17 Spectral Clustering 谱聚类

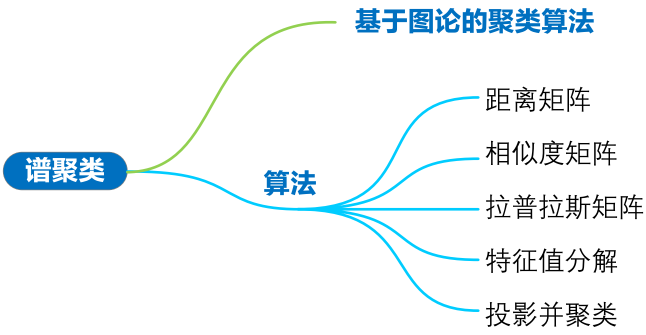
构造无向图，距离远的两点，权重值低；降维聚类

生命中最重要的问题，几乎都是概率问题。

The most important questions of life are indeed, for the most part, really only problems of probability.

—— 皮埃尔-西蒙·拉普拉斯 (Pierre-Simon Laplace) | 法国著名天文学家和数学家 | 1749 ~ 1827

- `sklearn.cluster.SpectralClustering()` 谱聚类算法
- `sklearn.datasets.make_circles()` 创建环形样本数据
- `sklearn.preprocessing.StandardScaler().fit_transform()` 标准化数据；通过减去均值然后除以标准差，处理后数据符合标准正态分布


谱聚类
基于图论的聚类算法
算法
距离矩阵
相似度矩阵
拉普拉斯矩阵
特征值分解
投影并聚类

17.1 谱聚类

谱聚类 (spectral clustering) 是一种基于图论的聚类算法，其特点是能够处理高维数据和非凸数据簇，并且对于数据分布的形态没有特殊要求。优点是可以在任意维度上进行聚类，并且不会受到噪声的影响。缺点是需要进行谱分解计算，计算量较大。

具体来说，谱聚类的思路是将样本数据看做是空间**节点** (node)，这些节点之间用**边** (edge) 连构成的**无向图** (undirected graph)，也叫**加权图**。无向图中，距离远的数据点，边的权重值低；距离近的数据点，在无向图中，边的权重值高。

用无向图聚类的过程很简单，切断无向图中权重值低的边，得到一系列子图。子图内部节点之间边的权重尽可能高，子图之间边权重尽可能低。将节点之间的相似度构成的矩阵称为邻接矩阵，通过对邻接矩阵进行谱分解，得到数据点的特征向量，进而将其映射到低维空间进行聚类。

流程

这个思路虽然简单，但是实际操作需要一系列矩阵运算。

首先，需要计算数据 $\boldsymbol{X}$ 之间的两两距离，并构造成距离矩阵 $\boldsymbol{D}$。然后，将距离转换成权重值，即**相似度** (similarity)，构造**相似度矩阵** (similarity matrix) $\boldsymbol{S}$，利用 $\boldsymbol{S}$ 可以绘制无向图。

之后，将相似度矩阵转化成**拉普拉斯矩阵** (Laplacian matrix) $\boldsymbol{L}$。最后，**特征值分解** (eigen decomposition) $\boldsymbol{L}$，相当于将 $\boldsymbol{L}$ 投影在一个低维度正交空间。在这个低维度空间中，用简单聚类方法对投影数据进行聚类，并得到原始数据聚类。

下面通过实例，我们一一讨论谱聚类这些步骤所涉及的技术细节。

17.2 距离矩阵

图 1 给出 12 个样本点在平面上位置。计算数据**两两距离** (pairwise distance)，$\boldsymbol{x}^{(i)}$和 $\boldsymbol{x}^{(j)}$两个点之间欧氏距离 $d_{i,j}$：

$$d_{i,j} = \sqrt{\left(\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}\right)^{\mathrm{T}} \left(\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}\right)} = \left\|\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}\right\| \tag{1}$$

其中，约定 $\boldsymbol{x}^{(i)}$ 和 $\boldsymbol{x}^{(j)}$ 均为列向量。

图 2 所示为热图描绘的 12 个样本点两两欧氏距离构造的对称矩阵 $\boldsymbol{D}$；注意，$\boldsymbol{D}$ 的对角线元素均为 0，这是因为观察点和自身之间距离为 0。色块颜色越浅，说明距离越近；色块颜色越深，说明距离越远。

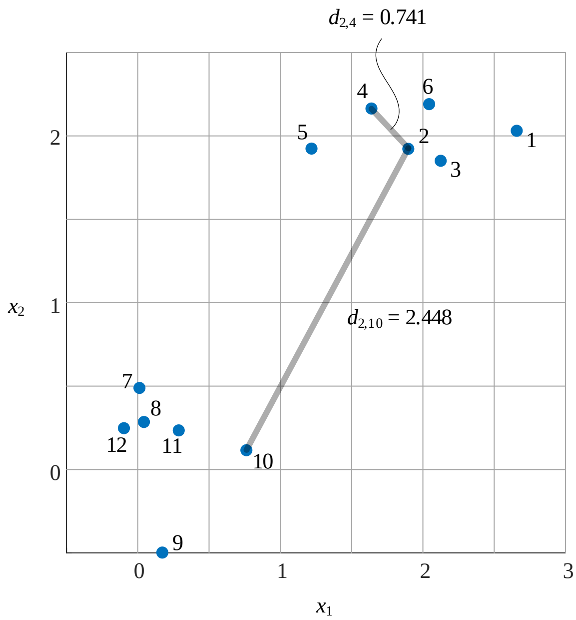


图 1. 12 个样本点平面位置

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0.3845	0.8692	0.7926	0.1449	0.2649	2.891	2.131	2.586	2.74	2.605	3.282
2	0.3845	0	0.4894	0.7412	0.2494	0.3749	2.629	1.929	2.356	2.448	2.333	2.975
3	0.8692	0.4894	0	0.851	0.7387	0.8378	2.439	1.865	2.228	2.207	2.134	2.695
4	0.7926	0.7412	0.851	0	0.7935	0.9998	3.276	2.638	3.039	3.054	2.972	3.546
5	0.1449	0.2494	0.7387	0.7935	0	0.2063	2.763	2.017	2.466	2.605	2.474	3.144
6	0.2649	0.3749	0.8378	0.9998	0.2063	0	2.65	1.875	2.335	2.514	2.369	3.063
7	2.891	2.629	2.439	3.276	2.763	2.65	0	0.867	0.4055	0.3493	0.3054	0.6349
8	2.131	1.929	1.865	2.638	2.017	1.875	0.867	0	0.4841	0.9092	0.6789	1.443
9	2.586	2.356	2.228	3.039	2.466	2.335	0.4055	0.4841	0	0.5782	0.354	1.028
10	2.74	2.448	2.207	3.054	2.605	2.514	0.3493	0.9092	0.5782	0	0.2384	0.5628
11	2.605	2.333	2.134	2.972	2.474	2.369	0.3054	0.6789	0.354	0.2384	0	0.7683
12	3.282	2.975	2.695	3.546	3.144	3.063	0.6349	1.443	1.028	0.5628	0.7683	0

图 2. 12 个样本点两两欧氏距离构造的成对距离矩阵 $\boldsymbol{D}$

17.3 相似度

然后利用 $d_{i,j}$ 计算 i 和 j 两点的相似度 $s_{i,j}$，“距离 → 相似度”的转换采用高斯核函数：

$$s_{i,j} = \exp\left(-\left(\frac{d_{i,j}}{\sigma}\right)^2\right) = \exp\left(-\frac{\left\|\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}\right\|^2}{\sigma^2}\right) \tag{2}$$

相似度取值区间为 (0, 1]。$\boldsymbol{x}^{(i)}$ 和 $\boldsymbol{x}^{(j)}$两个点距离越近，它们的相似性越高。任意点和自身的距离为 0，因此对应的相似度最大为 1。

$\sigma = 1$ 时，两两距离 $d_{i,j}$ 和相似度 $s_{i,j}$ 两者关系如图 3 所示。

图 1 中，点 $\boldsymbol{x}^{(2)}$和 $\boldsymbol{x}^{(10)}$之间欧氏距离为 $d_{2,10}$ = 2.448，点 $\boldsymbol{x}^{(2)}$和 $\boldsymbol{x}^{(4)}$之间欧氏距离为 $d_{2,4}$ = 0.741。利用上式，可以计算得到，点 $\boldsymbol{x}^{(2)}$和 $\boldsymbol{x}^{(10)}$之间相似度 $s_{2,10}$ = 0.0025，点 $\boldsymbol{x}^{(2)}$和 $\boldsymbol{x}^{(4)}$之间欧氏距离为 $s_{2,4}$ = 0.577。

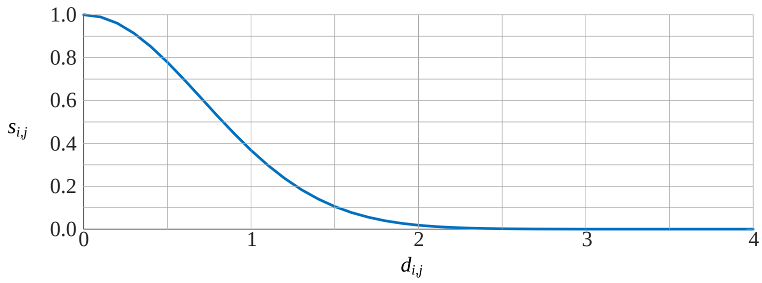


图 3. 欧氏距离和相似度关系

图 2 所示成对距离矩阵可以转化为图 4 所示**相似度矩阵** (similarity matrix) $\boldsymbol{S}$。$\boldsymbol{S}$ 也叫**邻接矩阵** (adjacency matrix)。相似度矩阵 $\boldsymbol{S}$ 的每个元素均大于 0。请大家注意，一些教材将两两距离矩阵 $\boldsymbol{D}$ 叫做相似度矩阵。从图 4 一眼就可以看出数据可以划分为两簇。

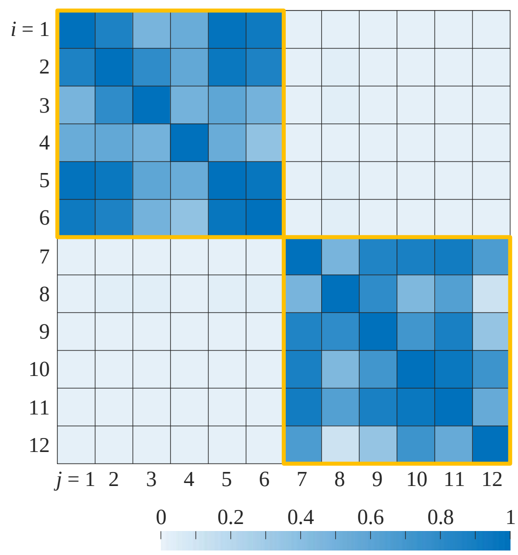


图 4. 12 个样本点两两相似度矩阵 $\boldsymbol{S}$

17.4 无向图

图 5 为相似度矩阵 $\boldsymbol{S}$ 无向图。图中绿色线越粗，表明两点之间的相似度越高，也就是两点距离越近。

切断相似度小于 0.001 两两元素之间的联系得到无向图图 6。图 7 为，切断相似度小于 0.005 两两元素之间的联系得到无向图。观察图 8 可以知道，当切断相似度小于 0.031 两两元素之间的联系，可以将原始数据划分为两簇。

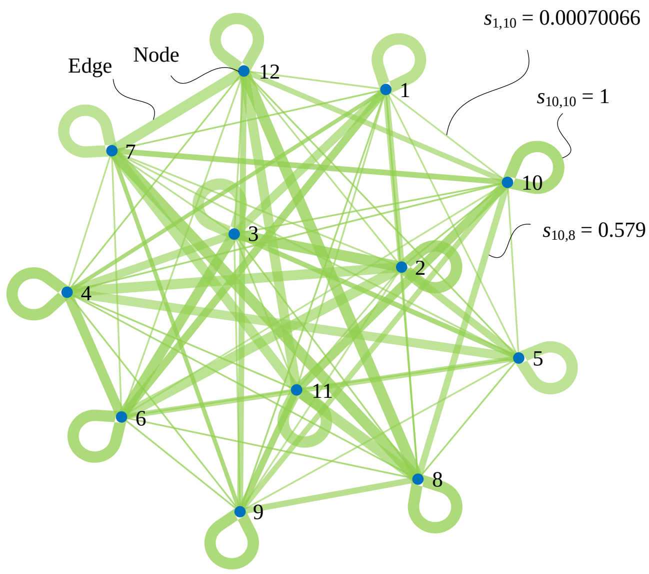


图 5. 相似度对称矩阵 S 无向图

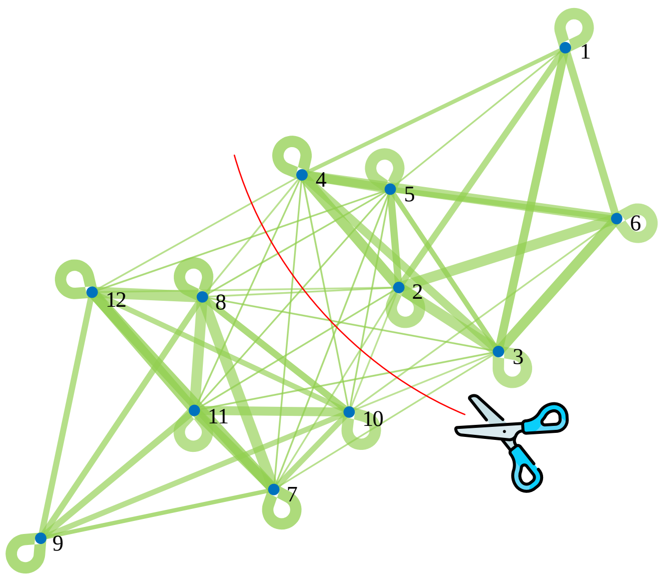


图 6. 当切断相似度小于 0.001 两两元素之间的联系得到无向图

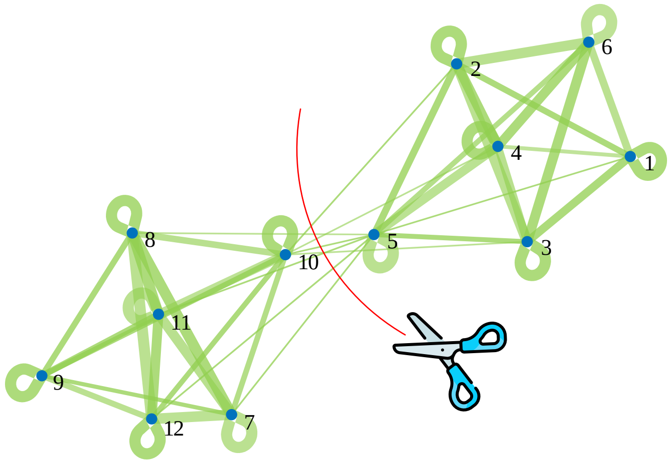


图 7. 当切断相似度小于 0.005 两两元素之间的联系得到无向图

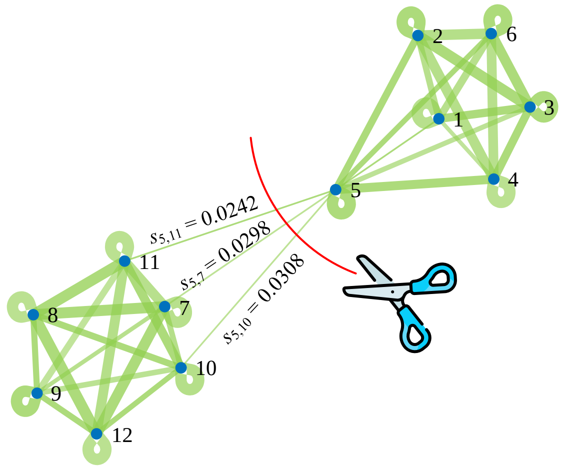


图 8. 当切断相似度小于 0.02 两两元素之间的联系得到无向图

17.5 拉普拉斯矩阵

度矩阵 (degree matrix) $\boldsymbol{G}$ 是一个对角阵。$\boldsymbol{G}$ 的对角线元素是对应相似度矩阵 $\boldsymbol{S}$ 对应列元素之和，即：

$$\boldsymbol{G}_{i,i} = \sum_{j=1}^{n} s_{i,j} = \operatorname{diag}\left(\boldsymbol{1}^{\mathrm{T}} \boldsymbol{S}\right) \tag{3}$$

	1	2	3	4	5	6	7	8	9	10	11	12
1	4.791	0	0	0	0	0	0	0	0	0	0	0
2	0	5.071	0	0	0	0	0	0	0	0	0	0
3	0	0	3.876	0	0	0	0	0	0	0	0	0
4	0	0	0	3.498	0	0	0	0	0	0	0	0
5	0	0	0	0	5.013	0	0	0	0	0	0	0
6	0	0	0	0	0	4.664	0	0	0	0	0	0
7	0	0	0	0	0	0	4.79	0	0	0	0	0
8	0	0	0	0	0	0	0	3.569	0	0	0	0
9	0	0	0	0	0	0	0	0	4.604	0	0	0
10	0	0	0	0	0	0	0	0	0	4.726	0	0
11	0	0	0	0	0	0	0	0	0	0	4.945	0
12	0	0	0	0	0	0	0	0	0	0	0	3.424

图 9. 12 个样本点两两相似度构造的度矩 $\boldsymbol{G}$

拉普拉斯矩阵

然后构造**拉普拉斯矩阵** (Laplacian matrix) $\boldsymbol{L}$。有三种方法构造拉普拉斯矩阵。

第一种叫做**未归一化拉普拉斯矩阵** (unnormalized Laplacian matrix)，具体定义如下：

$$\boldsymbol{L} = \boldsymbol{G} - \boldsymbol{S} \tag{4}$$

第二种叫做**归一化随机漫步拉普拉斯矩阵** (normalized random-walk Laplacian matrix)，也叫 Shi-Malik 矩阵，定义如下：

$$\boldsymbol{L}_{rw} = \boldsymbol{G}^{-1}\left(\boldsymbol{G} - \boldsymbol{S}\right) \tag{5}$$

第三种叫做**归一化对称拉普拉斯矩阵** (normalized symmetric Laplacian matrix)，也叫做 Ng-Jordan-Weiss 矩阵，如下：

$$\boldsymbol{L}_{s} = \boldsymbol{G}^{-1/2}\left(\boldsymbol{G} - \boldsymbol{S}\right)\boldsymbol{G}^{-1/2} \tag{6}$$

采用第一种方法获得拉普拉斯矩阵 $\boldsymbol{L}$，热图如图 10 所示。

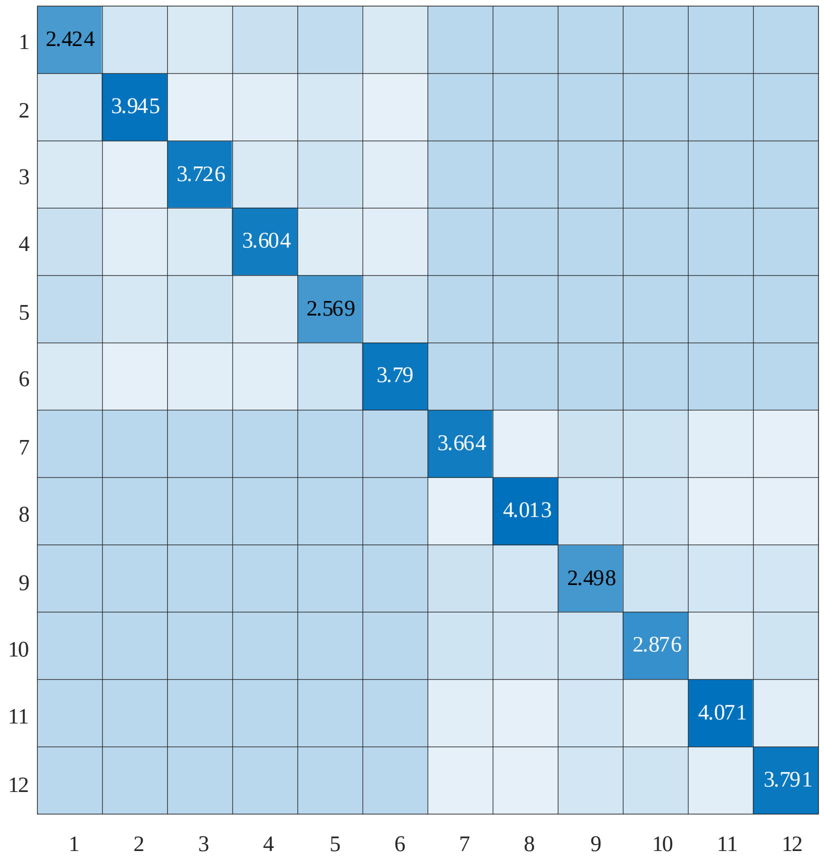


图 10. 12 个样本点两两相似度构造未归一化拉普拉斯矩阵 $\boldsymbol{L}$

17.6 特征值分解

对拉普拉斯矩阵 $\boldsymbol{L}$ 进行特征值分解：

$$\boldsymbol{L} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1} \tag{7}$$

其中

$$\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_{12} \end{bmatrix}, \quad \boldsymbol{V} = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \dots & \boldsymbol{v}_1 \end{bmatrix} \tag{8}$$

图 11 所示为拉普拉斯矩阵 $\boldsymbol{L}$ 特征值分解得到的特征值从小到大排序。按从小到大排列 λ 值，对应第 2 个，$\lambda_2 = 0.01285$，对应的特征向量 $\boldsymbol{v}_2 = [-0.300, -0.295, -0.297, -0.294, -0.275, -0.298, 0.283, 0.285, 0.288, 0.278, 0.284, 0.286]$。

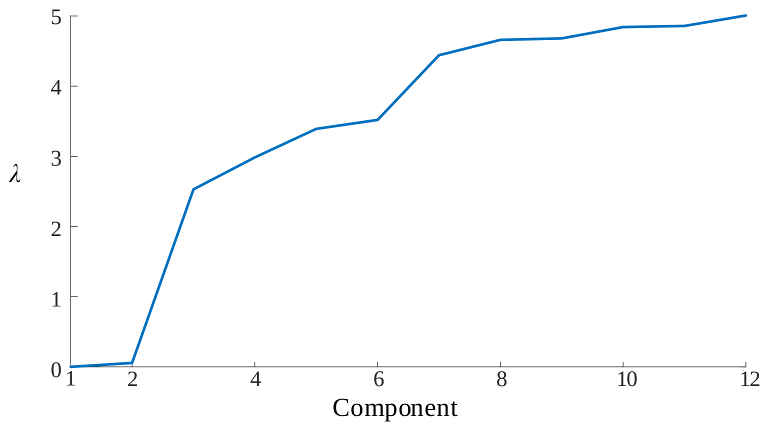


图 11. 拉普拉斯矩阵 $\boldsymbol{L}$ 特征值分解得到的特征值从小到大排序

图 12 和图 13 分别展示前两个特征向量的结果。相当于将拉普拉斯矩阵 $\boldsymbol{L}$ 投影到一个二维空间，具体如图 14 所示。在图 14 所示平面内，可以很容易将数据划分为两簇。

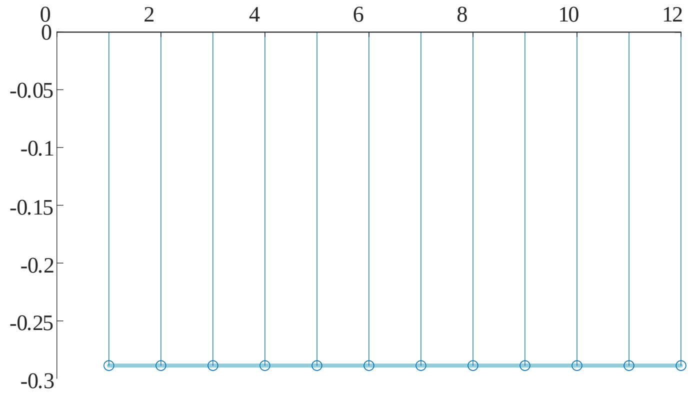


图 12. 特征向量 $\boldsymbol{v}_1$ 结果

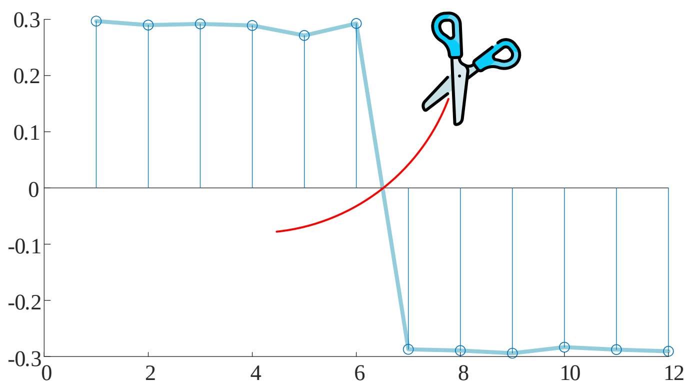


图 13. 特征向量 $\boldsymbol{v}_2$ 结果

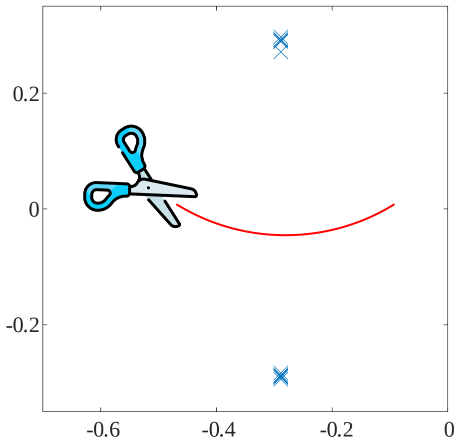


图 14. 矩阵 $\boldsymbol{L}$ 投影到低维度正交空间结果

图 15 所示为采用谱聚类算法对环形样本数据聚类结果。谱聚类的可调节参数包括：相似度矩阵可以使用不同的相似度度量方式。拉普拉斯矩阵可以采用不同类型。特征向量数量可以影响聚类效果。最终的聚类可以选择不同算法。

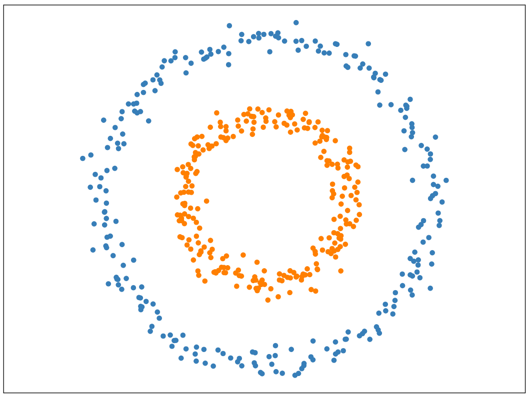


图 15. 环形样本数据聚类结果

代码 Bk7_Ch16_01.py 可以获得图 15。

谱聚类是一种基于图论的聚类算法，其特点是能够处理高维数据和非凸数据簇，并且对于数据分布的形态没有特殊要求。谱聚类通过将数据点看作图中的节点，将它们之间的相似度构成的矩阵称为邻接矩阵，通过对邻接矩阵进行谱分解，得到数据点的特征向量，进而将其映射到低维

空间进行聚类。优点是可以在任意维度上进行聚类，并且不会受到噪声的影响。缺点是需要进行谱分解计算，计算量较大。

请大家注意，拉普拉斯矩阵 $\boldsymbol{L}$ 为**半正定矩阵** (positive semi-definite matrix)。证明过程请参考 Ulrike von Luxburg 创作的 *A Tutorial on Spectral Clustering*。