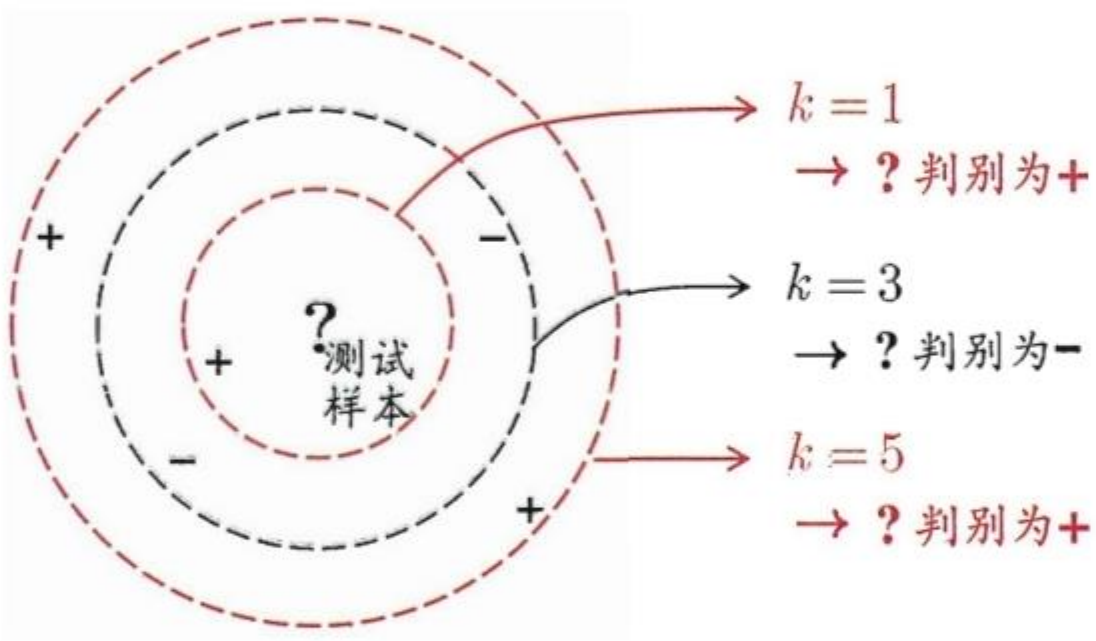
第十章 降维与度量学习

汇报人：何思成

K近邻学习



给定一个测试样本，基于一种距离度量寻找与其最相近的k个样本，根据这k个样本的信息来进行预测

- 分类任务：投票法
- 回归任务：平均法

K近邻学习

测试样本为x，最近邻样本为z，x与z类别标签不一致的概率（即最近邻分类器错误的概率）为：

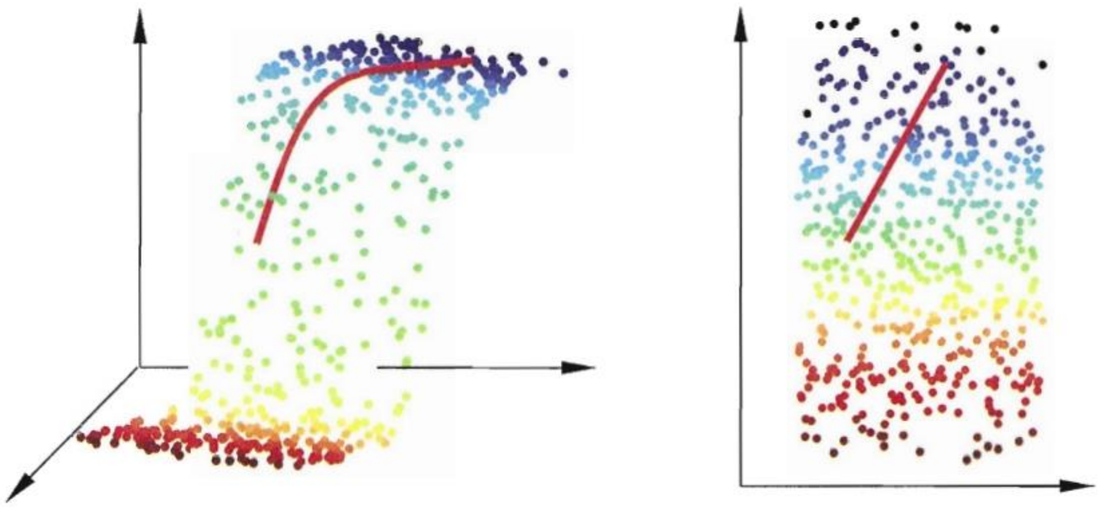
$$P(err) = 1 - \sum_{c \in \mathcal{Y}} P(c \mid \boldsymbol{x}) P(c \mid \boldsymbol{z}).$$

假设样本独立同分布且样本分布均匀，对于任意样本x，总能在任意近的范围内找到一个合适的训练样本z，1- P（c*| x）是贝叶斯最优分类器的结果有：

$$\begin{aligned} P(err) &= 1 - \sum_{c \in \mathcal{Y}} P(c \mid \boldsymbol{x}) P(c \mid \boldsymbol{z}) \\ &\simeq 1 - \sum_{c \in \mathcal{Y}} P^2(c \mid \boldsymbol{x}) \\ &\leqslant 1 - P^2(c^* \mid \boldsymbol{x}) \\ &= \left(1 + P(c^* \mid \boldsymbol{x})\right)\left(1 - P(c^* \mid \boldsymbol{x})\right) \\ &\leqslant 2 \times \left(1 - P(c^* \mid \boldsymbol{x})\right). \end{aligned}$$

KNN虽然简单，但是它的泛化错误率不超过贝叶斯最优分类器错误率的两倍

低维嵌入



事实上，一个任意测试样本x并不能总是在任意小的δ距离里找到测试样本

特征空间中样本维度越高，不仅数据样本越稀疏，度量测试样本和它的最近邻样本之间的距离也越困难

降维：把高维数据投影到低维空间，保持原始特征空间中样本之间的距离

假定 m 个样本在原始空间的距离矩阵为 $\mathbf{D} \in \mathbb{R}^{m\times m}$，其第 i 行 j 列的元素 $dist_{ij}$ 为样本 $\boldsymbol{x}_i$ 到 $\boldsymbol{x}_j$ 的距离. 我们的目标是获得样本在 d' 维空间的表示 $\mathbf{Z} \in \mathbb{R}^{d'\times m}$，$d' \leqslant d$，且任意两个样本在 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 $\mathbf{B}$ 为降维后样本的内积矩阵，$b_{ij} = \boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{z}_j$，有

$$\begin{aligned} dist_{ij}^2 &= \|\boldsymbol{z}_i\|^2 + \|\boldsymbol{z}_j\|^2 - 2\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{z}_j \\ &= b_{ii} + b_{jj} - 2b_{ij}\ . \end{aligned} \tag{10.3}$$

低维嵌入

为便于讨论，令降维后的样本 $\mathbf{Z}$ 被中心化，即 $\sum_{i=1}^{m} z_i = \mathbf{0}$. 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

$$\sum_{i=1}^{m} dist_{ij}^2 = \mathrm{tr}(\mathbf{B}) + m b_{jj} \ , \tag{10.4}$$

$$\sum_{j=1}^{m} dist_{ij}^2 = \mathrm{tr}(\mathbf{B}) + m b_{ii} \ , \tag{10.5}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{m} dist_{ij}^2 = 2m\ \mathrm{tr}(\mathbf{B}) \ , \tag{10.6}$$

其中 $\mathrm{tr}(\cdot)$ 表示矩阵的迹(trace)，$\mathrm{tr}(\mathbf{B}) = \sum_{i=1}^{m} \|z_i\|^2$. 令

$$dist_{i\cdot}^2 = \frac{1}{m}\sum_{j=1}^{m} dist_{ij}^2 \ , \tag{10.7}$$

$$dist_{\cdot j}^2 = \frac{1}{m}\sum_{i=1}^{m} dist_{ij}^2 \ , \tag{10.8}$$

$$dist_{\cdot\cdot}^2 = \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} dist_{ij}^2 \ , \tag{10.9}$$

求得：

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

此时我们得到的是z的内积，对bij进行特征值分解：

$$\mathbf{B} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{\mathrm{T}}$$

$$\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_d)$$

V是特征向量矩阵，Λ是由特征值构成的对角矩阵，假定其中有d * 个非零特征向量。
取其中d ″ << d个最大特征值构成新的对角矩阵，以及特征向量矩阵，得到：

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

低维嵌入

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

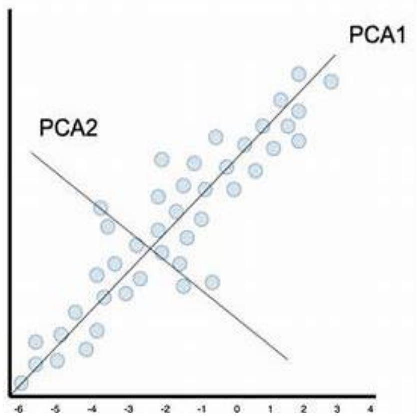
图 10.3 MDS 算法

如果最后几个特征值刚好为零，也就是说特征变换后没有数据丢失，此时就是线性降维：

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

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.

PCA



对于正交空间中的样本点，如何利用超平面对所有样本进行适当的表达?

- 最近重构性：样本点到这个个平面的距离都足够近
- 最大可分性：样本点在这个超平面上的投影都能尽可能分得开

假定数据样本进行了中心化, 即 $\sum_i \boldsymbol{x}_i = \mathbf{0}$; 再假定投影变换后得到的新坐标系为 $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_d\}$, 其中 $\boldsymbol{w}_i$ 是标准正交基向量, $\|\boldsymbol{w}_i\|_2 = 1$, $\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{w}_j = 0$ $(i \neq j)$. 若丢弃新坐标系中的部分坐标, 即将维度降低到 $d' < d$, 则样本点 $\boldsymbol{x}_i$ 在低维坐标系中的投影是 $\boldsymbol{z}_i = (z_{i1}; z_{i2}; \ldots; z_{id'})$, 其中 $z_{ij} = \boldsymbol{w}_j^{\mathrm{T}}\boldsymbol{x}_i$ 是 $\boldsymbol{x}_i$ 在低维坐标系下第 j 维的坐标. 若基于 $\boldsymbol{z}_i$ 来重构 $\boldsymbol{x}_i$, 则会得到 $\hat{\boldsymbol{x}}_i = \sum_{j=1}^{d'} z_{ij}\boldsymbol{w}_j$.

考虑整个训练集, 原样本点 $\boldsymbol{x}_i$ 与基于投影重构的样本点 $\hat{\boldsymbol{x}}_i$ 之间的距离为

$$\sum_{i=1}^{m}\left\|\sum_{j=1}^{d'} z_{ij}\boldsymbol{w}_j - \boldsymbol{x}_i\right\|_2^2 = \sum_{i=1}^{m}\boldsymbol{z}_i{}^{\mathrm{T}}\boldsymbol{z}_i - 2\sum_{i=1}^{m}\boldsymbol{z}_i{}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\boldsymbol{x}_i + \text{const}$$

$$\propto -\operatorname{tr}\left(\mathbf{W}^{\mathrm{T}}\left(\sum_{i=1}^{m}\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}\right)\mathbf{W}\right). \qquad (10.14)$$

$$\left\|\sum_{j}^{d'} z_{ij}\vec{w}_j - \vec{x}_i\right\|_2^2 = \left(\sum_{j}^{d'} z_{ij}\vec{w}_j^{T} - \vec{x}_i^{T}\right)\left(\sum_{j}^{d'} z_{ij}\vec{w}_j - \vec{x}_i\right)$$

$$= \underbrace{\left(\sum_{j}^{d'} z_{ij}\vec{w}_j^{T}\right)\left(\sum_{j}^{d'} z_{ij}\vec{w}_j\right)} - 2\sum_{j}^{d'} z_{ij}\vec{w}_j\cdot\vec{x}_i + \vec{x}_i^{T}\vec{x}_i$$

常数 const

$\begin{cases}\|\vec{w}_i\|_2 = 1 \\ \vec{w}_i\vec{w}_j = 0\end{cases}$

$$(\vec{z}_{i1}\vec{w}_1^{T} + \vec{z}_{i2}\cdot\vec{w}_2^{T} + \vec{z}_{i3}\cdot\vec{w}_3^{T} + \cdots \vec{z}_{id'}\cdot\vec{w}_{d'}^{T})(\vec{z}_{i1}\vec{w}_1 + \cdots + \vec{z}_{id'}\vec{w}_{d'})$$

$$= \sum_{j}^{d'} \vec{z}_{ij}\cdot\vec{z}_{ij}$$ 写成内积形式

$$= \vec{z}_i^{T}\cdot\vec{z}_i$$

$$\sum_{j}^{d'} \vec{z}_{ij}^{T}\cdot\vec{w}_j^{T}\cdot\vec{x}_i$$

$$= \vec{z}_i^{T}\cdot\begin{pmatrix}\vec{w}_1^{T}\\ \vec{w}_2^{T}\\ \vdots\\ \vec{w}_{d'}^{T}\end{pmatrix}\cdot\vec{x}_i$$

$$= \vec{z}_i^{T}\cdot\vec{z}_i$$

原式 $= -\sum_{i}^{d'} \vec{z}_i^{T}\cdot\vec{z}_i + \text{const}$

$$\sum_{i}^{d'}(\vec{w}_1^{T}\vec{x}_i \cdots \vec{w}_{d'}^{T}\vec{x}_i)\begin{pmatrix}\vec{w}_1^{T}\vec{x}_i\\ \vdots\\ \vec{w}_{d'}^{T}\vec{x}_i\end{pmatrix}$$

$$= \sum_{i}^{d'}(\vec{W}^{T}\vec{x}_i)\cdot(\vec{x}_i^{T}\cdot\vec{W})^{T}$$

$$\propto W^{T}\sum_{i}^{d'} x_i x_i^{T} W \quad \propto -tr\left(W^{T}\left(\sum_{i=1}^{m}\vec{x}_i\vec{x}_i^{T}\right)W\right)$$

原式正比于

PCA

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

基于最近重构性，上式应该被最小化，$\boldsymbol{w}_j$ 是标准正交基向量，$\sum_i \boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}$ 是协方差矩阵，得到：

$$\begin{aligned}&\min_{\mathbf{W}} \ -\mathrm{tr}\left(\mathbf{W}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W}\right)\\ &\text{s.t.}\ \mathbf{W}^{\mathrm{T}}\mathbf{W}=\mathbf{I}.\end{aligned}$$

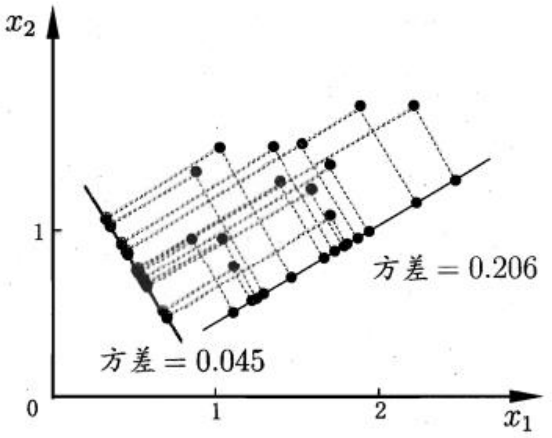


图 10.4 使所有样本的投影尽可能分开(如图中红线所示)，则需最大化投影点的方差

基于最大可分性，则应该使得投影后的样本点（$W^T x_i$）的方差最大化

于是，只需对协方差矩阵 $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ 进行特征值分解，将求得的特征值排序：$\lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant \lambda_d$，再取前 d' 个特征值对应的特征向量构成 $\mathbf{W}=(\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{d'})$. 这就是主成分分析的解. PCA 算法描述如图 10.5 所示.

$$\frac{\sum_{i=1}^{d'}\lambda_i}{\sum_{i=1}^{d}\lambda_i} \geqslant t\,.$$ 例如t =95%

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:

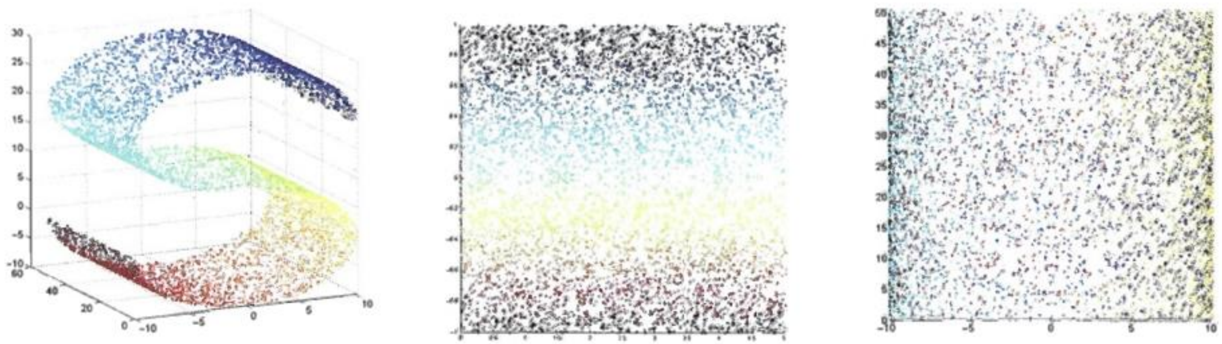
1: 对所有样本进行中心化: $\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i - \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{x}_i$;
2: 计算样本的协方差矩阵 $\mathbf{X}\mathbf{X}^{\mathrm{T}}$;
3: 对协方差矩阵 $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ 做特征值分解;
4: 取最大的 d' 个特征值所对应的特征向量 $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{d'}$.

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

图 10.5 PCA 算法

核化线性降维

从高维空间到低维空间的函数映射往往不是线性的，需要非线性映射才能得到合适的低维嵌入



(a) 三维空间中的观察　(b) 本真二维结构　(c) PCA 降维结果

核主成分分析（KPCA）：
首先假定数据从高维空间投影到W=（w1，w2....wd）所确定的超平面上，有：

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

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

假定z_i是由原始属性空间中的样本点x_i通过映射φ得到的

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

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

引入核函数：$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^{\mathrm{T}} \phi(\boldsymbol{x}_j) .$$

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

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

流形学习

流形是指在局部有和欧式空间有相同性质的空间·，能用欧氏距离进行距离计算，来进行降维

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，但局部具有欧式空间的性质，可以在局部建立降维映射关系，再推广到全局

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：

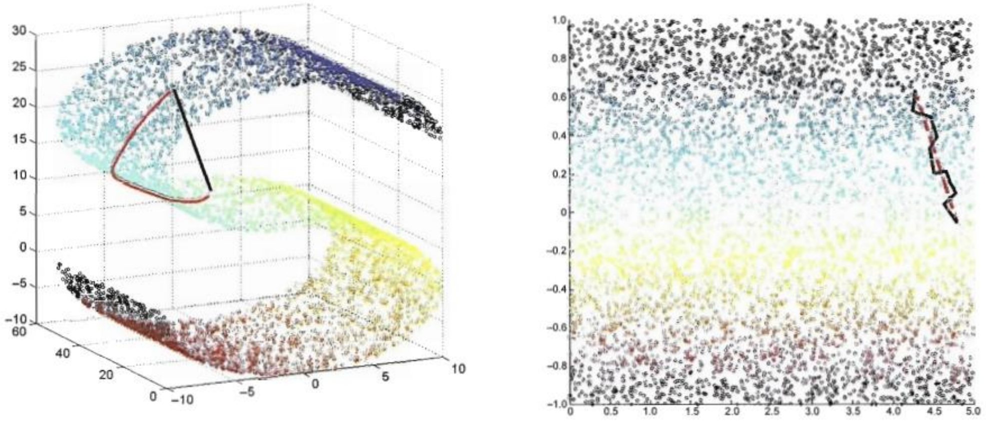
for $i=1,2,\ldots,m$ **do**
    确定 $\boldsymbol{x}_i$ 的 k 近邻;
    $\boldsymbol{x}_i$ 与 k 近邻点之间的距离设置为欧氏距离，与其他点的距离设置为无穷大;
end for
调用最短路径算法计算任意两样本点之间的距离 $\mathrm{dist}(\boldsymbol{x}_i,\boldsymbol{x}_j)$;
将 $\mathrm{dist}(\boldsymbol{x}_i,\boldsymbol{x}_j)$ 作为 MDS 算法的输入;
return MDS 算法的输出

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

1. 等密度映射

保持近邻样本之间的距离



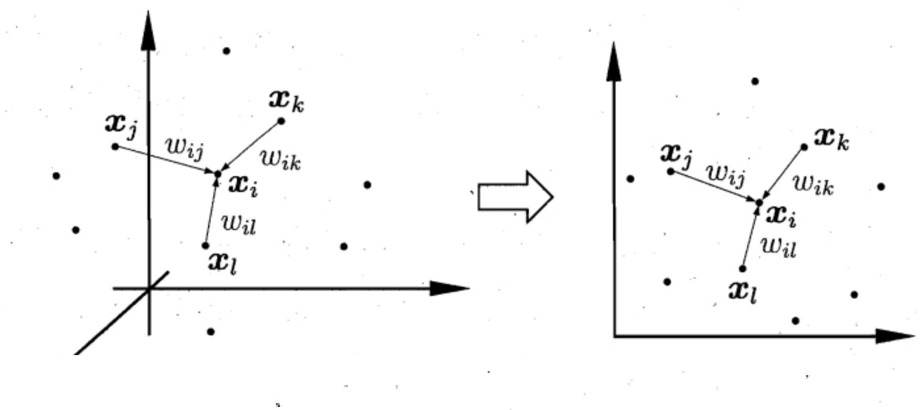
(a) 测地线距离与高维直线距离 (b) 测地线距离与近邻距离

图 10.7 低维嵌入流形上的测地线距离(红色)不能用高维空间的直线距离计算，但能用近邻距离来近似

近邻图构建：
1. 指定近邻点个数，例如欧氏距离最近的k个点作为近邻点
2. 指定距离阈值ε，距离小于ε被认为是近邻点

流形学习

2. 局部线性嵌入（LLE） 保持邻域内样本间的线性关系



$$\boldsymbol{x}_i = w_{ij}\boldsymbol{x}_j + w_{ik}\boldsymbol{x}_k + w_{il}\boldsymbol{x}_l$$

基于原样本点与线性重构后的点距离尽可能小并且线性关系保持不变，计算线性重构后的系数以及x_i对应的低维空间坐标z_i：

$$\min_{\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_m} \sum_{i=1}^{m} \left\| \boldsymbol{x}_i - \sum_{j\in Q_i} w_{ij}\boldsymbol{x}_j \right\|_2^2 \qquad (10.27)$$

$$\text{s.t.} \sum_{j\in Q_i} w_{ij} = 1,$$

$$\min_{\boldsymbol{z}_1,\boldsymbol{z}_2,\ldots,\boldsymbol{z}_m} \sum_{i=1}^{m} \left\| \boldsymbol{z}_i - \sum_{j\in Q_i} w_{ij}\boldsymbol{z}_j \right\|_2^2 . \qquad (10.29)$$

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:

1: **for** $i = 1, 2, \ldots, m$ **do**
2: 　确定 $\boldsymbol{x}_i$ 的 k 近邻;
3: 　从式(10.27)求得 w_{ij}, $j \in Q_i$;
4: 　对于 $j \notin Q_i$, 令 $w_{ij} = 0$;
5: **end for**
6: 从式(10.30)得到 $\mathbf{M}$;
7: 对 $\mathbf{M}$ 进行特征值分解;
8: **return** $\mathbf{M}$ 的最小 d' 个特征值对应的特征向量

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 10.10 LLE 算法

$$\mathbf{M} = (\mathbf{I} - \mathbf{W})^{\mathrm{T}}(\mathbf{I} - \mathbf{W}) , \qquad (10.30)$$

式10.29重写为：

$$\min_{\mathbf{Z}} \ \operatorname{tr}(\mathbf{Z}\mathbf{M}\mathbf{Z}^{\mathrm{T}}), \qquad (10.31)$$

$$\text{s.t.} \ \mathbf{Z}\mathbf{Z}^{\mathrm{T}} = \mathbf{I} .$$

对式10.31进行特征值分解，M中最小的d ‘个特征向量组成的矩阵就是Z^T

度量学习

降维的目的：高维数据降维过程中寻找一个合适的距离度量来把数据映射到低维空间使得在此空间学习比原始空间性能更好
度量学习：直接“学习”一个合适的距离度量

两个d维样本间的平方欧氏距离：

$$\mathrm{dist}^2_{\mathrm{ed}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = ||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2 = dist^2_{ij,1} + dist^2_{ij,2} + \ldots + dist^2_{ij,d} \,, \qquad (10.32)$$

考虑不同属性维度的重要性不同，引入属性权重w，则：

$$\begin{aligned}\mathrm{dist}^2_{\mathrm{wed}}(\boldsymbol{x}_i, \boldsymbol{x}_j) &= ||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2 = w_1 \cdot dist^2_{ij,1} + w_2 \cdot dist^2_{ij,2} + \ldots + w_d \cdot dist^2_{ij,d} \\ &= (\boldsymbol{x}_i - \boldsymbol{x}_j)^{\mathrm{T}} \mathbf{W} (\boldsymbol{x}_i - \boldsymbol{x}_j) \,, \end{aligned} \qquad (10.33)$$

其中 $w_i \geqslant 0$, $\mathbf{W} = \mathrm{diag}(\boldsymbol{w})$ 是一个对角矩阵, $(\mathbf{W})_{ii} = w_i$.

W替换为一个半正定对称矩阵M，得到“马氏距离”：

$$\mathrm{dist}^2_{\mathrm{mah}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i - \boldsymbol{x}_j)^{\mathrm{T}} \mathbf{M} (\boldsymbol{x}_i - \boldsymbol{x}_j) = ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2_{\mathbf{M}} \,,$$

M亦称为“度量矩阵”，度量学习实质上就是对M进行学习。

近邻成分分析（NCA）：NCA中样本之间的距离是通过学习的度量矩阵M定义的一种马氏距离。将M纳入近邻分类器的评价指标，通过优化该指标来求得M提高近邻分类器的性能

$$d_M(x_i, x_j) = \sqrt{(x_i - x_j)^T M (x_i - x_j)}$$

度量学习-近邻成分分析

对于每个样本x_i，希望它尽可能选择同类样本作为近邻。样本 i 选择样本 j 作为其近邻的概率：

$$p_{ij} = \frac{\exp(-\|x_i - x_j\|_M^2)}{\sum_{l \neq i} \exp(-\|x_i - x_l\|_M^2)}$$

$\|x_i - x_j\|_M^2$ 表示的是样本 i 和样本 j 在度量矩阵 M 下的距离平方：

$$\|x_i - x_j\|_M^2 = (x_i - x_j)^T M (x_i - x_j)$$

为了最大化正确分类的概率，NCA 的目标是最大化每个样本选择同类样本作为近邻的概率。损失函数为：

$$\mathcal{L}(M) = -\sum_i \log p_{i,\text{correct}}$$

$P_{i,correct}$即P_{ij}。通过最小化这个损失函数，NCA 学习到一个新的距离度量，使得同类样本在新的空间中更接近。

习题部分

10.3 在对高维数据降维之前应先进行“中心化”，常见的是将协方差矩阵 $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ 转化为 $\mathbf{X}\mathbf{H}\mathbf{H}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}$，其中 $\mathbf{H} = \mathbf{I} - \frac{1}{m}\mathbf{1}\mathbf{1}^{\mathrm{T}}$，试析其效果.

中心化的过程中用矩阵H去乘以原始数据矩阵X的时候，X的每一列的数据都会被减去这一列的平均值，使得这一列的平均值变为0。在这个过程中PCA超平面就能通过原点而不必考虑截距项，同时协方差矩阵中的均值项（即每个特征与其均值的差）变为0后，协方差矩阵更加简洁和易于分析。

10.5 降维中涉及的投影矩阵通常要求是正交的. 试述正交、非正交投影矩阵用于降维的优缺点.

正交：属性间独立，维数更低，耦合性小。
非正交：保留了属性间的联系。
正交投影矩阵还满足 最佳逼近定理。

10.7 试述核化线性降维与流形学习之间的联系及优缺点.

联系：都是非线性降维。
KPCA：先升到高维，在高维中是线性的，再用PCA降维。有核函数的优势，但计算量较大，核函数难选。新样本值可直接带入。难以兼顾泛化性和效率.
流形学习：可在保持原结构的条件下降维，假设和近似条件较多（只是假设流形存在），不能总适合数据的特点，计算复杂度较高，分类能力较弱（如短路断路问题），低维的维度数难确定，对噪声敏感

THANK YOU