第十章 降维与度量学习

王博：自动化（人工智能）学院
wangbo@hdu.edu.cn

目录

- k近邻学习
- MDS算法
- 主成分分析
- 核化线性降维
- 流形学习
- 度量学习

目录

k近邻学习

k近邻学习的工作机制

- 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。
  - 找到训练集中距离最近的k个样本，分类问题使用“投票法”获得预测结果，对于回归问题使用“平均法”获得预测结果。

K近邻学习没有显式的训练过程，训练时间开销为零，待收到测试样本后再进行处理。

K近邻算法

算法 3.1（k 近邻法）

输入：训练数据集

$$T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$$

其中，$x_i\in\mathcal{X}\subseteq\mathbf{R}^n$ 为实例的特征向量，$y_i\in\mathcal{Y}=\{c_1,c_2,\cdots,c_K\}$ 为实例的类别，$i=1,2,\cdots,N$；实例特征向量 x；

输出：实例 x 所属的类 y.

（1）根据给定的距离度量，在训练集 T 中找出与 x 最邻近的 k 个点，涵盖这 k 个点的 x 的邻域记作 $N_k(x)$；

（2）在 $N_k(x)$ 中根据分类决策规则（如多数表决）决定 x 的类别 y：

$$y=\arg\max_{c_j}\sum_{x_i\in N_k(x)}I(y_i=c_j),\quad i=1,2,\cdots,N;\quad j=1,2,\cdots,K$$

式 (3.1) 中，I 为指示函数，即当 $y_i=c_j$ 时 I 为 1，否则 I 为 0.

k近邻关键点

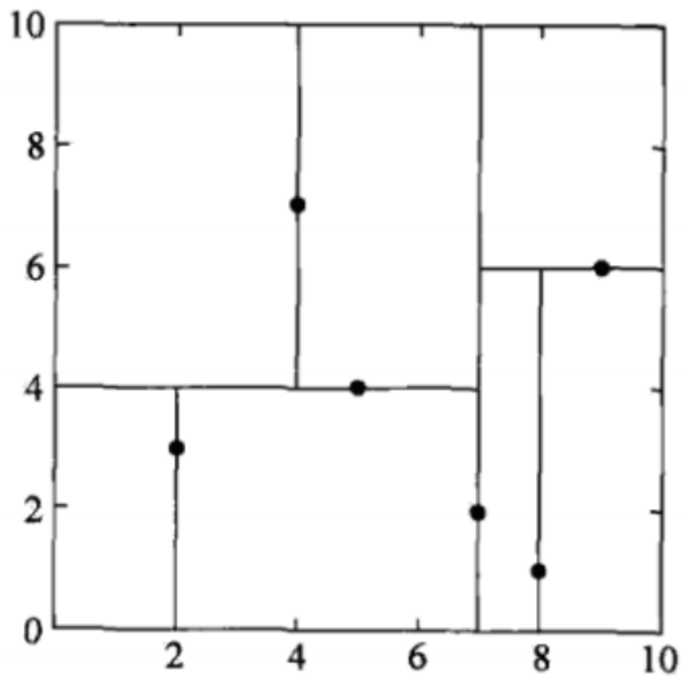
- k重要参数------交叉验证法
- 距离的度量
- 样本的选择
- 确定邻居涉及的计算

关键问题是如何对训练数据进行快速k近邻搜索

k近邻实现—kd树

Kd（ k-dimensional ）树存储训练数据

$$T=\{(2,3)^{\mathrm{T}},(5,4)^{\mathrm{T}},(9,6)^{\mathrm{T}},(4,7)^{\mathrm{T}},(8,1)^{\mathrm{T}},(7,2)^{\mathrm{T}}\}$$



k近邻学习

分析1NN二分类错误率 $P(err)$

- 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 $\boldsymbol{z}$ 与 $\boldsymbol{x}$ 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

分析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z。

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

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

- 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！

目录

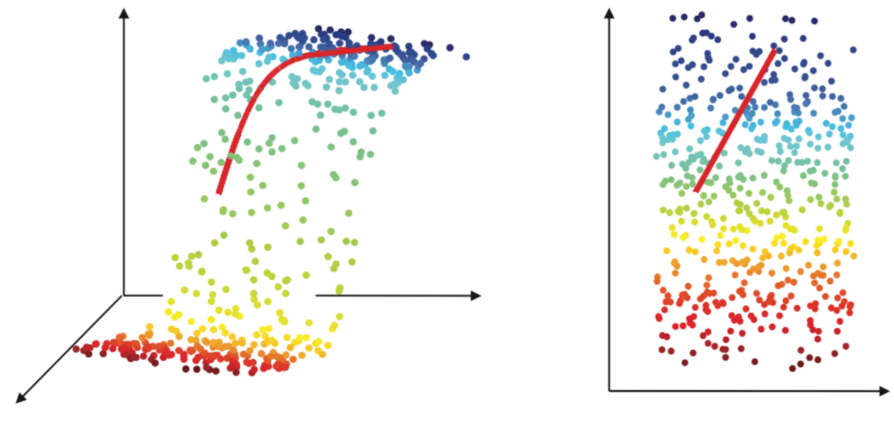
- k近邻学习
- MDS算法
- 主成分分析
- 核化线性降维
- 流形学习
- 度量学习

MDS算法

维数灾难 (curse of dimensionality)

- 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

MDS算法

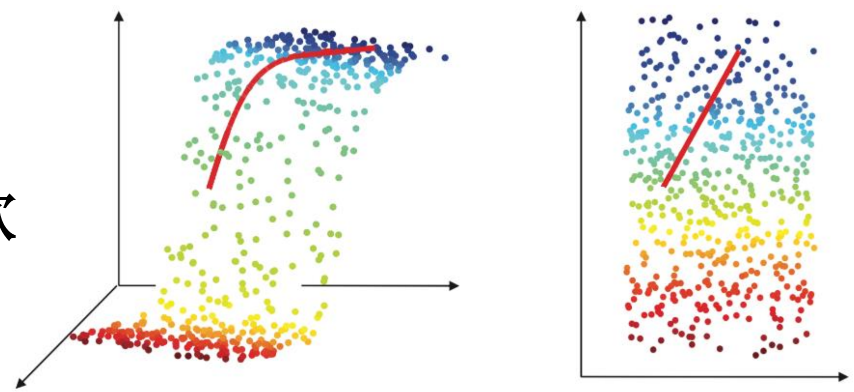
- 若要求原始空间中样本之间的距离在低维空间中得以保持，即得到“多维缩放”(Multiple Dimensional Scaling, MDS)：

- 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 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

MDS算法

- 为便于讨论，令降维后的样本 Z 被中心化，即 $\sum_{i=1}^{m} \boldsymbol{z}_i = \mathbf{0}$ 。显然，矩阵 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 $\mathrm{tr}(\cdot)$ 表示矩阵的迹(trace)，$\mathrm{tr}(\mathbf{B}) = \sum_{i=1}^{m} ||\boldsymbol{z}_i||^2$ 。令

$$\sum_{i=1}^{m} dist_{i\cdot}^2 = \mathrm{tr}(\mathbf{B}) + m b_{ij}, \quad \sum_{j=1}^{m} dist_{\cdot j}^2 = \mathrm{tr}(\mathbf{B}) + m b_{ij}, \quad \sum_{i=1}^{m}\sum_{j=1}^{m} dist_{\cdot\cdot}^2 = \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} dist_{ij}^2,$$

- 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：

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

MDS算法

- 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

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

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

目录

- k近邻学习
- MDS算法
- 主成分分析
- 核化线性降维
- 流形学习
- 度量学习

线性降维方法

- 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，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。

- 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

主成分分析(Principal Component Analysis, 简称PCA)

- 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？

- 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，$\sum_i x_i = 0$ ，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 $\boldsymbol{z}_i = (z_{i1}; z_{i2}; \ldots; z_{id'})$，$z_{ij} = \boldsymbol{w}_j^{\mathrm{T}} \boldsymbol{x}_i$ 是 $\boldsymbol{x}_i$ 在低维坐标下第 j 维的坐标，若基于 $\boldsymbol{z}_i$ 来重构 $\boldsymbol{x}_i$，则会得到

$$\hat{\boldsymbol{x}}_i = \sum_{j=1}^{d'} z_{ij} \boldsymbol{w}_j.$$

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

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

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

- 根据最近重构性应最小化上式。考虑到 $\boldsymbol{w}_j$是标准正交基，$\sum_i \boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}$是协方差矩阵，有

$$\min_{\mathbf{W}}\quad -\operatorname{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W})$$
$$\text{s.t.}\quad \mathbf{W}^{\mathrm{T}}\mathbf{W}=\mathbf{I}.$$

这就是主成分分析的优化目标。

主成分分析

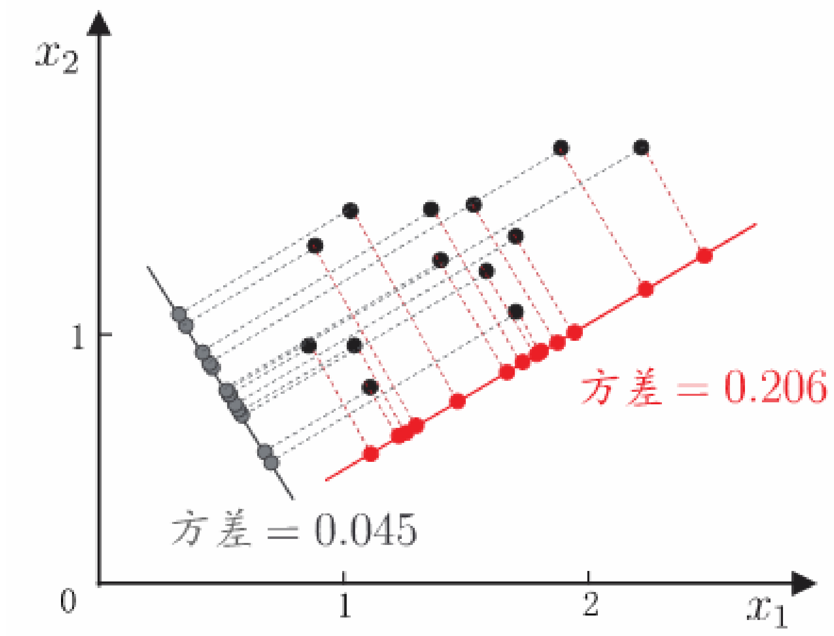
最大可分性

- 样本点 $\boldsymbol{x}_i$ 在新空间中超平面上的投影是 $\mathbf{W}^{\mathrm{T}}\boldsymbol{x}_i$，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。若投影后样本点的方差是 $\sum_i \mathbf{W}^{\mathrm{T}}\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}\mathbf{W}$，于是优化目标可写为

$$\max_{\mathbf{W}} \quad \mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W})$$
$$\mathrm{s.t.} \quad \mathbf{W}^{\mathrm{T}}\mathbf{W}=\mathbf{I}.$$

显然与下式等价：

$$\min_{\mathbf{W}} \quad -\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W})$$
$$\mathrm{s.t.} \quad \mathbf{W}^{\mathrm{T}}\mathbf{W}=\mathbf{I}.$$



主成分分析

PCA的求解

$$\max_{\mathbf{W}} \quad \mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W}) \qquad\qquad \min_{\mathbf{W}} \quad -\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W})$$

$$\text{s.t.} \quad \mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}. \qquad\qquad \text{s.t.} \quad \mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}.$$

- 拉格朗日函数为（包含矩阵内积运算）

$$\begin{aligned} L(\mathbf{W}, \Theta) &= -\mathrm{tr}\left(\mathbf{W}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W}\right) + \langle \Theta, \mathbf{W}^{\mathrm{T}}\mathbf{W} - \mathbf{I} \rangle \\ &= -\mathrm{tr}\left(\mathbf{W}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W}\right) + \mathrm{tr}\left(\Theta^{\mathrm{T}}(\mathbf{W}^{\mathrm{T}}\mathbf{W} - \mathbf{I})\right) \end{aligned}$$

- 矩阵求导 $\dfrac{\partial(\mathbf{x}^T\mathbf{A}\mathbf{x})}{\partial \mathbf{x}} = (\mathbf{A} + \mathbf{A}^T)\mathbf{x}$ $\qquad \dfrac{\partial(\boldsymbol{a}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{b})}{\partial \boldsymbol{X}} = \boldsymbol{X}\boldsymbol{b}\boldsymbol{a}^T + \boldsymbol{X}\boldsymbol{a}\boldsymbol{b}^T$

主成分分析

PCA的求解

$$L(\mathbf{W},\Theta) = -\operatorname{tr}\left(\mathbf{W}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W}\right) + \langle\Theta, \mathbf{W}^{\mathrm{T}}\mathbf{W}-\mathbf{I}\rangle$$
$$= -\operatorname{tr}\left(\mathbf{W}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W}\right) + \operatorname{tr}\left(\Theta^{\mathrm{T}}(\mathbf{W}^{\mathrm{T}}\mathbf{W}-\mathbf{I})\right)$$

- 对优化式使用拉格朗日乘子法可得

$$\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W} = \lambda\mathbf{W}.$$

只需对协方差矩阵 $\mathbf{X}\mathbf{X}^{\mathrm{T}}$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 $\mathbf{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{d'})$，这就是主成分分析的解。

主成分分析

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

主成分分析

- 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 $\mathbf{W}$ 值：

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

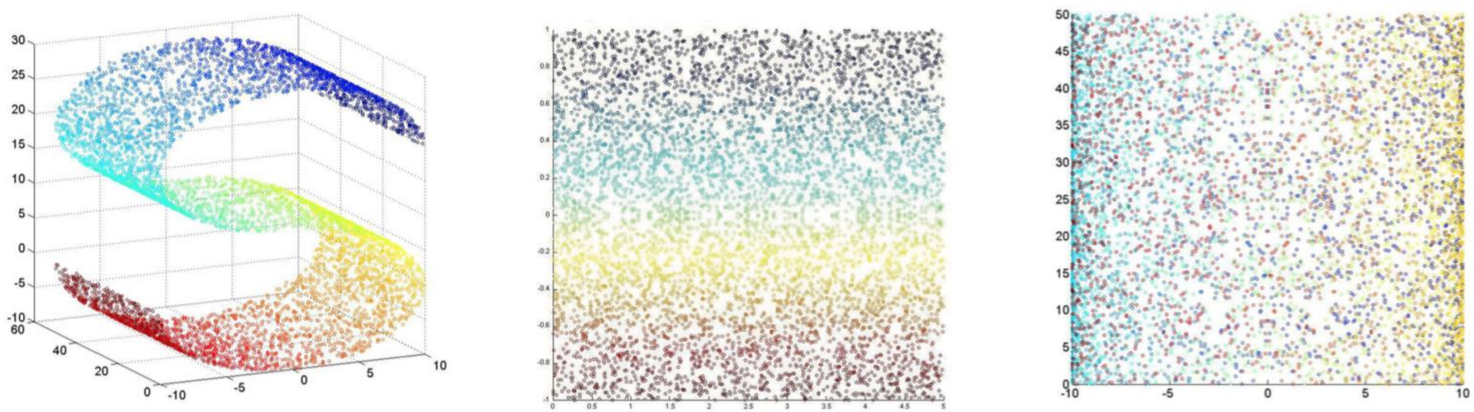
- 降维虽然会导致信息的损失，但一方面舍弃这些信息后能使得样本的采样密度增大，另一方面，当数据受到噪声影响时，最小的特征值所对应的特征向量往往与噪声有关，舍弃可以起到去噪效果。

目录

- k近邻学习
- MDS算法
- 主成分分析
- **核化线性降维**
- 流形学习
- 度量学习

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。
- 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）是基于核技巧对线性降维方法

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

$$\left(\sum_{i=1}^{m} x_i x_i^T\right) w_j = \lambda_j w_j \Rightarrow \left(\sum_{i=1}^{m} \phi(x_i)\phi(x_i)^T\right) w_j = \lambda_j w_j$$

- 可求得 $W = (w_1, \cdots, w_{d'})$
- 降维后 $z_i = W^T \phi(x_i)$

难点：ϕ 难以获得？？？？？？

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

核化线性降维

核化主成分分析（Kernelized PCA，简称KPCA）

- 考虑

$$w_j = \phi(X)\alpha_j = (\phi(x_1),\cdots\phi(x_m))\alpha_j,\ \left(\sum_{i=1}^{m}\phi(x_i)\phi(x_i)^T\right)w_j = \lambda_j w_j$$

- 得到

$$\phi(X)\phi(X)^T\phi(X)\alpha_j = \lambda_j\phi(X)\alpha_j$$

- 通常不清楚 ϕ 的具体形式，于是引入核函数

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

- 进一步

$$KK\alpha_j = \lambda_j K\alpha_j, K_{ij} = \kappa(x_i, x_j)$$

核化线性降维

核化主成分分析（Kernelized PCA，简称KPCA）

- 正交分解可得

$$A=(\alpha_1,\cdots,\alpha_{d'})$$

- 核化后的投影坐标zi

$$W=(w_1,\cdots,w_{d'})=\phi(X)(\alpha_1,\cdots,\alpha_{d'})=\phi(X)A$$

$$z_i=W^T\phi(x_i),$$

$$z_i=A^T\phi(X)^T\phi(x_i)=A^T\begin{bmatrix}\phi(x_1)^T\\ \vdots\\ \phi(x_N)^T\end{bmatrix}\phi(x_i)=A^T\begin{bmatrix}k(1,i)\\ \vdots\\ k(N,i)\end{bmatrix}$$

KPCA需对所有样本求和，它的计算开销较大。

目录

- k近邻学习
- 低维嵌入
- 主成分分析
- 核化线性降维
- **流形学习**
- 度量学习

流形学习

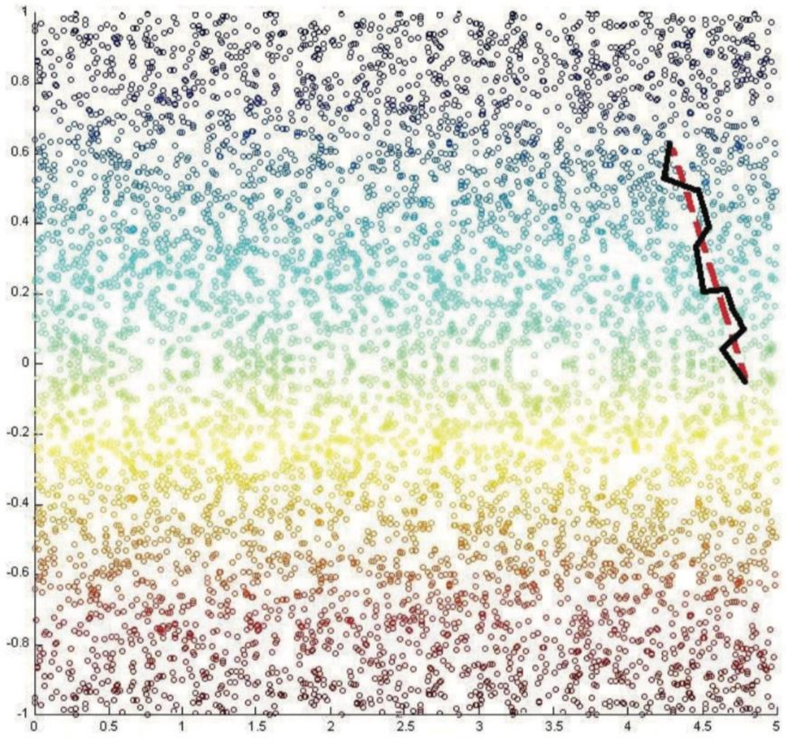
什么是流形?

- 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。

等度量映射

等度量映射(Isometric Mapping，Isomap)

- 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

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 $\mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j)$;
6: 将 $\mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ 作为 MDS 算法的输入;
7: **return** MDS 算法的输出

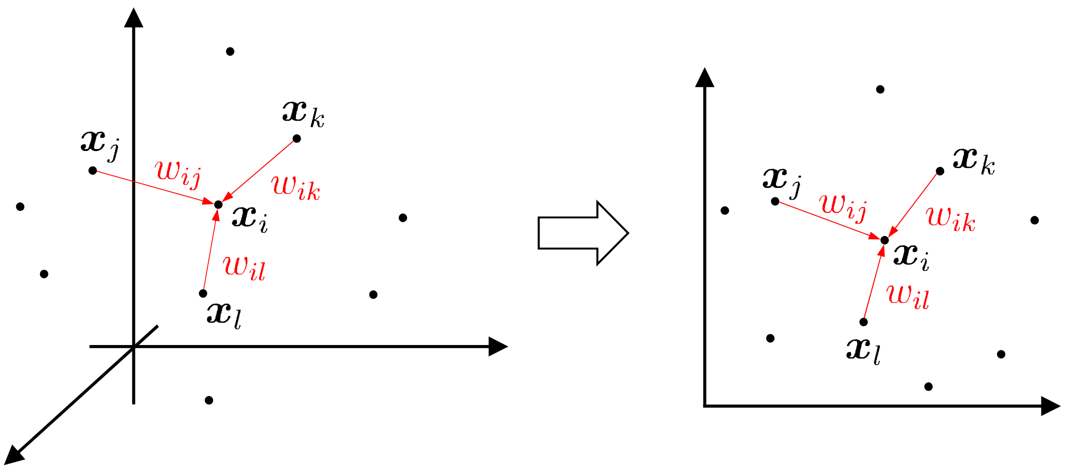
输出: 样本集 D 在低维空间的投影 $Z = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_m\}$.

图 10.8 Isomap 算法

局部线性嵌入

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

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 $\boldsymbol{w}_i$：

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

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

局部线性嵌入

令 $\boldsymbol{x}_i \in \mathbb{R}^{d\times 1}, Q_i = \{q_i^1, q_i^2, \dots, q_i^n\}$

$$
\begin{aligned}
\sum_{i=1}^{m}\left\|\boldsymbol{x}_i - \sum_{j\in Q_i} w_{ij}\boldsymbol{x}_j\right\|_2^2 &= \sum_{i=1}^{m}\left\|\sum_{j\in Q_i} w_{ij}\boldsymbol{x}_i - \sum_{j\in Q_i} w_{ij}\boldsymbol{x}_j\right\|_2^2 \\
&= \sum_{i=1}^{m}\left\|\sum_{j\in Q_i} w_{ij}(\boldsymbol{x}_i - \boldsymbol{x}_j)\right\|_2^2 \\
&= \sum_{i=1}^{m}\|\mathbf{X}_i\boldsymbol{w_i}\|_2^2 \\
&= \sum_{i=1}^{m}\boldsymbol{w_i}^{\mathrm{T}}\mathbf{X}_i^{\mathrm{T}}\mathbf{X}_i\boldsymbol{w_i}
\end{aligned}
$$

- 其中

$$\boldsymbol{w_i} = (w_{iq_i^1}, w_{iq_i^2}, \dots, w_{iq_i^n}) \in \mathbb{R}^{n\times 1}$$

$$\mathbf{X}_i = (\boldsymbol{x}_i - \boldsymbol{x}_{q_i^1}, \boldsymbol{x}_i - \boldsymbol{x}_{q_i^2}, \dots, \boldsymbol{x}_i - \boldsymbol{x}_{q_i^n}) \in \mathbb{R}^{d\times n}$$

局部线性嵌入

- 目标函数为：

$$\min_{\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_m} \sum_{i=1}^{m} \boldsymbol{w_i}^{\mathrm{T}} \mathbf{X}_i^{\mathrm{T}} \mathbf{X}_i \boldsymbol{w_i}$$

$$\text{s.t. } \boldsymbol{w_i}^{\mathrm{T}} \boldsymbol{I} = 1$$

- 拉格朗日函数为：

$$L(\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_m,\lambda) = \sum_{i=1}^{m} \boldsymbol{w_i}^{\mathrm{T}} \mathbf{X}_i^{\mathrm{T}} \mathbf{X}_i \boldsymbol{w_i} + \lambda_i \left(\boldsymbol{w_i}^{\mathrm{T}} \boldsymbol{I} - 1\right)$$

$$\boldsymbol{w_i} = -\frac{1}{2}\lambda_i \left(\mathbf{X}_i^{\mathrm{T}} \mathbf{X}_i\right)^{-1} \boldsymbol{I} \Rightarrow -\frac{1}{2}\lambda_i = \frac{1}{\boldsymbol{I}^{\mathrm{T}} \left(\mathbf{X}_i^{\mathrm{T}} \mathbf{X}_i\right)^{-1} \boldsymbol{I}}$$

局部线性嵌入

- 目标函数为：

$$\min_{\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_m} \sum_{i=1}^{m} \boldsymbol{w_i}^{\mathrm{T}} \mathbf{X}_i^{\mathrm{T}} \mathbf{X}_i \boldsymbol{w_i}$$

$$\text{s.t. } \boldsymbol{w_i}^{\mathrm{T}} \boldsymbol{I} = 1$$

- 拉格朗日函数为：

$$L(\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_m, \lambda) = \sum_{i=1}^{m} \boldsymbol{w_i}^{\mathrm{T}} \mathbf{X}_i^{\mathrm{T}} \mathbf{X}_i \boldsymbol{w_i} + \lambda_i \left(\boldsymbol{w_i}^{\mathrm{T}} \boldsymbol{I} - 1 \right)$$

$$\boldsymbol{w_i} = -\frac{1}{2}\lambda_i \left(\mathbf{X}_i^{\mathrm{T}} \mathbf{X}_i\right)^{-1} \boldsymbol{I} \quad \Rightarrow \quad -\frac{1}{2}\lambda_i = \frac{1}{\boldsymbol{I}^{\mathrm{T}} \left(\mathbf{X}_i^{\mathrm{T}} \mathbf{X}_i\right)^{-1} \boldsymbol{I}}$$

局部线性嵌入

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 $\boldsymbol{w}_i$ ：

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

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

其中 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 均为已知，令 $[P_i]_{jk} = (x_i - x_j)^T (x_i - x_k)$ ， w_{ij}有闭式解（前提是矩阵可逆）

$$\boldsymbol{w}_i = \frac{(\mathbf{X}_i^{\mathrm{T}}\mathbf{X}_i)^{-1}\boldsymbol{I}}{\boldsymbol{I}^{\mathrm{T}}(\mathbf{X}_i^{\mathrm{T}}\mathbf{X}_i)^{-1}\boldsymbol{I}} \Rightarrow w_{ij} = \frac{\sum_{k\in Q_i}\left[P_i^{-1}\right]_{jk}}{\sum_{j,k\in Q_i}\left[P_i^{-1}\right]_{jk}}$$

局部线性嵌入

局部线性嵌入（Locally Linear Embedding，LLE）

- LLE在低维空间中保持 w_i 不变，于是x_i对应的低维空间坐标 z_i 可通过下式求解：

$$\min_{z_1,z_2,\ldots,z_m} \sum_{i=1}^{m} \left\| z_i - \sum_{j \in Q_i} w_{ij} z_j \right\|_2^2$$

局部线性嵌入

$$Z=(z_1,\cdots,z_m)\in R^{d'\times m}, e_i=[0,\cdots,1,\cdots,0]^T, w_i=[w_{i1},\cdots,w_{im}]^T$$

$$\sum_{i=1}^{m}\left\|z_i-\sum_{j\in Q_i}w_{ij}z_j\right\|=\sum_{i=1}^{m}\|Ze_i-Zw_i\|^2$$

$$=\sum_{i=1}^{m}(e_i-w_i)^T Z^T Z(e_i-w_i)$$

$$=\sum_{i=1}^{m}tr(Z(e_i-w_i)(e_i-w_i)^T Z^T)$$

$$=\sum_{i=1}^{m}tr(Z(e_i-w_i)(e_i-w_i)^T Z^T)$$

$$=tr(Z(I-W)(I-W)^T Z^T)$$

- 则优化式可写为 $M=(I-W)(I-W)^T$

$$\min_{\mathbf{Z}}\ \mathrm{tr}(\mathbf{ZMZ}^{\mathrm{T}})$$

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

局部线性嵌入

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

目录

度量学习

低维线性嵌入，通过局部线性求得M

$$\min_{\mathbf{Z}} \operatorname{tr}(\mathbf{Z}\mathbf{M}\mathbf{Z}^{\mathrm{T}})$$
$$\text{s.t. } \mathbf{Z}\mathbf{Z}^{\mathrm{T}} = \mathbf{I}.$$

- 为何不直接尝试“学习”出一个合适的距离度量M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

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

其中 $dist_{ij,k}$ 表示 $\boldsymbol{x}_i$ 与 $\boldsymbol{x}_j$ 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

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

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

度量学习

- 考虑半正定对称矩阵 $\mathbf{M}=\mathbf{P}\mathbf{P}^{\mathrm{T}}$，于是就得到了马氏距离 (Mahalanobis distance)。

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

- 对 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 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 M 。

度量学习

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

$$p_{ij} = \frac{\exp(-||\boldsymbol{x}_i - \boldsymbol{x}_j||_{\mathbf{M}}^2)}{\sum_l \exp(-||\boldsymbol{x}_i - \boldsymbol{x}_l||_{\mathbf{M}}^2)},$$

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

$$p_i = \sum_{j \in \Omega_i} p_{ij},$$

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

度量学习

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

- 整个样本集上的留一法正确率为

$$\sum_{i=1}^{m} p_i = \sum_{i=1}^{m} \sum_{j \in \Omega_i} p_{ij}.$$

- 由 $p_{ij} = \frac{\exp(-||\boldsymbol{x}_i - \boldsymbol{x}_j||_{\mathbf{M}}^2)}{\sum_l \exp(-||\boldsymbol{x}_i - \boldsymbol{x}_l||_{\mathbf{M}}^2)}$ 和 $\mathbf{M} = \mathbf{P}\mathbf{P}^{\mathrm{T}}$，则NCA的优化目标为

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

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 $\mathbf{M}$。

度量学习

- 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

$$
\begin{aligned}
\min_{\mathbf{M}} \quad & \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{M}} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_{\mathbf{M}}^2 \\
\text{s.t.} \quad & \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{C}} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_{\mathbf{M}}^2 \geq 1, \\
& \mathbf{M} \succeq 0.
\end{aligned}
$$

总结

MDS算法

根据距离约束$\|x_i - x_j\| = \|z_i - z_j\|, \forall i, j$。找一个矩阵$B = Z^T Z$。对B进行正交分解求Z

PCA算法

$$z_i = W^T x_i, \hat{x}_i = z_{i1} w_i + \cdots z_{id'} w_{d'} = W z_i$$
$$\sum_{i=1}^{m} \|\hat{x}_i - x_i\|^2 \propto -tr(W^T \left(\sum_{i=1}^{m} x_i x_i^T\right) W) \Rightarrow \min_{W} -tr(W^T X X^T W), s.t. W^T W = I$$
拉格朗日乘子法对W求偏导，满足$X X^T = \Lambda W$.正交分解求W.

KCPA算法

$$\left(\sum_{i=1}^{m} \phi(x_i)\phi(x_i)^T\right) w_j = \lambda_j w_j, w_j = \phi(X)\alpha_j$$
$$\Rightarrow \phi(X)^T \phi(X)\phi(X)^T \phi(X)\alpha_j = \lambda_j \phi(X)^T \phi(X)\alpha_j \Rightarrow KK\alpha_j = \lambda_j K\alpha_j, K_{ij} = \kappa(x_i, x_j)$$

得到A后$z_i = W^T \phi(x_i) = A^T \phi(X)^T \phi(x_i) = A^T [k(1,i), \cdots, k(N,i)]^T$

等度量映射(Isomap)

局部流形下的K近邻+最短路+MDS算法

局部线性嵌入 (LLE)

$$\sum_{i=1}^{m} \left\| z_i - \sum_{j \in Q_i} w_{ij} z_j \right\|^2 = tr(ZMZ^T), M = (I - W)(I - W)^T, s.t.\ ZZ^T = I$$