

25

Spectral Clustering

谱聚类

构造无向图，降维聚类



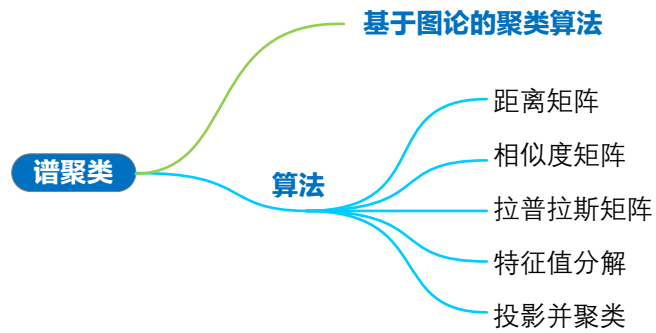
生命中最重要的问题，几乎都是概率问题。

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



25.1 谱聚类

谱聚类 (spectral clustering) 是一种基于**图论** (graph theory) 的聚类算法，能够处理高维数据，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。



《数据有道》专门介绍过有向图、无向图这些概念，请大家回顾。

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** (spectral decomposition)，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。



注意，谱分解是一种特殊的特征值分解。

流程

上述思路虽然简单，但是实际操作需要一系列矩阵运算。

首先，需要计算数据矩阵 X 内点与点的成对距离，并构造成距离矩阵 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) S ，利用 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) L 。

最后，**特征值分解** (eigen decomposition) L ，相当于将 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。

图 1 所示为谱聚类的算法流程。

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。

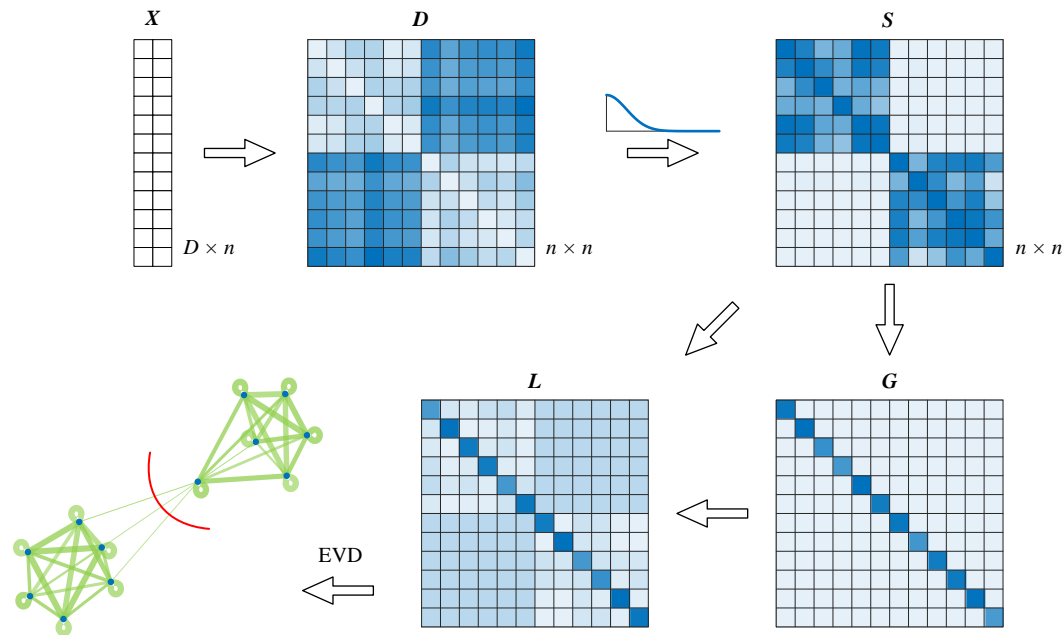


图 1. 谱聚类算法流程

25.2 距离矩阵

图 2 给出 12 个样本点在平面上位置。计算数据**成对距离** (pairwise distance), $\mathbf{x}^{(i)}$ 和 $\mathbf{x}^{(j)}$ 两个点之间欧氏距离 $d_{i,j}$

$$d_{i,j} = \|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\| \quad (1)$$

其中, 约定 $\mathbf{x}^{(i)}$ 和 $\mathbf{x}^{(j)}$ 均为列向量。注意, 这里的 $d_{i,j}$ 非负。

图 3 所示为热图描绘的 12 个样本点成对欧氏距离构造的矩阵 D 。色块颜色越浅, 说明距离越近; 色块颜色越深, 说明距离越远。

观察图 3, 显而易见矩阵 D 为**对称矩阵** (symmetric matrix), 也就是说

$$d_{i,j} = d_{j,i} \quad (2)$$

⚠ 注意, 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。

图 4 所示为计算成对距离矩阵 D 的原理图。

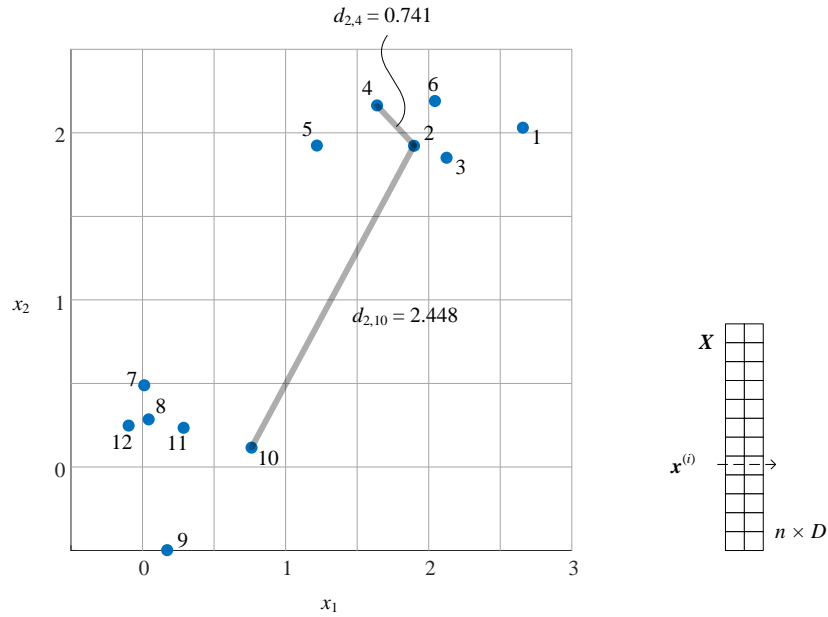
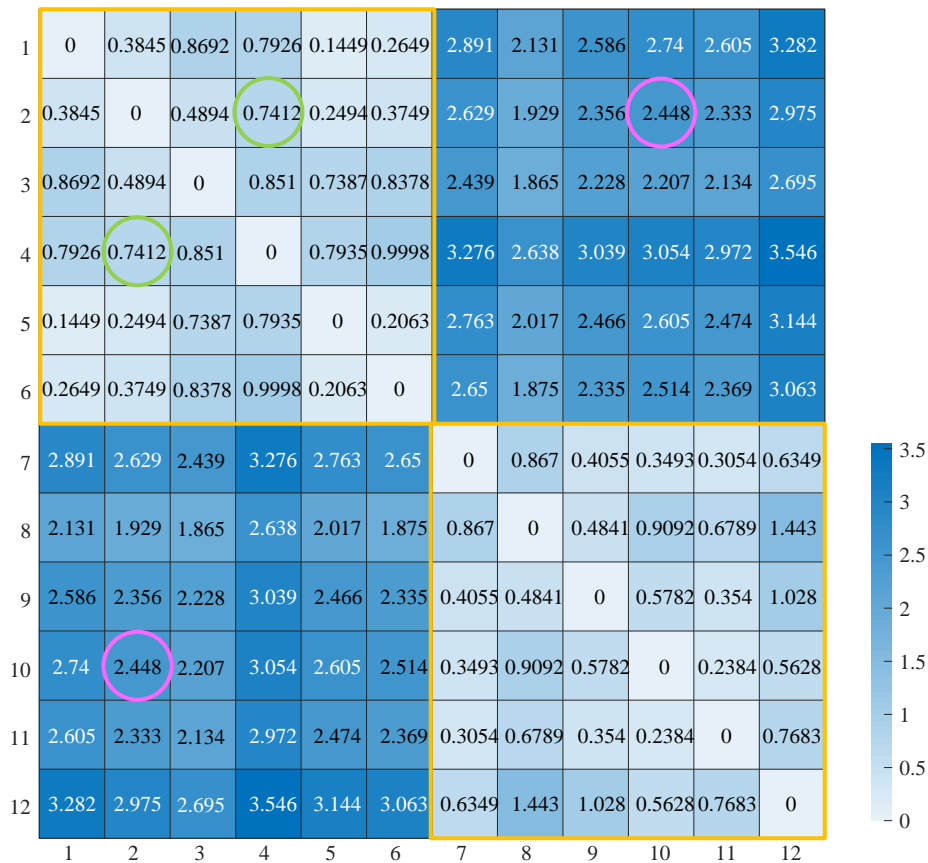
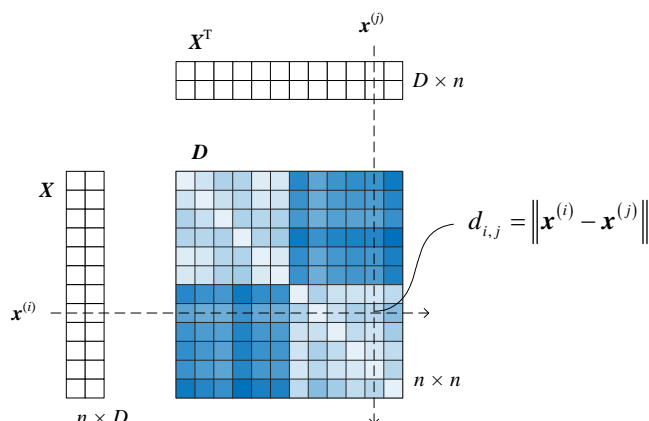


图 2. 12 个样本点平面位置

图 3. 12 个样本点成对欧氏距离构造的成对距离矩阵 D

图 4. 计算成对距离矩阵 D

25.3 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 \rightarrow 相似度”的转换采用高斯核函数:

$$s_{i,j} = \exp\left(-\left(\frac{d_{i,j}}{\sigma}\right)^2\right) = \exp\left(-\frac{\|x^{(i)} - x^{(j)}\|^2}{\sigma^2}\right) \quad (3)$$

相似度取值区间为 $(0, 1]$ 。

$x^{(i)}$ 和 $x^{(j)}$ 两个点距离越近, 它们的相似性越高, 越靠近 1; 反之, 距离越远, 相似度越低, 越靠近 0。任意点和自身的距离为 0, 因此对应的相似度为 1。

参数 $\sigma = 1$ 时, 成对距离 $d_{i,j}$ 和相似度 $s_{i,j}$ 两者关系如图 5 所示。

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

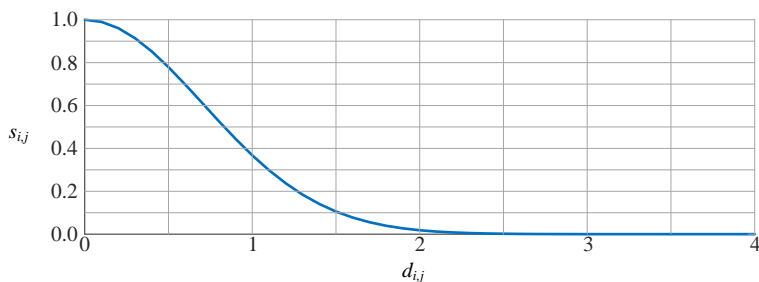


图 5. 欧氏距离和相似度关系

参数 σ 可调节, 图 6 所示为参数 σ 对 (3) 高斯函数的影响。

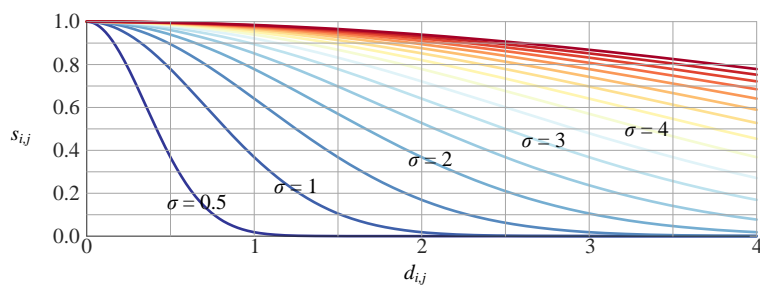
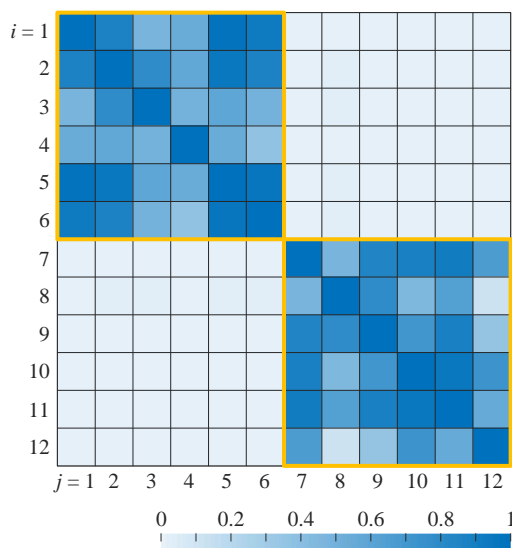
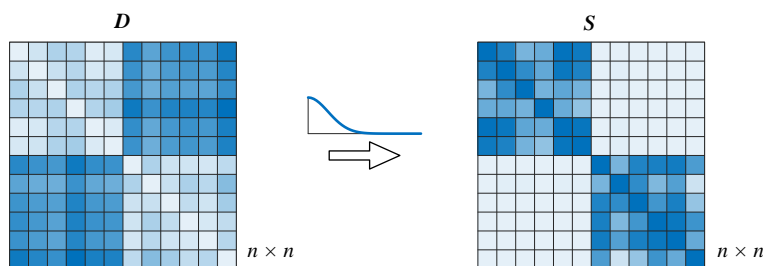
图 6. 参数 σ 对高斯函数的影响

图 3 所示成对距离矩阵转化为图 7 所示**相似度矩阵** (similarity matrix) S 。 S 也叫**邻接矩阵** (adjacency matrix)。相似度矩阵 S 的每个元素均大于 0。请大家注意，一些教材将成对距离矩阵 D 叫做相似度矩阵。从图 7 一眼就可以看出数据可以划分为两簇。

图 8 所示为距离矩阵 D 转化成相似度矩阵 S 的原理。

图 7. 12 个样本点成对相似度矩阵 S 图 8. 距离矩阵 D 转化成相似度矩阵 S

25.4 无向图

图 9 为相似度矩阵 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 成对元素之间的联系得到无向图图 10。

如图 11 所示，在图 10 基础上进一步切断相似度小于 0.005 成对元素之间的联系得到无向图。

观察图 12 可以知道，当切断相似度小于 0.031 成对元素之间的联系，可以将原始数据划分为两簇。

本章后文用特征值分解方法来完成簇划分。

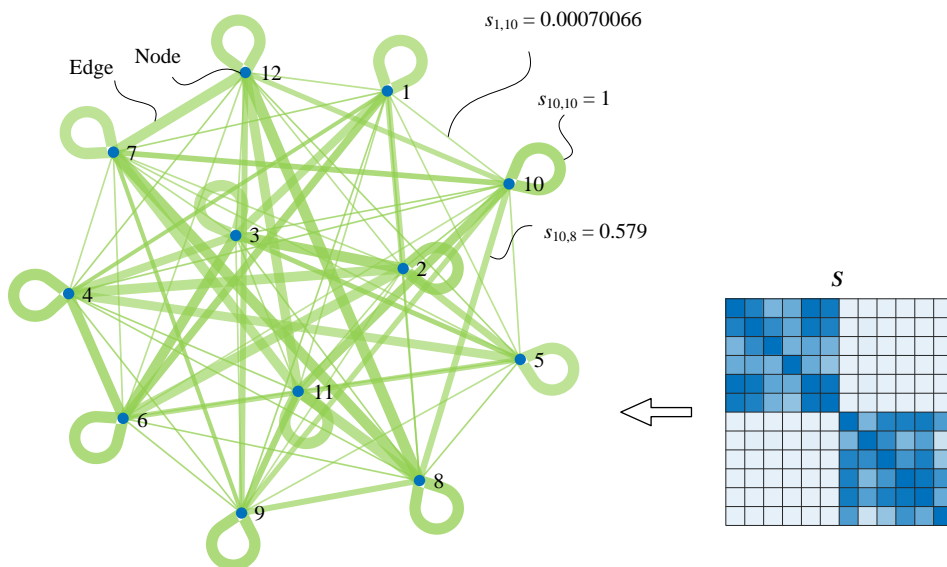
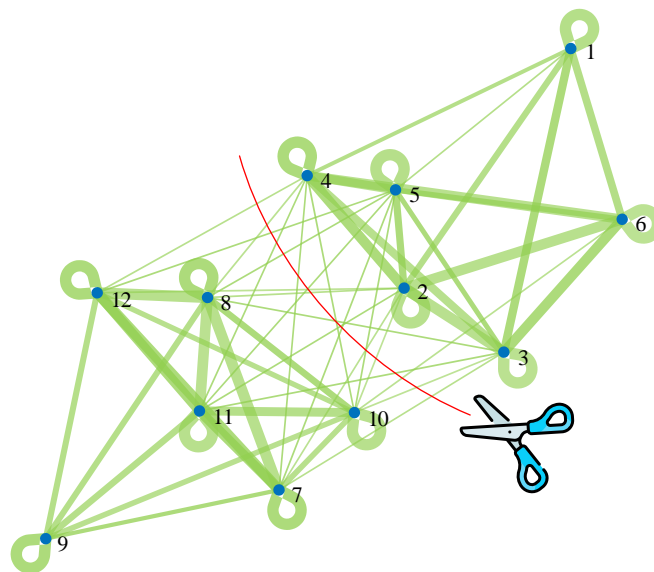


图 9. 相似度对称矩阵 S 无向图



本 PDF 文件为作者草稿，发布目的为方便大家在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com

图 10. 当切断相似度小于 0.001 成对元素之间的联系得到无向图

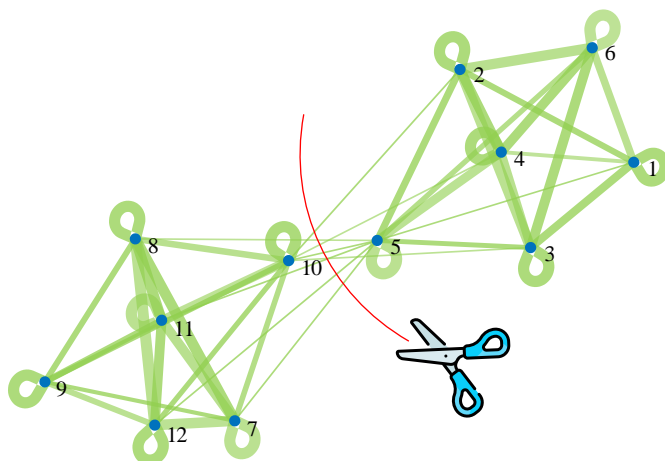


图 11. 当切断相似度小于 0.005 成对元素之间的联系得到无向图

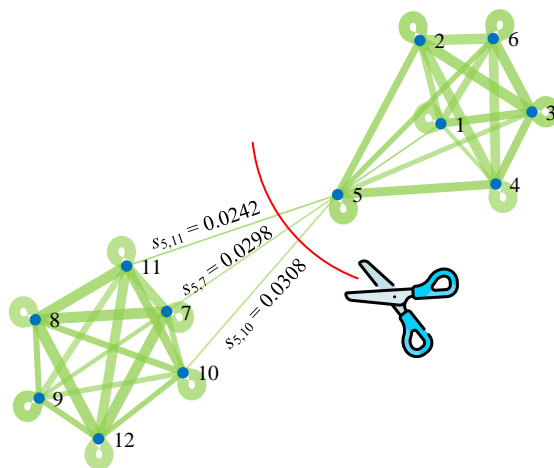


图 12. 当切断相似度小于 0.02 成对元素之间的联系得到无向图

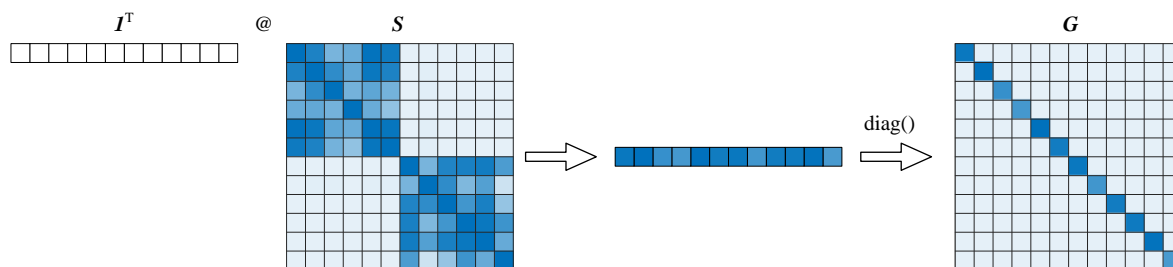
25.5 拉普拉斯矩阵

如图 13 所示，**度矩阵** (degree matrix) G 是一个对角阵。 G 的对角线元素是对应相似度矩阵 S 对应列元素之和，即：

$$G_{i,i} = \sum_{j=1}^n s_{i,j} = \text{diag}(I^T S) \quad (4)$$

图 14 所示为计算度矩阵 G 的原理。

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

图 13. 12 个样本点成对相似度构造的度矩阵 G 图 14. 计算的度矩阵 G 原理

拉普拉斯矩阵

然后构造**拉普拉斯矩阵** (Laplacian matrix) L 。有三种常用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：

$$L = G - S \quad (5)$$

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：

$$L_{rw} = G^{-1} (G - S) \quad (6)$$

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:

$$L_s = G^{-1/2} (G - S) G^{-1/2}$$
(7)

采用第一种方法获得拉普拉斯矩阵 L , 热图如图 15 所示。图 16 所示为用 (5) 计算 L 的原理。

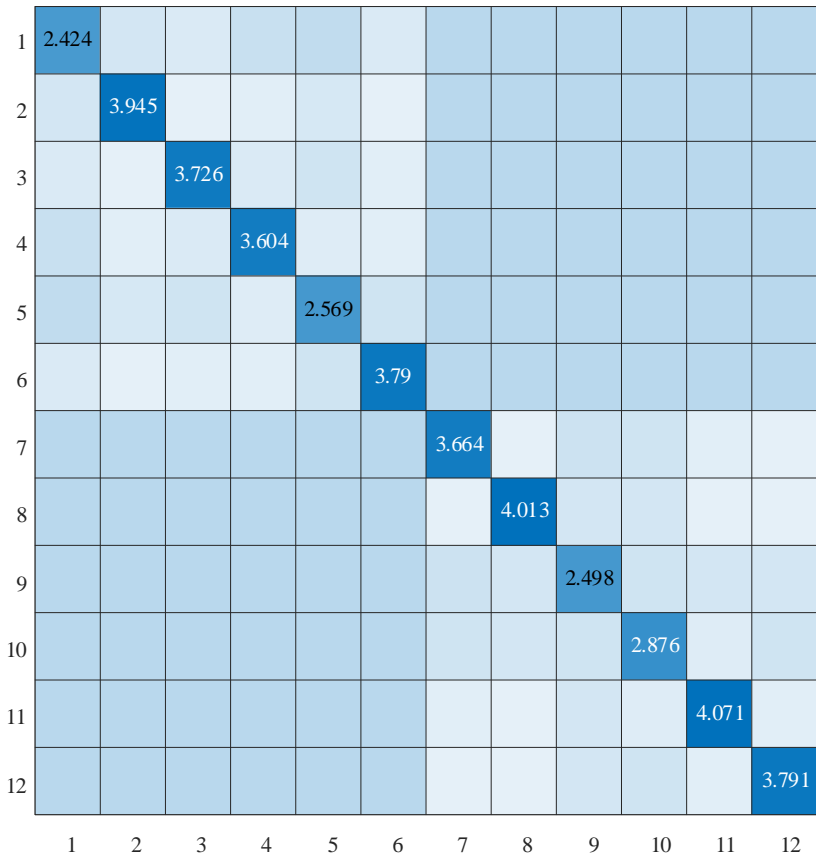


图 15. 12 个样本点成对相似度构造未归一化拉普拉斯矩阵 L

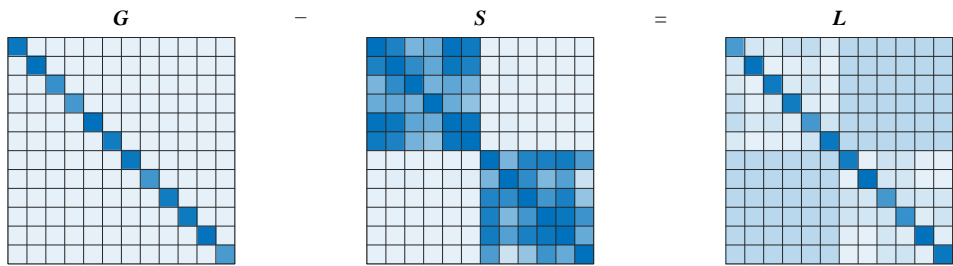


图 16. 计算未归一化拉普拉斯矩阵 L



请大家注意，拉普拉斯矩阵 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*。

25.6 特征值分解

对拉普拉斯矩阵 L 进行特征值分解：

$$L = V\Lambda V^{-1} \quad (8)$$

其中

$$\Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_{n2} \end{bmatrix}, \quad V = [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \dots \quad \mathbf{v}_1] \quad (9)$$

图 17 所示为拉普拉斯矩阵 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 值后，第 2 个特征值 $\lambda_2 = 0.01285$ ，对应的特征向量 $\mathbf{v}_2 = [-0.300, -0.295, -0.297, -0.294, -0.275, -0.298, 0.283, 0.285, 0.288, 0.278, 0.284, 0.286]$ 。

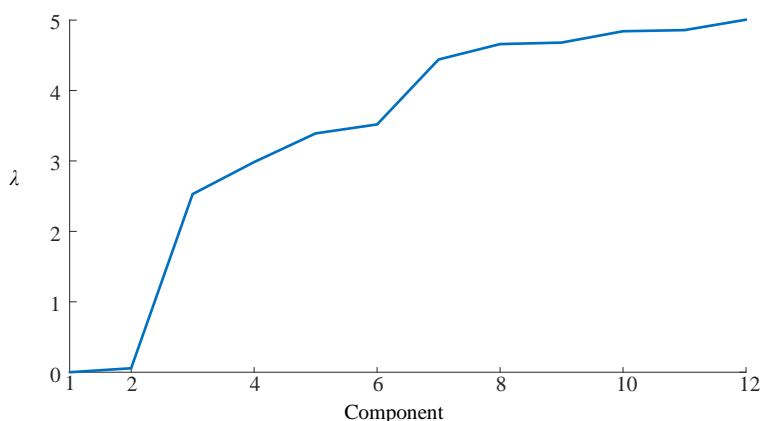


图 17. 拉普拉斯矩阵 L 特征值分解得到的特征值从小到大排序

图 18 和图 19 分别展示前两个特征向量的结果。相当于将拉普拉斯矩阵 L 投影到一个二维空间，具体如图 20 所示。在图 20 所示平面内，可以很容易将数据划分为两簇。

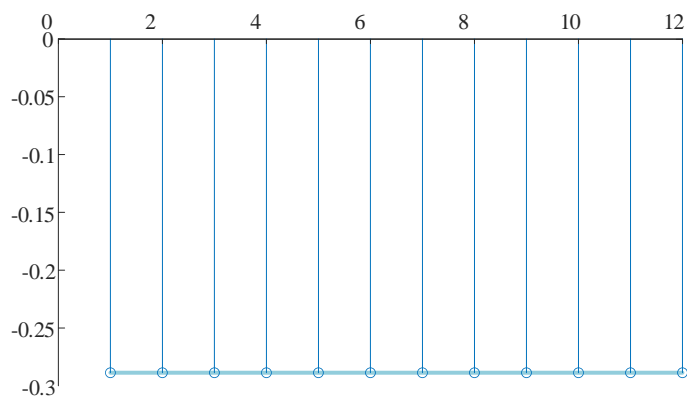
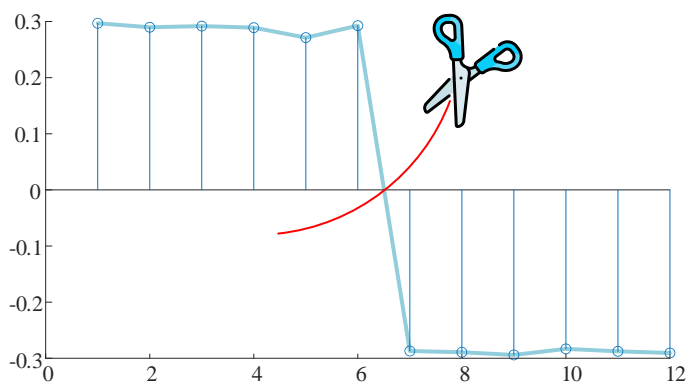
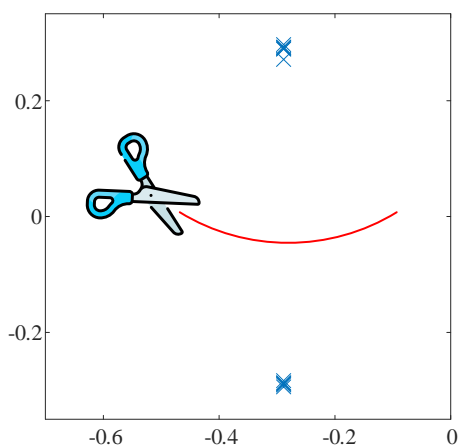
图 18. 特征向量 v_1 结果图 19. 特征向量 v_2 结果图 20. 矩阵 L 投影到低维度正交空间结果

图 21 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有很多。比如，高斯核函数中的参数 σ 。相似度矩阵也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。

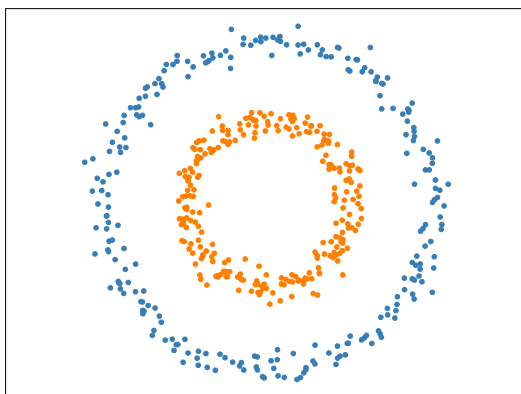


图 21. 环形样本数据聚类结果

代码 Bk7_Ch25_01.ipynb 可以获得图 21。下面聊聊其中核心语句。

```

n_samples = 500;
# 样本数据的数量

a dataset = datasets.make_circles(n_samples=n_samples,
                                  factor=.5, noise=.05)
# 生成环形数据

b X, y = dataset
# X 特征数据, y 标签数据

c X = StandardScaler().fit_transform(X)
# 标准化数据集

d spectral = cluster.SpectralClustering(
    n_neighbors = 20,
    assign_labels='discretize',
    eigen_solver="arpack",
    affinity="nearest_neighbors",
    n_clusters=2)
# 使用SpectralClustering算法对数据进行聚类

e y_pred = spectral.fit_predict(X)
# 返回每个样本的聚类标签

```

代码 1. 用 `sklearn.cluster.SpectralClustering()` 完成聚类 | Bk7_Ch25_01.ipynb

a 用 `sklearn.datasets.make_circles()` 生成环形结构两特征数据集。`n_samples` 指定生成的样本数量。`factor` 为控制内外环的大小的参数。`factor` 值在 0 到 1 之间，表示环的直径与内环直径之比。在这里，`factor=0.5` 表示外环直径是内环直径的两倍。`noise` 为添加到数据集中的高斯噪声的标准差。

b 将特征数据和标签数据分离。在聚类问题中，我们仅仅需要特征数据。

c 用 Scikit-learn 库中的 `StandardScaler` 来标准化数据集 `X`。数据处理结果的均值为 0，标准差为 1。

本 PDF 文件为作者草稿，发布目的为方便大家在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微视频均发布在 B 站——生姜 DrGinger: <https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com

d 用 `sklearn.cluster.SpectralClustering()` 完成聚类。`n_neighbors=20` 指定了用于构建 k 近邻图的邻居数目，即在图中每个数据点连接到其最近的 20 个邻居。

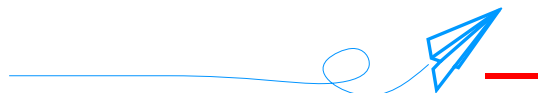
`assign_labels='discretize'` 表示在谱聚类过程中如何分配标签。在这里，它使用的是离散化的方法，将谱聚类的结果转换为离散的类别。

`eigen_solver="arpack"` 指定了求解特征值问题算法。

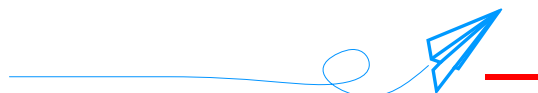
`affinity="nearest_neighbors"` 指定了用于计算相似度矩阵的方法。

`n_clusters=2` 指定了聚类的簇数目，即将数据分为两个簇。

e 对数据集进行谱聚类，并返回聚类标签。



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。



亲爱的同学们，读到这里，大家已经走完了 7 册的“鸢尾花书之旅”。

崭新的知识爆炸出现，尘封的知识被再次挖掘，已有的知识被跨学科应用，错误的理论被推翻、被修正 ... 前所未有的地，在人工智能的助力下，人类知识边界时刻延展、加速伸延。

鸢尾花书 7 册没有创造任何新知识；套用牛顿的话，面对知识的海洋，笔者仅仅打捞了几篓贝壳，将它们擦得闪亮，摆成了自以为漂亮的图案和大家分享。

面对这片浩瀚的充满未知的真理海洋，我们保持谦卑，保持好奇；与此同时，笔者始终坚信，那些热爱知识、不懈探索的读者朋友们，定能拓荒新领域，扩延人类知识星辰大海的边界。

笔者不敢奢求太多，只希望鸢尾花书能化作大家翅膀上几片羽毛，让同学们的羽翼更加丰满。

起风了。

虽百般不舍，飞走吧。

大鹏一日同风起，扶摇直上九万里！

带着乡亲们、鸡兔猪小伙伴们的期许和惦念，飞的更高些，飞的更远些！



懂的越多，便越自觉无知。

The more you know, the more you know you don't know.

—— 亚里士多德 (Aristotle) | 古希腊哲学家 | 384 ~ 322 BC