

第 5 章

图 计 算

图数据是节点和边的集合,其中节点表示实体或对象,边代表实体之间的关系或连接。图计算是一种处理图数据的计算模型,通过对节点和边进行操作和计算,发现节点间的模式和关联性等信息,解决复杂的图网络分析问题。本章将介绍图数据的统计特性、图遍历算法、图分割算法、社区发现算法以及图计算平台等相关内容。

5.1 问题导入

在同一篇论文中署名的作者,即表示他们之间存在合作关系。对某一领域科研工作者之间的合作关系进行挖掘,有助于了解该领域科研团队的研究方向及进展。合作者关系数据可以使用图网络来进行建模,其过程如图 5-1 所示。为实现这一目标,需要解决以下问题:

- (1) 如何使用图网络来表示实体之间的合作关系;
- (2) 如何对图网络中的节点和边进行统计分析,以描述网络节点的属性和结构特征;
- (3) 如何对图网络进行计算,以实现图的分割和节点的聚类任务。

围绕上述问题,本章将从图网络、图基础算法、社区发现等方面进行介绍。

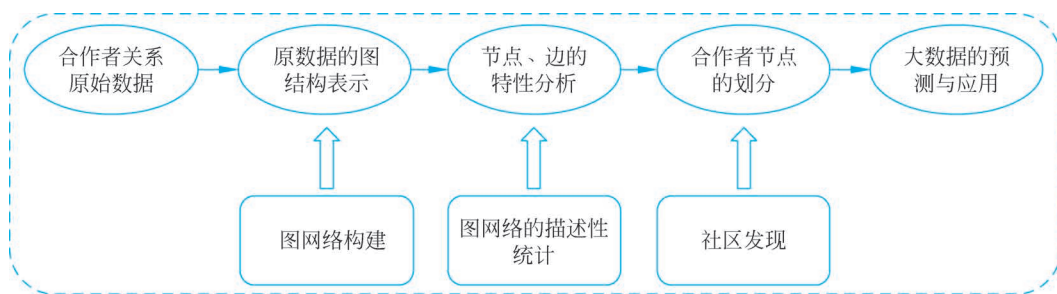


图 5-1 合作者关系建模过程

5.2 图网络

图网络常用于刻画社交网络、交易网络和通信网络等存在复杂网络关系的数据类型,是描述非结构化数据的一种重要方式。

5.2.1 图网络表示

常用 $G(V, E)$ 表示图网络, 其中 V 是由表示实体的节点构成的集合, E 是由表示节点之间关系的边构成的集合, 存在边连接的两个节点 v_i, v_j 用 $\langle v_i, v_j \rangle \in E$ 表示。图网络在现实生活中是广泛存在的, 可用于不同场景的数据建模与分析。

例 5.1 (社交网络) 20 世纪 70 年代, 美国社会学家 Zachary 观察了一家空手道俱乐部 34 名成员之间的交往情况。每名成员用一个节点来表示, 如果两名成员之间是交往频繁的朋友关系, 那么就在两个节点之间用一条边连接, 如图 5-2 所示。

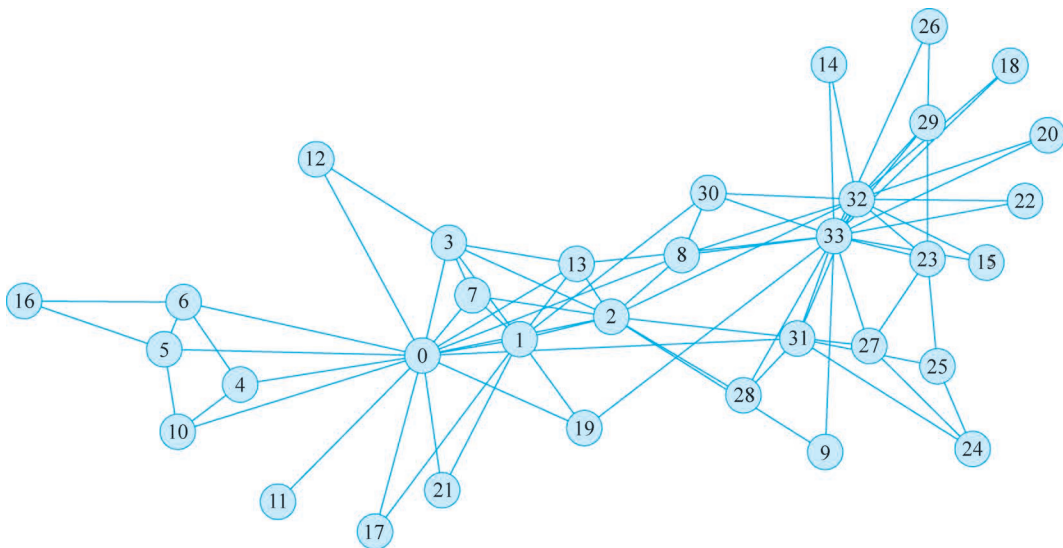
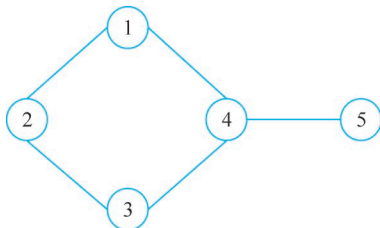


图 5-2 空手道俱乐部成员的社交网络

通过构造上述图网络, 可以模拟和分析现实世界中的复杂系统。但要在计算机存储或处理上述图网络, 还需要借助描述图网络中各节点之间连接关系的邻接矩阵。假设 $G(V, E)$ 为具有 n 个节点的图, 其邻接矩阵 $A = (a_{ij})_{n \times n}$ 定义为

$$a_{ij} = \begin{cases} 1, & \langle v_i, v_j \rangle \in E \\ 0, & \langle v_i, v_j \rangle \notin E \end{cases} \quad (5-1)$$

例 5.2 如图 5-3(a) 所示的图的邻接矩阵如图 5-3(b) 所示。



(a) 图

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

(b) 邻接矩阵

图 5-3 图及其邻接矩阵

5.2.2 网络结构分类

现实生活中可利用图网络表示的数据类型十分丰富。根据不同的标准, 可以将图进行不

同的划分。根据图网络中边是否有方向,可以将图网络分成有向图和无向图。根据图网络中边的连接方式,可以划分为 0-1 网络、加权网络、符号网络、双模网络和动态网络。根据图网络数据的表示模型,可以划分为 E-R 随机图模型、小世界网络和无标度图网络等。

1. 基于边方向的划分

1) 无向图

边没有方向的图称为无向图。如果 $\langle v_i, v_j \rangle \in E$, 那么同时有 $\langle v_j, v_i \rangle \in E$ 。因此, 无向图的邻接矩阵是对称的。例 5.1 中的社交网络是一个无向图。

2) 有向图

边有方向的图称为有向图。如果从节点 v_i 到节点 v_j 有一条边相连, 那么 $\langle v_i, v_j \rangle \in E$, 在邻接矩阵中 $a_{ij} = 1$ 。如果从节点 v_j 到节点 v_i 有没有边相连, 那么相应的在邻接矩阵中 $a_{ji} = 0$ 。因此在有向图中, 邻接矩阵不一定是对称的。

2. 基于边连接方式的划分

1) 0-1 网络

0-1 网络又称为无权网络。网络中的边仅表示相应的两个节点存在某种特殊的关系, 比如两个节点是朋友关系, 或者节点之间存在交易。0-1 网络可以由邻接矩阵完全表示, 矩阵中的元素取值为 0 或 1。

2) 加权网络

加权网络不仅能够反映节点之间是否存在联系, 还能反映节点之间联系的强弱。在加权网络中, 可以定义权重矩阵。给定一个有 n 个节点的加权网络 $G(V, E)$, 其权重矩阵为 $W = (w_{ij})_{n \times n}$, 如果节点 v_i 到节点 v_j 之间有边连接, 则 w_{ij} 为连接的边的权重; 如果节点 v_i 到节点 v_j 之间没有边连接, 那么 $w_{ij} = 0$ 。

3) 符号网络

符号网络是指边具有正或负符号属性的网络, 其中正号表示积极关系, 用“+”标识, 负号表示消极关系, 用“-”标识。符号网络可以被看作一种特殊的加权网络, 广泛存在于社会学、生物学和信息学等领域中。

4) 双模网络

双模网络是定义在两种不同类型节点上的网络, 且关系仅存在于不同类型的节点之间。比如我们要使用图网络结构表示图书馆图书的借阅关系, 读者和图书属于不同类型的节点。同一种类型的节点之间不会产生关系, 只有不同类型的节点之间可能存在边相连, 如图 5-4 所示。

5) 动态网络

动态网络中网络结构会随着时间的变化而变化, 包括节点数量的变化、边的变化以及边的权重的变化等。日常生活中存在的许多网络结构数据都具有丰富的时间信息, 节点之间的联系随着时间而动态演变。

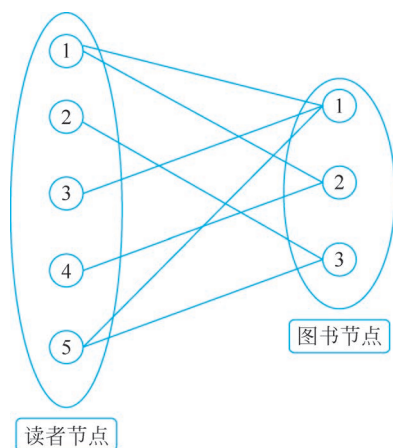


图 5-4 双模网络示意图

3. 基于表示模型的划分

1) E-R 随机图

E-R 随机图是由著名的数学家 Erdős 和 Rényi 提出的随机图模型生成的随机图。在这个模型中, 给定 n 个节点 $\{v_i, i = 1, 2, \dots, n\}$, 每两个节点 v_i, v_j 之间以概率 p 连接一条无向

边,而且任意两点之间存在边的概率都相互独立。

2) 小世界网络

小世界网络,就是相对于同等规模节点的随机网络,具有较短的平均路径长度和较大的聚类系数特征的网络模型。可以用来很好地建模真实网络的聚集性的特点。在现实世界的人际交往中,任意两个人只需要通过少数的几个中间人就能相互认识,具有小世界网络的特点。

3) 无标度网络

在一个包含多个节点的网络中,每个节点连接的边的数量差别很大。如果图网络中的节点的度满足幂律分布,即各节点之间的连接状况具有严重的不均匀分布性,网络中少数节点拥有极其多的连接,而大多数节点只有很少量的连接,这样的网络就称为无标度网络。幂律分布广泛地存在于物理学、计算机科学、社会科学等众多领域之中,使得无标度网络在对现实世界的建模中具有重要的意义。BA 模型(Barabási-Albert Model)是一种生成无标度网络的重要方法,其流程如下:

(1) 初始化 n_0 个相互连通的节点;

(2) 增加一个节点 v , 这个新增加的点与已有的节点 u 以概率 $p(u, v) = \frac{\deg(u)}{\sum_i \deg(i)}$ 生成一条无向边, 其中 $\deg(u)$ 表示网络中与节点 $u \in V$ 有边连接的节点的数量;

(3) 更新图网络中节点的度,不断重复步骤(2),直到生成足够丰富的节点。

5.2.3 网络描述性统计

接下来介绍常用的网络描述性统计量。

1. 节点的度

给定一个 n 个节点的无向图 $G(V, E)$, 如果邻接矩阵为 A , 节点 v_i 的度定义为

$$\deg(v_i) = \sum_{j=1}^n a_{ij} \quad (5-2)$$

表示与节点 v_i 通过边相连接的节点数量。

对于邻接矩阵为 A 的 n 个节点的有向图 $G(V, E)$, 每个节点的度包括入度和出度。节点 v_i 的入度表示从其他节点连向节点 v_i 的数量, 其定义为

$$\deg_+(v_i) = \sum_{j=1, j \neq i}^n a_{ji} \quad (5-3)$$

节点 v_i 的出度表示从节点 v_i 连向其他节点的数量, 其定义为

$$\deg_-(v_i) = \sum_{j=1, j \neq i}^n a_{ij} \quad (5-4)$$

2. 节点的中心性

节点的中心性是衡量网络中节点重要性的指标,它反映了节点在网络结构中的地位和作用,例如社交网络中最有影响力的人,互联网或城市网络中的关键基础设施节点以及疾病的超级传播者。常见的节点中心性度量指标包括度中心度、紧密中心度、介数中心度等。

1) 度中心度

度中心度(Degree Centrality)是通过与节点有边连接的邻居节点数目来计算节点的重要性。一个节点的邻居越多,显然该节点的重要性越强。对节点的度进行归一化,可以得到度中心度的定义。给定一个包含 n 个节点的无向图 $G(V, E)$, 节点 v_i 的度中心度为

$$C_d(v_i) = \frac{\deg(v_i)}{n-1} \quad (5-5)$$

其中, $0 \leq C_d(v_i) \leq 1$ 。

2) 紧密中心度

距离可以用来衡量两个非邻居节点之间的接近程度。如果节点 v_i 与其他节点的距离都很小,那么该节点越接近图的中心。这一特性可以用紧密中心度(Closeness)来衡量。其定义如下:

$$C_c(v_i) = \sum_{j \neq i} \frac{1}{d(v_j, v_i)} \quad (5-6)$$

其中, $d(v_j, v_i)$ 表示节点 v_i 到节点 v_j 之间的距离,可以通过迪杰斯特拉算法(Dijkstra's Algorithm)、贝尔曼-福特算法(Bellman-Ford Algorithm)和 A* 搜索算法等计算。

3) 介数中心度

在图网络中,两个非邻居节点需要通过其他节点进行连通。一个节点 v_i 充当中介作用的次数越高,说明其扮演桥梁作用的重要性越高。假如这个节点 v_i 消失了,那么其他节点之间的连接甚至可能断开。介数中心度(Betweenness Centrality)可以用来衡量节点在扮演桥梁作用中的重要性,其定义为

$$C_b(v_i) = \sum_{j, k \neq i} \frac{\sigma_{jk}(v_i)}{\sigma_{jk}} \quad (5-7)$$

其中, σ_{jk} 表示从节点 v_j 到节点 v_k 之间所有的路径的数量, $\sigma_{jk}(v_i)$ 表示其中途径节点 v_i 的路径数量。

3. 网络密度

在一个包含 n 个节点的网络 $G(V, E)$ 中,最多有 C_n^2 条边。如果节点之间都存在相连的边,那么这个上限就可以达到,此时的图网络就会显得稠密;如果只有较少的节点之间存在连接,此时的图网络就会显得稀疏。用网络密度来衡量,其定义为

$$\Delta = \frac{|E|}{C_n^2} \quad (5-8)$$

其中 $|E|$ 表示图 $G(V, E)$ 中的边数,式(5-9)表示在所有可能的连接中,有多大比例的边是实际存在的。在许多实际的问题中,网络的密度通常都是很低的。

例 5.3(合作者网络) 现在的科学研究通常需要多名研究者通力合作共同参与来完成。不同领域的合作者的规模可能会存在一定的区别。可以用图网络模型来建模科学研究合作者网络,网络中的节点代表一位学者,如果两位学者合作发表过学术论文,那么就在代表他们的节点之间连接一条边。arxiv ASTRO-PH(天体物理学)协作网络是基于提交到 arxiv 中天体物理学类别的论文建立的科学合作者网络模型,包括 18 772 个节点和 198 110 条边,且:

(1) 该合作者网络的密度为 0.11%,每个节点平均与 21.1 个节点有边连接。

(2) 可以借助 Python 库 networkx 计算该合作者网络的各节点的度等描述性统计指标。其中节点度的分布如图 5-5 所示。图中横坐标表示节点的度,纵坐标表示相应的节点数量。大多数作者的合作者都很少,只有少量的节点的度很大,与很多节点都存在连接。

(3) 如图 5-6 展示了合作者网络节点的度与度中心度、介数中心度和紧密中心度等指标之间的相关关系。其中横坐标表示节点的度,纵坐标表示节点的其他统计特性。可见节点的度与度中心度呈现严格的线性关系。

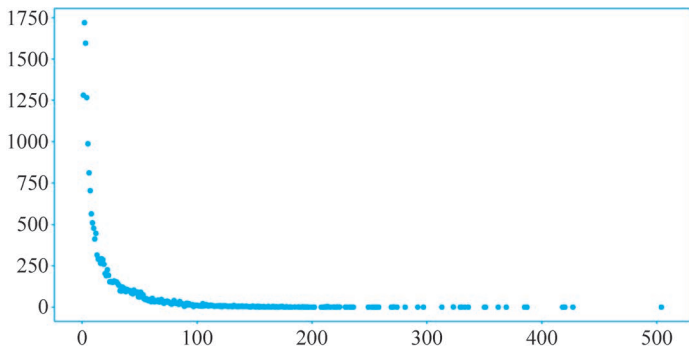


图 5-5 arxiv ASTRO-PH 合作者网络中节点度的统计

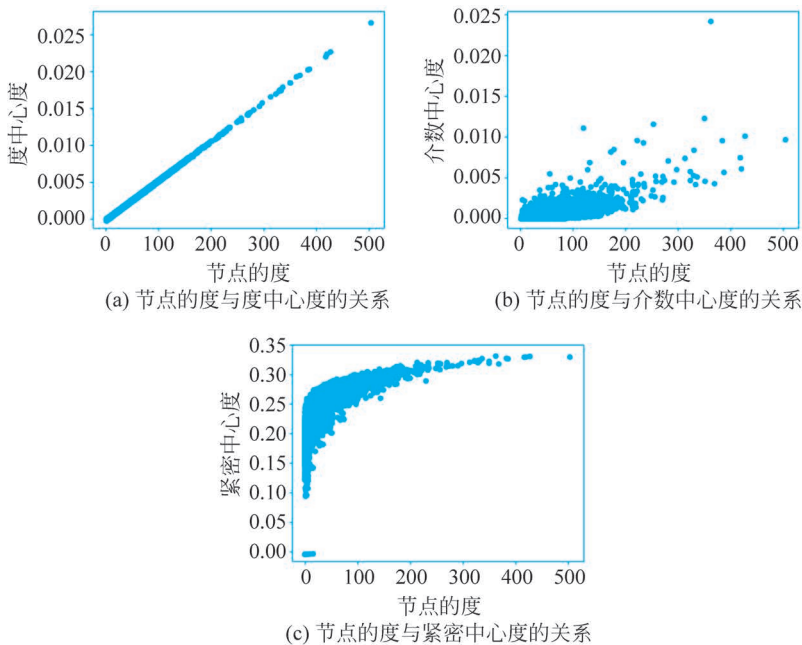


图 5-6 arxiv ASTRO-PH 合作者网络中节点的度与其他统计特征的散点图

5.3 图基础算法

前述介绍了图的基本概念、图的表示及描述性统计,本节主要介绍图的基本算法:图遍历和图分割。

5.3.1 图遍历

图遍历(Traversing Graph)是指从图中的某一节点出发,按照某种搜索方法沿着图中的边对图中的所有节点访问一次且仅访问一次。它是图的基本运算,在求解图的连通性问题、拓扑排序、关键路径和最短路径等算法中具有重要的作用。图遍历算法包括广度优先搜索(Breadth First Search,BFS)和深度优先搜索(Depth First Search,DFS)。

1. 广度优先搜索

给定图 $G(V, E)$ 和起始节点 v_0 的情况下,广度优先搜索算法从初始点 v_0 开始,依次访问

与其相邻的所有节点,然后由近及远,逐一访问相邻节点的相邻节点。通过广度优先搜索算法可以生成一棵根为 v_0 且包括 v_0 所有可达节点的广度优先树。

例 5.4 给定一个网络,如图 5-7 所示。请给出以节点 1 为起点的广度优先搜索结果。

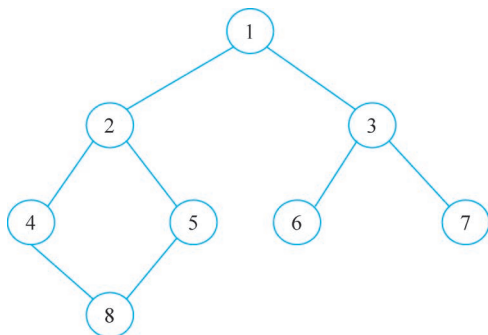


图 5-7 图网络示意图

根据广度优先搜索算法思想,有:

(1) 从节点 1 出发,首先搜索到它的相邻节点 2 和 3;

(2) 接下来搜索到节点 2 的相邻节点 4 和 5;

(3) 接下来搜索到节点 3 的相邻节点 6 和 7;

(4) 最后搜索到与节点 4 相邻的节点 8。此时所有的节点都已搜索到,于是得到从节点 1 出发的基于广度优先搜索算法的路径为

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8$$

2. 深度优先搜索

对于最新发现的节点 v ,如果它还有以此为起点而未搜索的边,就沿此边继续搜索下去。当以节点 v 为起点的所有边都已被探寻过,搜索将回溯到发现节点 v 的那条边的始节点。这一过程一直进行到已发现从源节点可达的所有节点为止。如果还存在未被发现的节点,则选择其中一个作为源节点并重复以上过程,整个过程反复进行直到所有节点都被发现为止。

例 5.5 考虑例 5.4 中的图网络,请给出以节点 1 为起点的深度优先搜索结果。

根据深度优先搜索算法思想,有:

(1) 从节点 1 出发,访问节点 1。

(2) 访问节点 1 后,访问第一个与节点 1 相连且未被访问的节点 2。之后,以节点 2 为新节点,继续搜索,依次访问节点 4、8、5。

(3) 搜索从节点 5 依次溯回至节点 8、4、2、1。

(4) 由于与节点 1 相连的节点 3 未被访问,访问节点 3,并从节点 3 出发依次访问节点 6、7。至此所有的节点都被搜索,于是算法终止。最终得到从节点 1 出发基于深度优先搜索算法的路径为

$$1 \rightarrow 2 \rightarrow 4 \rightarrow 8 \rightarrow 5 \rightarrow 3 \rightarrow 6 \rightarrow 7$$

3. Dijkstra 最短路径算法

Dijkstra 算法是一种基于贪心策略的路径寻优算法,主要用于求解图论中的单源最短路径问题。对给定的加权有向图 $G(V, E)$ 和源点 v_0 ,将所有节点划分成 S 与 $V-S$ 两个集合, S 中存放已找到的到 v_0 最短路径的节点(初始只包含源点 v_0), $V-S$ 中存放未找到到 v_0 最短路径的节点(初始为 $V-\{v_0\}$),然后计算 v_0 到 $V-S$ 中各节点最短路径的长度。若 S 中的节点与 $V-S$ 中节点 u 有边相连,则 v_0 到 u 的最短路径长度 $l(u) = \min_{v_i \in S} w(v_i, u) + l(v_i)$,

其中 $w(v_i, u)$ 为连接节点 v_i 与 u 的边的权重或长度(若 v_i 与 u 不相连, 则 $w(v_i, u) = \infty$; 若 S 中的节点与 u 没有边相连, 则 $l(u) = \infty$)。在此基础上, 将 v_0 到 $V-S$ 中路径长度最短的节点加入集合 S , 直到所有节点都加入集合 S 。Dijkstra 算法描述如表 5-1 所示。

表 5-1 Dijkstra 算法

-
- | |
|---|
| (1) 初始化 S 与 $V-S$, 其中 $S = \{v_0\}, V-S = V \setminus \{v_0\}$ |
| (2) 计算 v_0 到 $V-S$ 中各节点最短路径的长度。将 v_0 到 $V-S$ 中路径长度最短的节点 u 添加到集合 S 中, 并从 $V-S$ 中删除节点 u |
| (3) 重复步骤(2), 直到 $V-S$ 集合为空 |
-

例 5.6 考虑如图 5-8 中的网络, 请给出起点 A 到其他各节点的最短路径。

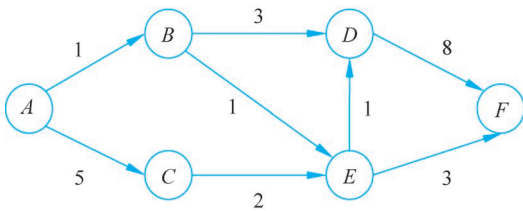


图 5-8 图网络示意图

根据 Dijkstra 算法, 有:

(1) 初始化 S 与 $V-S$, 令 $S = \{A(0)\}, V-S = \{B(1), C(5), D(\infty), E(\infty), F(\infty)\}$, 其中 $B(1)$ 表示当前起点 A 到节点 B 的最短路径长度为 1, $D(\infty)$ 表示当前起点 A 到节点 D 的最短路径长度为 ∞ , 其余类似。

(2) 将起点 A 到 $V-S$ 中路径长度最短的节点 B 添加到 S 中, 更新 S 、 $V-S$ 和节点 A 到 $V-S$ 中各节点最短路径的长度, 得 $S = \{A(0), B(1)\}, V-S = \{C(5), D(4), E(2), F(\infty)\}$ 。

(3) 将起点 A 到 $V-S$ 中路径长度最短的节点 E 添加到 S 中, 更新 S 、 $V-S$ 和节点 A 到 $V-S$ 中各节点最短路径的长度, 得 $S = \{A(0), B(1), E(2)\}$, 更新集合 $V-S = \{C(5), D(3), F(5)\}$ 。

(4) 将起点 A 到 $V-S$ 中路径长度最短的节点 D 添加到 S 中, 更新 S 、 $V-S$ 和节点 A 到 $V-S$ 中各节点最短路径的长度, 得 $S = \{A(0), B(1), E(2), D(3)\}, V-S = \{C(5), F(5)\}$ 。

(5) 由起始点到 C 与 F 的最短路径相等, 可随机挑选一个加入集合 S , 这里以 C 为例, 将其加入集合 S , 得 $S = \{A(0), B(1), E(2), D(3), C(5)\}, V-S = \{F(5)\}$ 。

(6) 将 $V-S$ 中剩余的 F 加入 S 中, 得 $S = \{A(0), B(1), E(2), D(3), C(5), F(5)\}$, 据此及算法过程可得起点 A 到各个节点的最短路径及长度。

5.3.2 图分割

令 $G(V, E)$ 表示一个加权无向图, 其权重矩阵为 $\mathbf{W}$ 。图分割是将图 G 划分成 2 个规模相同的子网络 G_1 和 G_2 。定义一个变量

$$p_i = \begin{cases} 1, & v_i \in G_1 \\ -1, & v_i \in G_2 \end{cases} \quad (5-9)$$

连接 G_1 中的节点与 G_2 中的节点的所有边的权重总和被称为“割”, 有

$$\text{cut}(G_1, G_2) = \sum_{v_i \in G_1, v_j \in G_2} w_{ij}$$

$$\begin{aligned}
&= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} (p_i - p_j)^2 \\
&= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} (p_i^2 - 2p_i p_j + p_j^2) \\
&= \frac{1}{2} \left(\sum_{i=1}^n \sum_{j=1}^n -2w_{ij} p_i p_j + \sum_{i=1}^n \sum_{j=1}^n 2w_{ij} \right) \\
&= \mathbf{p}^T (\mathbf{D} - \mathbf{W}) \mathbf{p}
\end{aligned} \tag{5-10}$$

其中, $\mathbf{p} = [p_1, p_2, \dots, p_n]^T$, $\mathbf{D}$ 是一个对角矩阵, 对角线上的元素为 $d_{ii} = \sum_{j=1}^n w_{ij}$, 也称为角度矩阵。定义半正定对称矩阵 $\mathbf{L} = \mathbf{D} - \mathbf{W}$, 称为拉普拉斯矩阵, 则图分割可以求解如下优化问题实现:

$$\begin{aligned}
&\min \quad \mathbf{p}^T \mathbf{L} \mathbf{p} \\
&\text{s. t.} \quad p_i^2 = 1, i = 1, 2, \dots, n
\end{aligned} \tag{5-11}$$

优化问题式(5-11) 是一个组合优化问题, 属于 NP 难问题。为解决该问题, 一种有效的方法是对该优化问题的约束条件进行松弛, 对应的优化问题可写为

$$\begin{aligned}
&\min \quad \mathbf{p}^T \mathbf{L} \mathbf{p} \\
&\text{s. t.} \quad \mathbf{e}^T \mathbf{p} = 0 \\
&\quad \mathbf{p}^T \mathbf{p} = n
\end{aligned} \tag{5-12}$$

其中, $\mathbf{e} = [1, 1, \dots, 1]^T$ 。 $\mathbf{L}$ 的最小特征值为 0, 此时对应的特征向量为 $\mathbf{e}$, 以 $\mathbf{e}$ 作为优化问题的解向量 $\mathbf{p}$ 不满足约束条件 $\mathbf{e}^T \mathbf{p} = 0$, 于是最优解应该在第二小特征值对应的特征向量 $\mathbf{p}_1$ 处。根据 $\mathbf{p}_1$ 的分量的正负性实现节点的划分。将特征向量分量值为正的节点划分到子图 G_1 , 特征向量分量值为负的节点划分到子图 G_2 , 从而实现网络的分割。

5.4 社区发现

在许多图网络结构数据中, 网络密度是很低的, 节点之间存在边的连接并不是随机的。比如在银行交易网络中, 亲戚朋友的银行账户之间相互转账交易是常发生的, 陌生人之间的转账交易则是很偶然的。图数据分析的一个重要任务是发现网络结构中节点的聚集特性, 即探索发现网络中存在的不同社区。

所谓社区, 是图网络中的一个子图。社区内部的节点与节点之间的连接很紧密, 而与其他社区的节点之间的连接比较稀疏。社区发现(Community Detection)是根据网络中节点之间的关系将节点划分到某一个社区之中, 在本质上是一个节点聚类的问题。社区发现算法主要包括 Girvan-Newman 算法、谱方法和 Louvain 算法等。

5.4.1 模块度

模块度(Modularity)是用来衡量一个社区划分结果性能的定量评价指标, 由 Newman 和 Girvan 于 2004 年首先提出来的。社区划分的结果, 应该使得社区内部节点之间连接紧密, 而社区之间的节点连接稀疏。模块度正是基于这一思想而定义的, 它是指网络中连接社区结构内部节点的边所占的比例减去另外一个与原网络结构度数分布一致的随机网络中连接社区结构内部节点的边所占比例的期望值。其严格的数学定义如下。

定义 5.1(模块度) 设图 G 包含 n 个节点和 m 条边, 其邻接矩阵为 $\mathbf{A}$, k_i 为节点 v_i 的度, 给定图 G 的社区结构 C , 社区结构的模块度可通过下式计算:

$$Q = \frac{1}{2m} \sum_{i,j} \left(a_{ij} - \frac{k_i k_j}{2m} \right) \delta(C_i, C_j) \quad (5-13)$$

其中, C_i 和 C_j 分别表示节点 v_i 和 v_j 所在的社区, 且有

$$\delta(C_i, C_j) = \begin{cases} 1, & C_i = C_j \\ 0, & C_i \neq C_j \end{cases}$$

模块度度量了社区结构情况, 值越大表明这种划分的社区结构越好, 可用于评价不同社区发现算法的优劣。式(5-14)只计算节点 v_i 和 v_j 属于同一个社区的情况, 如果两个节点不在同一个社区则忽略不计。因此, 模块度可以改写为

$$\begin{aligned} Q &= \frac{1}{2m} \sum_{i,j} \left(a_{ij} - \frac{k_i k_j}{2m} \right) \delta(C_i, C_j) \\ &= \frac{1}{2m} \sum_{c \in C} \sum_{v_i \in c} \sum_{v_j \in c} \left(a_{ij} - \frac{k_i k_j}{2m} \right) \end{aligned} \quad (5-14)$$

其中, $c \in C$ 是图 G 的一个社区。从式(5-15) 更容易理解模块度的计算思路。如果两个节点 v_i 和 v_j 之间有一条边, 而且这两个节点在同一个社区内部, 那么它们偏向于对模块度有积极的贡献; 反过来, 如果这两个节点间没有边, 而且这两个节点在同一个社区内部, 无论 $\frac{k_i k_j}{2m}$ 大小如何, 它们对模块度都会有负的贡献。

例 5.7 如图 5-9 所示, 图中有 6 个节点, 其中

$$m = 7, \quad k_1 = 2, \quad k_2 = 2, \quad k_3 = 3,$$

$$k_4 = 3, \quad k_5 = 2, \quad k_6 = 2$$

社区结构为 $C = \{\{1, 2, 3\}, \{4, 5, 6\}\}$, 计算该社区结构的模块度。

根据模块度的定义, 图 5-9 中社区结构的模块度为

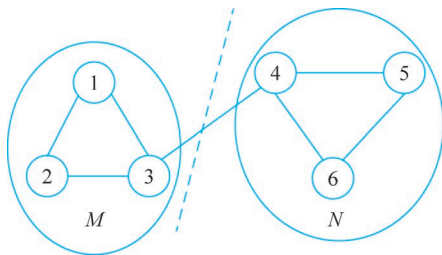


图 5-9 图网络结构示意图

$$\begin{aligned} Q_1 &= \frac{1}{2m} \sum_{i,j} \left(a_{ij} - \frac{k_i k_j}{14} \right) \delta(C_i, C_j) \\ &= \frac{1}{14} \left[\left(0 - \frac{k_1 k_1}{14} \right) + 2 \left(1 - \frac{k_1 k_2}{14} \right) + 2 \left(1 - \frac{k_1 k_3}{14} \right) + \right. \\ &\quad \left(0 - \frac{k_2 k_2}{14} \right) + 2 \left(1 - \frac{k_2 k_3}{14} \right) + \left(0 - \frac{k_3 k_3}{14} \right) + \\ &\quad \left(0 - \frac{k_4 k_4}{14} \right) + 2 \left(1 - \frac{k_4 k_5}{14} \right) + 2 \left(1 - \frac{k_4 k_6}{14} \right) + \\ &\quad \left. \left(0 - \frac{k_5 k_5}{14} \right) + 2 \left(0 - \frac{k_5 k_6}{14} \right) + \left(0 - \frac{k_6 k_6}{14} \right) \right] \\ &= \frac{5}{14} \end{aligned}$$

为了了解不同社区结构对模块度的影响, 请看下面的例 5.8。

例 5.8 考虑例 5.7 的图网络,使用不同的社区结构划分该网络。令其社区结构为 $C = \{\{1, 2\}, \{3, 4, 5, 6\}\}$, 即节点 1 和节点 2 属于一个社区, 节点 3、4、5、6 属于另一个社区, 如图 5-10 所示。计算该社区结构的模块度。

根据模块度的定义, 图 5-10 中社区结构的模块度为

$$Q_2 = \frac{1}{2m} \sum_{c \in C} \sum_{i \in c} \sum_{j \in c} \left(a_{ij} - \frac{k_i k_j}{2m} \right) = \frac{6}{49}$$

通过分析例 5.7 和例 5.8, $Q_2 < Q_1$, 调整节点 3 所属社区会导致模块度降低, 从而说明社区结构 $\{\{1, 2, 3\}, \{4, 5, 6\}\}$ 比社区结构 $\{\{1, 2\}, \{3, 4, 5, 6\}\}$ 效果更好。

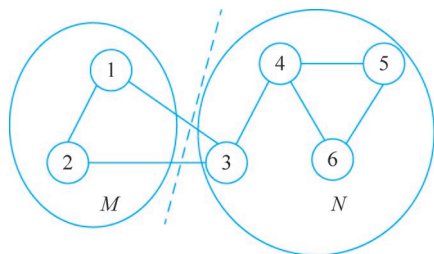


图 5-10 图网络结构示意图

5.4.2 GN 算法

传统的社区发现算法主要基于层次聚类的思想。首先将所有的节点当作一个社区, 每次使用算法将一个社区划分为两个社区, 不断地迭代, 实现自顶向下的方式发现图网络中的不同社区。

GN(Girvan-Newman)算法是由 Girvan 和 Newman 于 2002 年提出的社区发现领域的一个重要算法, 具有开拓性的意义。它的基本想法是, 在一个网络之中, 通过社区内部的边的最短路径相对较少, 而通过社区之间的边的最短路径的数目则相对较多。如果能够将连通不同社区的边删除, 那么就可以很自然地将图网络划分成不同的社区。为此需要衡量边在网络中的重要性。

给定一个图网络 $G(V, E)$, $(v_i, v_j) \in E$ 的边介数(Edge Betweenness)是指网络中所有最短路径中经过该边的路径的数目占最短路径总数的比例。边介数越大, 边在网络中的枢纽性越强。基于图网络的边介数, GN 算法的步骤如下:

- (1) 计算图网络 G 中每一条边的边介数;
- (2) 删除具有最高边介数的边;
- (3) 重新计算剩余所有边的边介数;
- (4) 重复步骤(2)和步骤(3), 直到形成既定的社区数量时停止。

从而产生一种类似层次聚类的效果, 形成一棵自上而下的树。GN 算法在每次删除边之后都会重复计算剩余节点的边介数, 从而具有较高的计算复杂度, 难以在大规模图网络学习中应用。同时 GN 算法的终止条件缺乏一个明确的衡量目标, 分裂过程可以持续进行下去直到每个节点都组成一个独立的社区。

例 5.9 考虑例 5.1 中空手道俱乐部的社交网络。

(1) 使用 GN 算法将该网络进行划分, 当选择划分成 2 个社区结构时, 划分结果如图 5-11 所示。可以很直观地观察到该俱乐部中存在分别以 0 号员工和 33 号员工为核心的 2 个小团体。

(2) 模块度是衡量社区发现结果的重要评价指标。当划分的社区数量和社区结构不同时, 划分结果具有不同的模块度, 如图 5-12 所示。从中可见, 将社区划分成 4 个子社区时, 具有最大的模块度。

(3) 根据最大化模块度的要求, 可以将该网络划分成 4 个社区, 如图 5-13 所示。

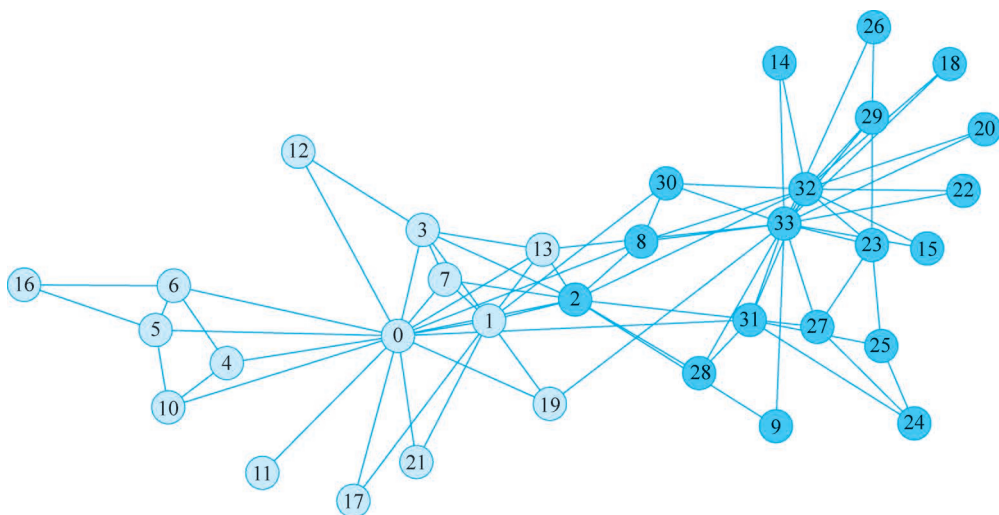


图 5-11 2 个社区的社区发现可视化

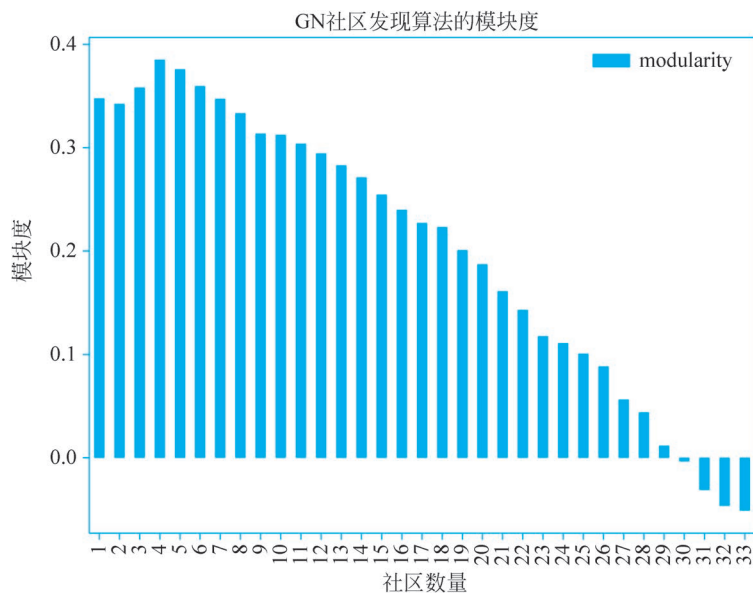


图 5-12 划分成不同社区数量的模块度

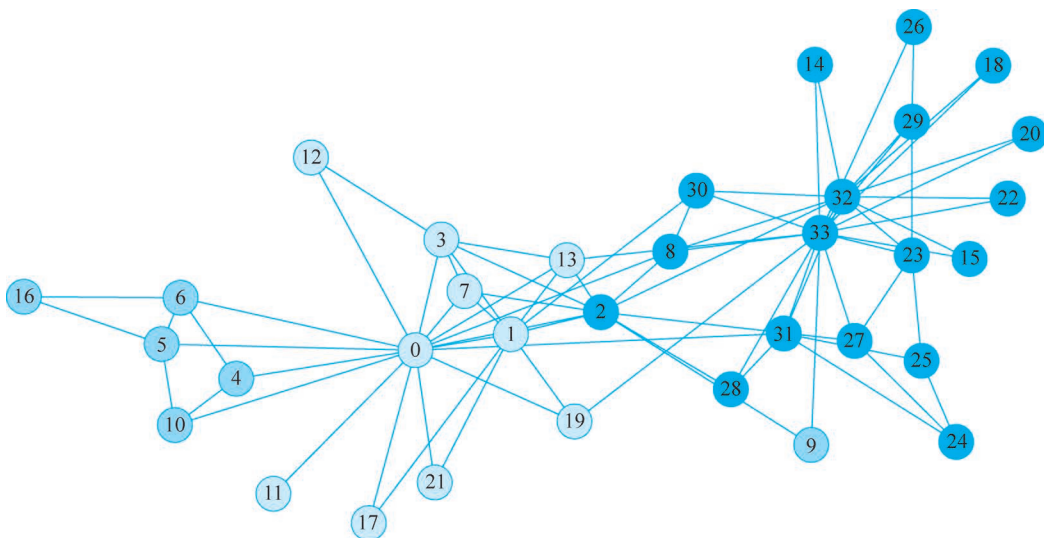


图 5-13 模块度最大化的 GN 算法的社区发现结果可视化

5.4.3 谱方法

在 5.3.2 节中,介绍了基于拉普拉斯矩阵的图分割算法,实现了将一个大规模的图划分成两个子图(社区)的目的。如果将这个过程继续进行下去,每次选择一个社区进行划分,直到满足要求为止。由于模块度能够较好地衡量社区划分的性能,因此也可以考虑使用模块度矩阵替换拉普拉斯矩阵,称为社区发现的谱方法。

把一个图 $G(V, E)$ 划分成 2 个社区 G_1 和 G_2 时,根据式(5-10)定义的变量 p_i ,以及模块度的定义,得到

$$\begin{aligned}
 Q &= \frac{1}{4m} \sum_{i,j} \left(a_{ij} - \frac{k_i k_j}{2m} \right) (p_i p_j + 1) \\
 &= \frac{1}{4m} \sum_{i,j} \left(a_{ij} - \frac{k_i k_j}{2m} \right) p_i p_j \\
 &= \frac{1}{4m} \mathbf{p}^T \mathbf{B} \mathbf{p}
 \end{aligned} \tag{5-15}$$

其中, $\mathbf{B} = (b_{ij})$ 称为模块度矩阵, $b_{ij} = a_{ij} - k_i k_j / 2m$ 。注意到在模块度矩阵 $\mathbf{B}$ 中,它的每一行和每一列之和都为 0,所以向量 $\mathbf{e} = (1, 1, \dots, 1)^T$ 为特征值 0 对应的特征向量。因此可以通过最大化一次划分的模块度来找到当前最优的划分,即

$$\begin{aligned}
 \max \quad & \mathbf{p}^T \mathbf{B} \mathbf{p} \\
 \text{s. t.} \quad & \mathbf{e}^T \mathbf{p} = 0 \\
 & \mathbf{p}^T \mathbf{p} = 1
 \end{aligned} \tag{5-16}$$

结合矩阵计算的知识,可以得到优化问题式(5-17)的最优值为模块度矩阵 $\mathbf{B}$ 的最大特征值,解向量 $\mathbf{p}$ 为其对应的单位特征向量。由此得到图网络中每一个节点的划分规则:

- (1) 如果解向量 $\mathbf{p}$ 的第 i 个元素 $p_i > 0$,那么节点 v_i 划分为社区 G_1 ;
- (2) 如果解向量 $\mathbf{p}$ 的第 i 个元素 $p_i < 0$,那么节点 v_i 划分为社区 G_2 。

图可能有多个社区,为了进一步实现更细致的社区结构,简单的做法是将每个划分出来的子社区再一分为二。但是这种做法并不合理,这是因为当划分一个子社区时,有些边可能会被

删除,最终整个图的社区结构的模块度会发生变化。每次最大化模块度,最终会导致一个错误的结果。

取而代之的是,每次划分一个子社区时,仅希望最大化模块度的增量 ΔQ 。假设一个子社区 $g = G_i, i = 1, 2$ 的大小为 n_g , 希望将社区 g 一分为二, 那么其模块度的增量 ΔQ 可以计算如下:

$$\begin{aligned}
 \Delta Q &= \frac{1}{2m} \left[\frac{1}{2} \sum_{i,j \in g} b_{ij} (p_i p_j + 1) - \sum_{i,j \in g} b_{ij} \right] \\
 &= \frac{1}{4m} \left(\sum_{i,j \in g} b_{ij} p_i p_j - \sum_{i,j \in g} b_{ij} \right) \\
 &= \frac{1}{4m} \sum_{i,j \in g} (b_{ij} - \delta_{ij} \sum_{k \in g} b_{ik}) p_i p_j \\
 &= \frac{1}{4m} \sum_{i,j \in g} (b_{ij} - \delta_{ij} \sum_{k \in g} b_{ik}) p_i p_j \\
 &= \frac{1}{4m} \mathbf{p}^T \mathbf{B}^{(g)} \mathbf{p}
 \end{aligned} \tag{5-17}$$

其中, $\frac{1}{2} \sum_{i,j \in g} b_{ij} (p_i p_j + 1)$ 和 $\sum_{i,j \in g} b_{ij}$ 分别表示子社区划分后和划分前模块度的大小, 当 $i = j$ 时 $\delta_{ij} = 1$, 否则其值为 0。因此, 只有当每次社区划分导致模块度增量 $\Delta Q > 0$ 时才进一步划分这个子社区, 否则这个子社区就不会再进一步划分。因此, 每次划分都会保证整个社区结构的模块度是在不断增加的。

基于模块度优化的社区发现算法思想是将社区发现问题转化为优化问题, 其优化目标是最大化整个社区结构的模块度。由于社区可以是图中任意节点的组合, 实践上很难通过枚举法找到模块度最大的社区划分或社区结构, 因此, 通常使用近似算法来获得模块度最大的社区结构。

5.5 GraphScope 简介

GraphScope 是阿里巴巴智能实验室研发并开源的一站式大规模图计算平台, 致力于解决实际生产场景中所涉及的图计算问题。它具有高效的跨引擎内存管理, 在业界首次支持 Gremlin 分布式编译优化, 同时支持算法的自动并行化和自动增量化处理动态图更新, 提供了企业级场景的极致性能。目前, GraphScope 已在包括风控、电商推荐、知识图谱和网络安全在内的多个互联网领域得到成功应用。接下来, 本节将介绍 GraphScope 的系统架构、重要特性和应用场景等。

1. 系统架构

GraphScope 开发的宗旨在于处理和分析海量的图结构数据。要实现以上目标, 需要多种计算组件进行交互, 包括集群管理软件、分布式执行引擎以及开发工具和算法库等。由于整个系统庞大且功能复杂, 在此主要对其中的算法层、引擎层和存储层进行介绍。图 5-14 给出了 GraphScope 的系统架构。

(1) 算法层。GraphScope 构建了丰富的图分析算法, 包括聚类算法、图挖掘算法和图模式匹配算法等; 此外还提供了丰富的图学习算法, 包括基于神经网络的算法与基于节点嵌入的算法。用户可以很方便地利用 GraphScope 提供的 Python 接口来调用相应的算法处理

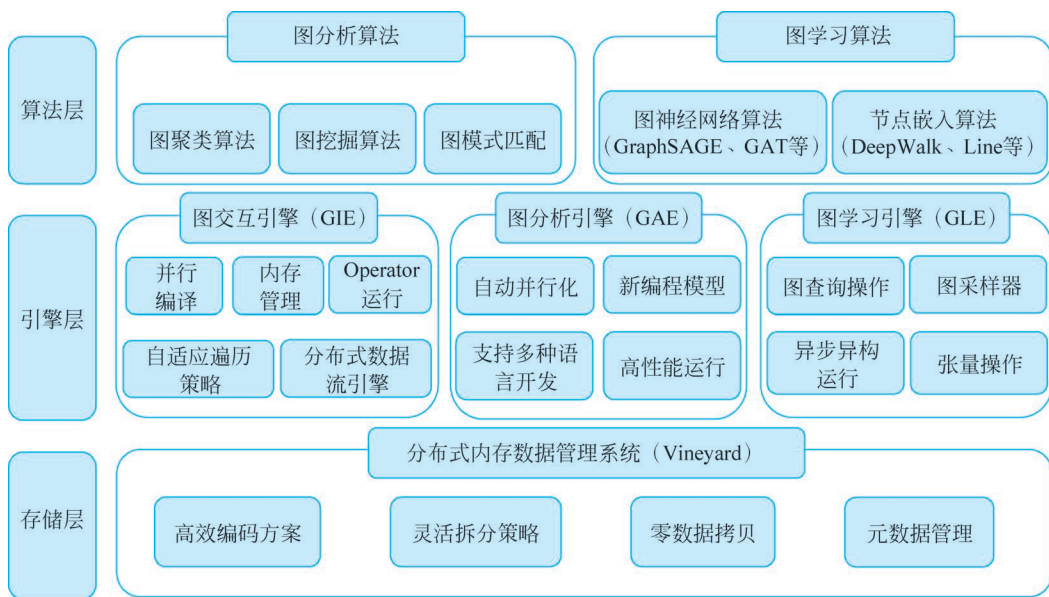


图 5-14 GraphScope 的系统架构

图分析任务。

(2) 引擎层。GraphScope 运行时由图交互引擎、图分析引擎和图学习引擎组成，分别负责提供存储和管理图数据的能力、执行各种图算法的自动并行化能力以及执行复杂图查询的能力。

(3) 存储层。实际应用中，图数据的规模通常较大，将数据加载到不同处理阶段（不同引擎之间）并保存输出需要花费非常大的代价。为解决以上问题，GraphScope 提出了分布式内存数据管理系统 Vineyard，用以支持管理数据的分区和跨引擎的数据处理任务，并为上层应用提供零拷贝的数据读取。

2. 重要特性

(1) 高性能。高性能引擎支持对 Gremlin 查询的并行化以及图分析算法的自动并行化机制。数据管理系统 Vineyard 提供了高效的图存储和数据交换，实现了跨系统的数据共享。

(2) 一站式处理。提供了一个一站式环境，用于在集群中执行各种并行图操作。

(3) 云原生。支持在多个云原生环境中进行部署和扩展，便于集成云上的各种服务，使其能够充分发挥云计算的优势。

(4) 易使用。提供了丰富和灵活的统一编程模型，覆盖了多种典型的图计算任务，方便用户快速构建自己的应用。

3. 应用场景

GraphScope 可以应用于多种场景，包括但不限于：

(1) 社交网络分析。通过图算法来分析社交网络中的用户关系和社区结构。

(2) 推荐系统。基于图的推荐算法可以帮助发现用户的兴趣和推荐相关内容。

(3) 欺诈检测。分析交易图可以帮助识别潜在的欺诈行为。

(4) 知识图谱构建和查询：利用 GraphScope 来构建复杂的知识图谱，并通过图查询来检索信息。

由于 GraphScope 是开源项目，它的发展受到社区贡献的推动，也意味着它在持续更新和改进中。感兴趣的用户和开发者可以从其官方 GitHub 仓库获取源代码，参与项目的开发和使用。

5.6 案例：基于谱聚类的图像分割

图聚类是无监督学习中的一个基本问题,在计算机科学及相关领域中有着广泛的应用。作为谱方法中的一种,谱聚类是解决图聚类问题中最易于实现的一类方法,允许对图数据和非图数据进行聚类分析,在计算机视觉、文本挖掘和语音识别等领域得到了较为广泛的应用。图像分割是指将图像划分成若干个具有独特性质的区域并提取出重要目标信息的过程。本节将通过一个案例介绍谱聚类算法在图像分割中的具体应用。

谱聚类是从图论中发展而来的,它以谱图划分作为理论基础,通过将聚类问题转化为图划分问题,实现任意形状数据的聚类。与传统的 K-means 聚类算法相比,谱聚类算法对数据分布的适应性更强,聚类性能也更优越且计算量小。其基本思想是将图像的像素点看作图的节点,用连接节点的边上权值表示样本之间的相似度,基于图像像素点之间的相似性,通过对像素点组成的图进行切分,使切图后不同子图间的边权重之和尽可能小,而子图内的边权重之和尽可能大,从而实现聚类的目的。

基于以上思想,谱聚类算法的基本步骤描述如表 5-2 所示。

表 5-2 基于谱聚类的图像分割

输入:	待分割图像,分割数 k
输出:	分割后的图像结果
(1)	对输入图像中的像素点,根据数据点之间的相似度构建相似矩阵 W
(2)	根据相似模型计算相似矩阵 W 的度矩阵 D ,并构建标准化的拉普拉斯矩阵 $L_s = D^{-1/2} L D^{-1/2}$,其中 $L = D - W$ 表示拉普拉斯矩阵
(3)	对矩阵 L_s 进行特征分解,获得其前 k 个较小的特征值与特征向量 $u_1, u_2, \dots, u_k$
(4)	将特征向量 u_i 组成的矩阵 U 按行进行归一化处理,得到归一化向量集 U_s ,其中归一化向量为图像像素集在特征空间上的一个映射
(5)	对特征向量集 U_s 使用 K-means 算法进行聚类分析,得到最终聚类结果

谱聚类算法的关键代码描述如下:

```
# 求解归一化的拉普拉斯矩阵并对特征向量集进行聚类分析
import scipy
import sgtl.graph
import scipy.sparse.linalg
from sklearn.cluster import Kmeans

def Clustering_eigenvectors (DatasetGraph, num_clusters, num_eigenvectors):
    # 计算归一化拉普拉斯矩阵
    laplacian_matrix = sgtl.graph.Graph.normalised_laplacian_matrix(DatasetGraph)
    _, eigvecs = scipy.sparse.linalg.eigsh(laplacian_matrix, num_eigenvectors, which='SM')

    # 对特征向量集执行 K-means 算法
    labels = KMeans(n_clusters=num_clusters).fit_predict(eigvecs[:, :num_eigenvectors])

    # 对不同的类别进行划分
    clusters = [[] for _ in range(num_clusters)]
    for idx, label in enumerate(labels):
        clusters[label].append(idx)

    return clusters
```

本案例测试图像来自 Berkeley 彩色图像数据库 BSDS500,该公开数据集支持下载。该数据集常用于图像分割和物体边缘检测分析。采用谱聚类算法进行图像分割分析,图 5-15 展示了基于谱聚类算法生成的图像分割结果。通过原始图和分割后的图可以看出,谱聚类算法可以将图像中的天鹅、蝴蝶以及山等重要目标信息识别出来,且对于一些细节信息也能取得较好的分割效果,如蝴蝶的触角等。



图 5-15 基于谱聚类算法生成的图像分割结果

5.7 本章小结

图计算作为人工智能中的一项重要技术,已成为支撑新兴数据驱动市场的重要引擎,在社交网络、知识图谱和计算机视觉等不同领域有着广泛的应用。本章首先介绍图网络结构数据的表示方法及其描述性统计特征,然后介绍了基于图网络的计算方法,包括图遍历和图分割算法。图遍历算法主要介绍了广度优先搜索算法和深度优先搜索算法。在不同的场景中,基于拉普拉斯矩阵的最小割是实现图分割的一种重要方法,在图像分割等领域中具有重要的应用。社区发现是图计算的一个重要的方向,可用于实现图网络中节点的聚类,本章通过一个例子介绍了 GN 算法等社区发现算法。最后,简要介绍了开源的图计算平台 GraphScope,并通过图像分割案例讲解了谱聚类算法的应用。

习题

1. 选择题

- (1) 图网络结构中的节点表示什么? ()
A. 数据对象 B. 数据属性 C. 数据关系 D. 数据属性值
- (2) 图网络结构中,节点之间的关系是通过什么来表示的? ()
A. 节点属性 B. 边属性 C. 边的方向 D. 边的权重
- (3) 图网络结构中,节点的度是指什么? ()
A. 节点的属性数量 B. 节点的属性值数量
C. 节点与其他节点的连接数量 D. 节点与边的连接数量

- (4) 以下哪种中心性指标强调节点在网络中的“桥梁”作用? ()
- A. 紧密中心度 B. 介数中心度 C. 度中心度 D. 特征向量中心度
- (5) 在有向图中, 一个节点的介数中心度是指()。
- A. 经过的最短路径数量 B. 经过的所有路径数量
- C. 经过的最短路径的比例 D. 经过的所有路径的比例
- (6) Dijkstra 算法适用于哪种类型的图? ()
- A. 有向图 B. 无向图 C. 有向加权图 D. 以上所有
- (7) 以下哪个选项是深度优先搜索的特点? ()
- A. 层层遍历, 每一层的所有节点都被访问完后再访问下一层
- B. 先访问近邻节点, 再访问远邻节点
- C. 按照从顶点到叶子的顺序访问节点
- D. 从起始节点开始, 沿着路径一直向前探索, 直到到达一个未被访问邻居的节点, 然后再回溯到上一个节点继续探索
- (8) 在图的最小割问题中, 以下哪项描述是正确的? ()
- A. 最小割是图中顶点集的一个分割, 使得分割后的两个子图之间的边权重之和最小
- B. 最小割是将图中的边分割为两部分, 使得两部分之间的边权重之和最大
- C. 最小割是图中顶点集的一个分割, 使得分割后的子图边权重之和最大
- D. 最小割是图中顶点集的一个分割, 使得分割后的子图顶点数最少
- (9) 在社区发现中, 哪个概念描述了节点与其社区内其他节点的连接紧密度? ()
- A. 模块度 B. 网络连通性 C. 节点的度 D. 紧密中心度
- (10) 以下哪种算法是基于模块度的社区发现算法? ()
- A. 随机游走 B. 谱聚类 C. 隐含图 GN 算法

2. 简答及计算题

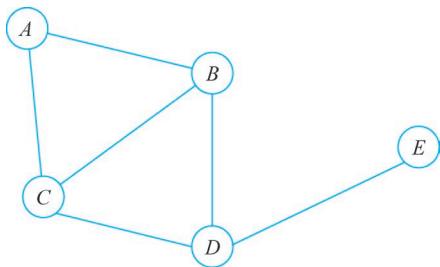
(1) 图网络结构中的节点表示什么? 节点之间的关系是通过什么来表示的?

(2) 对于一个包含 5 个节点的有向图, 使用 Dijkstra 算法计算从节点 1 到节点 5 的最短路径长度, 并给出最短路径。给定的加权矩阵如下:

$$\begin{bmatrix} 0 & 3 & \infty & 7 & \infty \\ 8 & 0 & 2 & \infty & \infty \\ 5 & \infty & 0 & 1 & \infty \\ 2 & \infty & \infty & 0 & 3 \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix}$$

(3) 给定一个有 10 个节点和 25 条边的无向图, 节点 A 的度为 5。计算节点 A 的度中心度。

(4) 给定一个无向图, 节点代表社交网络中的用户, 边代表用户之间的好友关系。使用模块度计算公式来计算社区划分的模块度。假设我们有以下好友关系数据:



假设我们已经通过社区发现算法得到了以下社区划分：

社区 1: A, B, C

社区 2: D, E

使用上述模块度计算公式, 计算该社区划分的模块度 Q 。

(5) 在社交媒体分析中, 图网络社区发现是一种常用的技术, 用于识别社交网络中的紧密相连的节点群体。这些社区可以帮助我们理解用户之间的互动模式, 发现潜在的兴趣群体或者影响力传播网络。假设我们有以下好友关系数据：

节点 A : B, C, D

节点 B : A, C, E

节点 C : A, B, D, E

节点 D : A, C, E, F

节点 E : B, C, D, F

节点 F : D, E

回答以下问题：

① 构建网络的邻接矩阵。

② 使用社区发现算法(如 Girvan-Newman 算法或模块度优化算法)来识别网络中的社区结构, 并给出算法的主要步骤。

③ 描述算法的输出结果, 包括社区的数量和每个社区的成员。

(6) 假设一个无向加权图的权重矩阵如下：

$$\mathbf{W} = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 3 \\ 2 & 3 & 0 \end{bmatrix}$$

计算其归一化拉普拉斯矩阵。

3. 思考题

(1) 银行用户之间通过相互交易从而产生一定的关系。把每一个银行账户看作一个节点, 如果账户 A 向账户 B 汇款, 那么就可以引出一条从节点 A 向节点 B 的边。如果账户 B 也向账户 A 汇款, 那么也可以引出一条从节点 B 向节点 A 的边。如何将大规模的交易数据通过图网络的形式表现出来, 并从中发现异常的交易模式和交易账户?

(2) GraphScope 是一款开源的大规模图计算平台, 能够用于解决实际生产场景中所涉及的图计算问题。试在 GraphScope 平台上搭建自己的社交网络分析环境, 并对题 3.1 涉及的金融反欺诈问题进行分析。