第 2 章 基于特征工程的图机器学习

引言

在机器学习中，特征用于描述数据对象的的属性或变量。它们对于产生准确和易于解释的预测模型，以及在各种数据分析任务中产生良好的结果至关重要。进一步的，特征工程是传统机器学习数据准备中的中心任务，算法的结果质量在很大程度上取决于可用特征的质量。特征工程通过筛选关键信息、减少噪声和冗余、降低维度和复杂度，提升模型性能和泛化能力，并增强模型的可解释性。图机器学习的基本方法（即基于特征工程的图机器学习方法）是抽取图中对象的特征，然后输入给机器学习方法。常用的邻接矩阵虽然包含了图的所有结构信息，但却过于稀疏，需要其他特征来提供更稠密以及特定方面的信息，所以引入了图特征工程。本章将介绍图的节点级特征、边级特征以及图级特征，分别可用于节点级、边级以及图级的机器学习任务。

本章学习目标

(1) 了解特征在机器学习中的重要性，以及特征工程在提高模型性能中的作用；
(2) 掌握节点级特征、边级特征和图级特征的基本概念及其在图机器学习中的应用；
(3) 理解中心性、局部聚类系数和图元度向量等节点级特征的计算方法和实际意义；
(4) 掌握基于距离、局部邻域重合和全局邻域重合的边级特征提取技术；
(5) 掌握图划分的相关概念，以及 WL-核方法等常用的图级特征的计算。

2.1 节点级特征

在图论和图机器学习中，节点级特征（Node-level Features）反映图中单个节点的特性。这些特征对于理解图结构、分析节点的重要性以及进行机器学习任务（如节点分类）具有关键作用。图结构由节点和边组成。节点级特征能够提供节点在图中作用的详细信息，帮助识别图中关键节点、了解节点之间的关系，甚至揭示图的整体结构。这些特征不仅可以反映节点的局部属性，还可以反映节点在图中的重要性。在机器学习任务中，节点级特征常被用作输入数据，用于训练模型。例如，在社交网络中，节点级特征可以用来预测用户的兴趣、行为或群体归属；在生物网络中，节点级特征可以帮助识别重要的基因或蛋白质。

节点级特征种类繁多，可以从不同维度对节点进行描述。常见的节点级特征主要有中心性（Centrality）、局部聚类系数（Local Clustering Coefficient）、图元度向量（Graphlet Degree Vector）等等。

2.1.1 中心性

在图中，节点的中心性（Centrality）度量该节点在图中的某种重要性。在图机器学习中，中心性指标常用于特征工程和网络分析。例如，中心性高的节点可能被标记为关键节点，在节点分类任务中具有重要参考价值。此外，中心性分析还可以帮助理解网络结构，识别潜在的社区或检测网络中的异常行为。本节中将介绍各种度量中心性的方法[3]。

1. 节点度

在图论中，度（Degree）是图中顶点的一个基本属性，是最明显和最直接的节点级特征和中心性的度量方式，表示与该顶点直接相连的边的数量[4]。它反映了节点在图中的连接程度，节点度捕捉到的信息默认了每一个邻居节点信息都是平等的。度在理解图的结构和性质中起着重要作用，也是图机器学习中的一个重要概念。下面将详细介绍图的度及其相关概念。

1) 定义

对于无向图，一个顶点的度是与该顶点相连的边的数量。记作$deg(v)$或$d(v)$。例如，若一个顶点v与三个其他顶点相连，则$deg(v)=3$。

在有向图中，度分为两类：

a.**入度**（In-degree）：指从其他顶点指向该顶点的边的数量，记作$deg^{-}(v)$。

b.**出度**（Out-degree）：指从该顶点指向其他顶点的边的数量，记作 $deg^{+}(v)$。

如图 2-1 所示的无向图，节点 1 的度为 2，节点 3 的度为 3，节点 7 的度为 2。

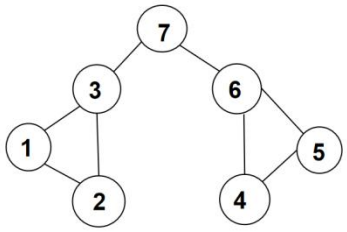


图 2-1 无向图示例图

2) 性质

无向图中的度性质：在无向图中，所有顶点的度的总和等于边数的两倍，即$\sum_{v\in V} deg\,(v)=2|E|$，其中，$V$是图的顶点集合，$E$是边的集合。

有向图中的度性质：在有向图中，所有顶点的入度之和等于出度之和，并且都等于边数，即：$\sum_{v\in V}\deg^{-}(v)=\sum_{v\in V}\deg^{+}(v)=|E|$。

3) 节点度在图机器学习中的应用

在图机器学习中，顶点的度可以用作顶点的一个特征，用于节点分类、链接预测等任务。度的概念还与图卷积神经网络（GCN）中的卷积操作相关，例如在标准的 GCN 中，卷积核的计算与节点的度有直接关系。总结而言，度是图的一个基本且重要的属性，它不仅在图论中有着广泛的应用，在图机器学习中也是一个不可忽视的关键概念。理解和应用度的知识能够帮助更好地分析图结构和设计有效的图学习算法。

节点度只是衡量一个节点有多少个邻居，但这不一定足以衡量节点在图中的重要性。为了获得更强大的重要性度量，可以考虑节点中心性的其他度量方式，以下介绍几个常用的其他中心性指标，分别是：特征向量中心性，中介中心性，接近中心性[4]。

2. 特征向量中心性

虽然基于度的中心性认为一个具有多个邻居的节点是重要的，但它会平等地对待所有的邻居。然而，邻居本身可能有不同的重要性；因此，它们可能会以不同的方式影响中心节点的重要性[3]。

特征向量中心性（Eigenvector Centrality）是一种更复杂的中心性度量，它不仅考虑节点的度，还考虑与该节点相连的节点的重要性。换句话说，如果一个节点与许多高中心性的节点相连，那么该节点的特征向量中心性也会较高，其公式为：

$$c_e(v_i)=\frac{1}{\lambda}\sum_{j=1}^{N}\boldsymbol{A}_{ij}\cdot c_e(v_j)$$

其中，$\boldsymbol{A}$是图的邻接矩阵，λ是一个常数，$c_e(v_i)$是节点v_i的特征向量中心性。

它可以被重写成一个矩阵的形式为：

$$\boldsymbol{c_e}=\frac{1}{\lambda}\boldsymbol{A}\cdot\boldsymbol{c_e}$$

其中$\boldsymbol{c_e}\in\mathbb{R}^N$是一个包含图中所有节点的中心性得分的向量。

显然，$\boldsymbol{c_e}$是矩阵$\boldsymbol{A}$的一个特征向量，λ是其对应的特征值。然而，给定一个邻接矩阵$\boldsymbol{A}$，存在多对特征向量和特征值。通常希望中心性得分是正的。因此，选择一个具有所有正元素的特征向量，而对于连通无向图，邻接矩阵的最大特征值对应的特征向量的所有分量都可以取正值。

因此，可以选择λ作为最大特征值，其对应的特征向量作为中心性得分向量[3]。

经过一系列的变形，以上计算方法实际上等价于如下定义：通过求解特征方程$\lambda \boldsymbol{c} = \boldsymbol{A}\boldsymbol{c}$，可以得到特征向量$\boldsymbol{c}$，当$\lambda$为最大特征值时，第$i$个节点的特征向量中心性即为向量$\boldsymbol{c}$的第$i$个元素。

例如，图 2-1 的邻接矩阵为

$$\begin{pmatrix} 0&1&1&0&0&0&0 \\ 1&0&1&0&0&0&0 \\ 1&1&0&0&0&0&1 \\ 0&0&0&0&1&1&0 \\ 0&0&0&1&0&1&0 \\ 0&0&0&1&1&0&1 \\ 0&0&1&0&0&1&0 \end{pmatrix}$$

其最大特征值为 2.3429，对应的单位特征向量为(0.335 0.335 0.450 0.335 0.335 0.450 0.384)，各元素对应了 1-7 节点的特征向量中心性的值。

3.中介中心性

中介中心性（Betweenness Centrality）又称为最短路径中心性（Shortest-path Centrality），其测量的是一个节点作为其他节点间最短路径“中介”的重要性。若一个节点在各节点对之间的最短路径上出现次数越多，则其在网络中的重要性越高。对某一个节点，表示为图中不含它的节点对连通最短路径经过它的比例。具有高中介中心性的节点通常是网络中的“桥梁”或“瓶颈”，它们在图中的信息传递中扮演关键角色。

中介中心性的计算公式为：

$$c_b(v_i) = \sum_{v_s \neq v_i \neq v_t} \frac{\sigma_{st}(v_i)}{\sigma_{st}}$$

其中，σ_{st}是从节点v_s到节点v_t的最短路径数量，而$\sigma_{st}(v_i)$是这些最短路径中经过节点v_i的路径数量。

为了使中介中心性得分在不同的图之间具有可比性，需要对其进行归一化。一种有效的方法是将中介中心性得分除以给定一个图的最大可能的中介中心性得分。在上式中，当所有节点对之间的最短路径都通过节点v_i时，$c_b(v_i)$可以达到最大值。即$\frac{\sigma_{st}(v_i)}{\sigma_{st}} = 1$，$\forall v_s \neq v_i \neq v_t$。在一个无向图中，总共有$\frac{(N-1)(N-2)}{2}$对节点。因此，一个节点的最大中介中心性得分为$\frac{(N-1)(N-2)}{2}$。然后，将节点$v_i$的归一化中介中心性得分$c_{nb}(v_i)$定义为：

$$c_{nb}(v_i) = \frac{2\sum_{v_s \neq v_i \neq v_t} \frac{\sigma_{st}(v_i)}{\sigma_{st}}}{(N-1)(N-2)}$$

例如，在图 2-1 中如要计算 7 的中介中心性，1、2、3 中任一节点到 4、5、6 的任一节点均要经过 7，其他的节点对则不需经过，则其中介中心性为 9。

3. 接近中心性

接近中心性（Closeness Centrality）是一个节点到图中所有其他节点的平均最短路径长度的倒数。接近中心性越高，意味着该节点能够更快地与其他节点“接触”或到达其他节点。因此，它反映了节点在网络中的传播效率。

接近中心性的计算公式为：

$$c_c(v_i) = \frac{n-1}{\sum_{t \in V} d(\mathrm{v,t})}$$

其中，$d(v,t)$是节点v和节点t之间的最短路径距离，n是图中节点数目。

例如，在图 2-1 中，节点 7 的接近中心性为 0.6。

4. 中心性的比较与选择

不同的中心性度量适用于不同类型的图和问题。度中心性适合简单的网络结构分析；特征

向量中心性更适合捕捉复杂的网络层次；中介中心性在网络控制与优化中非常有用；接近中心性则适合衡量节点的传播潜力。

总结而言，中心性是图分析中的关键工具，通过不同的中心性度量，可以从不同角度理解和分析图中的节点重要性，从而为图机器学习和网络科学研究提供重要的参考依据。

2.1.2 局部聚类系数

在图论中，聚类系数是图中节点倾向于聚类在一起的程度的度量。在大多数现实世界的网络中，尤其是社交网络中，节点倾向于创建紧密结合的群体，群体内部的联系密度相对较高，这种可能性往往大于两个节点之间随机建立联系的平均概率[5]。聚类系数主要有两种度量方式：局部和全局。

局部聚类系数（Local Clustering Coefficient）是图论中用于衡量单个节点在其邻居中形成闭合三角形（即完全子图）的程度的指标。该系数反映了图中节点之间的紧密程度或社群性，在社交网络中的群体发现等领域具有广泛应用。

简单来说，局部聚类系数衡量了一个节点的邻居之间有多大可能性彼此也是连接的。如果邻居之间完全连接，则局部聚类系数为 1；如果没有邻居之间的连接，则系数为 0。

对于度数为d_i的节点i，局部聚类系数定义为：

$$C_i = \frac{E_i}{T_i}$$

其中，E_i表示节点i的邻居实际存在的边的数量，T_i表示节点i的邻居可能（最多）存在的边的数量，$T_i = \frac{d_i \times (d_i - 1)}{2}$[2]。

$C_i = 0$ 表示节点i的邻居都没有相互链接；

$C_i = 1$ 表示节点i的邻居形成一个全连接图，即它们都相互链接；

$C_i = 0.5$ 意味着一个节点的两个邻居有 50%的机会链接。

网络的聚类系数即平均聚类系数：是所有节点的聚类系数的平均值，定义为

$$C = \frac{1}{N}\sum_i C_i$$

图 2-2 以带有 5 个节点的网络为例，展示了三种不同网络结构下节点 i的聚类系数：

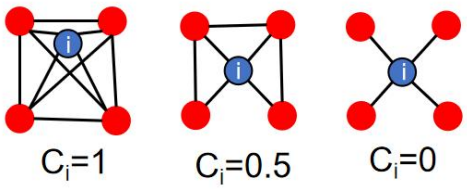


图 2-2 聚类系数图例

在图 2-1 中，节点 3 有三个邻居 1、2、7，并且这些邻居实际存在的边数为 1（1 和 2 之间有一条边），最大可能是 3 条边，所以局部聚类系数是$\frac{1}{3}$。

在图机器学习中，局部聚类系数可以作为节点的一个特征用于预测任务。高聚类系数的节点可能属于某一特定社区，低聚类系数的节点可能是社区之间的桥梁。

2.1.3 图元度向量

图元度向量（Graphlet Degree Vector）：对某一个节点，提取其周围不同图元（graphlet）种类（预先定义好）的个数。它通过考虑节点在各种小型图中的出现频率来捕捉节点的局部结构信息。这种方法为每个节点提供了一个多维的特征向量，其中每个维度对应于一种特定的图

元类型[4]。在理解图元度向量之前，首先要理解图元的概念。

图元的定义为“有根、连通的非同构子图”。简单来说，图元就是若干个节点构成的所有可能的连通图结构，它在较大的网络中作为局部结构的代表。在图 2-3 中可以看到，当仅有两个节点时，可能的图元结构只有一种；当有三个节点时，图元结构有三种，可能是线型连接的，或是构成一个三角形（这里值得注意的是，当以不同的节点为根节点时，图元是不同的，例如G_1结构代表了两种不同的图元表示）；以此类推，当考虑 4 个或 5 个节点时，构成的图元结构将会更多样。

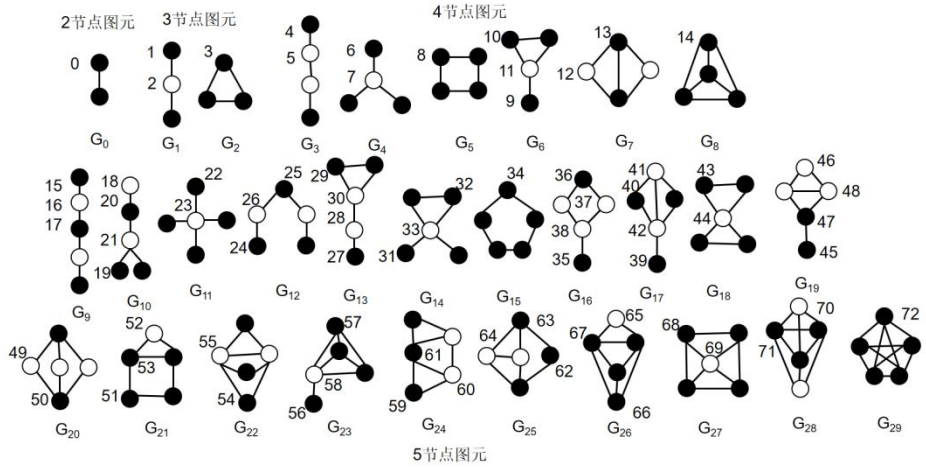


图 2-3 图元结构示例一

什么是图元度向量？指定图元，图元度向量统计了网络中以给定节点为根的各种图元的个数。以图 2-4 为例，网络G中的v是要观察的根节点，a-d 指定了 4 种不同的图元结构，以v为根节点对网络G中四种指定的图元结构进行计数统计，各图元出现的次数分别为 2,1,0,2，由此可以得到节点v的图元度向量为[2,1,0,2]。

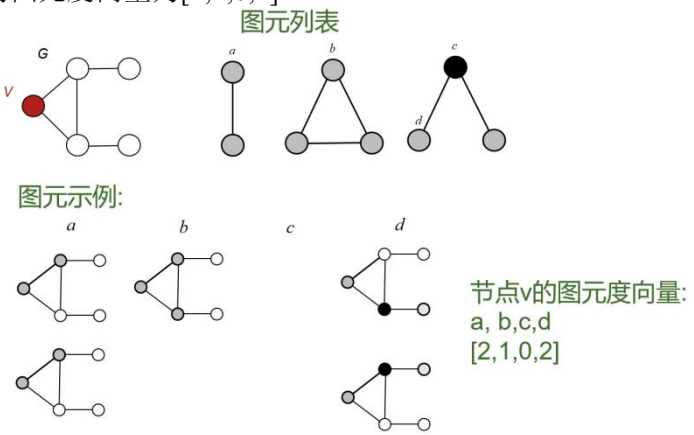


图 2-4 图元结构示例二

2.2 边级特征

传统的链接水平的特征可分为三类：

（1）基于距离的特征：将节点对之间的距离作为链接特征，如节点对之间的最短路径长度；

（2）局部邻域重合：能够捕捉节点对的邻居节点共享情况，相应的指标有：共同邻居数

量、Sorenson 指数、Salton 指数、Jaccard 系数、资源分配指数及 Adamic-Adar 指数；

（3）全局邻域重合：能够使用全局网络结构来为节点对进行打分，例如：Katz 指数，LHN 相似度以及随机游走方法。

2.2.1 基于距离的特征

基于距离的特征可以帮助理解图中节点间的可达性和连接紧密度。一般而言，若两个节点之间的距离越远，可认为该节点对之间链接的重要性越低，产生链接的可能性也越小。最短路径长度（Shortest-path Length）可以作为一种常见的节点对之间距离的衡量方式。在无向图中，最短路径长度可以衡量节点间的接近程度，而在有向图中，它还可以反映方向性。

例如，在图 2-1 中节点 1、7 之间最短路径长度为 2。

基于距离的特征有一个缺点在于，它忽略了路径的具体结构，且不能衡量两个节点之间的共同邻居信息，然而该信息对于新链接的预测往往是非常重要的。以社交网络为例，两个用户的共同好友越多，他们在未来成为好友的可能性通常会越高。在图 2-1 中，1、7 节点与 3、6 节点间的最短路径长度都是 2，但是这两对节点在图结构中的相似性显然是有所区别的。

2.2.2 局部邻域重合

局部邻域重合特征关注节点的直接邻居之间的重合程度，这些特征有助于理解节点的局部结构特性，但它无法衡量两个没有共同邻域节点之间的关系。其常用指标有以下几种：

1. 共同邻居数量

共同邻居数量（Common neighbors）统计了两个节点的共同邻居数量，即节点u的邻居集合与节点v的邻居集合的交集大小，以公式表示为：

$$S(u,v) = |N(u) \cap N(v)|$$

其中，使用$S(u,v)$表示节点u和v之间共同邻居数量，$N(u)$和$N(v)$分别表示节点u和v的邻居集合。两个节点的共同邻居数反映了它们在图中的局部领域重合程度。如果两个节点有大量共同的邻居，则它们可能具有较强的联系。使用共同邻居数量作为衡量指标的缺陷在于：度数越高的节点，与其他节点有共同邻居的可能性也越高，由此会影响衡量链接重要程度的准确性。例如，在图 2-1 中，1、7 节点的共同邻居数量为 1。

2. Sorenson 指数

Sorenson 指数（也称为 Dice 相似系数）是一种基于邻居重合的相似性度量，用于衡量两个节点共享邻居的比例。其计算公式为[4]：

$$\text{Sorenson}(u,v) = \frac{2\cdot|N(u) \cap N(v)|}{|N(u)| + |N(v)|} = \frac{2\cdot S(u,v)}{d_u + d_v}$$

其中$S(u,v)$表示节点u和v之间共同邻居的数量，$|N(u)|$和$|N(v)|$ 分别表示节点u和节点v的邻居集合的大小，即节点u和节点v的度d_u和d_v。

Sorenson 指数的取值范围为 0 到 1，值越大表示两个节点的邻居重合程度越高，其中 0 表示没有重合（即两个集合完全不同），1 表示完全匹配（即两个集合完全相同）。它是一种强调共同邻居相对总邻居数量的度量，适合在网络中识别节点间潜在的强连接。例如，图 2-1 中 1、7 节点 Sorenson 指数为$\frac{2\times1}{2+2} = 0.5$。

3. Salton 指数

Salton 指数（也称为余弦相似度）的计算同样基于节点的邻居信息，表示两个节点的共同邻居数与它们各自邻居数的几何平均值之比。其计算公式为[4]：

$$\text{Salton}(u,v) = \frac{2\cdot|N(u) \cap N(v)|}{\sqrt{|N(u)|\cdot|N(v)|}} = \frac{2\cdot S(u,v)}{\sqrt{d_u d_v}}$$

其中$\sqrt{|N(u)|\cdot|N(v)|}$是两个节点邻居数量的几何平均值，$S(u,v)$表示节点u和v之间共同邻居的

数量。如果两个节点的邻居完全相同，Salton 指数为 1，表示它们高度相似；如果两个节点没有共同邻居，Salton 指数为 0，表示它们完全不相似。

与 Sorenson 指数相比，Salton 指数更注重邻域大小的影响，避免了大度数节点对结果的过度影响。类似的指数还有 Jaccard 指数，是共同邻居数除以两个节点邻居数之和。例如，图 2-1 中 1、7 节点的 Salton 指数为$\frac{2\times1}{\sqrt{2\times2}}=1$。

4. 资源分配指数

资源分配指数（Resource Allocation Index，RA 指数）是一种基于节点邻居重合的相似性度量，广泛应用于图中的链接预测任务。它的思想来源于资源分配的概念，即如果两个节点共享更多具有较少邻居的共同邻居，那么这两个节点之间的联系更有可能被强化。资源分配指数的计算公式如下[4]：

$$\mathrm{RA}(u,v)=\sum_{w\in N(u)\cap N(v)}\frac{1}{|N(w)|}=\sum_{w\in N(u)\cap N(v)}\frac{1}{d_w}$$

其中$|N(w)|$表示与w相连的节点数量，即共同邻居w的度数d_w。例如，在图 2-1 中，1、7 节点的共同邻居只有 3，其度为 3，所以 RA 指数为$\frac{1}{3}$。

资源分配指数的核心思想是通过邻居节点进行“资源传递”。若两个节点u和v共享某个共同邻居w，那么w会将自己的“资源”分配给u和v。然而，如果邻居w的度数较大，则意味着w还与其他很多节点相连，因此它能够分配给u和v的“资源”较少。反之，度数较小的邻居w能分配更多的“资源”。该指数更关注那些度数较小的共同邻居。相比简单的共同邻居数量，资源分配指数加入了邻居节点度数的权重，进一步区分了不同邻居对节点相似性的重要性。

5. Adamic-Adar 指数

与上述 RA 指数相似，Adamic-Adar 指数认为，度数越高的共同邻居节点，在计算链接重要性时应该具有更低的影响力。因此，Adamic-Adar 指数的计算方法是：对两个节点的所有共同邻居节点的度数取 log 对数，并将取对数后的倒数相加求和，其数值越大，说明两个节点联系越紧密。计算公式如下所示：

$$\mathrm{AA}(u,v)=\sum_{w\in N(u)\cap N(v)}\frac{1}{\log|N(w)|}=\sum_{w\in N(u)\cap N(v)}\frac{1}{\log(d_w)}$$

该指数适用于发现较不明显的连接关系，常用于链接预测。

RA 指数和 AA 指数都给予了具有低程度的共同邻居更多的权重，因为直觉上认为共享的低度邻居比共享的高度邻居提供的信息更多。不同之处在于 RA 指数的权重分配基于共同邻居的度的倒数，而 AA 指数使用度的对数的倒数执行类似的计算。

2.2.3 全局邻域重合

局部邻域重合是链接预测的非常有效的方法，其主要是通过共同邻居来衡量节点之间的相似性。然而，仅依靠这种局部邻域信息可能无法充分反映图中更复杂的关系。在图中，有些节点之间可能没有局部邻域的重合，但它们仍然可能属于同一个社区[4]。例如在图 2-5 中，节点A与节点E之间不存在共同邻居时，局部邻域重合特征的取值总是为 0，但它们仍有可能在未来产生链接。实际上，局部邻域重合特征只能捕捉“2 跳”（2-hop）的邻居关系，在该例子中节点A与节点E为 3 跳的邻居关系，此时局部邻域重合特征显然是不适用的。

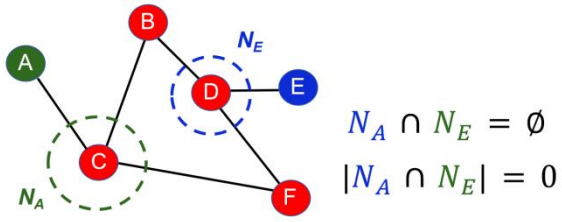


图 2-5 局部邻域重合特征无法捕捉节点 A 与 E 之间的关联信息

全局领域重合正是试图捕捉这种超越局部邻域的、更大范围的关系的方法。它考虑图的整体结构，用来衡量两个节点在全局图中的关系和相似性，适合用于捕捉节点间的深层次联系，适合在处理大规模图或社区结构明显的图时使用。在这里主要介绍 Katz 指数、LHN 相似度和随机游走方法。

1. Katz 指数

Katz 指数是一种基于路径的全局相似度度量，它不仅考虑两个节点间的直接连接，还会考虑通过中间节点的多跳路径。通过对节点之间的所有路径进行加权求和，可以捕捉节点间的间接关系，适合用于分析图的全局结构和预测潜在连接。

给定一对节点(u, v)，Katz 指数的计算公式可表示为[4]：

$$\mathrm{Katz}(u,v) = \sum_{l=1}^{\infty} \beta^l \boldsymbol{A}_{uv}^l$$

其中A_{uv}^l是长度为l的路径数量，$\beta^l \in \mathbb{R}^+$是用户定义的参数，通过对较长的路径赋予较低的权重，使得短路径的影响更大，长路径的贡献被逐步削弱。这个设计使得 Katz 指数既可以捕捉到局部结构，又能通过较长的间接路径探索全局结构。

每个长度的路径数量$\boldsymbol{A}_{uv}^l$可以利用邻接矩阵的幂来计算。例如：邻接矩阵$\boldsymbol{A}$中的元素$\boldsymbol{A}_{uv}$实际上对应了节点u与v之间长度为 1 的路径数量；矩阵的 2 次幂$\boldsymbol{A}^2$中的元素$\boldsymbol{A}_{uv}^2$对应了节点u与v之间长度为 2 的路径数量...以此类推，矩阵的l次幂$\boldsymbol{A}^l$中的元素$\boldsymbol{A}_{uv}^l$即对应了节点u与v之间长度为l的路径数量。通过如上方法可以计算任意节点对之间的 Katz 指数。

如果设$\beta = 0.5$,则图 2-1 中 1、6 节点分别有一条长度为 3 和 4 的路径，其 Katz 指数为 0.1875。

2. LHN 相似度

Katz 指数的一个问题是，它受到节点度的强烈偏差。在考虑高度节点时，Katz 指数通常会给出更高的总体相似性分数，因为高度节点通常会涉及更多的路径[4]。为了解决这一问题，LHN 相似度(Leicht, Holme, and Newman similarity)通过将实际观察到的路径数与期望路径数进行归一化，来消除这种偏差，设节点对为(v_1, v_2)，其计算公式：

$$LHN(v_1, v_2) = \frac{\boldsymbol{A}^i[v_1, v_2]}{E[\boldsymbol{A}^i[v_1, v_2]]}$$

其中$\boldsymbol{A}$表示图的邻接矩阵，$\boldsymbol{A}^i[v_1, v_2]$表示从节点$v_1$到节点$v_2$长度为$i$的路径数。邻接矩阵的幂可以直接用于计算特定路径长度的路径数。期望路径数$E[\boldsymbol{A}^i[v_1, v_2]]$是通过配置模型来计算的，该模型假设绘制的随机图与给定图的度数集合相同。基于这个假设，可以计算出两节点之间的期望边数，即长度为 1 的路径数为：

$$E[\boldsymbol{A}[v_1, v_2]] = \frac{d_{v_1} d_{v_2}}{2m}$$

其中，$\boldsymbol{A}[v_1, v_2]$表示节点v_1和v_2之间是否存在边，若$\boldsymbol{A}[v_1, v_2]$=*1*，则表示两节点之间有边，否则为 0;使用$m = |\mathcal{E}|$来表示图中边的总数。上式表明，在随机配置模型下，边的出现概率与两个节点度数的乘积成正比，即节点的度数越大，它们之间形成边的可能性越大。这可以由节点u有d_u条边，每条边有$\frac{d_v}{2m}$的概率到达节点v来解释。这里分母中的 2 是因为一个图中节点度数之和是边数的两倍。

对于长度为 2 的路径，可以进一步计算其概率：

$$E[\boldsymbol{A}^2[v_1, v_2]] = \frac{d_{v_1} d_{v_2}}{(2m)^2} \sum_{u \in V} (d_u - 1) d_u$$

其中，分别考虑从v_1到中间节点u的路径概率$\frac{d_{v_1} d_u}{2m}$，以及从u到v_2的路径概率$\frac{d_{v_2}(d_u - 1)}{2m}$（其中减

去 1 是因为u的一个边已经用于从v_1到u的入边），在这种情况下，路径的总概率是这两部分概率的乘积。

例如，试计算图 2-1 中，1、7 节点之间$i=2$ 时的 LHN 相似度。节点 1、7 度都为 2，其间有一条长度为 2 的路径，其中间节点 3 的度为 3。代入以上公式：$E[\boldsymbol{A}^2[v_1,v_2]]=\frac{2\times2}{(2\times8)^2}\times 2\times3=0.09375$。故 1、7 节点的 LHN 相似度为$\frac{1}{0.09375}=10.67$。

然而，随着路径长度增加，期望路径数的解析计算变得复杂，为了解决这个问题，Leicht[6]等人使用图的最大特征值来估算路径数的增长，因为图的邻接矩阵和特征值之间存在密切的关系。

任何对称矩阵（如图的邻接矩阵）都可以通过特征值分解来表示:$\boldsymbol{A}=\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{-1}$，其中$\boldsymbol{\Lambda}$是特征值对角矩阵，$\boldsymbol{Q}$是特征向量矩阵。对于$\boldsymbol{A}^{\boldsymbol{i}}=\boldsymbol{Q}\boldsymbol{\Lambda}^{\boldsymbol{i}}\boldsymbol{Q}^{-1}$，由于$\boldsymbol{\Lambda}$是对角矩阵，$\boldsymbol{\Lambda}^{\boldsymbol{i}}$ 的计算就变得非常简单，只需对角线上的每个特征值进行 i 次幂运算。当 i较大时，矩阵 $\boldsymbol{\Lambda}^{\boldsymbol{i}}$中最大特征值的影响会主导整体计算，因为其他特征值在幂运算后会迅速减小。

如果将$\boldsymbol{p}_{\boldsymbol{i}}\in\mathbf{R}^{|V|}$定义为从节点$u$到所有其他节点的长度为$i$的路径数的向量，那么对于较大的$i$，有：$\boldsymbol{A}\boldsymbol{p}_{\boldsymbol{i}}=\lambda\boldsymbol{p}_{\boldsymbol{i-1}}$，随着$i$变大，路径数的增长将越来越接近于最大特征值的影响，$\boldsymbol{p}_{\boldsymbol{i}}$最终会收敛到图的主特征向量。基于这种对大$i$的近似，以及$i=1$ 时的精确解，令λ是$\boldsymbol{A}$的最大特征值，可以得到：

$$E[\boldsymbol{A}^i[u,v]]=\frac{d_u d_v\lambda^{i-1}}{2m}$$

3. 随机游走方法

随机游走方法(Random walk methods)也是一种全局相似度度量的方法，通过在图中模拟节点之间的随机走动，捕捉图的全局结构信息。它可以发现两个节点间的潜在联系，即使它们没有直接相连或局部领域重合。

随机游走的原理是在图中从一个节点开始，按一定的概率跳转到相邻节点的过程。这个过程会持续进行，直到达到某个终止条件（如步数限制或返回初始节点的概率）。随机游走方法可以通过节点的访问频率或在特定节点处停止的概率来衡量节点间的相似性。

常见的随机游走方法有普通随机游走、SimRank 以及 PageRank 算法。

1) 普通随机游走：

普通随机游走在每一步中会等概率地选择一个邻居节点进行移动。假设随机游走从节点 u 开始，最终停止在节点v，那么可以通过统计从u到v的随机游走频率或概率，衡量u和v之间的联系强度。

2) SimRank:

SimRank 是另一种基于随机游走的相似性度量方法，它的核心思想是：两个节点如果与相似的邻居相连，那么它们自身也应该是相似的。SimRank 通过递归计算节点间的相似性，特别适合用于捕捉全局层面的节点关系。

SimRank 的定义基于以下递归关系：

$$\text{SimRank}(u,v)=\begin{cases}1 & ,if\ u=v\\ \dfrac{C}{|N(u)|\cdot|N(v)|}\sum_{\mathrm{i}=1}^{|\mathrm{N(u)}|}\sum_{\mathrm{j}=1}^{|\mathrm{N(v)}|}\text{SimRank}(N(u)_i,N(v)_j) & ,if\ u\neq v\end{cases}$$

其中 SimRank(u,v) 是节点u和节点v之间的相似度，$N(u)$ 和$N(v)$分别表示节点u和v的邻居集合，C是一个衰减因子，用于控制相似性的递减速度，表示路径越长，相似性越低。

3) PageRank:

PageRank 通过加入节点重启的机制，使得游走者有一定的概率回到特定的起始节点。这种方法更关注于某个特定节点的相对重要性。

定义随机游走的转移概率矩阵$\boldsymbol{P}=\boldsymbol{A}\boldsymbol{D}^{-1}$，其中$\boldsymbol{A}$是图的邻接矩阵，$\boldsymbol{D}$是度数矩阵，$\boldsymbol{D}^{-1}$用于对邻接矩阵进行归一化，使得每列的和为 1，表示每个节点到其邻居的转移概率。并计算递推方程：

$$\boldsymbol{q}_u = cP\boldsymbol{q}_u + (1-c)\boldsymbol{e}_u$$

在这个方程中，$\boldsymbol{e}_u$是节点u的一位指示向量（one-hot indicator vector），这是一个长度等于图中节点数的向量，其中第u个元素为 1，其他元素为 0。它表示从节点u开始的随机游走。而$\boldsymbol{q}_u[v]$表示从节点u开始的随机游走最终访问节点v的稳定概率。这里的c项决定了随机游走在每一步的重启概率。如果没有这个重启概率，随机游走的概率将简单地收敛到特征向量中心性的归一化变体。然而，加入重启概率后，反而获得了特定于节点u的重要性度量，因为随机游走会不断地被“传送”回该节点。$(1-c)\boldsymbol{e}_u$表示随机游走者在每一步有$(1-c)$的概率从节点u启动新的游走。

上述递推方程的解为：

$$\boldsymbol{q}_u = (1-c)(\boldsymbol{I} - c\boldsymbol{P})^{-1}\boldsymbol{e}_u$$

其中$\boldsymbol{I}$为单位矩阵，$(\boldsymbol{I} - c\boldsymbol{P})^{-1}$是矩阵求逆，表示在多次迭代中如何通过转移矩阵$\boldsymbol{P}$来递推计算出稳定的随机游走概率分布。

可以定义节点间的随机游走相似性度量为：

$$\mathrm{SRW}(u,v) = \boldsymbol{q}_u[v] + \boldsymbol{q}_v[u]$$

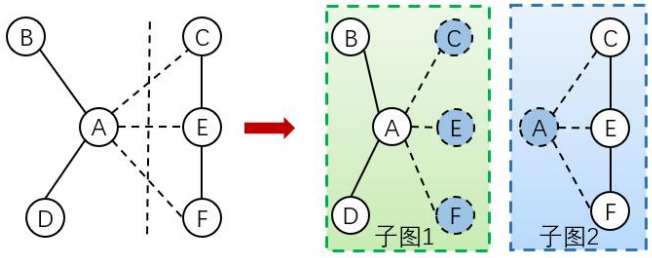
即节点对之间的相似性与从另一个节点开始的随机游走访问该节点的可能性成正比[4]。如果从节点u开始的随机游走经常到达节点v，或者反过来，从节点v开始的随机游走经常到达节点u，那么这两个节点就被认为是相似的。

2.3 图级特征

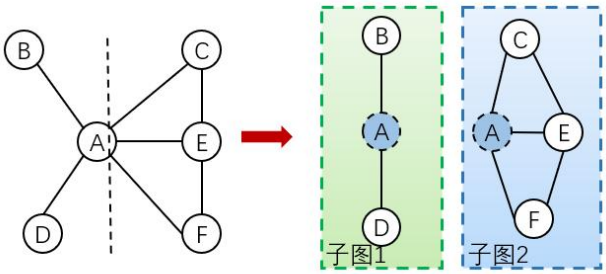
图级特征是用来描述某个子图，整个图或两个不同图或子图之间关系的一些特征。这些特征在许多图分析任务中起着核心作用，尤其是涉及图分类、聚类或图相似性计算等场景。图级特征不仅可以帮助理解图的全局结构，还能为后续的下游任务提供有力的支持。这一节将详细探讨图级特征的各个方面，包括图划分、图内部的特征、子图间的特征和不同图相似性特征。

2.3.1 图划分

图划分（Graph Partitioning）是图论中的一个重要概念，它涉及将图中的节点或边集合划分为若干个不相交的子集，是后续许多图级特征的基础。



(a) 点分割



(b) 边分割

图 2-6 点分割与边分割

按照对图数据的切分方式分类，图划分可以分为点分割（Vertex Partitioning or Edge-cut Partitioning）和边分割（Edge Partitioning or Vertex-cut Partitioning）。如图 2-6 所示，点分割是将图的节点分配到各个子图中，维持节点之间子图的完整性，但可能会造成某些节点之间的边被切割掉；同理，边分割是将图的边分配到各个子图中，每组分配的边构成子图，但可能会造成某些节点的冗余。根据不同目标，可以设计不同的指标来衡量划分算法划分的效果。

在图G中，设$A \subseteq V$表示图中节点的一个子集，而$\complement_V A$表示这个子集的补集，即$A \cup \complement_V A = V$，且$A \cap \complement_V A = \emptyset$。当图$G$的节点按点分割划分为$K$个不重叠的子集$A_1, A_2, \cdots, A_K$时，定义该划分的割值（Cut Value）为：

$$\mathrm{cut}(A_1, A_2, \cdots, A_K) = \frac{1}{2}\sum_{k=1}^{K} \mathrm{card}(\{(u,v) \in \varepsilon | u \in A_k, v \in \complement_V A_k\})$$

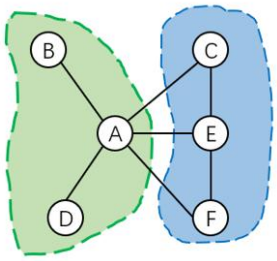


图 2-7 割值示意图

简单来说，割值表示跨越节点划分边界的边数的总和。如图 2-7 所示，当将图G节点划分为$A_1 = \{A, B, D\}, A_2 = \{C, E, F\}$时，可以计算

$$\mathrm{cut}(A_1, A_2) = \frac{1}{2}(3+3) = 3$$

一种定义最优划分的方法是选择一个能够最小化切割值的划分。然而，尽管有一些高效的算法可以解决这个问题，但这种方法存在一个问题，即它倾向于形成由单个节点组成的簇[7]。总的来说，图划分中的“割值”概念是评估划分效果的一个重要指标，而最优图划分则是在满足特定条件下最小化割值的一种划分。

2.3.2 图内部的特征

图内部特征，是对图整体特性的一种量化分析，它不仅可以帮助深入理解图的整体连接性，还能揭示图中节点和边的分布特征。通过这些特征可以判断图中节点的紧密程度、图中子结构的复杂度等。这些内部特征的提取对于后续的图分类、社区检测等任务具有重要意义。这一小节将介绍一些常用的图内部特征，包括平均度、连通度、内部密度等。

1. 空间特征

空间特征在图分析中占据重要地位，尤其是当试图分析图的几何形态时，这些特征能够提供直观的解释。通过对图中节点间距离、节点之间边的密度等几何信息的分析，通过空间特征可以更好地理解图的结构，发现图中的簇状结构和潜在的分层关系。这一部分将探讨一些重要的空间特征及其在实际应用中的意义。

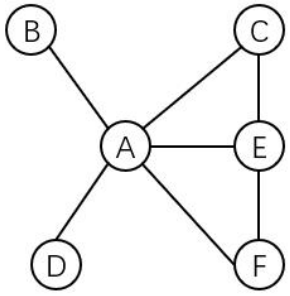


图 2-8 空间特征例 1

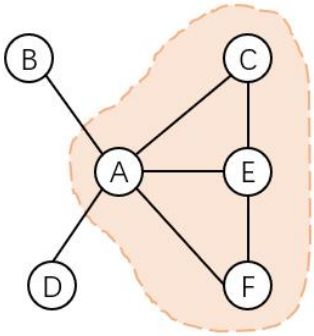


图 2-9 空间特征例 2

1) 内部边数

内部边数（Edge Inside）[9]直接表示图G内部的边的数量。

$$f(G) = \mathrm{card}(E_{in}^{G})$$

这个指标简单但有效，用于评估社区内节点之间的直接连通性。例如，可计算图 2-8 空间特征例 1 的内部边数为：$f(G) = 7$。

2) 内部密度

内部密度（Internal Density）[9]是衡量图G内部节点之间连边的密集程度，定义如下：

$$f(G) = \frac{\mathrm{card}(E_{in}^{G})}{N(N-1)/2}$$

公式中，$N(N-1)/2$ 是图 G 中可能形成的最大边数。因此，内部密度反映了图实际存在的边数相对于最大可能存在边数的比例。例如，可计算图 2-8 空间特征例 1 的内部密度为：$f(G) = \frac{\mathrm{card}(E_{in}^{G})}{N(N-1)/2} = \frac{7}{6(6-1)/2} \approx 0.47$。

3) 平均度

平均度（Average Degree）[9]是一个表达网络整体性质重要的参数。平均度的计算为：

$$f(G) = \frac{1}{N}\sum_{i=1}^{N} d_i = \frac{2\mathrm{card}(E_{in}^{G})}{N}$$

其中d_i是每个节点的度，$\mathrm{card}(E_{in}^{G})$是图$G$内部的边的数量，$N$是节点数。例如，可计算图 2-8 空间特征例 1 的平均度为：$f(G) = \frac{1}{N}\sum_{i=1}^{N} d_i = \frac{2\mathrm{card}(E_{in}^{G})}{N} = \frac{2\times 7}{6} \approx 2.33$。

度分布$P(d)$表示随机选择的节点的度为d的概率，则平均度的另一种表示方法为：

$$f(G) = \sum_{d=0}^{\infty} d\,P(d)$$

4) 超过中位数的度比例

超过中位数的比例（Fraction over Median Degree，FOMD）[9]衡量的是图中的一个子图G中节点的内部度数超过全图节点度数中位数的节点比例，计算方式如下：

$$f(G) = \frac{\mathrm{card}(\{u: u \in G, |\{(u,v): v \in G, (u,v) \in E\}| > d_m\})}{N}$$

其中，d_m表示整个图中节点度数的中位数。这一指标用于评估图内节点的连通性是否相对较强。

例如，可计算图 2-9 空间特征例 2 的 FOMD 为：

$$d_m = Q_{\frac{1}{2}}(\{1,1,2,2,3,5\}) = \frac{1}{2}(2+2) = 2$$

$$f(G) = \frac{\mathrm{card}(\{u: u \in G, |\{(u,v): v \in G, (u,v) \in E\}| > d_m\})}{\mathrm{card}(G)} = \frac{\mathrm{card}(\{A,E\})}{\mathrm{card}(\{A,C,E,F\})} = 0.5$$

5) 三角参与率

三角参与率（Triangle Participation Ratio，TPR）[9]是衡量图G中有多少比例的节点构成三角形（即与图内两个其他节点形成三角形）。计算方式如下：

$$f(G)=\frac{\mathrm{card}(\{u|u\in G,\{v,w\in G,(u,v)\in E,(u,w)\in E,(v,w)\in E\}\neq\emptyset\})}{N}$$

这个指标反映了图内节点之间的紧密连接程度。在社区检测领域，三角形是社团结构的基本单位，因此具有较高的 TPR 通常意味着该社区的联系更加紧密。例如，可计算图 2-9 空间特征例 2 的 TPR 为：

$$f(G)=\frac{\mathrm{card}(\{A,C,E,F\})}{\mathrm{card}(\{A,C,E,F\})}=1$$

6) 点连通度

在无向图G中，如果G包含一条从u到v的路径，则两个顶点u和v被称为是连通的。否则，它们被称为是非连通的。如果图中的每一对顶点都是连通的，则称G是连通图。

对于连通图$G=(V,E)$，如果存在一个顶点子集$A\subseteq V$使得G去除节点集A后不是连通图，则 A 是图 G 的一个点割集。大小为 1 的点割集又被称作割点。对于连通图G和整数k，若$\mathrm{card}(V)\geq k+1$且G不存在大小为$k-1$的点割集，则称图G是k-点连通的，而使得上式成立的最大的k被称作图G的点连通度（Vertex Connectivity），记作$k(G)$。

7) 边连通度

对于连通图$G=(V,E)$，若$F\subseteq E$且$G^{'}=(V,E\setminus F)$不是连通图，则F是图G的一个边割集。大小为 1 的边割集又被称作桥。对于连通图G和整数k，若G不存在大小为$k-1$的边割集，则称图G是k-边连通的，而使得上式成立的最大的k被称作图G的边连通度，记作$\lambda(G)$。

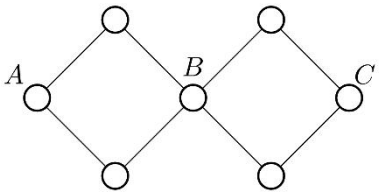


图 2-10 连通度示意图

例如，可以计算图 2-10 点连通度为 1，边连通度为 2。并且可以证明，一个图的点连通度总是小于其边连通度。

2. 谱特征

谱图理论通过分析图的拉普拉斯矩阵的特征值和特征向量来研究图的性质。本节将介绍图的拉普拉斯矩阵，并讨论其关键性质、特征值和特征向量，以及如何利用谱图理论构造图的谱特征。

1) 拉普拉斯矩阵

拉普拉斯矩阵（Laplacian Matrix）[18]是一种除邻接矩阵之外的另一种图矩阵表示。对于一个给定的图$G=(V,E)$，$\boldsymbol{A}$是该图的邻接矩阵，它的拉普拉斯矩阵被定义为如下形式：

$$\boldsymbol{L}=\boldsymbol{D}-\boldsymbol{A}$$

其中$\boldsymbol{D}=diag(d(v_1),d(v_2),\cdots,d(v_N))$是对角度矩阵。

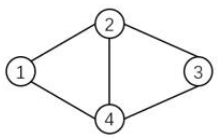
图	度矩阵	邻接矩阵	拉普拉斯矩阵
	$\begin{pmatrix}2&0&0&0\\0&3&0&0\\0&0&2&0\\0&0&0&3\end{pmatrix}$	$\begin{pmatrix}2&-1&-1&0\\-1&3&-1&-1\\-1&-1&2&-1\\0&-1&-1&3\end{pmatrix}$	$\begin{pmatrix}1&-1/\sqrt{6}&-1/\sqrt{6}&0\\-1/\sqrt{6}&1&-1/3&-1/2\sqrt{3}\\-1/\sqrt{6}&-1/3&1&-1/2\sqrt{3}\\0&-1/2\sqrt{3}&-1/2\sqrt{3}&1\end{pmatrix}$

图 2-11 图的度矩阵、邻接矩阵与拉普拉斯矩阵

标准化拉普拉斯矩阵的定义是上式的归一化版本，即

$$\boldsymbol{L}=\boldsymbol{D}^{-\frac{1}{2}}(\boldsymbol{D}-\boldsymbol{A})\boldsymbol{D}^{-\frac{1}{2}}=\boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}.$$

例如，可计算图 2-11 中的拉普拉斯矩阵。还存在一种形式的拉普拉斯矩阵，叫做随机游走拉普拉斯矩阵，在此不详细介绍，三种形式的拉普拉斯矩阵可整理成下表：

表 2-1 三种不同的拉普拉斯矩阵

方式	定义	适用场景
原始拉普拉斯[18]	$\boldsymbol{L}=\boldsymbol{D}-\boldsymbol{A}$	适用于基本的图聚类任务
标准化拉普拉斯[17]	$\boldsymbol{L}=\boldsymbol{I}-\boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$	适用于处理节点度数差异较大的图的切分问题
随机游走拉普拉斯[17]	$\boldsymbol{L}=\boldsymbol{I}-\boldsymbol{D}^{-1}\boldsymbol{A}$	适用于希望通过模拟随机游走过程来获取图的结构信息的聚类任务

注意，拉普拉斯矩阵是对称的，因为度矩阵$\boldsymbol{D}$和邻接矩阵$\boldsymbol{A}$均是对称的。设$\boldsymbol{f}$是一个向量，其第i个元素$\boldsymbol{f}_i$与节点v_i相关。将$\boldsymbol{L}$与$\boldsymbol{f}$相乘会得到一个新的向量$\boldsymbol{h}$：

$$\begin{aligned}\boldsymbol{h}&=\boldsymbol{L}\boldsymbol{f}\\&=(\boldsymbol{D}-\boldsymbol{A})\boldsymbol{f}\\&=\boldsymbol{D}\boldsymbol{f}-\boldsymbol{A}\boldsymbol{f}\end{aligned}$$

$\boldsymbol{h}$向量的第i个元素可以被表示为：

$$\begin{aligned}\boldsymbol{h}_i&=d(v_i)\boldsymbol{f}_i-\sum_{j=1}^{N}\boldsymbol{A}_{i,j}\boldsymbol{f}_i\\&=d(v_i)\boldsymbol{f}_i-\sum_{v_j\in N(v_i)}\boldsymbol{A}_{i,j}\boldsymbol{f}_i\\&=\sum_{v_j\in N(v_i)}(\boldsymbol{f}_i-\boldsymbol{f}_j)\end{aligned}$$

可以观察到，$\boldsymbol{h}_i$是节点v_i与其邻居$N(v_i)$在$\boldsymbol{f}$上的差值的总和。接下来计算$\boldsymbol{f}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{f}$：

$$\begin{aligned}\boldsymbol{f}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{f}&=\sum_{v_i\in V}\boldsymbol{f}_i\sum_{v_j\in N(v_i)}(\boldsymbol{f}_i-\boldsymbol{f}_j)\\&=\sum_{v_i\in V}\sum_{v_j\in N(v_i)}(\boldsymbol{f}_i\boldsymbol{f}_i-\boldsymbol{f}_i\boldsymbol{f}_j)\\&=\sum_{v_i\in V}\sum_{v_j\in N(v_i)}\left(\frac{1}{2}\boldsymbol{f}_i\boldsymbol{f}_i-\boldsymbol{f}_i\boldsymbol{f}_j+\frac{1}{2}\boldsymbol{f}_j\boldsymbol{f}_j\right)\\&=\frac{1}{2}\sum_{v_i\in V}\sum_{v_j\in N(v_i)}(\boldsymbol{f}_i-\boldsymbol{f}_j)^2\end{aligned}$$

因此，$\boldsymbol{f}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{f}$是相邻节点之间差值的平方和的一半。

换句话说，它衡量的是相邻节点的值有多大差异。很容易验证，对于任何可能的实向量$\boldsymbol{f}$，$\boldsymbol{f}^{\mathrm{T}}\boldsymbol{L}\boldsymbol{f}$总是非负的，这表明拉普拉斯矩阵是半正定的。

对于标准化的拉普拉斯，也具有类似的性质。假设$\boldsymbol{W}$是未经标准化的邻接矩阵，$\boldsymbol{A}$是标准化后的邻接矩阵$\boldsymbol{A}=\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$，$d_i:=d(v_i)$，则

$$\boldsymbol{A}_{ii}=0$$

$$\boldsymbol{A}_{ij}=1\Big/\sqrt{d_i d_j},(\mathrm{i},j)\in E$$

$$\boldsymbol{A}_{ij}=0,(\mathrm{i},j)\notin E\text{且}(\boldsymbol{A}\boldsymbol{v})_i=\frac{1}{\sqrt{d_i}}\sum_{j\in\{j|(i,j)\in\boldsymbol{E}\}}\frac{1}{\sqrt{d_j}}v_j$$

对比以下两式：

$$(\boldsymbol{Av})_i = \frac{1}{\sqrt{d_i}} \sum_{j\in\{j|(i,j)\in \boldsymbol{E}\}} \frac{1}{\sqrt{d_j}} v_j \qquad (\boldsymbol{Lv})_i = v_i - \frac{1}{\sqrt{d_i}} \sum_{j\in\{j|(i,j)\in \boldsymbol{E}\}} \frac{1}{\sqrt{d_j}} v_j$$

显然可以看出，$(\boldsymbol{Lv})_i$得到的是第i个节点相对于其邻居的变化量。

2) 特征值与特征向量

图G的标准化拉普拉斯矩阵$\boldsymbol{L}$是一个实对称矩阵，因此可以被正交对角化：

$$\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathrm{T}} = \boldsymbol{U}\begin{pmatrix}\lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n\end{pmatrix}\boldsymbol{U}^{\mathrm{T}}$$

其中$\boldsymbol{U} = (u_1, u_2, \cdots, u_n)\epsilon \mathbf{R}^{n\times n}$，且满足：

(1) 标准化拉普拉斯矩阵是半正定矩阵，它的特征值都是非负的。

假设λ是拉普拉斯矩阵$\boldsymbol{L}$的特征值，$\boldsymbol{u}$是对应的标准特征向量。根据特征值和特征向量的定义，有$\lambda\boldsymbol{u} = \boldsymbol{Lu}$。注意，$\boldsymbol{u}$是一个单位非零向量，即$\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u} = 1$。然后有，

$$\lambda = \lambda\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u} = \boldsymbol{u}^{\mathrm{T}}\lambda\boldsymbol{u} = \boldsymbol{u}^{\mathrm{T}}\boldsymbol{Lu} \geq 0$$

(2) 对于一个含有N个节点的图G，拉普拉斯矩阵一共有N个特征值和特征向量。并且，总是存在一个特征值等于 0。令$\boldsymbol{u_1} = \frac{1}{\sqrt{N}}(1,1,\cdots,1)$，显然$\boldsymbol{Lu_1} = 0 = 0\boldsymbol{u_1}$，即$\boldsymbol{u_1}$是特征值 *0* 的特征向量。

(3) 特征向量之间满足$\boldsymbol{u_i}\boldsymbol{u_i}^{\mathrm{T}} = 1$ 且$\boldsymbol{u_i}\boldsymbol{u_j}^{\mathrm{T}} = 0$。

(4) $\boldsymbol{L}$的特征值描述了其对应单位特征向量在图上变化量的强度。由上文，知道$\boldsymbol{f}^{\mathrm{T}}\boldsymbol{Lf}$度量了信号$\boldsymbol{f}$在图上变化的平方和。那么令$\boldsymbol{f} = \boldsymbol{u_i}$，有

$$\boldsymbol{u_i}^{\mathrm{T}}\boldsymbol{Lu_i} = \boldsymbol{u_i}^{\mathrm{T}}\lambda_i\boldsymbol{u_i} = \lambda_i$$

即λ_i的值越大，表明其对应单位特征向量所对应的图信号在图上的变化越剧烈。故λ_i也称对应特征向量的平滑度。

图 2-12 是不同特征值对应的特征向量的实例，可以看到特征值越大，其对应的特征向量越不平滑：

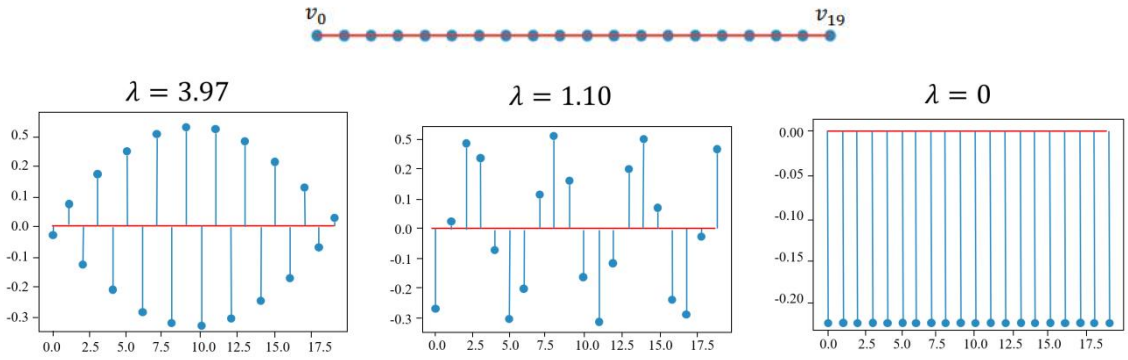


图 2-12 不同特征值对应的特征向量

图的连通分量可以写作一个指示向量，其对应连通分量的顶点位置上为 1，而其他位置为 0。例如，如果图G中有两个连通分量，那么对于第一个连通分量，所有属于该连通分量的顶点在指示向量中对应的位置为 1，其他位置为 0；对于第二个连通分量，同样有一个类似的指示向量。拉普拉斯矩阵还有一个重要的性质，与谱聚类相关，这里不加证明的给出。令G是一个无向有权图，则原始拉普拉斯矩阵L的特征值 0 的几何重数k等于图G中连通分量的数量。

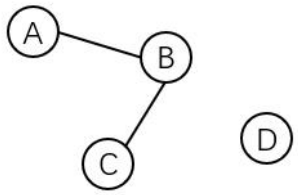


图 2-13 指示向量示意图

例如，可以写出图 2-13 的原始拉普拉斯矩阵为：

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A} = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

容易计算出特征值 0 对应的特征向量分别是：$\boldsymbol{v_1} = (1,1,1,0), \boldsymbol{v_2} = (0,0,0,1)$，其几何重数 $k = 2$，而图中连通分量的个数也是 2。

这个特征值所在的特征空间是由这些连通分量的指示向量生成的。换句话说，这些特征值为 0 对应的特征向量实际上是各个连通分量的指示向量。

除了上述应用外，也可以利用分解后得到的特征向量进行聚类，这称为谱聚类。如 NJW 算法[20]是谱聚类的一种方法，该算法使用最大的k个特征向量进行图的划分。

2.3.3 子图间的特征

社区结构是网络研究中的一个重要概念，尤其在数据挖掘和社交网络分析领域受到广泛关注。尽管社区的定义并不十分明确，但一般认为社区结构是将网络中的节点划分为若干个集合，其中每个集合内的节点之间连接密集，而集合之间的连接稀疏，如图 2-14 所示。

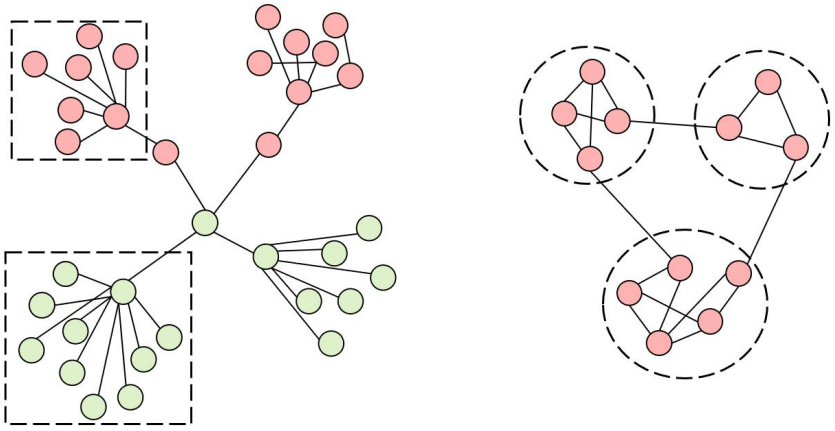


图 2-14 图的社区结构

社区分析通常包括两个阶段：首先，从网络中检测出有意义的社区结构；其次，评估所检测到的社区结构的合理性。由于没有一个被普遍接受的社区定义，社区结构的定义也不尽相同，每种定义都通过不同的指标来证明其合理性。

子图间特征分析是社区检测中的重要步骤，尤其在复杂网络和社交网络的研究中，这一分析能够识别网络中的紧密连接群体。通过对比子图之间的连接强度、扩展系数等指标可以评估子图之间的关系，判断社区结构的合理性。这一节将介绍如何通过子图间特征分析来识别和评估图中的社区结构。

1.基于连通性

1) 比值割

比值割（Ratio Cut）[12]的定义如下：

$$\mathrm{RatioCut}(A_1,\cdots,A_K)=\frac{1}{2}\sum_{k=1}^{K}\frac{\mathrm{card}(\{(u,v)\in E|u\in A_k,v\in \complement_V A_k\})}{\mathrm{card}(A_k)}$$

其中，$A_1,\cdots,A_K$表示将图节点划分为K个子集，$\mathrm{card}(A_k)$表示第k个子集中的节点数，E表示图的边集合，(u,v)是一条边。通过最小化比值割，既可以最小化割集中的边数，又可以确保每个子集A_k的大小相对较大，从而避免划分后产生过小的子集。这种方法在图划分和社区检测中非常常用。

2) 扩展系数

扩展系数（Expansion）[9]是衡量，在同一图G下，子图ω中每个节点指向子图ω外部的边的平均数量，计算公式如下：

$$f(\omega)=\frac{\mathrm{card}(E_\omega^{out})}{\mathrm{card}(\omega)}$$

其中，$\mathrm{card}(E_\omega^{out})$表示从子图$\omega$指向外部节点的边的数量，$\mathrm{card}(\omega)$表示子图的节点数量。在社区检测领域，扩展系数表示社区内部节点在外部的连接强度，即每个节点平均有多少条边连接到社区外部。

3) 割比率

割比率（Cut Ratio）[13]0是衡量从子图中离开的边占所有可能的边的比例，计算公式如下：

$$f(\omega)=\frac{\mathrm{card}(E_\omega^{out})}{\mathrm{card}(\omega)\cdot\mathrm{card}(G\setminus\omega)}$$

其中，$\mathrm{card}(G\setminus\omega)$表示全图中不属于子图$\omega$的节点数量。在社区检测领域，割比率反映了社区边缘节点有多强的倾向连接到社区外部，越高的割比率表示社区内外的分界越模糊。

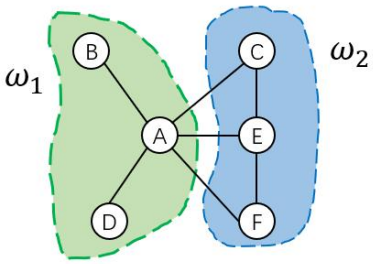


图 2-15 割比例示意图

例如，可分别基于连通性，计算图 2-15 的各项指标：

- 比割值：$RatioCut(A_1,A_2)=\frac{1}{2}\sum_{k=1}^{2}\frac{\mathrm{card}(\{(u,v)\in E:u\in A_k,v\in \complement_V A_k\})}{\mathrm{card}(A_k)}=\frac{1}{2}\left(\frac{3}{3}+\frac{3}{3}\right)=1$；
- 扩展系数：$f(\omega_1)=\frac{\mathrm{card}(E_\omega^{out})}{\mathrm{card}(\omega)}=\frac{3}{3}=1$；
- 割比率：$f(\omega_1)=\frac{\mathrm{card}(E_\omega^{out})}{\mathrm{card}(\omega)\cdot\mathrm{card}(G\setminus\omega)}=\frac{3}{3\cdot 3}\approx 0.33$。

2.基于网络模型

网络模型是用于描述和理解网络结构及动态特性的理论模型。通过这些模型可以更加深入理解和模拟现实世界中的各种网络，如社交网络、互联网、生物网络等。比如在一个随机图模型中，每对节点之间都以相同的概率连接来生成网络。基于网络模型的特征通过对比给定的图与一个网络模型生成的图的差距，给出特征的值。

1) 模块度

模块度（Modularity）是一种衡量网络中社区结构质量的指标。要计算一个网络的模块度，

需要构造一个具有相同节点度分布的随机网络作为参照。通俗地来说，模块度的物理含义是：在社团内，实际的边数与随机情况下的边数的差距。如果差距比较大，说明社团内部密集程度显著高于随机情况，社团划分的质量较好。

无权无向图的模块度定义为：

$$Q_{ud}=\sum_{\omega\in\Omega}\left[\frac{\mathrm{card}(E_\omega^{in})}{\mathrm{card}(E)}-\left(\frac{2\mathrm{card}(E_\omega^{in})+\mathrm{card}(E_\omega^{out})}{2\mathrm{card}(E)}\right)^2\right]$$

其中，$\mathrm{card}(E_\omega^{in})$表示子图$\omega$内部的边数，$c\mathrm{ard}(E_\omega^{out})$表示从子图$\omega$指向外部的边数，$\mathrm{card}(E)$表示图的总边数。$\frac{\mathrm{card}(E_\omega^{in})}{\mathrm{card}(E)}$表示社区内部的边占总边数的比例。$\frac{2\mathrm{card}(E_\omega^{in})+\mathrm{card}(E_\omega^{out})}{2\mathrm{card}(E)}$表示社区内的节点度数和与图中所有节点度数和的比值。

另一种定义方式为[16]：

$$Q_{ud}=\frac{1}{2\mathrm{card}(E)}\sum_{i,j}\left(\boldsymbol{A}_{ij}-\frac{d(i)d(j)}{2\mathrm{card}(E)}\right)\delta_{\omega_i,\omega_j}$$

其中，$\boldsymbol{A}_{ij}$是邻接矩阵的元素，如果节点i和j之间有边则为 1，否则为 0；$d(i)$和$d(j)$ 分别表示节点i和节点j的度数。事实上，如果把同一个图中的边随机放置，则节点i和节点j之间边数的期望值是$\frac{d(i)d(j)}{2\mathrm{card}(E)}$，请读者自行证明。$\delta_{\omega_i,\omega_j}$是 Kronecker delta 函数，当$i$和$j$属于同一社区时返回 1，否则为 0。$\frac{1}{2\mathrm{card}(E)}\sum_{i,j}\boldsymbol{A}_{ij}$表示子图内部实际的边数的比例，$\frac{1}{2\mathrm{card}(E)}\sum_{i,j}\frac{d(i)d(j)}{2\mathrm{card}(E)}$表示随机情况下社区内部期望的边数的比例，因此，模块度的定义可以看作是，在社区内部的边的比例，减去边随机放置时社区内部期望边数的比例。模块度值越高，表示图中的子图结构越明显，即子图内部的节点连接紧密，子图之间的连接稀疏。

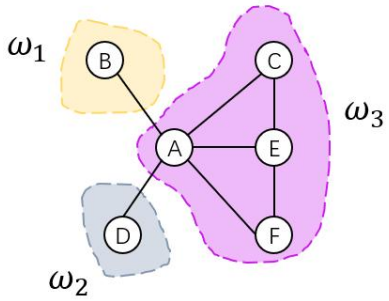


图 2-16 模块度示意图

例如，可以使用第一种定义计算图 2-16 的模块度：

$$\begin{aligned}Q_{ud}&=\sum_{\omega\in\Omega}\left[\frac{\mathrm{card}(E_\omega^{in})}{\mathrm{card}(E)}-\left(\frac{2\mathrm{card}(E_\omega^{in})+\mathrm{card}(E_\omega^{out})}{2\mathrm{card}(E)}\right)^2\right]\\&=\left[0-\left(\frac{1}{2\times7}\right)^2\right]+\left[0-\left(\frac{1}{2\times7}\right)^2\right]+\left[\frac{5}{7}-\left(\frac{10+2}{2\times7}\right)^2\right]\\&=-\frac{3}{98}\approx-0.03\end{aligned}$$

2.3.4 不同图间相似性特征

在图数据分析中，不同图的相似性度量是一个关键问题，特别是在图分类、图匹配等任务中，如何有效地衡量图与图之间的相似性，直接影响到模型的性能。本节将介绍几种常用的图

相似性度量方法，尤其是图核方法。图核方法不仅能够捕捉图的结构信息，还能够通过核函数将图特征映射到高维隐特征空间，从而更精确地进行相似性分析。通过这些方法可以更加深入地理解图的全局特性和相互关系。

1. 图核方法概述

核方法（Kernel methods）是一种广泛使用的基于图水平的特征表示方法。虽然数学上核方法的核心是不需显式设计全图的向量特征，而是设计一个核函数$K(G,G^{'})\in\mathbf{R}$去计算两个图的相似度；然而，基于特征工程的图机器学习经常使用的图核方法确实显示给出了每个图的向量特征，并且直接用点乘去计算二者的相似度，详细的过程可参见后文具体核方法的讲解。

可以使用图词袋法（Bag-of-Words，BoW）获得图的全局特征。例如，可以根据图中节点的度数、中心性或聚类系数来计算直方图以用作图级表示。但这些方法的缺点是这些节点信息大多是一些局部信息，可能会错过图中重要的全局信息。

2. 图元核

图元核方法的基本思想是对图网络中的各种图元（Graphlet）进行计数从而得到图的向量表示，进而利用该向量表示计算内积来衡量图之间的相似度。

图元是在固定数量节点下构成的任意图结构。例如，图 2-17 展现了在一个图中可能存在的四种不同的由三个节点构成的图元。注意与前文的节点级特征中的图元不同，这里的图元没有一个特定的根节点，因此相同节点个数下图元的种类也较少一些。

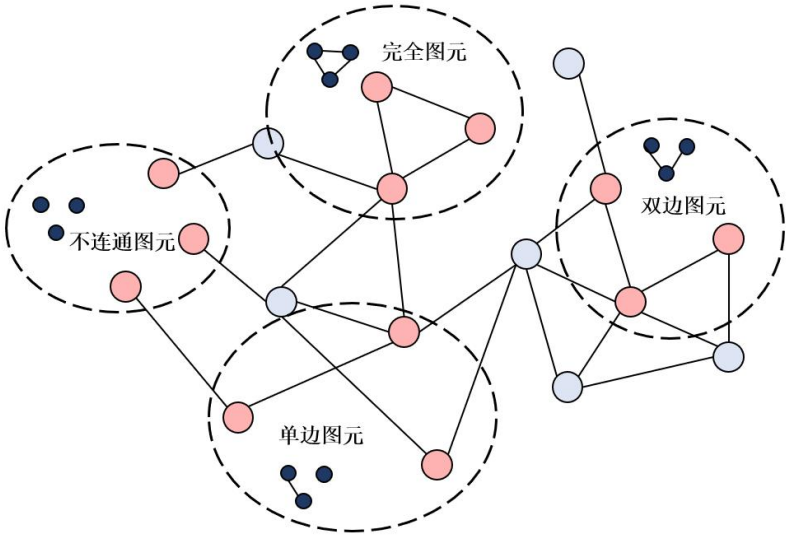


图 2-17 图元示意图

假设使用符号g_i表示第i种图元，含有k个节点的n_k种图元构成的列表记作$G_k=(g_1,g_2,\cdots,g_{n_k})$，当给定一个图$G$时，可以定义图元统计向量为$\boldsymbol{f_G}\in\mathbf{R}^{n_k}$，其中$(\boldsymbol{f_G})_i=\#(g_i\subseteq G)$，$i=1,2,\cdots,n_k$，即图$G$中该种图元的个数。图 2-18 对$k=3$情况下某个图$G$的图元向量生成方法进行了示例说明，在该图中，$\boldsymbol{f_G}=(1,6,3,0)$。

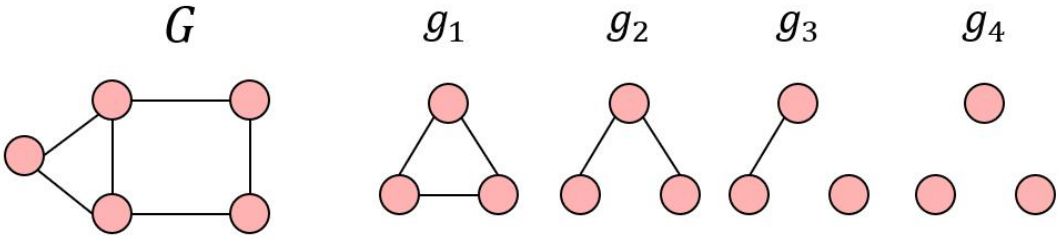


图 2-18 图元计数示意图

给定两个图G和G'，图元核计算方法为：$K(G,G')=\boldsymbol{f}_G^{\mathrm{T}}\boldsymbol{f}_{G'}$，即两个图的图元统计向量的内积。有的时候两个图规模不一致可能会导致图核值偏斜程度严重，因此在计算图元核之前可以先对图元统计向量进行归一化操作：$\boldsymbol{h}_G=\frac{\boldsymbol{f}_G}{Sum(\boldsymbol{f}_G)}$。

但是计算 Graphlet 的开销非常大。在一个大小为n的图上，通过枚举法计算k个节点图元的数量需要耗费$O(n^k)$的时间。

3. Weisfeiler-Lehman 核方法

Weisfeiler-Lehman (WL) 核方法是一种通过迭代的邻域聚合策略来改进基本的词袋方法的一种图核算法。它的核心思想是提取比单一节点的局部邻域图包含更多信息的节点级特征，并将这些更丰富的特征聚合成图级表示。WL 核方法的步骤如下：

a. 初始标签赋值：首先，给每个节点分配一个初始标签$l^{(0)}(v)$。比如，这个初始标签可以设置为节点的度数，即$l^{(0)}(v)=d_v$，其中d_v是节点v的度数，$v\in V$代表所有节点；

b. 标签更新：接下来，通过对节点邻域内的当前标签集进行哈希处理，迭代地为每个节点分配一个新标签。新标签的计算公式为：

$$\mathrm{l}^{(\mathrm{i})}(\mathrm{v})=HASH(\{\{\mathrm{l}^{(\mathrm{i}-1)}(\mathrm{u})|\forall u\in N(v)\}\})$$

其中，双花括号表示多重集（Multi-Set），HASH 函数则将每个独特的多重集映射到一个唯一的新标签。

c. 特征表示和核计算：在运行K次重新标签迭代（即步骤 b）后，每个节点都有了一个新标签$l^{(K)}(v)$，这个标签总结了节点v的K邻域结构。然后，可以计算这些标签在图上的直方图或其他统计特征，用作图的特征表示。最后，通过测量两幅图每次迭代各标签频数的差异来计算 WL 核。

Weisfeiler-Lehman 核方法相比于图元核方法拥有更高的运行效率，总体时间复杂度为$O(|E|)$，因此在实际应用中也更加广泛。也有书将赋予和更新的节点标签称为颜色，因为该核方法采用了一个名为“Color Refinement”的算法，下面基于这种叫法给出 WL 核方法的具体例子：

首先，如图 2-19 给定两个图，为每个节点指定一个初始颜色，这里颜色使用数字作为替代。

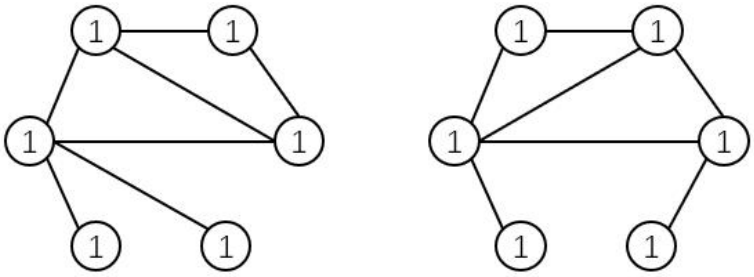


图 2-19 节点颜色初始化

接下来，为每个节点聚合邻居节点的颜色信息，以第一个图左上角的节点为例，它有三个邻居节点，因此如图 2-20 聚合后的信息变成了(1, 111)：

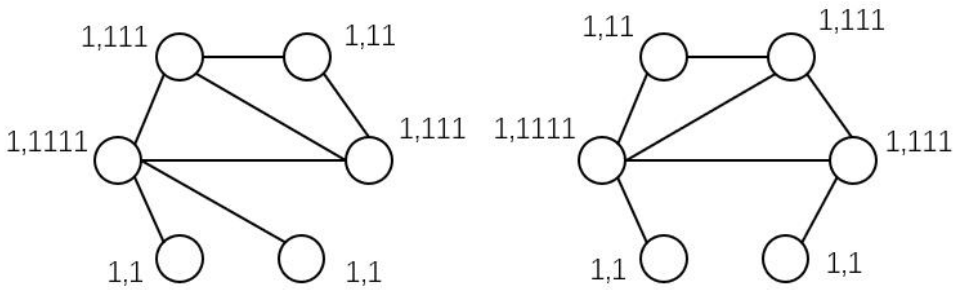


图 2-20 聚合邻居颜色

根据 HASH 表映射每个节点聚合后的颜色，仍然以第一个图左上角节点为观察对象，经过 HASH 映射，如图 2-21，它由(1, 111)映射成了对应的颜色 4：

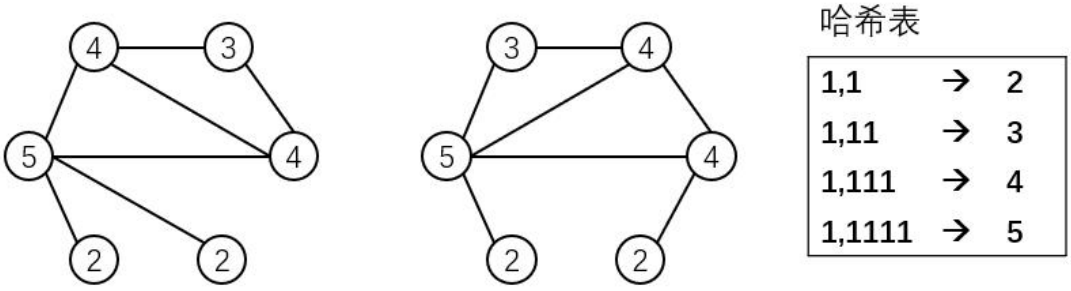


图 2-21 颜色哈希

假设经过 2 轮迭代完成了 Color Refinement 过程，如图 2-22 和 2-23：

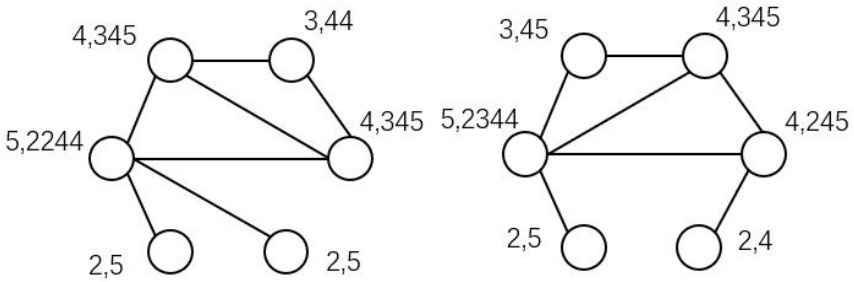


图 2-22 第二轮颜色聚合

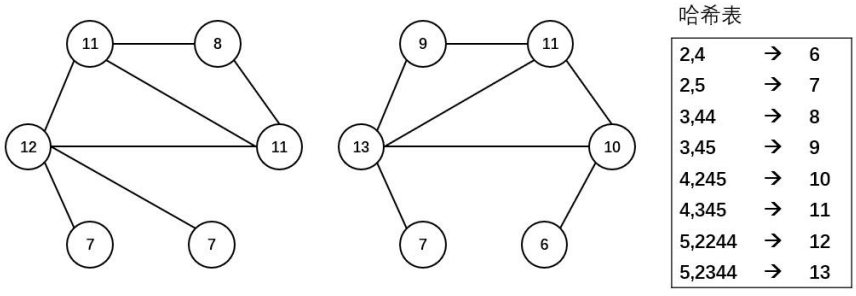


图 2-23 第二轮颜色哈希

WL 核此时对 Color Refinement 过程中每种颜色对应节点的数量进行计数统计从而得到图的向量特征表示，如图 2-24:

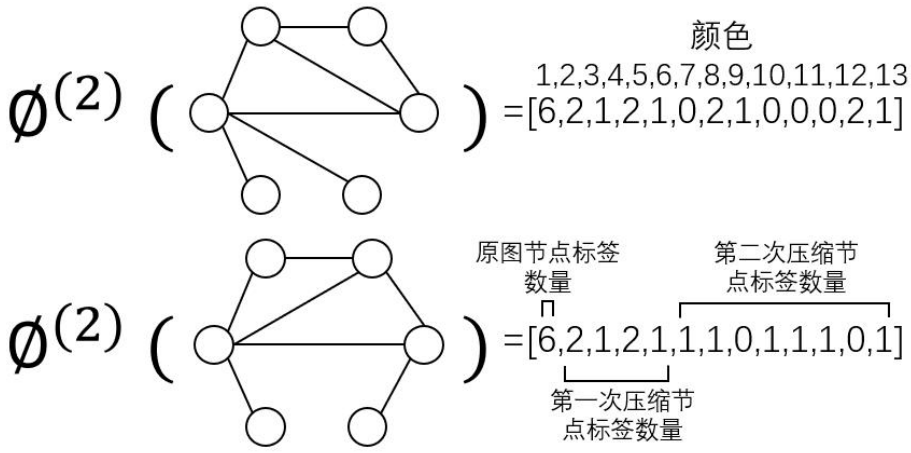


图 2-24 计算图的向量特征表示

完成如上所有步骤后，WL 核的值即可通过颜色统计向量的内积计算得到，在上图例子中：

$$K_{WL}^{(2)}(G,G^{'})=\emptyset^{(2)}(G)^{\mathrm{T}}\emptyset^{(2)}(G^{'})=49$$

2.4 本章小结

本章主要介绍了图机器学习中特征工程的基本概念，分别对图的节点级特征、边级特征和图级特征的概念和计算方法，以及它们在图分析任务中的应用进行了详细介绍，为读者学习后续关于图机器学习的章节内容奠定了基础。

本章引言介绍了特征及其在机器学习中的重要性，强调特征工程在数据准备中的核心作用及其对模型性能的提升。第一节讨论了节点级特征，包括中心性、局部聚类系数和图元度向量，它们从不同角度刻画了一个节点的性质。第二节介绍边级特征，包括基于距离的特征、局部邻域重合及全局邻域重合，这些特征刻画了一个节点对之间在图中有多强的联系。第三节讲述图级特征，涵盖图划分、图内部特征、子图间特征以及图核方法，这些特征得以从整体上对图的各个方面进行定量的描述。遇到实际问题时，需要根据问题的需求灵活选择这些特征，以求对具体的图数据有着更深刻的了解，并让机器学习方法发挥出更好的性能。

扩展阅读材料

(1) Ma Y, Tang J. Deep learning on graphs[M]. Cambridge University Press, 2021:17-30.

(2) Hamilton W L. Graph representation learning[M]. Morgan & Claypool Publishers, 2020:9-27.

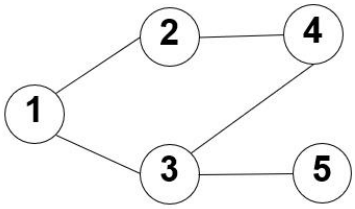
(3) Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs[J]. Advances in neural information processing systems. 2017(30).

(4) Chakraborty T, Dalmia A, Mukherjee A, et al. Metrics for community analysis: A survey[J]. ACM Computing Surveys (CSUR), 2017, 50(4): 1-37.

(5) Shervashidze N, Schweitzer P, Van Leeuwen E J, et al. Weisfeiler-lehman graph kernels[J]. Journal of Machine Learning Research, 2011, 12(9).

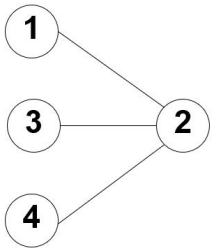
习题

(1) 考虑一个无向图，包含 5 个节点，节点间的连接关系如下：



计算每个节点的局部聚类系数，并分析图中节点的社群性。

(2) 给定一个无向图，包含 4 个节点，节点间的连接关系如下：



计算以下边级特征：

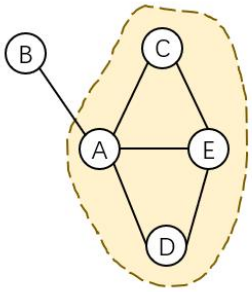
①节点 1 和节点 3 之间的最短路径长度。

②节点 1 和节点 3 的共同邻居数量。

③使用 Jaccard 系数计算节点 1 和节点 3 的局部邻域重合度。

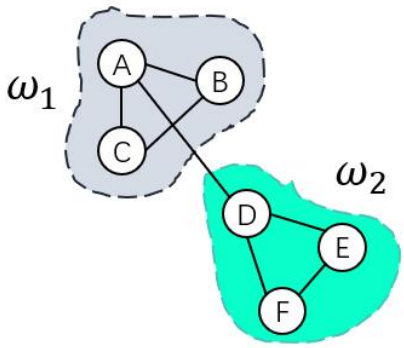
④使用 Adamic-Adar 指数计算节点 1 和节点 3 的局部邻域重合度。

(3) 给定一个无向图，包含 5 个节点，节点间的连接关系如下：



计算一下上面黄色区域子图的超过中位数的比例和三角参与率。

(4) 给定一个无向图，包含 6 个节点，节点间的连接关系如下：



如图所示，将上述图分为ω_1和ω_2两个子图。请在上述划分下，计算其比割值、拓展系数和割比率。

(5) 请使用模块度的第二种定义计算图 2-16。

(6) 请证明：图G中节点 i 和节点 j 之间节点边数的期望值为$\frac{d(i)d(j)}{2\mathrm{card}(E)}$。

(7) 请证明：模块度的第一种定义等价于第二种定义。

(8) 请绘制出含有 4 个节点的全部图元。

参考文献

[1] Dong G, Liu H. Feature engineering for machine learning and data analytics[M]. CRC press, 2018.

[2] Nargesian F, Samulowitz H, Khurana U, et al. Learning Feature Engineering for Classification[C]//Ijcai. 2017, 17: 2529-2535.

[3] Ma Y, Tang J. Deep learning on graphs[M]. Cambridge University Press, 2021.

[4] Hamilton W L. Graph representation learning[M]. Morgan & Claypool Publishers, 2020.

[5] Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs[J]. Advances in neural information processing systems, 2017, 30.

[6] Leicht, Elizabeth A., Petter Holme, and Mark EJ Newman. Vertex similarity in networks[J]. Physical Review E—Statistical, Nonlinear, and Soft Matter Physics. 73.2 (2006): 026120.

[7] Stoer M, Wagner F. A simple min-cut algorithm[J]. Journal of the ACM (JACM), 1997, 44(4): 585-591.

[8] West D B. Introduction to graph theory[J]. 2001.

[9] Radicchi F, Castellano C, Cecconi F, et al. Defining and identifying communities in networks[J]. Proceedings of the national academy of sciences, 2004, 101(9): 2658-2663.

[10] Ma Y, Tang J. Deep learning on graphs[M]. Cambridge University Press, 2021.

[11] Leskovec J, Lang K J, Mahoney M. Empirical comparison of algorithms for network community detection[C]//Proceedings of the 19th international conference on World wide web. 2010: 631-640.

[12] Hamilton W L. Graph representation learning[M]. Morgan & Claypool Publishers, 2020.

[13] Fortunato S. Community detection in graphs[J]. Physics reports, 2010, 486(3-5): 75-174.

[14] Shi J, Malik J. Normalized cuts and image segmentation[J]. IEEE Transactions on pattern analysis and machine intelligence, 2000, 22(8): 888-905.

[15] Flake G W, Lawrence S, Giles C L. Efficient identification of web communities[C]//Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. 2000: 150-160.

[16] Newman M E J. Modularity and community structure in networks[J]. Proceedings of the national academy of sciences, 2006, 103(23): 8577-8582.

[17] Von Luxburg U. A tutorial on spectral clustering[J]. Statistics and computing, 2007, 17: 395-416.

[18] Bolla M. Spectral clustering and biclustering: Learning large graphs and contingency tables[M]. John Wiley & Sons, 2013.

[19] Trillos N G, Slep D, Von Brecht J, et al. Consistency of Cheeger and ratio graph cuts[J]. Journal of Machine Learning Research, 2016, 17(181): 1-46.

[20] Ng A Y, Jordan M. Weiss. Y., “On spectral clustering: analysis and an algorithm,” [J]. Advances in neural information processing systems, 2002, 14: 84