机器学习：无监督学习(Unsupervised Learning)之聚类(Clustering)，降维(Dimension Reduction)

机器学习：无监督学习(Unsupervised Learning)之聚类(Clustering)，降维(Dimension Reduction)
1. 复习：支撑向量机
1.1. 数学描述
1.2. 问题转化
1.3. 问题求解
1.4. 问题拓展
课程考核方法
2. 无监督学习简介
2.1. 概述
3. 聚类
3.1. $K-Means$ 算法
3.2. $Mean-Shift$ 算法
3.2.1. 均值漂移向量
3.2.2. 算法流程
4. 降维分析
4.1. PCA 主成分分析
4.2. ED 特征值分解
4.3. SVD 奇异值分解(用于求解 PCA 中 w 的特征值)
5. 降维分析-流形学习($Manifold\ Learning$)
5.1. LLE (局部线性嵌入，$Locally\ Linear\ Embedding$)
5.2. SNE (随机近邻嵌入，$Stochastic\ Neighbor\ Embedding$)
5.3. LE (谱嵌入，拉普拉斯特征映射 $Laplacian\ Eigenmap$)
5.3.1. 归一化拉普拉斯矩阵
5.3.2. 流形降维 - 拉普拉斯映射

---

1. 复习：支撑向量机

目标：找到"最佳"分类面，分割平面。

意义：新样本可以进行预测。

1.1. 数学描述

样本：$(x^{(i)}, y^{(i)}) \quad i = 1, \cdots m$ 。

其中 $x^{(i)} \in \mathbb{R}^n,\ y^{(i)} \in \{-1, +1\}$ 。

参数化模型 $f(w, b) = w^T X + b$ 。

1.2. 问题转化

从二维点到线的距离得到启发，定义集合间隔 $\gamma^{(i)}$ ，得到优化问题。

难求解的 $\max\limits_{w,b} \min\limits_{i} \gamma^{(i)}$ $\Leftrightarrow$ 好求解的有约束的二次规划问题。

1.3. 问题求解

凸优化问题（拉格朗日乘子）

利用 $Slater$ 条件 $\Rightarrow d^* = p^*$ 。

利用 KKT 条件求解此问题 $(w^*,\ b^*)$ 。

1.4. 问题拓展

线性不可分：特征映射。

计算复杂度高，通过核方法求解，核方法也可以降低复杂度。

课程考核方法

1.抄笔记 30% (群里要求)

2.加入 QQ 群 10%

3.上课奖励分 5%

4.大作业 60%：

给定两个数据集(下周二， 4月13日)，监督学习的问题。

上交程序，和两页以内的报告($Latex$)。

- 标题
- 作者，联系方式，单位
- 摘要(100 字以内)
- 简介(文献引用)
- 结果(公式，表格，图片)
- 结论
- 参考文献

一定不能抄袭！！！ 认真试了都行。

时间节点：5月15日， $23:59$ 。

可以加一些对课程的建议。

2. 无监督学习简介

常见的一般有两种，数据的无监督聚类的分类算法，以及数据的无监督降维算法。

$K-means$ (K 均值聚类算法)，$Mean-Shift$ 均值漂移聚类算法，PCA 主成分分析(降维)，流形学习(降维)之 LLE, SNE, LE 。

2.1. 概述

定义：在监督学习和强化学习中，典型的任务是分类和回归，且需要使用到人工预先准备好的范例。但是无监督学习中，不需要人力来输入标签，或者说，用于训练的数据本身是没有标签的。

意义：

- 根据食物的本身属性去分辨事物。无监督学习过程中，训练样本的标记信息是未知的，无监督学习可以通过对无标记训练样本的学习来解释数据的内在性质和规律，为进一步的数据分析提供基础。
- 与监督学习的方法结合，作半监督学习。
- 用于神经网络的隐藏层的感知函数定义。

3. 聚类

物以类聚，人以群分。

方法：怎么去聚类?

定义距离的度量方式：一维平面上： $d(x_i, x_j) = |x_i - x_j|$ ，二维可用欧氏距离。

3.1. $K-Means$ 算法

很典型的基于距离的聚类算法，采用距离作为相似性的评价指标，即认为两个对象的距离越近，其相似度就越大。该算法认为簇是由距离靠近的对象组成的，因此把得到紧凑且独立的簇作为最终目标。

假定输入样本为 $S = x_1, x_2, \cdots, x_m$ ，则算法步骤为：

1、选择初始的 k 个类别中心 $\mu_1 \mu_2 \cdots \mu_k$ 。

2、对于每个样本 x_i ，将其标记为距离类别中心最近的类别，即：

$$label_i = \arg\min_{1 \le j \le k} ||x_i - x_j||$$

3、将每个类别中心更新为隶属该类别的所有样本的均值：

$$\mu_j = \frac{1}{|c_j|} \sum_{i \in c_j} x_i$$

4、重复最后两布，直到类别中心的变化小于某阈值。

5、终止条件：迭代次数 / 簇中心变化率 / 最小平方误差 MSE 。

目标函数及其求解：

记 K 个簇中心为 $\mu_1\mu_2,\cdots,\mu_k$ ，每个簇的样本数目为 $N_1,N_2,\cdots,N_k$ 。

使用平方误差作为目标函数：

$$J(\mu_1,\mu_2,\cdots,\mu_k=\frac{1}{2}\sum_{j=1}^{K}\sum_{i=1}^{N_j}||x_i-\mu_j||^2$$

N_j 是给定的中心点 j 附近样本点的个数。

对关于 $\mu_1,\mu_2,\cdots,\mu_k$ 的函数求偏导，其驻点为：

$$\frac{\partial J}{\partial \mu_j}=-\sum_{i=1}^{N_j}(x_i-\mu_j)\rightarrow 0\Rightarrow\mu_j=\frac{1}{N_j}\sum_{i=1}^{N_j}x_i$$

3.2. $Mean-Shift$ 算法

$Mean-Shift$ 均值漂移算法是一个典型的无监督学习算法。和 $K-Means$ 根据距离度量来标记样本所属的最近类型，并通过迭代反复更新多个类别中心的思想不同；MS 算法是计算区域内的漂移向量，通过漂移向量更新点和对应区域的位置，最后再计算新的区域里的漂移向量，循环往复，直到找到概率密度函数的极大值点。所以MS 算法常应用于数据聚类，图像分割，目标跟踪，以及概率密度函数估计。

3.2.1. 均值漂移向量

MS 算法是Lesson 3.5 参数估计(MLE, MAP, Bayes, KNN, Parzen, GMM, EM算法)的核密度估计和我们接下来要介绍的梯度上升法结合的算法。寻找概密函数的极大值点即为寻找核密度函数的极大值点，求函数的极大值可以采用梯度上升的方法来优化。

根据Lesson 3.5 参数估计(MLE, MAP, Bayes, KNN, Parzen, GMM, EM算法)的内容，给定核函数 $K(x)$ ，在任一点 x 处的概率密度函数的估计值根据所有的样本点计算：

$$p(x)=\frac{1}{nh^d}\sum_{i=1}^{n}K(\frac{x-x_i}{h})$$

h 是核函数的窗口半径，是人工设定的正参数。核函数要保证函数值 $K(\frac{x-x_i}{h})$ 随着待估计点 x 离样本点 x_i 的距离增加而递减。核函数的剖面函数(核密度函数)可以写为：

$$f_{h,K}(x)=\frac{c_{k,d}}{nh^d}\sum_{i=1}^{n}k\left(\left|\left|\frac{x-x_i}{h}\right|\right|^2\right)$$

计算它的梯度值，由于：

$$\nabla_x k\left(\left|\left|\frac{x-x_i}{h}\right|\right|^2\right)=k'\left(\left|\left|\frac{x-x_i}{h}\right|\right|^2\right)\frac{2}{h^2}(x-x_i)$$

带回核密度函数，得到：

$$\nabla f_{h,K}(x) = \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n}(x - x_i)k'\Big(\Big|\Big|\frac{x - x_i}{h}\Big|\Big|^2\Big)$$

如果定义 $g(x) = k'(x)$ ，将 $-k'(x)$ 替换为 $g(x)$ ，可以得到：

$$\begin{aligned}
\nabla f_{h,K}(x) &= \frac{2c_{k,d}}{nh^{d+2}} \left(\sum_{i=1}^{n}\Big(g\Big(\Big|\Big|\frac{x - x_i}{h}\Big|\Big|^2\Big)x_i\Big) - \sum_{i=1}^{n}\Big(g\Big(\Big|\Big|\frac{x - x_i}{h}\Big|\Big|^2\Big)x\Big) \right) \\
&= \frac{2c_{k,d}}{nh^{d+2}} \left(\sum_{i=1}^{n}\Big(g\Big(\Big|\Big|\frac{x - x_i}{h}\Big|\Big|^2\Big) \frac{\sum_{j=1}^{n} g\Big(\Big|\Big|\frac{x - x_j}{h}\Big|\Big|^2\Big)}{\sum_{j=1}^{n} g\Big(\Big|\Big|\frac{x - x_j}{h}\Big|\Big|^2\Big)} x_i\Big) - \sum_{i=1}^{n}\Big(g\Big(\Big|\Big|\frac{x - x_i}{h}\Big|\Big|^2\Big)x\Big) \right) \\
&= \frac{2c_{k,d}}{nh^{d+2}} \left(\sum_{j=1}^{n} g\Big(\Big|\Big|\frac{x - x_j}{h}\Big|\Big|^2\Big) \left(\sum_{i=1}^{n} \frac{g\Big(\Big|\Big|\frac{x - x_i}{h}\Big|\Big|^2\Big)}{\sum_{j=1}^{n} g\Big(\Big|\Big|\frac{x - x_j}{h}\Big|\Big|^2\Big)} x_i \right) - \sum_{i=1}^{n}\Big(g\Big(\Big|\Big|\frac{x - x_i}{h}\Big|\Big|^2\Big)x\Big) \right) \\
&= \frac{2c_{k,d}}{nh^{d+2}} \Big[\sum_{i=1}^{n} g\Big(\Big|\Big|\frac{x - x_i}{h}\Big|\Big|^2\Big)\Big] \left[\frac{\sum_{i=1}^{n} x_i g(||\frac{x - x_i}{h}||^2)}{\sum_{i=1}^{n} g(||\frac{x - x_i}{h}||^2)} - x \right]
\end{aligned}$$

即：

$$\nabla f_{h,K}(x) = \frac{2c_{k,d}}{nh^{d+2}} \Big[\sum_{i=1}^{n} g\Big(\Big|\Big|\frac{x - x_i}{h}\Big|\Big|^2\Big)\Big] \left[\frac{\sum_{i=1}^{n} x_i g(||\frac{x - x_i}{h}||^2)}{\sum_{i=1}^{n} g(||\frac{x - x_i}{h}||^2)} - x \right]$$

这也就是均值漂移算法的核心迭代公式。$\sum_{i=1}^{n} g\Big(\Big|\Big|\frac{x - x_i}{h}\Big|\Big|^2\Big)$ 是一个正数，因为剖面函数 $k(x)$ 是一个减函数，因此 $g(x) = -k'(x) > 0$ 。

对于上式，如果使用核函数 G 的剖面函数 $g(x)$ ，那么，第一项就等于 $f_{h,G}$ ：

$$f_{h,G}(x) = \frac{c_{k,d}}{nh^{d}} \sum_{i=1}^{n} g\Big(\Big|\Big|\frac{x - x_i}{h}\Big|\Big|^2\Big)$$

第二项就相当于一个 $mean\ shift$ 均值漂移向量：

$$m_{h,G}(x) = \frac{\sum_{i=1}^{n} x_i g(||\frac{x - x_i}{h}||^2)}{\sum_{i=1}^{n} g(||\frac{x - x_i}{h}||^2)} - x$$

这是使用了核函数 G 进行加权之后的 x_i 和 x 之间的插值。上述式子就可以表示为：

$$\nabla f_{h,K}(x) = f_{h,G}(x) \frac{2c_{k,d}}{h^2 c_{g,d}} m_{h,G}(x)$$

利用梯度上升法优化，核心的迭代公式为：

$$x_{t+1} = x_t + \eta \cdot m_t$$

η 是步长系数，可表示为 $f_{h,G}(x)$ ，因为 $f_{h,G}(x)$ 本身也是个常数。m_t 即为第 t 次迭代时的均指向量 $m_{h,G}(x)$ 。

3.2.2. 算法流程

- a. 选择空间中 x 为圆心，以 h 为半径为半径，做一个高维球，采样落在所有球内的所有点 x_i 。
- b. 根据上面的式子计算 $m_{h,G}(x)$ 。

  如果 $m_{h,g}(x) < \varepsilon$ (人工设定)，退出程序。

  如果 $m_{h,G}(x) > \varepsilon$, 则利用迭代公式计算 $x_{t+1} = x_t + \eta \cdot m_t$ ，返回 a。

最后讨论一下 MS 算法的收敛问题，$m_{h,G}(x)$ 变形得到：

$$m_{h,G}(x) = \frac{1}{2}h^2 \frac{c_{g,d}}{c_{k,d}} \frac{\nabla f_{h,K}(x)}{f_{h,G}(x)}$$

这个式子表明，在 x 点处，用核函数 G 计算出的均值漂移向量正比于用 K 计算出的核密度函数梯度归一化后的值。因为概率密度函数的梯度值指向的是概率密度函数增加最快的方向，而均值向量又和这个梯度正比例相关，所以 MS 算法能保证每次迭代指向的也都是概率密度函数增加最快的方向。

4. 降维分析

降维简单解释：用少量的特征代替整体的特征。

概念化定义：采用某种映射方法，将原高维空间中的数据点映射到低维度空间中。降维的本质是学习一个映射函数 $f: x->y$ ，其中 x 是原始数据点的表达，目前最多使用向量表达形式。y 是数据点映射后的低维向量表达，通常 y 的维度小于 x 的维度(当然提高维度也是可以的，混合特征)。f 可能是显式的或隐式的，线性的或非线性的。

方法：PCA, LDA, LLE, SNE, LE ...

这其中，LDA 线性判别分析的降维方法是有监督学习，在Lesson 5 监督学习之分类(Perceptron, Fisher, Logistic, Softmax, Bayes)有详细讲到，这里不重复说明。

4.1. PCA 主成分分析

对于二维数据，数据点沿着 x 轴方向的投影即可视为其降维后的数据。

PCA 是线性降维方法，目标是通过某种线性投影，将高维数据映射到低维的空间中表示。

而我们希望找到一个最佳的投影方向，期望在所投影的维度上方差最大。即以此使用较少的数据维度，同时保住较多的原数据点的特性。

数学定义：一个正交化线性变换，把数据变换到一个新的坐标系统中，使得这一个数据的投影的第一大方差在第一个坐标(称为第一主成分)上，第二大方差在第二个坐标(第二大主成分)上，以此类推。

假设有样本：

$$x^{(i)}, i = 1, 2 \cdots, m \quad x_i \in \mathbb{R}^n$$
$$\text{目标 } f: x \to y \quad y \in \mathbb{R}^n$$

使得降维后在该维度上的方差最大。

定义：$X = [X^{(1)}, \cdots, X^{(m)}]$ 。

投影：

$$\begin{bmatrix} X_1^{(1)} & \cdots & X_1^{(m)} \\ X_2^{(1)} & \cdots & X_2^{(m)} \end{bmatrix} w = [1 \quad 0],\ w \in \mathbb{R}^{k \times n} \quad (k\text{是要降的维数}) X_p = w^T x$$

即希望找到 $w^* \in \mathbb{R}^{n \times k}$，使得 $\min\limits_{w} Var(w^T x)$ 。

不失一般性，假设 $\sum\limits_{j=1}^{n} x_j^{(i)} = 0$ 。

假设 X 零经验均值，则数据集 X 的第一主成分 W_1 可以定义为如下形式：

$$\begin{aligned} W_1 &= \arg\max_{||w||=1} Var\{W^T X\} \\ &= \arg\max_{||w||_2=1} E\{W^T X X^T W\} \\ &= \arg\max W^T X X^T W, \quad s.t\, W_1^T W_1 = 1 \end{aligned}$$

$W^T X X^T W$ 为投影完之后数据的方差。

求解的优化问题等价为：

$$\begin{aligned} &\Leftrightarrow \arg\max \frac{W_1^T X X^T W_1}{W_1^T W_1} \\ &\Leftrightarrow^{(\Phi \triangleq X X^T)} \arg\max \frac{W_1^T \Phi W_1}{W_1^T W_1} \end{aligned}$$

根据Lesson 5 监督学习之分类(Perceptron, Fisher, Logistic, Softmax, Bayes)瑞利商的定理推导，当 W_1 满足：

$$\Phi W_1 = \lambda_{\max}(\Phi) W_1 \text{ 时}$$

取最大值，即瑞利商：

$$\frac{W_1^T \Phi W_1}{W_1^T W_1} = \lambda_{\max}(\Phi)$$

求解 Φ 的特征值和特征向量可以用下面 4.3 讲到的 SVD 奇异值分解。

为了得到第 K 个主成分首先需要从数据集 X 中减去前 $K-1$ 个主成分，可以定义为如下形式：

$$\hat{X} = X - \sum_{i=1}^{k-1} W_i W_i^T X$$

从而可求得，则数据集 X 第 K 个主成分 W_k 可以定义为如下形式：

$$\begin{aligned} W_k &= \arg\max_{||w||=1} Var W^T \hat{X} \\ &= \arg\max_{||w||=1} E\{W^T X X^T W\} \end{aligned}$$

PCA 核心就是选择合适的投影方向，将数据从高维降到低维。降维后可以用 $K-Means$ 或者 SVM 等算法把两类值分开。

PCA 的优点：

- 可消除特征之间的相关影响。
- 可减少指标选择的工作量。
- 可进行将未处理降低数据维度。
- 完全无参数限制。

PCA 缺点：

- 无法通过参数化等方法对处理过程进行干预。
- 在非高斯分布的情况下，选取的结果不一定是最优的。

拓展：

可以将之前的 SVM 中用到的核函数运用到 PCA 中，形成核 PCA 方法，对于数据进行非线性降维。

如果期望不以方差作为数据信息的衡量因素，而希望分析出数据的主要影响因子亦或是分析出数据的独立成分则可用 $Factor\ Analysis$ (因子分析)，ICA (独立成分分析)等方法。

4.2. ED 特征值分解

方阵 A 的特征值：$Av = \lambda v$ 。

如果矩阵 A 有 n 个线性无关的向量，矩阵 Q 可逆，其相似对角化：$Q^{-1}AQ = \Lambda$

方阵 A 的特征值分解

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

Q 的列为矩阵 A 的特征向量组成的矩阵，Λ 为一个对角矩阵，对角线上的元素为对应特征向量的特征值。

一个 n 阶矩阵可以进行特征值分解的充要条件是它有 n 个线性无关的特征向量。通常这些特征向量 v_i 都是单位化的。

- 可用于求逆，$A^{-1} = Q\Lambda^{-1}Q^{-1}$ 。Λ 的逆矩阵很好计算，为主对角线元素的倒数
- 可用于计算多项式，设 $f(x) = a_n x^n + \cdots + a_1 x$ 。

$$f(A) = f(Q\Lambda Q^{-1}) = Qf(\Lambda)Q^{-1}$$

特别的，有 $A^n = Q\Lambda^n Q^{-1}$ 。

如果 A 是实对称矩阵，则正交化($Q^TAQ = \Lambda$)后，$A = Q\Lambda Q^T$ 。

4.3. SVD 奇异值分解(用于求解 PCA 中 w 的特征值)

奇异值分解可以用于 PCA 问题的求解：

$$w_1 = \arg\max_{||w_1||=1} w_1^T XX^T w_1$$

根据Lesson 5 监督学习之分类(Perceptron, Fisher, Logistic, Softmax, Bayes)瑞利商的推导，该最大化问题可转换为求 X^TX 的最大的广义特征值和对应的广义特征向量，最大特征值就是最优解 w 。而求解该问题，需要让 XX^T 作奇异值分解。

4.2. 介绍的特征值分解有很多局限，比如说变幻的矩阵**必须是方阵**。

奇异值分解是一个能使用于任意矩阵的一种分解的方法。其思路是，对 A^TA 和 AA^T 进行特征值分解(这两个矩阵有相同的非 0 特征值)，(A^TA) 是一个方阵，可分解为 $Q\Sigma Q^{-1}$ 。

奇异值分解原理：

$$(A^TA)v_i = \lambda_i v_i$$

设 $A \in R^{m\times n}$，其中 $m \geqslant n$ ，则有：

$$U^TAV = \Sigma$$

U 为 m 阶正交矩阵，其列为矩阵 A 的左奇异向量，也是 AA^T 的特征向量。

V 为 n 阶正交矩阵，其行为矩阵 A 的右奇异向量，也是 A^TA 的特征向量。

Σ 为如下形式的 $m \times n$ 阶矩阵。

$$\Sigma = \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \sigma_n \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots \end{pmatrix} = \begin{pmatrix} \Sigma_n \\ 0_{(m-n)\times n} \end{pmatrix}$$

其中 σ_i 为 A 的奇异值，是 AA^T 特征值的非负平方根，也是 A^TA 特征值的非负平方根： $\sigma_i = \sqrt{\lambda_i}$ 。式中， λ_i 是 A^TA 和 AA^T 的特征值，即 A 的奇异值 $= (A^TA)$ 的特征值。

对于 $U^TAV = \Sigma$ ，前式左乘 U ，右乘 V^T ，由于 U, V 均为正交矩阵，则有

$$A = U\Sigma V^T$$

此即为矩阵的奇异值分解的表达式。

$$A^TA = V\Sigma^T\Sigma V^T \quad \text{此即为} A^TA \text{的特征值分解}$$
$$AA^T = U\Sigma\Sigma^TU^T \quad \text{此即为} AA^T \text{的特征值分解}$$

如果 A 是对称矩阵($A^TA = AA^T = AA$)，则 A^TA 和 AA^T 的特征值分解相同，这意味着 U, V 相同，A 的奇异值为其特征值的绝对值 $\sigma = \sqrt{\lambda^2} = |\lambda|$ 。

最终得到了 Σ 矩阵，通过 Σ 矩阵，就能得到对应的 AA^T 的特征值和对应的特征向量，也就能得到 PCA 的投影结果。

例子：假设 PCA 对应的训练数据为 $A = \begin{bmatrix} -1 & 3 \\ 3 & 1 \\ 1 & 1 \end{bmatrix}$ 。求解第一主成分的目标函数可表示为：$w_1 = \arg\max_{||w_1||=1} w_1^TAA^Tw_1$ ，需要求解 w_1 的最优值作为数据降维($y = w^TX$)的特征矩阵。

解：根据广义瑞利商，当 $AA^Tw_1=\lambda_{\max}(AA^T)w_1$ 时，上式取最大值，此时 w_1 取最优解。又因为 A 不是一个方阵，无法做特征值分解，因此该优化问题可以转为求解 AA^T 的特征值和对应的特征向量，即 A 的奇异值分解问题。

$$AA^T=\begin{pmatrix}10&0&2\\0&10&4\\2&4&2\end{pmatrix}\quad A^TA=\begin{pmatrix}11&1\\1&11\end{pmatrix}$$

对 AA^T,A^TA 做特征值分解，AA^T 的特征值为 $12,10,0$ ，A^TA 的特征值为 $12,10$ 。因此 A 的非零奇异值为 $\sigma_1=\sqrt{12},\sigma_2=\sqrt{10}$ 。

计算 AA^T,A^TA 的特征向量并单位化，可以得到：

$$U=\begin{pmatrix}\frac{1}{\sqrt{6}}&\frac{2}{\sqrt{6}}&\frac{1}{\sqrt{6}}\\\frac{2}{\sqrt{5}}&-\frac{1}{\sqrt{5}}&0\\\frac{1}{\sqrt{30}}&\frac{2}{\sqrt{30}}&-\frac{5}{\sqrt{30}}\end{pmatrix}\quad V=\begin{pmatrix}\frac{1}{\sqrt{2}}&\frac{1}{\sqrt{2}}\\\frac{1}{\sqrt{2}}&-\frac{1}{\sqrt{2}}\end{pmatrix}$$

做奇异值分解，可以得到：

$$U^TAV=\begin{pmatrix}\sqrt{12}&0\\0&\sqrt{10}\\0&0\end{pmatrix}$$

则 $\lambda_{\max}=\sqrt{12}$ ，代回 $AA^Tw_1=\lambda_{\max}w_1$ 求解该方程，即可得到最终需要求解的 w_1 。

5. 降维分析-流形学习($Manifold\ Learning$)

流形学习的思想：

流形是集合中的一个概念，它是高维空间中的低维几何结构，例如三维空间的球面就是一个二维流形，给定半径之后，其方程可以用两个参数(经纬度)来比表示。可以简单将流形理解成曲线、曲面在高维空间的推广。

假设数据是均匀采样于一个高维欧氏空间中的低维流形。流形学习就是从高维采样数据中恢复低维流形结构，并求出相应的嵌入映射，以实现维数约简或者数据可视化，它是从观测到的现象中去寻找事物的本质或者内在规律。

对于降维，流型学习要保证降维之后，数据同样满足与高维空间中流形有关的几何约束关系。

与 PCA 不同，它在高维空间中要发现低维结构。常见的有局部线性嵌入，随机近邻嵌入和谱嵌入等算法。

5.1. LLE (局部线性嵌入，$Locally\ Linear\ Embedding$)

PCA 无法实现三维 S 形状或者滚筒分布的数据。

而 LLE (局部线性嵌入)对于一个流形数据学习的结果能得到更好的结果。

局部线性嵌入是一种非线性降维算法，它能够使降维后的数据较好地保持原有的流型结构，是一种最经典的流型学习算法。它的核心思想是，每一个高维数据点都可以由近邻点的线性加权组合构造得到，降维后低维空间需要保持同样的线性加权关系。

ps：但是有些情况下它也并不适用。如果数据分布在整个封闭的球面上，LLE 则不能将它映射到二维空间，且不能保持原有的数据流型。那么我们在处理数据中，首先假设数据不是分布在闭合的球面或者椭球面上。

思想：首先假设数据在较小的局部是线性的(局部线性性)，也就是说，某一个数据可以由它邻域中的几个样本来线性表示。

$$x_0 = w_1x_1 + w_2x_2 + w_3x_3$$

目标：降维后上述等式依然近似成立。

$$\tilde{x}_0 = w_1\tilde{x}_1 + w_2\tilde{x}_2 + w_3\tilde{x}_3$$

其中 x_i 为对应 x_i 降维后的数据点。

1. 首先要确定邻域大小的选择。假设这个值为 k 。我们可以通过和 KNN 一样的思想通过距离度量比如说欧氏距离来选择某样本的 k 个最近邻。

$$N_i = KNN(x_i, k), \quad N_i = [x_{i1}, \cdots, x_{ik}]$$

2. 确定高维空间中的局部线性权重。

   $Loss\ function$:

$$J(W) = \sum_{i=1}^{N} \left|\left| x_i - \sum_{j=1}^{k} w_{ij}x_{ij} \right|\right|_2^2, \quad s.t \sum_{j=1}^{k} w_{ij} = 1$$

3. 矩阵化 $Loss\ function$

$$J(W) = \sum_{i=1}^{N} \left|\left| x_i - \sum_{j=1}^{k} w_{ij}x_{ij} \right|\right|_2^2 = \sum_{i=1}^{N} \left|\left| \sum_{j=1}^{k} w_{ij}x_i - \sum_{j=1}^{k} w_{ij}x_{ij} \right|\right| = \sum_{i=1}^{N} \left|\left| \sum_{j=1}^{k} w_{ij}(x_i - x_{ij}) \right|\right|_2^2$$

$$= \sum_{i=1}^{N} ||(x_i - x_{ij})W_i|| = \sum_{i=1}^{N} W_i^T(x_i - x_{ij})^T(x_i - x_{ij})W_i$$

   其中，$W_i = [w_{i1}, w_{i2}, cdots, w_{ik}]^T \in R^{k\times 1}$ 。

4. 矩阵化 $Loss\ function$

$$J(W) = \sum_{i=1}^{N} W_i^T(x_i - x_{ij})^T(x_i - x_{ij})W_i = \sum_{i=1}^{N} W_i^T Z_i W_i$$

   而约束 $\sum_{j=1}^{k} w_{ij} = 1$ 可化为：

$$\sum_{j=1}^{k} w_{ij} = W_i^T \mathbf{1}_{1\times k} = 1$$

5. 拉格朗日乘数法求解 W ：

$$L(W) = \sum_{i=1}^{N} W_i^T Z_i W_i + \lambda(W_i^T \mathbf{1}_{1\times k} - 1)$$

令 $Z_i = (x_i - x_{ij})^T (x_i - x_{ij})$ 。

对 $L(W)$ 求 W 的导数并令其为 0：

$$W_i = -\frac{1}{2}\lambda Z_i^{-1} \mathbf{1}_{1\times k}$$

总结：即输入 $X = \{x_1, x_2, \cdots, x_N\}$ 。最小化 $Loss\ Function: J(W)$ 得到结果权重 $W = [W_1, W_2, cdots, W_N] \in R^{k\times N},\ W_i = [w_{i1}, , w_{i2}, \cdots, w_{ik}]^T,\ W_i \in R^{k\times 1}$ 。(求解方式：化成矩阵形式和拉格朗日乘子法来求解这个最优化问题)

6. 数据映射到低维空间，求解函数：

$$J(Y) = \sum_{i=1}^{N} \left|\left| y_i - \sum_{j=1}^{k} w_{ij} y_{ij} \right|\right|_2^2, \quad s.t \sum_{i=1}^{N} y_i = 0,\ \sum_{i=1}^{N} y_i y_i^T = N I_{d\times d}$$

其中低维空间向量 $Y = [y_1, y_2, \cdots, y_N]^T, d \times N$ 。即 $X \to W \to Y$ ， X, Y 都可以写成邻居的线性组合。

7. 目标损失函数矩阵化：

$$J(Y) = \sum_{i=1}^{N} \left|\left| y_i - \sum_{j=1}^{k} w_{ij} y_{ij} \right|\right|_2^2 = \sum_{i=1}^{N} ||YI_i - YW_i||_2^2 = tr(Y^T (I-W)^T (I-W) Y)$$

其中， $w_{ij} = \begin{cases} = 0, \text{ 非}y_i\text{临近的}y_{ij} \\ \neq 0, \ y_i\text{临近的}y_{ij} \end{cases}$ ，从而 $W_i = [w_{i1}, \cdots, w_{iN}] \in R^{N\times 1}$。

对于非 y_i 临近的 y_{ij} ，令其对应 $w_{ij} = 0$ 。

令 $M \triangleq (I - W^T (I - W)$ 为每个样本点的连接矩阵。($(W \in R^{N\times N})$)

8. 同之前的步骤一样，再次利用拉格朗日乘数法：

$$L(Y) = tr(Y^T M Y) + \lambda(Y^T Y - NI)$$

对 $L(Y)$ 求 Y 的导数并令其为 0：

$$MY = -\lambda Y$$

Y 其实是 M 的特征向量构成的矩阵，为了将数据降到 d 维，我们只需要取 M 的最小的 d 个非零特征值对应的特征向量，而一般第一个最小的特征值接近 0 (由于 M 的最小特征值为 0 不能反映数据特征，此时对应的特征向量为全 1 ，谱嵌入算法中也有类似的分析)，我们将其舍弃，取前 $[2, d+1]$ 个特征值对应的特征向量。

不同假设下，求不同的优化问题，得到类似的优化过程。

即先找到关系的线性系数矩阵，且保留线性矩阵的线性关系，根据 W ，分解 M 最后降维得到 Y 。

思路总结：LLE 算法认为每一个数据点都可以由近邻点的线性加权组合构造得到。算法的主要步骤如下：

1. 寻找每个样本点的 k 个近邻点。
2. 由每个样本点的近邻点计算出该样本点的局部重建权值矩阵。

3. 由该样本点的局部重建权值矩阵和其近邻点计算出该样本点的输出值。

5.2. SNE (随机近邻嵌入，$Stochastic\ Neighbor\ Embedding$)

随机近邻嵌入主要是基于Lesson 7 信息论与决策树中的 KL 散度来衡量两个概率分布之间的差异。它将向量组 $x_i, i=1,\cdots,l$ 变换降维到低维空间 $y_i, i=1,\cdots,l$ 。要求变换之后的向量组保持原始向量组在高维空间中的某些几何结构信息。

SNE 主要基于如下思想：高维空间中距离很近的点投影到低维空间之后也要保持这种近邻关系，这种关系可通过概率体现。假设高维空间中 x_j 以 $p_{j|i}$ 的概率成为 x_i 的邻居，可将样本之间的欧氏距离转为概率值，借助于正态分布，概率的计算公式可以写为：

$$p_{j|i}=\frac{\exp(-||x_i-x_j||^2)/2\sigma^2}{\sum_{k\neq i}\exp(-||x_i-x_k||^2)/2\sigma^2}$$

σ_i 表示以 x_i 为中心的正态分布的标准差，这个概率计算公式类比于 $Softmax$ 回归。除以分母是归一化为概率。由于不关心一个点与自身的相似度，$p_{i|i}=0$ 。投影到低维后也要保持这个关系，假设 x_i, x_j 的低维映射 y_i, y_j 的近邻概率记为 $q_{i|j}$ ，标准差设为 $\frac{1}{\sqrt{2}}$ ，即：

$$q_{j|i}=\frac{\exp(-||y_i-y_j||^2)}{\sum_{k\neq i}\exp(-||y_i-y_k||^2)}$$

上面的定义是点 x_i 和它一个邻居以及低维映射的 y_i 和一个邻居的关系。如果考虑所有点，这些概率值构成了一个离散分布 p_i ，即所有样本点成为 x_i 邻居的概率，这也是一个多项分布。低维空间对应的是 q_i 。降维的目标是让这两个概率分布尽可能接近，因此需要很亮两个概率分布之间的差距，可以通过 KL 散度衡量，即目标为最小化如下函数：

$$\min L(y_i)=\min\sum_{i=1}^{l}D_{KL}(p_i|q_i)=\min\sum_{i=1}^{l}p_{j|i}\ln\frac{p_{j|i}}{q_{j|i}}$$

把上面两个概率计算公式带入 KL 散度，既可以得到关于 y_i 的函数。求解上式的极小值，即可以得到 x_i 降维后的结果 y_i 。

5.3. LE (谱嵌入，拉普拉斯特征映射 $Laplacian\ Eigenmap$)

这部分涉及一点点图论的内容，下面文字中的"图"都是特指 $Graph$ ，不是图像的意思。

5.3.1. 归一化拉普拉斯矩阵

图 G 的邻接矩阵 W：假设图有 n 个结点，则 $W_{n\times n}$ 的每个元素 w_{ij} 表示边 (i,j) 的权重。如果两点没有边连接，则 W 对应的元素为 0 。无向图的邻接矩阵都是对称矩阵。

图 G 的加权度矩阵 D： 它是一个对角矩阵，其主对角线上的元素为每个顶点的加权度，即 $d_{ii}=d_i=\sum_{j=1}^{n}w_{ij}$ 。如果图中存在孤立结点，则加权度矩阵为奇异矩阵(非满秩)。

图 G 的拉普拉斯矩阵 L： 定义为加权度矩阵和邻接矩阵之差。

$$L=D-W$$

拉普拉斯矩阵 L 的性质：

- 对任意向量 $f \in \mathbb{R}^n$ 有：

$$f^T L f = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} (f_i - f_j)^2$$

- L 是对称半正定矩阵。
- L 的最小特征值为 0 ，其对应的特征向量为常向量 1 ，所有分量为 1 。
- L 有 n 个非负实数特征值，并且满足：

$$0 = \lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_n$$

L 不依赖于邻接矩阵 W 的主对角线元素。除主对角线元素之外，其他位置的元素都相等的各种不同的矩阵 W 都有相同的拉普拉斯矩阵。因此，图中的自环不影响其 L 矩阵。

重要结论：假设 G 是一个有非负权重的无向图，其 L 的特征值 0 的重数 k 等于图的连通分量的个数(参考数据结构定义：连通图表示图中的任何两点都存在路径，图的连通分量就是图的极大连通子图)。假设图的连通分量为 $A_1, \cdots, A_k$ ，则特征值 0 的特征空间由这些连通分量对应的向量 $1_{A_1}, \cdots, 1_{A_k}$ 组成。

图 G 的归一化拉普拉斯矩阵：有两种形式的归一化。

第一种为对称归一化：

$$L_{sym} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$$

$D^{\frac{1}{2}}$ 是 D 的所有元素的正平方根得到的，$D^{-\frac{1}{2}}$ 是其逆矩阵，也就是 $D^{\frac{1}{2}}$ 对角线元素的倒数(对角矩阵相关结论)。

第二种为随机漫步归一化：

$$L_{rw} = D^{-1} L = I - D^{-1} W$$

L_{sym}, L_{rw} 的性质：

- 对任意向量 $f \in \mathbb{R}^n$ 有：

$$f^T L_{sym} f = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} (\frac{f_i}{\sqrt{d_{ii}}} - \frac{f_j}{\sqrt{d_{jj}}})^2$$

- 当前仅当 λ 是 L_{sym} 的特征值，并且特征向量为 $w = D^{\frac{1}{2}} u$ 时，λ 是 L_{rw} 的特征值，u 是对应的特征向量。
- 当前仅当 λ, u 是 $Lu = \lambda D u$ 这一广义特征值问题的解时，λ 是 L_{rw} 的特征值，u 是对应的特征向量。
- 0 是 L_{rw} 的特征值，对应的特征向量为常向量 1 。0 是 L_{sym} 的特征值，对应的特征向量为 $D^{\frac{1}{2}} 1$ 。
- L_{sym}, L_{rw} 都是半正定矩阵，有 n 个非负实数特征值，并且满足：

$$0 = \lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_n$$

与拉普拉斯矩阵 L 类似的重要结论：假设 G 是一个有非负权重的无向图，其归一化拉普拉斯矩阵 L_{sym}, L_{rw} 的特征值 0 的重数 k 等于图的连通分量的个数。假设图的连通分量为 $A_1, \cdots, A_k$ ，对于矩阵 L_{rw} ，特征值 0 的特征空间由这些连通分量对应的向量 $1_{A_1}, \cdots, 1_{A_k}$ 组成；对于矩阵 L_{sym} ，特征值 0 的特征空间由这些连通分量对应的向量 $D^{\frac{1}{2}}1_{A_1}, \cdots, D^{\frac{1}{2}}1_{A_k}$ 组成。

5.3.2. 流形降维 - 拉普拉斯映射

$Laplacian\ Eigenmap$ 利用了图论的思想，主要思想是，为样本点构造带权重的图，然后计算图的拉普拉斯矩阵，对该矩阵进行特征值分解，得到投影的结果。这个结果对应于将样本点投影到低维空间，且保持了其在高维空间中的相对距离信息。

假设训练集为不带标签的样本点 $x_1, \cdots, x_n \in \mathbb{R}^D$ ，降维的目标是映射到 $y_1, \cdots, y_n \in \mathbb{R}^d$ ，其中 $d \ll D$ 。假设 $x_1, \cdots, x_k \in M$ ， M 为一个嵌入 $\mathbb{R}^D$ 空间里的流形。

对于这组数据点，首先需要根据欧氏距离度量或者近邻算法，构造了带权重的图(样本集的相似度图)，并计算三个图矩阵 W, D, L。假设图是连通的，不连通也可以将算法分别作用到各连通分量上。目标是将这组向量映射到一维直线上，且保证在高维空间中相邻的点，映射之后也要距离尽可能近。因此，目标函数可定义为：

$$\min_y \sum_{i=1}^n \sum_{j=1}^n (y_i - y_j)^2 w_{ij}$$

这个损失函数意味着，如果高维空间中 x_i, x_j 很近， w_{ij} 就会很大，那么 y_i, y_j 也就必须尽可能近，否则损失值会很大。如果高维空间中 x_i, x_j 很远， w_{ij} 就会很小，那么 y_i, y_j 很远，也不会导致很大的损失。

根据拉普拉斯矩阵中的性质， $f^T L f = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij}(f_i - f_j)^2$ ，可以得到：

$$\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (y_i - y_j)^2 w_{ij} = y^T L y$$

同时我们可以添加一个约束条件 $y^T D y = 1$ 消除 y 的缩放冗余，因为 y, ky 本质上是一种降维投影的结果。D 提供了一个对图顶点的衡量，d_{ii} 过大，则其对应的第 i 个顶点提供的信息越多，这也符合我们的直观认识，即一个顶点的所有边的总权重越大，其在图中的作用(信息)也就越大。

因此上面的最优化问题可以转换为：

$$\min_y y^T L y \quad s.t.\ y^T D y = 1$$

同样利用拉格朗日乘子函数求解：

$$L(y, \lambda) = y^T L y + \lambda(y^T D y - 1)$$

对 y 求梯度并令其为 0 ，可以得到： $Ly = -\lambda D y$ 。同时左乘 D^{-1} 可以得到：

$$D^{-1} L y = \lambda y$$

整个求解过程和Lesson 5 监督学习之分类(Perceptron, Fisher, Logistic, Softmax, Bayes)的瑞利商其实是类似的。上式也是随机漫步归一化拉普拉斯矩阵的特征值求解问题。由于要最小化 $y^T L y$ ，因为 0 对应的为常向量 1 ，投影后坐标均为 1 ，无有用信息，因此最终的算法输出，是求除了 0 之外的一个最小的特征值对应的特征向量。

将算法从一维直线推广到高维，假设将向量投影到 d 维空间，降维结果是一个 $n \times d$ 的矩阵，即为 $Y = [y_1\ y_2\ cdots\ y_d]$，其第 i 行为第 i 个样本点投影后的坐标。算法的目标函数为：

$$\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}||y_i - y_j||^2 w_{ij} = tr(Y^T L Y)$$

$tr(\cdot)$ 表示矩阵的迹。这等价于下面的优化问题：

$$\min_{y} tr(Y^T L Y) \quad s.t.\ Y^T D Y = I$$

这个最优解与上面的解相同，是最小的 d 个非 0 特征值对应的特征向量，这些向量按照列构成了最终的投影矩阵 Y 。

最后介绍整个 LE 谱嵌入的降维算法流程：

1. 构造样本相似度的图。顶点为样本，边为每个顶点与它的邻居样本之间的相似度，相似度计算可用欧氏距离或者近邻法。
2. 计算图的邻接矩阵 W ，加权矩阵 D ，拉普拉斯矩阵 L 。
3. 特征映射。根据上面的拉普拉斯乘子算法推导，实际上是求解如下的广义特征值和特征向量问题：

   $$Lf = \lambda D f$$

4. 假设 $f_0, \cdots, f_{k-1}$ 是这个广义特征值问题的解，按照特征值的大小升序排列，根据前面的结论，$D^{-1}L$ 半正定且特征值满足：

   $$0 = \lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_n$$

5. 去掉值为 0 的特征值 λ_0 ，用剩下的 d 个特征值对应的特征向量按列构成降维结果 Y ，向量 x_i 的投影结果就是这 d 个特征向量的第 i 个分量构成的向量：

   $$x_i \rightarrow (f_1(i), \cdots, f_d(i))^T$$

   这也对应着矩阵 Y 的第 i 行。