

作者	生姜 DrGinger
脚本	生姜 DrGinger
视频	崔崔 CuiCui
开源学习资源	https://github.com/Visualize-ML
平台	https://www.youtube.com/@DrGinger_Jiang https://space.bilibili.com/3546865719052873 https://space.bilibili.com/513194466

15.4 拉普拉斯矩阵和谱聚类



本节你将掌握的核心技能：

- ▶ 谱聚类基本流程。
- ▶ 无权无向图邻接矩阵是对称矩阵，元素为 1 表示有连接，0 表示无连接。
- ▶ 无向图度矩阵是对角矩阵，对角线元素是节点的度数，反映节点的活跃程度。
- ▶ 度数高的节点通常是社交网络的核心人物，度数低的节点往往处于边缘地位。
- ▶ 拉普拉斯矩阵通过度矩阵减去邻接矩阵得到，体现图的连接结构。
- ▶ 归一化处理可以消除节点度数差异，防止高连接节点主导聚类结果。
- ▶ 选取最小特征值对应的特征向量，节点在低维空间中位置相近代表社交关系紧密。

本节介绍如何利用谱聚类完成人际网络分析。谱聚类中的核心要素是拉普拉斯矩阵，这个矩阵通过前文介绍的度矩阵、邻接矩阵构造。重要的运算工具就是大家习以为常的特征值分解。

人际关系网络

人际交往中的好友关系图就是典型的无向图，节点表示人，边表示两人之间的好友关系。空手道俱乐部会员关系图是一张经典的无向图，用于描述一群俱乐部成员之间的社交关系。

在图 1 这张无向图中，节点表示俱乐部中的成员；边表示两个成员之间的关系，例如是否经常一起训练或社交。图中数字为俱乐部成员编号。

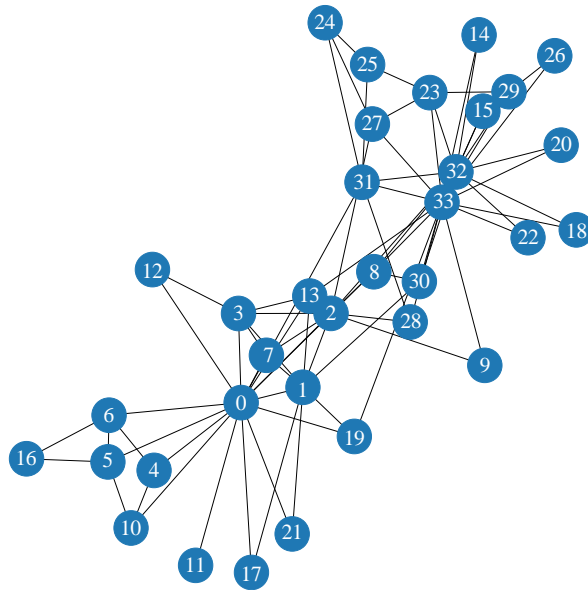


图 1. 空手道俱乐部会员无向图

这是一个常用于研究网络结构的著名数据集，由 Zachary 在 1977 年的研究中提出，这个无向图也因此得名 Zachary's Karate Club。

它最初被用来分析群体内的社交动态和分裂现象。在 Zachary 的研究中，由于俱乐部领导人之间的冲突，会员最终分裂成了两个群体。这种分裂可以通过无向图中的社区检测算法自动识别。

谱聚类

下面我们用**谱聚类** (spectral clustering) 对图 1 节点聚类。

聚类 (clustering) 是一种将数据根据相似性划分成若干**簇** (cluster) 的方法。在聚类中，同一簇内的数据彼此之间应当尽可能相似，不同簇之间的数据则尽可能不同。

简单来说，聚类就是把相似的东西归到一起，把不相似的东西分开。

谱聚类的基本流程如下：

- ▶ 首先，给定一组数据或一个图结构，先计算其无向图的**邻接矩阵**。
- ▶ 接着，计算**度矩阵**。
- ▶ 然后，根据**度矩阵**、**邻接矩阵**，计算图的**拉普拉斯矩阵**。
- ▶ 计算**归一化拉普拉斯矩阵**。
- ▶ 对**归一化拉普拉斯矩阵**进行特征值分解。
- ▶ 对应的特征向量按升序排列，选取前几个最小特征值对应的特征向量，构造低维空间。
- ▶ 最后，在这个低维空间中，对节点进行聚类。

下面，我们就按照这个思路展开本节讨论。

邻接矩阵

根据图 1 这幅无向图节点之间的连接关系，我们首先计算得到如图 2 所示的邻接矩阵 A 。

本章前文提过，无向图的邻接矩阵 A 为对称矩阵。

特别对于无权无向图，邻接矩阵 A 元素为 0 意味着节点之间不存在边；元素为 1 意味着节点存在一条边。

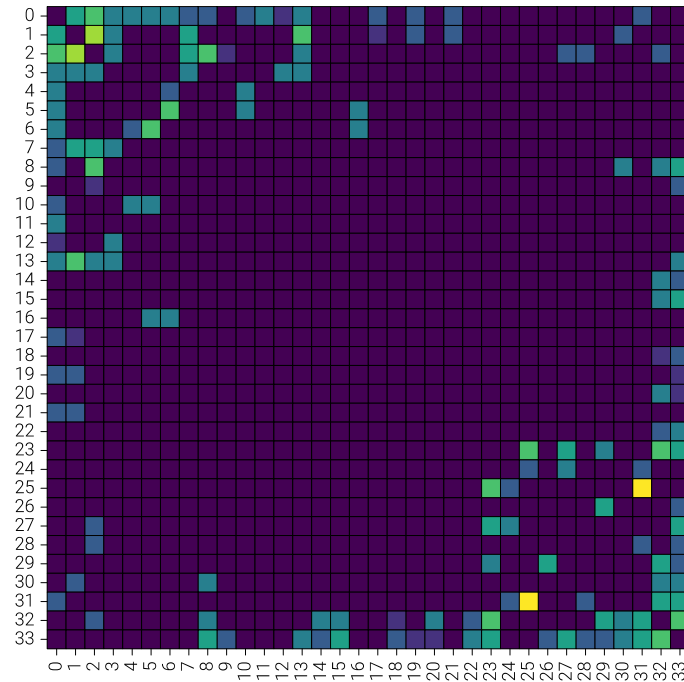


图 2. 邻接矩阵

度矩阵

接着，计算度矩阵 D 。

如图 3 所示，无向图度矩阵 D 是一个对角矩阵，每个对角线元素是对应节点的度数，即节点连接的边的数量之和。

? 请大家回顾无向图度矩阵、邻接矩阵两者关系，并思考如何用 Python 通过邻接矩阵计算度矩阵。

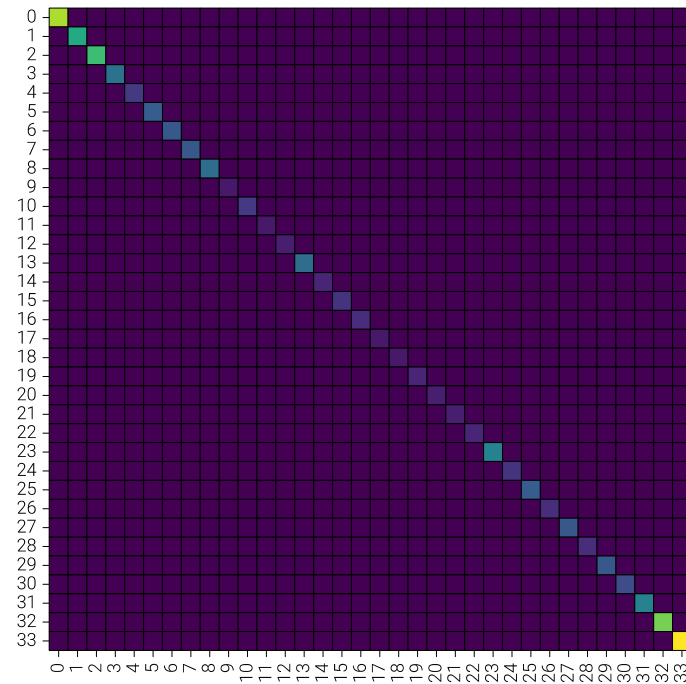


图 3. 度矩阵

节点度数是指一个节点连接的边的数量。节点度数越高，意味着该节点在网络中越活跃、越重要，它与更多的其他节点建立了直接联系，通常在信息传播、资源分配或社交网络中起到关键作用。

图 4 所示为各个节点度数的柱状图。容易发现，编号为 0、33 节点的度数特别高；他俩相当于群体中的“核心人物”。

相反，节点度数越低，意味着该节点在网络中的联系较少，它可能是边缘角色，参与度较低，或者在结构上处于孤立的位置。

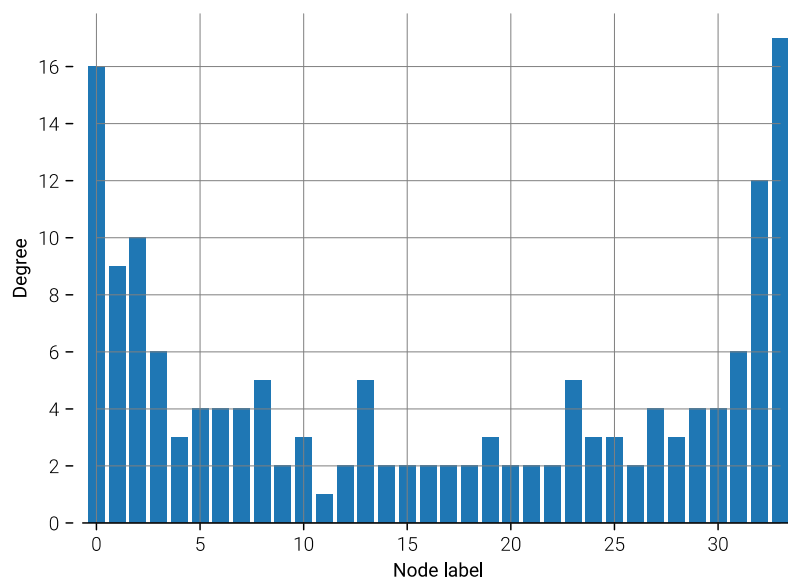


图 4. 各个节点的度

总体上，度数反映了节点在网络中的直接影响力和参与程度，是衡量节点地位的重要指标。图5所示为利用度数大小渲染节点，我们似乎已经发现了两个“朋友圈”。

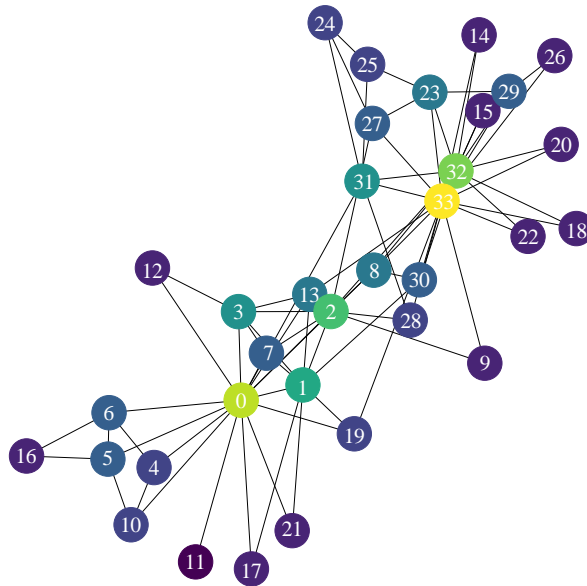


图 5. 根据度数渲染节点

拉普拉斯矩阵

然后，根据度矩阵 D 和邻接矩阵 A ，计算图的拉普拉斯矩阵 (Laplacian matrix) L

$$L = D - A \quad (1)$$

对于无向图，度矩阵 D 和邻接矩阵 A 都是对称矩阵，显然拉普拉斯矩阵 L 也是对称矩阵，具体如图6所示。

拉普拉斯矩阵是图结构的一种重要表示，它反映了节点之间的连通关系和整体结构特性。拉普拉斯矩阵的特征值和特征向量还能揭示网络内部的分区结构。

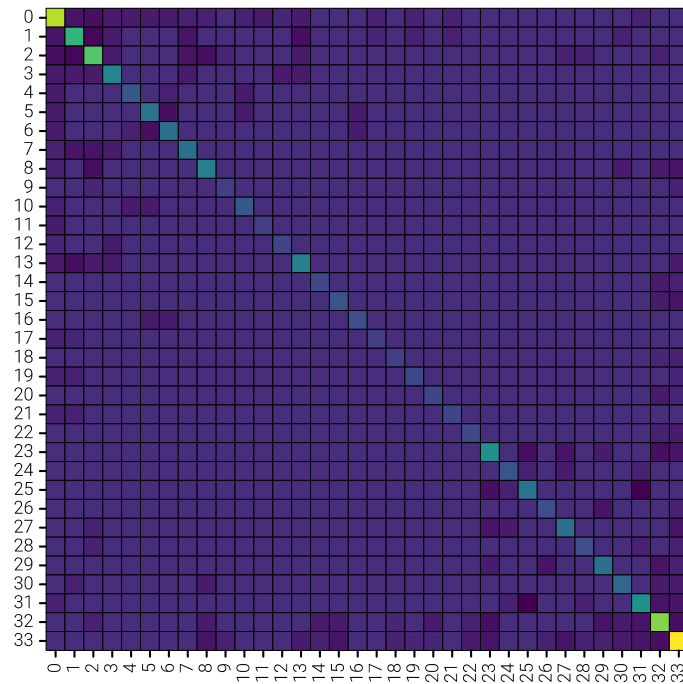


图 6. 拉普拉斯矩阵

归一化拉普拉斯矩阵

进一步，为了消除节点度数大小对计算的影响，通常需要计算归一化拉普拉斯矩阵 (symmetrically normalized Laplacian)

$$L_{\text{normalized}} = D^{-1/2} L D^{-1/2} = D^{-1/2} (D - A) D^{-1/2} \quad (2)$$

即在拉普拉斯矩阵 L 两边分别乘以度矩阵 D 的负平方根，相当于对 L 的行列分别进行缩放。

如图 7 所示，归一化拉普拉斯矩阵的主对角线元素均为 1。

原始的拉普拉斯矩阵在度数分布差异很大的图中，容易让高连接节点主导整体结构，导致聚类或分割结果失衡。而归一化处理后，相当于把每个节点的连接强度进行了标准化，让所有节点在计算中处于更公平的位置。

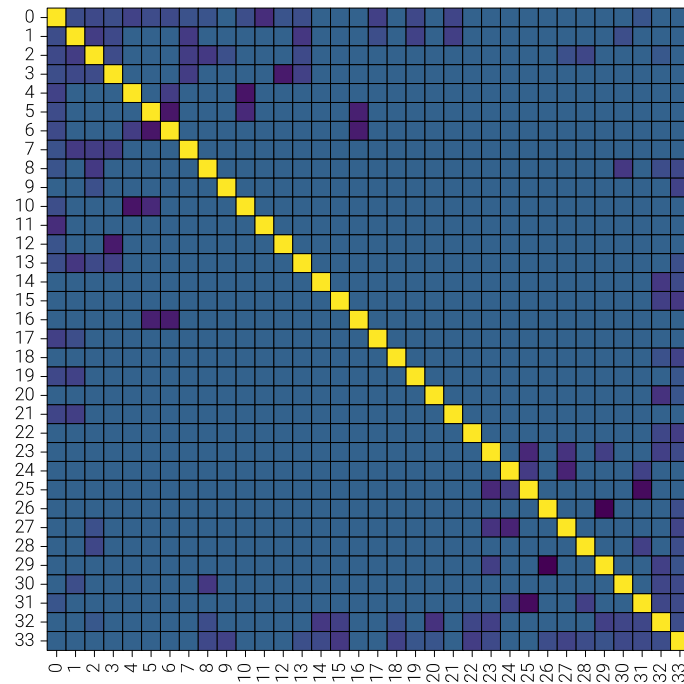


图 7. 归一化拉普拉斯矩阵

特征值分解

得到归一化拉普拉斯矩阵后，进行特征值分解，

$$L_{\text{normalized}} = V \Lambda V^T \quad (3)$$

结果如图 8 所示。

由于归一化拉普拉斯矩阵为实对称矩阵，因此图 8 为谱分解。

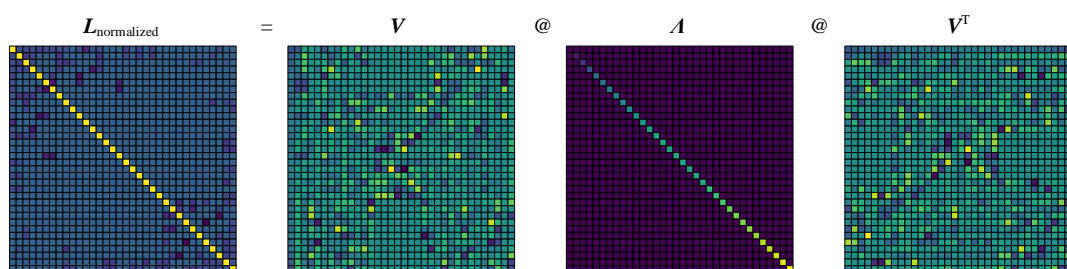


图 8. 归一化拉普拉斯矩阵的特征值分解 (谱分解)

根据特征值的大小，将对应的特征向量按升序排列，如图 9、图 10 所示。

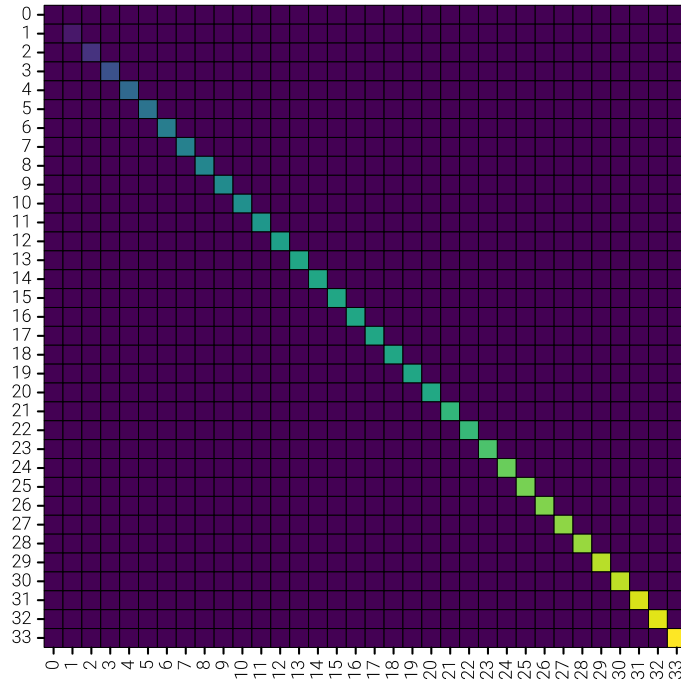


图 9. 特征值对角方阵

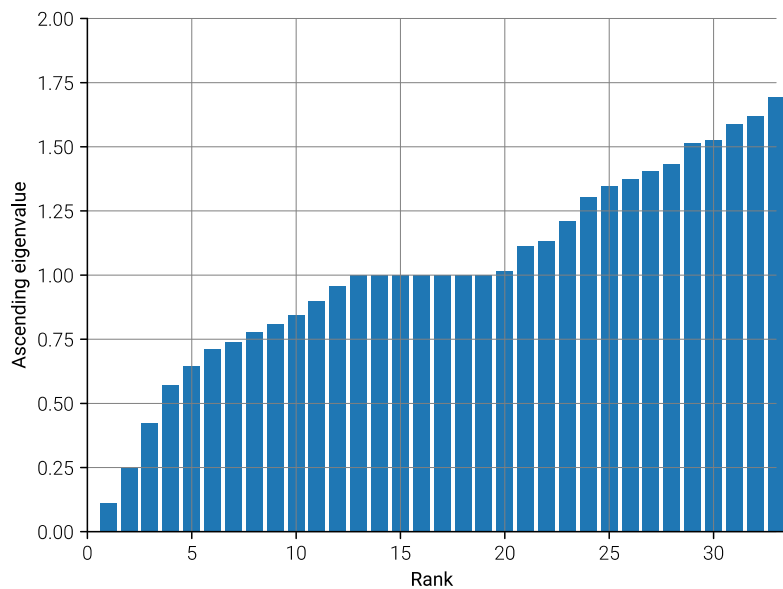


图 10. 特征值递增排列

由于图 1 无向图相对简单，我们选取前 2 个最小特征值对应的特征向量（图 11 中正交矩阵的前两列 v_1 、 v_2 ），然后再这些特征向量构成新的低维空间的表示成员关系，如图 12 所示。

图 12 红色散点纵轴坐标大于 0，蓝色散点纵轴坐标小于 0；实际上，我们已经将成员分割成了两簇，具体如图 13 所示。

归一化拉普拉斯矩阵本质上体现了节点之间的相似性。谱分解后，特征向量把每个节点映射到了一个新的低维空间中，在这个空间里，相互连接紧密的节点，在特征向量空间的位置会很接近；连接稀疏的节点，在特征向量空间的位置会比较远。

主成分分析中，我们选择最大特征值对应的投影方向是因为在这个方向上，数据分布得最为分散，信息量最大。而谱聚类的目的是聚合，找到的这个方向是让连接紧密的节点更接近，从而方便形成簇。

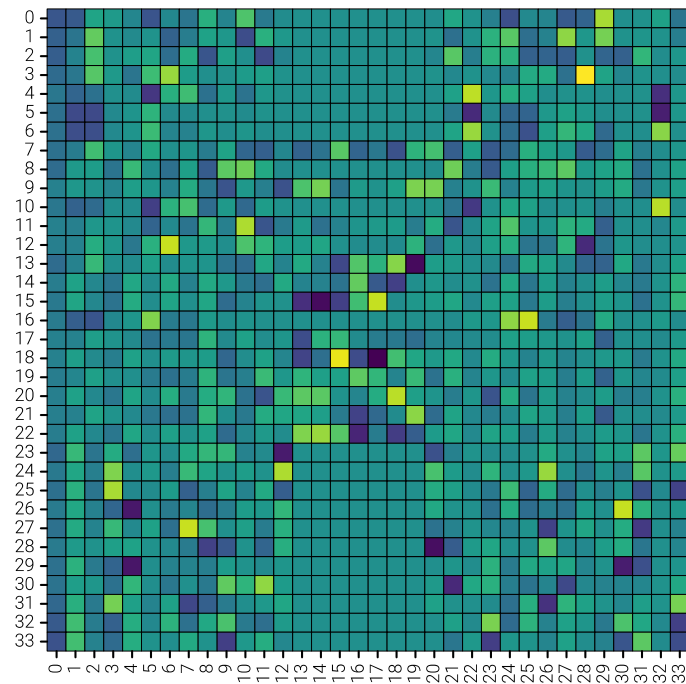


图 11. 正交矩阵 V

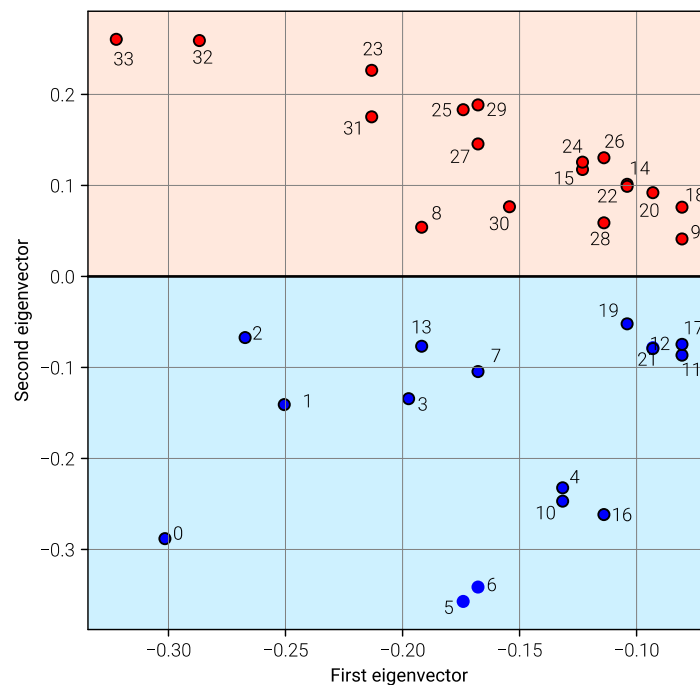


图 12. 前两个特征向量构成的散点图

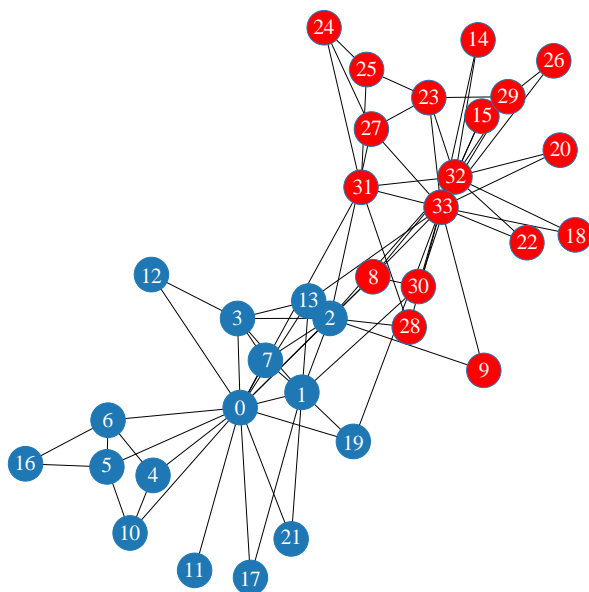


图 13. 分割无向图



LA_15_04_01.ipynb 完成本节谱分解所有运算以及可视化。

尾声

把线性代数学懂、用好是《线性代数不难》的追求。希望在 Python 编程、几何化思维、Manim 数学动画的助力下，《线性代数不难》真的让线性代数“不难”！

线性代数的学习，不应是枯燥无味的数字游戏，更不应该是一段死记硬背公式的痛苦旅程；线性代数的学习，应是一场几何视觉的盛宴，一次深入数字丛林的探险。

《线性代数不难》希望帮助读者打开一扇窗，看到线性代数背后丰富的几何直觉、应用价值，甚至是数学之美。真正掌握线性代数，意味着能以更高的视角审视数据、模型、现象与世界。

当然，本书也只是一个起点。线性代数的天地广阔而深远。希望本书能够在大家心中种下一颗种子，让我们在今后的学习与工作中，继续探索、不断成长。

数学之美，不在于复杂，而在于透过抽象，洞察背后的秩序与本质。愿每一位读者，都能以自由而自信的姿态，带着炽热的好奇心，将线性代数化为自己探索世界的不懈力量。