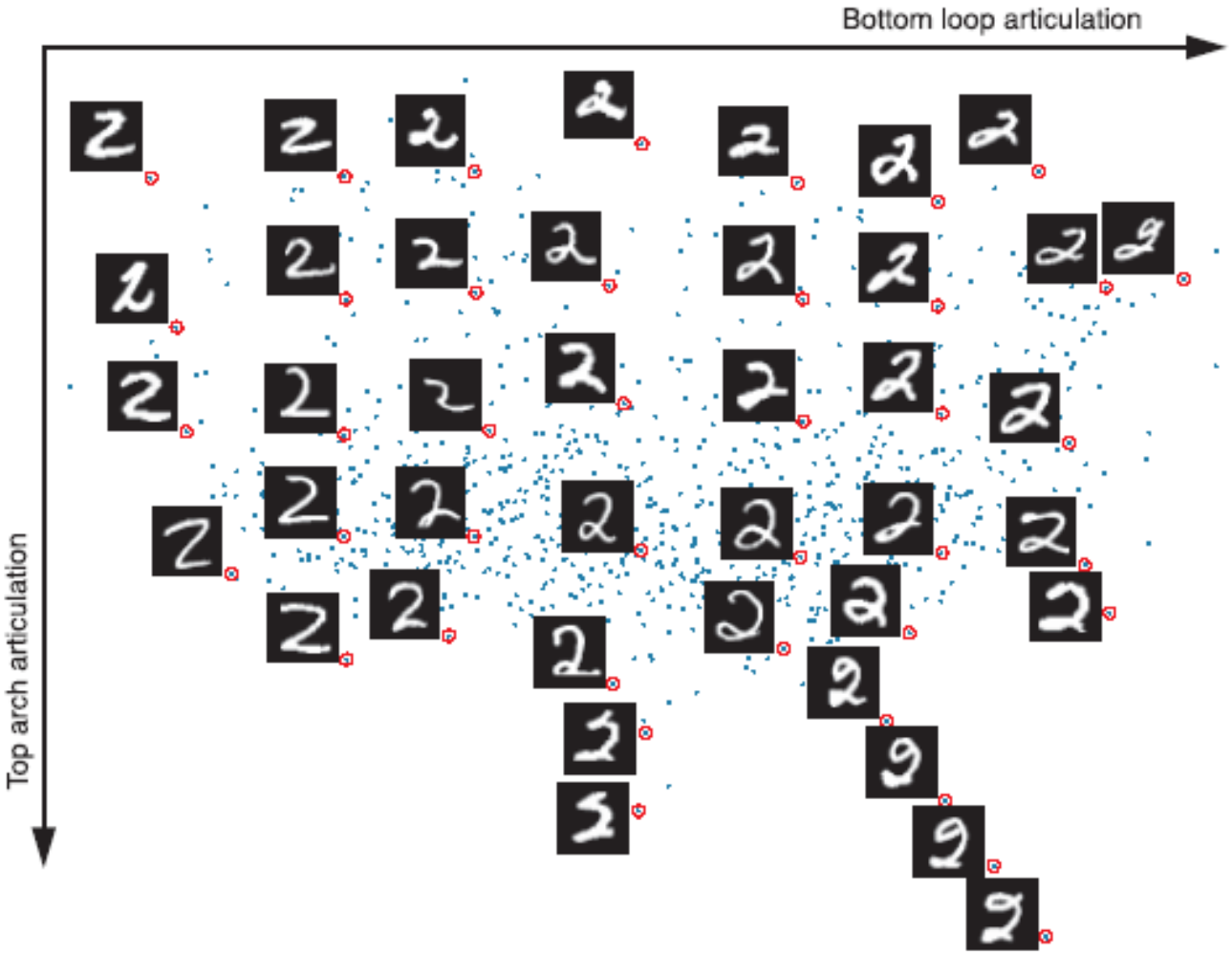


Figure from
http://isomap.stanford
.edu/handfig.html

ISOMAP实验结果



Figures from
ISOMAP paper

Interpolation on Straight Lines in the Projected Co-ordinates

Figures from ISOMAP paper

代表性算法-1

- ISOMAP (Isometric feature mapping)
  - 前提假设
    - 数据所在的低维流形与欧式空间的一个子集整体等距
    - 该欧式空间的子集是一个凸集
  - 思想核心
    - 较近点对之间的测地距离用欧式距离代替
    - 较远点对之间的测地距离用最短路径来逼近
  - 算法特点
    - 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
    - 计算点对间的最短路径比较耗时

ISOMAP - summary

- Inherits features of MDS and PCA:
  - guaranteed asymptotic convergence to true structure
  - Polynomial runtime
  - Non-iterative
  - Ability to discover manifolds of arbitrary dimensionality
- Perform well when data is from a single well sampled cluster
- Few free parameters
- Good theoretical base for its metrics preserving properties

Problems with ISOMAP

- Embeddings are biased to preserve the separation of faraway points, which can lead to distortion of local geometry
- Fails to nicely project data spread among multiple clusters
- Well-conditioned algorithm but computationally expensive for large datasets

Improvements to ISOMAP

- Conformal Isomap – capable of learning the structure of certain curved manifolds
- Landmark Isomap – approximates large global computations by a much smaller set of calculation
- Reconstruct distances using k/2 closest objects, as well as k/2 farthest objects

重点介绍的几个方法

- 等距映射(ISOMAP)

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.

- 局部线性嵌入(LLE)

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, vol. 290, pp. 2323--2326, 2000.

- 拉普拉斯特征映射(Laplacian Eigenmap)

M. Belkin, P. Niyogi, Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, Vol. 15, Issue 6, pp. 1373 –1396, 2003 .

代表性算法-2

- LLE (Locally linear embedding)
  - 显式利用“局部线性”的假设
  - 保持局部邻域几何结构 – 重构权重
  - 权重对样本集的几何变换具有不变性

代表性算法-2

- LLE (Locally linear embedding)
  - 前提假设
    - 采样数据所在的低维流形在局部是线性的
    - 每个采样点均可以利用其近邻样本进行线性重构表示
  - 学习目标
    - 低维空间中保持每个邻域中的重构权值不变
    - 在嵌入映射为局部线性的条件下，最小化重构误差
    - 最终形式化为特征值分解问题

LLE算法示意图

1. Compute the neighbors of each data point, $\vec{X}_i$.
2. Compute the weights W_{ij} that best reconstruct each data point $\vec{X}_i$ from its neighbors, minimizing the cost in Equation (1) by constrained linear fits.
3. Compute the vectors $\vec{Y}_i$ best reconstructed by the weights W_{ij}, minimizing the quadratic form in Equation (2) by its bottom nonzero eigenvectors.

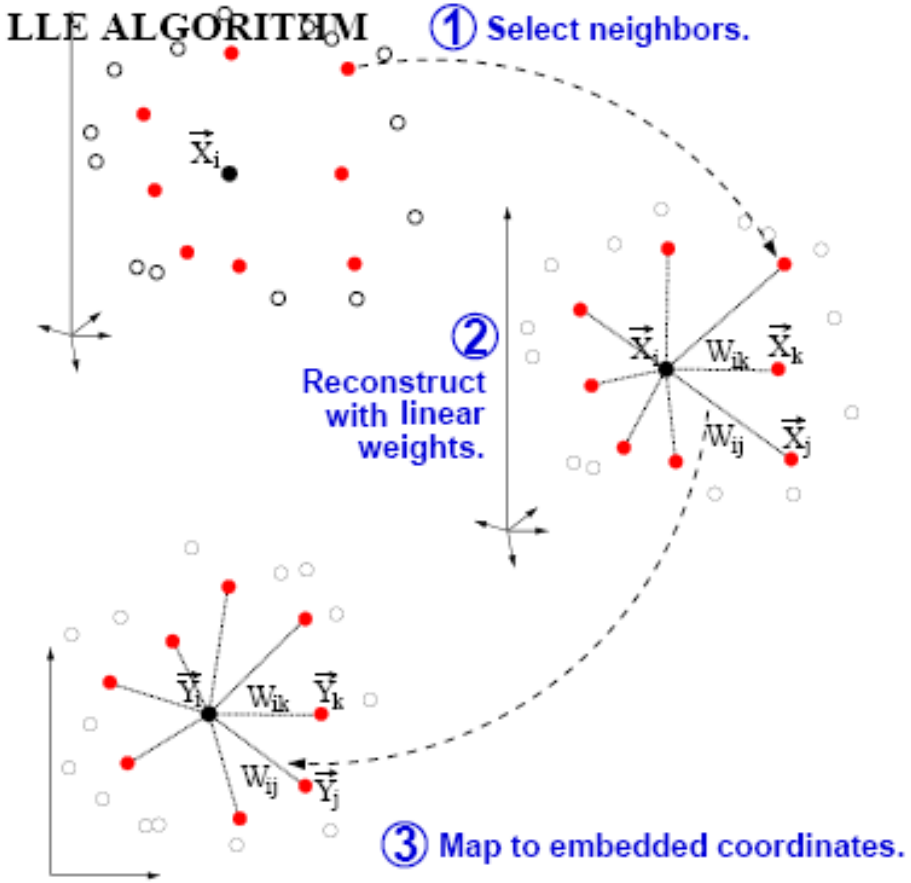


Figure 2: Summary of the LLE algorithm, mapping high dimensional inputs $\vec{X}_i$ to low dimensional outputs $\vec{Y}_i$ via local linear reconstruction weights W_{ij}.

LLE算法流程

1 计算每一个点 X_i 的近邻点, 一般采用 K 近邻或者 ε 邻域.

2 计算权值 W_{ij}, 使得把 X_i 用它的 K 个近邻点线性表示的误差最小, 即通过最小化 $\left\| X_i - W_{ij} X_j \right\|$ 来求出 W_{ij}.

$$min\ \varepsilon(\boldsymbol{W}) = \sum_i \left\| \boldsymbol{x}_i - \sum_j w_{ij} \boldsymbol{x}_{ij} \right\|^2$$

3 保持权值 W_{ij} 不变, 求 X_i 在低维空间的象 Y_i, 使得低维重构误差最小.

$$min\ \phi(\boldsymbol{Y}) = \sum_i \left\| \boldsymbol{y}_i - \sum_j w_{ij} \boldsymbol{y}_{ij} \right\|^2$$

LLE算法的求解

1 计算每一个点 X_i 的近邻点.

2 对于点 X_i 和它的近邻点的权值 W_{ij},

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

其中 $$\sum_j w_j=1$$

构造局部协方差矩阵

$$G^i_{jk}=(X_i-\eta_j)\bullet(X_i-\eta_l),\eta_j,\eta_l\text{为}X_i\text{的近邻点.}$$

则求解得 $$W_{ij}=\frac{\sum_k G^{i\ -1}_{jk}}{\sum_{lm} G^{i\ -1}_{lm}}$$

LLE算法的求解

1 计算每一个点 X_i 的近邻点.

2 对于点 X_i 和它的近邻点的权值 W_{ij} ,

3 在低维空间固定**Wij,**求

$$min\ \phi(\boldsymbol{Y}) = \sum_i \left\| \boldsymbol{y}_i - \sum_j w_{ij} \boldsymbol{y}_{ij} \right\|^2$$

满足约束

$$\sum_i \vec{Y}_i = \vec{0} \quad \text{（中心化），} \quad \frac{1}{N} \sum_i \vec{Y}_i \vec{Y}_i^T = I \quad \text{（单位协方差）}$$

又

$$\Phi(Y) = \sum_{ij} M_{ij} \left(\vec{Y}_i \cdot \vec{Y}_j \right) \qquad M = (I - W)^T (I - W)$$

问题最终转化为求解**My=λy**的特征值分解问题。

LLE算法的求解

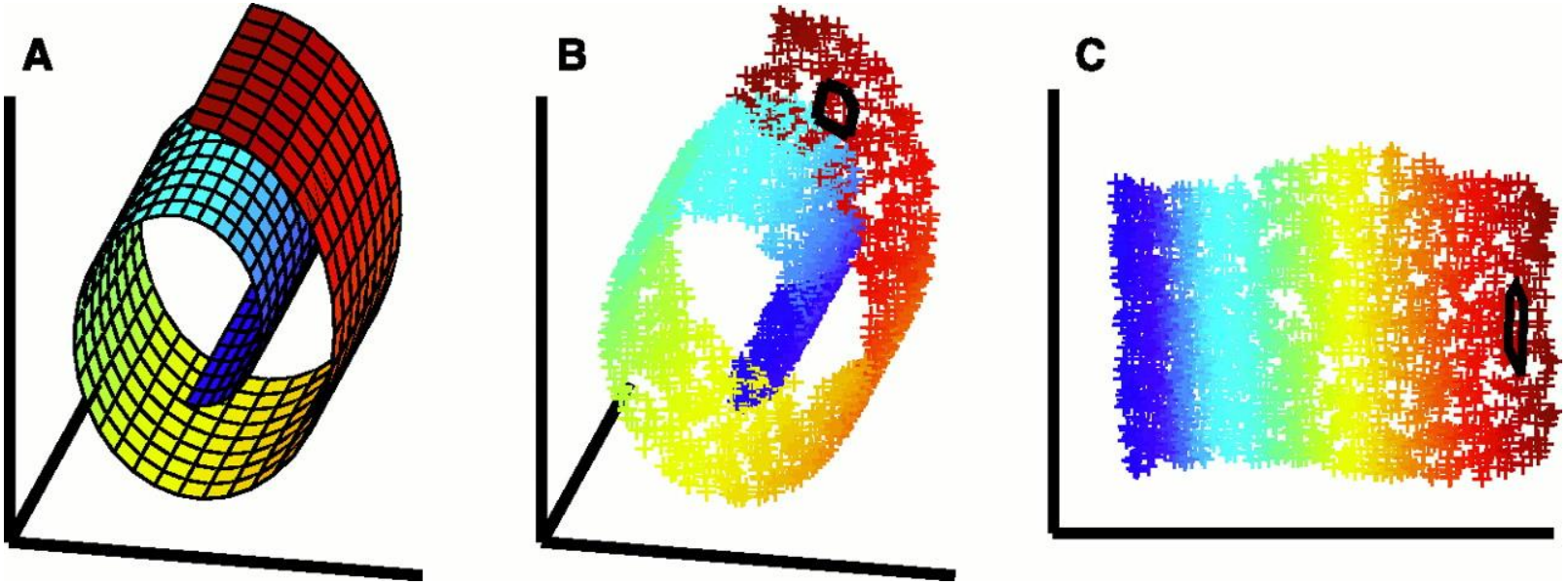
1 计算每一个点 X_i 的近邻点.

2 对于点 X_i 和它的近邻点的权值 W_{ij} ,

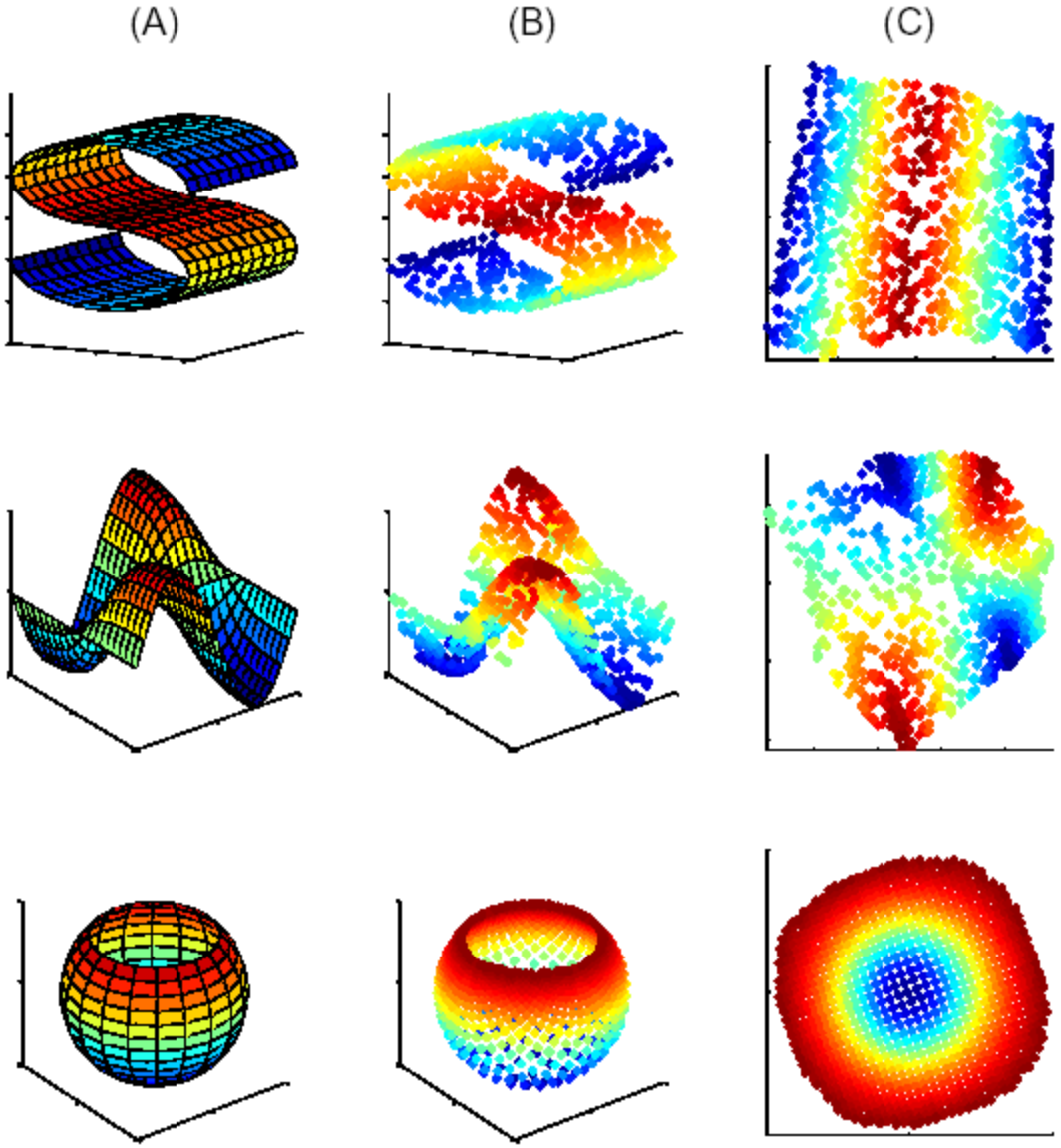
$$W_{ij}=\frac{\sum_k G^i_{jk}{}^{-1}}{\sum_{lm} G^i_{lm}{}^{-1}},\quad \text{其中 } G^i_{jk}=(X_i-\eta_j)\bullet(X_i-\eta_l),\ \eta_j,\eta_l \text{为} X_i \text{的近邻点.}$$

3 令 $W=(W_{ij})$, $M=(I-W)^{\mathrm{T}}(I-W)$, 低维嵌入是 M 的最小的第 **2**到第 **d**+**1** 个特征向量.

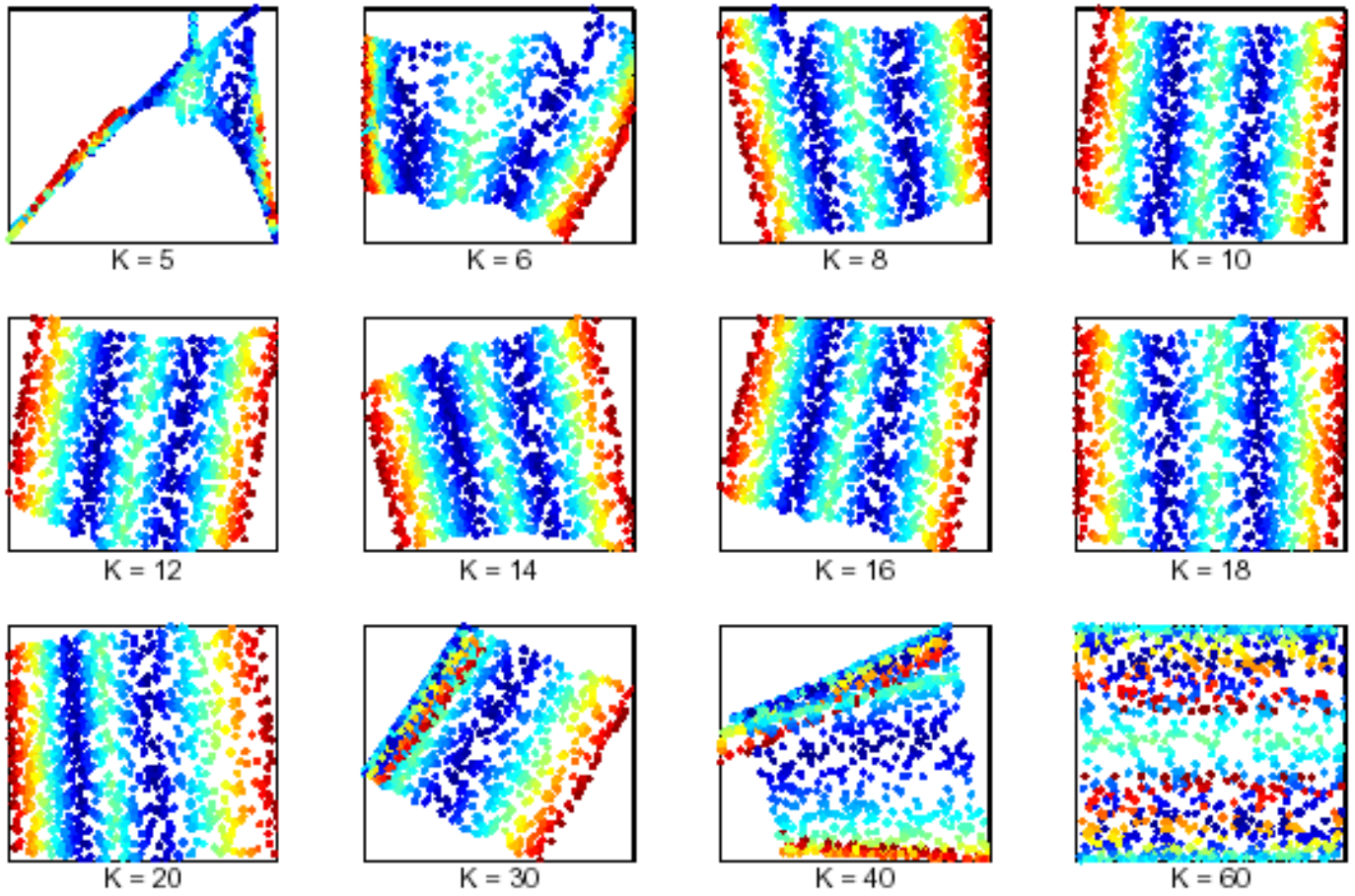
LLE实验结果



LLE实验结果



LLE实验结果



邻域参数的影响

LLE实验结果

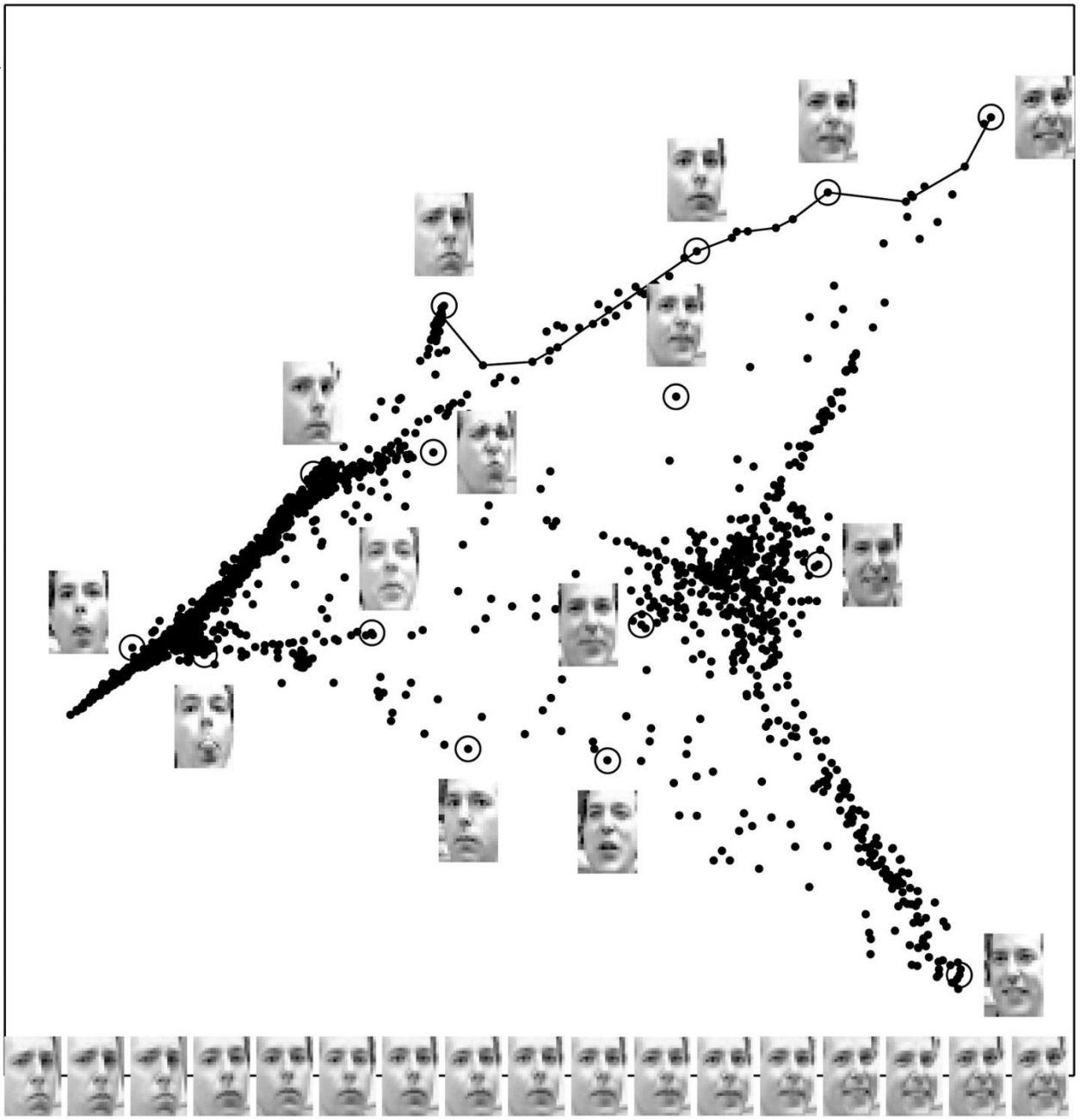
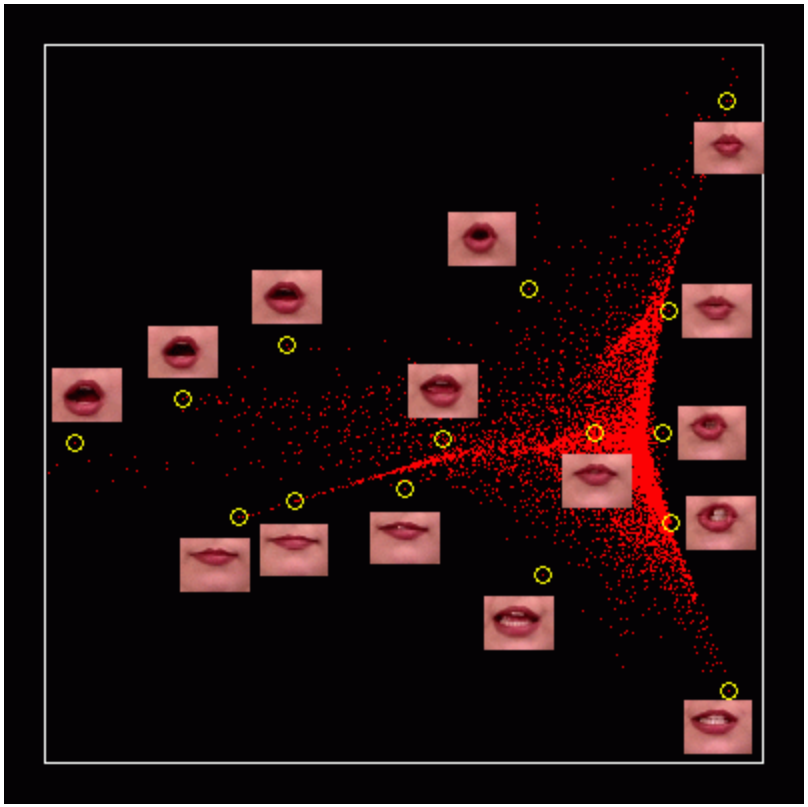
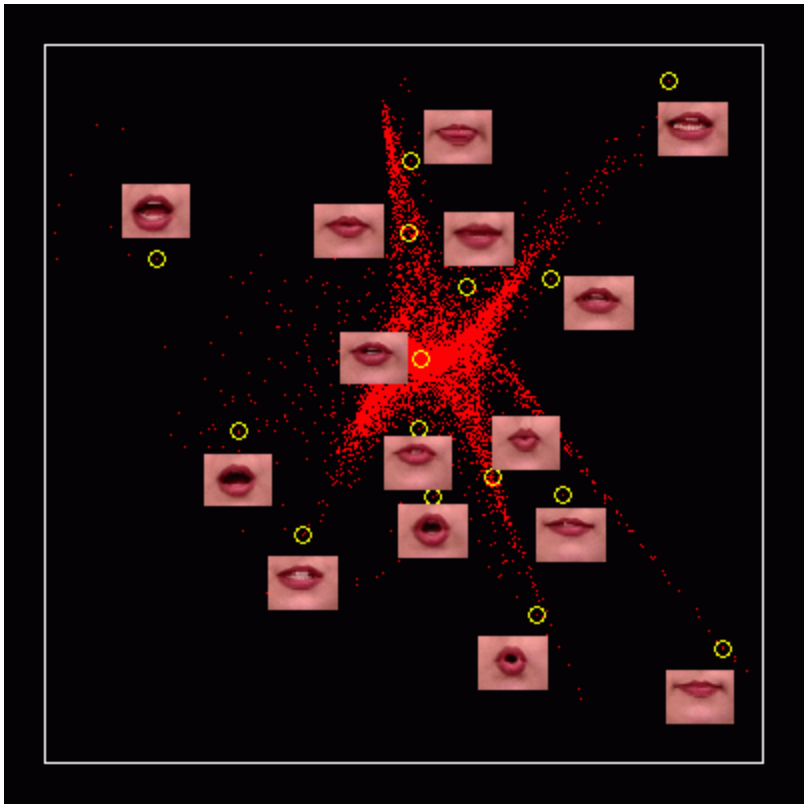


Figure from
LLE paper

LLE实验结果





代表性算法-2

- LLE (Locally linear embedding)
  - 优点
    - 算法可以学习任意维的局部线性的低维流形
    - 算法归结为稀疏矩阵特征值计算，计算复杂度相对较小
  - 缺点
    - 算法所学习的流形只能是不闭合的
    - 算法要求样本在流形上是稠密采样的
    - 算法对样本中的噪声和邻域参数比较敏感

Numerical Issues

- Covariance matrix used to compute W can be ill-conditioned, regularization needs to be used
- Small eigen values are subject to numerical precision errors and to getting mixed
- But, sparse matrices used in this algorithm make it much faster then Isomap

重点介绍的几个方法

- **等距映射(ISOMAP)**

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.

- **局部线性嵌入(LLE)**

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, vol. 290, pp. 2323--2326, 2000.

- **拉普拉斯特征映射(Laplacian Eigenmap)**

M. Belkin, P. Niyogi, Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, Vol. 15, Issue 6, pp. 1373 –1396, 2003 .

代表性算法-3

- LE (Laplacian Eigenmap)
  - 基本思想：在高维空间中离得很近的点投影到低维空间中的象也应该离得很近.
  - 求解方法：求解图拉普拉斯算子的广义特征值问题.

拉普拉斯算子

设 **M** 是光滑的黎曼流形, f 是 **M** 上的光滑函数, ∇f 是 f 的梯度, 则称线性映射

$$\Delta:\ C^{\infty}(M)\to C^{\infty}(M),\quad \Delta f=-\mathrm{div}(\nabla f)$$

为 **M** 上的拉普拉斯算子, 其中**div**是散度算子.

图上的拉普拉斯算子

设 **G** 是一个图, **v** 是它的顶点, d_v 是 **v** 的自由度, **w(u,v)** 是连接顶点**u,v** 的边的权值,令

$$l(u,v)=\begin{cases}\mathrm{d}_v - w(u,v) & u=v\\ -w(u,v) & u,v\text{是连接的}\\ 0 & \text{其它}\end{cases}$$

$L = T^{-\frac{1}{2}} l T^{-\frac{1}{2}}$, 其中 T 是对角矩阵,对角线的元素为 $\sum_{\substack{u\\ u\sim v}} -w(u,v)$, 则称 L 为图 **G** 上的拉普拉斯算子.

Laplacian Eigenmap 算法流程

1 从样本点构建一个近邻图, 图的顶点为样本点, 离得很近两点用边相连 **(K**近邻或 ε邻域**).**

2 给每条边赋予权值 如果第 ***i***个点和第 **j** 个点不相连, 权值为**0**，否则 $W_{ij}=1$ **;**

3 计算图拉普拉斯算子的广义特征向量, 求得低维嵌入.

令***D***为对角矩阵 $D_{ii}=\sum_j W_{ji}, L=D-W$,***L***是近邻图上的拉普拉斯算子, 求解广义特征值问题 $Lf=\lambda Df$.

Laplacian Eigenmap实验结果(1)

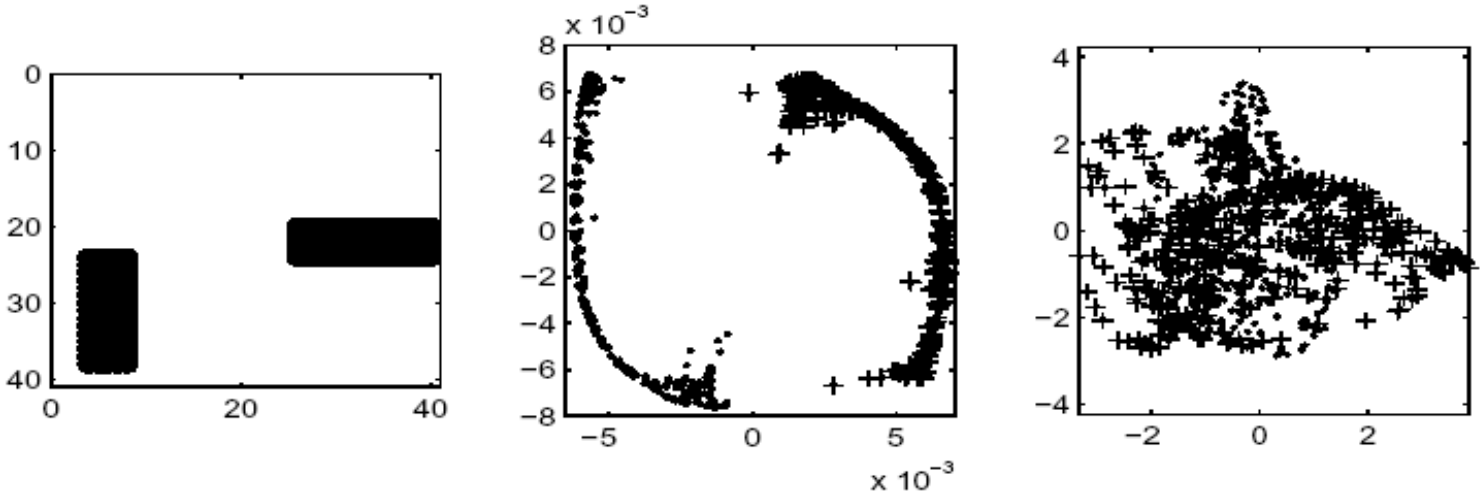
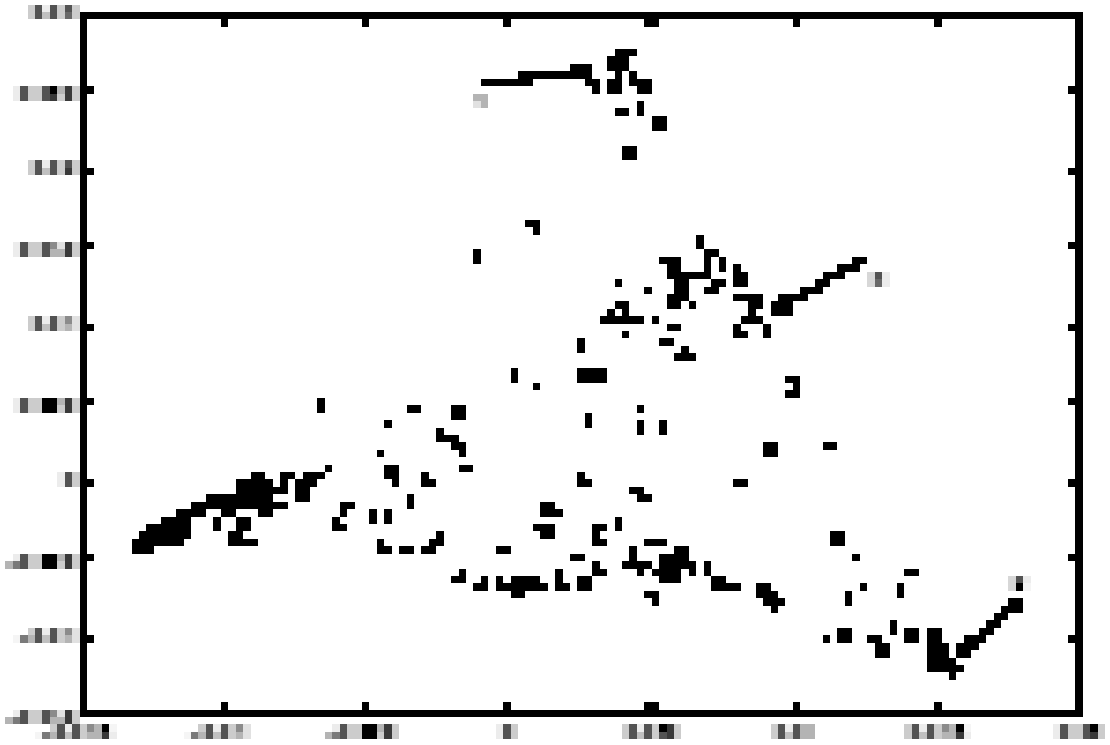


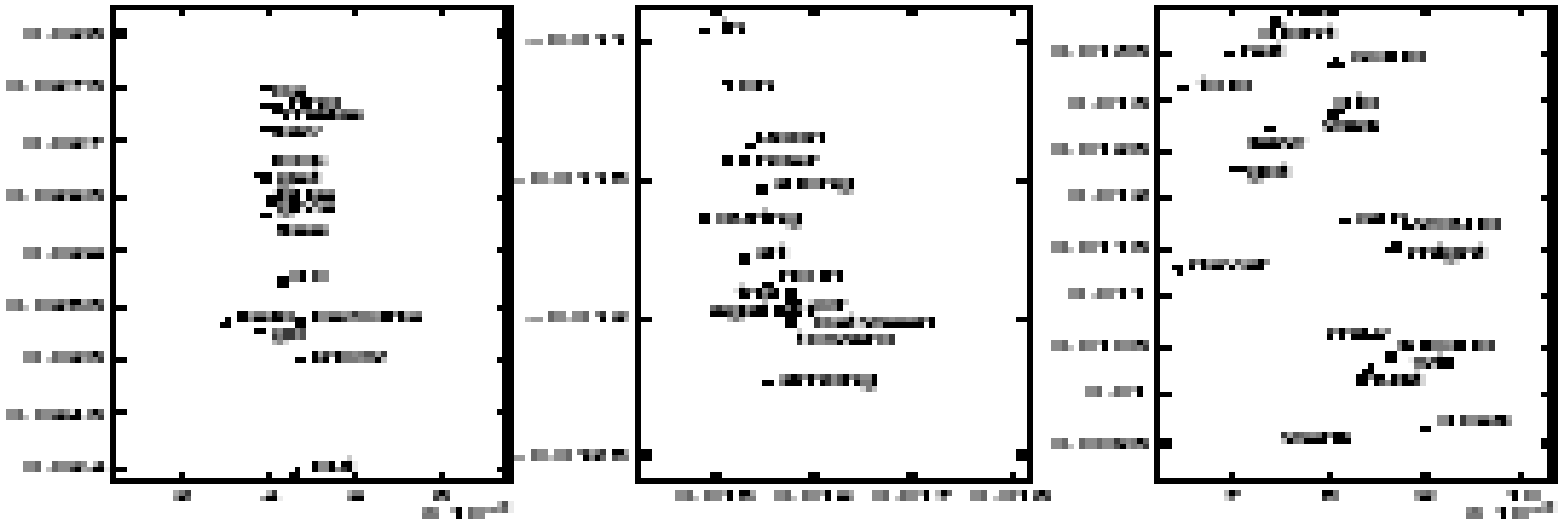
Figure 3: The left panel shows a horizontal and a vertical bar. The middle panel is a two dimensional representation of the set of all images using the Laplacian eigenmaps. The right panel shows the result of a principal components analysis using the first two principal directions to represent the data. Blue dots correspond to images of vertical bars and red '+' signs correspond to images of horizontal bars.

Laplacian Eigenmap实验结果(2)



300 most frequent words of the Brown corpus represented in the spectral domain

Laplacian Eigenmap实验结果(2)



The first is exclusively infinitives of verbs, the second contains prepositions and the third mostly modal and auxiliary verbs. We see that syntactic structure is well-preserved.

代表性算法-3

- LE (Laplacian Eigenmap)
  - 优点
    - 算法是局部非线性方法，与谱图理论有很紧密的联系.
    - 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，效率非常高
    - 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
  - 缺点
    - 同样对算法参数和数据采样密度较敏感
    - 不能有效保持流形的全局几何结构

经典方法小结

- 优点
  - 非参数：不需要对流形的很多参数作假设
  - 非线性：基于流形内在几何结构，体现现实数据的本质
  - 求解简单：转化为求解优化问题，通常采用特征值分解，而不需要采用迭代算法
- 缺点
  - 对观察数据存在流形结构的假设
  - 需要调节较多的算法参数，如k-NN的邻域参数k
  - 对数据采样稠密性、均匀性以及噪声数据的敏感性

研究难点与未来方向

- 如何进行统一有效的定量化评估
  - 真实数据 vs. 人工数据
  - 理论分析依据
  - 评估指标：一致性，收敛率，稳定性，复杂度…
- 如何求解测试数据的out-of-sample问题
  - 线性近似
  - 回归方法
- 如何确定低维目标空间的维数
- 如何进行监督式推广应用于分类问题

参考文献

- Roweis, S. T. and L. K. Saul (2000). "Nonlinear dimensionality reduction by locally linear embedding " Science **290**(5500): 2323-2326.
- Tenenbaum, J. B., V. de Silva, et al. (2000). "A global geometric framework for nonlinear dimensionality reduction " Science **290**(5500): 2319-2323.
- Vlachos, M., C. Domeniconi, et al. (2002). "Non-linear dimensionality reduction techniques for classification and visualization." Proc. of 8th SIGKDD, Edmonton, Canada.
- de Silva, V. and Tenenbaum, J. (2003). “Global versus local methods for nonlinear dimensionality reduction”, Advances in Neural Information Processing Systems,15.
- Law, Martin. Nonlinear Dimensionality Reduction and Manifold Learning. 2005.
- Lin, Zhouchen. A Glance over Manifold Learning. 2008.
- 杨剑. 流形学习问题. 2004.