微博热门话题分析

马洋　　项思琪
（1.电子工程系，无 58，2015011181
2.电子工程系，无 56，2015011120）

摘要：在新浪微博中，每天都有许多热门话题引发微博用户的广泛讨论。分析这些热门话题的特征，如粉丝在地理上的分布、粉丝间组成的社交网络等，对于发现新浪微博中有相似兴趣的用户群体，理解信息在社交网络中传播的行为有重要意义。本次实验中我们将实现一个网络爬虫，爬取恋与制作人超话的粉丝信息和讨论用户信息，从多个角度提取该话题的特征，将结果可视化地展现出来并做进一步的讨论和分析。

关键词：网络爬虫；社区发现；可视化

1 实验方案设计

1.1 网络爬虫

网络爬虫是一个自动提取网页的程序，它为搜索引擎从万维网上下载网页，是搜索引擎的重要组成。网络爬虫按照系统结构和实现技术，大致可以分为通用网络爬虫、聚焦网络爬虫、增量式网络爬虫、深层网络爬虫。实际的网络爬虫系统通常是几种爬虫技术相结合实现的。以通用爬虫为例，它从一个或若干初始网页的 URL 开始，获得初始网页上的 URL，在抓取网页的过程中，不断从当前页面上抽取新的 URL 放入队列，直到满足系统的一定停止条件。

本实验要求我们实现网络爬虫，爬取一个时效热门话题的粉丝信息和带话题微博的用户信息，并分析该热门话题的特征。考虑到新浪对于网页版微博的维护很全面，我们选择了难度较小的移动版微博进行数据爬取。

1.2 Gephi

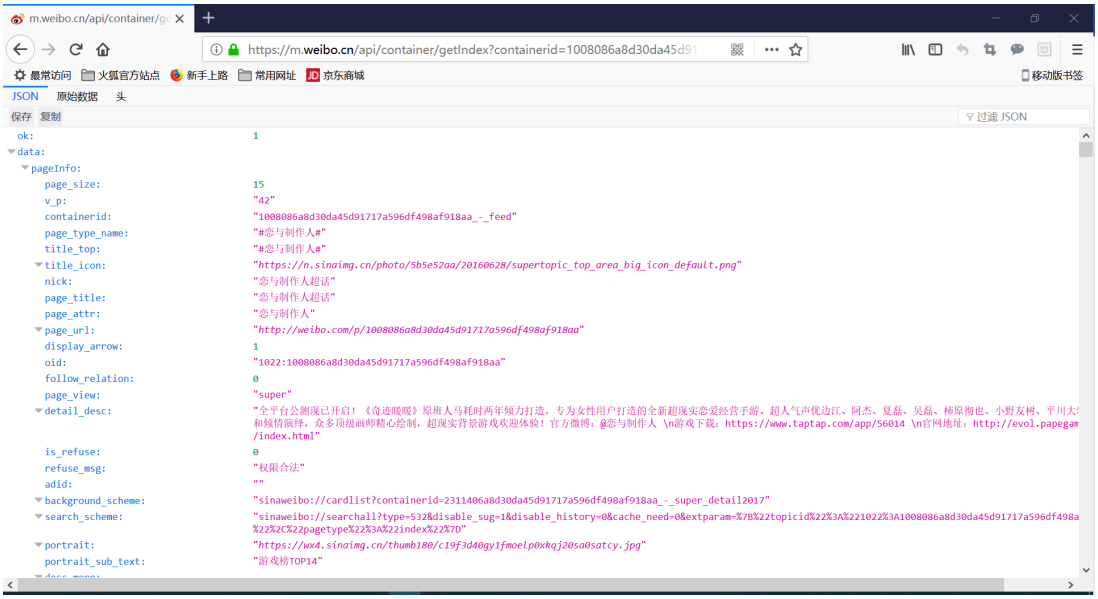
Gephi 是一款开源免费跨平台基于 JVM 的复杂网络分析软件，其主要用于各种网络和复杂系统，动态和分层图的交互可视化与探测开源工具，具体可以用作探索性数据分析，链接分析，社交网络分析，生物网络分析等。

Gephi 是基于模块化的思想设计的，其真正的体现了高内聚低耦合的思想。每个模块负责不同的职责，比如有专门负责图形结构的 Graph，有专门用于布局的 Layout。本次实验中我们将主要利用 Gephi 来完成社区发现和可视化的工作。最初这部分我们是打算用 Louvain 算法来做，但后来考虑到数据规模比较小，选择可视化的软件会更直接和方便。

2 实现方法描述

2.1 微博爬虫实现

爬虫的主要原理是从若干个初始网页的 url 开始，抽取获得新的 url 放入队列，并解析 url 网页内容来获取需要的信息。在一开始已经说过，由于网页版微博（www.weibo.com）的数据爬取复杂性，我们本次的爬虫工作主要针对移动版微博（www.weibo.cn）。在移动版微博中，我们可以通过修改 url 获得 json 格式的网页内容，而 python 中可以直接解析 json 格式，这在很大程度上降低了数据处理的难度。在我们修改了 url 之后，html 中的信息将会以 json 的格式显示出来。如超级话题“恋与制作人”的主页第一页的 url 地址 https://m.weibo.cn/api/container/getIndex?containerid=1008086a8d30da45d91717a596df498af918aa_-_feed&page=1，用火狐浏览器打开可以直接显示如下：



在本问题中，需要爬取的信息包括超级话题的粉丝信息及粉丝的关注/粉丝列表、超级话题的发帖用户信息及用户的关注/粉丝列表、超级话题的帖子的时间信息。其中超级话题的粉丝列表可以从超级话题的名人堂中获得，本次实验中我们用到的 url 展示如下：

恋与制作人名人堂 url：

https://m.weibo.cn/api/container/getIndex?containerid=2311406a8d30da45d91717a596df498af918aa_-_super_newfans&page=%d

恋与制作人帖子 url：

https://m.weibo.cn/api/container/getIndex?containerid=1008086a8d30da45d91717a596df498af918aa_-_feed&page=%d

id 为 XXX 的用户的关注列表 url：

https://weibo.cn/XXX/follow?page=%d

id 为 XXX 的用户的粉丝列表 url：

https://weibo.cn/XXX/fans/page=%d

在具体的爬虫实现中，我们主要分为以下几个步骤：

1、获取微博 cookie，模拟登录

在爬取微博网页的时候，访问 m.weibo.cn 需要通过登陆获取 cookie，然后访问相关页面时服务器根据 cookie 确定访问者是谁，返回相应的信息。在本次实验中，为了防止

因访问过于频繁而被封号，我们手动注册了多个微博账号，轮流爬取微博网页。由于 cookie 可能会过期，因此实验五中在浏览器上登陆，然后将 cookie 复制到爬虫里进行访问这中方法是行不通的。我们选择的新方案是在每次开始爬取网页前，首先运行 redis_cookies.py 来实时获取当下的微博 cookie，完成模拟登陆。

2、初始化任务

在实现过程中，我们利用 Asyncio 和 Redis 模块实现了多线程异步 IO。

Asyncio 是 python 中一个用于实现高并发的模块，而 Redis 是一个 key-value 数据库，用于存储网页 url 和相应的任务名。Redis 的模块采用的是动态链接库的方式，可以启动的时候加载，也可以在运行时加载。

针对超级话题用户的爬虫任务，在初始化的时候将指定页数（此处默认为 500 页）的网页 URL 放入到 Redis 数据库中，并设定任务名方便后续操作。针对超级话题粉丝的爬虫任务也是同理，在初始化的时候将指定页数的网页 URL 放入到 Redis 数据库中。

程序整体的运行框架由几个 python 文件构成，文件清单及各个文件的具体作用展示如下：

(1) redis_cookies.py：用于实时获取微博 cookie，完成模拟登陆过程，为爬取网页做好准备。

(2) init.py：将默认页数（此处为 500 页）的名人堂或发帖信息的 URL 放入 Redis 数据库中，等待下一步操作。

(3) start.py 和 Weibo_cn_async.py：从 Redis 数据库中取出任务，并根据需求放入新的任务到 Redis 数据库中，同时将获取到的信息从“生产者”发送到“经销商”处 (“生产者”和“经销商”的具体含义之后解释)。

(4) consumer.py：“消费者”从“经销商”处获得爬取到的信息，保存成 json 文件共后续处理。其中 start.py，Weibo_cn_async.py 和 consumer.py 共同构成 Kafka 分布式消息系统，能够以高吞吐量处理所有动作流数据。

3、Kafka 与数据爬取

在数据爬取过程中，我们依然利用 Redis 实现类似栈的装入和推出过程，在不同的任务中推出指定任务名的网页 URL。其中用户信息任务名为 user，关注列表任务名为 follower，粉丝列表任务名为 fan。

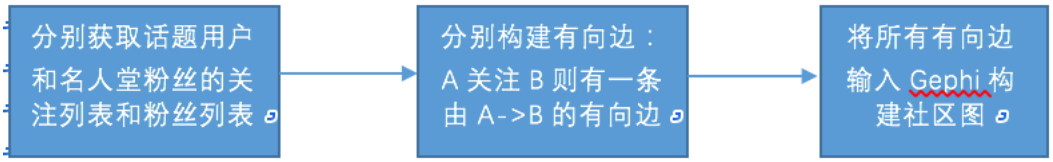
在记录网页爬虫爬取到的内容时，我们需要用到 Kafka。Kafka 是一种高吞吐量的分布式发布订阅消息系统，通过 O(1)的磁盘数据结构提供消息的持久化，即使对数以 TB 的消息存储也能够保持长时间的稳定性能。刚刚初始化任务中我们提到过的“生产者”，“经销商”，“消费者”对应的其实就是 Kafka 中的“producer”，“broker”和“consumer”。

“Broker”是 Kafka 集群包含一个或多个服务器，或者简单理解，可以把它看成消息汇集的中心，“Producer”发布消息到此处而“Consumer”从此处读取信息。“Topic”指发布到 Kafka 集群上的消息类别，物理上不同 Topic 的消息分开存储，但逻辑上用户只需指定消息的 Topic 即可生产或消费数据，并不需要关心数据存于何处。解析 json 文件获取到相应信息后，weiboasync.py 中 producer 会把数据发至指定 topic 的 broker 中，之后 consumer.py 中的 consumer 会从指定 topic 的 broker 中获得相应数据存储成 json 文件，即可用于展开后续数据分析。

2.2 Gephi 与社区分割

在数据分析部分，本次实验指导书提出了两个基本要求。一是分别统计关注话题的粉丝和讨论用户的地点分布,年龄分布和性别分布；二是分别分析粉丝和讨论用户各自组成的社交网络。第一个要求比较基础，只需要对爬虫获取的 json 文件做基本的数据清洗工作即可，不在此赘述。第二个要求涉及社区分割原理，最初我们打算使用 Louvain 算法来解决这个问题，但后来发现要处理的数据规模并不大，直接应用可视化的方法就能解决了。

本次我们借助 Gephi 这款网络分析软件来进行社区分割，下面我们以构建有向图形式的社交网络为例，给出具体的操作流程：



在构建有向边的过程中，考虑到话题用户和名人堂粉丝的关注、粉丝列表中有许多与话题无关的用户，为了使研究范围更集中，我们选择只在话题范围内构建关系图。例如对于名人堂粉丝 A 的某一个粉丝 B，如果 B 也是此话题的名人堂粉丝，则建立一条 B->A 的有向边，否则忽略此粉丝 B。

Gephi 要求的输入的数据格式为：节点 1id 节点 2id。

而经过数据清洗的 json 文件只能够得到简易版的粉丝或者关注列表。以粉丝列表为例，每行的格式为：用户 id 粉丝 1id 粉丝 2id ……

为了达到 Gephi 要求的输入格式，我们另写了一段代码来处理粉丝列表和关注列表，代码框架如下：

```
class Graph { ... };
Graph::Graph(int n) { ... }
Graph::~Graph() { ... }
bool Graph::loadInfo(string input) { ... }
bool Graph::buildGraph() { ... }
bool Graph::writeResult(string output) { ... }
void main()
{
    string input = "lyzzr_follow_ming.txt";
    string output = "lyzzr_follow_ming";
    Graph g(3222);
    g.loadInfo(input);
    g.buildGraph();
    g.writeResult(output);
}
```

设计 Graph 类用于数据的输入、输出，图的建立：

```
class Graph
{
public:
    Graph(int n);
    ~Graph();
    bool loadInfo(string input);
    bool buildGraph();
    bool writeResult(string output);
private:
    int N;                          // 用户数
    vector<double> user;            // 话题下用户
    vector<double>* fans;           // 每个用户分别的 粉丝/关注
    int** A;                        // 邻接矩阵A(i,j)=1代表两人有 粉丝/关注 关系
    int count;
};
```

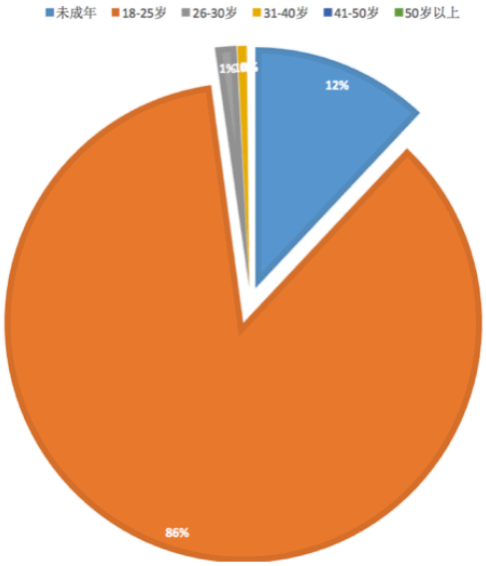
分别处理粉丝列表和关注列表后，合并并用 excel 去重，再输入到 Gephi 中即可构建出社区关系图。实现上述功能的完整代码保存在网络构建文件夹的 buildGraph.cpp 中。

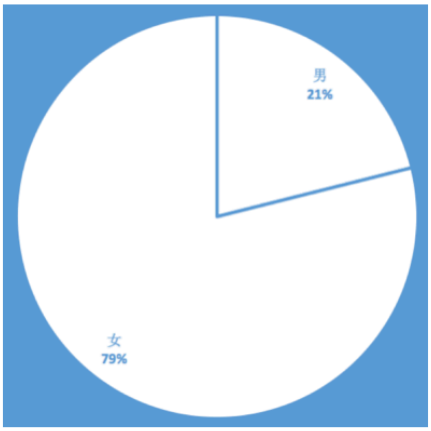
3 实现结果分析

3.1 话题用户信息分析

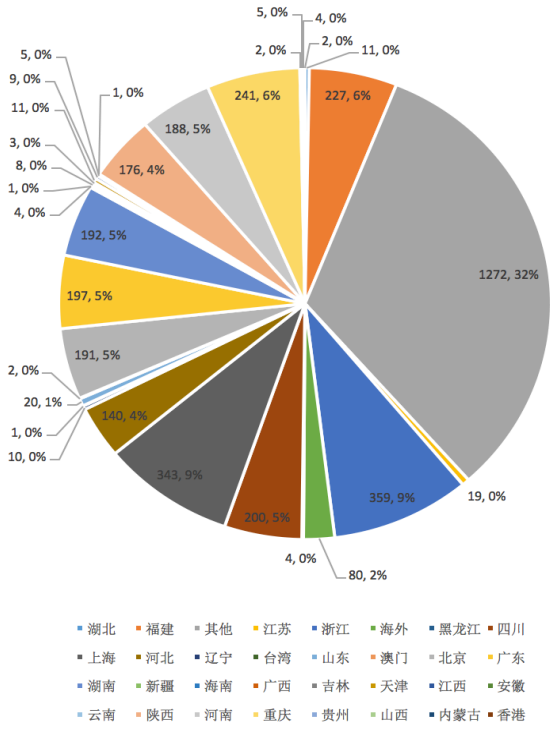
话题用户即是指在超级话题中发帖进行讨论的用户，可以通过爬取超话中的帖子来确定他们的 id，并进一步进入他们的主页来获得更详细的信息。由于超话中的帖子是按照最后评论时间排列的，因此在爬取的过程中会有很多重复信息，经过去重操作之后，我们得到了 3928 名话题用户的信息，分析整理如下：

(1)年龄分布与性别分布：

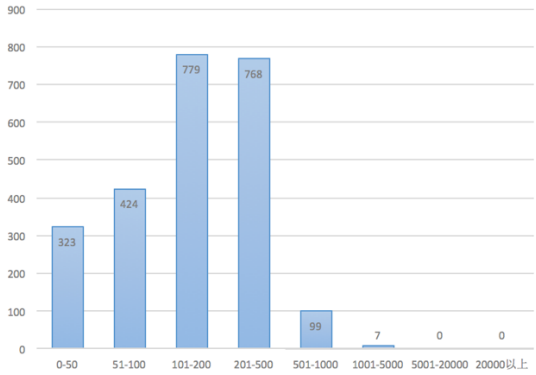




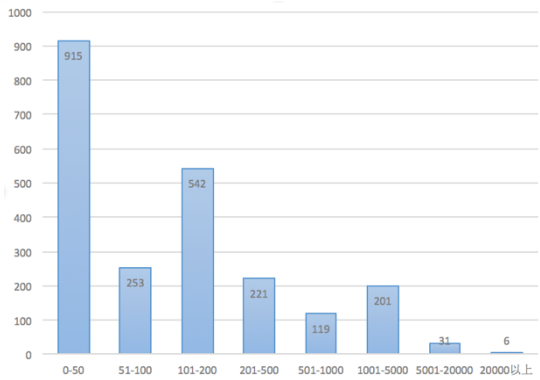
(2)地域分布：



(3)关注数分布：



(4)粉丝数分布：

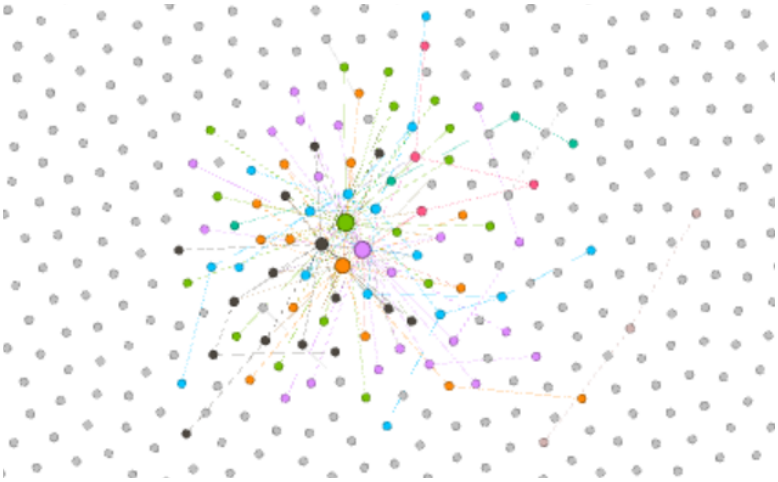


从上述统计结果可以看出，“恋与制作人”超级话题用户的年龄集中在 18-25 岁，也有一部分未成年用户，但 25 岁以上的用户寥寥无几。用户中 79%都是女性，剩下 21%的用户资料显示为男，但由于微博默认的性别就是男性，所以实际的女性比例应该还会更高一些。

上述结果和该话题本身的特征非常契合，因为恋与制作人本身就是一款针对女性开发的手机游戏，主要的受众就是 15-25 岁的女性。而微博上的超级话题是该手游的衍生产物，自然而然也就带上了游戏玩家本身的特点。

除了年龄和性别分布之外，该超级话题用户很多都没有填写自己所处的地点，而填写者中来自上海，北京和广东的比较多。大部分用户的关注数和粉丝数都在 500 以内，也有少量用户的粉丝数超过 5000，属于“恋与制作人”超级话题中的大 V 人物。

在社区分割部分，我们最初只使用了简单的无向图，即 A 和 B 之间只要 A 关注 B 或者 B 关注 A，二者之间就有一条连线。将数据输入 Gephi 中，得到的分割结果如下：

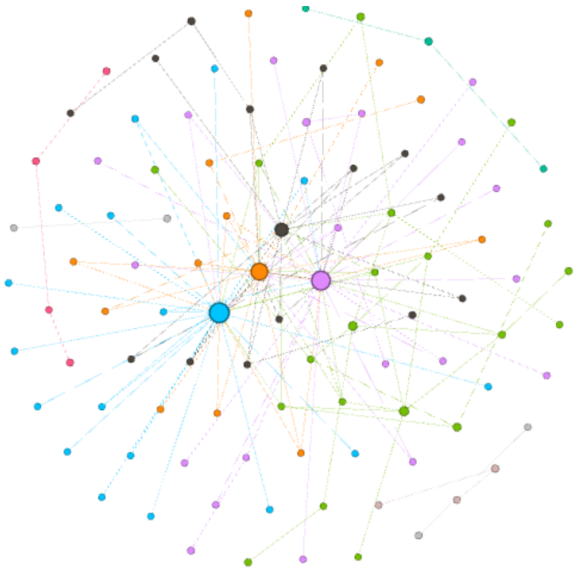


可以看到孤岛的数目非常多（上图中我们还截掉了一些孤岛），说明很多超话用户之间是不存在粉丝或者关注关系的，她们大多数人只是单纯地参与话题讨论。不过从上图中也能明显看到，存在一些较大的节点和其他节点之间关系密切。这些大的节点中有“恋与制作人”手游的官方微博，还有一些针对游戏的攻略组微博。除此之外，为游戏中男主配音的几位配音演员也收到了很多超话用户的关注。

在验收的时候助教提示我们可以对关注关系和粉丝关系做一些区分，试验看看能不能得出其他结果。因此之后我们又基于超话用户的关注列表和粉丝列表构建了有向图，绘制

时，A 关注 B 则边从 A 指向 B。图中的节点大小按照节点的入度进行排序，节点越大表示该微博账号粉丝数越多。节点颜色采用 Gephi 自带的模块化工具划分不同模块后上色，在节点数量不大时可以大致代表社区关系。

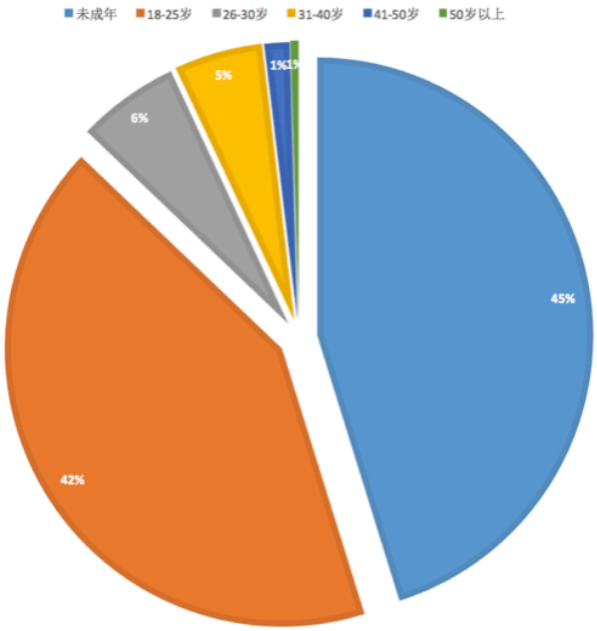
此外，在绘制有向图时我们去除了孤岛节点，最终得到的结果如下：

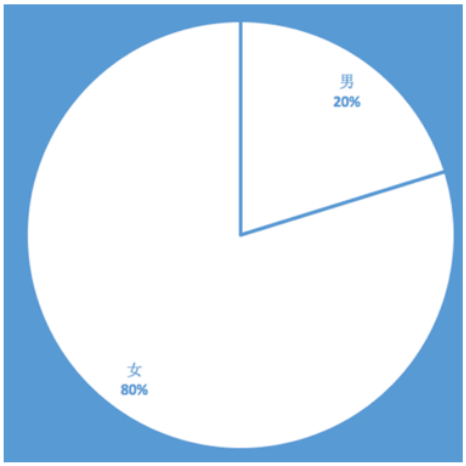


3.2 话题粉丝信息分析

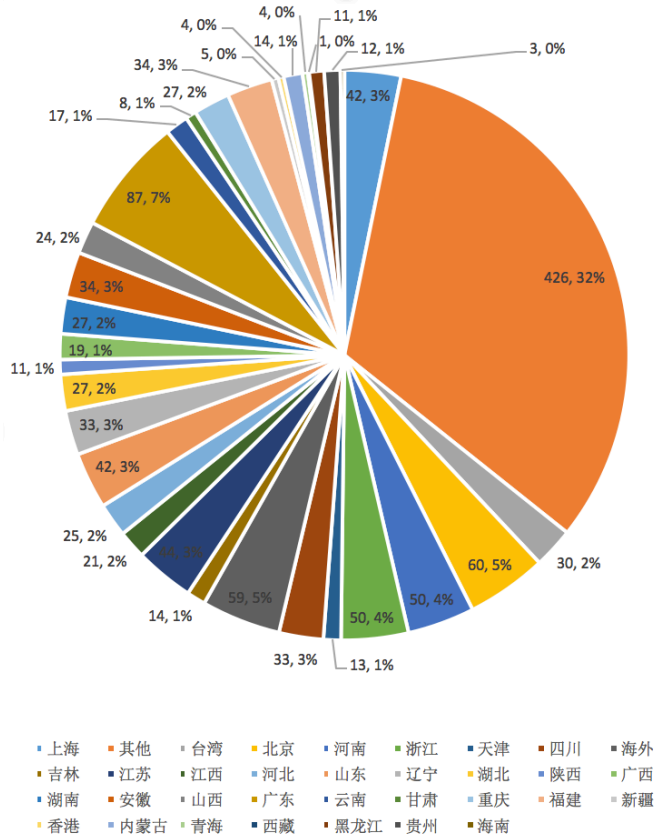
话题粉丝指关注相应超级话题的粉丝，获取途径是超话名人堂的新进粉丝栏目，可以进一步进入他们的主页来获得更详细的信息。我们最终得到了 1311 名粉丝的信息，分析整理如下：

(1) 年龄分布与性别分布：

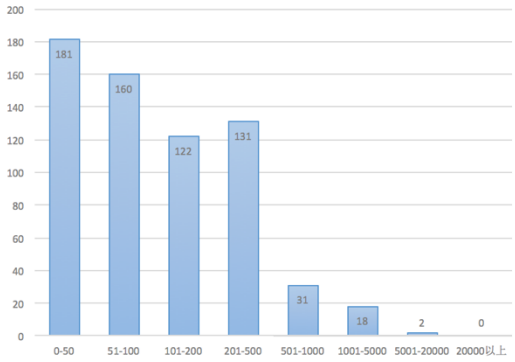




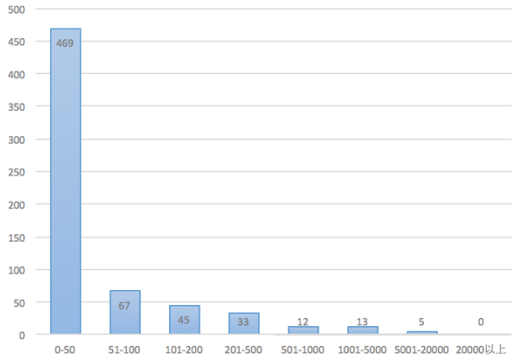
(2) 地域分布：



(3)关注数分布：



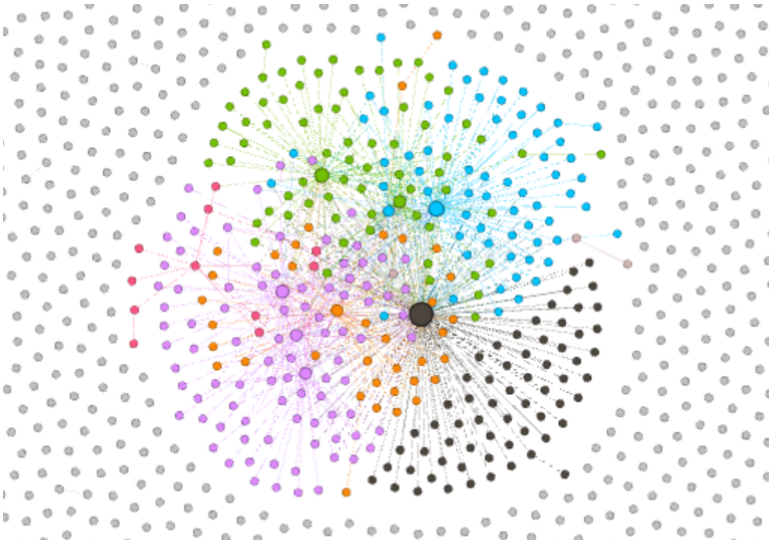
(4)粉丝数分布：



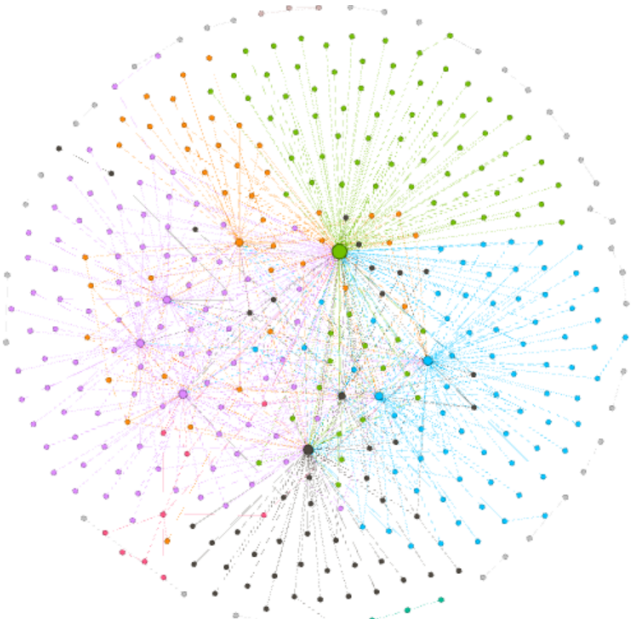
从上述统计结果可以看出，“恋与制作人”超话粉丝中的未成年人数更为庞大，但用户年龄集中在 25 岁以下，且绝大多数为女性这两个特征还是没有改变的。除了年龄和性别分布之外，该超级话题粉丝中有 32%都没有填写自己所处的地点，而填写者的地域分布比较均匀，北京、河南、浙江和海外的粉丝相对更多一些。

大部分超话粉丝的关注数都在 500 以内，粉丝数都在 100 以内，也有少量用户的粉丝数和关注数超过 1000。

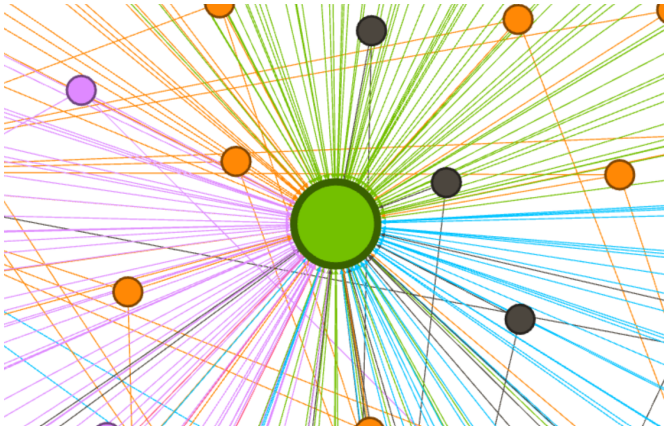
采用和分析超话用户类似的方法，我们也对“恋与制作人”超话粉丝进行了社区分割分析。在无向图中得到的社区分割结果如下页图所示（截掉了一部分孤立节点），可以看出相比于超话用户，超话粉丝之间的社区关系更为密切，向大节点汇集的趋势也更为明显。图中最大的黑色节点即为“恋与制作人”手游的官方微博，很多超级话题的粉丝同时也是官博的粉丝，这是一个非常合理的现象。



之后我们又基于有向图重新进行了社区分割，得到的结果如下：

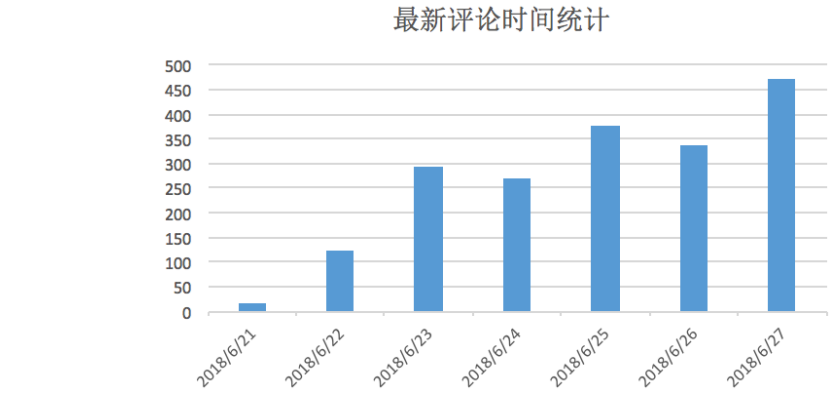


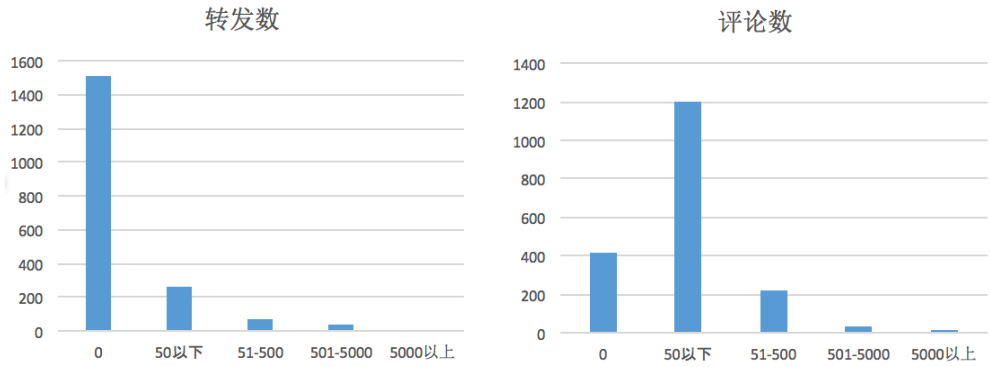
图中的节点越大代表粉丝数目越多，上图中绿色的节点为恋与制作人官方微博账号，具体的细节图展示如下：



3.3 话题讨论帖情况概览

在对话题用户和话题粉丝的信息进行分析之后，我们也对话题中的发帖情况进行了局部性的探究。我们抓取了从 2018.6.21 到 2018.6.27 期间超话中的帖子信息，拿到了 1337 个用户发送的 1883 个不重复帖子。这些帖子的最新评论时间，转发数和评论数统计整理如下：





由于我们所能抓取的数据有限，因此最新评论时间更晚的帖子数量会较多。此外，大部分帖子的转发数和评论数都在 50 以下，超过 5000 转发和评论的高热度帖子比较少。帖子的评论数总体来说会比转发数更多，因为对微博用户而言，评论比转发更为方便快捷。

4 实验总结

本次实验我们小组选择的课题是微博热门话题分析，在实验中也选择了组员个人比较感兴趣的“恋与制作人”超话进行数据爬取和信息整理。这是本学期这门课程的最后一次实验，也是任务量最大难度最高的一次实验。在实验过程中，无论是最初对爬虫的探索，对数据的整理和清洗，还是之后的归类分析，使用 Gephi 来生成并可视化社区分割的结果，都是我们之前不太熟悉的领域，对我们而言是很大的挑战。我们在这次实验中投入了大量的精力和时间，在艰辛之余，也收获了对爬虫技术更深刻的了解，进一步锻炼了自己编写 python、C++等代码的技巧，并对网络分析和社区分割算法有了初步的认识。

当然，除了宝贵的收获之外，本次实验中我们也有一些未能实现的遗憾。在验收时助教曾跟我们提到可以使用深度优先算法或者广度优先算法来爬取数据，这样建立出的网络联系可能会更为紧密。但我们之后发现移动版微博中并不会显示某用户是否关注了某超话，这条线索就很遗憾的断掉了，基于这个结构可以延伸出的一些探索也只能暂时搁置。非常感激老师和助教在实验过程中为我们提供的指导和帮助，但很可惜我们没能实现您们所期待的“意料之外，情理之中”，我们会在这门课程之外继续努力，争取在未来探索出更有价值的结果。

（最终上交的压缩包中含有实验报告和网络爬虫、数据清洗与分析、网络构建等文件夹）