基于 Facebook、Cora 数据集的社区挖掘和节点分析

小组成员：曹竞禹、蒯文啸、王承乾

一、研究意义

社交网络与我们的日常生活息息相关。社区发现和网络属性的描述反映了社区的具体特征及构成演化，节点识别代表着从整体到局部的逻辑方法，有助于把握社交网络的主次趋势。链接预测与节点分类任务有助于我们从机器学习、深度学习的角度分析社交网络的现实意义。因此我们小组基于 Facebook 和 Cora 数据集对上述问题做了较为详细的解答。

二、研究过程

1. Facebook 数据集

(1) Facebook 社区发现

由于 Facebook 好友关系的双向关注机制，这里建立无向图模型，利用 Louvain 算法进行社区发现。

1) Louvain 算法介绍：

基于模块度 modularity 来衡量。模块度的物理意义是社区内部所有边的权重之和减去与社区相连的边权重之和。在无权图中，所有边的权重都可以视为 1。

$$Q = \frac{1}{2m}\sum_{i,j}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta(c_i, c_j)$$

$$\delta(u, v) = \begin{cases} 1, when\ u = v \\ 0, otherwise \end{cases}$$

Louvain 算法采用两步迭代，因此划分社区速度很快，对于点多边少的图，聚类效果非常明显。算法流程如下：

1.初始时将每个顶点当作一个社区，社区个数与顶点个数相同。

2.依次将每个顶点与之相邻顶点合并在一起，计算它们的模块度增益是否大于 0，如果大于 0，就将该结点放入该相邻结点所在社区。

3.迭代第二步，直至算法稳定，即所有顶点所属社区不再变化。

4.将各个社区所有节点压缩成为一个结点，社区内点的权重转化为新结点环的权重，社区间权重转化为新结点边的权重。

5.重复步骤 1-3，直至算法稳定。

2）可视化：

在代码实现中，我们调用 NetworkX 包，使用其中 community 的 best_partition 方法，默使用 Louvain 社区发现，用 matplotlib 绘制后的结果见图 1。

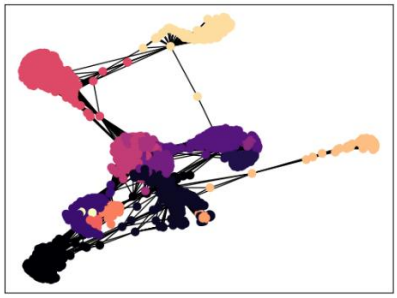

图 1 Facebook 社区发现的 matplotlib 可视化

将 python 输出的 gexf 格式文件导入 Gephi，利用合适的布局可视化结果见图 2。

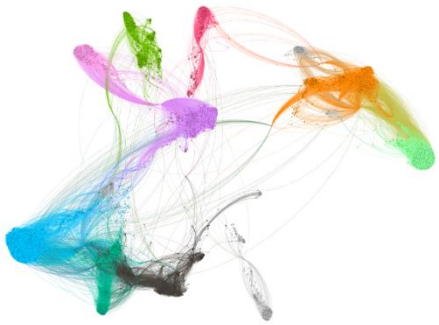

图 2 Facebook 社区发现的 Gephi 可视化

(2) Facebook 网络属性

1）图的直径、连通子图、局部聚类系数、平均最短路径：

为了了解 Facebook 社交网络的大小，我们首先计算了它的节点数和边数。

图的直径是指任意两个节点间最短路径的最大值，直径只存在于连通图中。由 NetworkX 分析得到 Facebook 只有一个连通子图，即所有节点是连通的，从而可以计算直径。平均最短路径度量了一个节点到另一个节点平均来说需要遍历的边的数量。

局部聚类系数可以反映一个节点的邻居节点之间聚集的程度。用 d_i 表示节点 i 的邻居节点 N_i 实际连接边数，D_i 表示节点 i 的邻居节点 N_i 全连接可形成的边数，局部聚类系数的计算公式为：

$$C_i = \frac{d_{N_i}}{D_{N_i}}$$

Facebook 图中有 267 个节点的局部聚类系数达到最大值 1，意味着这些节点的邻居节点是互相连接的，在这个数据集中的实际意义是某个人的好友全部也互为好友。对整个网络可用平均聚类系数来分析，反映的是平均而言某人的好友间都互相认识的比例。n 为节点总数，平均聚类系数的计算公式为：

$$C = \frac{1}{n}\sum_{i=1}^{n}\frac{d_{N_i}}{D_{N_i}}$$

网络属性结果列举如下：

平均最短路径长度：3.6925
连通子图数：1
直径：8

平均聚类系数：0.6055
共有267个节点的局部聚类系数达到最大

图 3 Facebook 网络属性

2）度分布：

度分布是度数为 x 的节点个数在整个网络中所占比例，可以反映这个网络中大部分节点与多少个附近节点相连。度分布的计算为 $P_x = \frac{m}{n}$，其中 m 为度数为 x 的节点个数。

度分布的计算可以通过调用 NetworkX 库返回各度数的频数得到。节点的度满足幂律分布，因此可以表示$P_x = ax^{-b}, a$为幂律截距，b为幂律指数。

若对公式同时取对数则变形为$\ln P_x = -b\ln x + \ln a$是一个线性关系，通过图示结果验证了度分布与函数关系匹配。从度分布图看出 Facebook 数据集中节点的度数集中在 0 到 50 之间。

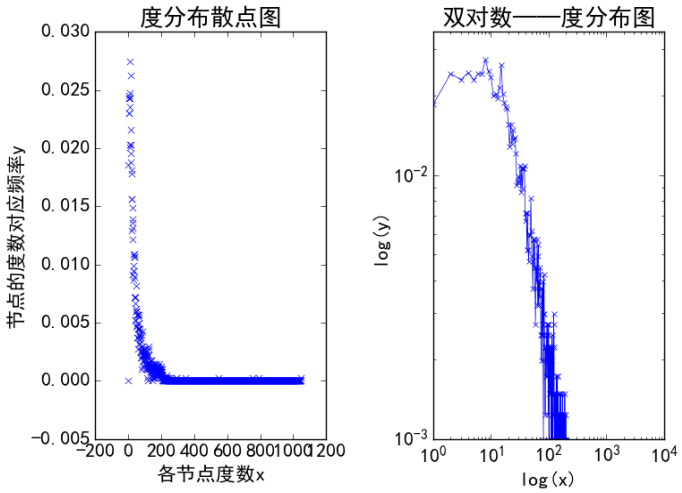

图 4 Facebook 度分布图

(3) 节点中心性识别

1）度中心性：

度中心性是拥有最大度数的节点，顾名思义就是一个点与其他点直接连接的总和，即：

$$C(v) = d$$

比如想知道某个人在网络社交圈中处于哪种程度的凝聚力，也就是要去判断所有节点的中心性强弱，这时就需要知道这个人和多少人有链接，这就是度中心度的作用。

但是在实际情况中，连接是有方向的，于是便有了点入中心度（或入度）和点出中心度（或出度），公式如下：

$$C_d(v_i) = d_i^{in}$$
$$C_d(v_i) = d_i^{out}$$
$$C_d(v_i) = d_i^{in} + d_i^{out}$$

入度表现一个人的受关注程度。入度高的人有可能会引导这个网络圈交流的内容、视角、深

度、广度等问题。出度表现一个人关注他人的程度。出度高的人，在网络中能够从很多的其他成员那里获得丰富的信息。在学习网络中可能就是知识、方法等；在娱乐圈中或许就是八卦新闻。

2）特征向量中心性：

通过结合无向图中邻居节点（或有向图的入度邻居）的重要性来概括（某个节点的）度中心性，计算公式为：

$$C_e(v_i) = \sum_{j=1}^{n} \frac{1}{\lambda} A_{j,i} C_e(v_j)$$

其中，λ是固定常量， A是邻接矩阵

假设C_e是所有节点中心性（值）的向量，则上式可表示为:$\lambda C_e = A^T C_e$因此，C_e是A^T（无向图中，$A^T = A$）的特征向量，λ则是对应的特征值。每个节点的中心值最好都> 0，因此应寻找各维度均> 0的特征向量。通过 Perron-Frobenius 定理，可以通过求解A^T的最大特征值对应的特征向量，即得到图中所有节点的特征向量中心性。

3）PageRank 中心性：

PageRank 中心性的核心在于这样的想法，即任一节点的出边不应该贡献其全部中心性质。因此在有向图中，一个中心性很高的节点（权威节点）如有很多出边，其贡献过来的中心性应当除以它的出度，即：

$$C_p(v_i) = \alpha \sum_{j=1}^{n} \frac{A_{j,i} C_p(v_j)}{d_o} + \beta$$

如果将上式写为：

$$C_p = \alpha A^T D^{-1} C_p + \beta 1$$

其中，D是度对角矩阵。当λ是$A^T D^{-1}$的最大特征值时，取$\alpha < \frac{1}{\lambda}$。

在计算每个节点的 PageRank 值时，通过不断迭代得到最后的收敛值，迭代公式为：

$$PR(p_i) = \alpha \sum_{p_j \in M_{pi}} \frac{PR(p_j)}{L(p_j)} + \frac{1-\alpha}{N}$$

其中，$L(p_j)$是节点p_j的出链数量。如果没有$\frac{1-\alpha}{N}$，网络中存在没有出链的网页，则会造成所有节点的 PR 值最终收敛于 0；若存在出链只指向自己的网页，则会造成最终收敛时只有该节点 PR 值 为 1，其他节点 PR 值都为 0。

4）接近中心性：

计算的是一个点到其他所有点的距离的总和，总和越小说明这个点到其他所有点的路径越短，也就说明这个点距离其他所有点越近。接近中心度体现的是一个点与其他点的近邻程度。接近中心性被定义为距离的倒数：

$$C(x) = \frac{1}{\sum_y d(y,x)}$$

在有向图中会得到入接近中心度和出接近中心度。入接近中心度是通过计算走向一个点的边来测量出其他点到达这个点的容易程度，一个点的入接近中心度越高，说明其他点到这个点

越容易。出接近中心度指的是一个点到达其他点的容易程度，通过一个点到其他点的最短距离的和的倒数，接近中心度越大，这个点到其他点越容易。
因此入接近中心度表达的是整合力，出接近中心度表达的是辐射力。

5）中介中心性

计算经过一个点的最短路径的数量。经过一个点的最短路径的数量越多，就说明它的中介中心度越高。如果一个大的社交网络中包含了几个小组，那么中介中心度高的人就起到将这些小组连接起来的作用。

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{st}$$

节点v的最终中介中心性是网络中所有节点对的最短路径经过该节点的σ_{st}之和。
因此，中介中心性的计算方法一般分为两步：
① 计算所有节点对的最短路径数和最短路径长度；
② 针对单个节点，加总其所有节点对依赖值。

节点中心性的最后结果汇总如下：

	度	特征向量	PageRank	中介	接近
最大值	1045	0. 095407	0. 007615	0. 480518	0. 459699
索引节点	107	352	1821	107	107

图 5 节点识别结果

4.链接预测

链接预测的思想是度量两个未连接节点的相似性，并连接相似程度高的节点。首先引入常用的节点相似性度量指标。

1） 节点相似性度量指标 Preferential Attachment，Jaccard，Adamic-Adar ：

假设用 N_i 表示节点 i 的邻居节点，则$|N_i \cap N_j|$表示的是节点 i 和 j 的公共邻居个数，$|N_i \cup N_j|$表示节点 i 和 j 所有的邻居个数。
① Preferential attachment 依赖于两个节点的邻居数，如果节点的邻居数都很大，可认为它们连接的概率也很大。计算公式：

$$S(i,j) = |N_i| * |N_j|$$

② Jaccard 系数是对公共邻居个数的标准化，如果两个节点的公共邻居较多，那么认为这两个节点联系紧密是合理的，即有高相似性。计算公式：

$$S(i,j) = \frac{|N_i \cap N_j|}{|N_i \cup N_j|}$$

③ Adamic-Adar 系数如下：

$$S(i,j) = \sum_{k \in N_i \cap N_j} \frac{1}{lnN(k)}$$

Adamic-Adar 的思想是先找出两个节点的公共邻居，然后用公共邻居的邻居数来度量相似性。公共邻居数目越少，相似性越高。直观理解是公共邻居的邻居数目较少时，却恰好作为 i 和 j 的公共邻居，凸显了公共邻居作为“中间人”的重要性。

在实践中，我们先对 Facebook 网络图随机删除 25%的边，然后利用剩余的图网络信息作为训练集，利用节点的相似性程度度量指标进行链接预测，绘制 ROC-AUC 曲线。
下图为三种系数的链接预测比较，Facebook 数据集中 jaccard 系数有最好的预测准确率 0.99，Preferential attachment 系数只利用了两个节点的度数，考虑的因素较少预测效果低于另外两种预测指标。总体来说，我们预测的连接与实际的连接较为接近，链接预测效果较好。

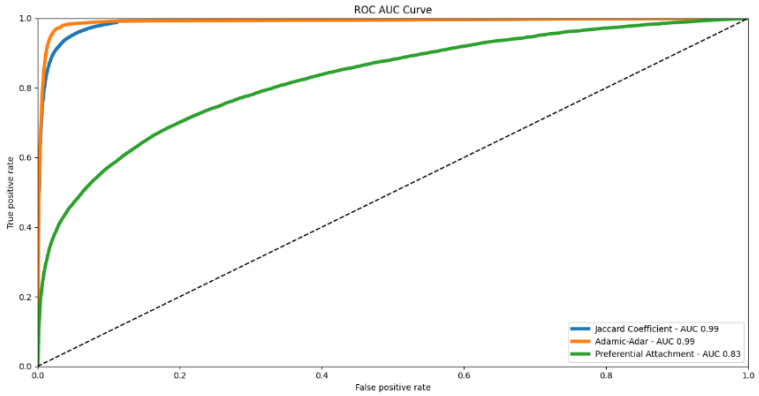

图 6 Facebook 数据集链接预测

2. Cora 数据集

(1) Cora 数据集介绍

Cora 数据集分为“Cora.sites”与“Cora.content”两个数据集。
前者包含 2708 个样本点和 10556 条边，每个样本点都代表一篇科学论文，其中第二列的论文编号引用了第一列的论文。后者是每个节点的特征矩阵，每篇论文都由一个 1433 维的 one-hot 词向量表示，即每个样本点有 1433 个特征。在数据集的最后一列，所有论文被划分为 7 个类别，每篇论文都至少引用了一片其他论文，或者被其他论文引用。

(2) 社区发现

同 Facebook 社交网络，社区可视化结果见图 7.

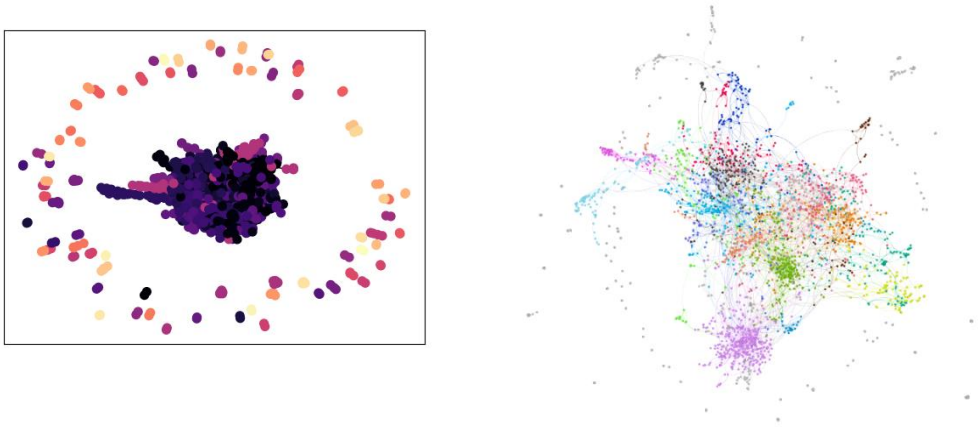

图 7 Cora 社区发现的 matplotlib 与 Gephi 可视化

(3) 链接预测

同 Facebook 中链接预测的方法，ROC 曲线和围成的面积 AUC 展示如下：

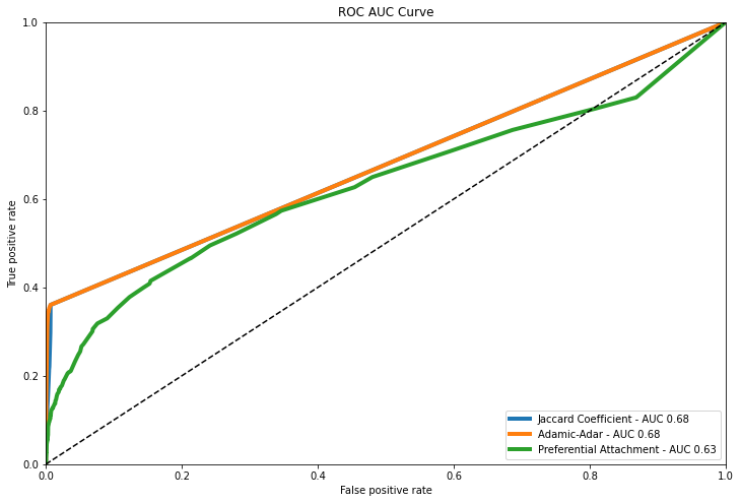

图 8 Cora 数据集链接预测

(4) 节点分类

1）图神经网络 GCN 简介：

由于 Facebook 数据集不包含节点的类别标签，因此这也是我们重新寻找数据集的理由。在这里我们使用图神经网络(GCN)进行训练。图神经网络是一种非常强大的在图上进行机器学习的神经网络框架，即使是随机初始化的单层 GCN 也可以生成网络中节点的有用的特征表示。对于给定的一个图$G = (V, E)$,GCN 的特征矩阵和一个图的结构表示矩阵（通常为邻接矩阵），因此我们需要对数据集做一些预处理的工作。

2）预处理：

对数据集的预处理工作具体包含从数据集中提取论文词向量特征矩阵、保留论文样本点和所属类别、将类别转化为矩阵存储、同时在训练神经网络前构造了邻接矩阵。

3）网络训练——添加自循环：

由于节点的聚合表示只是其邻居节点特征的集合，并不包括自己本身的特征。训练过程中，在应用传播规则之前将恒等矩阵添加到邻接矩阵来实现自循环能够使得节点的信息也被保留在邻接矩阵中，这样做可以进一步提高神经网络的预测精度。

4）分类测试结果：

我们输入了前 500 个样本的特征矩阵、邻接矩阵，对它们一共进行 200 次迭代，缩小训练集、增加迭代次数是希望避免过拟合的发生，最终分类精度达到 68%。如果添加节点自循环，分类精度可以达到 74%。

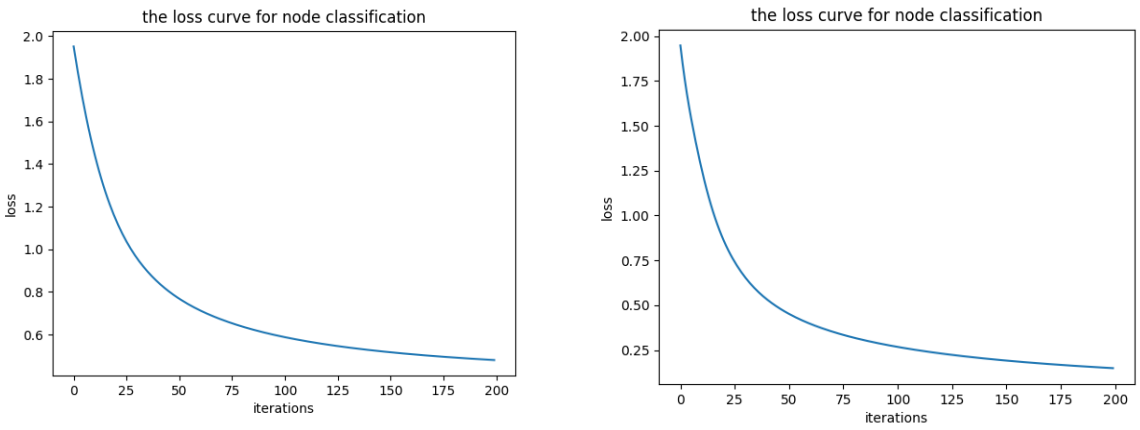

图 9 GCN 在 Cora 节点分类任务上的损失函数（添加自循环前与添加自循环后）

三、总结不足

1.在社区划分和计算最短路径时还是以调包来实现。希望以后能够根据算法原理做一些自己编写源码的尝试;
2.在链接预测中可以额外使用传统的机器学习算法来帮助判断预测精度;
3.为了节省运行时间，使用了单层图神经网络（因为尝试双层的预测精度并没有很大的提升），之后可以使用早停止等策略来进一步优化分类精度。

四、小组分工

曹竞禹 20210180052:
① 节点识别任务中 5 种中心性的度量
② ppt 的 Cora 数据集介绍展示部分
③ Cora 数据集特征矩阵的提取
④

蔺文啸 20210840018:
① 使用 Louvain 算法完成社区划分与 matplotlib 和 Gephi 的可视化;
② Facebook、Cora 数据集的 Jaccard Coefficient、Adamic-Adar、Preferential Attachment 三种相似性度量指标的计算、绘制 ROC-AUC 曲线完成链接预测评价;
③ 构建邻接矩阵、标签转化为列表存储的数据预处理，训练图神经网络 GCN 完成 Cora 数据集的节点分类;
④ ppt 的社区划分、链接预测、节点分类展示；ppt 的汇报。

王承乾 20210840013:
① 计算 Facebook 社交网络的网络属性（度数、直径，平均最短路径长度，局部聚类系数，平均聚类系数）、绘制度分布图和双对数度分布图;
② ppt 的 Facebook 网络属性展示部分。