博雅大数据学院
BOYA BIG DATA COLLEGE

第八部分 降维方法

何向南
hexn@ustc.edu.cn

17 April 2020

为什么要降维

- 虽然许多实际是高维的，但是数据“内在的维数”可能很低
  - E.g., 许多特征之间有很高的相关性
  - 举例：

    通信数据原始有7维：入网时间、套餐价格、每月话费，每月流量、每月通话时长欠费金额、欠费月数

    “内在维度”可能只有3个：用户忠诚度、消费能力、欠费指数

- 维度灾难：模型的复杂度和计算量随着数据维度的增大而指数增长
  - 降维可降低模型复杂度，减少模型训练时间

- 降维可以作为特征提取第一种手段

降维的分类

- 常用的线性降维方法
  - 主成分分析（PCA）
  - 线性判别分析（LDA）
- 常用的非线性降维方法
  - 多维尺度变换（MDS）
  - 局部线性嵌入（LLE）

- 方差和标准差是用于衡量单个变量离散度的指标：
  - 方差：$\mathrm{var}(x) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$
  - 标准差：$\mathrm{std}(x) = \sqrt{\mathrm{var}(x)}$
- 协方差用于衡量两个变量之间的相关程度：
  - 协方差：$\mathrm{cov}(x, y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n-1}$

特征值分解

- 设 $\mathbf{A}$ 是 n 阶方阵，如果有常数 λ 和 n 维非零列向量 α 的关系式

$$\mathbf{A}\boldsymbol{\alpha} = \lambda\boldsymbol{\alpha}$$

成立，则称 λ 为方阵 $\mathbf{A}$ 的特征值，非零向量 $\boldsymbol{\alpha}$ 称为方阵 $\mathbf{A}$ 的对应于特征值 λ 的特征向量

特征值分解

- 设方阵 $A \in \mathbb{R}^{n \times n}$ 有 n 个线性无关的特征向量，存在一个特征分解公式：

  $A = U\Lambda U^{-1}$

- 其中 U 的列向量是方阵 A 的单位化后的特征向量，Λ 矩阵中的对角元素是方阵 A 的特征值

- $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n) = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix}$

- 表达式 $A = U\Lambda U^{-1}$ 称为方阵的特征值分解，此时方阵就被特征值和特征向量唯一表示

- 对$A = U\Lambda U^{-1}$进行一个变换，可得$U^{-1}AU = \Lambda = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix}$

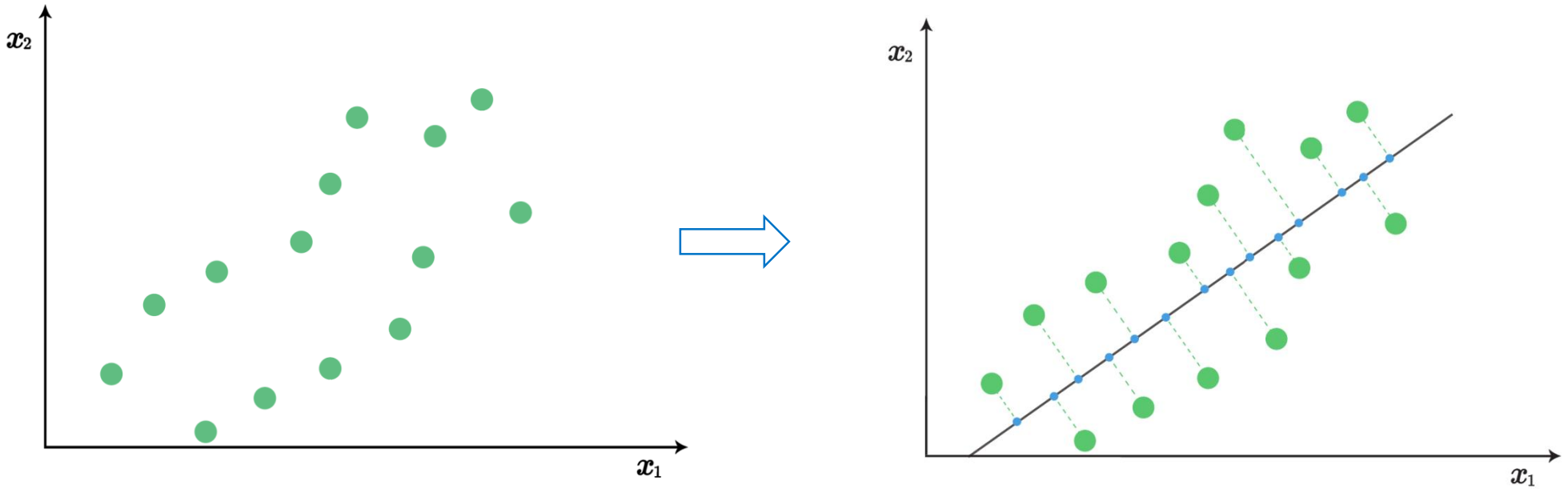
- 根据实对称矩阵单位化可知$U^{-1} = U^{T}$

主成分分析

主成分分析

- 主成分分析（Principal Component Analysis，PCA），最早在1901年由Karl Pearson提出
- 在人脸识别和图像压缩等领域得到了广泛的应用
- 基本思想：构造原始特征的一系列线性组合形成低维的特征，以**去除数据的相关性**，并使降维后的数据最大程度地保持原始高维数据的**方差信息**

如何将左下图中的二维数据投影到一维，使得数据的方差最大化地保留？



PCA算法原理

- 假设数据集 $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\}$，每个样本表示成 d 维向量，且每个维度均为连续型特征。数据集 D 也可以表示成一个 $n \times d$ 的矩阵 $\mathbf{X}$
- 为了便于描述，我们进一步假设每一维特征的均值均为零（已经标准化），且使用一个 $d \times l$ 的线性转换矩阵 $\mathbf{W}$ 来表示将 d 维的数据降到 l 维($l < d$)空间的过程，降维后的数据用 $\mathbf{Y}$ 表示，有

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{W}$$

PCA的数学模型

- 降维后数据的方差为

$$\begin{aligned}\mathrm{var}(\mathbf{Y}) &= \frac{1}{n-1}\mathrm{tr}(\mathbf{Y}^{\mathrm{T}}\mathbf{Y}) \\ &= \frac{1}{n-1}\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{W}) \\ &= \mathrm{tr}(\mathbf{W}^{\mathrm{T}}\frac{1}{n-1}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{W}).\end{aligned}$$

- 原始数据集的协方差矩阵$\boldsymbol{\Sigma} = \frac{1}{\mathbf{n-1}}\mathbf{X}^{\mathbf{T}}\mathbf{X}$，则PCA的数学模型为

$$\begin{aligned}\max_{\mathbf{W}} \quad & \mathrm{tr}(\mathbf{W}^{\mathrm{T}}\boldsymbol{\Sigma}\mathbf{W}) \\ \text{s.t.} \quad & \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{w}_i = 1 \quad i \in \{1, 2, \cdots, l\}.\end{aligned}$$

PCA模型求解

- 使用**拉格朗日乘子法**将上述带约束的最优化问题转化为无约束的最优化问题，对应的拉格朗日函数为

$$L(\mathbf{W}, \boldsymbol{\lambda}) = \mathrm{tr}(\mathbf{W}^{\mathrm{T}}\boldsymbol{\Sigma}\mathbf{W}) - \sum_{i=1}^{l} \lambda_i (\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{w}_i - 1)$$

其中 $\{\lambda_1, \lambda_2, \ldots, \lambda_l\}$ 为拉格朗日乘子，$\mathbf{w_i}$ 为矩阵 $\mathbf{W}$ 第 i 列

- 对 $\mathbf{w_i}$ 求偏导并令导数为零，有

$$\boldsymbol{\Sigma}\boldsymbol{w}_i = \lambda_i \boldsymbol{w}_i$$

- 即我们要求的转换矩阵 $\mathbf{W}$ 的每一个列向量 $\mathbf{w_i}$ 都是数据 $\mathbf{X}$ 的协方差矩阵 $\boldsymbol{\Sigma}$ 的特征向量，而 λ_i 为对应的特征值

PCA模型求解

Recall PCA的优化目标：

$$\max_{\mathbf{W}} \quad \mathrm{tr}(\mathbf{W}^{\mathrm{T}}\boldsymbol{\Sigma}\mathbf{W})$$

$$\text{s.t.} \quad \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{w}_i = 1 \quad i \in \{1, 2, \cdots, l\}.$$

- 由$\boldsymbol{\Sigma}\boldsymbol{w}_i = \lambda_i \boldsymbol{w}_i$，且$\mathrm{tr}(W^T \boldsymbol{\Sigma} W) = \sum_{i=1}^{l} w_i^T \boldsymbol{\Sigma} w_i$，有 $\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\boldsymbol{\Sigma}\mathbf{W}) = \sum_i^l \lambda_i$
- 故PCA最优化的方差等于原数据集 $\mathbf{X}$ 的协方差矩阵的特征值之和
- 要使上述方差最大，我们只要首先求得 $\boldsymbol{\Sigma}$ 的特征向量和特征值，然后取最大的 l 个特征值对应的特征向量组成转换矩阵 $\mathbf{W}$ 即可

PCA算法步骤

算法 7 PCA 算法

输入: 数据矩阵 $\mathbf{X}$，降维后样本维数 l.

输出: 转换矩阵 $\mathbf{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_l)$.

1: 对于 $\mathbf{X}$ 中的每一个样本 $\boldsymbol{x}_i$ 进行中心化处理:

$$\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i - \boldsymbol{m},$$

其中 $\boldsymbol{m} = \frac{1}{n}\sum_{j=1}^{n} \boldsymbol{x}_j$ 为样本均值;

2: 计算协方差矩阵 $\boldsymbol{\Sigma} = \frac{1}{n-1}\mathbf{X}^{\mathrm{T}}\mathbf{X}$;

3: 对协方差矩阵 $\boldsymbol{\Sigma}$ 做特征值分解并将特征值降序排序: $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$;

4: 取最大的前 l 个特征值相对应的特征向量 $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_l$ 组成转换矩阵 $\mathbf{W}$.

PCA小结

- 通过确定降维前后方差保留比例选择降维后的样本维数 l，可预先设置一个方差比例阈值如90%

$$\mathrm{var}_{\mathrm{ratio}}(l) = \frac{\sum_{i=1}^{l} \lambda_i}{\sum_{j=1}^{d} \lambda_j}$$

- 从数据重构角度来看，通过矩阵$\mathbf{W}^{\mathbf{T}}$从 $\mathbf{Y}$ 可以得到重构数据为 $\mathbf{XWW}^{\mathbf{T}}$，且重构误差为

$$\|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{W}^{\mathrm{T}}\|_{\mathrm{F}}^{2}$$

- 最小化上述重构误差等价于最大化降维后的方差

PCA小结

- 低维空间的特征都写成高维特征的线性加权形式，为**理解低维数据提供了便利**
- 主成分分析没有需要调参的超参数，具有**全局最优解**，不存在局部最优解的问题
- **计算效率较高**，其时间复杂度为$O(d^3)$，空间复杂度为$O(d^2)$
- 当数据中存在线性结构且方差较大时，通常可以选择主成分分析作为首选降维方法
- 方差最大化有时并不是我们感兴趣的优化目标

PCA应用实例 1

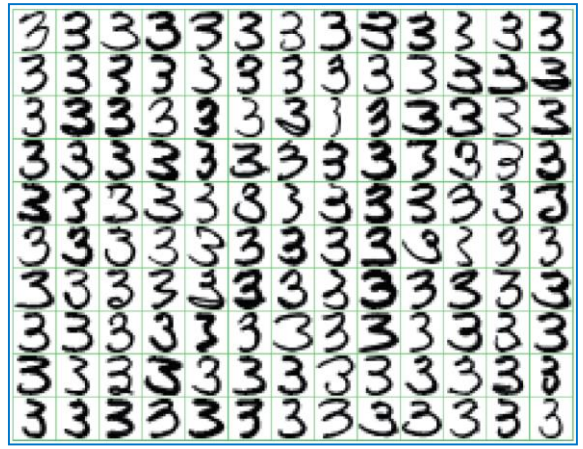
- 美国统计学家斯通在1947年关于国民经济的研究中，得到了17个反映国民收入与支出的变量要素，例如雇主补贴、消费资料、生产资料、纯公共支出、股息、利息、外贸平衡等
- 在进行主成分分析后，以97.4%的精度，用3个新变量就取代了原来的17个变量
- 根据经济学知识，斯通给这3个新变量分别命名为总收入F1，总收入变化率F2和经济发展或衰退趋势F3

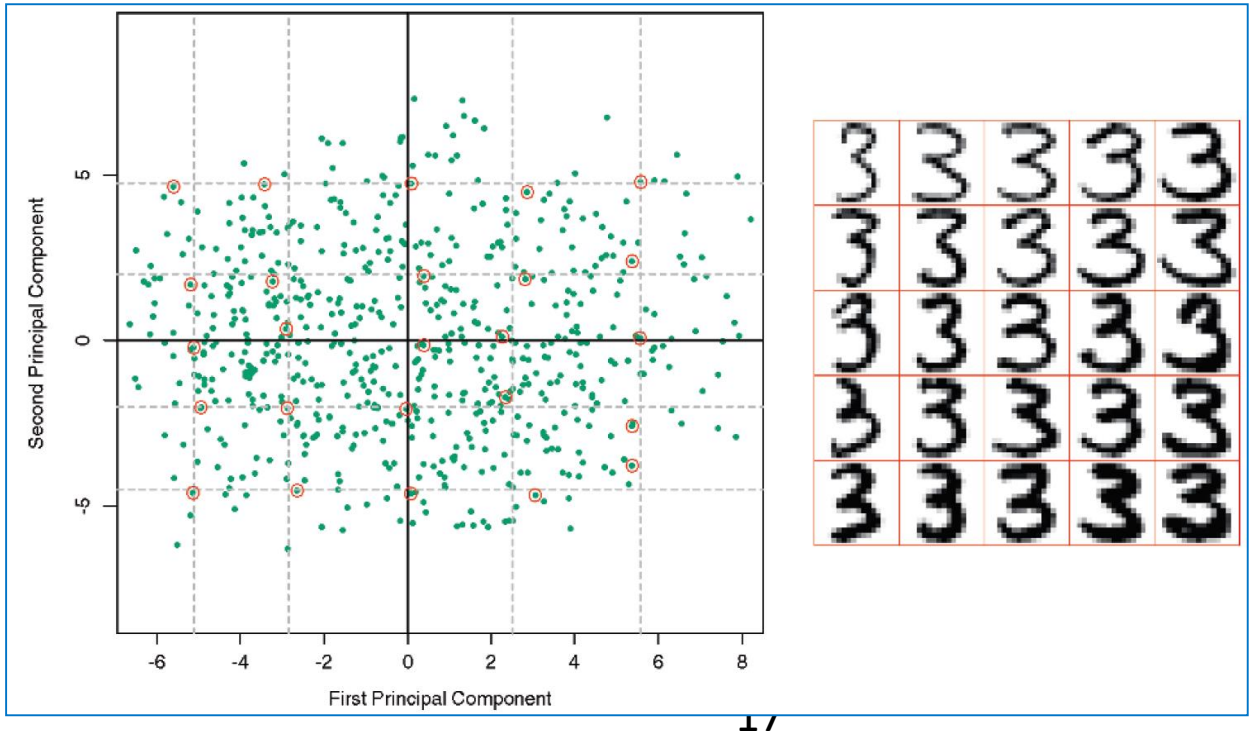
PCA应用实例 2

- 手写数字识别
- 数据集由658个手写数字3的图像组成，每张图片大小为16 x 16

$$\hat{f}(\lambda) = \bar{x} + \lambda_1 v_1 + \lambda_2 v_2$$

$$= \bar{x} + \lambda_1 \cdot + \lambda_2 \cdot .$$





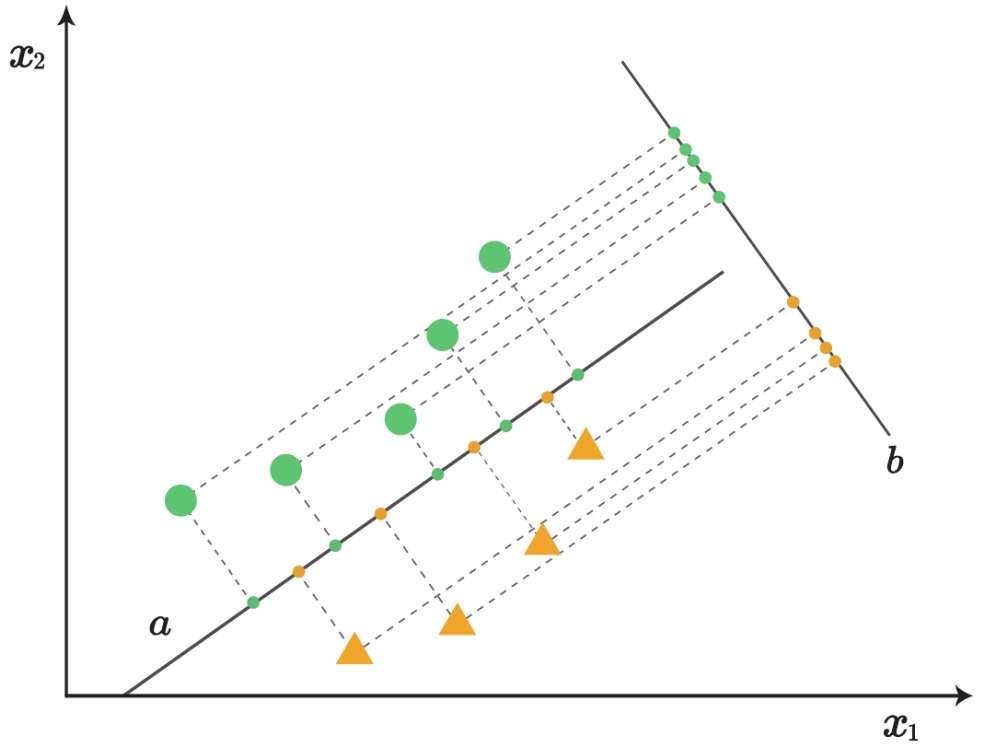
线性判别分析

线性判别分析

- 线性判别分析（Linear Discriminant Analysis, LDA）是一种典型的**有监督**的线性降维方法。由Ronald Fisher于1936年提出，也称为Fisher判别分析（ Fisher Discriminant Analysis ）
- 基本思想：使用数据的**类别信息**，将高维的样本线性投影到低维空间中，使得低维表示最有利于对数据进行分类

LDA原理示意

- 在右图中，原始样本分为两类 (分别用圆形和三角形表示)
- 使用PCA我们将得到直线a，降维后两类数据无法很好区分
- LDA能够找到直线b，使得降维后两类数据很好地区分开



基本记号

- 数据集为 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_n, y_n)\}$
- 样本标签 y_i 的取值范围$\{1,2,\ldots,C\}$，第 c 类的样本数量为 n_c，且有 $n = \sum_{i=1}^{C} n_i$
- 第 c 类样本均值：$\boldsymbol{m}_c = \frac{1}{n_c}\sum_{i=1}^{n_c} \boldsymbol{x}_i$，所有样本的均值：$\boldsymbol{m} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i$
- 通过线性转换矩阵 $\mathbf{W} \in \mathbb{R}^{d\times l}$ 将数据降到 l 维：$\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{W}$

LDA基本原理

- LDA的目标是利用样本的类别标签信息，找到一个利于数据分类的线性低维表示
- 这个目标可以从两个角度来量化
  - 第一个角度是使得降维后相同类样本尽可能近。使用类内离散度(within-class scatter)（类内样本的方差）度量
  - 第二个角度是使得降维后不同类样本尽可能远。使用类间离散度(between-class scatter)（不同类样本的均值的方差）度量

类内离散度和类间离散度

- 类内离散度（within-class scatter）
  - 降维前：第 c 类样本的类内离散度矩阵：$\mathbf{S}_c = \sum_{i=1}^{n_c}(\boldsymbol{x}_i - \boldsymbol{m}_c)(\boldsymbol{x}_i - \boldsymbol{m}_c)^{\mathrm{T}}$.

    总的类内离散度矩阵：$\mathbf{S}_w = \sum_{c=1}^{C}\frac{n_c}{n}\mathbf{S}_c$.
  - 降维后在$\boldsymbol{w}$方向的类内离散度为：$\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_w\boldsymbol{w}$
- 类间离散度 (between-class scatter)
  - 降维前：$\mathbf{S}_b = \sum_{c=1}^{C}\frac{n_c}{n}(\boldsymbol{m}_c - \boldsymbol{m})(\boldsymbol{m}_c - \boldsymbol{m})^{\mathrm{T}}$.
  - 降维后$\boldsymbol{w}$方向的类间离散度为：$\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_b\boldsymbol{w}$

LDA的数学模型

- LDA的目标是使得投影后数据更利于分类
- 所以要求投影后，类内散度 $\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_w\boldsymbol{w}$ 尽量小，类间散度 $\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_b\boldsymbol{w}$尽量大.
  优化目标:

$$\max_{\boldsymbol{w}} \quad \frac{\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_b\boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_w\boldsymbol{w}}$$

- 该优化问题等价于

$$\max_{\boldsymbol{w}} \quad \boldsymbol{w}^{\mathrm{T}}\mathbf{S}_b\boldsymbol{w}$$

$$\text{s.t.} \quad \boldsymbol{w}^{\mathrm{T}}\mathbf{S}_w\boldsymbol{w} = 1$$

LDA的模型求解

- 使用拉格朗日乘子法求解该问题，对应的拉格朗日函数为

$$L(\boldsymbol{w}, \lambda) = \boldsymbol{w}^{\mathrm{T}} \mathbf{S}_b \boldsymbol{w} - \lambda(\boldsymbol{w}^{\mathrm{T}} \mathbf{S}_w \boldsymbol{w} - 1)$$

- 对 $\boldsymbol{w}$ 求偏导并等于零，则有

$$\frac{\partial L(\boldsymbol{w}, \lambda)}{\partial \boldsymbol{w}} = 2\mathbf{S}_b \boldsymbol{w} - 2\lambda \mathbf{S}_w \boldsymbol{w},$$

$$\mathbf{S}_b \boldsymbol{w} = \lambda \mathbf{S}_w \boldsymbol{w}$$

- 即 $\mathbf{S}_w^{-1} \mathbf{S}_b \boldsymbol{w} = \lambda \boldsymbol{w}$

要求的投影方向 $\boldsymbol{w}$ 恰好是矩阵 $\mathrm{S_w^{-1}S_b}$的特征向量

LDA算法的求解方法

- 由于 $\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_b\boldsymbol{w} = \boldsymbol{w}^{\mathrm{T}}\lambda\mathbf{S}_w\boldsymbol{w} = \lambda\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_w\boldsymbol{w}$ ，且 $\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_w\boldsymbol{w} = 1$ ，最优化问题等价于

$$\max_{\boldsymbol{w}} \quad \lambda$$

$$\text{s.t.} \quad \mathbf{S}_w^{-1}\mathbf{S}_b\boldsymbol{w} = \lambda\boldsymbol{w}$$

- 因此，如果要使用LDA将数据降到l维，则需要求矩阵 $\mathbf{S}_w^{-1}\mathbf{S}_b$ 的前 l 个最大的特征值和对应的特征向量。将这 l 个投影向量$(\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_l)$按列组合，得到转换矩阵 $\mathbf{W}$

算法 8 LDA 算法

输入: 数据集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_n, y_n)\}$，降维后样本维数 l.

输出: 转换矩阵 $\mathbf{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_l)$.

1: 计算数据集的均值 $\boldsymbol{m}$ 和每一类数据的均值 $\boldsymbol{m}_c$:

$$\boldsymbol{m} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i, \quad \boldsymbol{m}_c = \frac{1}{n_c}\sum_{i=1}^{n_c}\boldsymbol{x}_i;$$

2: 计算类内离散度矩阵 $\mathbf{S}_w = \sum_{c=1}^{C}\frac{n_c}{n}\mathbf{S}_c$;

3: 其中 $\mathbf{S}_c$ 为第 c 类样本的类内离散度矩阵，根据 (8.13) 进行计算;

4: 计算类间离散度矩阵 $\mathbf{S}_b$:

$$\mathbf{S}_b = \sum_{c=1}^{C}\frac{n_c}{n}(\boldsymbol{m}_c - \boldsymbol{m})(\boldsymbol{m}_c - \boldsymbol{m})^{\mathrm{T}};$$

5: 计算矩阵 $\mathbf{S}_w^{-1}\mathbf{S}_b$，并对其做特征值分解，将矩阵 $\mathbf{S}_w^{-1}\mathbf{S}_b$ 的特征值降序排序;

6: 选取前 l 个特征值对应的特征向量 $(\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_l)$ 按列组合成转换矩阵 $\mathbf{W}$.

收缩判别分析

- 如果矩阵 $\mathbf{S_w}$ 无法求逆?

  例如，当我们的数据集 D 中每一类数据的样本数量 $\mathbf{n_c}$ 远小于样本维数 d 时，矩阵 $\mathbf{S_w}$ 将无法求逆。此时我们可以将矩阵 $\mathbf{S_w}$ 按照下式进行调整

$$\mathbf{S}_w = (1-\lambda)\mathbf{S}_w + \lambda\mathbf{I}$$

  其中 $\lambda > 0$为收缩参数

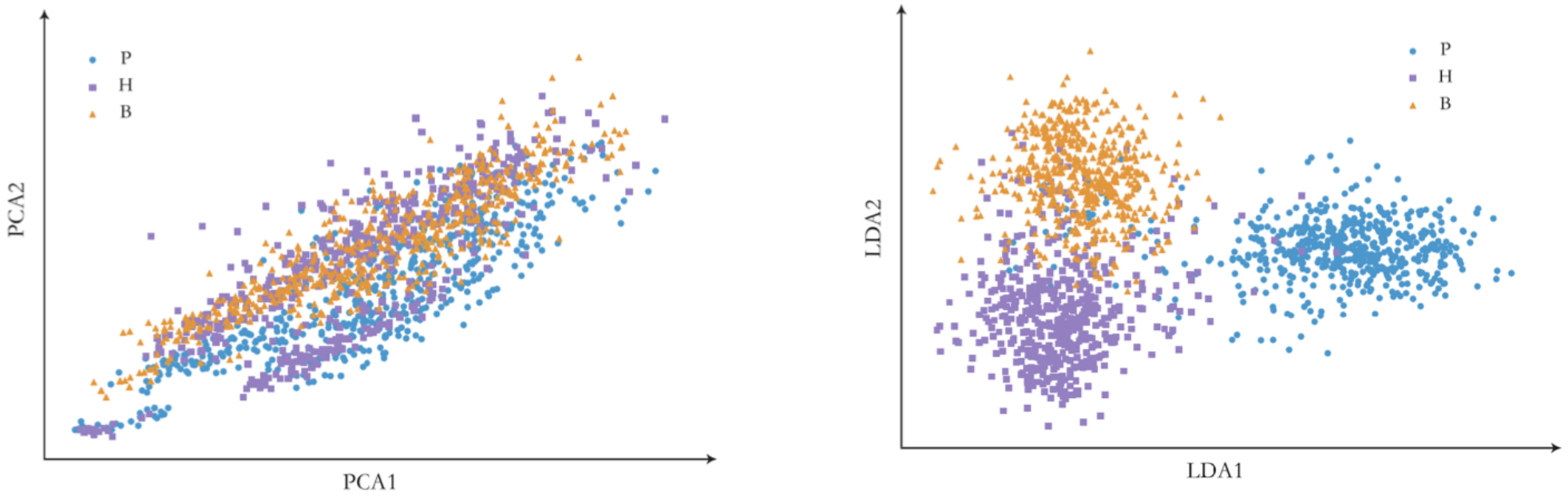
- 此时的算法称为收缩判别分析(shrinkage discriminant analysis)

LDA与PCA区别

- 基本思想不同
  - PCA选择样本投影具有最大方差的方向，最大化保留了数据的内部信息
  - LDA则考虑标签信息，使得投影后不同类之间的样本距离最大化，以及同类样本距离最小化
- 学习模式不同
  - PCA属于无监督式学习，适用范围更广，但并不能保证数据降维后数据易于分析
  - LDA属于有监督学习，同时具有分类和降维的能力

LDA与PCA的区别示例

光学字符识别数据集：26个大写英文字母的20,000个样本，特征16维



光学字符数据上PCA和LDA降维的结果对比

多维尺度变换

多维尺度变换

- 多维尺度变换(Multi-dimensional Scaling，MDS)的目标是找到数据的低维表示，使得降维前后样本之间的相似度信息尽量得以保留
- 在一些实际问题中，我们可能无法直接观察到样本的特征，而只能获取到样本之间的距离或相似度
- 多维尺度变换能够只利用样本间的距离信息，找到每一个样本的特征表示，且在该特征表示下样本的距离与原始的距离尽量接近
- 当降维后的维度较低时（<=3），可作为一种数据可视化方法

MDS的优化目标

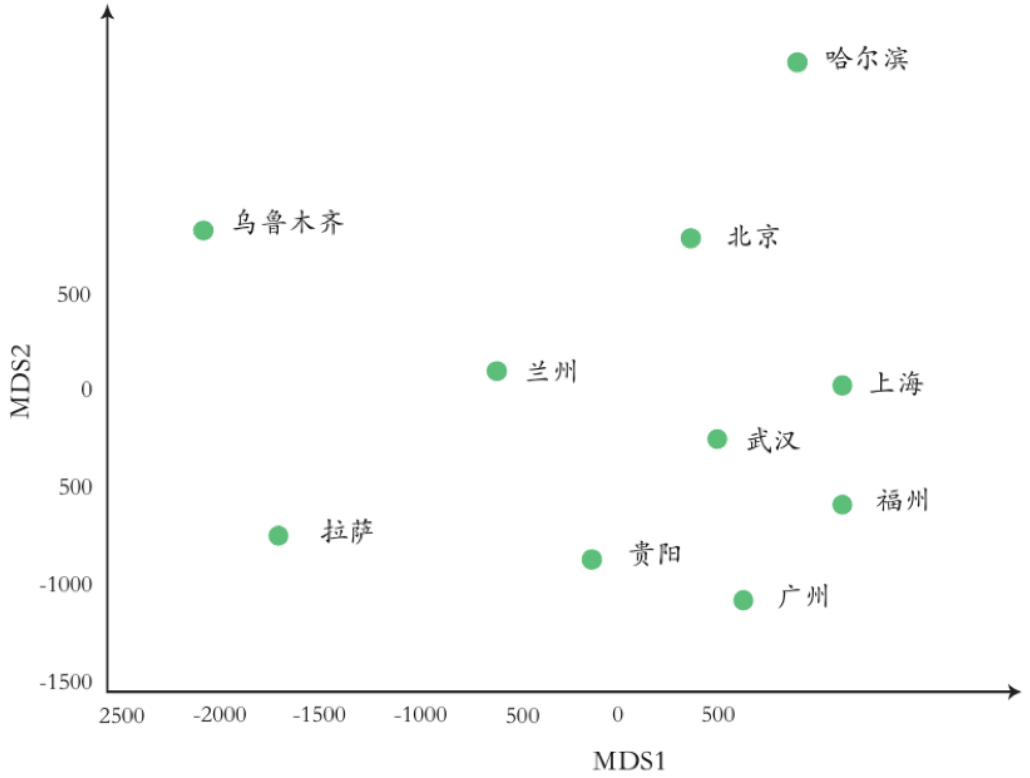
- 假设数据集包含 n 个样本，数据集用矩阵表示为 $\mathbf{X} \in \mathbb{R}^{n \times d}$，则可由 $\mathbf{X}$ 计算得到距离矩阵 $\mathbf{D} \in \mathbb{R}^{n \times n}$，其元素为 d_{ij}
- 若使用欧式距离来计算样本间的距离， 假设 $\boldsymbol{z}_i$为样本 $\boldsymbol{x}_i$在低维空间中的表示，则MDS的优化目标为

$$\min_{z_1, z_2, \cdots, z_n} \quad \sum_{i \neq j} \left(||\boldsymbol{z}_i - \boldsymbol{z}_j||_2 - d_{ij} \right)^2$$

MDS示例：根据城市间距离还原城市坐标

表 8.1: 中国部分城市之间的距离 (单位：公里)

	北京	上海	哈尔滨	乌鲁木齐	贵阳	兰州	福州	拉萨	广州	武汉
北京	0	1064	1055	2417	1734	1187	1558	2563	1888	1049
上海	1064	0	1675	3268	1527	1717	610	2902	1213	683
哈尔滨	1055	1675	0	3061	2769	2192	2282	3558	2791	1992
乌鲁木齐	2417	3268	3061	0	2571	1624	3463	1602	3281	2766
贵阳	1734	1527	2769	2571	0	1087	1256	1560	763	870
兰州	1187	1717	2192	1624	1087	0	1840	1376	1699	1148
福州	1558	610	2282	3463	1256	1840	0	2786	693	698
拉萨	2563	2902	3558	1602	1560	1376	2786	0	2311	2227
广州	1888	1213	2971	3281	763	1699	693	2311	0	839
武汉	1049	683	1992	2766	870	1148	698	2227	839	0



MDS将城市映射到二维空间的结果

度量MDS与非度量MDS

- 度量多维尺度变换(metric MDS)，即对于任意样本 i 、样本 j 和样本 k ，距离矩阵 $\mathbf{D}$ 满足以下条件：

$$d_{ik} + d_{jk} \geq d_{ij}$$

- 对于度量多维尺度变换，如果我们选择欧式距离来计算样本间的距离，则得到的低维表示将和主成分分析的结果是一致的
- 当不等式的条件不满足时，MDS不再以保持样本距离为目标，而是尽量保持样本间距离的次序，此时的方法称为非度量多维尺度变换(non-metric MDS)

MDS的其他应用

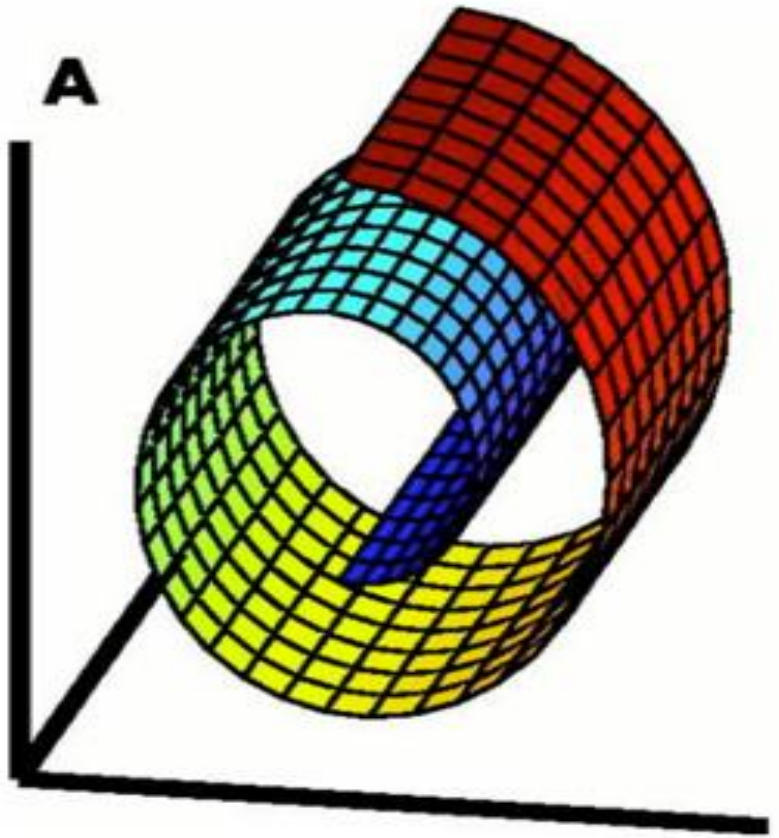
- 在社交网络分析领域，如果将节点之间的最短路径当作节点之间的距离且维度 l 选择为2维，MDS可以作为一种**图布局**(graph layout)算法来应用
- 在化学领域，MDS还被用于**分子构象**(molecular conformation)，即用来分析分子的空间结构

局部线性嵌入(Local Linear Embedding, LLE)

- Motivation: 在高维空间中，样本距离可能无法很好的刻画数据的特点

基于流形学习的非线性降维方法

- 举例：如果数据分布在高维空间的一个流形上，仅依靠高维空间的距离，无法反应其特点
  - 流形(manifold)是一种特殊的拓扑空间，局部具有欧式空间的性质
  - 流形是欧式空间中曲线、曲面等概念的推广



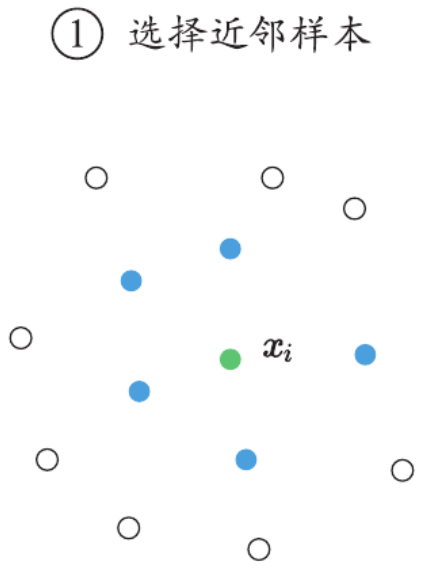
局部线性嵌入

- 局部线性嵌入（Locally Linear Embedding，LLE）：假设数据分布在高维空间中的低维度流形内

- 基本想法：将数据降到低维空间中，但是保留数据局部的线性关系
- 每一个样本点可以写成其 k 个近邻点的**线性组合**，从高维嵌入（embedding）到低维时尽量保持该局部线性关系

LLE算法的步骤

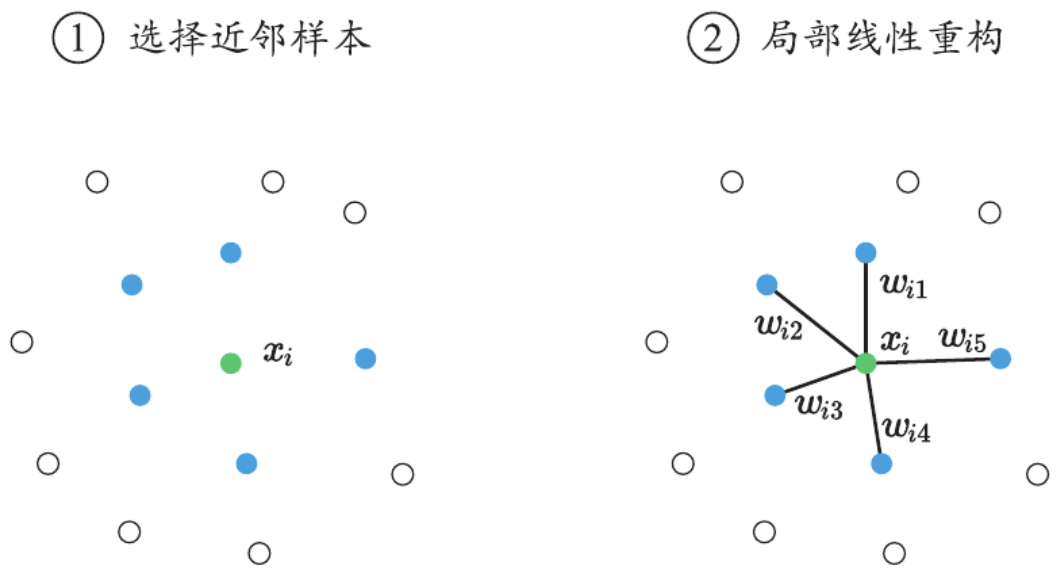
- 第一步，**选择近邻样本**。对于数据集中的每一个$\mathbf{x_i}$，使用 K近邻算法找到 k (通常选择 $k \leq d$)个近邻样本



LLE算法的步骤

- 第二步，**局部线性重构**。假设样本之间的线性关系使用矩阵 $\mathbf{W} \in \mathbb{R}^{n \times n}$表示，其元素 w_{ij}表示样本 j 在重构样本 i 时的系数。我们需要最小化样本的重构误差

$$\min_{\mathbf{W}} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \sum_{j} w_{ij}\boldsymbol{x}_j\|_2^2$$

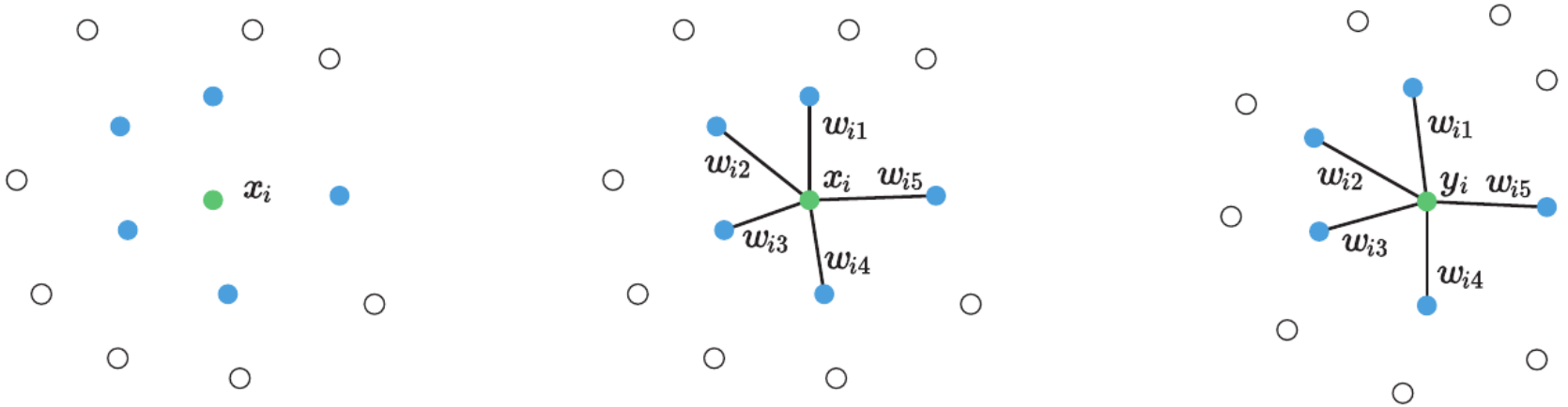


LLE算法的步骤

- 第三步，**寻找低维表示**。使用权重 $\mathbf{W}$ ，找到数据的低维表示$\mathbf{Y}$。同理，我们需要最小化低维样本的重构误差

$$\min_{\mathbf{Y}} \sum_{i=1}^{n} \|\boldsymbol{y}_i - \sum_j w_{ij}\boldsymbol{y}_j\|_2^2$$

① 选择近邻样本　② 局部线性重构　③ 寻找低维表示



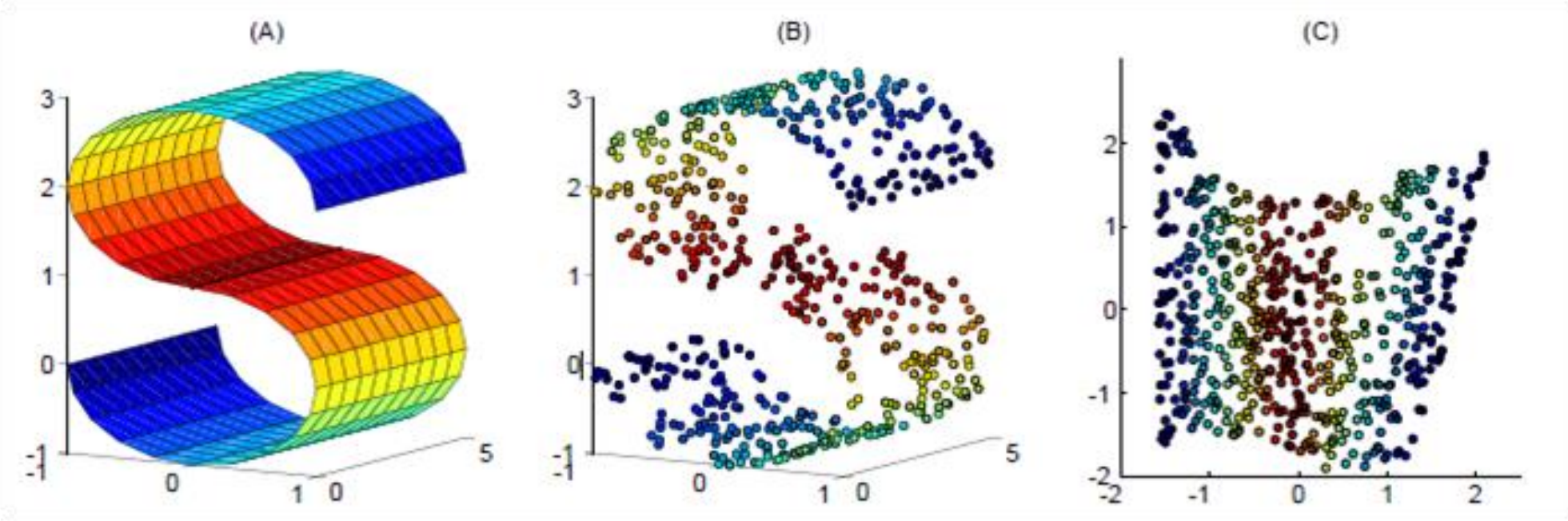
LLE算法的优缺点

- 优点
- LLE只**包含两个**需要调参的**超参数**：近邻样本数量 k 和正则化系数 λ；
- LLE算法得到的是**全局最优解**，不会陷入局部最优值。这些优点使得LLE在图像识别、图像分类和数据可视化领域得到广泛的应用。

- 缺点
- 由于需要使用K近邻算法寻找近邻样本，当**样本量不足**时，样本与其近邻样本距离可能较远，导致降维效果变差；
- LLE对**噪音和异常值**比较**敏感**，因而当数据中存在异常值时效果可能受到影响；
- **没有显式的映射**对应关系（例如PCA、LDA: Y=XW）
- LLE假设样本分布在一个单独的光滑**流形**，对于**分类问题该假设是不成立的。**

LLE示例

- 使用LLE将三维数据（图B）映射到二维（图C）之后，映射的数据仍能保持原有的数据流形（红色的点相互靠近，蓝色的也相互靠近）



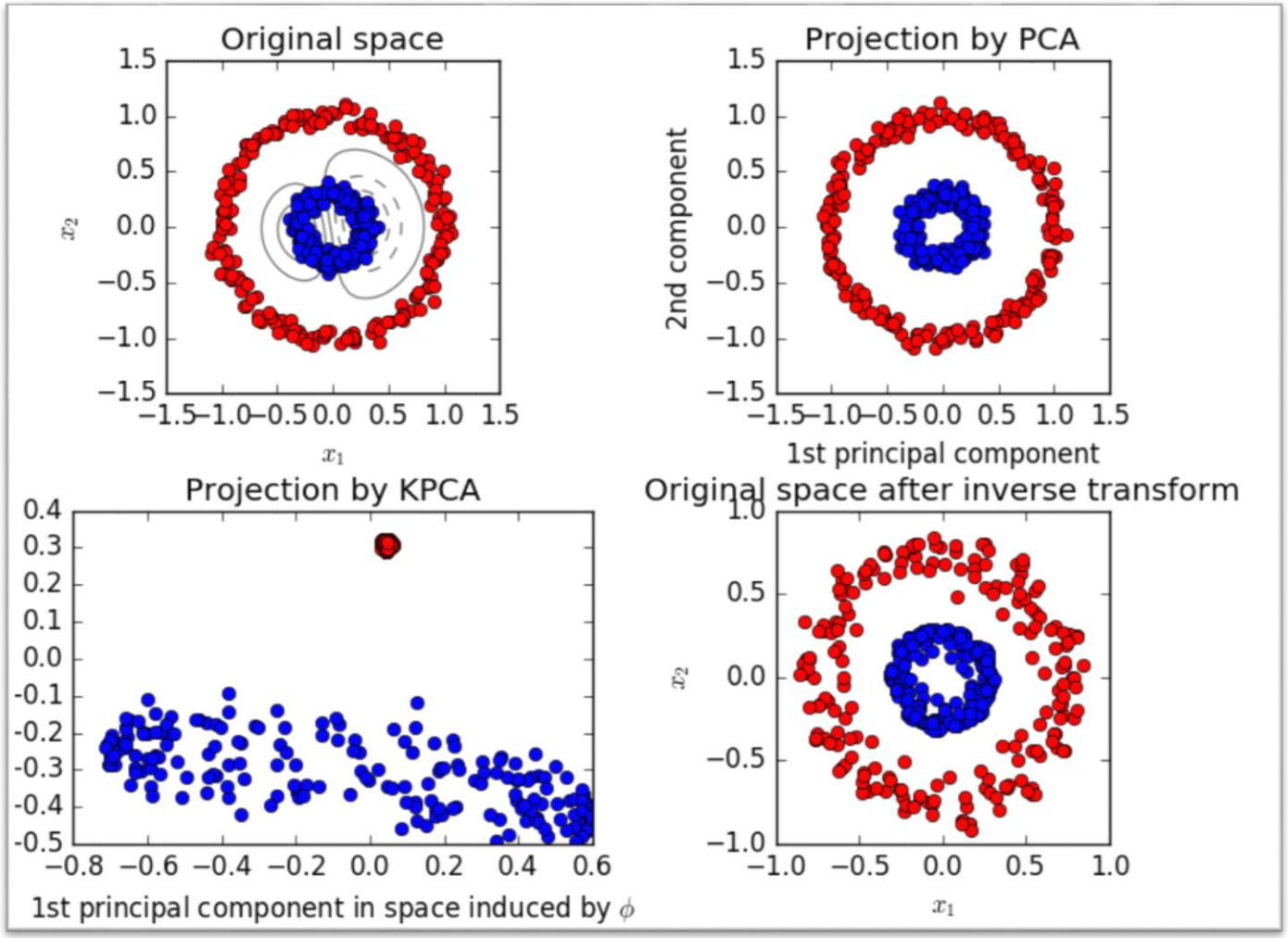
其他降维方法

核主成分分析

- 核主成分分析 (Kernel PCA, KPCA)利用核方法来弥补传统主成分分析只能进行线性降维的缺陷
- 基本思想：非线性映射 ϕ 将原始数据集 $\mathbf{X}$ 映射到更高维的空间中。在更高维空间中的样本 $\phi(\mathbf{X})$ 中利用 PCA 进行降维
- 主要应用：信号去噪和人脸识别

核主成分分析

- 通过核方法找到使得数据线性可分的高维空间，再进行PCA降维；
- 类似地，有Kernel LDA方法。

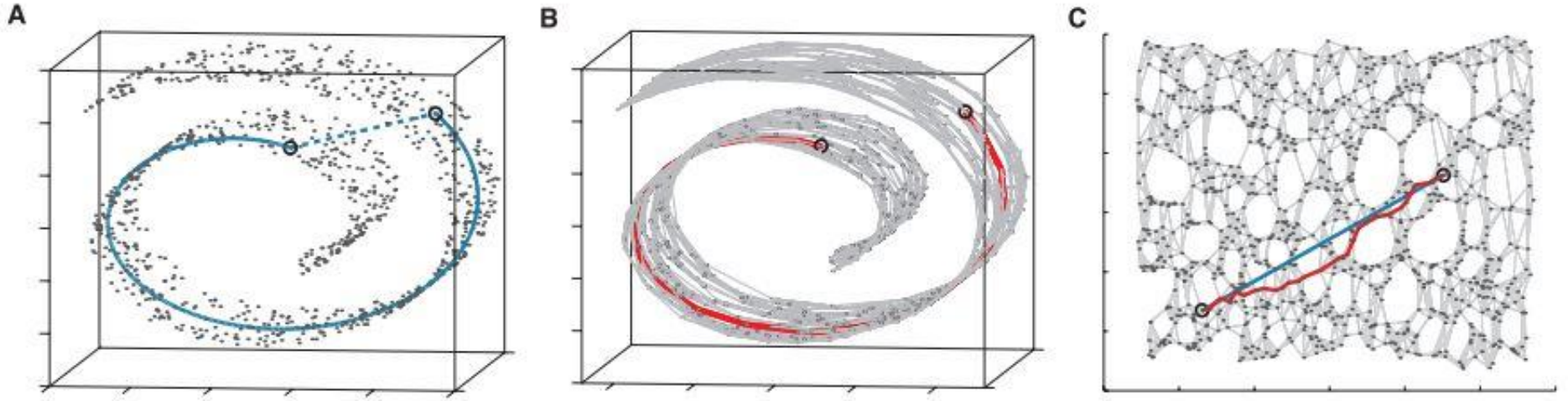


等度量映射

- 等度量映射(Isometric mapping，Isomap)是基于 MDS 改进的降维方法
- 用**测地距离（曲线距离）**作为空间中两点距离，原来是用欧氏距离，从而将位于某维流形上的数据映射到一个欧氏空间上
- 优点：Isomap 能较好保留数据的全局结构
- 缺点：对噪音比较敏感，而且计算复杂度也比较高

等度量映射

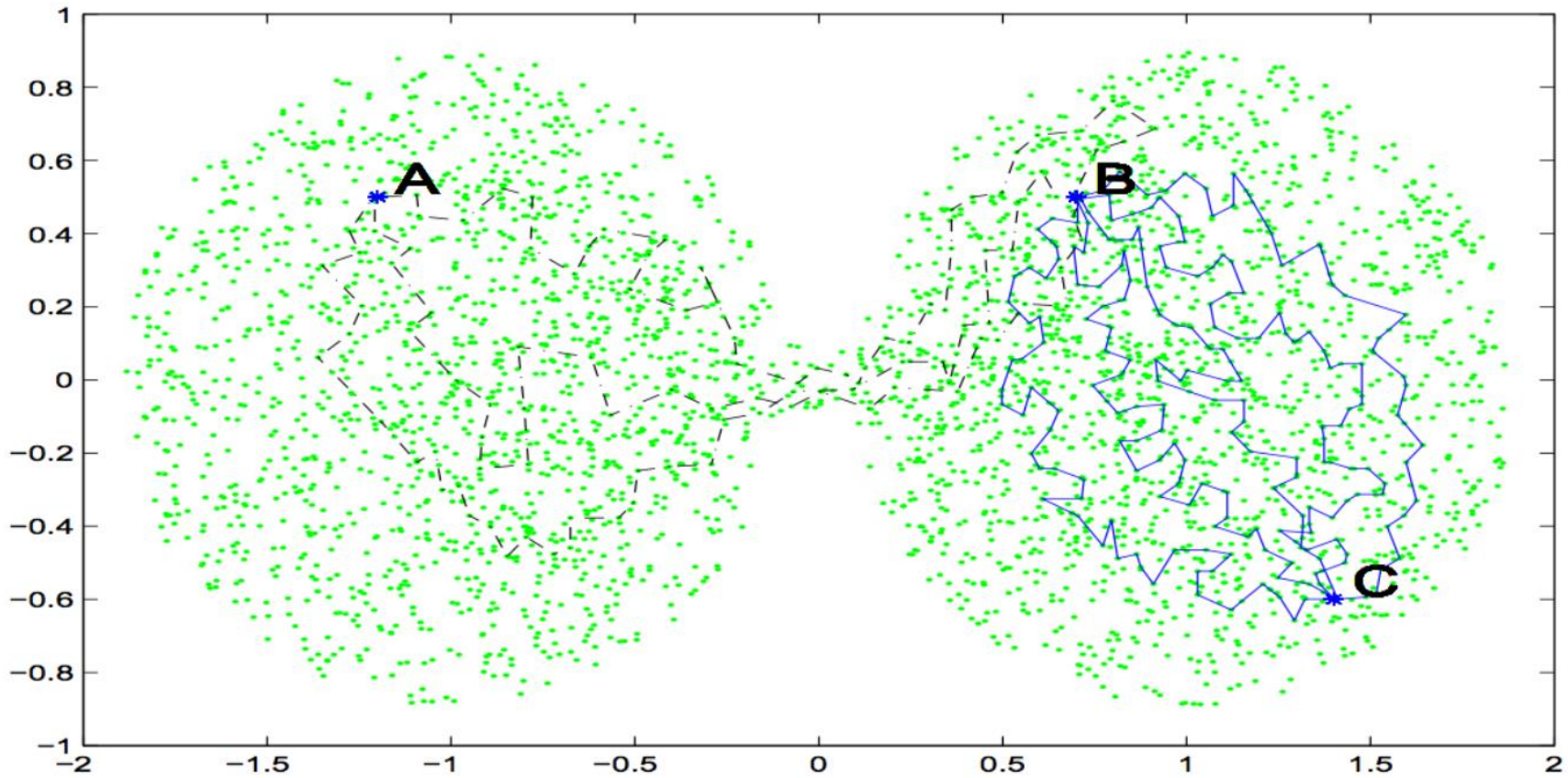
- 流形内两个相距较远的点在欧式空间中距离很近（图A）
- 距离较远的点使用测地距离(geodesic distance)代替欧式距离
- 测地距离通过图（节点为数据点，边为K个近邻点之间的边）中最短路径路径来近似（图B）



扩散映射

- 扩散映射(diffusion maps)是一类基于动态系统的非线性降维方法
- 基本思想：通过构建一个扩散图，使用图中的**扩散距离**来度量样本之间的相似度。与Isomap 使用最短路径来度量样本相似性不同的是，扩散映射本质上考虑了样本间的所有路径
- 优点：扩散映射避免了矩阵分解等操作，具有较好的抗噪能力

扩散映射



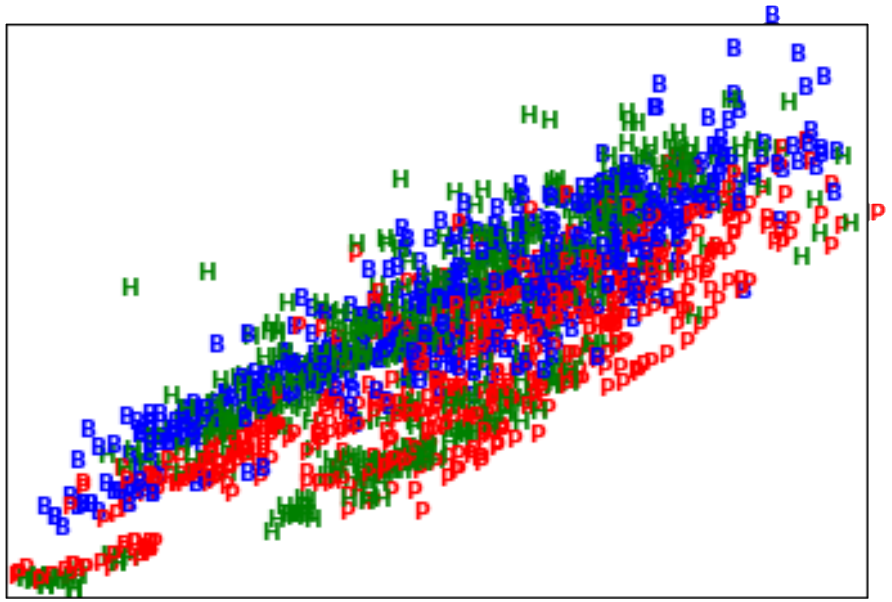
更多的降维方法

- **t分布随机近邻嵌入**(t-distributed Stochastic Neighbor Embedding, t-SNE)是一种适合对高维数据进行可视化的非线性降维方法。t-SNE使用两个样本间的条件概率来计算相似度，在降维时尽量保持这种条件概率信息
- 更多的降维方法包括**自编码器**(autoencoders)、**塞曼映射**(Sammon mapping)和**拉普拉斯特征映射**(Laplacian eigenmaps)等
- 不同非线性降维方法的对比分析，以及在人工和真实任务下的表现，可以参考文献
  - Laurens Van Der Maaten, Eric Postma, and Jaap Van den herik. “Dimensionality reduction: a comparative view”, In: J Mach Learn Res 10 (2009), pp.66-71

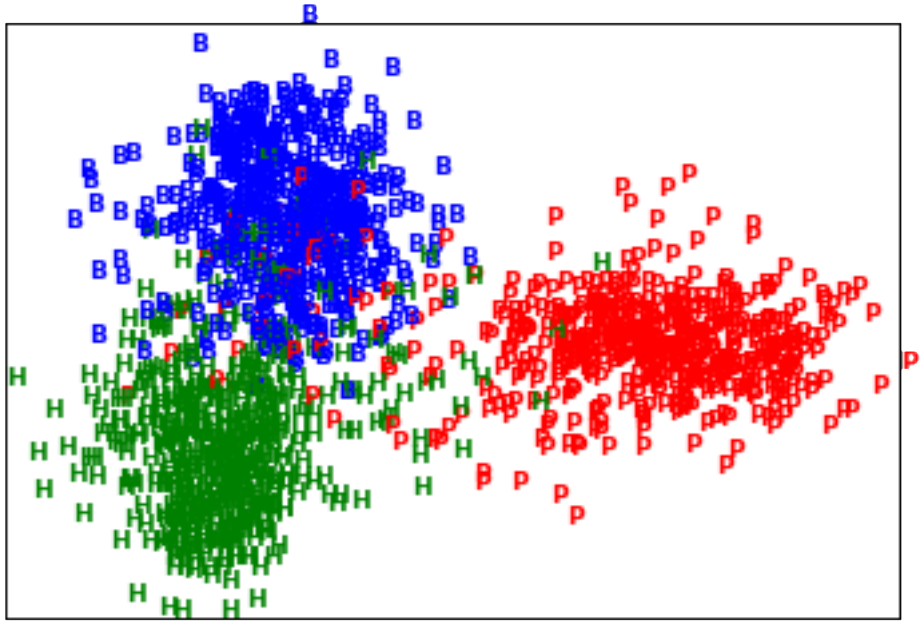
光学字符数据降维

- 降维方法在光学字符数据集中的应用

PCA

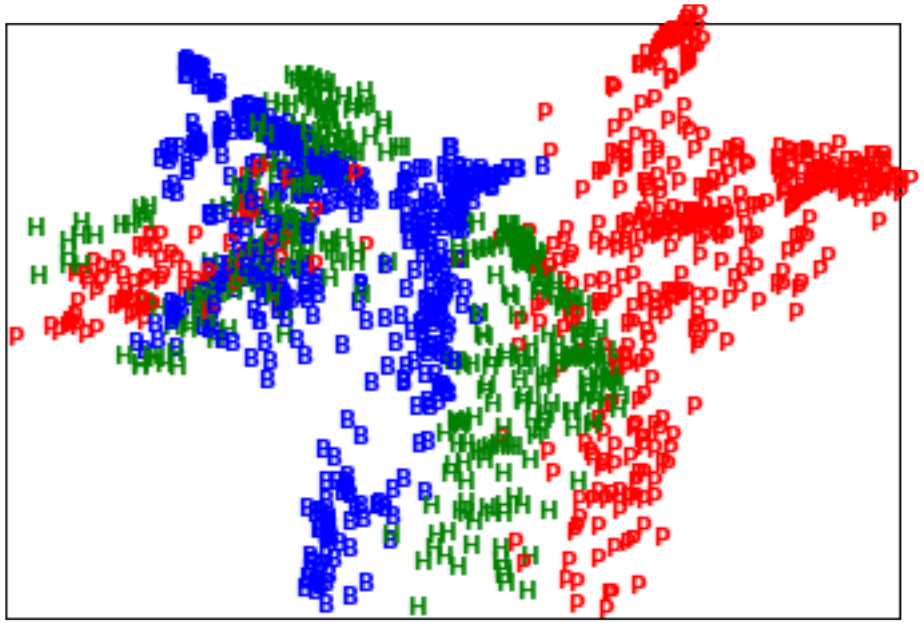


LDA

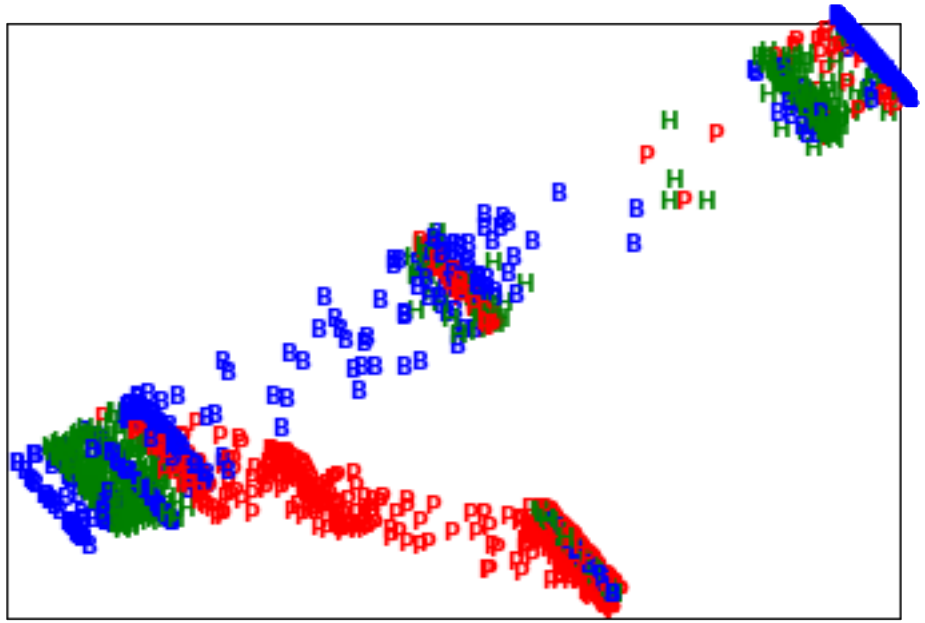


光学字符数据降维

Isomap

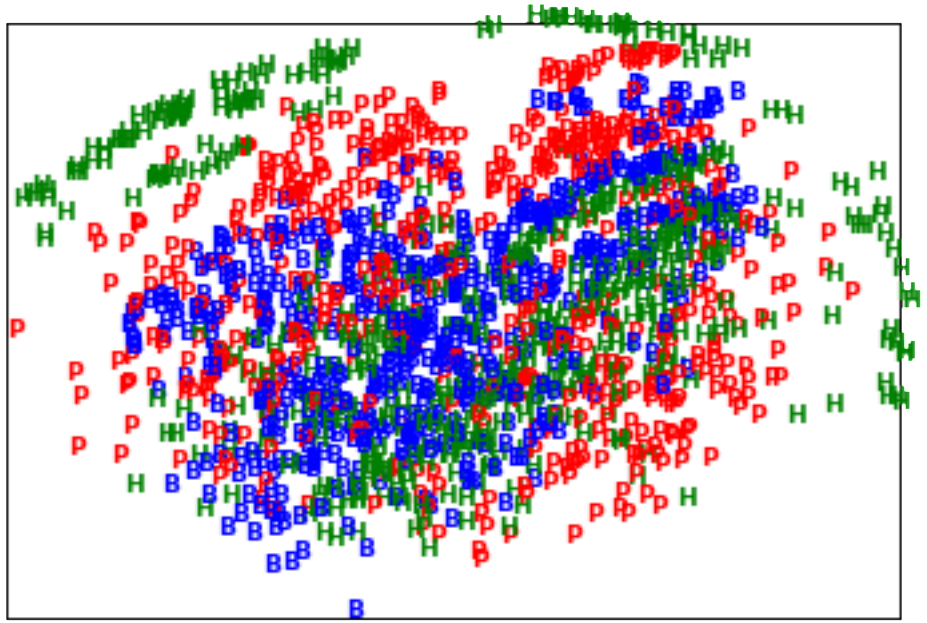


LLE

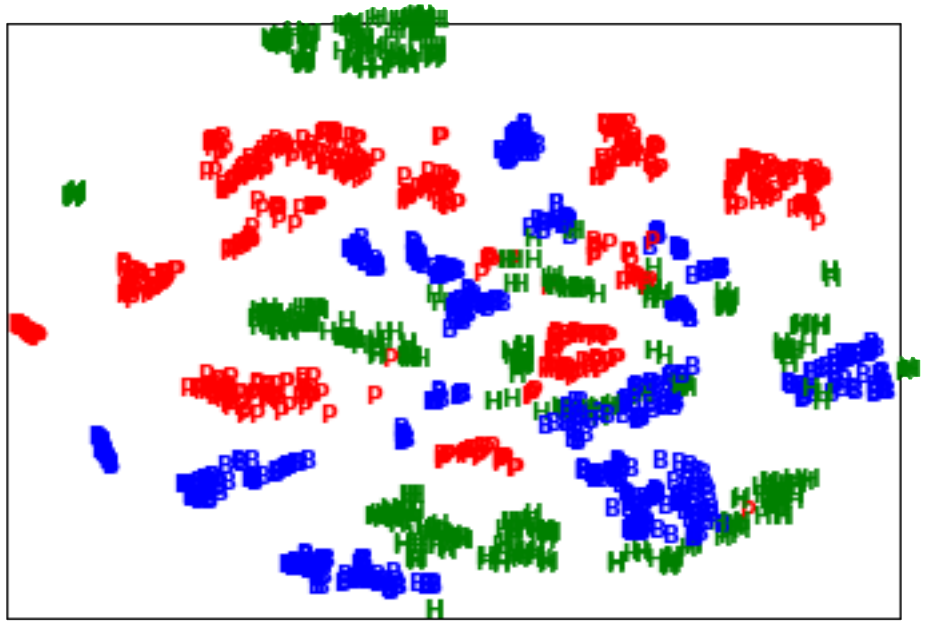


光学字符数据降维

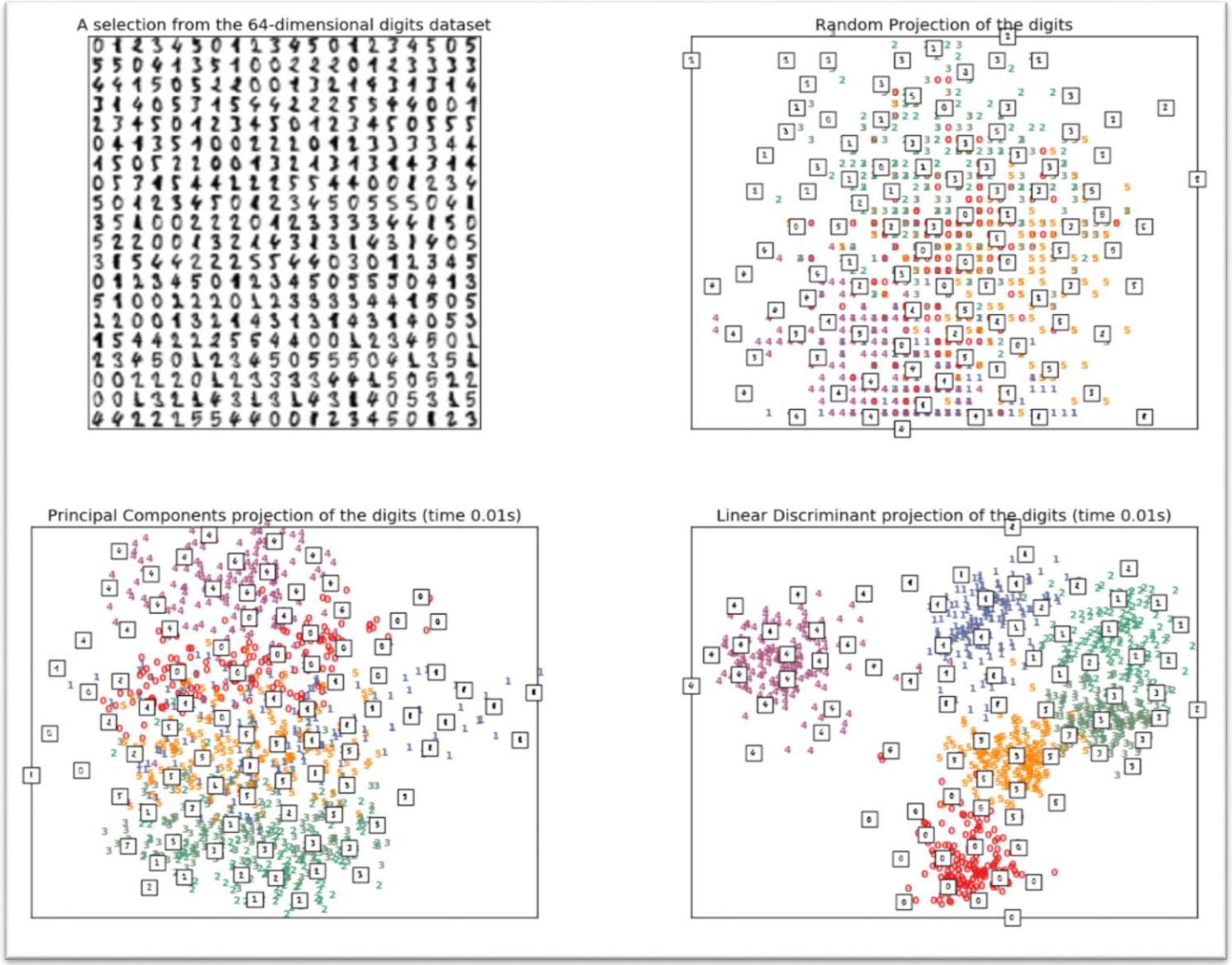
MDS



t-SNE



手写数字数据降维

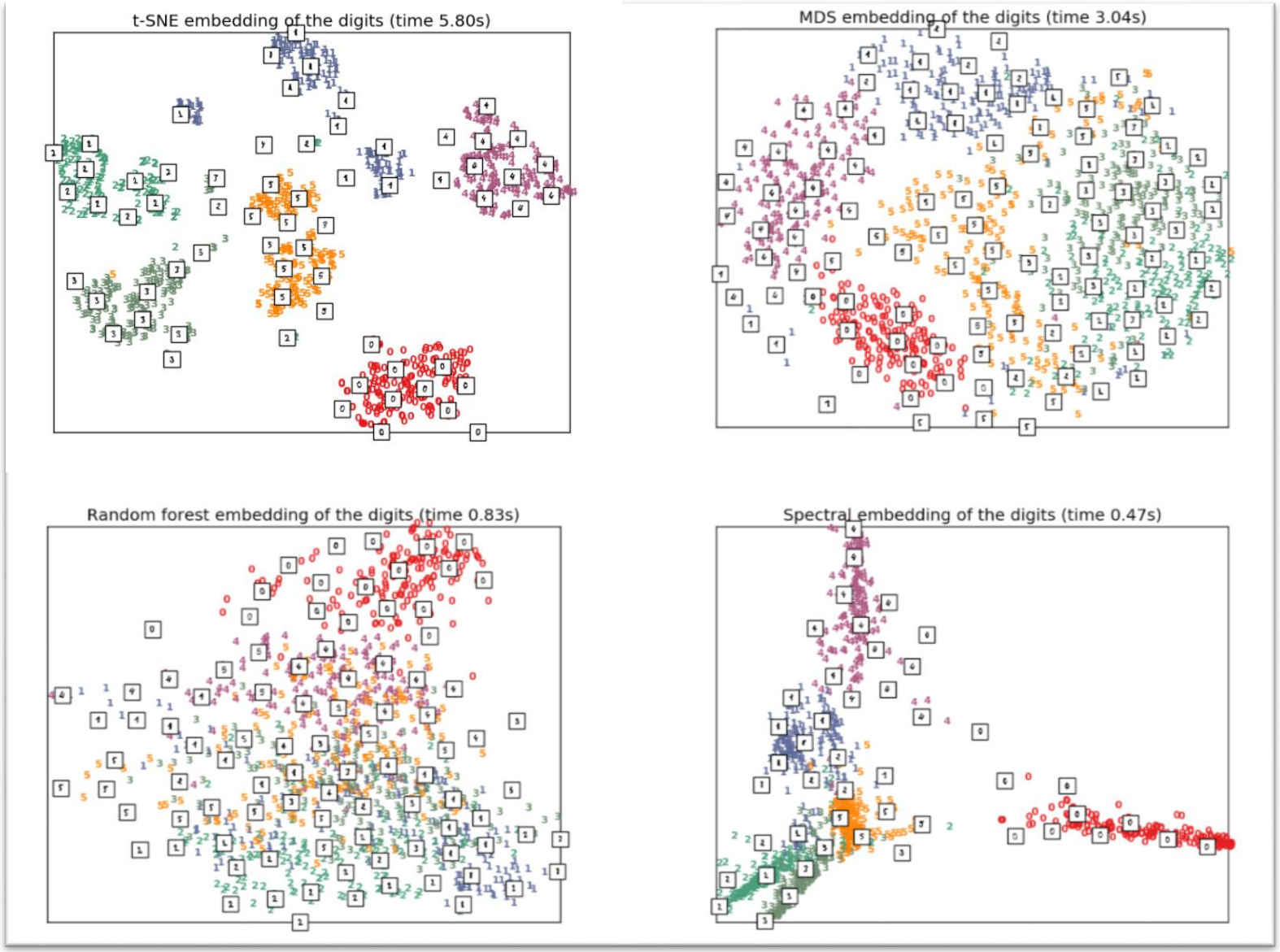


(Ref: scikit-learn-examples)

手写数字数据降维

(Ref: scikit-learn-examples)

手写数字数据降维



(Ref: scikit-learn-examples)

扫描二维码发现更多

数据酷客公众号

数据酷客官网