北京郵電大學
Beijing University of Posts and Telecommunications

第二章

基于特征工程的图机器学习

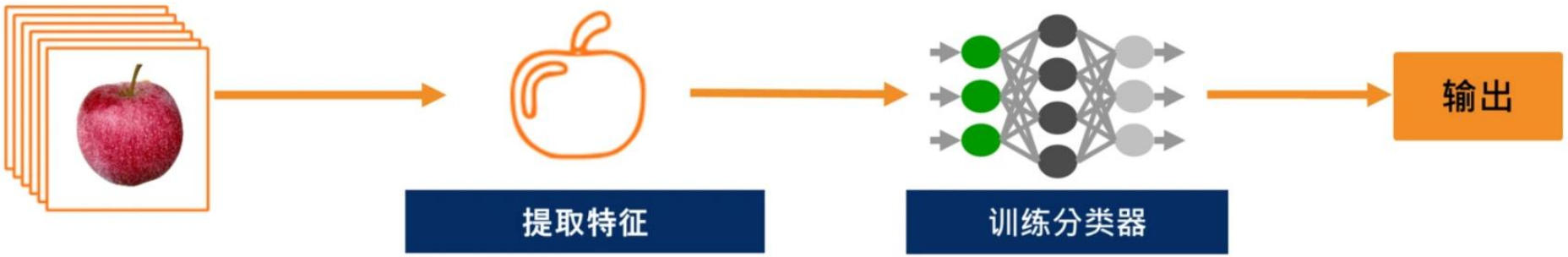
石川 教授
数据科学与服务中心 计算机学院

内容提纲

- 概述
- 2.1 节点级特征
- 2.2 边级特征
- 2.3 图级特征

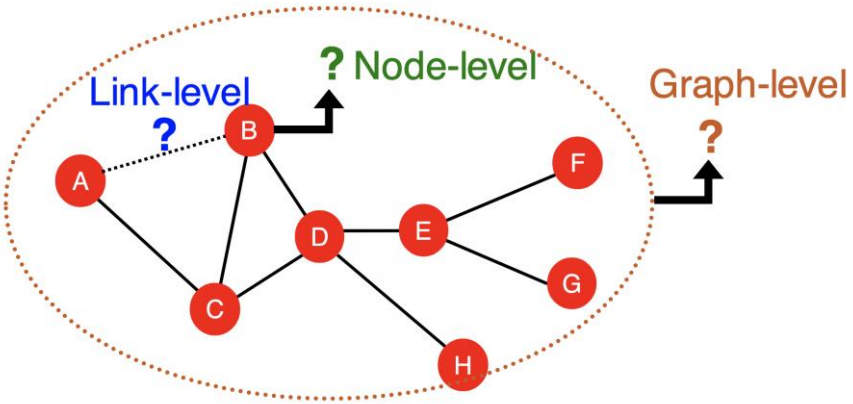
概述

- 在机器学习中，**特征**是用于描述单个数据对象的某些方面的属性或变量。
- 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。
- 邻接矩阵过于稀疏，需要其他特征来提供更稠密以及特定方面的信息。



概述

- 图的节点层面、连接层面、子图/全图层面都需要提取特征成为向量/矩阵后，才能给传统机器学习方法进行处理
- 合理的特征工程将帮助传统机器学习方法发挥更好的效果。
- 虽然如今的图神经网络可以直接作用于图上，不再必须显式抽取图的特征。但是在有些情况下仍需要这些特征的辅助，如节点特征的初始化等。
- 新兴的图transformer模型也依赖显式抽取的图的特征



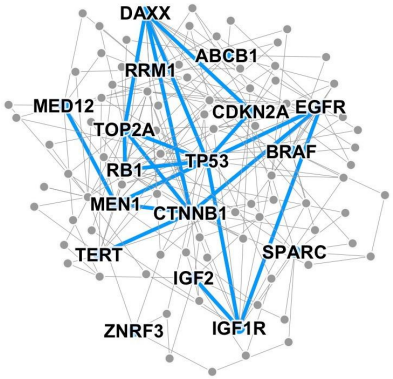
2.1 节点级特征

- 2.1 节点级特征
  - 2.1.1 中心性
    - 节点度
    - 接近中心性
    - 特征向量中心性
    - 中介中心性
  - 2.1.2 局部聚类系数
  - 2.1.3 图元度向量

2.1 节点级特征

- **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。
- 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。

社交网络



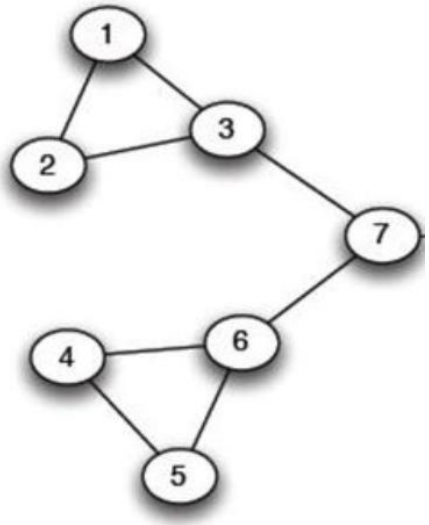
生物网络

2.1.1 中心性

- 在图中，节点的中心性度量该节点在图中的某种重要性。
- 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。
- 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
- 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。

2.1.1 中心性

- 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。
  - 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)*。
  - 在有向图中，入度（In-degree）指从其他顶点指向该顶点的边的数量，记作$deg^-(v)$。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



2.1.1 中心性

- 性质：
  - 在无向图中，所有顶点的度的总和等于边数的两倍，即

$$\sum_{v \in V} \deg(v) = 2|E|$$

  - 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

$$\sum_{v \in V} \deg^{-}(v) = \sum_{v \in V} \deg^{+}(v) = |E|$$

  其中，V是图的顶点集合，E是边的集合。

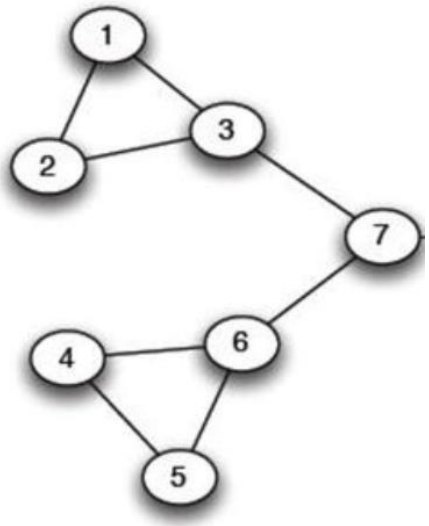
- 节点度只是衡量一个节点有多少个邻居，但这不一定足以衡量节点在图中的重要性。

2.1.1 中心性

- **接近中心性**（Closeness Centrality）衡量的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：

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

  - 其中，d(v,t)是节点 v和节点 t 之间的最短路径距离, n是节点总数
  - 右图中节点7的接近中心性为0.6



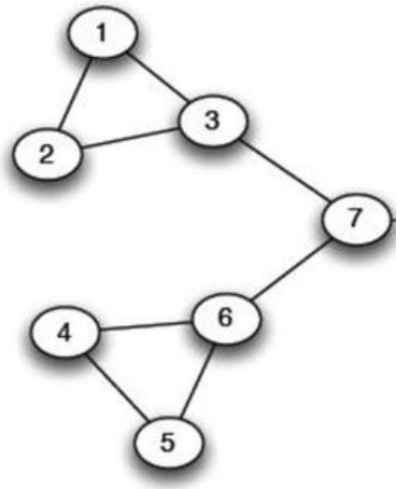
2.1.1 中心性

- **特征向量中心性**（Eigenvector Centrality）进一步考虑与该节点相连的节点的重要性。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 $c_e(v_i) = \frac{1}{\lambda}\sum_{j=1}^{N} A_{ij} \cdot c_e(v_j)$。其中，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。
- 该公式可以被重写为一个矩阵的形式：$c_e = \frac{1}{\lambda} A \cdot c_e$，其中$c_e \in \mathbb{R}^N$是一个包含图中所有节点的中心性得分的向量。
- 通过求解特征方程$\lambda c = Ac$，可以得到单位特征向量c，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，可以选择最大特征值对应的单位特征向量各元素作为各节点中心性得分。

2.1.1 中心性

- 右图的邻接矩阵为

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



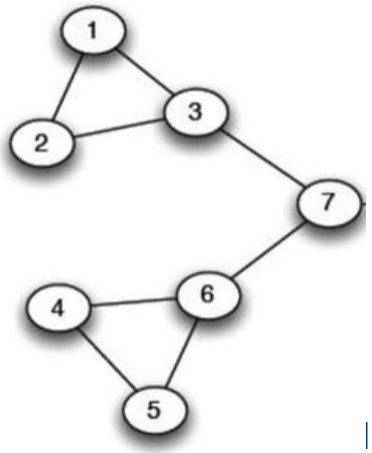
- 其最大特征值为2.3429，对应的单位特征向量为

$$(0.335\ 0.335\ 0.450\ 0.335\ 0.335\ 0.450\ 0.384)$$

各元素对应了**1-7**节点的特征向量中心性的值

2.1.1 中心性

- **中介中心性**（Betweenness Centrality）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。
- 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，$\sigma_{st}$ 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的路径数量。
- 右图中如要计算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



2.1.1 中心性

- 不同的中心性度量有不同的适用场景。
  - 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
- 需要根据图的类型以及具体的问题，依据不同中心性的特点选择合适的中心性度量。

2.1.2 局部聚类系数

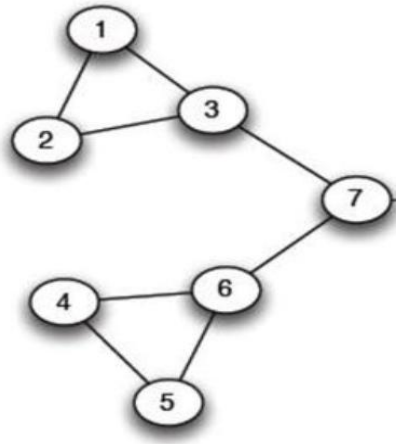
- **聚类系数**是图中节点倾向于聚类在一起的程度的度量。
- **局部聚类系数**（Local Clustering Coefficient）是图论中用于衡量单个节点在其邻居中形成闭合三角形（即完全子图）的程度的指标。该系数反映了图中节点之间的紧密程度或社群性，在社交网络中的群体发现等领域具有广泛应用。
- 局部聚类系数衡量了一个节点的邻居之间有多大可能性彼此也是连接的。如果邻居之间完全连接，则局部聚类系数为1；如果没有邻居之间的连接，则系数为0。
- 对于度数为d_i的节点i，局部聚类系数定义为 $C_i = \frac{E_i}{T_i}$ 。其中，E_i表示节点i的邻居之间实际存在的边的数量，T_i表示节点i的邻居可能（最多）存在的边的数量，$T_i = \frac{d_i \times (d_i - 1)}{2}$ 。

2.1.2 局部聚类系数

- C_i=0如果节点i的邻居都没有相互链接。
- C_i=1如果节点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。
- 聚类系数也可以作为一个图级别的特征。整个图的聚类系数即平均聚类系数：是所有节点的集聚系数的平均值，定义为

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

- 在右图中，节点 3的邻居1、2、7实际存在1条边，最大可能是3条边，所以局部聚类系数是1/3.

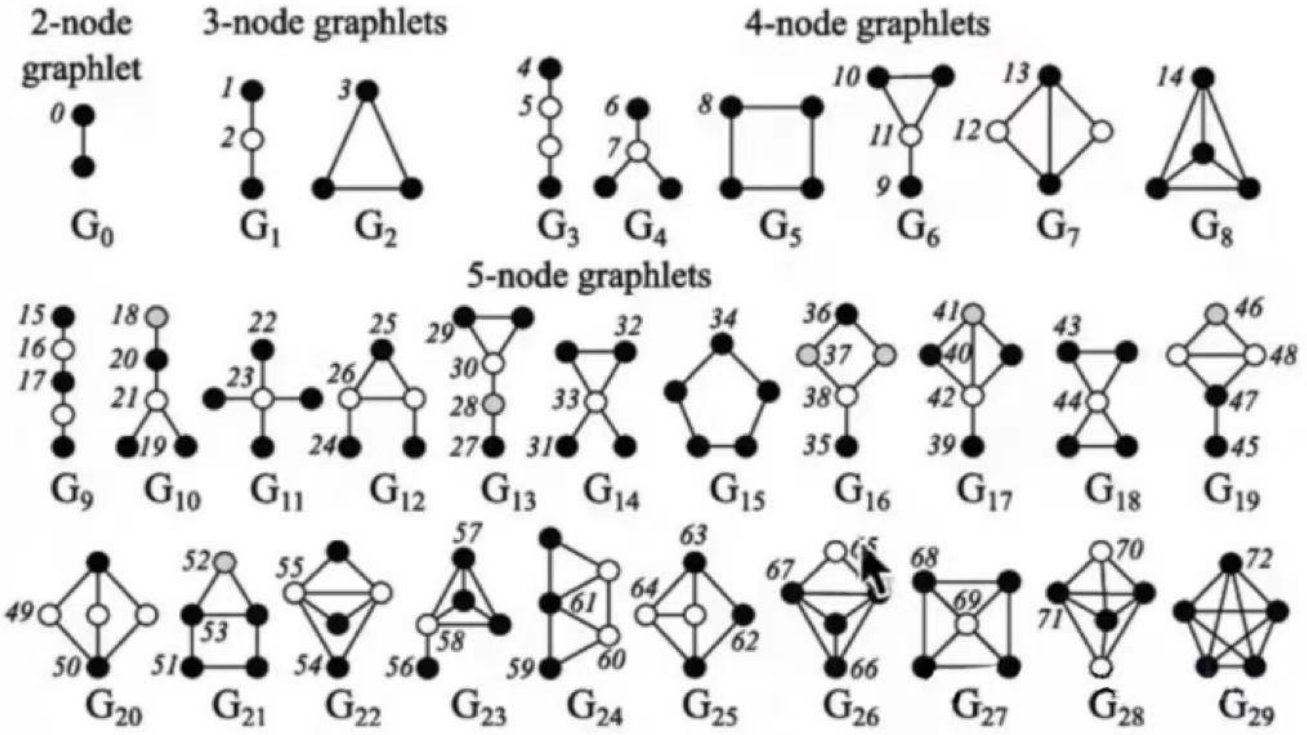


2.1.3 图元度向量

- **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。
- **图元**（graphlet）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。

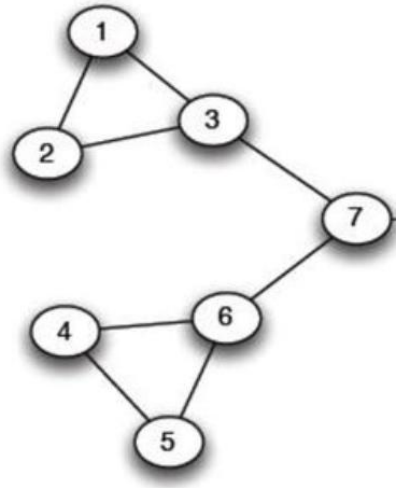
2.1.3 图元度向量

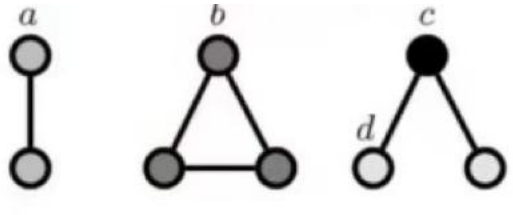
- 在下图中可以看到，当仅有两个节点时，可能的图元结构只有一种；当有三个节点时，图元结构有三种，可能是线型连接的，或是构成一个三角形（这里值得注意的是，当以不同的节点为根节点时，图元是不同的，例如G_1结构代表了两种不同的图元表示）；以此类推，当考虑4个或5个节点时，构成的图元结构将会更多样。



2.1.3 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。以下图为例，1是要观察的根节点，a-d指定了4种不同的图元结构，以1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2,1,0,1，由此可以得到节点v的图元度向量为[2,1,0,1]。



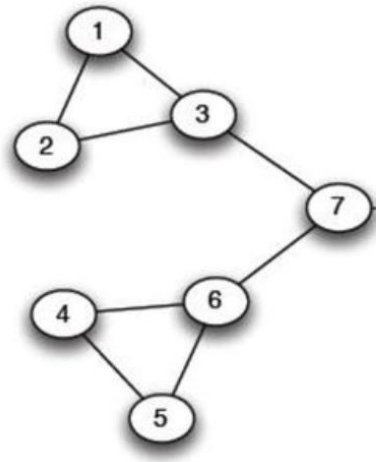


2.2 边级特征

- 2.2 边级特征
  - 2.2.1 基于距离的特征
  - 2.2.2 局部邻域重合
    - 共同邻居数量
    - Sorenson指数
    - Salton指数
    - 资源分配指数
  - 2.2.3 全局邻域重合共同邻居数量
    - Katz指数
    - LHN相似度
    - 随机游走方法

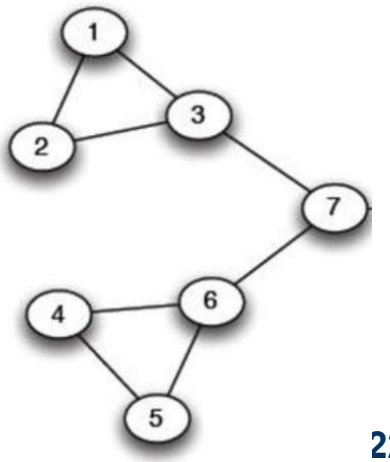
2.2.1 基于距离的特征

- **基于距离的特征**（Distance-based features）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。
- 由此，**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
  - 在有向图中，最短路径长度还可以反映方向性。
- 右图中节点1、7之间最短路径长度为2



2.2.2 局部邻域重合

- 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。
- 如图所示，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。
- **局部邻域重合**（Local neighborhood overlap）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，适用于衡量距离小于等于2节点之间的关系。



2.2.2 局部邻域重合

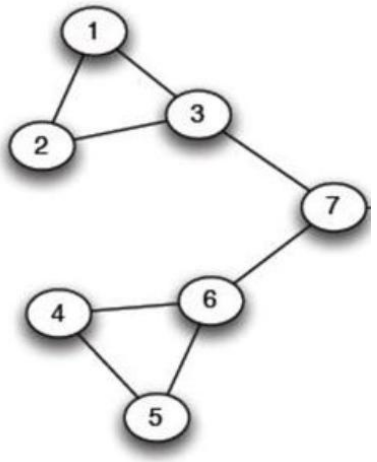
- **共同邻居数量**（Common neighbors）统计了两个节点的共同邻居数量，以公式表示为：$S[u,v] = |N(u) \cap N(v)|$，

  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。

- 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。

- 图中1、7节点的共同邻居数量为1

- 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。



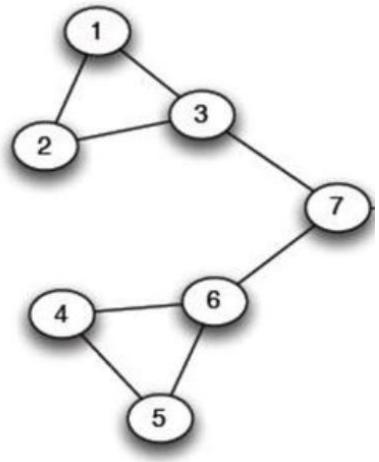
2.2.2 局部邻域重合

- **Sorenson指数**（也称为Dice相似系数）用于衡量两个节点共享邻居的比例，其计算公式为：

$$\text{Sorenson}(u,v) = \frac{2\cdot|N(u)\cap N(v)|}{|N(u)|+|N(v)|} = \frac{2\cdot S[u,v]}{du+dv}$$

- 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。图中1、7节点Sorenson指数为$\frac{2\times1}{2+2}=0.5$

- 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。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。

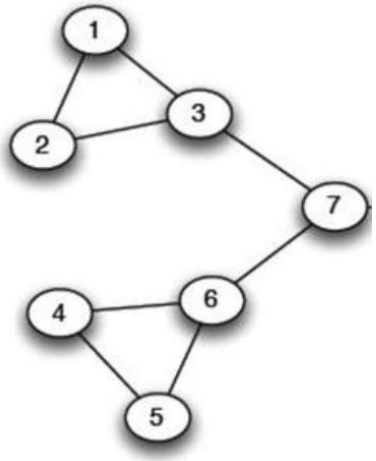


2.2.2 局部邻域重合

- Salton指数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：

$$\mathbf{Salton}(u,v)=\frac{2\cdot|N(u)\cap N(v)|}{\sqrt{|N(u)|\cdot|N(v)|}}=\frac{2\cdot S[u,v]}{\sqrt{\mathrm{dudv}}}$$

- 其中 $\sqrt{|N(u)|\cdot|N(v)|}$ 是两个节点邻居数量的几何平均值。
- 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。



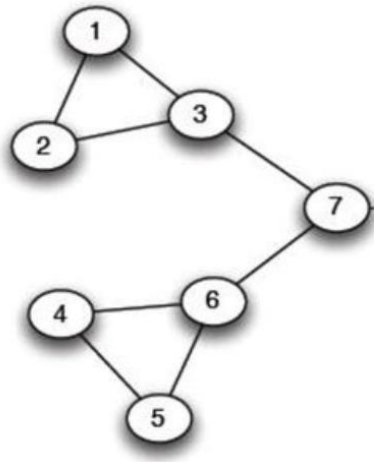
- 右图中1、7节点的Salton指数为$\frac{2\times1}{\sqrt{2\times2}}=1$
- 类似的指数还有Jaccard指数，是是共同邻居（交集）的数量除以所有涉及的邻居（并集）的数量

2.2.2 局部邻域重合

- **资源分配指数**（Resource Allocation Index，RA指数）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。计算公式为：

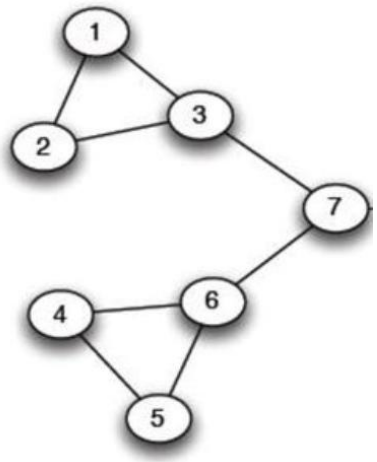
$$\mathrm{RA}(u,v)=\sum_{w\in N(u)\cap N(v)}\frac{1}{|N(w)|}=\sum_{w\in N(u)\cap N(v)}\frac{1}{\mathrm{d}_w}$$

- 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。
- 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。
- 图中1、7节点的共同邻居只有3，其度为3，所以RA指数为1/3
- 类似的还有AA指数，是共同邻居的度取log后再求和



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。
- 例如在下图中，节点 1与节点 6 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）以内的邻居关系，在该例子中节点 1 与节点 6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。
- 全局邻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。它考虑图的整体结构，适合在处理大规模图或社区结构明显的图时使用。



2.2.3 全局邻域重合

- **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。

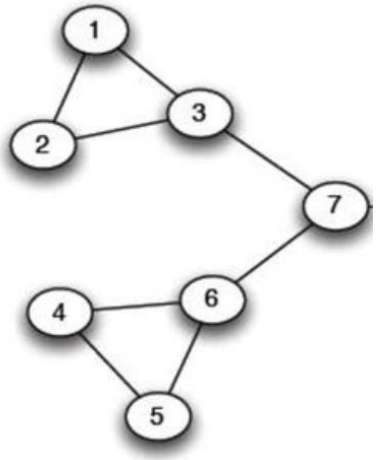
- 其计算公式为：$Katz[u,v] = \sum_{l=1}^{\infty} \beta^l A_{uv}^l$

- 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。

2.2.3 全局邻域重合

- 每个长度的路径数量A_{uv}^{l}可以利用邻接矩阵的幂来计算。例如：邻接矩阵A中的元素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A^2中的元素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A^l中的元素A_{uv}^{l}即对应了节点u与v之间长度为l的路径数量。这样可以计算任意节点对之间的Katz指数。
- 如果设$\beta = 0.5$,则图中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
- Katz指数的一个问题是，它受到节点度的强烈偏差。在考虑大度数节点时，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。



2.2.3 全局邻域重合

- 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)，其计算公式为：

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

- 其中A表示图的邻接矩阵，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$E[A^i[v_1, v_2]]$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。两节点之间的期望边数，即长度为1的路径数为：$E[A[v_1, v_2]] = \frac{d_{v1} d_{v2}}{2m}$

2.2.3 全局邻域重合

$$E[A[v_1, v_2]] = \frac{d_{v1} d_{v2}}{2m}$$

- 其中，A[v_1,v_2]表示节点v1和v2之间是否存在边，m表示图中边的总数。上式表明，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（即通过一个中间节点连接的路径），可以进一步计算其概率：

$$E[A^2[v_1 v_2]] = \frac{d_{v1} d_{v2}}{(2m)^2} \sum_{u \epsilon V} (d_u - 1) d_u 。$$

其中，分别考虑从v_1到中间节点u的路径概率$\frac{d_{v1} d_u}{2m}$，以及从u到v_2的路径概率$\frac{d_{v2}(d_u - 1)}{2m}$（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），路径的总概率是这两部分概率的乘积。

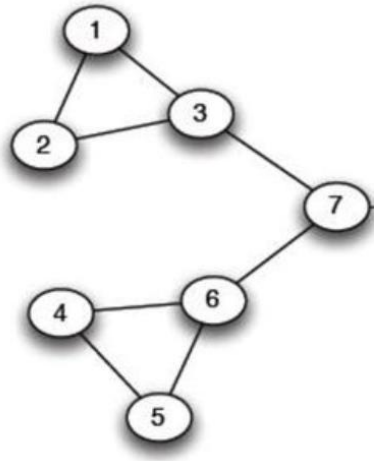
2.2.3 全局邻域重合

$$E[A^2[v_1v_2]] = \frac{d_{v1}d_{v2}}{(2m)^2}\sum_{u\epsilon V}(d_u - 1)d_u。$$

- 试计算图中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：

$$E\left[A^2[v_1v_2]\right] = \frac{2\times 2}{(2\times 8)^2}2\times 3 = \ 0.09375$$

$$A^2[v_1v_2] = 1$$



- 故1、7节点的LHN相似度为

$$\frac{1}{0.09375} = 10.67$$

2.2.3 全局邻域重合

- **随机游走方法**(Random walk methods)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。
- **原理**: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。
- **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。

2.2.3 全局邻域重合

- **SimRank**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：

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

- 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。

2.2.3 全局邻域重合

- **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。
- 随机游走的转移概率矩阵$P=AD^{-1}$，其中A是图的邻接矩阵，D是度数矩阵，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：

$$q_u = cPq_u + (1-c)e_u$$

- 在该方程中，e_u是节点u的一位指示向量，是一个长度等于图中节点数的向量，表示从节点 u 开始的随机游走。$q_u[v]$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。
- 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。

2.2.3 全局邻域重合

- 上述递推方程的解为：

$$q_u = (1-c)(I-cP)^{-1}e_u$$

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

- 节点间的随机游走相似性度量为：

$$S_{RW}[u,v] = q_u[v] + q_v[u]$$

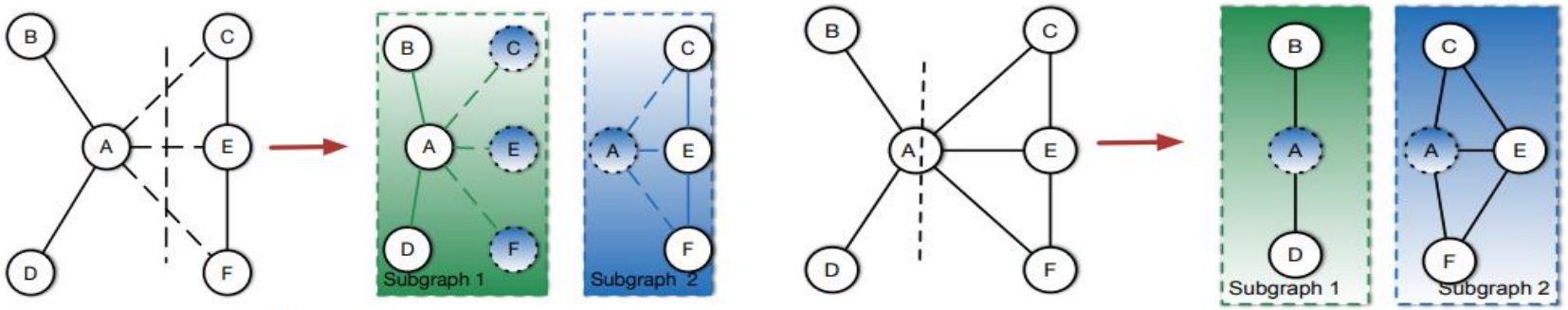
即节点对之间的相似性与从另一个节点开始的随机游走访问该节点的可能性成正比。如果从节点u开始的随机游走经常到达节点v，或者反过来，从节点v 开始的随机游走经常到达节点u ，那么这两个节点就被认为是相似的。

2.3 图级特征

- 2.3 图级特征
  - 2.3.1 图划分
  - 2.3.2 图内部的特征
    - 空间特征
    - 谱特征
  - 2.3.3 子图间的特征
    - 基于连通性
    - 基于网络模型
  - 2.3.4 不同图间相似性特征
    - 图核方法概述
    - 图元核
    - Weisfeiler-Lehman核方法

2.3.1 图划分

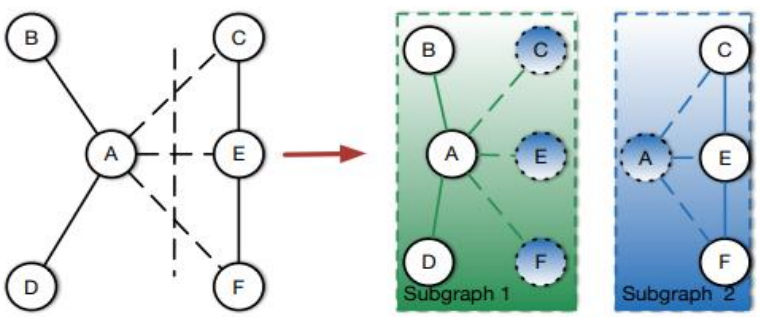
- **图划分（Graph Partitioning）**是图论中一个重要的概念，涉及将图中的节点或边集合划分为若干个不相交的子集，是后续许多图级特征的基础。
- 根据对**图数据的切分方式**分类，图划分可以分为
  - 点分割（Vertex Partitioning or Edge-cut Partitioning）
  - 边分割（Edge Partitioning or Vertex-cut Partitioning）



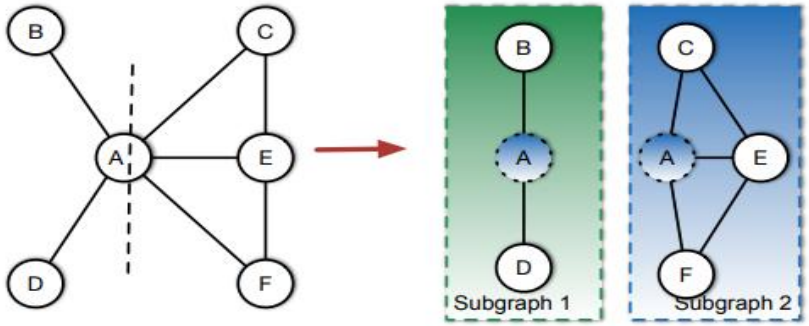
(a) edge-cut　　(b) vertex-cut

2.3.1 图划分

- 点分割：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，**如下图(a)所示。
- 边分割：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，**如下图(b)所示。
- 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**。



(a) edge-cut



(b) vertex-cut

2.3.1 图划分

- 割值（Cut Value）
  - 在图G中，设$A \subseteq V$表示图中节点的一个子集，$\overline{A}$表示这个子集的补集，即$A \cup \overline{A} = V$，且$A \cap \overline{A} = \emptyset$。当将图G中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，定义该划分下的割值为：

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

  - 简单来说，割值表示跨越节点划分边界的边数总和。
- 图划分中的“割值”概念是评估划分效果的一个重要指标，满足特定条件下最小化割值的划分则被称为最优图划分。

2.3.2 图内部的特征

- **图内部的特征：**对图整体特性的一种量化分析，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。
  - **空间特征：**空间特征在图分析中占据重要地位，尤其是当试图理解图的几何形态时，这些特征能够提供直观的解释。
  - **谱特征：**谱图理论主要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。
- 由这两大类特征，逐渐发展出后面章节介绍的空域图神经网络和谱域图神经网络。

2.3.2 图内部的特征—空间特征

- 内部边数（Edge Inside）：表示图G内部的边的数量

$$\mathrm{f(G)} = \left|E_{in}^{G}\right|$$

- 内部密度（Internal Density）：衡量图G内部节点之间连边的密集程度

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

- 平均度（Average Degree）：表示图G中每个节点的平均度

$$f(G) = \frac{1}{N}\sum_{i=1}^{N} d_i = \frac{2\left|E_{in}^{G}\right|}{N}$$

2.3.2 图内部的特征—空间特征

- 超过中位数的比例（Fraction over Median Degree，FOMD）：衡量图中的一个子图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：

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

其中，d_m表示整个图中节点度数的中位数

- 超过中位数的比例（Triangle Participation Ratio，TPR）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）：

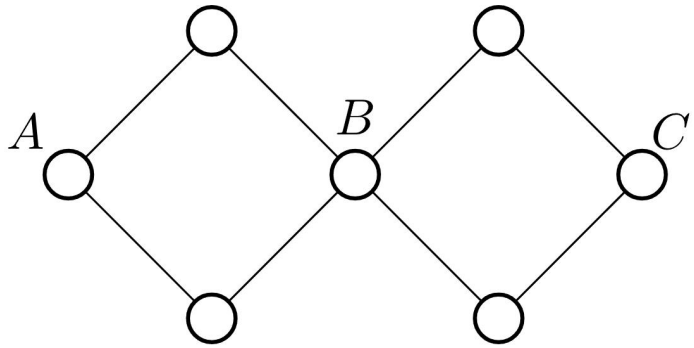
$$\mathrm{f(G)} = \frac{|u: u \in G, \{v,w \in G, (u,v) \in E, (u,w) \in E, (v,w) \in E\} \neq \emptyset|}{N}$$

2.3.2 图内部的特征—空间特征

- **连通图：**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。
- 点连通度
  - **点割集：**对于连通图G=(V,E)，如果**存在一个顶点子集A ⊆ V且G[V\A]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$|V| \geq k+1$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，记作k(G)。

2.3.2 图内部的特征—空间特征

- 边连通度
  - **边割集：**对于连通图G=(V,E)，若**F ⊆ E且G(V,E\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λ(G)。
- 下图点连通度为1，边连通度为2。可以证明，一个图的点连通度总是小于等于其边连通度。



2.3.2 图内部的特征—谱特征

- 常用的图拉普拉斯矩阵有三种，这三种矩阵表达的含义相似，但性质上略有区别。

方式	定义	适用场景
原始拉普拉斯	$\mathbf{L}=\mathbf{D}-\mathbf{A}$	适用于基本的图聚类任务
标准化拉普拉斯	$\mathbf{L}=\mathbf{I}-\mathbf{D}^{-1/2}\boldsymbol{A}\mathbf{D}^{-1/2}$	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	$\mathbf{L}=\mathbf{I}-\mathbf{D}^{-1}\boldsymbol{A}$	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

2.3.2 图内部的特征—谱特征

- 原始拉普拉斯矩阵
  - 设f是图上的一个信号向量，其第i个元素$\mathbf{f[i]}$与节点$\boldsymbol{v_i}$相关。将L与f相乘，会得到一个新的向量h

$$\mathbf{h = Lf = (D - A)f = Df - Af}$$

h向量的第i个元素可以被表示为：

$$\boldsymbol{h[i] = d(v_i) \cdot f[i] - \sum_{j=1}^{N} A_{i,j} \cdot f[j]}$$

$$\boldsymbol{= d(v_i) \cdot f[i] - \sum_{v_j \in N(v_i)} A_{i,j} \cdot f[j]}$$

$$\boldsymbol{= \sum_{v_j \in N(v_i)} (f[i] - f[j]))}$$

  - h[i]是节点v_i与其邻居$\mathrm{N}(v_i)$在f上的差值的总和

2.3.2 图内部的特征—谱特征

- 原始拉普拉斯矩阵
  - 尝试计算$f^T L f$

$$f^T L f = \sum_{v_i \in V} f[i] \sum_{v_j \in N(v_i)} (f[i] - f[j])$$

$$= \sum_{v_i \in V} \sum_{v_j \in N(v_i)} (f[i] \cdot f[i] - f[i] \cdot f[j])$$

$$= \sum_{v_i \in V} \sum_{v_j \in N(v_i)} \left(\frac{1}{2} f[i] \cdot f[i] - f[i] \cdot f[j] + \frac{1}{2} f[j] \cdot f[j] \right)$$

$$= \frac{1}{2} \sum_{v_i \in V} \sum_{v_j \in N(v_i)} (f[i] - f[j])^2$$

  - $f^T L f$是相邻节点之间差值的平方和的一半

2.3.2 图内部的特征—谱特征

- 标准化拉普拉斯矩阵
  - 假设W是未经标准化的邻接矩阵，A是标准化后的邻接矩阵$A = D^{-1/2}WD^{-1/2}$，$d_i := d(v_i)$，则，
    - ① $A_{ii} = 0$，
    - ② $A_{ij} = 1/\sqrt{d_i d_j}, (i,j) \in E$，
    - ③ $A_{ij} = 0, (i,j) \notin E$且$(Av)_i = \frac{1}{\sqrt{d_i}} \sum_{j:(i,j)\in E} \frac{1}{\sqrt{d_j}} v_j$
  - 类似的，将一个信号向量右乘标准化拉普拉斯矩阵也可度量该信号在图中的变化量：

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

  - $(Lv)_i$得到的是第i个节点相对于其邻居的变化量

2.3.2 图内部的特征—谱特征

- 标准化拉普拉斯矩阵的特征值与特征向量
- 性质
  - 标准化拉普拉斯矩阵是对称的
  - 标准化拉普拉斯矩阵是半正定矩阵，特征值非负
  - 对于一个含有N个节点的图G（这里为正则图），标准化拉普拉斯矩阵总存在特征值0：令$\mathrm{u}_1 = \frac{1}{\sqrt{\mathrm{N}}}(1, \cdots, 1)$，显然$\mathrm{Lu}_1 = 0 = 0\mathrm{u}_1$，即$\mathrm{u}_1$是特征值0的特征向量
  - 本章以下均讨论标准化拉普拉斯矩阵

2.3.2 图内部的特征—谱特征

- 特征值与单位特征向量
  - 单位特征向量之间满足$\mathrm{u_i u_i^T} = 1$且$\mathrm{u_i u_j^T} = 0$
  - 特征值满足$\lambda_\mathrm{i} = \mathrm{u_i^T L u_i}$
  - 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$f^T L f$度量了信号f在图上变化的平方和。那么令$f = u_i$，有

$$u_i^T L u_i = u_i^T \lambda_i u_i = \lambda_i$$

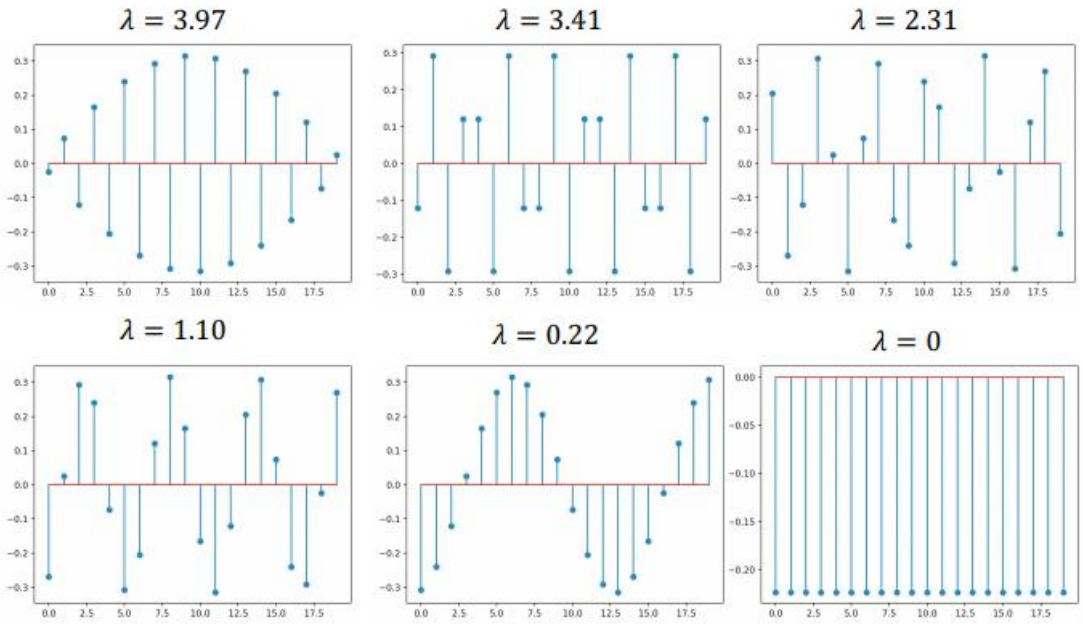
即λ_i的值越大，表面其对应单位特征向量所对应的图信号在图上的变化越剧烈。故λ_i也称对应特征向量的平滑度。

2.3.2 图内部的特征—谱特征

- 特征值与特征向量

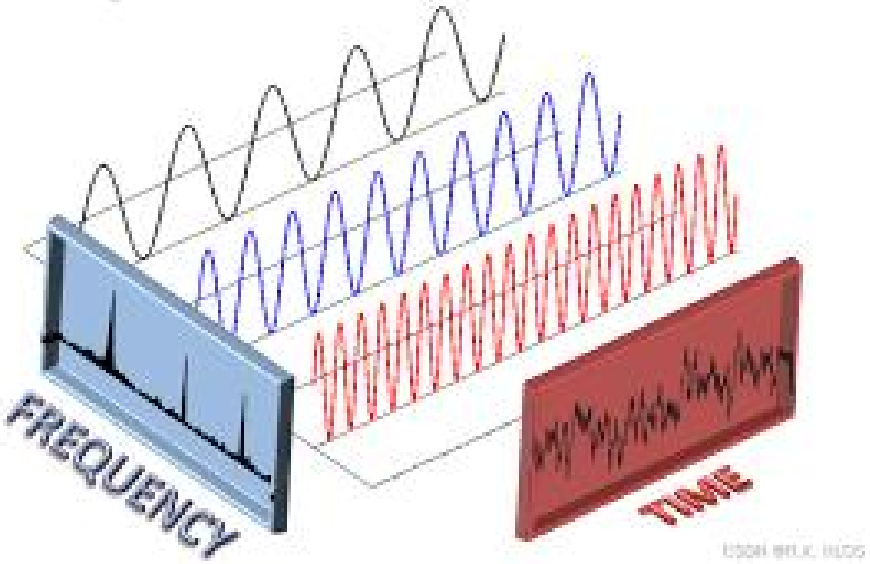
  - 以下是不同特征值对应的特征向量的实例，可以看到特征值越大，其对应的特征向量越不平滑

  - 如果将上述分解得到的U^T与一个图信号向量x相乘，即$\hat{\mathrm{x}} = \mathrm{U}^{\mathrm{T}}x$，则信号x可以通过拉普拉斯矩阵的特征向量进行线性加权来表示。根据正交矩阵的性质，逆变换也容易得到，即$x = U\hat{x}$。



2.3.2 图内部的特征—谱特征

- 图傅里叶变换
  - 从数学的角度讲，与U^T相乘的过程本质是将信号x向U^T构成的空间进行投影。上述过程被称为图傅里叶变换，拉普拉斯矩阵的特征向量被称为傅里叶基，通过傅里叶变换后的信号$\hat{x}$被称为傅里叶系数。
  - 不同特征值对应的特征向量也可以看作不同变化剧烈程度的图信号，也就是可以看作图上不同频率的基。这样，图傅里叶变换与普通的傅里叶变换有着异曲同工之妙，都是将一个信号转换到频域上。

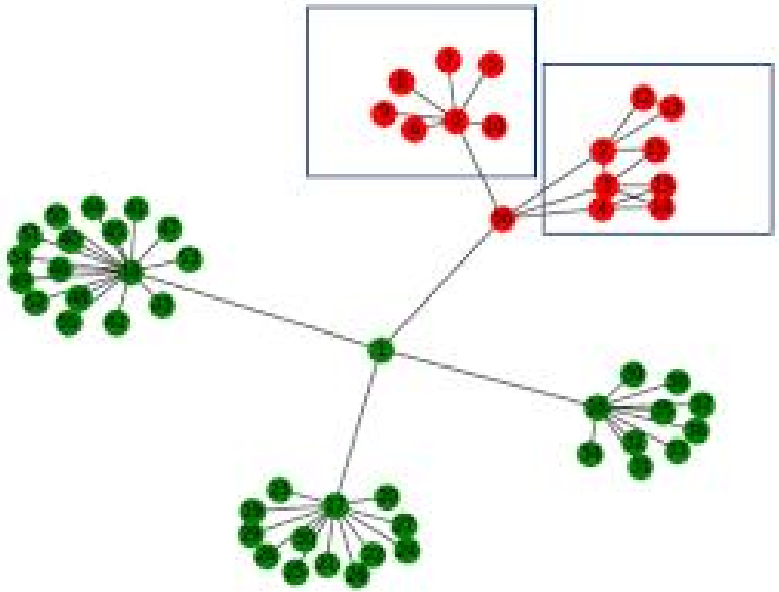


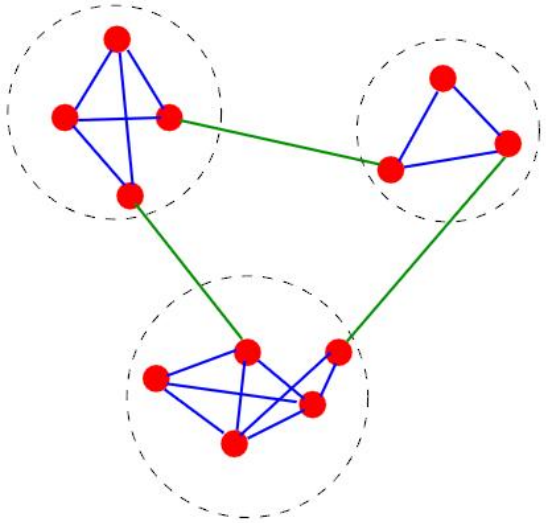
2.3.2 图内部的特征—谱特征

- 特征值与特征向量
  - 指示向量：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。
  - 令G是一个无向有权图，则**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**。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。

2.3.3 子图间的特征

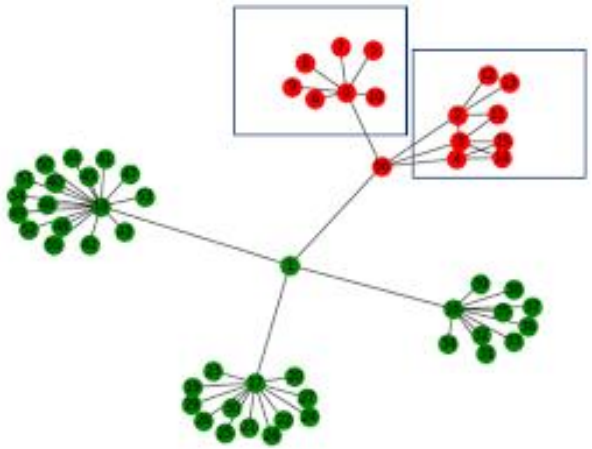
- 社区结构是网络研究中的一个重要概念，尤其在数据挖掘和社交网络分析领域受到广泛关注。
- 一般认为社区结构是将网络中的节点划分为若干个集合，其中每个集合内的节点之间连接密集，而集合之间的连接稀疏。

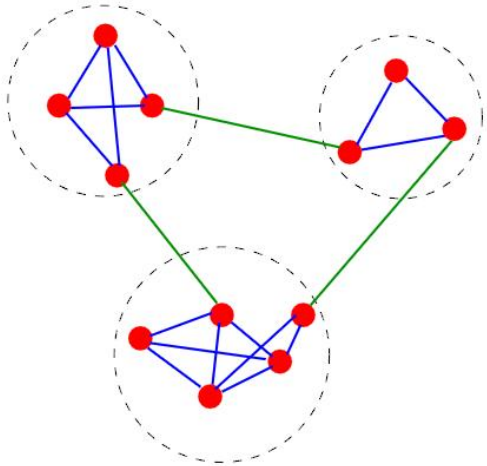




2.3.3 子图间的特征

- 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。
- 子图间特征分析是社区检测中的重要步骤，这一分析能够帮助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。





2.3.3 子图间的特征—基于连通性

- 比割值（Ratio Cut）：令$A_1, \cdots, A_K$表示将图划分为K个子集，$|A_k|$表示第k个子集中的节点数

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

- 扩展系数（Expansion）：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

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

其中$|E_\omega^{out}|$表示从子图ω指向外部的边数量，$|\omega|$表示子图的节点数量。

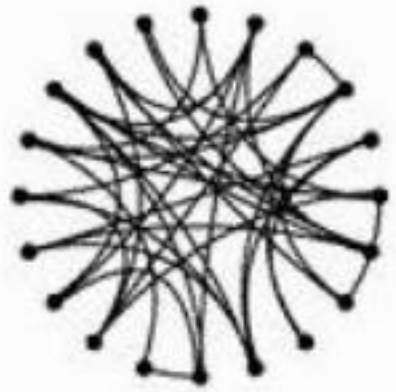
- 割比率（Cut Ratio）：衡量从子图中离开的边占所有可能的边的比例

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

其中，$|G \setminus \omega|$表示全图中不属于子图ω的节点数量。

2.3.3 子图间的特征—基于网络模型

- 网络模型：用于描述和理解网络结构及动态特性的理论模型。这些模型帮助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。
- 比如在一个随机网络模型中，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。
- 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：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。



2.3.3 子图间的特征—基于网络模型

- 模块度
  - 无权无向图的定义为

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

其中，$|E_{\omega}^{in}|$表示子图ω内部的边数，$|E_{\omega}^{out}|$表示从子图ω指向外部的边数，|E|表示图的总边数。

  - $\frac{|E_{\omega}^{in}|}{|E|}$表示社区内部的边占总边数的比例
  - $\frac{|E_{\omega}^{in}|+|E_{\omega}^{out}|}{2|E|}$是社区连接到内部和外部的边占总边数的比例

2.3.3 子图间的特征—基于网络模型

- 模块度
  - 另一种定义方式为

$$Q_{ud} = \frac{1}{2|E|}\sum_{i,j}[A_{ij} - \frac{d(i)d(j)}{2|E|}]\delta_{\omega_i,\omega_j}$$

其中，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，$\delta_{\omega_i,\omega_j}$是Kronecker delta 函数，当i和j属于同一子图时返回1，否则为0。

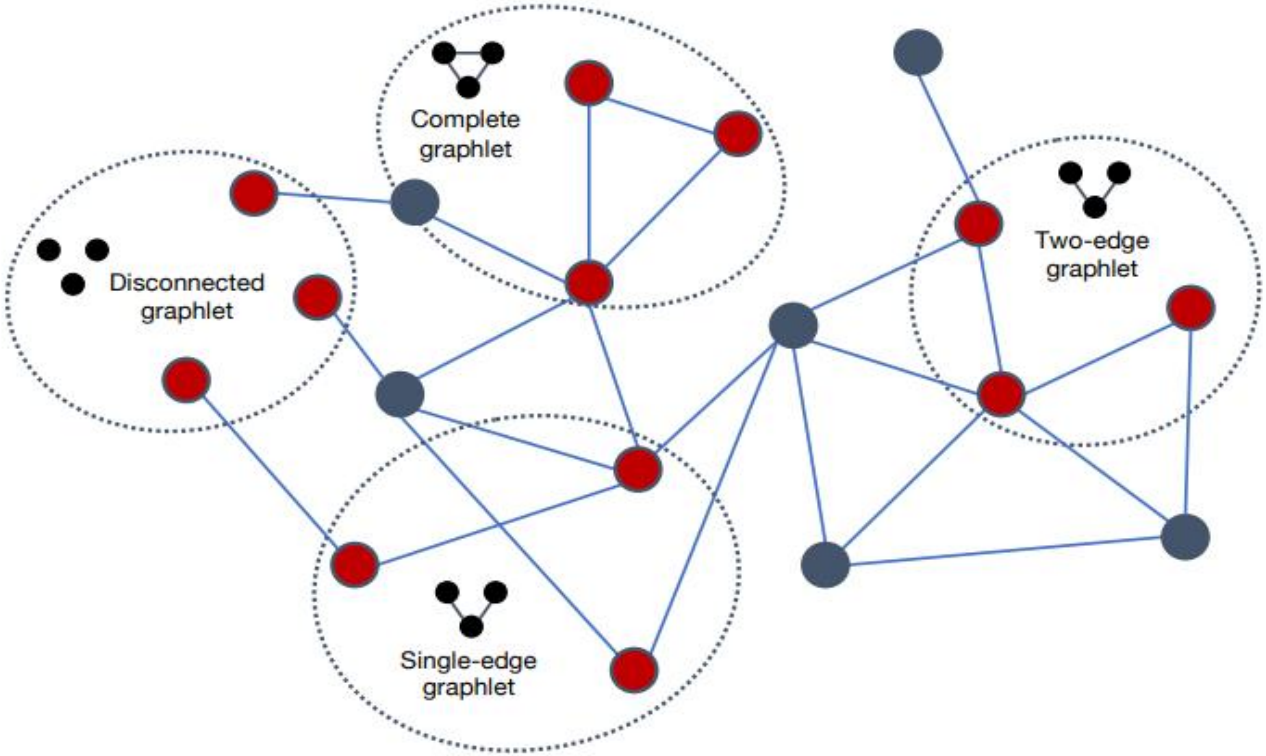
  - $\frac{1}{2|E|}\sum_{i,j}A_{ij}$表示子图内部实际的边数的比例
  - 事实上，如果把同一个图中的边随机放置，则节点i和节点j之间边树的期望值是$\frac{d(i)d(j)}{2|E|}$，$\frac{1}{2|E|}\sum_{i,j}\frac{d(i)d(j)}{2|E|}$表示随机情况下社区内部期望的边数的比例
  - 因此，模块度的定义可以看作是，在社区内部的边的比例，减去边随机放置时社区内部期望边数的比例。模块度值越高，表示图中的子图结构越明显，即子图内部的节点连接紧密，子图之间的连接稀疏。

2.3.4 不同图间相似性特征—图核方法概述

- 核方法（Kernel methods）：一种广泛使用的图水平的特征表示方法。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$K(G, G') \in \boldsymbol{R}$去计算两个图的相似度
- 然而，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。
- 可以对图使用词袋法（Bag-of-Words，BOW）以获得图的全局向量特征。
  - 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。

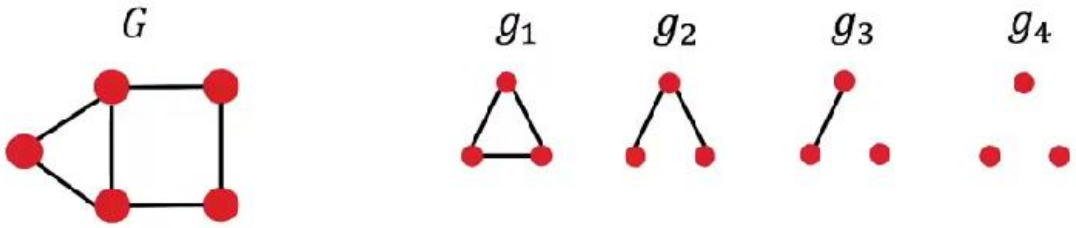
2.3.4 不同图间相似性特征—图元核

- 图元（Graphlet）：在固定数量节点下构成的任意图结构。
  - 例如，下图展现了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。



2.3.4 不同图间相似性特征—图元核

- 图元核方法的基本思想：对图网络中的各种图元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。
- 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$\mathrm{f_G} \in \mathrm{R^{n_k}}$，其中$(f_G)_i = \#(g_i \subseteq G)$，$\mathrm{i} = 1,2,\cdots,\mathrm{n_k}$，即图$G$中该种图元的个数。
- 下图对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，$f_G = (1,6,3,0)$。



2.3.4 不同图间相似性特征—图元核

- 图元核计算方法：给定两个图G和G'，$\mathrm{K(G,\ G')} = f_G^T f_{G'}$，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：

$$\mathrm{h_G} = \frac{f_G}{Sum(f_G)} \qquad \mathrm{K(G, G')} = h_G^T h_{G'}$$

- 但是计算图元是非常昂贵的。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$\mathrm{O(n^k)}$的时间。

2.3.4 不同图间相似性特征—WL核方法

- Weisfeiler-Lehman（WL）核方法：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。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

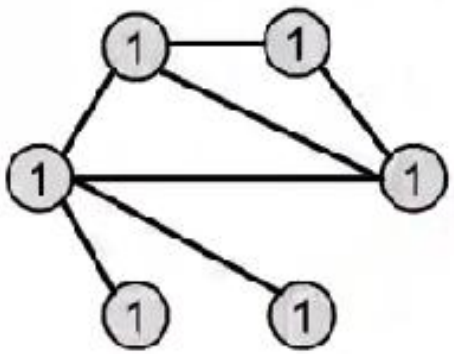
- WL核方法的步骤如下：
  - 初始标签赋值：首先，给每个节点分配一个初始标签$l^{(0)}(v)$。比如，这个初始标签可以设置为节点的度数，也可以是其他值。
  - 标签更新：接下来，通过对节点邻域内的当前标签集进行哈希处理，迭代地为每个节点分配一个新标签。新标签的计算公式为：

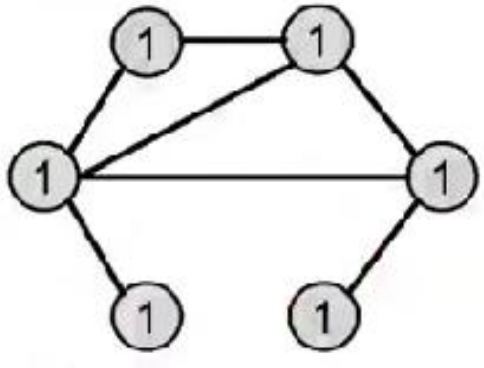
$$l^{(\mathrm{i})}(v) = HASH\left(\left\{\left\{l^{(i-1)}(u)\middle|\forall u \in N(v)\right\}\right\}\right)$$

  - 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。
  - 特征表示和核计算：在运行K次重新标签迭代（即步骤2）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。最后，通过测量两幅图的最终标签集向量之间的差异来计算WL核。

2.3.4 不同图间相似性特征—WL核方法

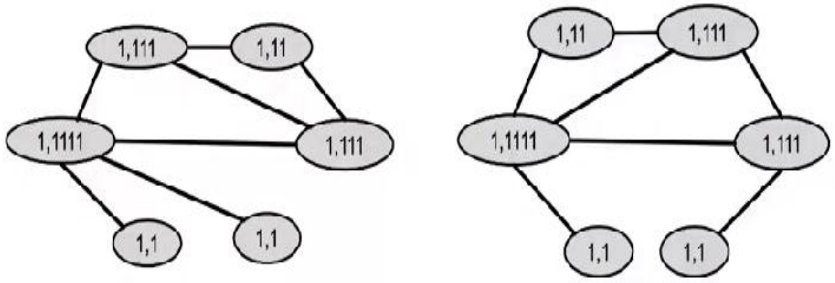
- WL核方法相比于图元核方法拥有更高的运行效率，总体时间复杂度为$O(|E|)$，因此在实际应用中也更加广泛。
- 也有书将赋予和更新的节点标签称为颜色，因为该核方法采用了一个名为“Color Refinement”的算法，下面基于这种叫法给出WL核方法的具体例子。首先，给定两个图，为每个节点指定一个初始颜色，这里颜色使用数字作为替代



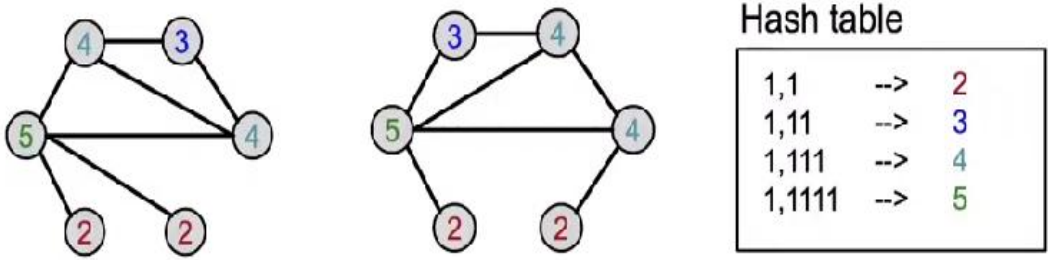


2.3.4 不同图间相似性特征—WL核方法

- 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，因此聚合后的信息变成了(1, 111)：

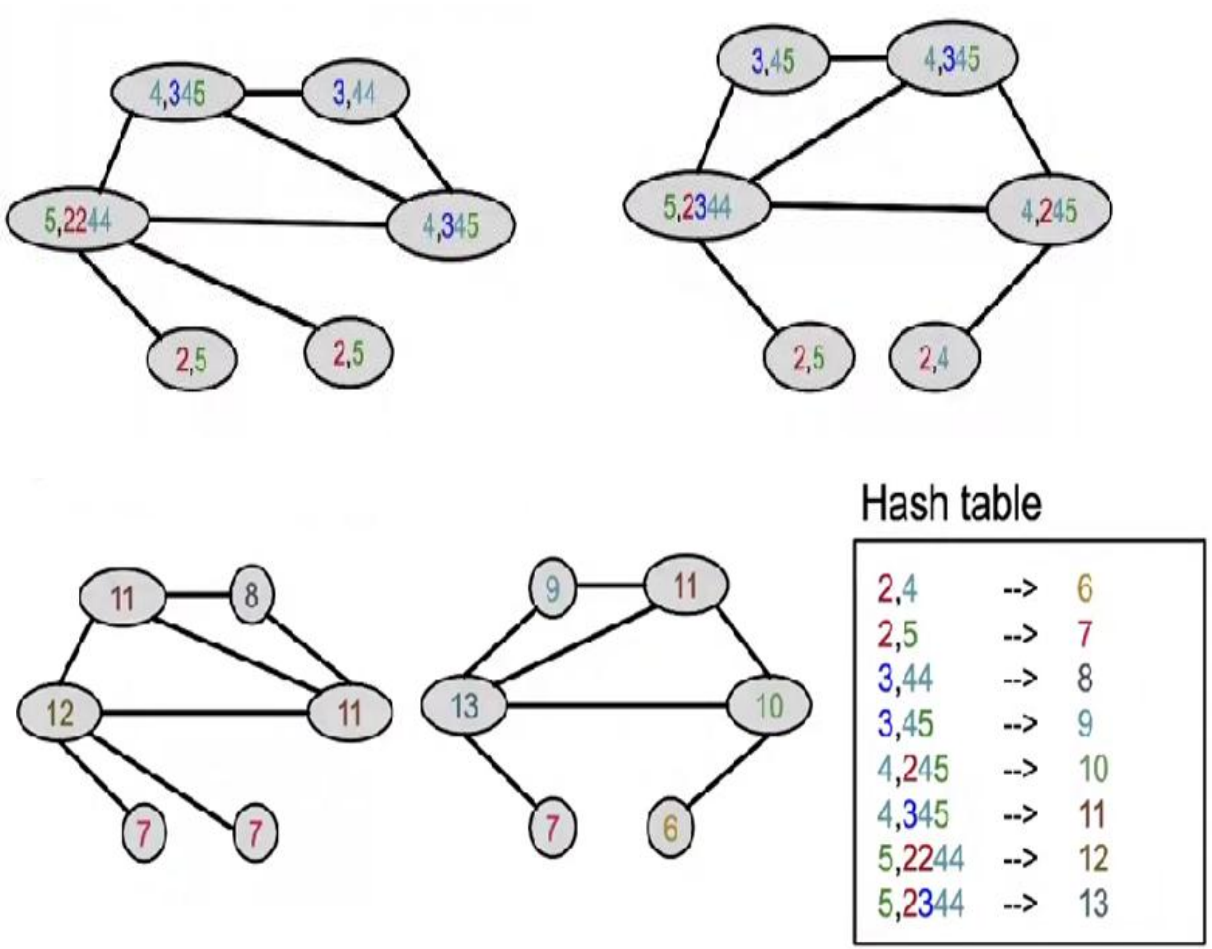


- 根据HASH表映射每个节点聚合后的颜色，仍然以第一个图左上角节点为观察对象，经过HASH映射，它由(1, 111)映射成了对应的颜色4：



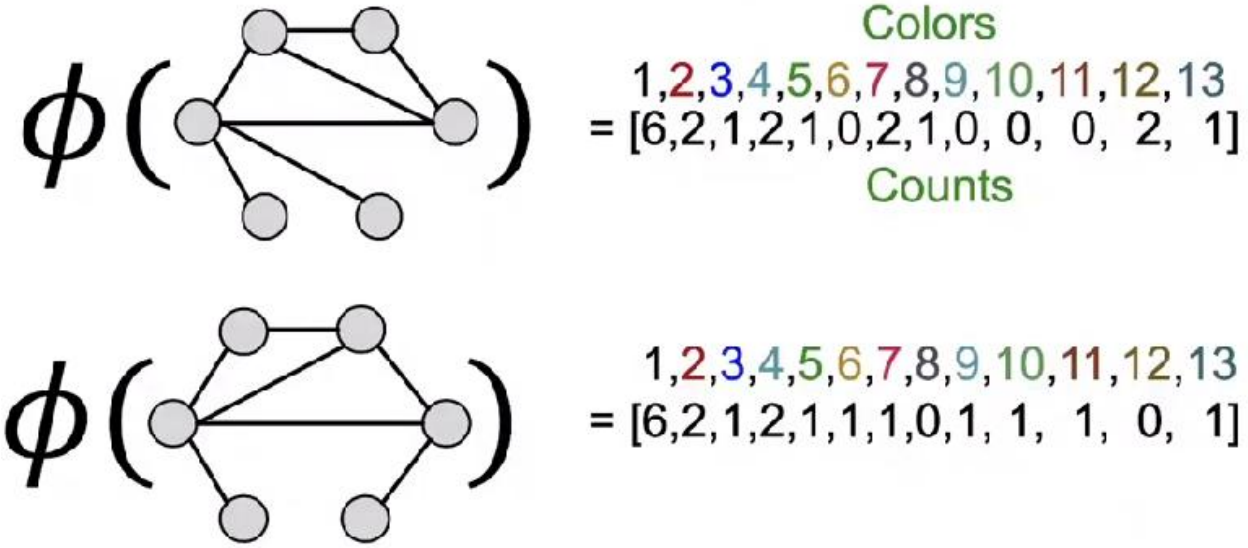
2.3.4 不同图间相似性特征—WL核方法

- 假设经过两轮迭代完成了Color Refinement过程：



2.3.4 不同图间相似性特征—WL核方法

- 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特征向量表示：



- 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

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