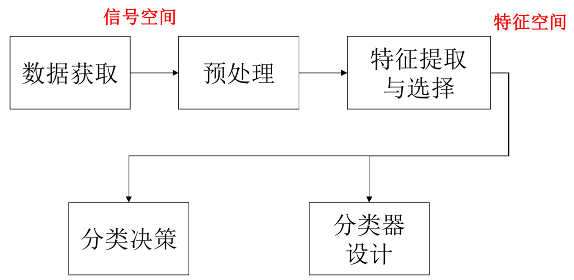
第十章：降维与度量学习

大纲

- 前言
- 多维缩放
- 主成分分析
- 流形学习
- 度量学习

前言



特征的选择与提取举例

- 细胞自动识别：
  - 原始测量：（正常与异常）细胞的数字图像
  - 原始特征（特征的形成，找到一组代表细胞性质的特征）：细胞面积，胞核面积，形状系数，光密度，核内纹理，和浆比
  - 压缩特征：原始特征的维数仍很高，需压缩以便于分类
    - 特征选择：挑选最有分类信息的特征
    - 特征提取：数学变换
      - 傅立叶变换或小波变换
      - 用PCA方法作特征压缩(Principle Component Analysis)

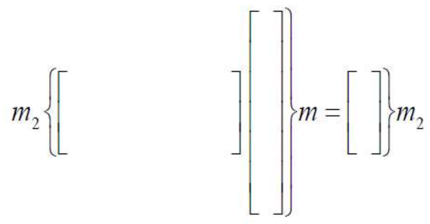
特征的选择与提取

- 两类提取有效信息、压缩特征空间的方法：特征提取和特征选择
- **特征提取** (extraction)：用映射（或变换）的方法把原始特征变换为较少的新特征

将 m 个特征变为 m_2 个新特征　　　　--- 二次特征

- **特征选择**(selection) ：从原始特征中挑选出一些最有代表性，分类性能最好的特征

从 m 个特征中选择 m_1 个， $m_1 < m$ （人为选择、算法选择）



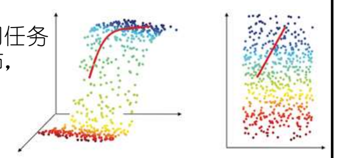
或从等维变换中选择若干个（cf，K—L，PCA）

必要时也可进行升维（非线性）变换

低维嵌入

- 缓解维数灾难的一个重要途径是降维(dimension reduction)
  - 即通过某种数学变换，将原始高维属性空间转变为一个低维“子空间” (subspace)，在这个子空间中样本密度大幅度提高，距离计算也变得更为容易。

- 为什么能进行降维？
  - 数据样本虽然是高维的，但与学习任务密切相关的也许仅是某个低维分布，即高维空间中的一个低维“嵌入” (embedding)，因而可以对数据进行有效的降维。



(a) 三维空间中观察到的样本点　(b) 二维空间中的曲面

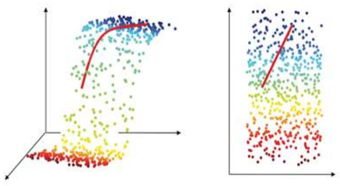
图 10.2　低维嵌入示意图

多维缩放(MDS)

- 若要求原始空间中样本之间的距离在低维空间中得以保持，即得到“多维缩放” (Multiple Dimensional Scaling, MDS)：

- 假定有m个样本，在原始空间中的距离矩阵为 $\mathbf{D} \in \mathbb{R}^{m\times m}$，其第i行j列的元素$dist_{ij}$为样本 $\boldsymbol{x}_i$ 到 $\boldsymbol{x}_j$ 的距离。
- 目标是获得样本在d'维空间中的欧氏距离等于原始空间中的距离，即 $||\boldsymbol{z}_i - \boldsymbol{z}_j|| = dist_{ij}$.
- 令 $\mathbf{B} = \mathbf{Z}^{\mathrm{T}}\mathbf{Z} \in \mathbb{R}^{m\times m}$，其中$\mathbf{B}$为降维后的内积矩阵，$b_{ij} = \boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{z}_j$，有

$$\begin{aligned} dist_{ij}^2 &= ||\boldsymbol{z}_i||^2 + ||\boldsymbol{z}_j||^2 - 2\boldsymbol{z}_i^T\boldsymbol{z}_j \\ &= b_{ii} + b_{jj} - 2b_{ij}. \end{aligned}$$



(a) 三维空间中观察到的样本点　(b) 二维空间中的曲面

图 10.2　低维嵌入示意图

多维缩放

- 为便于讨论，令降维后的样本$\mathbf{Z}$被中心化，即$\sum_{i=1}^{m} \boldsymbol{z}_i = \mathbf{0}$。显然，矩阵$\mathbf{B}$的行与列之和均为零，即

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

易知 $\sum_{i=1}^{m} dist_{ij}^2 = \mathrm{tr}(\mathbf{B}) + m b_{jj}$, $\sum_{j=1}^{m} dist_{ij}^2 = \mathrm{tr}(\mathbf{B}) + m b_{ii}$, $\sum_{i=1}^{m}\sum_{j=1}^{m} dist_{ij}^2 = 2m\ \mathrm{tr}(\mathbf{B})$,

其中tr(·)表示矩阵的迹(trace), $\mathrm{tr}(\mathbf{B}) = \sum_{i=1}^{m} \|\boldsymbol{z}_i\|^2$。令

$$dist_{i\cdot}^2 = \frac{1}{m}\sum_{j=1}^{m} dist_{ij}^2 \ , \quad dist_{\cdot j}^2 = \frac{1}{m}\sum_{i=1}^{m} dist_{ij}^2 \quad dist_{\cdot\cdot}^2 = \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} dist_{ij}^2$$

由此即可通过降维前后保持不变的距离矩阵D求取内积矩阵B ：

$$b_{ij} = -\frac{1}{2}(dist_{ij}^2 - dist_{i\cdot}^2 - dist_{\cdot j}^2 + dist_{\cdot\cdot}^2)$$

多维缩放

- 对矩阵$\mathbf{B}$做特征值分解(eigenvalue decomposition)$\mathbf{B} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{\mathrm{T}}$，其中$\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_d)$为特征值构成的对角矩阵，$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$为特征向量矩阵，假定其中有$d^*$个非零特征值，它们构成对角矩阵$\mathbf{\Lambda}_* = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_{d*})$，$\mathbf{V}$为特征向量矩阵。令$\mathbf{V}_*$表示相应的特征矩阵，则$\mathbf{Z}$可表达为$\mathbf{Z} = \lambda_*^{1/2}\mathbf{V}_*^{\mathrm{T}} \in \mathbb{R}^{d^* \times m}$。

- 在现实应用中为了有效降维，往往仅需降维后的距离与原始空间中的距离尽可能接近，而不必严格相等。此时可取$d' \ll d$个最大特征值构成对角矩阵$\tilde{\mathbf{\Lambda}} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_{d'})$，令$\tilde{\mathbf{V}}$表示相应的特征向量矩阵，则$\mathbf{Z}$可表达为

$$\mathbf{Z} = \tilde{\mathbf{\Lambda}}^{1/2}\tilde{\mathbf{V}}^{\mathrm{T}} \in \mathbb{R}^{d' \times m}.$$

多维缩放

MDS算法的描述

输入: 距离矩阵 $\mathbf{D} \in \mathbb{R}^{m \times m}$, 其元素 $dist_{ij}$ 为样本 $\boldsymbol{x}_i$ 到 $\boldsymbol{x}_j$ 的距离;
低维空间维数 d'.

过程:

1: 根据式(10.7)–(10.9)计算 $dist_{i\cdot}^2$, $dist_{\cdot j}^2$, $dist_{\cdot\cdot}^2$;
2: 根据式(10.10)计算矩阵 $\mathbf{B}$;
3: 对矩阵 $\mathbf{B}$ 做特征值分解;
4: 取 $\tilde{\mathbf{\Lambda}}$ 为 d' 个最大特征值所构成的对角矩阵, $\tilde{\mathbf{V}}$ 为相应的特征向量矩阵.

输出: 矩阵 $\tilde{\mathbf{V}}\tilde{\mathbf{\Lambda}}^{1/2} \in \mathbb{R}^{m \times d'}$, 每行是一个样本的低维坐标

图 10.3 MDS 算法

线性降维方法

- 一般来说，欲获得低维子空间，最简单的是对原始高维空间进行线性变换。给定d维空间中的样本$\mathbf{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m) \in \mathbb{R}^{d \times m}$，变换之后得到$d' \leq d$维空间中的样本

$$\mathbf{Z} = \mathbf{W}^{\mathrm{T}}\mathbf{X},$$

其中$\mathbf{W} \in \mathbb{R}^{d \times d'}$是变换矩阵，$\mathbf{Z} \in \mathbb{R}^{d' \times m}$是样本在新空间中的表达。

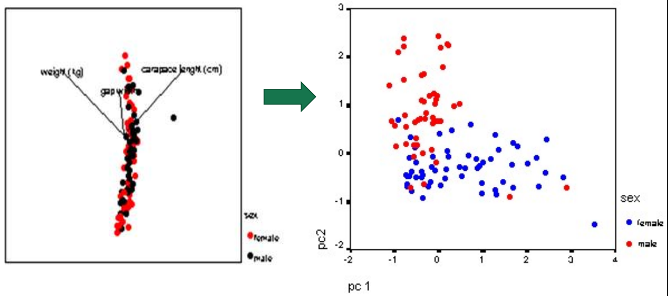
- 变换矩阵$\mathbf{W}$可视为d'个d维属性向量。换言之，$\boldsymbol{z}_i$是原属性向量$\boldsymbol{x}_i$在新坐标系$\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{d'}\}$中的坐标向量。若$\boldsymbol{w}_i$与$\boldsymbol{w}_j (i \neq j)$正交，则新坐标系是一个正交坐标系，此时$\mathbf{W}$为正交变换。显然，新空间中的属性是原空间中的属性的线性组合。

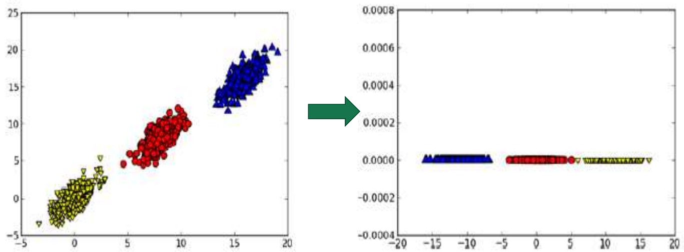
- 基于线性变换来进行降维的方法称为线性降维方法，对低维子空间性质的不同要求可通过对$\mathbf{W}$施加不同的约束来实现。

主成分分析（PCA）

- 主成分分析(或称主分量分析，principal component analysis)由皮尔逊(Pearson,1901)首先引入，后来被霍特林(Hotelling,1933)发展。
- 在PCA中，我们感兴趣的是找到一个从原d维输入空间到新的k维空间的具有最小信息损失的映射

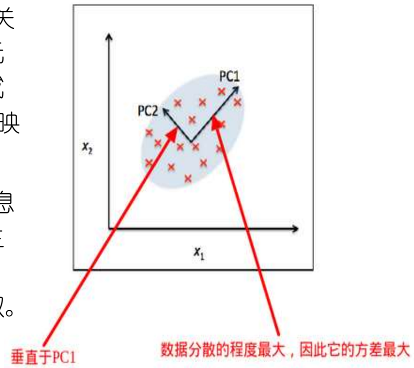
13





PCA的基本思想

- PCA可以把可能具有相关性的高维变量合成线性无关的低维变量，称为主成分；n维数据集可以通过映射降成k维子空间；
- 准则：压缩数据时让信息损失最小化，即第一个主成分是从数据差异最大（方差最大）的方向提取。依次处理



PCA的数学描述

- 原特征表示:x=$[x_1,......,x_p]^T$
- 新特征表示:x'=$[x_1',......,x_k']^T$

$$x_i' = \sum_{j=1}^{p} \alpha_{ij} x_j = \alpha_i^T x$$

$$x' = W^T x, \alpha_i^T \alpha_i = 1 \qquad \alpha_i = [\alpha_{i1,}\ldots\ldots \alpha_{ip}]^T$$

$$W = [\alpha_{1,}\ldots\ldots \alpha_k] i = 1,...k$$

PCA的公式推导

- 考虑第一个新特征x_1':

$$\mathrm{var}(x_1') = E(x_1'^2) - E(x_1')^2$$

$$= E[\alpha_1^T x x^T \alpha_1] - E(\alpha_1^T x) E(x^T \alpha_1)$$

$$= \alpha_1^T \Sigma \alpha_1$$

目标函数:

$$\underset{\alpha_1}{\mathrm{argmax}}\ \mathrm{var}(x_1) = \underset{\alpha_1}{\mathrm{argmax}}\ \alpha_1^T \Sigma \alpha_1$$

PCA的公式推导

$$f(\alpha_1) = \alpha_1^T \Sigma \alpha_1 - \lambda (\alpha_1^T \alpha_1 - 1) = 0$$

对α_1求导: $\Sigma \alpha_1 - \lambda \alpha_1 = 0$

$$\mathrm{Var}(x_1) = \alpha_1^T \Sigma \alpha_1 = \lambda \alpha_1^T \alpha_1$$

α_1为Σ的最大特征向量对应的特征向量

PCA的公式推导

除了满足协方差最大外，第二个特征还要与第一个特征无关，即：

$$E(x_2 x_1) - E(x_2) E(x_1) = 0$$

$$E[\alpha_2^T x x^T \alpha_1] - E[\alpha_2^T x] E(x^T \alpha_1) = 0$$

$$\alpha_2^T \Sigma \alpha_1 = 0 \qquad f(\alpha_2) = \alpha_2^T \Sigma \alpha_2 - \lambda_2 (\alpha_2^T \alpha_2 - 1) - \lambda_2' \alpha_2^T \alpha_1$$

$$\alpha_2^T \alpha_1 = 0 \qquad \Sigma \alpha_2 - \lambda_2 \alpha_2 = 0$$

α_2为Σ的第二大特征向量对应的特征向量

PCA计算过程

- 去除平均值
- 计算协方差矩阵：$\Sigma = \frac{1}{m}\sum_{i=1}^{m} x^{(i)} {x^{(i)}}^{T}$
- 计算特征向量和特征值：　$\Sigma\alpha_1 = \lambda\alpha_1$
- 将特征值从大到小排列
- 保留最上面的k个特征向量
- 将数据转换到上述k个特征向量构建的新空间

$$W = [\alpha_1, \ldots, \alpha_k]$$

$$X' = W^T X$$

主成分分析

PCA算法

输入： 样本集 $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$;
　　　低维空间维数 d'.

过程：

1: 对所有样本进行中心化：$\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i - \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{x}_i$;
2: 计算样本的协方差矩阵 $\mathbf{X}\mathbf{X}^{\mathrm{T}}$;
3: 对协方差矩阵 $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ 做特征值分解;
4: 取最大的 d' 个特征值所对应的特征向量 $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{d'}$.

输出： 投影矩阵 $\mathbf{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{d'})$.

图 10.5　PCA 算法

PCA计算过程

Data =

x	y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

T

$\bar{x} = 1.81 \quad \bar{y} = 1.91$

DataAdjust =

x	y
.69	.49
-1.31	-1.21
.39	.99
.09	.29
1.29	1.09
.49	.79
.19	-.31
-.81	-.81
-.31	-.31
-.71	-1.01

PCA计算过程

DataAdjust =

x	y
.69	.49
-1.31	-1.21
.39	.99
.09	.29
1.29	1.09
.49	.79
.19	-.31
-.81	-.81
-.31	-.31
-.71	-1.01

$$C = \begin{bmatrix} cov(x,x) & cov(x,y) \\ cov(y,x) & cov(y,y) \end{bmatrix}$$

$$C = \begin{bmatrix} .61655556 & .615444444 \\ .615444444 & .716555556 \end{bmatrix}$$

PCA计算过程

$$\text{Cov} = \begin{bmatrix} .61655556 & .615444444 \\ .615444444 & .716555556 \end{bmatrix}$$

$$eigenvalues = \begin{bmatrix} 0.490833989 \\ 1.28402771 \end{bmatrix}$$

$$\begin{bmatrix} -0.735178656 & -0.677873399 \\ 0.677873399 & -0.735178656 \end{bmatrix}$$

将特征值按照从大到小的顺序排序，选择其中最大的k个。

我们选择其中最大的那个，这里是1 .28402771，对应的特征向量是：(-0.677873399，-0.735178656)^T

PCA计算过程

$$\text{FinalData}(m*k) = \text{DataAdjust}(m*n) \times \text{EigenVectors}(n*k)$$

DataAdjust =

x	y
.69	.49
-1.31	-1.21
.39	.99
.09	.29
1.29	1.09
.49	.79
.19	-.31
-.81	-.81
-.31	-.31
-.71	-1.01

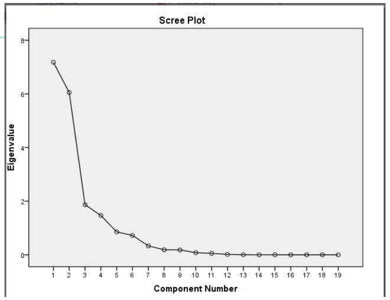
Transformed Data (Single eigenvector)

x
-.827970186
1.77758033
-.992197494
-.274210416
-1.67580142
-.912949103
.0991094375
1.14457216
.438046137
1.22382056

- 总方差中属于主成分Z_i的比例为：

$$\frac{\sum_{i=1}^{d'} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \geqslant t$$

称为主成分z_i的贡献率。

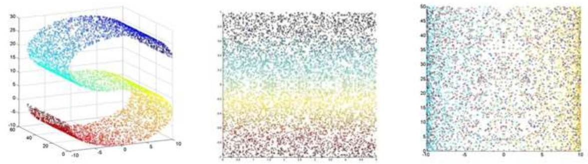


- 第一主成分z的贡献率最大，表明它解释原始变量$x_1 \dots x_n$的能力最强，而z_1，……,z_n的解释能力依次递减。主成分分析的目的就是为了减少变量的个数，因而一般是不会使用所有主成分的，忽略一些带有较小方差的主成分将不会给总方差带来大的影响。

27

核化线性降维

- 线性降维方法假设从高维空间到低维空间的函数映射是线性的，然而，在不少现实任务中，可能需要非线性映射才能找到恰当的低维嵌入：



(a) 三维空间中的观察　(b) 本真二维结构　(c) PCA 降维结果

图 10.6　三维空间中观察到的 3000 个样本点，是从本真二维空间中矩形区域采样后以 S 形曲面嵌入，此情形下线性降维会丢失低维结构．图中数据点的染色显示出低维空间的结构．

核化线性降维

核化主成分分析 (Kernelized PCA, 简称KPCA)

- 非线性降维的一种常用方法，是基于核技巧对线性降维方法进行“核化”（kernelized)。
- 假定我们数据映射到高维特征空间中，在高维特征空间把数据投影到由 W 确定的超平面上，即PCA欲求解

去均值化

$$\left(\sum_{i=1}^{m} \boldsymbol{z}_i \boldsymbol{z}_i^{\mathrm{T}}\right) \mathbf{W} = \lambda \mathbf{W}.$$

- 其中$\boldsymbol{z}_i$是样本点$\boldsymbol{x}_i$在高维特征空间中的像。令 $\boldsymbol{\alpha}_i = \frac{1}{\lambda} \boldsymbol{z}_i^{\mathrm{T}} \mathbf{W}$,

$$\mathbf{W} = \frac{1}{\lambda}\left(\sum_{i=1}^{m} \boldsymbol{z}_i \boldsymbol{z}_i^{\mathrm{T}}\right) \mathbf{W} = \sum_{i=1}^{m} \boldsymbol{z}_i \frac{\boldsymbol{z}_i^{\mathrm{T}} \mathbf{W}}{\lambda} = \sum_{i=1}^{m} \boldsymbol{z}_i \boldsymbol{\alpha}_i.$$

核化线性降维

核化主成分分析 (Kernelized PCA, 简称KPCA)

- 假定$\boldsymbol{z}_i$是由原始属性空间中的样本点$\boldsymbol{x}_i$通过映射ϕ产生，即

$$\boldsymbol{z}_i = \phi(\boldsymbol{x}_i),\ i = 1, 2, \ldots, m.$$

$$\mathbf{W} = \sum_{i=1}^{m} \boldsymbol{z}_i \boldsymbol{\alpha}_i$$

- 若ϕ能被显式表达出来，则通过它将样本映射至高维空间，再在特征空间中实施PCA即可，即有

$$\left(\sum_{i=1}^{m} \phi(\boldsymbol{x}_i) \phi(\boldsymbol{x}_i)^{\mathrm{T}}\right) \mathbf{W} = \lambda \mathbf{W}.$$

并且

$$\mathbf{W} = \sum_{i=1}^{m} \phi(\boldsymbol{x}_i) \boldsymbol{\alpha}_i.$$

核化线性降维

核化主成分分析 (Kernelized PCA, 简称KPCA)

- 一般情形下，我们不清楚ϕ的具体形式，于是引入核函数

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^{\mathrm{T}} \phi(\boldsymbol{x}_j).$$

- 又由$\mathbf{W} = \sum_{i=1}^{m} \phi(\boldsymbol{x}_i) \boldsymbol{\alpha}_i$，代入优化式$\left(\sum_{i=1}^{m} \phi(\boldsymbol{x}_i) \phi(\boldsymbol{x}_i)^{\mathrm{T}}\right) \mathbf{W} = \lambda \mathbf{W}$，有

$$\mathbf{K}\mathbf{A} = \lambda \mathbf{A}.$$

其中$\mathbf{K}$为κ对应的核矩阵，$(\mathbf{K})_{ij} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j), \mathbf{A} = (\boldsymbol{\alpha}_1; \boldsymbol{\alpha}_2; \ldots; \boldsymbol{\alpha}_m)$.

- 上式为特征值分解问题，取$\mathbf{K}$最大的d'个特征值对应的特征向量得到解。

核化线性降维

核化主成分分析 (Kernelized PCA, 简称KPCA)

- 对新样本$\boldsymbol{x}$，其投影后的第j $(j = 1, 2, \ldots, d')$维坐标为

$$\begin{aligned} z_j = \boldsymbol{w}_j^{\mathrm{T}} \phi(\boldsymbol{x}) &= \sum_{i=1}^{m} \alpha_i^j \phi(\boldsymbol{x}_i)^{\mathrm{T}} \phi(\boldsymbol{x}) \\ &= \sum_{i=1}^{m} \alpha_i^j \kappa(\boldsymbol{x}_i, \boldsymbol{x}). \end{aligned}$$

其中$\boldsymbol{\alpha}_i$已经过规范化，α_i^j是$\boldsymbol{\alpha}_i$的第j个分量。由该式可知，为获得投影后的坐标，KPCA需对所有样本求和，因此它的计算开销较大。

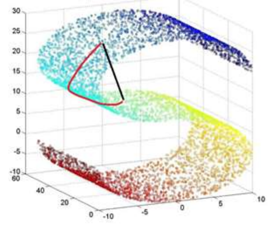
流形学习

- 流形学习(manifold learning)是一类借鉴了拓扑流形概念的降维方法。“流形”是在局部与欧氏空间同胚的空间，换言之，它在局部具有欧氏空间的性质，能用欧氏距离来进行距离计算。
- 若低维流形嵌入到高维空间中，则数据样本在高维空间的分布虽然看上去非常复杂，但在局部上仍具有欧氏空间的性质，因此，可以容易地在局部建立降维映射关系，然后再设法将局部映射关系推广到全局。
- 当维数被降至二维或三维时，能对数据进行可视化展示，因此流形学习也可被用于可视化。

流形学习

等度量映射(Isometric Mapping，Isomap)

- 低维流形嵌入到高维空间之后，直接在高维空间中计算直线距离具有误导性，因为高维空间中的直线距离在低维嵌入流形上不可达。而低维嵌入流形上两点间的本真距离是“测地线”(geodesic)距离。

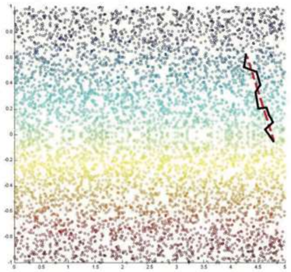


(a) 测地线距离与高维直线距离

流形学习

等度量映射(Isometric Mapping，Isomap)

- 测地线距离的计算：利用流形在局部上与欧氏空间同胚这个性质，对每个点基于欧氏距离找出其近邻点，然后就能建立一个近邻连接图，图种近邻点之间存在连接，而非近邻点之间不存在连接，于是，计算两点之间测地线距离的问题，就转变为计算近邻连接图上两点之间的最短路径问题。
- 最短路径的计算可通过Dijkstra算法或Floyd算法实现。得到距离后可通过多维缩放方法获得样本点在低维空间中的坐标。



(b) 测地线距离与近邻距离

流形学习

等度量映射(Isometric Mapping，Isomap)

输入： 样本集 $D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_m\}$;
近邻参数 k;
低维空间维数 d'.

过程：

for $i=1,2,\ldots,m$ **do**
  确定 $\boldsymbol{x}_i$ 的 k 近邻;
  $\boldsymbol{x}_i$ 与 k 近邻点之间的距离设置为欧氏距离，与其他点的距离设置为无穷大;
end for
调用最短路径算法计算任意两样本点之间的距离 $\mathrm{dist}(\boldsymbol{x}_i,\boldsymbol{x}_j)$;
将 $\mathrm{dist}(\boldsymbol{x}_i,\boldsymbol{x}_j)$ 作为 MDS 算法的输入;
return MDS 算法的输出

输出： 样本集 D 在低维空间的投影 $Z=\{\boldsymbol{z}_1,\boldsymbol{z}_2,\ldots,\boldsymbol{z}_m\}$.

图 10.8 Isomap 算法