4 降维方法

降维是一类应用广泛的问题。它的本质是压缩编码或特征提取。所谓压缩编码，就是构建一个编码器，将样本编码到隐空间向量；再构造一个解码器，将隐空间向量恢复为高维数据。可以在有监督或无监督的情况下进行降维。本章先介绍通过线性变换进行降维的方法，再介绍非线性的降维方法，最后介绍一类借助神经网络设计的降维方法——自编码器。

4.1 线性降维方法

4.1.1 主成分分析 PCA

PCA 是最常用的线性降维算法。它本质上是在空间中寻找 d 个坐标轴，使样本在其上的投影方差依次最大。

算法	**主成分分析**
算法简述	对于降维问题，依次选取 d 个样本方差最大的 正交投影方向作为基，得到降维后向量
已知	样本 $X \in R^{n\times N}$
求	线性变换 $\hat{P} \in R^{n\times d}$ 使得 $X' = X\hat{P}$ 方差最大
解类型	闭式解
算法步骤	1. 将每个样本减去样本均值 $\tilde{\mathbf{x}}^{(i)} = \mathbf{x}^{(i)} - \bar{\mathbf{x}}$ 2. 计算维度协方差矩阵 $C = \tilde{X}\tilde{X}^T \in R^{n\times n}$ 3. 协方差矩阵进行相似对角化分解 $C = Q^T\Lambda Q$ 4. 使用单位矩阵 R 将 Λ 重排序为 $\tilde{\Lambda}$, 使得 $\tilde{\Lambda}$ 的特征值依序从大到小. 即 $\Lambda = R^T\tilde{\Lambda}R$ 5. 取 R^TQ^T 的前 d 列为 $\hat{P}$

4.1.2 局部保持投影 LPP

算法	**局部保持投影**
算法简述	对于线性降维问题，最小化以原空间距离加权的投影后样本距离
已知	样本 $X \in R^{n\times N}$
求	线性变换 $A \in R^{d\times n}$ 使得加权投影后样本距离 $J(A) = \mathrm{tr}(A^TX^tLXA)$ 最小
解类型	闭式解
算法步骤	1. 使用 k 近邻算法构建邻接图（无方向） 2. 对相邻边计算权重 $w_{ij} = \exp\{-\frac{\|\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|\|^2}{\sigma^2}\}$ 3. 计算度矩阵 D 和拉普拉斯矩阵 L 4. 取 $(X^TDX)^{-1}X^TLX$ 的最小 d 个特征值对应特征向量为 A

LPP 也是一种线性降维方法，它的算法目标也是找一个矩阵 $A \in R^{d\times n}$ 将 X 投影到 $Y = AX$。和 PCA 不同，LPP 的优化目标是带权重的投影后样本距离

$$J(A)=\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}w_{ij}||\mathbf{y}^{(i)}-\mathbf{y}^{(j)}||^2=\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}w_{ij}||A\mathbf{x}^{(i)}-A\mathbf{x}^{(j)}||^2 \tag{4.1}$$

其中 w_{ij} 是和高维空间距离相关的权重，距离越近权重越大，距离越远权重越小。具体来说，算法首先寻找每个样本 $\mathbf{x}^{(i)}$ 的 k 近邻，将近邻之间的权重置为

$$w_{ij}=\exp\left\{-\frac{||\mathbf{x}^{(i)}-\mathbf{x}^{(j)}||^2}{\sigma^2}\right\}$$

即径向基函数。其余样本间权重一律为 0. 事实上，此时各样本已经组成了一个带权图。除了权重矩阵 W，还可以计算度矩阵 D 和拉普拉斯矩阵 L。

对优化目标进行展开

$$\begin{aligned}
J(A)=&\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}w_{ij}||A\mathbf{x}^{(i)}-A\mathbf{x}^{(j)}||^2\\
=&\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}w_{ij}(A\mathbf{x}^{(i)}-A\mathbf{x}^{(j)})^T(A\mathbf{x}^{(i)}-A\mathbf{x}^{(j)})\\
=&\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}w_{ij}({\mathbf{x}^{(i)}}^TA^TA\mathbf{x}^{(i)}+{\mathbf{x}^{(j)}}^TA^TA\mathbf{x}^{(j)}-2{\mathbf{x}^{(i)}}^TA^TA\mathbf{x}^{(j)})\\
=&\sum_{i=1}^{N}d_{ii}{\mathbf{x}^{(i)}}^TA^TA\mathbf{x}^{(i)}-\sum_{i=1}^{N}\sum_{j=1}^{N}w_{ij}{\mathbf{x}^{(i)}}^TA^TA\mathbf{x}^{(j)}\\
=&\operatorname{tr}(A^TX^T(D-W)XA)\\
=&\operatorname{tr}(A^TX^TLXA)
\end{aligned}$$

由于投影矩阵 A 有无尺度性，因此增加一个关于投影后向量模长的约束

$$A^TX^TDXA=I$$

即

$$\begin{aligned}
\min_A \quad & J(A)=\operatorname{tr}(A^TX^TLXA)\\
\text{s.t.} \quad & A^TX^TDXA=I
\end{aligned} \tag{4.2}$$

使用 Lagrange 乘子法，得到 Lagrange 函数

$$\mathcal{L}(A,\Lambda)=\operatorname{tr}(A^TX^TLXA)+\operatorname{tr}(\Lambda(A^TX^TDXA-I))$$

求偏导

$$\begin{aligned}
&\frac{\partial\mathcal{L}}{\partial A}=2X^TLXA-2X^TDXA\Lambda=0\\
&X^TLXA=X^TDXA\Lambda\\
&(X^TDX)^{-1}X^TLXA=A\Lambda
\end{aligned}$$

即 A 的列向量是矩阵 $(X^TDX)^{-1}X^TLX$ 的特征向量。为最小化目标函数，取最小的 d 个特征值对应的特征向量组成 A 即可。

注：LPP 和后面列出的 LE 方法一脉相承，LPP 本质是在 LE 的基础上将变换线性化，可能更加不精确，但可以学到变换而不只是嵌入。

4.2 非线性降维方法

4.2.1 核 PCA

低维空间中的非线性复杂分布往往可以在映射到高维空间后变得简单。核 PCA 的思想就是在核函数衍生出的高维空间中对 $\phi(\mathbf{x})$ 进行降维，从而获得非线性空间中的主成分。然而，一般的核函数不能显式提供 $\phi(\mathbf{x})$，强行计算的成本也高。因此需要使用变换技巧将其都转化成核函数的计算。

算法	核主成分分析
算法简述	对于降维问题，在核空间中选取 d 个样本方差最大的正交投影方向 作为基，得到降维后向量
已知	样本 $X \in R^{n\times N}$
求	变换 $f: R^n \to R^d$ 使得 $X' = f(X)$ 方差最大
解类型	闭式解
算法步骤	1. 计算样本核矩阵 K: $k_{ij} = \kappa(x^{(i)}, x^{(j)})$ 2. 计算中心化核矩阵 $\tilde{K} = K - K\mathbf{1}_N - \mathbf{1}_N K + \mathbf{1}_N K \mathbf{1}_N$ 3. 对中心化核阵进行特征分解 $\tilde{K}P = \Lambda P$, $\Lambda = \text{diag}\{\lambda_1, \lambda_2, ..., \lambda_n\}$ 是从大到小排序后的特征值矩阵，$P = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n\}$ 4. 计算降维后样本 $X' = P_d^T \tilde{K}$，其中 $P_d = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_d\}$ 为 P 的前 d 列

首先要对 $\phi(\mathbf{x})$ 进行去中心化，使其均值为 0. 虽然我们无法直接计算嵌入向量，但我们只关心嵌入向量的 Gram 矩阵，因此我们需要的是去中心化向量 $\tilde{\mathbf{x}}$ 的核矩阵 $\tilde{K}$。

设 $\phi(X) = [\phi(\mathbf{x}^{(1)}), \phi(\mathbf{x}^{(1)}), ..., \phi(\mathbf{x}^{(1)})] \in R^{m\times N}(m > n)$. 设元素全为 $\frac{1}{N}$ 的 $N \times N$ 矩阵为 $\mathbf{1}_N$，设元素全为 1 的 n 维列向量为 $\mathbf{e}_N$，则有核矩阵

$$K = \phi(X)^T \phi(X)$$

嵌入向量均值

$$\bar{\phi}(\mathbf{x}) = \frac{\sum_{i=1}^N \phi(\mathbf{x}^{(i)})}{N} = \frac{1}{N}\phi(X)\mathbf{e}_N$$

去中心化向量

$$\tilde{\phi}(X) = \phi(X) - \bar{\phi}(\mathbf{x})\mathbf{e}_N = (I - \frac{1}{N}\mathbf{e}_N\mathbf{e}_N^T)\phi(X) = (I - \mathbf{1}_N)\phi(X)$$

去中心化 Gram 矩阵（去中心化核矩阵）

$$\begin{aligned}\tilde{K} =& \tilde{\phi}(X)^T \tilde{\phi}(X) \\ =& (I - \mathbf{1}_N)^T \phi(X)^T \phi(X)(I - \mathbf{1}_N) \\ =& (I - \mathbf{1}_N)^T K (I - \mathbf{1}_N) \\ =& K - \mathbf{1}_N K - K\mathbf{1}_N + \mathbf{1}_N K \mathbf{1}_N\end{aligned}$$

我们需要在核空间中找到 d 个正交的单位向量 $\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_d$ 作为基，于是 $\phi(\mathbf{x})$ 在这组基上的投影就构成了降维后的 R^d 坐标向量。

$$
\begin{aligned}
&\max_{\mathbf{p}} \sum_{i=1}^{N} \left\langle \mathbf{p}, \tilde{\phi}(\mathbf{x}^{(i)}) \right\rangle^2 \\
=&\max_{\mathbf{p}} \sum_{i=1}^{N} (\mathbf{p}^T \tilde{\phi}(\mathbf{x}^{(i)}))^2 \\
=&\max_{\mathbf{p}} \mathbf{p}^T \left(\sum_{i=1}^{N} \tilde{\phi}(\mathbf{x}^{(i)}) \tilde{\phi}(\mathbf{x}^{(i)}) \right) \mathbf{p} \\
=&\max_{\mathbf{p}} \mathbf{p}^T C_e \mathbf{p}
\end{aligned}
$$

其中中间的矩阵记作 C_e，它是嵌入后的协方差矩阵。

$$C_e = \sum_{i=1}^{N} \tilde{\phi}(\mathbf{x}^{(i)}) \tilde{\phi}(\mathbf{x}^{(i)}) = \tilde{\phi}(X) \tilde{\phi}(X)^T$$

正如 PCA 算法中所推导的，当要求一组相互正交的 p 使每个 p 都让二次型目标取极值时，应当取 C_e 的对应最大的几个特征值的特征向量，即

$$C_e \mathbf{p} = \tilde{\phi}(X) \tilde{\phi}(X)^T \mathbf{p} = \lambda \mathbf{p}$$

然而，协方差矩阵 C_e 的值可能是无法计算或计算成本很大的，因此无法直接相似对角化。将上式变形

$$\mathbf{p} = \frac{1}{\lambda} \tilde{\phi}(X) \tilde{\phi}(X)^T \mathbf{p} = \frac{1}{\lambda} \tilde{\phi}(X) \alpha \tag{4.3}$$

其中

$$\alpha = \tilde{\phi}(X)^T \mathbf{p} \tag{4.4}$$

是一个 N 维列向量。也就是说，根据式4.3，$\mathbf{p}$ 是各 $\phi(\mathbf{x})$ 的线性组合。代入式4.3中

$$\frac{1}{\lambda} \tilde{\phi}(X) \tilde{\phi}(X)^T \tilde{\phi}(X) \alpha = \tilde{\phi}(X) \alpha$$

两边左乘 $\tilde{\phi}(X)^T$，有

$$\frac{1}{\lambda} \tilde{\phi}(X)^T \tilde{\phi}(X) \tilde{\phi}(X)^T \tilde{\phi}(X) \alpha = \tilde{\phi}(X)^T \tilde{\phi}(X) \alpha$$

即

$$\frac{1}{\lambda} \tilde{K} \tilde{K} \alpha = \tilde{K} \alpha$$

$$\tilde{K} \alpha = \lambda \alpha$$

因此，$\tilde{K}\alpha$ 是 $\tilde{K}$ 的特征向量，即 α 是 $\tilde{K}$ 的特征向量。这样，我们可以由 $\tilde{K}$ 的前 d 大的特征值得到 d 个 $\mathbf{p}$ 对应的 α。

接下来只需要将 $\tilde{\phi}(X)$ 投影到 $P = [\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_d]$ 上即可。为了投影方便，需要将 P 的列向量归一化，即

$$\mathbf{p}^T\mathbf{p}=1$$

由式4.3，有

$$(\frac{1}{\lambda}\tilde{\phi}(X)\alpha)^T(\frac{1}{\lambda}\tilde{\phi}(X)\alpha)=1$$

$$\frac{1}{\lambda^2}\alpha^T\tilde{\phi}(X)^T\tilde{\phi}(X)\alpha=1$$

$$\frac{1}{\lambda^2}\alpha^T\tilde{K}\alpha=1$$

$$\frac{1}{\lambda^2}\alpha^T\lambda\alpha=1$$

$$\alpha^T\alpha=\lambda$$

因此只要放缩 α 使其模长为 $\sqrt{\lambda}$, 就可以使 $\mathbf{p}$ 归一化。此时有

$$f(\mathbf{X})=(\tilde{\phi}(X)^TP)^T=[\alpha_1,\alpha_2,...,\alpha_d]^T$$

即，样本 $\mathbf{x}^{(i)}$ 的降维结果就是求出的各 α 的第 i 行元素组成的 d 维向量。

4.2.2 多维度缩放 MDS

算法	**多维度缩放**
算法简述	对于降维问题，保持降维后各样本间距不变
已知	样本 $X\in R^{n\times N}$
求	$f(X)$，其中变换 $f:R^n\to R^d$ 使得 $\|\|\mathbf{x}^{(i)}-\mathbf{x}^{(j)}\|\|=\|\|f(\mathbf{x}^{(i)})-f(\mathbf{x}^{(j)})\|\|$
解类型	闭式解
算法步骤	1. 计算原空间中样本两两间距 $d_{ij}=\|\|\mathbf{x}^{(i)}-\mathbf{x}^{(j)}\|\|$ 2. 计算矩阵 B：$b_{ij}=-\frac{1}{2}(d_{ij}^2-\frac{1}{N}\sum_{k=1}^N d_{kj}^2-\frac{1}{N}\sum_{l=1}^N d_{il}^2+\frac{1}{N^2}\sum_{k=1}^N\sum_{l=1}^N d_{kl}^2)$ 3. 对 B 矩阵做相似对角化分解 $B=P^T\Lambda P$， 取 P^T 中对应前 d 大特征值的 d 行组成 P_d， 对应特征值组成 Λ_d，$f(X)=P_d\Lambda_d^{\frac{1}{2}}\in R^{d\times N}$ 即为所求

假设样本 $X\in R^{n\times N}$ 降维后变成 $f(X)=Z=[\mathbf{z}^{(1)},\mathbf{z}^{(2)},...,\mathbf{z}^{(N)}]\in R^{d\times N}$。可以写出降维后内积矩阵

$$B=Z^TZ$$

其中 $b_{ij}=\mathbf{z}^{(i)^T}\mathbf{z}^{(j)}$。设原空间中样本两两间距

$$d_{ij}=||\mathbf{x}^{(i)}-\mathbf{x}^{(j)}||$$

有距离约束

$$
\begin{aligned}
d_{ij}^2 =&||\mathbf{z}^{(i)} - \mathbf{z}^{(j)}||^2 \\
=&\mathbf{z}^{(i)^T}\mathbf{z}^{(i)} - 2\mathbf{z}^{(i)^T}\mathbf{z}^{(j)} + \mathbf{z}^{(j)^T}\mathbf{z}^{(j)} \\
=&b_{ii} - 2b_{ij} + b_{jj}
\end{aligned} \tag{4.5}
$$

不妨设降维后样本是中心化的，即

$$
\sum_{i=1}^{N} \mathbf{z}^{(i)} = \mathbf{0}
$$

则 B 矩阵有行和列和为 0 的约束

$$
\begin{aligned}
\sum_{i=0}^{N} b_{ij} =&0 \\
\sum_{j=0}^{N} b_{ij} =&0
\end{aligned}
$$

此时考虑 D 矩阵的行列平方和

$$
\begin{aligned}
d_{\cdot j}^2 =& \sum_{i=0}^{N} d_{ij}^2 = \mathrm{tr}(B) + Nb_{jj} \\
d_{i\cdot}^2 =& \sum_{j=0}^{N} d_{ij}^2 = \mathrm{tr}(B) + Nb_{ii} \\
d_{\cdot\cdot}^2 =& \sum_{i=0}^{N}\sum_{j=0}^{N} d_{ij}^2 = 2N\,\mathrm{tr}(B)
\end{aligned}
$$

代入式4.5中，即可求出 B

$$
\begin{aligned}
b_{ij} =& -\frac{1}{2}(d_{ij}^2 - b_{ii} - b_{jj}) \\
=& -\frac{1}{2}(d_{ij}^2 - \frac{1}{N}(d_{i\cdot}^2 - \frac{1}{2N}d_{\cdot\cdot}^2) - \frac{1}{N}(d_{\cdot j}^2 - \frac{1}{2N}d_{\cdot\cdot}^2)) \\
=& -\frac{1}{2}(d_{ij}^2 - \frac{1}{N}d_{i\cdot}^2 - \frac{1}{N}d_{\cdot j}^2 + \frac{1}{N^2}d_{\cdot\cdot}^2)
\end{aligned} \tag{4.6}
$$

对 B 矩阵做相似对角化分解 $B = P^T\Lambda P$，取 P 中对应前 k 大特征值的 d 列组成 P_d，对应特征值组成 Λ_d，即可得到 $Z = \Lambda_d^{\frac{1}{2}}P_d$

4.2.3 等距特征映射 ISOMAP

MDS 方法保留了样本在高维空间中的距离，但有时数据是分布在高维空间中的低维流形上的，此时应当尝试保留流形上的距离。ISOMAP 通过流形最短路径估计这个距离。

算法	等距特征映射
算法简述	对于降维问题，保持降维后各样本间距不变
已知	样本 $X \in R^{n\times N}$
求	变换 $f: R^n \to R^d$ 使得 $\|\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|\| = \|\|f(\mathbf{x}^{(i)}) - f(\mathbf{x}^{(j)})\|\|$
解类型	确定解
算法步骤	1. 计算高维空间中样本的 k 近邻，将其对称化，保证形成连通图 2. 将每个样本和其 k 近邻的流形距离设置为其欧氏距离 3. 对于非近邻点对，使用 Dijkstra 算法求出最短距离作为流形距离 4. 以样本间流形距离作为 D 矩阵，应用 MDS 算法，得到降维结果

由于几乎每对点之间都要计算最短距离，且 Dijkstra 最短距离的计算复杂度是 $O(N\log N)$（邻接图边数和节点数成线性关系），所以 ISOMAP 算法总时间复杂度至少为 $O(N^3\log N)$。对于数据量较大的情况，最好避免使用这个算法。

4.2.4 局部线性嵌入 LLE

算法	局部线性嵌入
算法简述	对于降维问题，保持降维后各样本间距不变
已知	样本 $X \in R^{n\times N}$
求	回归系数 $W \in R^{N\times N}$ 使得 $\min J_i(W) = \|\|\mathbf{x}^{(i)} - \sum_{j=1}^N w_{ij}\mathbf{x}^{(j)}\|\|^2$ 且满足近邻约束 嵌入向量 $Y \in R^{d\times N}$ 使得 $\min J(Y) = \sum_{i=1}^N \|\|\mathbf{y}^{(i)} - \sum_{j\in N(i)} w_{ij}\mathbf{y}^{(j)}\|\|^2$
解类型	闭式解
算法步骤	1. 计算高维空间中样本的 k 近邻，将其对称化，保证形成连通图 2. 逐样本计算近邻方差 $C_i = (\mathbf{x}^{(j)} - \mathbf{x}^{(i)})^T(\mathbf{x}^{(k)} - \mathbf{x}^{(i)})$ 3. 计算回归系数 $\tilde{\mathbf{w}}_i = \frac{C_i^{-1}\mathbf{1}_N}{\mathbf{1}_N^T C_i^{-1}\mathbf{1}_N}$ 4. 取 $(W-I)(W-I)^T$ 中最小的 d 个特征值对应的特征行向量组成 Y

相比于以上几种方法关注全局性质的保持，LLE 和 LPP 类似，都更关注数据局部性质的保持。LLE 认为，任何一个样本都可以通过近邻样本的线性组合来近似。如果在低维空间中也能保持和高维空间中一样的线性组合关系，就是保留了局部特征。因此 LLE 计算分为两步，首先寻找最优线性组合系数，其次计算最优低维表示。

假设有样本 $X \in R^{n\times N}$。用 kNN 算法可以构建其邻接图（非对称）。设有系数矩阵 $W = [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_N] \in R^{N\times N}$。若 $\mathbf{x}^{(j)}$ 不是 $\mathbf{x}^{(i)}$ 的 k 近邻，则 $w_{ij} = 0$。其余系数用于拟合，即

$$\mathbf{x}^{(i)} \approx \sum_{j=1}^N w_{ji}\mathbf{x}^{(j)} = X\mathbf{w}_i$$

有约束

$$\sum_{j=1}^N w_{ji} = 1$$

最优参数就是最小化回归的平方误差

$$\min_W J(W) = \sum_{i=1}^N ||\mathbf{x}^{(i)} - \sum_{j\in N(i)} w_{ji}\mathbf{x}^{(j)}||^2$$

其中 $N(i)$ 表示第 i 个样本的 K 近邻的下标集合。W 的一行只和单个样本有关，记 $\tilde{\mathbf{w}}_i \in R^k$ 为第 i 个样本的 K 近邻的加权系数向量. 考虑第 i 个样本的回归平方误差

$$\begin{aligned}
J_i(W) =& ||\mathbf{x}^{(i)} - \sum_{j=1}^N w_{ji}\mathbf{x}^{(j)}||^2 \\
=& ||\sum_{j=1}^N w_{ji}(\mathbf{x}^{(i)} - \mathbf{x}^{(j)})||^2 \\
=& \sum_{j=1}^K\sum_{k=1}^K w_{ji}w_{ki}(\mathbf{x}^{(j)} - \mathbf{x}^{(i)})^T(\mathbf{x}^{(k)} - \mathbf{x}^{(i)}) \\
=& \sum_{j=1}^K\sum_{k=1}^K w_{ji}w_{ki}c_{jk} \\
=& \mathbf{w}_i^T C_i \mathbf{w}_i
\end{aligned}$$

设 $\mathbf{1}_K$ 代表全 1 的 K 维列向量, 约束项可以写为

$$\mathbf{1}_N^T\mathbf{w}_i = 1$$

即

$$\begin{aligned}
\min_{\mathbf{w}_i} \quad & J_i(\mathbf{w}_i) = \mathbf{w}_i^T C_i \mathbf{w}_i \\
\text{s.t.} \quad & \mathbf{1}_N^T\mathbf{w}_i = 1
\end{aligned} \tag{4.7}$$

使用 Lagrange 乘子法

$$\mathcal{L}(\tilde{\mathbf{w}}_i, \lambda) = \tilde{\mathbf{w}}_i^T C_i \tilde{\mathbf{w}}_i + \lambda(\mathbf{1}_N^T\tilde{\mathbf{w}}_i - 1)$$

对 $\tilde{\mathbf{w}}_i$ 求偏导

$$\frac{\partial\mathcal{L}}{\partial\tilde{\mathbf{w}}_i} = 2C_i\tilde{\mathbf{w}}_i + \lambda\mathbf{1}_N = 0$$

即

$$\tilde{\mathbf{w}}_i = -\frac{\lambda}{2}C_i^{-1}\mathbf{1}_N$$

使用约束条件归一化，有

$$\tilde{\mathbf{w}}_i = \frac{C_i^{-1}\mathbf{1}_N}{\mathbf{1}_N^T C_i^{-1}\mathbf{1}_N} \tag{4.8}$$

另一方面，设 $\mathbf{x}^{(i)}$ 对应低维表示为 $\mathbf{y}^{(i)}$。优化目标为最小化平方误差

$$\min_Y J(Y) = \sum_{i=1}^N ||\mathbf{y}^{(i)} - \sum_{j\in N(i)} w_{ji}\mathbf{y}^{(j)}||^2$$

由于 Y 有非尺度性，加入约束条件

$$\frac{1}{N}YY^T = I$$

重写目标函数为

$$\begin{aligned}
\min J(Y) &= \sum_{i=1}^{N} ||\mathbf{y}^{(i)} - \sum_{j=1}^{N} w_{ji}\mathbf{y}^{(j)}||^2 \\
&= \sum_{i=1}^{N} ||\mathbf{y}^{(i)} - Y\mathbf{w}_i||^2 \\
&= \sum_{i=1}^{N} (\mathbf{y}^{(i)} - Y\mathbf{w}_i)^T(\mathbf{y}^{(i)} - Y\mathbf{w}_i) \\
&= \operatorname{tr}(Y^TY) + \operatorname{tr}(W^TY^TYW) - 2\operatorname{tr}(Y^TYW) \\
&= \operatorname{tr}((W-I)^TY^TY(W-I))
\end{aligned}$$

即

$$\begin{aligned}
\min_{Y} \quad & J(Y) = \operatorname{tr}((W-I)^TY^TY(W-I)) \\
\text{s.t.} \quad & \frac{1}{N}YY^T = I
\end{aligned} \tag{4.9}$$

Lagrange 乘子法

$$\mathcal{L}(Y,\lambda) = \operatorname{tr}((W-I)^TY^TY(W-I)) + \lambda(YY^T - NI))$$

对 Y 求偏导

$$\frac{\partial \mathcal{L}}{\partial Y} = 2Y(W-I)(W-I)^T + 2\lambda Y = 0$$

即

$$Y(W-I)(W-I)^T = \lambda Y$$

取 $(W-I)(W-I)^T$ 中最小的 d 个特征值对应的特征行向量组成 Y 即可。

4.2.5 拉普拉斯特征图 LE

算法	**拉普拉斯特征图**
算法简述	对于降维问题，最小化以原空间距离加权的降维后样本距离
已知	样本 $X \in R^{n\times N}$
求	嵌入向量 $Y \in R^{d\times N}$ 使得使得加权投影后样本距离 $J(Y) = \operatorname{tr}(Y^TLY)$ 最小
解类型	闭式解
算法步骤	1. 使用 k 近邻算法构建邻接图（对称化） 2. 对相邻边计算权重 $w_{ij} = \exp\left\{-\frac{\lVert\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\rVert^2}{\sigma^2}\right\}$ 3. 计算度矩阵 D 和拉普拉斯矩阵 L 4. 取 $LD^{-1}L$ 的最小 d 个特征值对应特征行向量为 Y

LE 和 LPP 思路一致，优化目标都是带权重的投影后样本距离，只不过优化变量直接是投影后的向量

$$J(Y) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}||\mathbf{y}^{(i)} - \mathbf{y}^{(j)}||^2 = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}||\mathbf{y}^{(i)} - \mathbf{y}^{(j)}||^2$$

其中 w_{ij} 是和高维空间距离相关的权重，距离越近权重越大，距离越远权重越小。具体来说，算法首先寻找每个样本 $\mathbf{x}^{(i)}$ 的 k 近邻，将近邻之间的权重置为

$$w_{ij} = \exp\left\{-\frac{||\mathbf{x}^{(i)} - \mathbf{x}^{(j)}||^2}{\sigma^2}\right\}$$

即径向基函数。其余样本间权重一律为 0. 事实上，此时各样本已经组成了一个带权图。除了权重矩阵 W，还可以计算度矩阵 D 和拉普拉斯矩阵 L。

对优化目标进行展开

$$\begin{aligned}
J(Y) =& \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}||\mathbf{y}^{(i)} - \mathbf{y}^{(j)}||^2 \\
=& \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}(\mathbf{y}^{(i)} - \mathbf{y}^{(j)})^T(\mathbf{y}^{(i)} - \mathbf{y}^{(j)}) \\
=& \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}({\mathbf{y}^{(i)}}^T\mathbf{y}^{(i)} + {\mathbf{y}^{(j)}}^T\mathbf{y}^{(j)} - 2{\mathbf{y}^{(i)}}^T\mathbf{y}^{(j)}) \\
=& \sum_{i=1}^{N} d_{ii}{\mathbf{y}^{(i)}}^T\mathbf{y}^{(i)} - \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}{\mathbf{y}^{(i)}}^T\mathbf{Y}^{(j)} \\
=& \operatorname{tr}(Y^TY(D - W)) \\
=& \operatorname{tr}(Y^TYL) \\
=& \operatorname{tr}(YLY^T)
\end{aligned}$$

由于 Y 有无尺度性，因此增加一个约束

$$YDY^T = I$$

即

$$\begin{aligned}
\min_{Y} \quad & J(Y) = \operatorname{tr}(YLY^T) \\
\text{s.t.} \quad & YDY^T = I
\end{aligned} \tag{4.10}$$

使用 Lagrange 乘子法，得到 Lagrange 函数

$$\mathcal{L}(Y, \Lambda) = \operatorname{tr}(YLY^T) + \operatorname{tr}(\Lambda(YDY^T - I))$$

求偏导

$$\begin{aligned}
&\frac{\partial \mathcal{L}}{\partial Y} = 2YL - 2YD\Lambda = 0 \\
&YL = \Lambda YD \\
&YLD^{-1} = Y\Lambda D
\end{aligned}$$

即 Y 的行向量为 LD^{-1} 矩阵的行特征向量。为最小化目标函数，取最小的 d 个特征值对应的行特征向量组成 Y 即可。