社区挖掘与关联分析项目报告

加兴华	龙泽飞	周宇轩
23210980044	23210980060	23210980102

2024 年 6 月 28 日

Abstract

本项目专注于运用社交网络挖掘技术来深化对社区结构和节点关联性的理解。我们以 Cora 数据集为基础，运用了 Louvain、Infomap 和标签传播算法来识别社区结构，并通过一系列中心性度量方法对节点的重要性进行了细致评估。在这一坚实基础上，我们进一步探索了图数据的机器学习任务，特别是节点分类和链接预测。利用图神经网络，包括 GCN 和 GAT，我们在节点分类任务上实现了显著的准确度提升。同样，在链接预测任务中，GAT 和 PEG 模型也展现了卓越的性能，准确预测了学术文献间的潜在引用关系。

1 引言

1.1 研究背景

社交网络的广泛普及催生了海量的网络数据，这些数据记录了人们的互动并映射了复杂的社交结构。社区挖掘和关联分析作为关键技术，对于揭示社交网络中的结构特性和信息传播模式至关重要。本项目以 Cora 数据集为研究对象，专注于识别网络中的社区结构，评估节点重要性，并探索图数据的机器学习任务，包括节点分类和链接预测。

1.2 成员分工

- 龙泽飞：社区挖掘与网络分析
- 加兴华：机器学习节点分类任务，pre
- 周宇轩：机器学习连接预测任务

2 方法

2.1 数据集

Cora 数据集是一个公开的图结构数据集，广泛用于图机器学习研究。该数据集包含 2708 篇科学论文，每篇论文作为图中的一个节点，并且通过 5429 条边相互连接，这些

边代表论文之间的引用关系。每个节点都有 1433 个特征，这些特征是使用词袋模型从论文的文本中提取的，并且论文被分为 7 个不同的研究领域。在本研究中，我们采用了标准的 Cora 数据集划分，将数据集分为训练集、验证集和测试集，以评估我们所提出的模型的性能。

2.2 分析类算法

本节将介绍我们在 Cora 数据集上应用的关键分析技术，旨在揭示网络的社区结构和节点的重要性。我们选择了几种代表性的算法，包括社区挖掘和网络分析方法，为图数据的机器学习任务奠定基础。

2.2.1 社区挖掘

Louvain 算法 Louvain 算法 [1] 是一种贪婪优化方法，专注于通过最大化模块度 Q 来发现社区结构。模块度 Q 量化了社区内部连接的紧密程度，定义为：

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

其中 A_{ij} 是节点 i 和 j 之间的边的权重，k_i 是节点 i 的度，m 是图中所有边的权重之和，$\delta(C_i, C_j)$ 是指示函数，当节点 i 和 j 属于同一个社区时 $\delta(C_i, C_j) = 1$，否则 $\delta(C_i, C_j) = 0$。该算法通过两个阶段——局部优化和社区聚合——不断迭代，直至模块度不再显著增加。算法具体流程参见算法 1。

Infomap 算法 Infomap [4] 算法基于信息论，通过最小化随机游走过程中的编码长度来识别社区。编码长度 $L(M)$ 的定义为：

$$L(M) = q_{\curvearrowright} H(Q) + \sum_{i=1}^{m} p^i_{\circlearrowright} H(P^i),$$

其中 $q_{\curvearrowright}$ 是进入任意模块的概率，$H(Q)$ 是进入模块的概率分布的熵，$p^i_{\circlearrowright}$ 是模块 i 内部的随机游走概率，$H(P^i)$ 是模块 i 内部节点的概率分布的熵。通过最小化编码长度，算法能够找到最优的社区划分。算法流程参见 2。

标签传播算法 基于节点之间标签的传播，标签传播算法通过节点互相传递标签来确定社区结构。算法不需要预设社区数量。标签传播算法的基本思想是通过节点间标签的迭代更新，使得每个节点逐步采用其邻居节点中出现频率最高的标签，最终实现社区划分。算法在收敛时，每个社区内的节点将拥有相同的标签。算法 3 展示了标签传播的完整过程。

2.2.2 网络分析

中心性 中心性是复杂网络分析中用来衡量节点重要性或影响力的一种指标，有助于理解网络中节点在结构上的中心位置和在网络功能中的贡献程度。我们使用六种不同指标来衡量网络的中心性：度中心性，特征向量中心性，Katz 中心性，PageRank 中心性，介数中心性，接近度中心性。

度分布 度分布用于描述网络中节点度数的分布情况。节点的度数指的是与该节点直接相连的其他节点的数量。度分布具体定义为：在一个网络中，节点的度数 k 出现的概率 P(k)，即有多少比例的节点具有 k 个连接。度分布揭示了网络的整体结构特征。

平均节点距离 平均节点距离是复杂网络理论中用来衡量网络结构紧密程度的重要指标之一。它指的是网络中任意两个节点之间的平均最短路径长度，即通过最少的边连接两个节点所需的平均步数。具体计算平均节点距离的方法包括使用图论中的最短路径算法，来找到每对节点之间的最短路径，并将所有最短路径长度进行平均。平均节点距离的大小直接反映了网络中信息或影响传播的速度和效率。

2.3 模型算法

在图数据的机器学习任务中，节点分类（Node Classification）和链接预测（Link Prediction）是两个核心问题。节点分类的目标是利用图的结构信息和节点特征来预测未标记节点的类别。链接预测则旨在推断图中节点间可能存在的链接，这对于推荐系统、社交网络分析和生物网络研究等应用至关重要。

接下来，我们将介绍用于这些任务的基类模型算法。

2.3.1 Node2Vec

Node2Vec [2] 是一种用于图数据的节点嵌入技术，它基于 DeepWalk 算法进行了改进，通过调整随机游走的概率来平衡嵌入结果的 “同质性” 和 “结构性”。Node2Vec 算法通过生成节点的随机游走路径，然后利用这些路径来学习节点的向量表示。这种方法能够更好地捕捉网络的结构特性，从而提高节点分类、链接预测等任务的性能。

在数学上，Node2Vec 算法的核心是通过随机游走生成的路径序列来学习节点的嵌入。游走过程中，每个节点在每一步都会根据预定义的概率分布选择下一个节点。这种概率分布可以通过调整探索（广度优先）和利用（深度优先）之间的平衡来控制。

2.3.2 GCN

图卷积网络（GCN）[3] 是一种经典的神经网络模型，它直接在图结构数据上操作，并利用节点的局部连接模式来学习节点的嵌入表示。GCN 的核心思想是通过局部特征聚合来捕获全局图结构信息。在数学上，GCN 的层与层之间的特征传播遵循以下公式：

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

其中，$\tilde{A}$ 表示添加了自连接的无向图的邻接矩阵，$\tilde{D}$ 是度矩阵，$W^{(l)}$ 是层特定的可训练权重矩阵，$H^{(l)}$ 表示第 l 层的输出，而 σ 是激活函数，例如 ReLU。

2.3.3 GAT

图注意力网络（Graph Attention Networks, GAT）[5] 是一种在传统图神经网络上引入了自注意力机制的网络。模型的输入是一系列节点的表示 $\mathbf{h} = \{\vec{h}_1, \vec{h}_2, \ldots, \vec{h}_N\}$，$\vec{h}_i \in \mathbb{R}^F$，

其中 N 是节点的数量，F 是节点表示中特征的维度。模型输出新的节点表示 $\mathbf{h}' = \{\vec{h}'_1, \vec{h}'_2, \ldots, \vec{h}'_N\}$，$\vec{h}'_i \in \mathbb{R}^{F'}$。模型首先使用线性变换将原始输入映射到高维空间再做自注意力操作：

$$e_{ij} = a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)\alpha_{ij} = \text{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k\in\mathcal{N}_i}\exp(e_{ik})}$$

其中 $\mathcal{N}_i$ 是与图中与节点 i 相连的节点集合，我们只计算节点 i 与这个集合里的节点的注意力分数。更新的节点表示为：

$$\vec{h}'_i = \sigma\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}\mathbf{W}\vec{h}_j\right)$$

如果引入多头自注意力机制，则采用张量拼接操作聚合多头的信息，更新的节点表示为：

$$\vec{h}'_i = \prod_{k=1}^{K}\sigma\left(\sum_{j\in\mathcal{N}_i}\alpha^k_{ij}\mathbf{W}^k\vec{h}_j\right)$$

其中 $\|$ 代表张量拼接操作。而对于网络的最后一层（预测层）的多头自注意力机制，则采用平均操作聚合多头信息，更新的节点表示为：

$$\vec{h}'_i = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}_i}\alpha^k_{ij}\mathbf{W}^k\vec{h}_j\right)$$

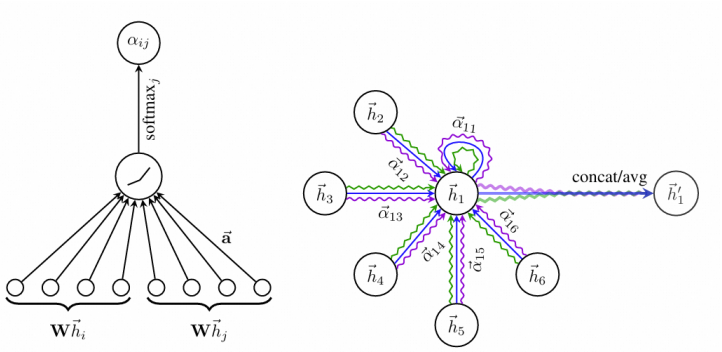


图 1: 左图表示注意力分数的计算机制，右图表示多头自注意力的聚合机制

2.3.4 PEG

PE-Stable GNN（简称 PEG）[6] 是一种保持了排列等变、稳定的利用了位置编码（Position Encoding, PE）的图神经网络。

对于图 A 和节点特征 X，引入位置特征 Z，则排列等变、稳定的图神经网络层 PEG 如下：

$$g_{\mathbf{PEG}}(A, X, Z) = \left(\psi\left[\left(\hat{A}\odot\Xi\right)XW\right], Z\right), \text{ where } \Xi_{uv} = \phi\left(\|Z_u - Z_v\|\right), \forall u, v \in [N]$$

表 1: 不同算法的社区挖掘结果

	社区数	模块度
Louvain	104	**0.812**
Infomap	287	0.727
标签传播	454	0.672

其中，ψ 是 element-wise 的激活函数，ϕ 是一个 MLP 层，$\odot$ 是哈德玛积操作。

3 实验结果与分析

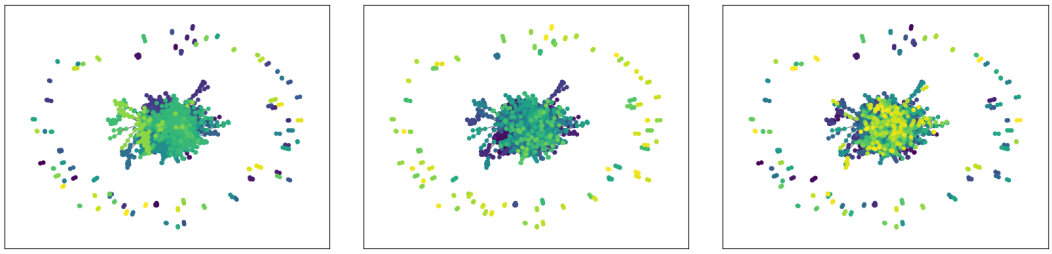
3.1 实验环境

本研究的实验环境主要基于 PyTorch 框架和 NetworkX 框架。NetworkX 提供了一系列用于创建、操作和研究复杂网络的算法和功能。PyTorch 作为一个开源的机器学习库，为深度学习模型的构建和训练提供了坚实的基础。我们主要选择 PyTorchGeometric 库来实现图神经网络（GNNs）。各实验的超参数设置详见代码。

3.2 社区挖掘与网络分析

3.2.1 社区挖掘

图 2 用弹簧布局展示了不同算法在 Cora 学术论文网络数据集上的实验结果。我们可以看到，3 种算法基于不同的依据，给出了明确清晰的划分结果，中心区节点社区在可视化中重合度高，而边缘节点社区明显分离。进一步地，经过社区挖掘，我们得到的社区数量和模块度如表 1 所示，从模块度角度来看，Louvain 算法取得了最好的效果。



(a) Louvain 算法　(b) Infomap 算法　(c) 标签传播算法

图 2: 社区挖掘算法实验结果对比

3.2.2 关联分析

中心性　图 3 在子图上展示了 6 种中心性测量方法，从左到右，从上到下分别使用：度中心性、特征向量中心性、Katz 中心性、PageRank 中心性、介数中心性和紧密中心性。我们规定，达到最大中心性的 50% 的节点用红色标注，达到最大中心性 25% 的节点用蓝色标注。从图中不难看出，在不同的测量方法下，高中心性节点的数量变化很大。

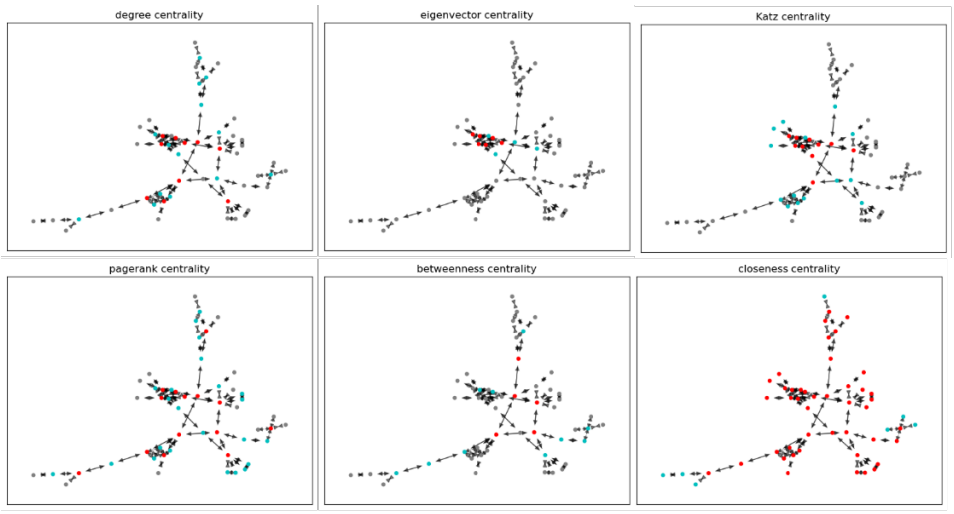
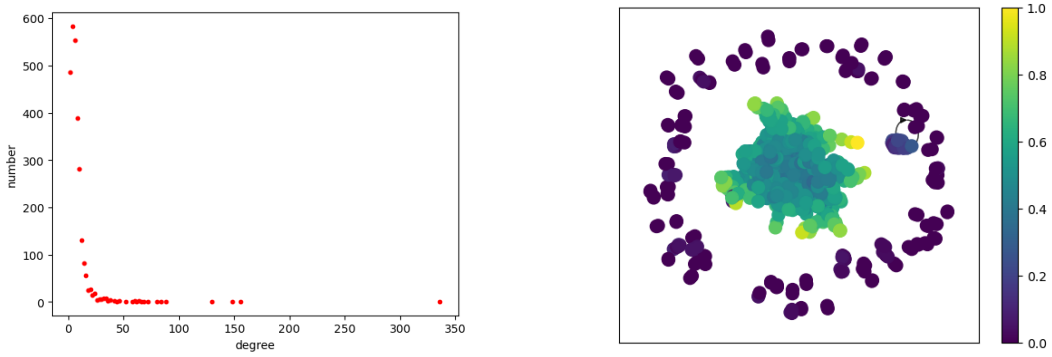


图 3: 网络的 6 个中心性

度分布 从图 4a中可以看出，Cora 网络的度分布呈现出典型的幂律分布特征：大多数节点的度较低，少数节点的度非常高。这种分布意味着在 Cora 网络中，大部分论文被引用的次数较少，只有少数几篇论文被大量引用。这种分布模式在学术引文网络中很常见。



(a) 度分布　　(b) 平均节点距离

图 4: Cora 网络属性

平均节点距离 Cora 网络的平均节点距离为 6.3103，反映出该图结构总体较为稀疏。图 4b 展示了各节点到其可达节点的平均距离。整张图明确地被分为外围和中心区域两个部分。外围的颜色为紫色，平均距离很短，说明这些节点的引用文献比较少或冷门，节点较为孤立。同时，中心区域从内部到外部颜色逐渐从绿色变为黄色，说明中心区域内部连接相较于外部的距离更短，联系更加紧密，而外部更为稀疏。这种现象表明，Cora 网络中存在一个核心区域，节点之间的连接相对紧密，信息可以较快地在核心区域内传播。

表 2: 不同模型在测试集上的分类准确率

模型	准确率 (%)
Node2Vec	73.8
GCN	82.3
GAT	81.8

3.3 节点分类

在本研究中，我们选取了 Node2Vec 作为节点分类任务的基准模型，同时运用了图卷积网络（GCN）和图注意力网络（GAT）进行比较分析。本节旨在综合探讨这三种模型在节点分类性能上的表现，并结合 t-SNE 可视化结果提供深入的讨论。

3.3.1 准确率分析

如表 2 所示，首先，Node2Vec 作为我们的性能基线，提供了一个重要的参照点。它通过随机游走机制在图中生成节点的序列化表示，为我们理解节点的局部和全局网络结构提供了一种有效途径。然而，在 Cora 数据集上的实验结果显示，Node2Vec 的测试准确率为 73.8%，这为我们后续的模型改进设定了起点。

接着，GCN 和 GAT 模型的引入显著提升了分类任务的性能。GCN 利用图卷积操作有效地聚合了节点的邻域特征，实现了 82.3% 的测试准确率，成为目前表现最佳的模型。GAT 则通过引入注意力机制，赋予了模型动态加权邻居节点的能力，其测试准确率达到了 81.8%，虽略低于 GCN，但依然显示出了强大的潜力。

3.3.2 t-SNE 可视化

此外，通过 t-SNE 算法对节点进行的可视化分析进一步验证了 GCN 和 GAT 模型的优势，如图 5 所示。在二维空间中，这两种模型得到的节点嵌入表示展现出了更好的类别分离度，这不仅直观地反映了模型的分类效果，也揭示了它们在捕捉节点特征方面的有效性。

综合考虑准确率和可视化结果，我们可以得出结论，GCN 和 GAT 模型在节点分类任务上的性能均优于传统的 Node2Vec 方法。这些图神经网络模型不仅在分类准确率上取得了突破，而且在学习节点表示方面也展现了更高的能力。

3.4 链接预测

在本任务中，我们使用 Cora 学术论文网络数据集，基于 PEG 的开源代码框架 [6]，分别训练了 GAT 和 PEG（各 200 个 epoch），并在测试集上进行链接预测任务——预测两篇学术文献之间是否存在引用关系，我们的评估指标包括 AUC、AP 和 F1 指标。

(a) Node2Vec (b) GCN (c) GAT

图 5: 不同模型分类结果 t-SNE 可视化

表 3: Cora 测试集结果

模型	AUC	AP	F1
GAT	0.9360	0.9416	0.8434
PEG	0.9658	0.9674	0.8217

4 结论

本项目通过应用 Louvain、Infomap 和标签传播算法成功识别了 Cora 数据集中的社区结构。网络分析揭示了节点的重要性和网络的幂律分布特性。GCN 和 GAT 在节点分类任务中表现优异，而 GAT 和 PEG 在链接预测任务中展现了高准确率。

References

[1] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[2] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[3] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[4] Martin Rosvall, Daniel Axelsson, and Carl T Bergstrom. The map equation. *The European Physical Journal Special Topics*, 178(1):13–23, 2009.

[5] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[6] Haorui Wang, Haoteng Yin, Muhan Zhang, and Pan Li. Equivariant and stable positional encoding for more powerful graph neural networks. In *International Conference on Learning Representations*, 2022.

附录

Algorithm 1 Louvain 算法

输入: 网络图 $G = (V, E)$
输出: 社区划分结果
初始化每个节点为一个独立社区，并按照式**??**计算初始的模块度 Q
repeat
  阶段 1: 局部优化
  for 每个节点 $i \in V$ **do**
    根据公式**??**，计算节点 i 从当前社区移动到相邻社区 C' 的模块度增益 ΔQ
    将节点 i 移动到使 ΔQ 最大的社区 C'
  end for
  阶段 2: 社区聚合
  生成一个新的网络图，其中每个社区作为一个新的节点，节点之间的边权重为原图中对应社区之间的总边权重
  重新计算新图的模块度 Q
until 模块度 Q 不再增加
返回最终的社区划分

Algorithm 2 Infomap 算法

输入: 网络图 $G = (V, E)$
输出: 社区划分结果
初始化每个节点为一个独立社区
repeat
  计算节点的进入概率和离开概率
  for 每个节点 $i \in V$ **do**
    计算节点 i 的进入概率 $p_{enter}(i)$ 和离开概率 $p_{exit}(i)$
  end for
  最小化描述编码长度
  for 每个节点 $i \in V$ **do**
    尝试将节点 i 移动到相邻社区 C'，计算移动后的编码长度变化 ΔL
    将节点 i 移动到使 ΔL 最小的社区 C'
  end for
  社区合并
  将移动后社区合并，形成新的社区结构
until 编码长度 L 不再显著减少
返回最终的社区划分

Algorithm 3 标签传播算法

输入: 网络图 $G = (V, E)$
输出: 社区划分结果
初始化每个节点 $i \in V$ 的标签 $L_i = i$
repeat
  将所有节点按随机顺序更新标签
  for 每个节点 $i \in V$ **do**
    统计节点 i 的邻居节点中每个标签出现的次数，节点 i 更新为邻居节点中出现频率最高的标签

$$L_i = \arg\max_{L_j} \sum_{j \in \mathcal{N}(i)} \delta(L_j, L), \tag{1}$$

    其中 $\mathcal{N}(i)$ 表示节点 i 的邻居节点集合, δ 是指示函数, 当 $L_j = L$ 时 $\delta(L_j, L) = 1$，否则 $\delta(L_j, L) = 0$。
  end for
until 标签不再变化或达到最大迭代次数
返回最终的社区划分，即节点的标签