周志华 著

MACHINE LEARNING

机器学习

清华大学出版社

崔磊

Tel: 15829735700(M)
QQ：362626744
E-Mail: leicui@nwu.edu.cn

本章课件致谢：
丁尧相

第十章：降维与度量学习

大纲

- k近邻学习
- 多维缩放
- 主成分分析
- 流形学习
- 度量学习

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。
  - 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，对于回归问题使用“平均法”获得预测结果。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。
    - 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。
    - 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。

“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

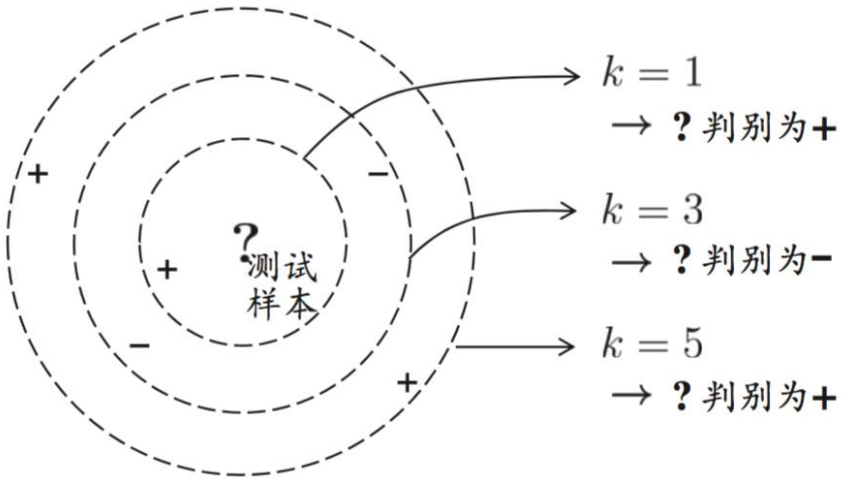


图 10.1 k 近邻分类器示意图. 虚线显示出等距线; 测试样本在 $k = 1$ 或 $k = 5$ 时被判别为正例, $k = 3$ 时被判别为反例.

- 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

分析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

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！

低维嵌入

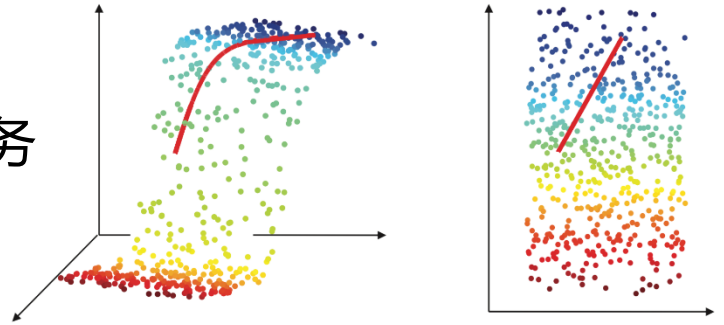
维数灾难 (curse of dimensionality)

- 上述讨论基于一个重要的假设：任意测试样本$\boldsymbol{x}$附近的任意小的δ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δ=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$(10^3)^{20} = 10^{60}$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)
  - 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。

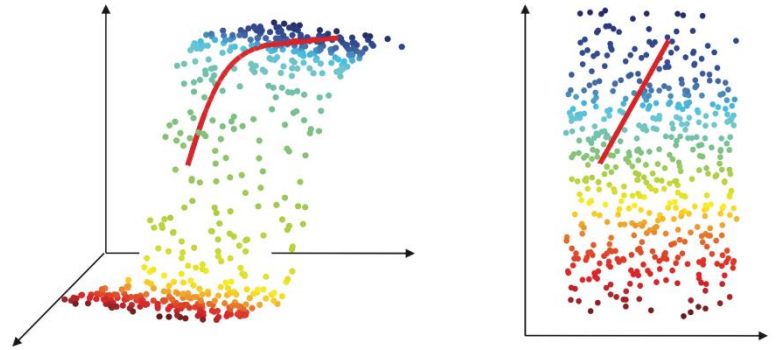


(a) 三维空间中观察到的样本点　　(b) 二维空间中的曲面

图 10.2　低维嵌入示意图

多维缩放

- 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 $||\boldsymbol{z}_i - \boldsymbol{z}_j|| = dist_{ij}$.



(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

- 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

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

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

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 $\mathbf{D}$ 求取内积矩阵 $\mathbf{B}$：

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

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

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

- PCA是一种常见的数据分析方法，常用于高维数据的降维，可用于提取数据的主要特征分量。
- 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。

相关线性代数基本知识

向量表示与基变换

1、内积

两个向量的 A 和 B 内积我们知道形式是这样的：

$$(a_1, a_2, \cdots, a_n) \cdot (b_1, b_2, \cdots, b_n)^{\mathsf{T}} = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

内积运算将两个向量映射为实数，其计算方式非常容易理解，那么内积的物理意义是什么呢？

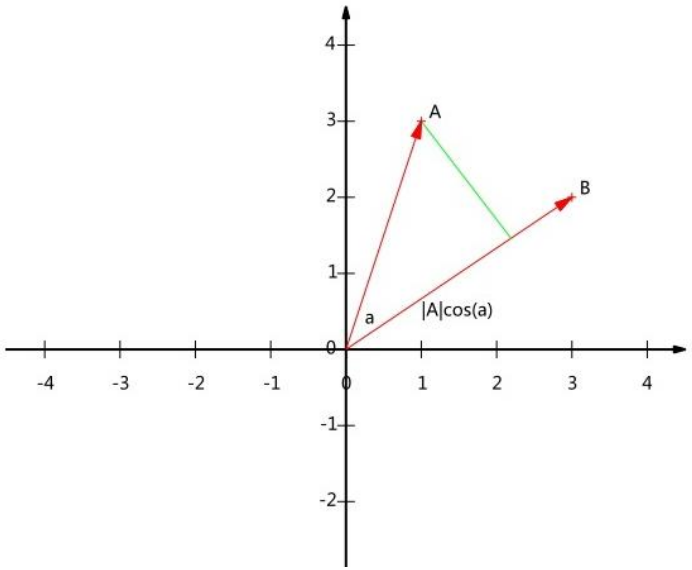
相关线性代数基本知识

向量表示与基变换

1、内积

我们从几何角度来分析，为了简单起见，我们假设 A 和 B 均为二维向量，则：

$$A = (x_1, y_1), \quad B = (x_2, y_2) \; A \cdot B = |A||B|cos(\alpha)$$



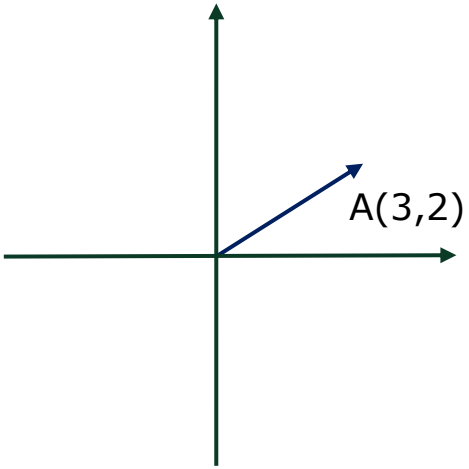
$$A \cdot B = |A|cos(a)$$

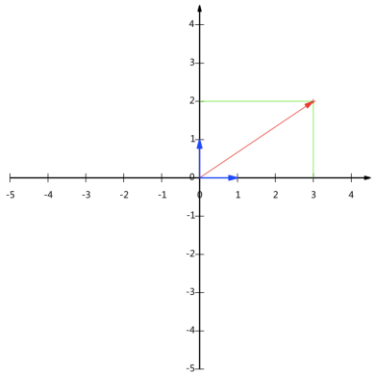
相关线性代数基本知识

向量表示与基变换

2、基

在我们常说的坐标系中，向量 (3,2) 其实隐式引入了一个定义：以 x 轴和 y 轴上正方向长度为 1 的向量为标准。向量 (3,2) 实际是说在 x 轴投影为 3 而 y 轴的投影为 2。**注意投影是一个标量，所以可以为负。**





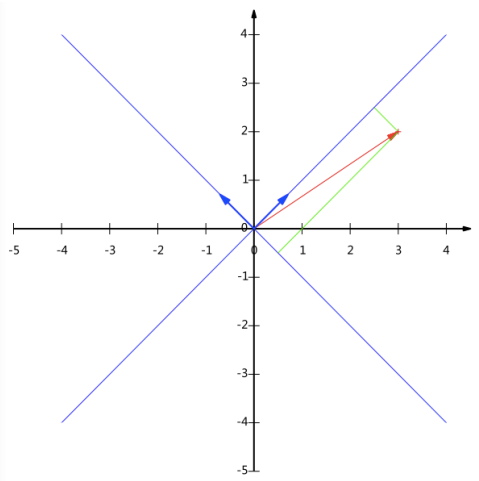
$x(1,0)^{\mathsf{T}} + y(0,1)^{\mathsf{T}}$

相关线性代数基本知识

向量表示与基变换

3、基变换的矩阵表示

练习：对于向量 (3,2) 这个点来说，在 $(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}})$ 和 $(-\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}})$ 这组基下的坐标是多少?



$$\begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 5/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix}$$

相关线性代数基本知识

向量表示与基变换

3、基变换的矩阵表示

推广一下，如果我们有 m 个二维向量，只要将二维向量按列排成一个两行 m 列矩阵，然后用“基矩阵”乘以这个矩阵就可以得到了所有这些向量在新基下的值。

例如对于数据点（1,1），（2,2），（3,3）来说，想变换到之前的那组基上，则可以表示为：

$$\begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} = \begin{pmatrix} 2/\sqrt{2} & 4/\sqrt{2} & 6/\sqrt{2} \\ 0 & 0 & 0 \end{pmatrix}$$

相关线性代数基本知识

向量表示与基变换

3、基变换的矩阵表示

进一步，我们可以把它写成通用的表示形式：

$$\begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_R \end{pmatrix} \begin{pmatrix} a_1 & a_2 & \cdots & a_M \end{pmatrix} = \begin{pmatrix} p_1 a_1 & p_1 a_2 & \cdots & p_1 a_M \\ p_2 a_1 & p_2 a_2 & \cdots & p_2 a_M \\ \vdots & \vdots & \ddots & \vdots \\ p_R a_1 & p_R a_2 & \cdots & p_R a_M \end{pmatrix}$$

其中 p_i 是一个行向量，表示第 i 个基，a_j 是一个列向量，表示第 j 个原始数据记录。实际上也就是做了一个向量矩阵化的操作。

两个矩阵相乘的意义是将右边矩阵中的每一列向量 变换到左边矩阵中以每一行行向量为基所表示的空间中去。也就是说一个矩阵可以表示一种线性变换。

主成分分析

最大可分性

上面我们讨论了选择不同的基可以对同样一组数据给出不同的表示，如果基的数量少于向量本身的维数，则可以达到降维的效果。

但是我们还没回答一个最关键的问题：如何选择基才是最优的。或者说，如果我们有一组 N 维向量，现在要将其降到 K 维（K 小于 N），那么我们应该如何选择 K 个基才能最大程度保留原有的信息？

一种直观的看法是：希望投影后的投影值尽可能分散，因为如果重叠就会有样本消失。当然这个也可以从熵的角度进行理解，熵越大所含信息越多。

主成分分析

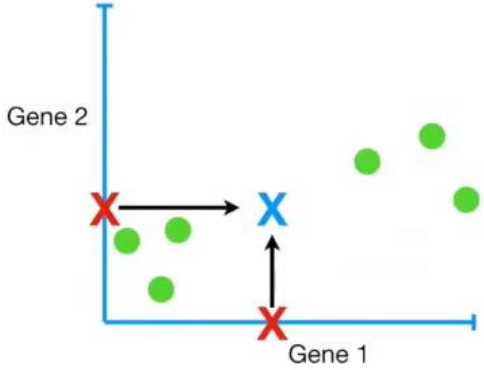
最大可分性

1、方差

$$Var(a) = \frac{1}{m}\sum_{i=1}^{m}(a_i - \mu)^2$$

中心化

$$Var(a) = \frac{1}{m}\sum_{i=1}^{m}a_i^2$$



中心化

主成分分析

最大可分性

2、协方差

在一维空间中我们可以用方差来表示数据的分散程度。而对于高维数据，我们用协方差进行约束，协方差可以表示两个变量的**相关性**。为了让两个变量尽可能表示更多的原始信息，我们希望它们之间不存在**线性相关性**，因为相关性意味着两个变量必然存在重复表示的信息。

$$Cov(a,b) = \frac{1}{m-1}\sum_{i=1}^{m}(a_i - \mu_a)(b_i - \mu_b)$$

中心化

$$Cov(a,b) = \frac{1}{m}\sum_{i=1}^{m}a_i b_i$$

主成分分析

最大可分性

至此，我们得到了降维问题的优化目标：**将一组 N 维向量降为 K 维，其目标是选择 K 个单位正交基，使得原始数据变换到这组基上后，各变量两两间协方差为 0，而变量方差则尽可能大（在正交的约束下，取最大的 K 个方差）。**

3、协方差矩阵

我们看到，最终要达到的目的与**变量内方差及变量间协方差**有密切关系。因此我们希望能将两者统一表示，仔细观察发现，两者均可以表示为内积的形式，而内积又与矩阵相乘密切相关。于是我们有：

假设我们只有 a 和 b 两个变量，那么我们将它们按行组成矩阵 X：

$$X = \begin{pmatrix} a_1 & a_2 & \cdots & a_m \\ b_1 & b_2 & \cdots & b_m \end{pmatrix}$$

主成分分析

最大可分性

3、协方差矩阵

$$X = \begin{pmatrix} a_1 & a_2 & \cdots & a_m \\ b_1 & b_2 & \cdots & b_m \end{pmatrix}$$

$$\frac{1}{m}XX^{\mathsf{T}} = \begin{pmatrix} \frac{1}{m}\sum_{i=1}^{m} a_i^2 & \frac{1}{m}\sum_{i=1}^{m} a_i b_i \\ \frac{1}{m}\sum_{i=1}^{m} a_i b_i & \frac{1}{m}\sum_{i=1}^{m} b_i^2 \end{pmatrix} = \begin{pmatrix} Cov(a,a) & Cov(a,b) \\ Cov(b,a) & Cov(b,b) \end{pmatrix}$$

我们很容易被推广到一般情况：

设我们有 m 个 n 维数据记录，将其排列成矩阵 $X_{n,m}$ ，设 $C = \frac{1}{m}XX^T$ ，则 C 是一个对称矩阵，其对角线分别对应各个变量的方差，而第 i 行 j 列和 j 行 i 列元素相同，表示 i 和 j 两个变量的协方差。

主成分分析

最大可分性

4、矩阵对角化

根据我们的优化条件，**我们需要将除对角线外的其它元素化为 0，并且在对角线上将元素按大小从上到下排列（变量方差尽可能大）**，这样我们就达到了优化目的。

设原始数据矩阵 X 对应的协方差矩阵为 C，而 P 是一组基按行组成的矩阵，设 Y=PX，则 Y 为 X 对 P 做基变换后的数据。设 Y 的协方差矩阵为 D，我们推导一下 D 与 C 的关系：

$$
\begin{aligned}
D &= \frac{1}{m} Y Y^T \\
&= \frac{1}{m} (PX)(PX)^T \\
&= \frac{1}{m} PXX^T P^T \\
&= P(\frac{1}{m} XX^T) P^T \\
&= PCP^T
\end{aligned}
$$

主成分分析

最大可分性

4、矩阵对角化

至此，我们离 PCA 还有仅一步之遥，我们还需要完成对角化。

由上文知道，协方差矩阵 C 是一个是对称矩阵，在线性代数中实对称矩阵有一系列非常好的性质：

1. 实对称矩阵不同特征值对应的特征向量必然正交。
2. 设特征向量 λ 重数为 r，则必然存在 r 个线性无关的特征向量对应于 λ ，因此可以将这 r 个特征向量单位正交化。

由上面两条可知，一个 n 行 n 列的实对称矩阵一定可以找到 n 个单位正交特征向量，设这 n 个特征向量为 $e_1, e_2, \cdots, e_n$ ，我们将其按列组成矩阵： $E = (e_1, e_2, \cdots, e_n)$ 。

主成分分析

最大可分性

4、矩阵对角化

则对协方差矩阵 C 有如下结论：

$$E^T C E = \Lambda = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix}$$

其中 Λ 为对角矩阵，其对角元素为各特征向量对应的特征值（可能有重复）。

到这里，我们发现我们已经找到了需要的矩阵 P：$P = E^{\mathsf{T}}$ 。

P 是协方差矩阵的特征向量单位化后按行排列出的矩阵，其中每一行都是 C 的一个特征向量。如果设 P 按照 Λ 中特征值的从大到小，将特征向量从上到下排列，则用 P 的前 K 行组成的矩阵乘以原始数据矩阵 X，就得到了我们需要的降维后的数据矩阵 Y。

主成分分析

最大可分性

5、拉格朗日乘子法

在叙述求协方差矩阵对角化时，我们给出希望变化后的变量有：**变量间协方差为 0 且变量内方差尽可能大。**然后我们通过实对称矩阵的性质给予了推导，此外我们还可以把它转换为最优化问题利用拉格朗日乘子法来给予推导。

我们知道样本点 x_i 在基 w 下的坐标为：$(x_i, w) = x_i^T w$，于是我们有方差：

$$
\begin{aligned}
D(x) &= \frac{1}{m}\sum_{i=1}^{m}(x_i^T w)^2 \\
&= \frac{1}{m}\sum_{i=1}^{m}(x_i^T w)^T (x_i^T w) \\
&= \frac{1}{m}\sum_{i=1}^{m} w^T x_i x_i^T w \\
&= w^T (\frac{1}{m}\sum_{i=1}^{m} x_i x_i^T) w
\end{aligned}
$$

主成分分析

最大可分性

5、拉格朗日乘子法

我们看到 $\frac{1}{m}\sum_{i=1}^{m} x_i x_i^T$ 就是原样本的协方差，我们另这个矩阵为 Λ ，于是我们有：

$$\begin{cases} max\{w^T \Lambda w\} \\ s.t. w^T w = 1 \end{cases}$$

然后构造拉格朗日函数：

$$L(w) = w^T \Lambda w + \lambda(1 - w^T w)$$

对 w 求导：

$$\Lambda w = \lambda w$$

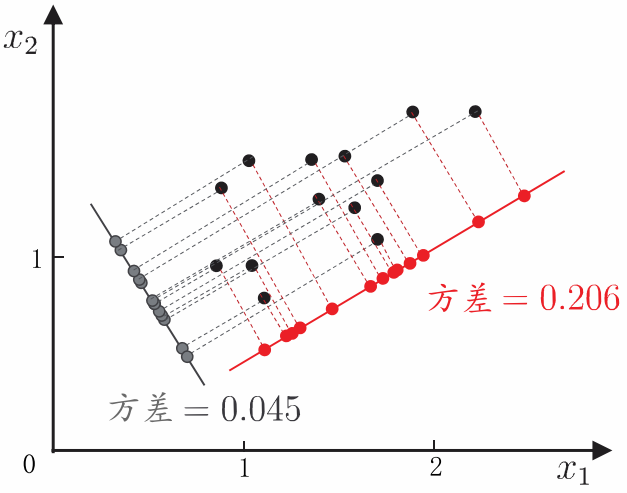
此时我们的方差为：$D(x) = w^T \Lambda w = \lambda w^T w = \lambda$

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

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



主成分分析

最近重构性

- 对样本进行中心化，$\sum_i \boldsymbol{x}_i = \mathbf{0}$，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

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

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

- 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

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

这就是主成分分析的优化目标。

显然与 $\begin{aligned}\max_{\mathbf{W}} \quad & \mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{W}) \\ \text{s.t.} \quad & \mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}.\end{aligned}$ 等价。

主成分分析

PCA的求解

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

- 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

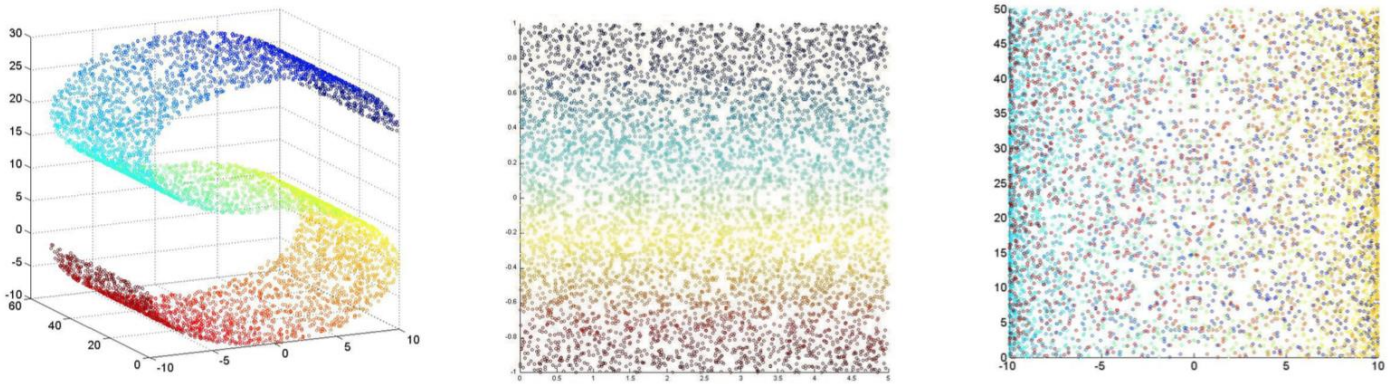
主成分分析

PCA算法性质

- **缓解维度灾难**：PCA 算法通过舍去一部分信息之后能使得样本的采样密度增大（因为维数降低了），这是缓解维度灾难的重要手段；
- **降噪**：当数据受到噪声影响时，最小特征值对应的特征向量往往与噪声有关，将它们舍弃能在一定程度上起到降噪的效果；
- **过拟合**：PCA 保留了主要信息，但这个主要信息只是针对训练集的，而且这个主要信息未必是重要信息。有可能舍弃了一些看似无用的信息，但是这些看似无用的信息恰好是重要信息，只是在训练集上没有很大的表现，所以 PCA 也可能加剧了过拟合；
- **特征独立**：PCA 不仅将数据压缩到低维，它也使得降维之后的数据各特征相互独立；

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)。
- 假定我们将在高维特征空间中把数据投影到由$\mathbf{W}$确定的超平面上，即PCA欲求解

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

- 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

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

$$= \sum_{i=1}^{m}\alpha_i^j\kappa(\boldsymbol{x}_i,\boldsymbol{x}).$$

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。

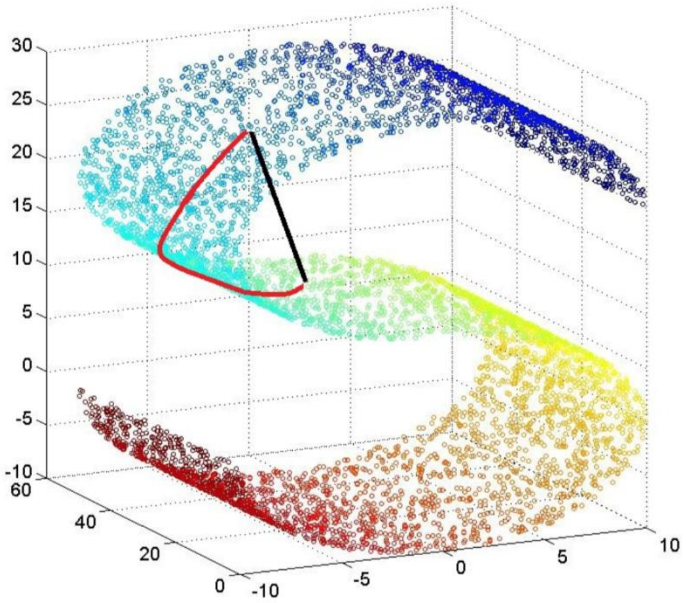
- 若低维流形嵌入到高维空间中，则数据样本在高维空间的分布虽然看上去非常复杂，但在局部上仍具有欧氏空间的性质，因此，可以容易地在局部建立降维映射关系，然后再设法将局部映射关系推广到全局。

- 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。

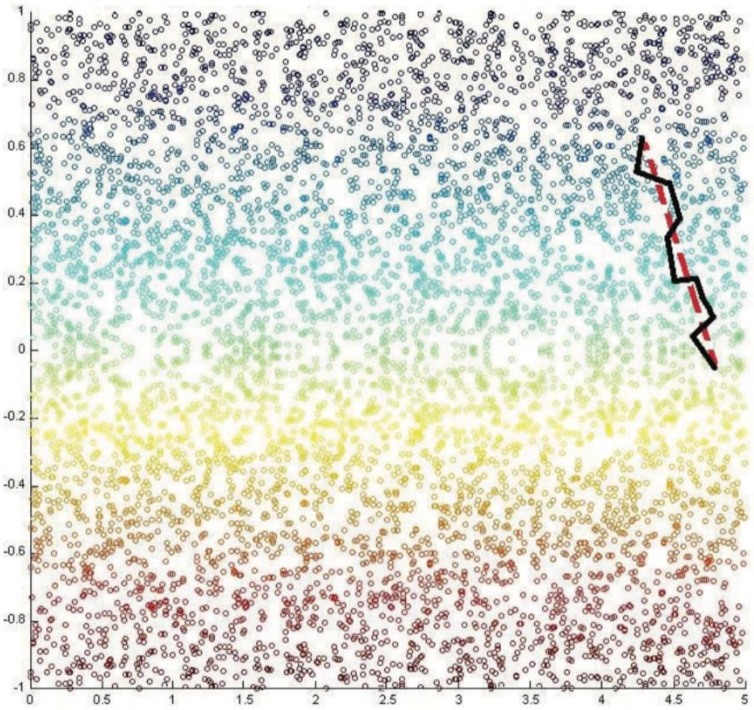


(a) 测地线距离与高维直线距离

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:

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

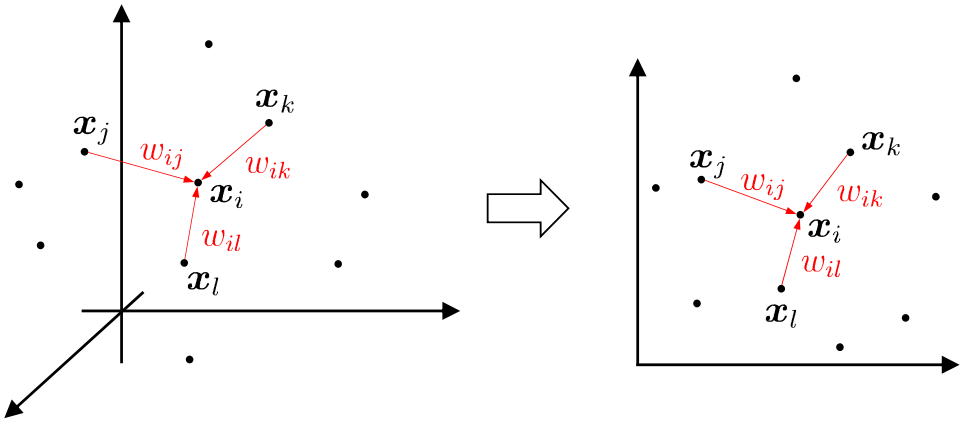
输出: 样本集 D 在低维空间的投影 $Z = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_m\}$.

图 10.8 Isomap 算法

流形学习

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。



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

流形学习

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 $C_{jk} = (\boldsymbol{x}_i - \boldsymbol{x}_j)^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{x}_k)$,$w_{ij}$ 有闭式解

$$w_{ij} = \frac{\sum_{k\in Q_i} C_{jk}^{-1}}{\sum_{l,s\in Q_i} C_{ls}^{-1}}.$$

流形学习

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：

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

- 令$\mathbf{Z} = (\mathrm{z}_1, \mathrm{z}_2, \ldots, \mathrm{z}_m) \in \mathbb{R}^{d' \times m}, (\mathbf{W})_{ij} = w_{ij},$

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

- 则优化式可重写为右式，并通过特征值分解求解。

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

流形学习

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

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

- 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

- $\mathbf{W}$的非对角元素均为零，这意味着坐标轴是正交的，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$\mathbf{W}$替换为一个普通的半正定对称矩阵$\mathbf{M}$，于是就得到了马氏距离(Mahalanobis distance)。

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

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

- 对$\mathbf{M}$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$\mathbf{M}$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$\mathbf{M}$。

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

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

- 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

- 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 $\mathcal{C}$ 与“勿连”(cannot-link)约束集合 $\mathcal{M}$：

$(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{C}$ 表示$\boldsymbol{x}_i$与$\boldsymbol{x}_j$相似，$(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{M}$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，于是可通过求解下面这个凸优化问题获得适当的度量矩阵 $\mathbf{M}$：

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

- 其中约束 $\mathbf{M} \succeq 0$ 表明$\mathbf{M}$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 $\mathbf{M}$，若 $\mathbf{M}$ 是一个低秩矩阵，则通过对 $\mathbf{M}$ 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 $\mathbf{M}$ 的秩 $\mathrm{rank}(\mathbf{M})$，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 $\mathbf{P} \in \mathbb{R}^{d \times \mathrm{rank}(\mathbf{M})}$，能用于降维目的。

小结

- k近邻学习
- 多维缩放
- 主成分分析
- 流形学习
- 度量学习