Github合作关系网络的社团发现及分析

1. 引言

Github作为目前世界上包含最庞大的开发者规模的合作社区，截至2021年，在其上工作的开发者已逾7300万，它所包含的最重要的信息之一就是用户之间错综复杂的合作关系。在该社区中，用户和项目是其最值的关注的元素。其中，用户之间可以有follow的关系，同时，用户可以对喜欢的项目进行star或者fork操作[1]。而对于项目来说，用户可以创建项目，并对自己创建或者fork的项目进行修改，更新或删除等操作。

Github中的仓库可以由多个用户共同完成，用户的commit操作可以对仓库进行更新。在大型开源项目中，Pull Request简称PR是提交代码修改的重要操作。该功能允许一个用户在自己的分支进行修改后，在管理员的允许之下将自己的分支合并到主分支。除此之外，非代码贡献者的issue也是用于计算合作关系时的重要考虑因素。

本报告旨在基于对用户之间的合作关系，建立起用户之间的类似于社交网络的合作网络，并对其中存在的关系较为密切的子网络即社团进行聚类和分割。该项任务可作为发掘用户的技术领域的基础和子任务之一，也可以为开源社区的建设提供参考。

1. 问题描述

本研究研究的主要对象是Github用户合作网络，它是以用户为节点，用户间的合作关系为边构建的加权无向图，其中边的权重为两节点间合作影响力取值。本研究的合作定义为两个用户对同一个仓库有贡献，此时计为合作一次。其中，用户以显式节点出现，而仓库隐式计算为合作影响力取值。当存在两个用户对同一个仓库有贡献时，合作影响力以加权计数的统计方式记录并体现在网络中。这样做的目的是简化网络，避免生成过于复杂的异质图。

本次进行实验的目标是选取Github上2021年的活跃开发者，构建起合作网络。并据此对网络总体特征和节点重要性进行计算，并将计算结果作为社团划分的依据，按照不同的特征和粒度可视化分析社团结构。从而达到基于用户的提交信息寻找同一领域内的用户，并最终实现开源用户领域区分的目的。项目所采用的数据集是从Github上爬取的Github用户行为数据集。

1. 方法

实验首先需要构建Github中活跃用户的开发者合作网络。利用阿里云的SQL语句查询功能，可以对每个用户使用一条语句将其与每个其他活跃用户的合作次数提取出来，然后处理得到每条边对应的权值。由于网络构建的性质，对于两个有合作关系的开发者来说，其任意一方对另一方必然都存在边，且具有相等的合作次数。在构建网络的过程中，需要对每个活跃用户在2021年内使用SQL中的聚集函数计算其与其他活跃用户的合作次数并记录下来。

在包含社团结构的网络中，一个社团内部的节点之间连接比较紧密，而处于不同社团的节点连接就比较稀疏。在Github开发者合作网络中，不同开发者构成的社团结构关系到网络中技术领域、开发者所在的社区功能划分。对网络整体的了解和分析，有助于发现Git开发者合作网络中的动力学传播并发掘其中隐藏的规律，甚至于帮助预测网络不同社区中开发者的行为演化。

​ Github开发者合作网络划分的意义在于对合作网络中的单个开发者节点进行分析，通过开发者节点所在社团的位置发掘该开发者节点在社团中的功能—若节点位于社团比较中心的位置，说明这个开发者在社团中可以起到领导和影响其他开发者的功能；而位于社团结构边缘的节点，应该具有不同社团之间信息传输的功能。

发现社团结构一定程度上揭示了真实网络中的现象，即网络由不同社团构成，社团又由小社团构成。得到网络社团结构后，可以对网络的细节行为进一步详细观察和控制，有助于更加简单方便和有针对性地分析和处理网络中的问题。

1. 评价

对原始数据可视化得到的所有用户合作关系图，如图4-1所示。图中共有1001个节点，共有8341条边。其中，表示两个节点之间的边越粗，表示其权重越大，否则其权重越小；节点的绿色饱和度越高，则与其连接的节点数量越多，否则它的度越小。图中可以明显看出，该数据集中节点之间联系即边的权重大多较小，也即两个固定节点之间的协助次数平均来说比较少，这在一定程度上也在节点的度上能体现出来。

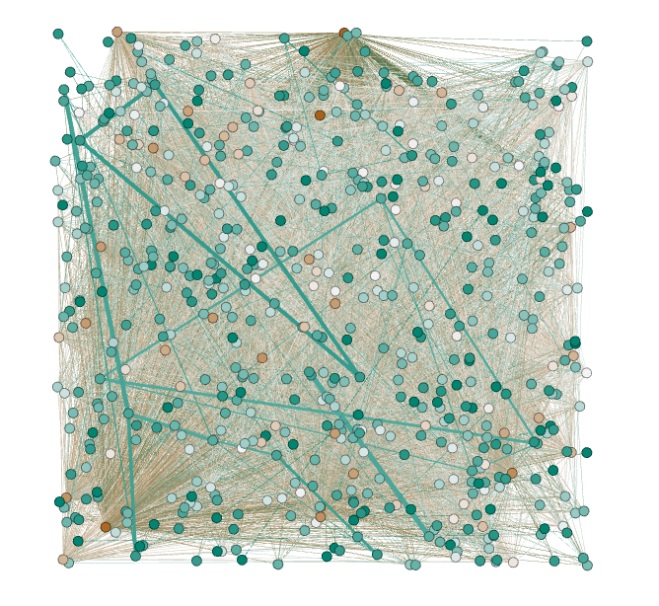


图4-1 所有用户的协作关系图

合作关系在平均值以上的用户，分为两个方面，一是与多个用户进行合作，二是与其他所有用户的合作次数权重在均值以上。分别如图4-2所示，左图为计算得到所有用户平均合作过的用户数量为8341人次，此时协作关系图中共有588个节点，连接节点的共有1214条边；右图2 为计算得到用户之间的平均合作次数约为25163931次，共有142个节点，它们之间共有177条边。

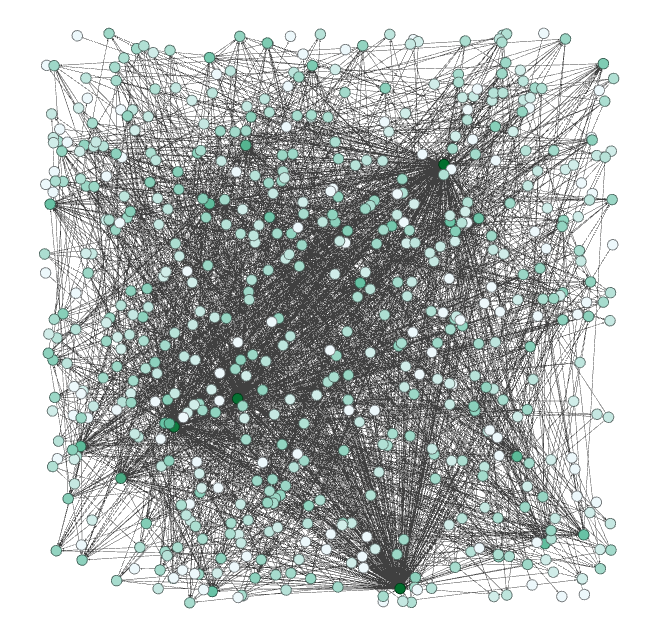
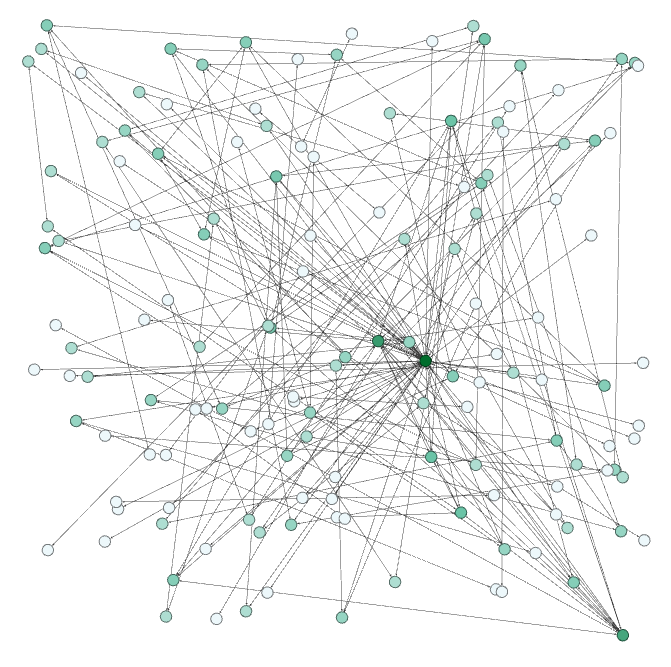
 

图4-2 合作关系在平均值以上用户协作关系图

在一个复杂网络中，聚类系数是用于度量图中节点聚类程度的指标。在真实世界的社交网络中，节点往往以相对高密度的联系为特征，形成紧密的群体，其概率往往大于随机网络中节点之间联系的平均概率。聚类系数在某种程度上类似于社会网络中的“物以类聚，人以群分”特性，它描述网络中节点的邻居节点之间也互为邻居节点的比例。其中，节点之间的平均聚类系数给出了节点聚集或报团的总体迹象。

同时，在Github开发者合作无向图中，平均路径长度指图中所有节点对之间的平均图距离。互相连接的节点的图距离为1。直径是最长的任何两个节点之间的距离，表明两个最遥远的节点相距多远。

Github开发者及其合作关系构成的无向图中，按照上述计算公式计算得到，两个任意节点之间的最长距离，即网络直径为6；所有节点对之间的平均图距离，即平均路径长度为2.1117842784568897；描述一个节点聚类或报团迹象的聚类系数指标如图4-3所示；这里取单个聚类系数的平均值，得到网络节点平均聚类系数，达到0.829；Git开发者合作网络中三角形总数为93128。

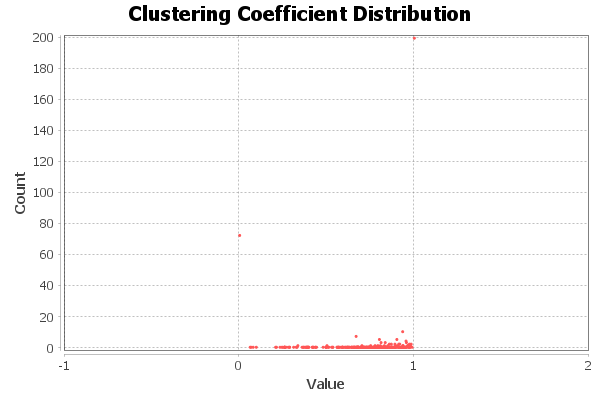


图4-3网络平均聚类系数分布

从数值以及变化趋势来看，网络直径为6，符合“六度空间”基本理论，此外，网络节点对平均路径长度很小，只有2.1左右，网络节点平均聚类系数，达到0.829，因此，Github开发者合作网络显著符合小世界效应。

在明确Github开发者合作网络所具有的性质之后，就可以使用选择算法对该网络进行社区发现和区分。本项目首先尝试了基于Louvain算法尝试探索了基于模块度的社团挖掘方法，发掘社团结构66个，结果如图4-4、4-5所示。显然，可视化效果比较差。

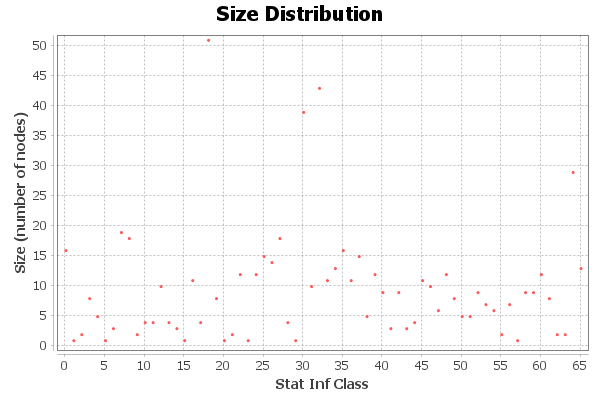


图4-4发现社团数目66个

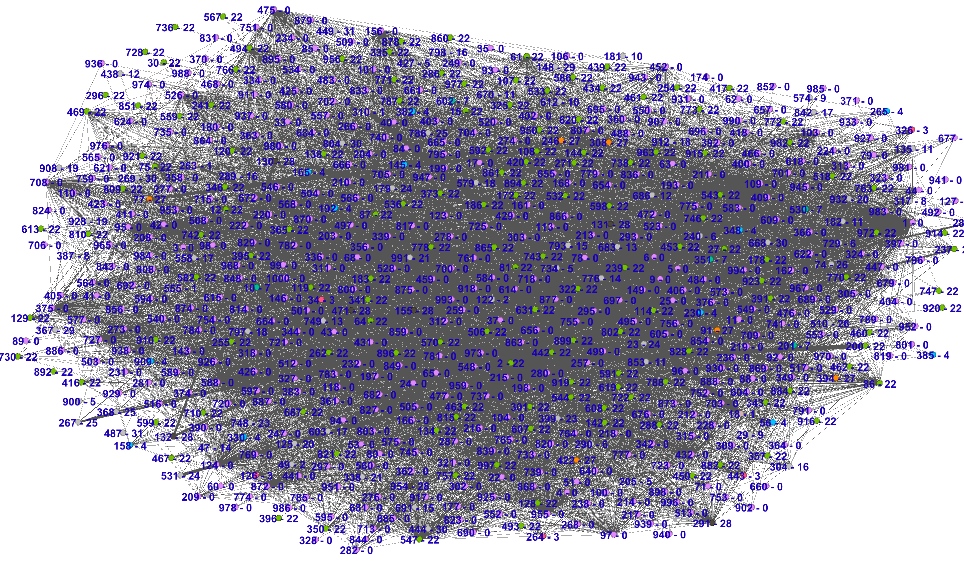


图4-5社团可视化

然后本项目在调整之后继续使用Louvain算法探索了基于模块度的社团挖掘方法，发掘社团结构32个，结果如图4-6 和图4-7所示。可视化效果没有明显改善。

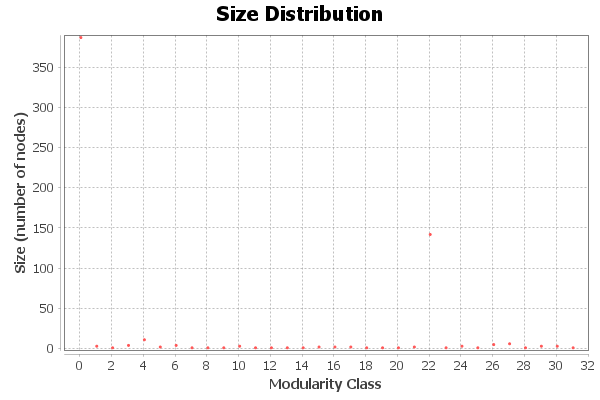


图4-6社团发现数目32个

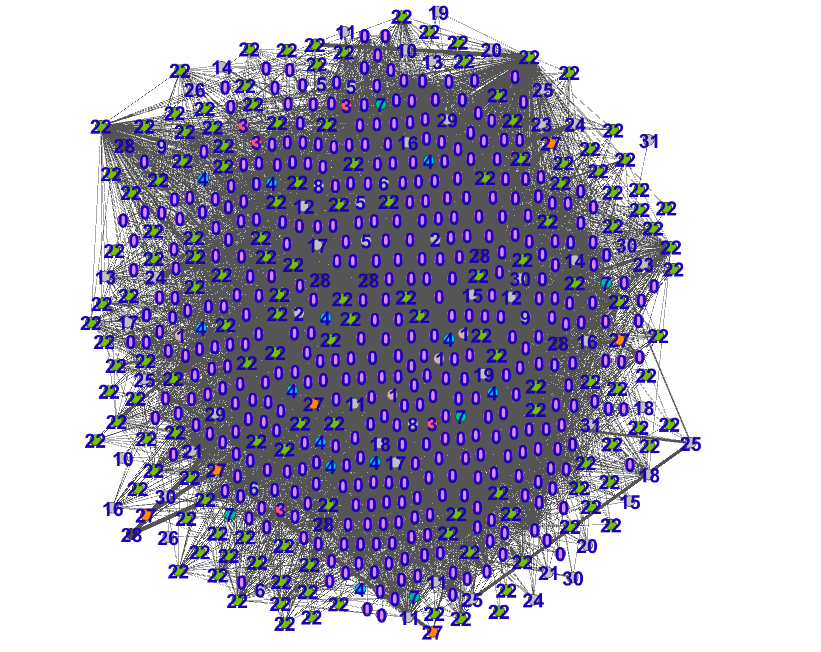


图4-7社团可视化(32个社团)

因此，本研究采用k-means算法对图中的节点进行聚类分析。在进行聚类分析之前，我们首先将使用node2vec算法学习节点特征，生成维度为128的特征向量，node2vec学习特征的具体步骤如下：

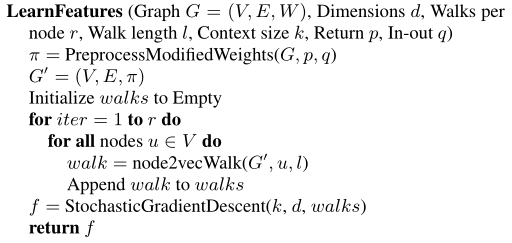
首先，walks用来存储随机游走，先初始化为空；

然后，一共要进行r次循环，每一次循环要为图中每个节点都生成一个长度为l的游走序列；

接着，第iter次循环中对所有节点都利用node2vec算法生成一个随机游走序列walk，然后将其加入walks；

最后，得到所有节点的r个随机游走序列后，利用SGD方法得到每一个节点的向量表示。

特征学习算法伪代码如下：



迭代过程如图4-8所示：

|  |  |
| --- | --- |
| https://img-blog.csdnimg.cn/20200705210942517.png?x-oss-process=image/watermark,type_ZmFuZ3poZW5naGVpdGk,shadow_10,text_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L0N5cmlsX0tJ,size_16,color_FFFFFF,t_70 | https://img-blog.csdnimg.cn/20200705211013393.png?x-oss-process=image/watermark,type_ZmFuZ3poZW5naGVpdGk,shadow_10,text_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L0N5cmlsX0tJ,size_16,color_FFFFFF,t_70 |
| 初始化输入节点 | 第1次迭代 |
| https://img-blog.csdnimg.cn/20200705211044231.png?x-oss-process=image/watermark,type_ZmFuZ3poZW5naGVpdGk,shadow_10,text_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L0N5cmlsX0tJ,size_16,color_FFFFFF,t_70 | https://img-blog.csdnimg.cn/20200705211236192.png?x-oss-process=image/watermark,type_ZmFuZ3poZW5naGVpdGk,shadow_10,text_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L0N5cmlsX0tJ,size_16,color_FFFFFF,t_70 |
| 第2次迭代 | 第9次迭代 |
| 图4-8 K-means 的迭代过程 | |

随后，利用K-means算法对得到的节点的128维向量进行聚类，我们在这里将节点聚集为4类，即将K值设为5，基本步骤如下：

1）设定聚类中心（即分为K类）；

2）算出每一个点到这K个聚类中心的距离，然后把该点分配给距离它最近的一个聚类中心；

3）更新聚类中心，算出每一个类别里面所有点的平均值，作为新的聚类中心；

4）给定迭代此次数，不断重复步骤2,3，达到该迭代次数后自动停止.

算法迭代次数的可视化结果，将其映射到二维平面上如图6所示。通过图中的信息，我们将数据分为4类时，其中大部分节点被分类为第1类（黄色），第2、3类次之，第4类（黑色）最少。

经过对节点数据的分析，四类数据的中心点特征均与Top50用户协作次数的特征有较高的相似度，即该中心型分布网络的用户协作中心在其Top50的用户中，其他用户分布相对靠近网络的边缘。

1. 相关工作

GitHub 可以说是程序员开源精神之所系。在GitHub开源社区构建的网络中，程序员和编程爱好者探寻自己感兴趣的项目、分享源代码、交流学习,逐步形成联系紧密的群,又称簇(Cluster)或社区(Communitie)。对Github网络进行社区划分或聚类,使得社区内部个体之间的交往比与社区外部的交往更频繁,有助于迅速找到需要的资料,加强精准交流和合作，有利于社会计算任务的完成。

目前，关于社区划分和聚类的国内外研究成果丰硕。用于复杂网络的社区划分方法主要分为四种：以节点为中心的社区划分、以群组为中心的社区划分、以网络为中心的社区划分和以层次为中心的社区划分[2]。本研究尝试了多种社区划分方法，最后重点研究了三种方法。

Lizhi Zhang等开发了一种基于种植分区模型的非参数贝叶斯公式的原则性方法来推断网络中的分类社区[3]，这种方法成功地在网络中找到了具有统计意义的分类模块。这与模块化最大化等替代方案不同，后者在人工和经验示例中都系统地过度拟合；而前者没有明显的分辨率限制，只要有统计证据，就可以发现任意数量的社区。该研究提出的公式符合模型选择程序，并揭示分类性是否实际上是主导的大规模混合模式，展示了如何确定更可靠的分类。

Tigo P以随机块模型及其度校正和重叠推广为基础[4]，使用贝叶斯推理从网络数据中提取大规模模块化结构。该研究通过增加贝叶斯层次结构来避免欠拟合，通过关注非参数公式以避免过拟合的推理方法，并支持模型选择。最后阐明了网络中模块结构可检测性的基本限制。

Blondel等提出了一种基于模块化优化的启发式方法，这是提取大型网络社区结构的简单方法[5]。它在计算时间方面优于所有其他已知的社区检测方法。此外，通过模块化度量，该方法检测到的社区质量非常好。因此，我们在研究中使用该方法进行Github社区划分并取得了良好效果。

1. 结论

Github网络作为一个复杂网络，应该具有包括小世界网络的性质。目前相关研究表明，Github合作网络的聚集系数大，平均路径小，满足小世界特征。从一个社会学理论视角出发，小世界网络符合“六度分隔理论”。在Github合作网络中，这样的小世界具体指任意两个开发者可以通过非常小的常量跳数相关联。同时，聚集系数也是重要衡量指标，即一个节点和另外两个节点都有边，则这两个节点之间有边的概率。显然，技术领域相似的开发者合作相对来说更为密切，所以Github合作网络具有较大的聚集系数[6]。

此外，Github合作网络中节点的度应该大致满足幂律分布，与真实世界有很大相似性。通俗地说，节点的度符合极少数节点拥有极多边，即“二八定律”，从而导致网络中边的分布呈现长尾特征。

该项目在生产中具有实际意义，并且比较广泛。通过对用户角色的划分，可以确定该用户的技术栈，并最终用于Github用户画像的生成。同时，本项目对于开源项目的发起人和决策者具有指导意义[7]。或是企业对开发者的面试和招聘，都可以使用类似的分类的技术来对开发者进行评估和遴选。

参考文献

[1]王姗姗. GitHub用户数据分析与研究[D].大连理工大学,2018.

[2]Lei Tang, Huan Liu 著，文益民，闭应洲译. 社会计算-社区发现和社会媒体挖掘[B]. 机械工业出版社, 2013,12.

[3] Statistical inference of assortative community structures Lizhi Zhang, Tiago P. Peixoto Phys. Rev. Research 2 043271 (2020) https://dx.doi.org/10.1103/PhysRevResearch.2.043271

[4] Bayesian stochastic block modeling. Tigo P. Peixoto Chapter in “Advances in Network Clustering and Blockmodeling,” edited by P. Doreian, V. Batagelj, A. Ferligoj (Wiley, 2019).

[5] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, Etienne Lefebvre, Fast unfolding of communities in large networks, in Journal of Statistical Mechanics: Theory and Experiment 2008 (10), P1000。

[6]王玲,董振伟,张庆芝.开源软件项目发起人个体特征对项目绩效的影响——基于GitHub的实证研究[J].科技管理研究,2021,41(20):124-130.

[7]蒋竞,吴秋迪,张莉.开源社区评审过程度量体系及其实证研究[J].软件学报,2021,32(12):3698-3709.DOI:10.13328/j.cnki.jos.006127.