第十章：降维与度量学习

主讲：连德富 特任教授 | 博士生导师

邮箱：liandefu@ustc.edu.cn

手机：13739227137

主页：http://staff.ustc.edu.cn/~liandefu

k近邻学习

- k近邻(k-Nearest Neighbor, kNN)学习是一种常用的监督学习方法:
  - 确定训练样本，以及某种距离度量。
  - 对于某个给定的测试样本，找到训练集中距离最近的k个样本
  - 对于分类问题使用“投票法”获得预测结果

    投票法：选择这k个样本中出现最多的类别标记作为预测结果。
  - 对于回归问题使用“平均法”获得预测结果。

    平均法：将这k个样本的实值输出标记的平均值作为预测结果。
  - 还可基于距离远近进行加权平均或加权投票，距离越近的样本权重越大。

“懒惰学习”与“急切学习”

- K近邻学习没有显式的训练过程，属于“懒惰学习”

“懒惰学习”(lazy learning):

此类学习技术在训练阶段仅仅是把样本保存起来，训练时间开销为零，待收到测试样本后再进行处理。

“急切学习”(eager learning):

在训练阶段就对样本进行学习处理的方法。

k近邻分类示意图

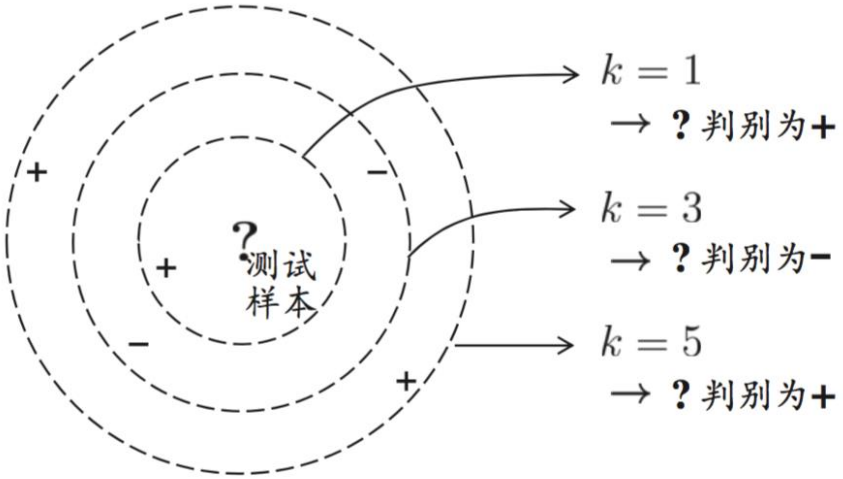


图 10.1 k 近邻分类器示意图. 虚线显示出等距线; 测试样本在 $k = 1$ 或 $k = 5$ 时被判别为正例, $k = 3$ 时被判别为反例.

- k近邻分类器中的k是一个重要参数，当k取不同值时，分类结果会有显著不同。
- 另一方面，若采用不同的距离计算方式，则找出的“近邻”可能有显著差别，从而也会导致分类结果有显著不同。

k近邻学习—1NN二分类错误率$P(err)$

- 暂且假设距离计算是“恰当”的，即能够恰当地找出k个近邻

- 我们来对“最近邻分类器”（1NN，即k=1）在二分类问题上的性能做一个简单的讨论。

- 给定测试样本$\boldsymbol{x}$，若其最近邻样本为$\boldsymbol{z}$，则最近邻出错的概率就是$\boldsymbol{x}$与$\boldsymbol{z}$类别标记不同的概率，即

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

k近邻学习—1NN二分类错误率$P(err)$

- 假设样本独立同分布，且对任意$\boldsymbol{x}$和任意小正整数δ，在$\boldsymbol{x}$附近δ距离范围内总能找到一个训练样本

- 换言之，对任意测试样本，总能在任意近的范围内找到$P(err) = 1 - \sum_{c \in \mathcal{Y}} P(c|\boldsymbol{x})P(c|\boldsymbol{z})$中的训练样本$\boldsymbol{z}$

- 令 $c^\star = \operatorname{argmax}_{c \in \mathcal{Y}} P(c|\boldsymbol{x})$表示贝叶斯最优分类器的结果，有

$$
\begin{aligned}
P(err) &= 1 - \sum_{c \in \mathcal{Y}} P(c|\boldsymbol{x})P(c|\boldsymbol{z}) \\
&\simeq 1 - \sum_{c \in \mathcal{Y}} P^2(c|\boldsymbol{x}) \\
&\leq 1 - P^2(c^\star|\boldsymbol{x}) \\
&= \big(1 + P(c^\star|\boldsymbol{x})\big)\big(1 - P(c^\star|\boldsymbol{x})\big) \\
&\leq 2\big(1 - P(c^\star|\boldsymbol{x})\big)
\end{aligned}
$$

最近邻分类虽简单，但它的泛化错误率不超过贝叶斯最优分类器错误率的两倍！

维数灾难 (curse of dimensionality)

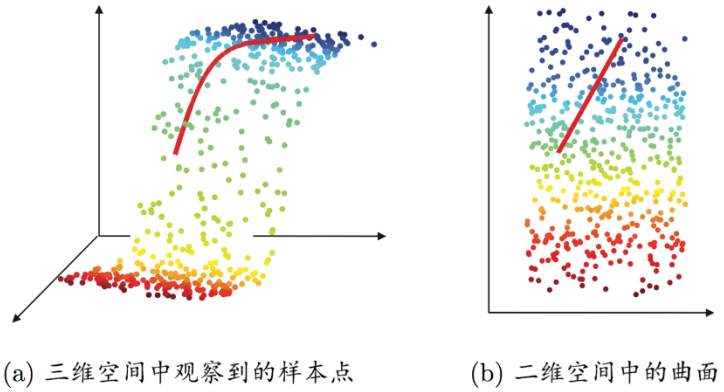
- 上述讨论基于一个重要的假设：任意测试样本 附近的任意小的距离范围内总能找到一个训练样本，即训练样本的采样密度足够大，或称为“密采样”。
- 然而，这个假设在现实任务中通常很难满足
  - 若属性维数为1，当$\delta = 0.001$，仅考虑单个属性，则仅需1000个样本点平均分布在归一化后的属性取值范围内，即可使得任意测试样本在其附近0.001距离范围内总能找到一个训练样本，此时最近邻分类器的错误率不超过贝叶斯最优分类器的错误率的两倍。若属性维数为20，若样本满足密采样条件，则至少需要1000^{20}个样本
  - 现实应用中属性维数经常成千上万，要满足密采样条件所需的样本数目是无法达到的天文数字
  - 许多学习方法都涉及距离计算，而高维空间会给距离计算带来很大的麻烦，例如当维数很高时甚至连计算内积都不再容易

在高维情形下出现的数据样本稀疏、距离计算困难等问题，是所有机器学习方法共同面临的严重障碍，被称为“维数灾难”

低维嵌入

- 缓解维数灾难的一个重要途径是降维(dimension reduction)
  - 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$\boldsymbol{D} \in \mathbb{R}^{m \times m}$，其第$i$行$j$列的元素$dist_{ij}$为样本$i$到$j$的距离

- 目标是获得样本在$d'(\ll d)$维空间中的欧氏距离等于原始空间中的距离，即$\left\|\boldsymbol{z}_i - \boldsymbol{z}_j\right\| = \mathrm{dist}_{ij}$

- 令$\boldsymbol{B} = \boldsymbol{Z}^{\mathrm{T}}\boldsymbol{Z}$，其中$\boldsymbol{B}$为降维后的内积矩阵，$b_{ij} = \boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{z}_j$，有

$$\mathrm{dist}_{ij}^2 = \|\boldsymbol{z}_i\|^2 + \left\|\boldsymbol{z}_j\right\|^2 - 2\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{z}_j$$

$$= b_{ii} + b_{jj} - 2b_{ij}$$

多维缩放（MDS）

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

$$\sum_i b_{ij} = \sum_i \boldsymbol{z}_i^{\mathrm{T}} \boldsymbol{z}_j = \boldsymbol{z}_i^{\mathrm{T}} \sum_i \boldsymbol{z}_j = \boldsymbol{z}_i^{\mathrm{T}} 0 = 0 \qquad \sum_j b_{ij} = 0$$

$$\sum_{j=1}^{m} \mathrm{dist}_{ij}^2 = \sum_{j=1}^{m} \left(b_{ii} + b_{jj} - 2b_{ij}\right) = m b_{ii} + \mathrm{tr}(\boldsymbol{B})$$

$$\sum_{i=1}^{m} \mathrm{dist}_{ij}^2 = \sum_{i=1}^{m} \left(b_{ii} + b_{jj} - 2b_{ij}\right) = m b_{jj} + \mathrm{tr}(\boldsymbol{B})$$

$$\sum_{i=1}^{m}\sum_{j=1}^{m} dist_{ij}^2 = \sum_{i} \left(m b_{ii} + \mathrm{trace}(\boldsymbol{B})\right) = 2\, m\, \mathrm{tr}(\boldsymbol{B})$$

多维缩放（MDS）

$$\mathrm{dist}_{i\cdot}^2 = \frac{1}{m}\sum_{j=1}^{m}\mathrm{dist}_{ij}^2 = \frac{1}{m}\left(mb_{ii} + \mathrm{tr}(\boldsymbol{B})\right) = b_{ii} + \frac{1}{m}\mathrm{tr}(\boldsymbol{B})$$

$$\mathrm{dist}_{\cdot j}^2 = \frac{1}{m}\sum_{i=1}^{m}\mathrm{dist}_{ij}^2 = \frac{1}{m}\left(mb_{jj} + \mathrm{tr}(\boldsymbol{B})\right) = b_{jj} + \frac{1}{m}\mathrm{tr}(\boldsymbol{B})$$

$$\mathrm{dist}_{\cdot\cdot}^2 = \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} dist_{ij}^2 \quad = \frac{1}{m^2}\,2\,m\,\mathrm{tr}(\boldsymbol{B}) \qquad = 2\frac{1}{m}\,\mathrm{tr}(\boldsymbol{B})$$

$$\mathrm{dist}_{ij}^2 = b_{ii} + b_{jj} - 2b_{ij}$$

$$\mathrm{dist}_{i\cdot}^2 + \mathrm{dist}_{\cdot j}^2 - \mathrm{dist}_{\cdot\cdot}^2 = b_{ii} + b_{jj} \quad = \mathrm{dist}_{ij}^2 + 2b_{ij}$$

$$b_{ij} = \frac{1}{2}\left(\mathrm{dist}_{i\cdot}^2 + \mathrm{dist}_{\cdot j}^2 - \mathrm{dist}_{\cdot\cdot}^2 - \mathrm{dist}_{ij}^2\right)$$

多维缩放（MDS）

$$b_{ij} = \frac{1}{2}\left(\mathrm{dist}_{i\cdot}^2 + \mathrm{dist}_{\cdot j}^2 - \mathrm{dist}_{\cdot\cdot}^2 - \mathrm{dist}_{ij}^2\right)$$

原空间中距离矩阵

$\boldsymbol{D} \in \mathbb{R}^{m\times m}$ ⇒ $\boldsymbol{B} = \boldsymbol{Z}^{\top}\boldsymbol{Z}$

降维后的内积矩阵

- 对矩阵$\boldsymbol{B}$做特征值分解(eigenvalue decomposition) $\boldsymbol{B} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{\top}$，其中$\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_d)$为特征值构成的对角矩阵，$\boldsymbol{V}$为特征向量矩阵
- $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$为特征向量矩阵，假定其中有$d^{\star}$个非零特征值， 它们构成对角矩阵$\boldsymbol{\Lambda}_{\star} = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_{d^{\star}})$。令$\boldsymbol{V}_{\star}$表示相应的特征矩阵，

$$\boldsymbol{Z} = \boldsymbol{\Lambda}_{\star}^{1/2}\boldsymbol{V}_{\star} \in \mathbb{R}^{d^{\star}\times m}$$

- 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$\tilde{\boldsymbol{\Lambda}} = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_{d'})$，令$\tilde{\boldsymbol{V}}$表示相应的特征向量矩阵，

$$\boldsymbol{Z} = \tilde{\boldsymbol{\Lambda}}^{1/2}\tilde{\boldsymbol{V}}$$

多维缩放（MDS）

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 $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m) \in \mathbb{R}^{d \times m}$，变换之后得到$d' \leq d$维空间中的样本

$$\boldsymbol{Z} = \boldsymbol{W}^{\top} \boldsymbol{X},$$

其中$\boldsymbol{W}$是变换矩阵，$\boldsymbol{Z}$是样本在新空间中的表达

- 变换矩阵$\boldsymbol{W}$可视为d'个d维属性向量。换言之，z_i是原属性向量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$\boldsymbol{W}$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$\boldsymbol{W}$施加不同的约束来实现。

主成分分析

- 对于正交属性空间中的样本点，如何用一个超平面对所有样本进行恰当的表达？

- 容易想到，若存在这样的超平面，那么它大概应具有这样的性质：

  最近重构性：样本点到这个超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能尽可能分开

- 基于最近重构性和最大可分性，能分别得到主成分分析的两种等价推导。

主成分分析—最近重构性

- 对样本进行中心化，即 $\sum_{i=1}^{m} \boldsymbol{x}_i = 0$

- 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$d' < d$，则样本点在低维坐标系中的投影是$\boldsymbol{z}_i = (z_{i1}, z_{i2}, \ldots, z_{id'})$，$z_{ij}$是$\boldsymbol{x}_i$在低维坐标下第$i$维的坐标。基于$\boldsymbol{z}_i$来重构$\boldsymbol{x}_i$，则会得到

$$\hat{\boldsymbol{x}}_i = \sum_{j=1}^{d'} z_{ij} \boldsymbol{w}_j = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{d'}) \boldsymbol{z}_i$$

$$= \boldsymbol{W} \boldsymbol{z}_i$$

$$\boldsymbol{W}^{\mathrm{T}} \boldsymbol{W} = \boldsymbol{I}_{d'}$$

主成分分析—最近重构性

- 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

$$\sum_{i=1}^{m}\|\hat{\boldsymbol{x}}_i-\boldsymbol{x}_i\|_2^2=\sum_{i=1}^{m}\|\boldsymbol{W}\boldsymbol{z}_i-\boldsymbol{x}_i\|_2^2=\sum_{i=1}^{m}\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{z}_i-2\sum_{i}^{m}\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}_i+\sum_{i}\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_i$$

$$\min_{\boldsymbol{W}}\sum_{i=1}^{m}\|\hat{\boldsymbol{x}}_i-\boldsymbol{x}_i\|_2^2=\sum_{i=1}^{m}\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{z}_i-2\sum_{i}^{m}\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}_i+\sum_{i}\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_i$$

$$\Leftrightarrow\ \max_{\boldsymbol{W}}\sum_{i=1}^{m}\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}_i=\sum_{i=1}^{m}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}_i)^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}_i=\sum_{i=1}^{m}\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}_i=\mathrm{tr}\left(\boldsymbol{W}^{\mathrm{T}}\sum_{i=1}^{m}(\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}})\,\boldsymbol{W}\right)$$

$$=\mathrm{tr}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W})$$

$$\boldsymbol{Z}=\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\quad\Rightarrow\quad\boldsymbol{z}_i=\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}_i$$

主成分分析—最近重构性

$$\min_{\boldsymbol{W}} \sum_{i=1}^{m} \|\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i\|_2^2 \quad \Leftrightarrow \quad \max_{\boldsymbol{W}} \operatorname{tr}(\boldsymbol{W}^{\top}\boldsymbol{X}\boldsymbol{X}^{\top}\boldsymbol{W}) \quad \text{s.t. } \boldsymbol{W}^{\top}\boldsymbol{W} = \boldsymbol{I}_{d'}$$

- 这就是主成分分析的优化目标

- 由于$\sum_{i=1}^{m} \boldsymbol{x}_i = 0$，$\frac{1}{m-1}\boldsymbol{X}\boldsymbol{X}^{\top} = \frac{1}{m-1}\sum_i \boldsymbol{x}_i\boldsymbol{x}_i^{\top}$ 是协方差矩阵

主成分分析—最大可分性

- 样本点$\boldsymbol{x}_i$在新空间中超平面上的投影是$\boldsymbol{z}_i = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}_i$，若所有样本点的投影能尽可能分开，则应该使得投影后样本点的方差最大化。

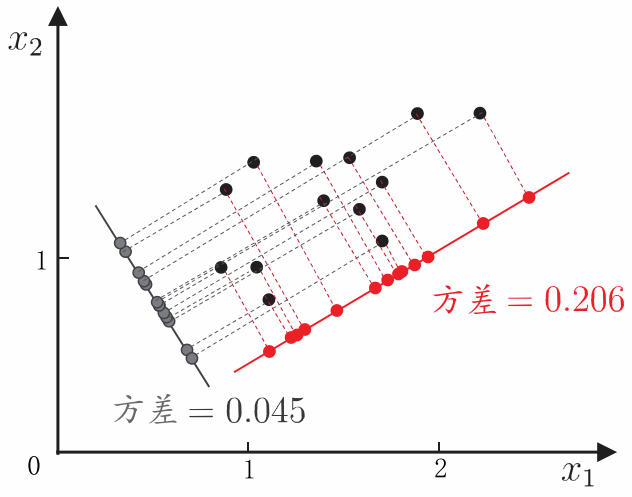
- 由于$\sum_{i=1}^{m}\boldsymbol{x}_i = 0$，$\sum_{i=1}^{m}\boldsymbol{z}_i = \boldsymbol{W}^{\mathrm{T}}\sum_{i=0}^{m}\boldsymbol{x}_i = 0$

- 投影后样本点的方差是$\frac{1}{m-1}\sum_{i=1}^{m}\boldsymbol{z}_i\boldsymbol{z}_i^{\mathrm{T}}$，优化目标为

$$\max_{\boldsymbol{W}} \operatorname{tr}\left(\sum_{i=1}^{m}\boldsymbol{z}_i\boldsymbol{z}_i^{\mathrm{T}}\right) = \operatorname{tr}\left(\sum_{i=1}^{m}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{W}\right)$$

$$\Leftrightarrow \quad \max_{\boldsymbol{W}} \operatorname{tr}\left(\boldsymbol{W}^{\mathrm{T}}\left(\sum_{i=1}^{m}\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}\right)\boldsymbol{W}\right)$$

$$\Leftrightarrow \quad \max_{\boldsymbol{W}} \operatorname{tr}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W})$$



主成分分析—求解

$$\max_{\boldsymbol{W}} \operatorname{tr}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}) \quad \text{s.t. } \boldsymbol{W}^{\mathrm{T}}\boldsymbol{W} = \boldsymbol{I}_{d'}$$

- 使用拉格朗日乘子法可得

$$\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W} = \boldsymbol{\Lambda}\boldsymbol{W}$$

只需对协方差矩阵$\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$进行特征值分解，并将求得的特征值排序：$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$，再取前$d'$个特征值对应的特征向量构成$\boldsymbol{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{d'})$

这就是主成分分析的解。

主成分分析—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

主成分分析—维度选择

- 降维后低维空间的维数d'通常是由用户事先指定，或通过在d'值不同的低维空间中对k近邻分类器（或其它开销较小的学习器）进行交叉验证来选取较好的d'值。
- 对PCA，还可从重构的角度设置一个重构阈值，例如$t = 95\%$，然后选取使下式成立的最小d'值：

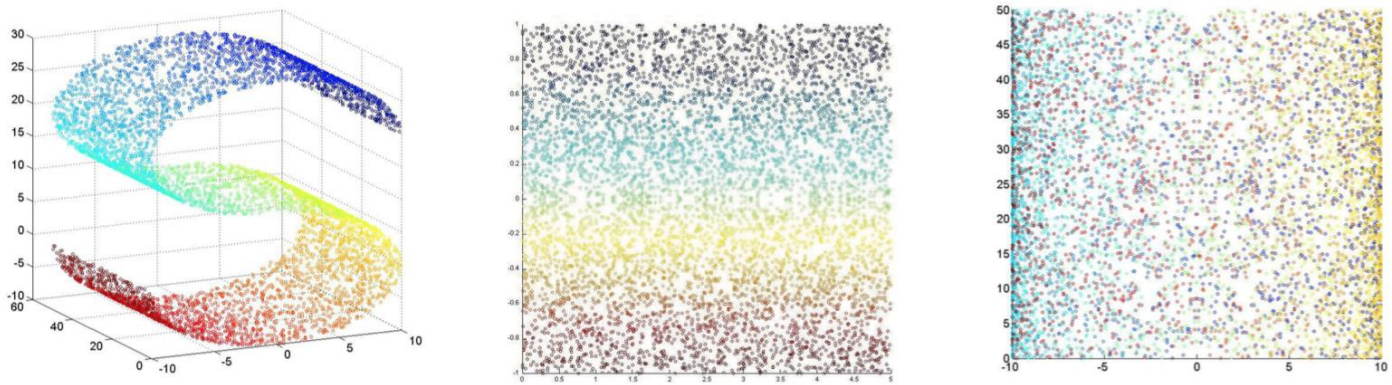
$$\frac{\sum_{i=1}^{d'} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \geq t$$

- PCA仅需保留$\boldsymbol{W}$与样本的均值向量即可通过简单的向量减法和矩阵-向量乘法将新样本投影至低维空间中

- 降维虽会导致信息的损失，但一方面舍弃这些信息后能使得样本的采样密度增大，另一方面，当数据受到噪声影响时，最小的特征值所对应的特征向量往往与噪声有关，舍弃可以起到去噪效果

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．

核化主成分分析 (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)。

- 假定我们将在高维特征空间中把数据$\phi(\boldsymbol{x})$投影到由$\boldsymbol{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_d)$确定的超平面上，即PCA欲求解

$$\left(\sum_{i=1}^{m}\phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_i)^{\mathrm{T}}\right)\boldsymbol{w}_j = \lambda_j \boldsymbol{w}_j \quad \Rightarrow \quad \sum_{l=1}^{m} K_{il}\alpha_l^j = \lambda_j \alpha_i^j \quad \Rightarrow \quad \boldsymbol{K}\boldsymbol{\alpha}^j = \lambda_j \boldsymbol{\alpha}^j$$

$$\phi(\boldsymbol{x}_i)^{\mathrm{T}}\left(\sum_{l=1}^{m}\phi(\boldsymbol{x}_l)\phi(\boldsymbol{x}_l)^{\mathrm{T}}\right)\boldsymbol{w}_j = \lambda_j \phi(\boldsymbol{x}_i)^{\mathrm{T}}\boldsymbol{w}_j \qquad \sum_{l=1}^{m} K_{il}\phi(\boldsymbol{x}_l)^{\mathrm{T}}\boldsymbol{w}_j = \lambda_j \phi(\boldsymbol{x}_i)^{\mathrm{T}}\boldsymbol{w}_j$$

$$\boldsymbol{w}_j = \frac{1}{\lambda_j}\left(\sum_{i=1}^{m}\phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_i)^{\mathrm{T}}\right)\boldsymbol{w}_j = \sum_{i=1}^{m}\phi(\boldsymbol{x}_i)\frac{\phi(\boldsymbol{x}_i)^{\mathrm{T}}\boldsymbol{w}_j}{\lambda_j} = \sum_{i=1}^{m}\phi(\boldsymbol{x}_i)\alpha_i^j$$

核化主成分分析 (KPCA)

KPCA欲求解 $\boldsymbol{K}\boldsymbol{\alpha}^j = \lambda_j \boldsymbol{\alpha}^j$

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

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

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

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

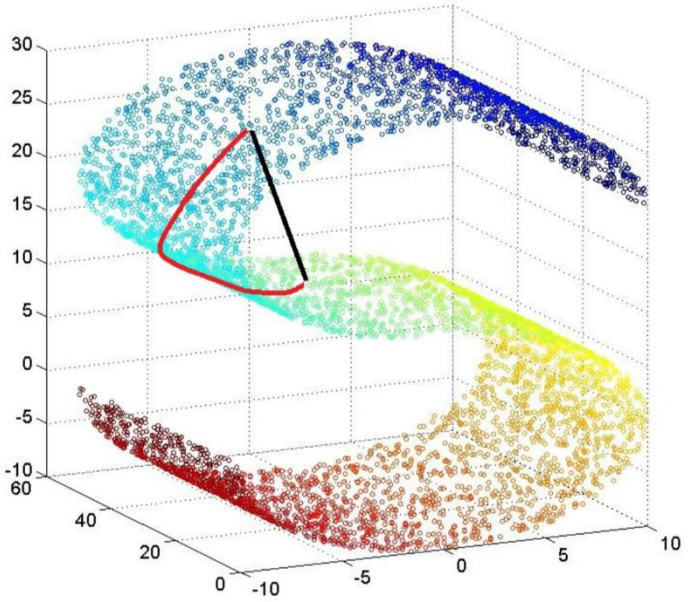
- 流形学习(manifold learning)是一类借鉴了拓扑流形概念的降维方法。“流形”是在局部与欧氏空间同胚的空间，换言之，它在局部具有欧氏空间的性质，能用欧氏距离来进行距离计算。

- 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。

等度量映射(Isometric Mapping，Isomap)

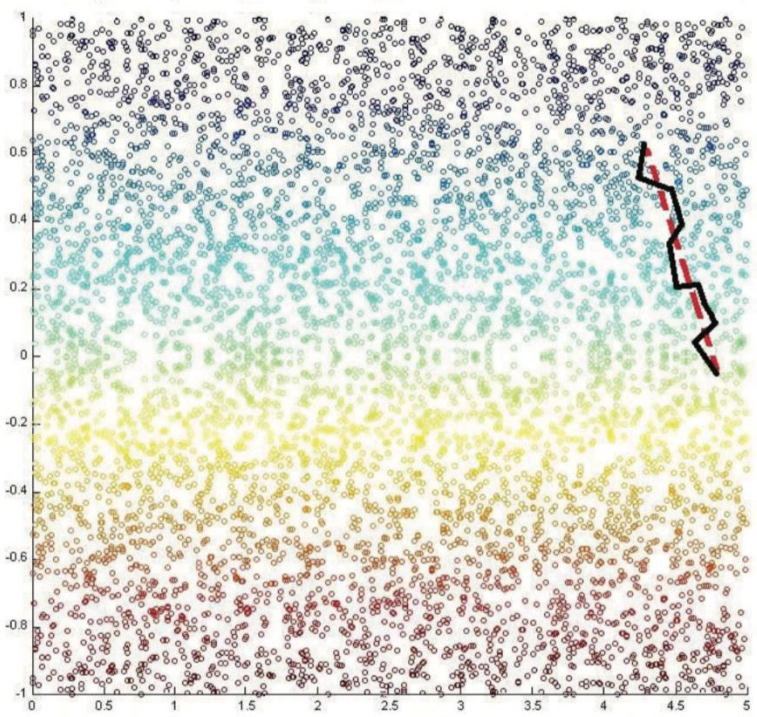
- 低维流形嵌入到高维空间之后，直接在高维空间中计算直线距离具有误导性，因为高维空间中的直线距离在低维嵌入流形上不可达。而低维嵌入流形上两点间的本真距离是“测地线”(geodesic)距离。



(a) 测地线距离与高维直线距离

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，图中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

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:

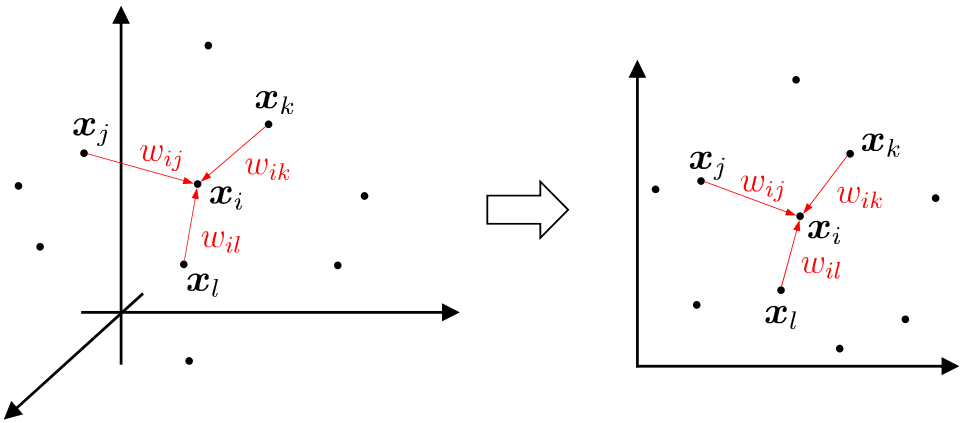
1: **for** $i = 1, 2, \ldots, m$ **do**
2: 确定 $\boldsymbol{x}_i$ 的 k 近邻;
3: $\boldsymbol{x}_i$ 与 k 近邻点之间的距离设置为欧氏距离, 与其他点的距离设置为无穷大;
4: **end for**
5: 调用最短路径算法计算任意两样本点之间的距离 $\text{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j)$;
6: 将 $\text{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ 作为 MDS 算法的输入;
7: **return**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

局部线性嵌入 (Locally Linear Embedding，LLE)

- 局部线性嵌入试图保持邻域内的线性关系，并使得该线性关系在降维后的空间中继续保持。



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

局部线性嵌入 (Locally Linear Embedding，LLE)

- LLE先为每个样本$\boldsymbol{x}_i$找到其近邻下标集合Q_i，然后计算出基于Q_i的中的样本点对$\boldsymbol{x}_i$进行线性重构的系数$\boldsymbol{w}_i$

$$\min_{\boldsymbol{w}_i} \left\| \boldsymbol{x}_i - \sum_{j \in Q_i} w_{ij} \boldsymbol{x}_j \right\|_2^2 \quad \text{s.t.} \sum_{j \in Q_i} w_{ij} = 1 \quad \Rightarrow \quad \boldsymbol{w}_i^{\mathrm{T}} \mathbf{1} = 1$$

$$\Rightarrow \quad \min_{\boldsymbol{w}_i} \left\| \boldsymbol{x}_i \mathbf{1}^{\mathrm{T}} \boldsymbol{w}_i - \boldsymbol{X}_i^{\mathrm{T}} \boldsymbol{w}_i \right\|_2^2 \qquad \boldsymbol{w}_i = \left[w_{i_1}, w_{i_1}, \ldots, w_{i_{|Q_i|}} \right]^{\mathrm{T}}$$

$$\Rightarrow \quad \min_{\boldsymbol{w}_i} \left\| \left(\boldsymbol{x}_i \mathbf{1}^{\mathrm{T}} - \boldsymbol{X}_i^{\mathrm{T}} \right) \boldsymbol{w}_i \right\|_2^2 \qquad \boldsymbol{X}_i = \left[\boldsymbol{x}_{i_1}, \boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_{|Q_i|}} \right]^{\mathrm{T}}$$

- 引入拉格朗日乘子，求梯度，并令其梯度等于0可得

$$\left(\boldsymbol{x}_i \mathbf{1}^{\mathrm{T}} - \boldsymbol{X}_i^{\mathrm{T}} \right)^{\mathrm{T}} \left(\boldsymbol{x}_i \mathbf{1}^{\mathrm{T}} - \boldsymbol{X}_i^{\mathrm{T}} \right) \boldsymbol{w}_i = \lambda \mathbf{1}$$

局部线性嵌入 (Locally Linear Embedding，LLE)

$$\left(\boldsymbol{x}_i\mathbf{1}^{\mathsf{T}}-\boldsymbol{X}_i^{\mathsf{T}}\right)^{\mathsf{T}}\left(\boldsymbol{x}_i\mathbf{1}^{\mathsf{T}}-\boldsymbol{X}_i^{\mathsf{T}}\right)\boldsymbol{w}_i=\lambda\mathbf{1} \quad \Rightarrow \quad \boldsymbol{C}\boldsymbol{w}_i=\lambda\mathbf{1}$$

令$C_{jk}=\left(\boldsymbol{x}_i-\boldsymbol{x}_j\right)^{\mathsf{T}}\left(\boldsymbol{x}_i-\boldsymbol{x}_j\right)$ $\Rightarrow$ $\boldsymbol{w}_i=\lambda\boldsymbol{C}^{-1}\mathbf{1}$

- w_{ij}的最终闭形式解为 $$w_{ij}=\frac{\sum_{k\in Q_i}C_{jk}^{-1}}{\sum_{l,s\in Q_i}C_{ls}^{-1}}$$

局部线性嵌入 (Locally Linear Embedding，LLE)

- LLE在低维空间中保持$\boldsymbol{w}_i$不变，于是$\boldsymbol{x}_i$对应的低维空间坐标$\boldsymbol{z}_i$可通过下式求解：

$$\min_{\boldsymbol{z}_1,\boldsymbol{z}_2,\ldots,\boldsymbol{z}_m} \sum_{i=1}^{m} \left\| \boldsymbol{z}_i - \sum_{j \in Q_i} w_{ij} \boldsymbol{z}_j \right\|_2^2 = \sum_{i=1}^{m} \|\boldsymbol{z}_i - \boldsymbol{Z}^{\top} \boldsymbol{W}_i\|_2^2$$

$$(\boldsymbol{W})_{ij} = w_{ij}$$

$$= \|\boldsymbol{Z}^{\top} - \boldsymbol{Z}^{\top} \boldsymbol{W}\|_F^2$$

$$\min_{\boldsymbol{W}} \|\boldsymbol{Z}^{\top} - \boldsymbol{Z}^{\top} \boldsymbol{W}\|_F^2 = \mathrm{tr}(\boldsymbol{Z}^{\top} \underbrace{(\boldsymbol{I} - \boldsymbol{W})(\boldsymbol{I} - \boldsymbol{W})^{\top}}_{\boldsymbol{M}} Z)$$

$$\text{s.t.}\ \boldsymbol{Z}^{\top} \boldsymbol{Z} = \boldsymbol{I}$$

该优化可以通过特征值分解得以求解

局部线性嵌入 (Locally Linear Embedding，LLE)

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

度量学习—研究动机

- 在机器学习中，对高维数据进行降维的主要目的是希望找到一个合适的低维空间，在此空间中进行学习能比原始空间性能更好。事实上，每个空间对应了在样本属性上定义的一个距离度量，而寻找合适的空间，实质上就是在寻找一个合适的距离度量。那么，为何不直接尝试“学习”出一个合适的距离度量呢？

度量学习

- 欲对距离度量进行学习，必须有一个便于学习的距离度量表达形式。对两个d维样本$\boldsymbol{x}_i$和$\boldsymbol{x}_j$，它们之间的平方欧氏距离可写为

$$dist_{ed}^2(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 = dist_{ij,1}^2 + dist_{ij,2}^2 + \cdots + dist_{ij,d}^2$$

- 其中$dist_{ij,k}^2$表示 $\boldsymbol{x}_i$和$\boldsymbol{x}_j$在第k维上的距离。若假定不同属性的重要性不同，则可引入属性权重$\boldsymbol{w}$，得到

$$dist_{ed}^2(\boldsymbol{x}_i, \boldsymbol{x}_j) = w_1 \cdot dist_{ij,1}^2 + w_2 \cdot dist_{ij,2}^2 + \cdots + w_d \cdot dist_{ij,d}^2$$

$$= (\boldsymbol{x}_i - \boldsymbol{x}_j)^{\mathsf{T}} \boldsymbol{W} (\boldsymbol{x}_i - \boldsymbol{x}_j)$$

- 其中$w_i \geq 0, \boldsymbol{W} = \mathrm{diag}(\boldsymbol{w})$是一个对角矩阵，$(\boldsymbol{W})_{ii} = w_{ii}$，可通过学习确定。

度量学习

- $\boldsymbol{W}$的非对角元素均为零，意味着坐标轴正交，即属性之间无关
- 但现实问题中往往不是这样，例如考虑西瓜的“重量”和“体积”这两个属性，它们显然是正相关的，其对应的坐标轴不再正交。
- 为此将$\boldsymbol{W}$替换为一个普通的半正定对称矩阵$\boldsymbol{M}$，于是就得到了马氏距离(Mahalanobis distance)

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

- 其中$\boldsymbol{M}$亦称“度量矩阵”，而度量学习则是对$\boldsymbol{M}$进行学习。注意到为了保持距离非负且对称，$\boldsymbol{M}$必须是（半）正定对称矩阵，即必有正交基$\boldsymbol{P}$使得$\boldsymbol{M}$能写为$\boldsymbol{M} = \boldsymbol{P}\boldsymbol{P}^{\mathrm{T}}$

- 对$\boldsymbol{M}$进行学习当然要设置一个目标。假定我们是希望提高近邻分类器的性能，则可将$\boldsymbol{M}$直接嵌入到近邻分类器的评价指标中去，通过优化该性能指标相应地求得$\boldsymbol{M}$

近邻成分分析(Neighbourhood Component Analysis, NCA)

- 近邻成分分析在进行判别时通常使用多数投票法，邻域中的每个样本投1票，邻域外的样本投0票。不妨将其替换为概率投票法。对于任意样本$\boldsymbol{x}_j$，它对$\boldsymbol{x}_i$分类结果影响的概率为

$$p_{ij} = \frac{\exp\left(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_M^2\right)}{\sum_l \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_l\|_M^2)}$$

- 当$i = j$时，p_{ij}最大。显然，$\boldsymbol{x}_i$对$\boldsymbol{x}_j$的影响随着它们之间距离的增大而减小。

近邻成分分析(Neighbourhood Component Analysis, NCA)

- 若以留一法(LOO)正确率的最大化为目标，则可计算$\boldsymbol{x}_i$的留一法正确率，即它被自身之外的所有样本正确分类的概率为

$$p_i = \sum_{j \in \Omega_i} p_{ij} \qquad p_{ij} = \frac{\exp\left(-\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_M^2\right)}{\sum_l \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_l\|_M^2)}$$

其中Ω_i表示与$\boldsymbol{x}_i$属于相同类别的样本的下标集合

- 整个样本集上的留一法正确率为 $\sum_{i=1}^{m} p_i = \sum_{i}^{m} \sum_{j \in \Omega_i} p_{ij}$

- 由于$\boldsymbol{M} = \boldsymbol{P}\boldsymbol{P}^{\mathrm{T}}$，则NCA的优化目标为

$$\left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)^{\mathrm{T}} \boldsymbol{M} \left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)$$

$$\Rightarrow \left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)^{\mathrm{T}} \boldsymbol{P}\boldsymbol{P}^{\mathrm{T}} \left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)$$

$$\min_{\boldsymbol{P}} 1 - \sum_{i=1}^{m} \sum_{j \in \Omega_i} \frac{\exp\left(-\left\|\boldsymbol{P}^{\mathrm{T}}\boldsymbol{x}_i - \boldsymbol{P}^{\mathrm{T}}\boldsymbol{x}_j\right\|_2^2\right)}{\sum_l \exp(-\|\boldsymbol{P}^{\mathrm{T}}\boldsymbol{x}_i - \boldsymbol{P}^{\mathrm{T}}\boldsymbol{x}_l\|_2^2)}$$

求解即可得到最大化近邻分类器LOO正确率的距离度量矩阵$\boldsymbol{M}$

度量学习

- 实际上，我们不仅能把错误率这样的监督学习目标作为度量学习的优化目标，还能在度量学习中引入领域知识。
- 若已知某些样本相似、某些样本不相似，则可定义“必连”(must-link) 约束集合$\mathcal{M}$ 与“勿连”(cannot-link)约束集合$\mathcal{C}$：
  - $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{M}$，表示$\boldsymbol{x}_i$与$\boldsymbol{x}_j$相似
  - $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{C}$，表示$\boldsymbol{x}_i$与$\boldsymbol{x}_j$不相似

- 显然，我们希望相似的样本之间距离较小，不相似的样本之间距离较大，可通过求解下面这个凸优化问题获得适当的度量矩阵$\boldsymbol{M}$

$$\min_{\mathrm{M}} \sum_{(x_i,x_j)\in\mathcal{M}} \left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_{\boldsymbol{M}}^2 \quad \text{s.t.} \sum_{(x_i,x_j)\in\mathcal{C}} \left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_{\boldsymbol{M}}^2 \geq 1 \text{ and } \boldsymbol{M} \succcurlyeq 0$$

$\boldsymbol{M}$必须是半正定的

- 上式要求在不相似样本间的距离不小于1的前提下，使相似样本间的距离尽可能小

度量学习

- 不同的度量学习方法针对不同目标获得“好”的半正定对称距离度量矩阵$\boldsymbol{M}$，若$\boldsymbol{M}$是一个低秩矩阵，则通过对$\boldsymbol{M}$进行特征值分解，总能找到一组正交基，其正交基数目为矩阵$\boldsymbol{M}$的秩$\mathrm{rank}(\boldsymbol{M})$，小于原属性数$d$。
- 于是，度量学习学得的结果可衍生出一个降维矩阵$\boldsymbol{P} \in \mathbb{R}^{d \times \mathrm{rank}(\boldsymbol{M})}$

习题

- 记$\mathrm{err}^{\star}(\boldsymbol{x}) = 1 - \max_{c \in \mathcal{Y}} P(c|\boldsymbol{x})$，$\mathrm{err}(\boldsymbol{x}) = 1 - \sum_c P(c|\boldsymbol{x})P(c|\boldsymbol{z})$，其中$\boldsymbol{z}$为$\boldsymbol{x}$的最近邻，试证明在样本无穷多时

$$\mathrm{err}^{\star}(\boldsymbol{x}) \leq \mathrm{err}(\boldsymbol{x}) \leq \mathrm{err}^{\star}(\boldsymbol{x})\left(2 - \frac{|\mathcal{Y}|}{|\mathcal{Y}|-1} \times \mathrm{err}^{\star}(\boldsymbol{x})\right)$$

  提示：柯西-施瓦兹不等式$(\sum_i a_i)^2 \leq n\left(\sum_i a_i^2\right)$

- 10.4