

華中科技大學

大数据系统综合实践

院 系 计算机学院

专业班级 大数据 2102 班

姓 名 张钧玮

学 号 U202115520

指导教师 石宣化

2024 年 10 月 30 日

目 录

1	绪论	1
2	实验环境	1
3	实验目的	1
4	实验概况	2
4.1	项目结构	2
5	实验内容	3
5.1	GN 算法	3
5.2	Louvain 算法	3
5.3	GPU 加速	4
6	实验结果	5
6.1	功能数据集: Cit-HepPh	5
6.2	性能数据集: soc-LiveJournal1	6
7	总结	7

1 绪论

本次大数据系统综合实践聚焦于大规模图数据中社区发现算法的设计与性能优化，分别使用 Louvain 算法和 Girvan-Newman 算法来进行社区发现，比较不同算法的性能差距，对算法进行性能优化。项目代码开源于 <https://github.com/hakureiReimu/network-community-detection>

2 实验环境

本次实验环境使用 conda 进行环境管理,配置如下

表 2-1 实验环境配置

实验依赖	具体版本
CPU	AMD Ryzen 7 6800H
操作系统	windows 11
python	3.8.20
networkx	3.1
python-louvain	0.16

conda 详细环境见项目根目录"requirements.txt"

3 实验目的

本实验旨在通过大规模图数据中社区发现算法的设计与性能优化，帮助学生深入理解图计算系统的工作原理和性能优化机制，并学会使用图计算框架进行大规模图数据分析和处理。通过此实验，学生将能够掌握图计算的基本原理，编写比较复杂的图算法程序并进行性能调优。

4 实验概况

4.1 项目结构

项目使用模块化设计。使用 input 文件夹保存数据集，output 文件夹保存结果，过程日志保存在 log 文件夹，src 文件夹下存放源代码。

源代码分为 data_loader.py、community_detection.py、utils.py、logger_config.py、main.py 五个文件，其中 data_loader.py 用于加载数据，community_detection.py 用于运行社区发现，utils.py 用于工具函数，logger_config.py 用于配置日志，main.py 用于调用其他模块。

项目结构如下：

```
├── README.md
├── input
│   ├── Cit-HepPh.txt
│   └── soc-LiveJournal1.txt
├── log
│   ├── app_20241030_084820.log
│   ├── app_20241030_085709.log
│   ├── app_20241030_085713.log
│   ├── app_20241030_091020.log
│   ├── app_20241030_091654.log
│   └── app_20241030_093912.log
├── output
│   └── communities.output
├── requirements.txt
├── src
│   ├── __pycache__
│   │   ├── community_detection.cpython-38.pyc
│   │   ├── data_loader.cpython-38.pyc
│   │   ├── logger_config.cpython-38.pyc
│   │   └── utils.cpython-38.pyc
│   ├── community_detection.py
│   ├── data_loader.py
│   ├── logger_config.py
│   ├── main.py
│   └── utils.py
```

5 directories, 20 files

5 实验内容

5.1 GN 算法

5.1.1 概述

Girvan-Newman (GN) 算法是一种用于社区发现的算法，主要用于检测网络中自然形成的社区结构。它通过逐步移除网络中“桥接”不同社区的关键边，来识别社区。GN 算法是一种基于边介数（edge betweenness）来进行社区划分的层次聚类算法。

核心概念：

1. 边介数：在图中，两点之间的最短路径数量可以有多条。当某条边位于更多的最短路径上时，它的“边介数”就更高，意味着它更可能是两个社区之间的桥接边。
2. 社区：GN 算法认为网络中自然形成的社区是通过较少的边连接的，而这些连接不同社区的边往往具有较高的边介数。

工作流程：

1. 计算每条边的边介数：对于网络中的所有边，计算边介数（即通过该边的最短路径数量）。
2. 删除最高边介数的边：找到边介数最大的边并删除，将删除后的网络重新计算边介数。这样逐渐将网络划分成多个连通分支。

重复步骤 1 和 2：继续删除边介数最高的边，直到整个网络分解成多个社区（即多个连通分支）。每次删除边时，重新计算剩余边的边介数。

5.1.2 缺点

计算复杂度高：GN 算法的时间复杂度较高（ $O(|V| \times |E|)$ ），因此只适用于节点数相对较少的图。实际上该算法的时间复杂度已经到了几乎无法成功跑完大数据集的地步。

5.2 Louvain 算法

5.2.1 概述

Louvain 算法是一种高效的社区发现算法，用于检测大型网络中的社区结构。它通过模块度 (modularity) 最大化的方式将网络划分成多个社区，是一种基于模块度优化的分层聚类算法。

核心概念：

1. 社区：在网络中，节点之间的连接密度较高，而不同社区之间的连接相对较少。
2. 模块度 (Modularity)：模块度是衡量网络划分质量的指标。其值越高，说明社区划分越合理（即社区内部的连接密度更高，社区之间的连接密度更低）。

工作流程：

Louvain 算法分为两个主要阶段，逐层迭代，直到模块度不再显著提升：

局部优化阶段：

初始化时，每个节点被视为一个独立的社区。对每个节点，将它移动到相邻节点的社区中，以寻找最大化模块度的分配。重复这一过程，直到不能进一步提高模块度。社区聚合阶段：

将当前划分结果中的每个社区视为一个超级节点，重新构建网络。在新网络上，重复局部优化和社区聚合，直到模块度不再显著提升。

5.2.2 优点

Louvain 算法的时间复杂度较低，适合大规模网络的社区发现。

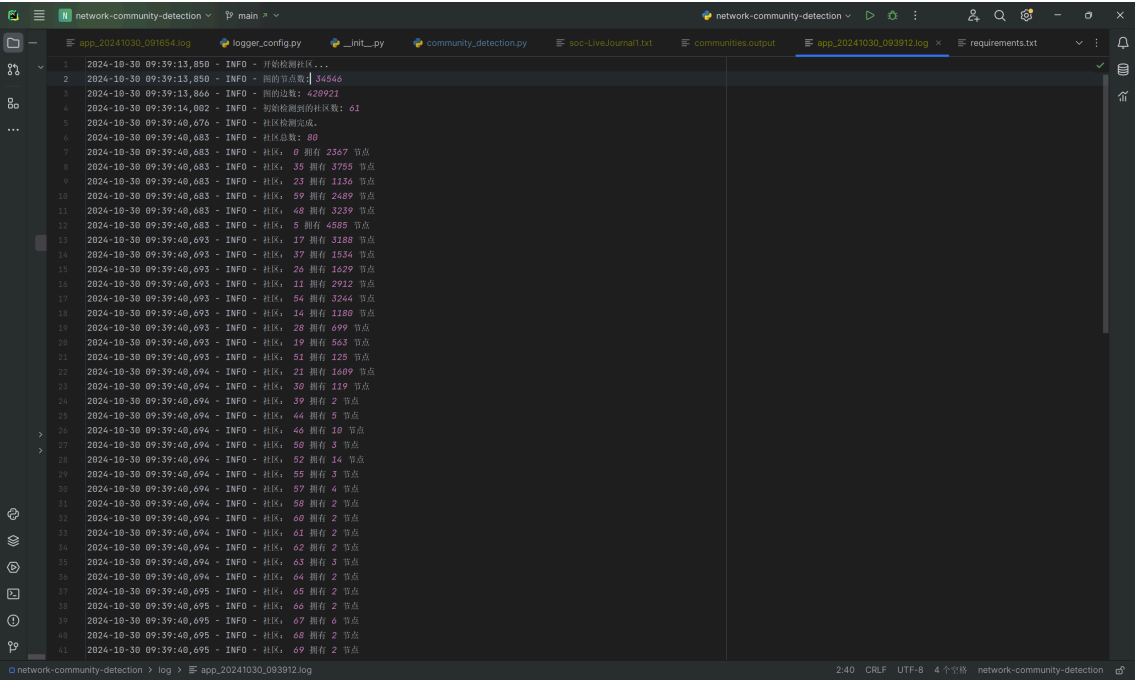
5.3 GPU 加速

矩阵数据乘法运算的本质是 $N*N$ 个并行计算，属于是高并行度低冲突的算法，因此可以使用 GPU 加速。因为 Louvain 算法中计算模块度的算法是并行的矩阵乘法，因此在 networkx 库中，scipy 库中的 csr_matrix 会使用 gpu 进行稀疏矩阵的乘法运算，从而实现 GPU 加速。

6 实验结果

6.1 功能数据集：Cit-HepPh

运行结果如下：



```
1 2024-10-30 09:39:13,850 - INFO - 开始检测社区...
2 2024-10-30 09:39:13,850 - INFO - 图的节点数: 34546
3 2024-10-30 09:39:13,866 - INFO - 图的边数: 420921
4 2024-10-30 09:39:14,002 - INFO - 检测得到社区数: 61
5 2024-10-30 09:39:40,676 - INFO - 社区检测完成。
6 2024-10-30 09:39:40,683 - INFO - 社区总数: 80
7 2024-10-30 09:39:40,683 - INFO - 社区: 0 拥有 2367 节点
8 2024-10-30 09:39:40,683 - INFO - 社区: 35 拥有 3795 节点
9 2024-10-30 09:39:40,683 - INFO - 社区: 23 拥有 1136 节点
10 2024-10-30 09:39:40,683 - INFO - 社区: 59 拥有 2489 节点
11 2024-10-30 09:39:40,683 - INFO - 社区: 48 拥有 3239 节点
12 2024-10-30 09:39:40,683 - INFO - 社区: 5 拥有 4585 节点
13 2024-10-30 09:39:40,693 - INFO - 社区: 17 拥有 3188 节点
14 2024-10-30 09:39:40,693 - INFO - 社区: 37 拥有 1534 节点
15 2024-10-30 09:39:40,693 - INFO - 社区: 26 拥有 1629 节点
16 2024-10-30 09:39:40,693 - INFO - 社区: 12 拥有 2912 节点
17 2024-10-30 09:39:40,693 - INFO - 社区: 54 拥有 3244 节点
18 2024-10-30 09:39:40,693 - INFO - 社区: 14 拥有 1780 节点
19 2024-10-30 09:39:40,693 - INFO - 社区: 28 拥有 699 节点
20 2024-10-30 09:39:40,693 - INFO - 社区: 19 拥有 563 节点
21 2024-10-30 09:39:40,693 - INFO - 社区: 51 拥有 125 节点
22 2024-10-30 09:39:40,694 - INFO - 社区: 21 拥有 1609 节点
23 2024-10-30 09:39:40,694 - INFO - 社区: 30 拥有 119 节点
24 2024-10-30 09:39:40,694 - INFO - 社区: 39 拥有 2 节点
25 2024-10-30 09:39:40,694 - INFO - 社区: 44 拥有 5 节点
26 2024-10-30 09:39:40,694 - INFO - 社区: 46 拥有 10 节点
27 2024-10-30 09:39:40,694 - INFO - 社区: 50 拥有 3 节点
28 2024-10-30 09:39:40,694 - INFO - 社区: 52 拥有 14 节点
29 2024-10-30 09:39:40,694 - INFO - 社区: 55 拥有 3 节点
30 2024-10-30 09:39:40,694 - INFO - 社区: 57 拥有 4 节点
31 2024-10-30 09:39:40,694 - INFO - 社区: 58 拥有 2 节点
32 2024-10-30 09:39:40,694 - INFO - 社区: 60 拥有 2 节点
33 2024-10-30 09:39:40,694 - INFO - 社区: 61 拥有 2 节点
34 2024-10-30 09:39:40,694 - INFO - 社区: 62 拥有 2 节点
35 2024-10-30 09:39:40,694 - INFO - 社区: 63 拥有 3 节点
36 2024-10-30 09:39:40,694 - INFO - 社区: 64 拥有 2 节点
37 2024-10-30 09:39:40,695 - INFO - 社区: 65 拥有 2 节点
38 2024-10-30 09:39:40,695 - INFO - 社区: 66 拥有 2 节点
39 2024-10-30 09:39:40,695 - INFO - 社区: 67 拥有 6 节点
40 2024-10-30 09:39:40,695 - INFO - 社区: 68 拥有 2 节点
41 2024-10-30 09:39:40,695 - INFO - 社区: 69 拥有 2 节点
```

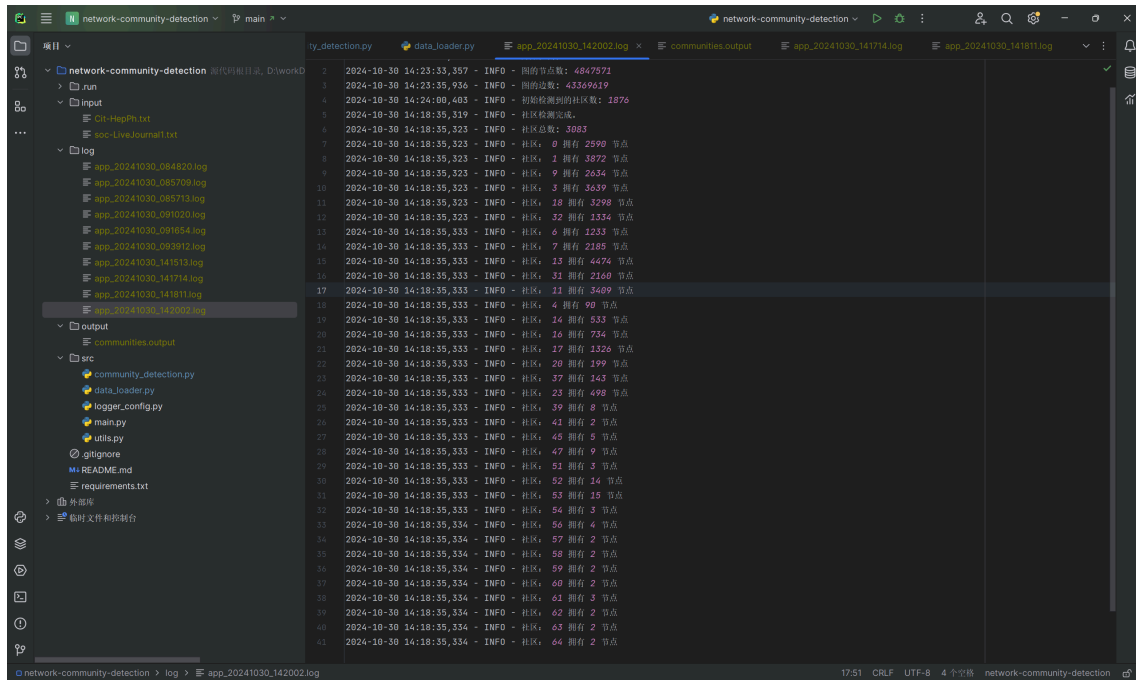
图 6-1 Louvain 算法运行结果

运行时间：

2024-10-30 14:18:35,340 - INFO - 算法总时间:22.58 seconds

6.2 性能数据集：soc-LiveJournal1

运行结果如下：



```
2 2024-10-30 14:23:33,357 - INFO - 图的节点数: 4847571
3 2024-10-30 14:23:35,936 - INFO - 图的边数: 43369619
4 2024-10-30 14:24:00,403 - INFO - 初始检测到的社区数: 1876
5 2024-10-30 14:18:35,319 - INFO - 社区检测开始
6 2024-10-30 14:18:35,323 - INFO - 社区总数: 3083
7 2024-10-30 14:18:35,323 - INFO - 社区: 0 拥有 2598 节点
8 2024-10-30 14:18:35,323 - INFO - 社区: 1 拥有 3872 节点
9 2024-10-30 14:18:35,323 - INFO - 社区: 9 拥有 2634 节点
10 2024-10-30 14:18:35,323 - INFO - 社区: 3 拥有 3639 节点
11 2024-10-30 14:18:35,323 - INFO - 社区: 18 拥有 3298 节点
12 2024-10-30 14:18:35,323 - INFO - 社区: 32 拥有 1336 节点
13 2024-10-30 14:18:35,333 - INFO - 社区: 6 拥有 1233 节点
14 2024-10-30 14:18:35,333 - INFO - 社区: 7 拥有 2185 节点
15 2024-10-30 14:18:35,333 - INFO - 社区: 13 拥有 4474 节点
16 2024-10-30 14:18:35,333 - INFO - 社区: 31 拥有 2160 节点
17 2024-10-30 14:18:35,333 - INFO - 社区: 11 拥有 3409 节点
18 2024-10-30 14:18:35,333 - INFO - 社区: 4 拥有 90 节点
19 2024-10-30 14:18:35,333 - INFO - 社区: 14 拥有 633 节点
20 2024-10-30 14:18:35,333 - INFO - 社区: 16 拥有 734 节点
21 2024-10-30 14:18:35,333 - INFO - 社区: 17 拥有 1326 节点
22 2024-10-30 14:18:35,333 - INFO - 社区: 20 拥有 199 节点
23 2024-10-30 14:18:35,333 - INFO - 社区: 37 拥有 143 节点
24 2024-10-30 14:18:35,333 - INFO - 社区: 23 拥有 498 节点
25 2024-10-30 14:18:35,333 - INFO - 社区: 39 拥有 8 节点
26 2024-10-30 14:18:35,333 - INFO - 社区: 41 拥有 2 节点
27 2024-10-30 14:18:35,333 - INFO - 社区: 45 拥有 5 节点
28 2024-10-30 14:18:35,333 - INFO - 社区: 49 拥有 9 节点
29 2024-10-30 14:18:35,333 - INFO - 社区: 53 拥有 5 节点
30 2024-10-30 14:18:35,333 - INFO - 社区: 52 拥有 14 节点
31 2024-10-30 14:18:35,333 - INFO - 社区: 53 拥有 15 节点
32 2024-10-30 14:18:35,333 - INFO - 社区: 54 拥有 3 节点
33 2024-10-30 14:18:35,334 - INFO - 社区: 56 拥有 4 节点
34 2024-10-30 14:18:35,334 - INFO - 社区: 57 拥有 2 节点
35 2024-10-30 14:18:35,334 - INFO - 社区: 58 拥有 2 节点
36 2024-10-30 14:18:35,334 - INFO - 社区: 59 拥有 2 节点
37 2024-10-30 14:18:35,334 - INFO - 社区: 60 拥有 2 节点
38 2024-10-30 14:18:35,334 - INFO - 社区: 61 拥有 3 节点
39 2024-10-30 14:18:35,334 - INFO - 社区: 62 拥有 2 节点
40 2024-10-30 14:18:35,334 - INFO - 社区: 63 拥有 2 节点
41 2024-10-30 14:18:35,334 - INFO - 社区: 64 拥有 2 节点
```

图 6-2 Louvain 算法运行结果

运行时间:

2024-10-30 15:05:42,135 - INFO - 算法总时间:1801.15 seconds

7 总结

为了跑通算法花了很长时间，在小数据集时 GN 算法还算有效，但是在大数据集上 GN 算法的时间复杂度太高，无法跑通。而 Louvain 算法在大数据集上有很好的表现，比较好的完成了实验的要求。