

目录

[1.项目概览 代码目录结构设计与运行环境 4](#_Toc43044614)

[1.1代码目录结构 4](#_Toc43044615)

[1. 2目录及项目运行步骤说明 4](#_Toc43044616)

[1.3项目运行环境 4](#_Toc43044617)

[2.前置准备 项目数据读取与预整理 5](#_Toc43044618)

[2.1代码与算法分析 5](#_Toc43044619)

[2.3 数据分析 6](#_Toc43044620)

[3.建立模型 建立无向图模型 6](#_Toc43044621)

[3.1代码与算法分析 6](#_Toc43044622)

[3.2运行结果与数据分析 7](#_Toc43044623)

[4.实验报告 连通分支数量、规模、电影类别统计 8](#_Toc43044624)

[4.1代码与算法分析 8](#_Toc43044625)

[4.2运行结果 9](#_Toc43044626)

[4.3数据分析 9](#_Toc43044627)

[5.实验报告 连通分支直径计算 9](#_Toc43044628)

[5.1代码与算法分析 9](#_Toc43044629)

[5.2运行结果 11](#_Toc43044630)

[5.3 数据分析 11](#_Toc43044631)

[6.实验报告 连通分支规模、直径、平均星级作图 11](#_Toc43044632)

[6.1代码与算法分析 11](#_Toc43044633)

[6.2 运行结果1 (所有连通分支全部作出的情况) 13](#_Toc43044634)

[6.3 结果分析1(所有连通分支全部作出的情况) 14](#_Toc43044635)

[6.4 运行结果2（只作前后20个分支的情况） 14](#_Toc43044636)

∎[6.4.1 调整后的代码 14](#_Toc43044637)

∎[6.4.2 作图结果2 15](#_Toc43044638)

∎[6.4.3结果分析2（只作前后20个分支的情况） 16](#_Toc43044639)

[7.实验报告 有关周星驰的演艺信息统计 16](#_Toc43044640)

[7.1代码与算法分析 16](#_Toc43044641)

[7.1运行结果1 (共同出演者总共的统计) 17](#_Toc43044642)

[7.2 运行结果2 (共同出演者分别逐个统计) 17](#_Toc43044643)

[8.自选报告 基于网络排序算法的演员影响力排名 18](#_Toc43044644)

[8.1自选题目与课内内容的联系与算法简介 18](#_Toc43044645)

[8.2代码与算法分析 18](#_Toc43044646)

[8.3运行结果 19](#_Toc43044647)

[∎8.3.1 演员影响力排名列表: 19](#_Toc43044648)

[∎8.3.2 PageRank因子密度分布图: 20](#_Toc43044649)

[∎8.3.3 最具影响力前5名演员位置子图: 20](#_Toc43044650)

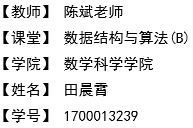
[8.4 算法发现与分析 21](#_Toc43044651)

[9.项目展望 21](#_Toc43044652)

[10.算法各部运行时间简计 22](#_Toc43044653)

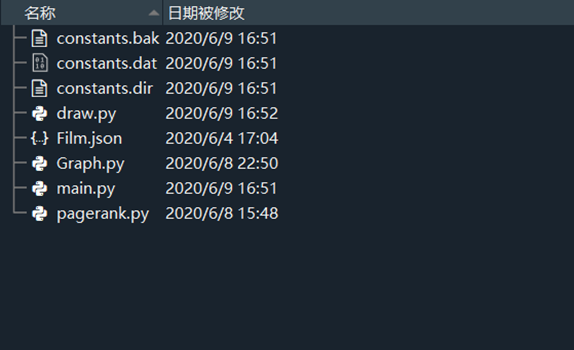
[11.附录 22](#_Toc43044654)

演员的小世界实验报告



# 1.项目概览 代码目录结构设计与运行环境

## 1.1代码目录结构

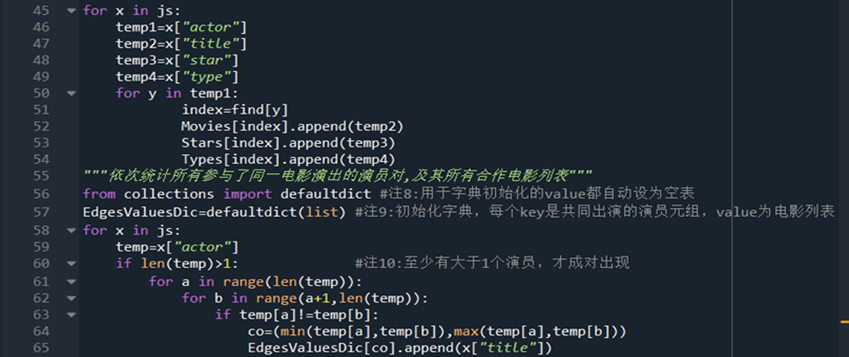


## 1. 2目录及项目运行步骤说明

## 1.3项目运行环境

# 2.前置准备 项目数据读取与预整理

## 2.1代码与算法分析



2.2运行结果[[1]](#footnote-1)

**∎**整理完成了：Actors、Movies、Stars、Types、EdgesValuesDic 5个列表或者字典对象，输出示例如下：

## 

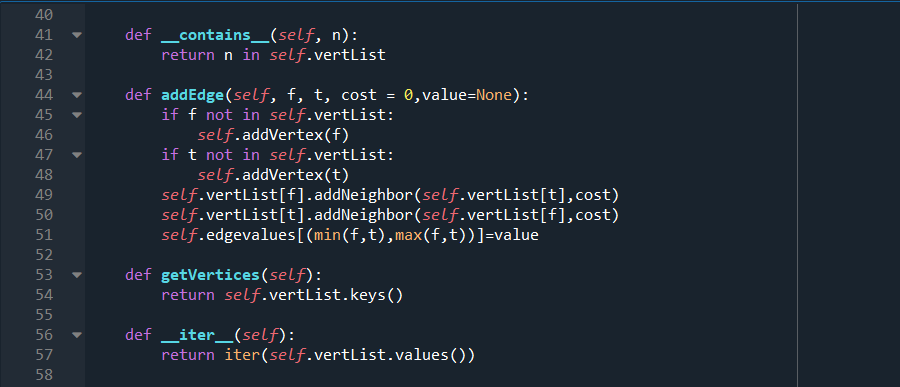
## 2.3 数据分析

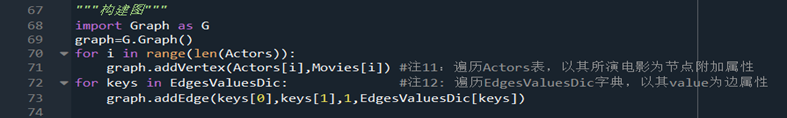
# 3.建立模型 建立无向图模型

## 3.1代码与算法分析

**∎**对应main.py 67行至73行以及Graph.py 7至57行，截图分析如下：

1. **∎**Graph.py模块7至57行：构建图模型，包括节点和图对象，边属性储存在edgevalues字典中：

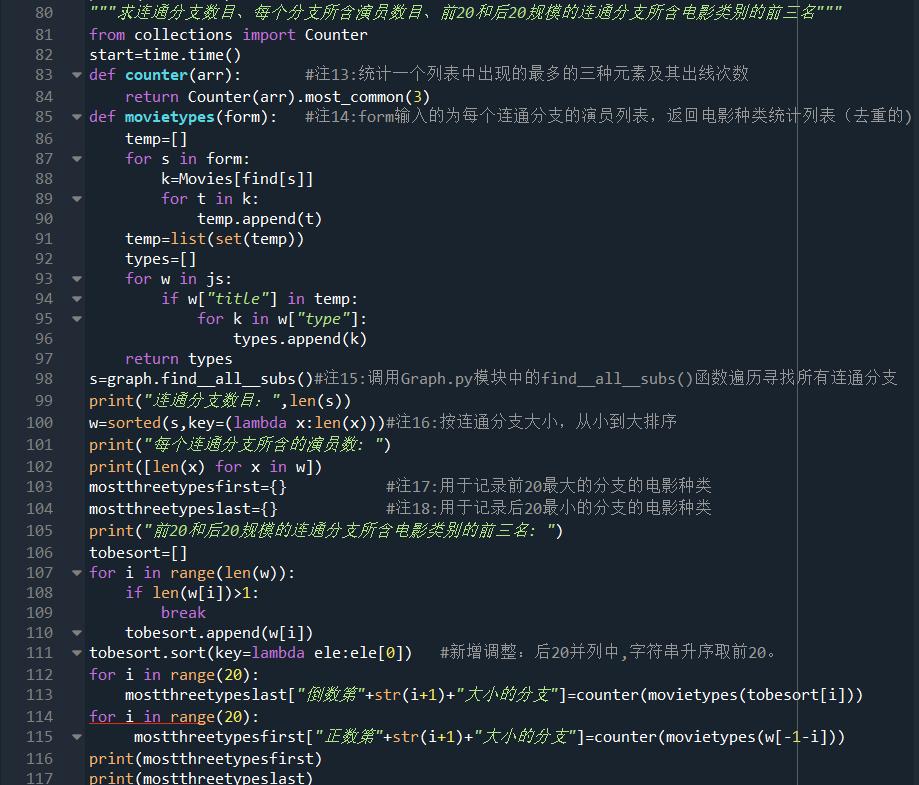
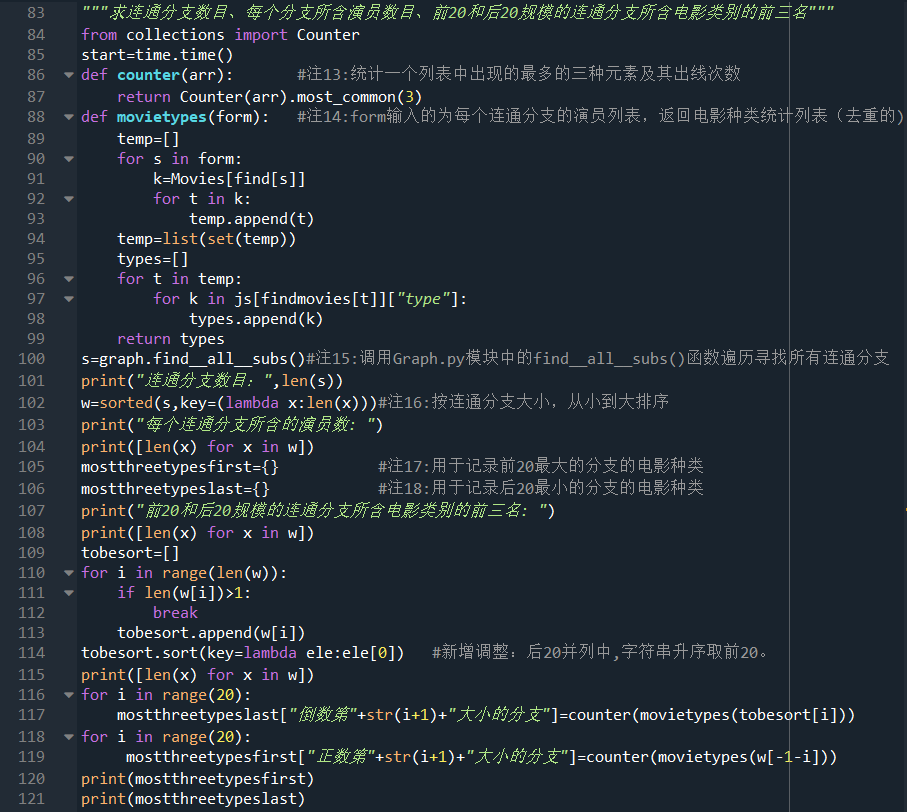
（2）**∎** main.py模块67至73行：利用之前的Actors和Movies表格建立无向图模型。

## 3.2运行结果与数据分析

**∎**运行结果构建了一个以演员为节点，所演电影列表为附加属性；有共同出现为边，以共同出演电影列表为边附加属性的自定义的无向图Graph对象。

# 4.实验报告 连通分支数量、规模、电影类别统计[[2]](#footnote-2)

## 4.1代码与算法分析

（1）**∎** main.py 83至121行：调用Graph.py的find\_all\_subs方法，统计连通分支数、规模、电影类别：

（2）**∎** Graph.py第59至77行: 定义find\_\_all\_\_subs方法，寻找所有连通分支，返回为多重列表。

**∎** 用基于深度遍历DFS算法: 用栈非递归实现的分离连通分支的算法,总运行和统计时间只需2.61s。



4.2运行结果 [[3]](#footnote-3)

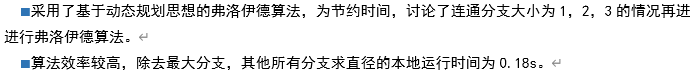
 

## 4.3数据分析

# 5.实验报告 连通分支直径计算

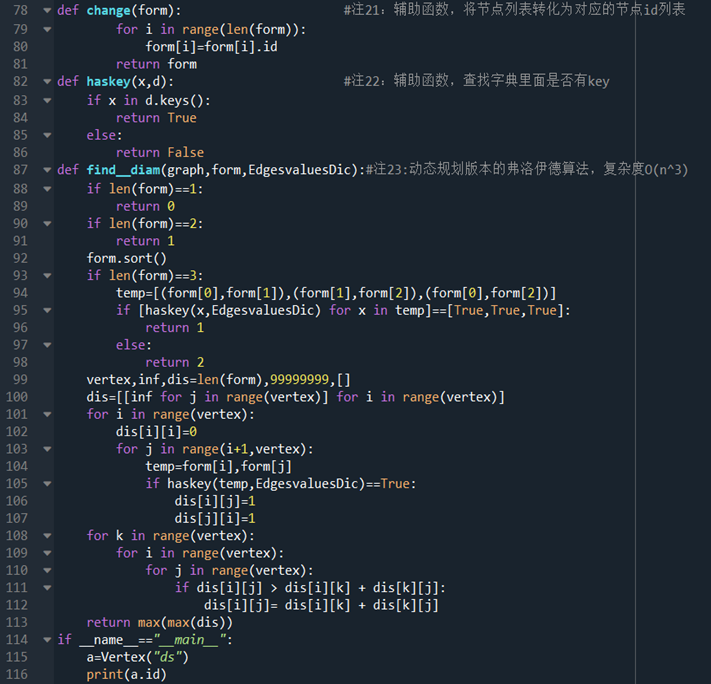
## 5.1代码与算法分析

1. Graph.py部分：

**∎**第78至116行：定义基于弗洛伊德算法寻找任意子图的直径的方法，时间复杂度为O(N\*\*3)

（2）main.py部分：

**∎** 第104行至117行: 调用算法求直径列表，可考虑shelve储存(但事实上时间成本很小，也可不储存)，以方便下次调用直径列表：

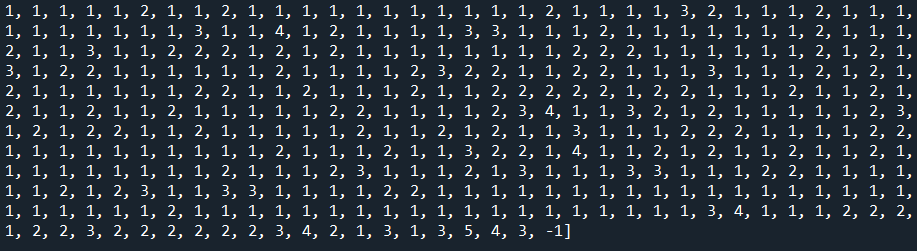






## 5.2运行结果





## 5.3 数据分析

# 

# 6.实验报告 连通分支规模、直径、平均星级作图

## 6.1代码与算法分析

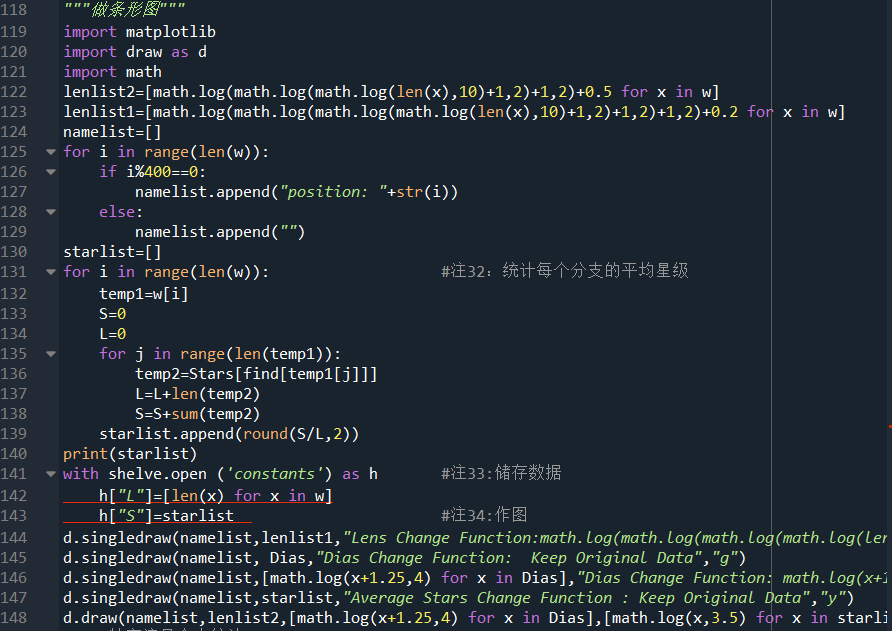
（1）draw.py模块

**∎**定义4种不同功能的作图函数(见6.2运行结果)，在draw.py内作其中1种y轴截断图：

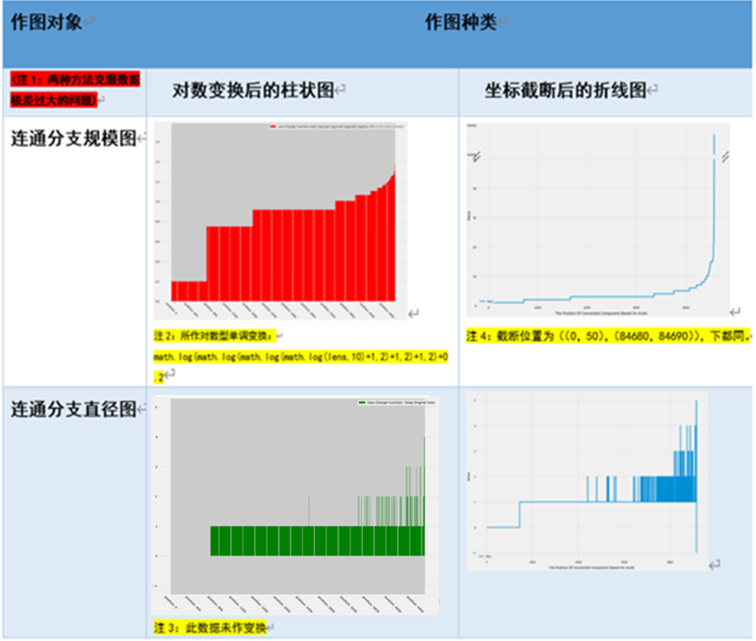




1. **∎** main.py第118至148行：作出对数变换数据的非截断的条形图：

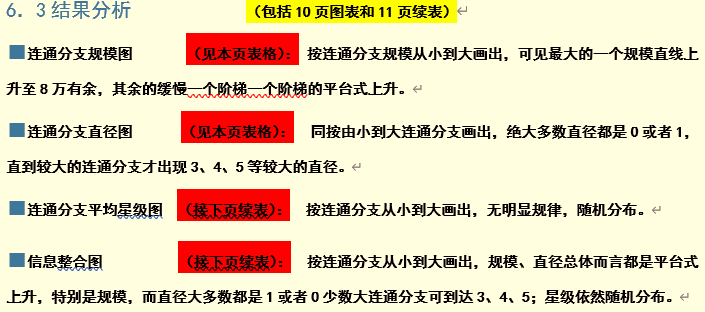


6. 2 运行结果1 (所有连通分支全部作出的情况)[[4]](#footnote-4)

****



## 6. 3 结果分析1(所有连通分支全部作出的情况)



## 6. 4 运行结果2（只作前后20个分支的情况）

### 6.4.1 调整后的代码

在主实验报告和整合py程序中：所作的是全部连通分支的对数变换柱形图，但由于第3题的实际操作要求更新，下也只作出前20和后20分支的对数变换后的柱形图，对应代码74至91行：

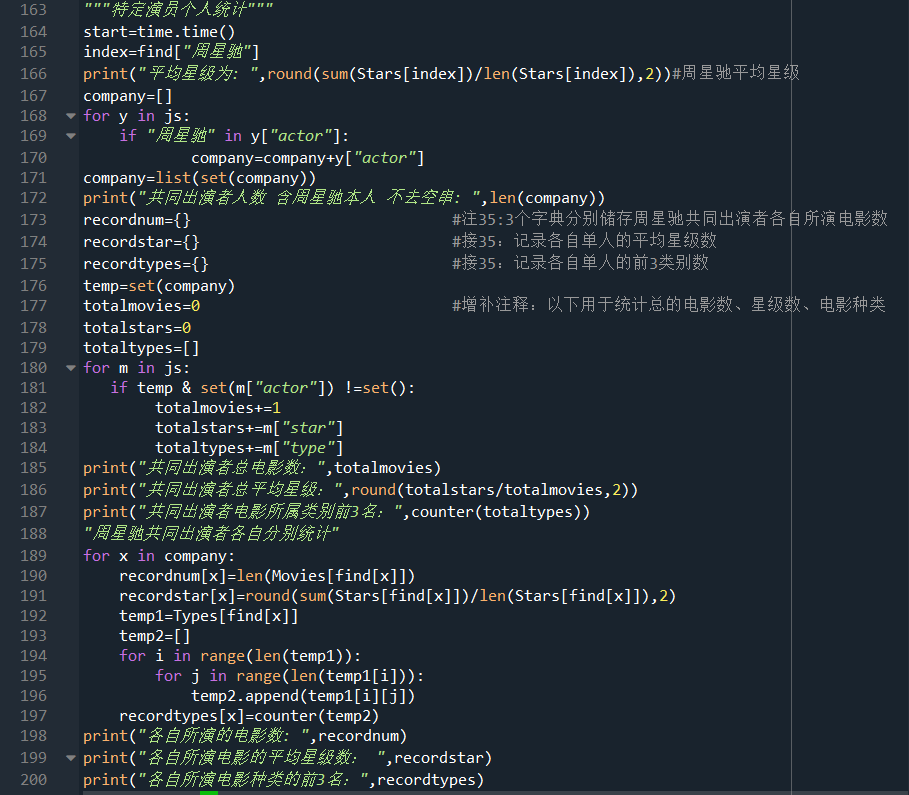
### 6.4.2 作图结果2

### 6.4.3结果分析2（只作前后20个分支的情况）

简析：由6.4.2节运行结果图可见，三个信息的变化趋势，由小分支到大分支，与主报告全部作出的情况完全一致（参见主报告或北大网盘提交全部输出图片文件夹）。

# 7.实验报告 有关周星驰的演艺信息统计

## 7.1代码与算法分析

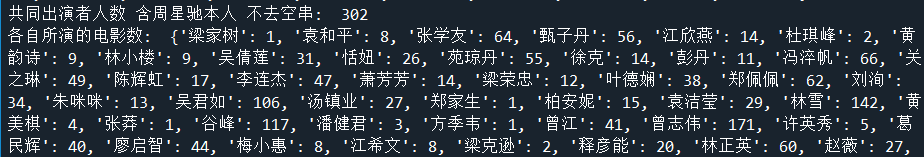
**∎**main.py第163行至第200行：直接利用js表、find字典和Stars表遍历即可得单人演员的统计；遍历js表，可得所有共同出演者的总信息：

## [7.1运行结果1 (共同出演者总共的统计)](#_Toc43027822)

## 7.2 运行结果2 (共同出演者分别逐个统计)

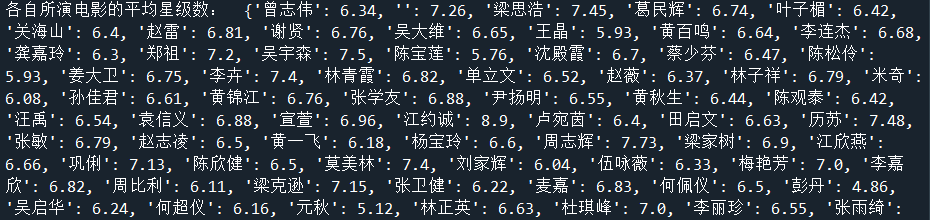
（1）**∎**下展示周星驰平均星级数，共同出演者数目，以及周星驰部分共同出演者们各自所演电影数：

注1：共同出演者各自所演电影数为部分截屏，注意到有一个空串演员，其来源是数据质量缺陷。

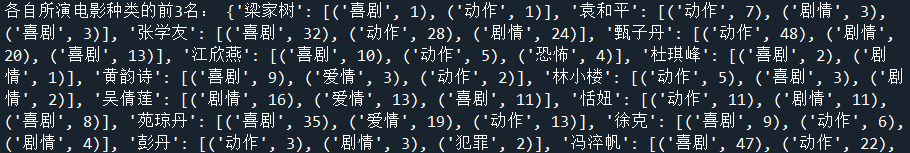


（2）**∎**周星驰共同出演者们的平均星级数：

注2：以下为部分截屏内容

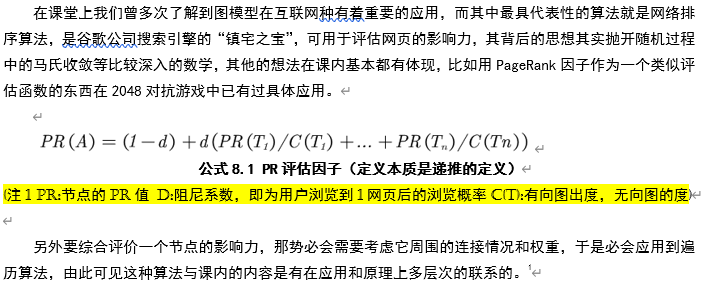


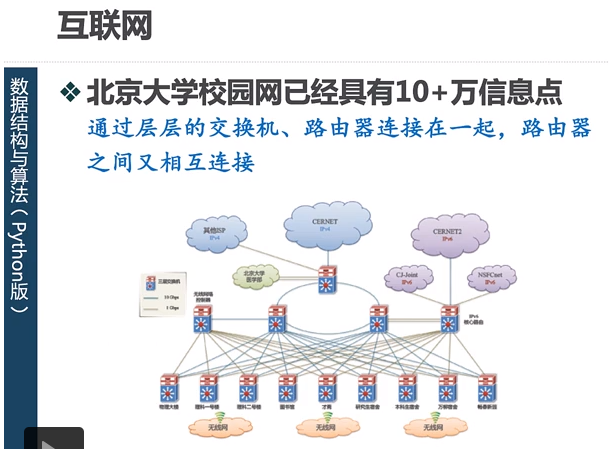
（3）**∎**周星驰共同出演者们的电影种类前3名：

注3：以下为部分截屏内容

# 8.自选报告 基于网络排序算法的演员影响力排名

## 8.1自选题目与课内内容的联系与算法简介

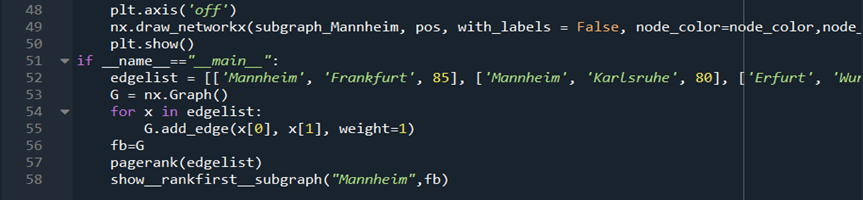
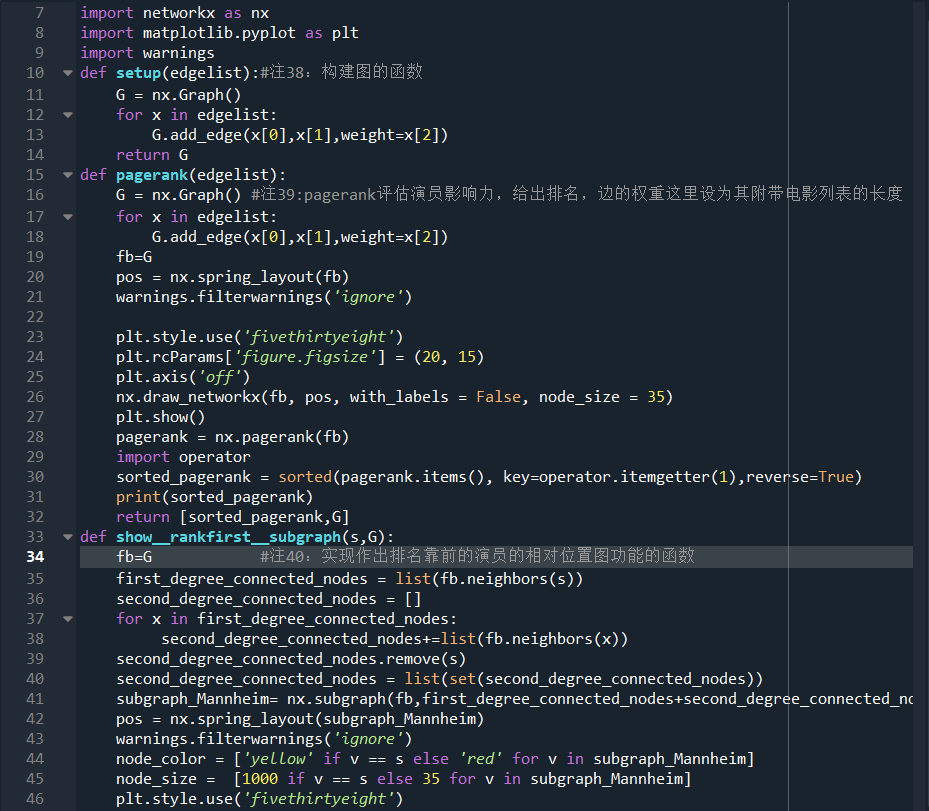
[[5]](#footnote-5)



**图 8.1.1 图 8.1.2**

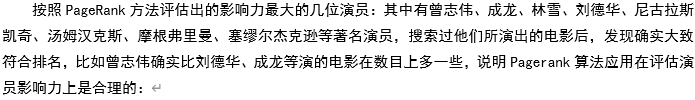
## 8.2代码与算法分析

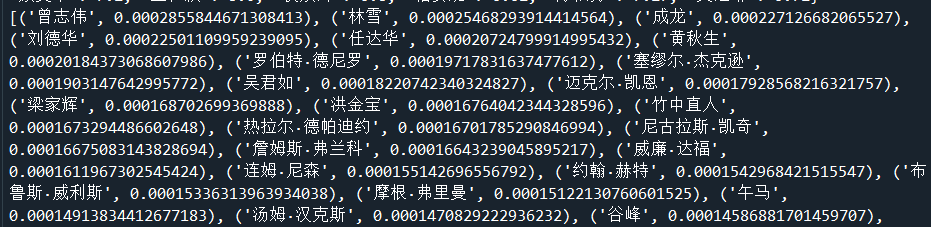




## 8.3运行结果

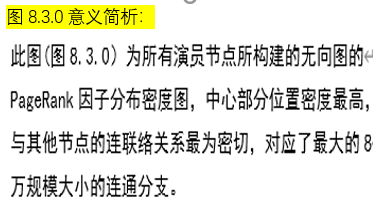
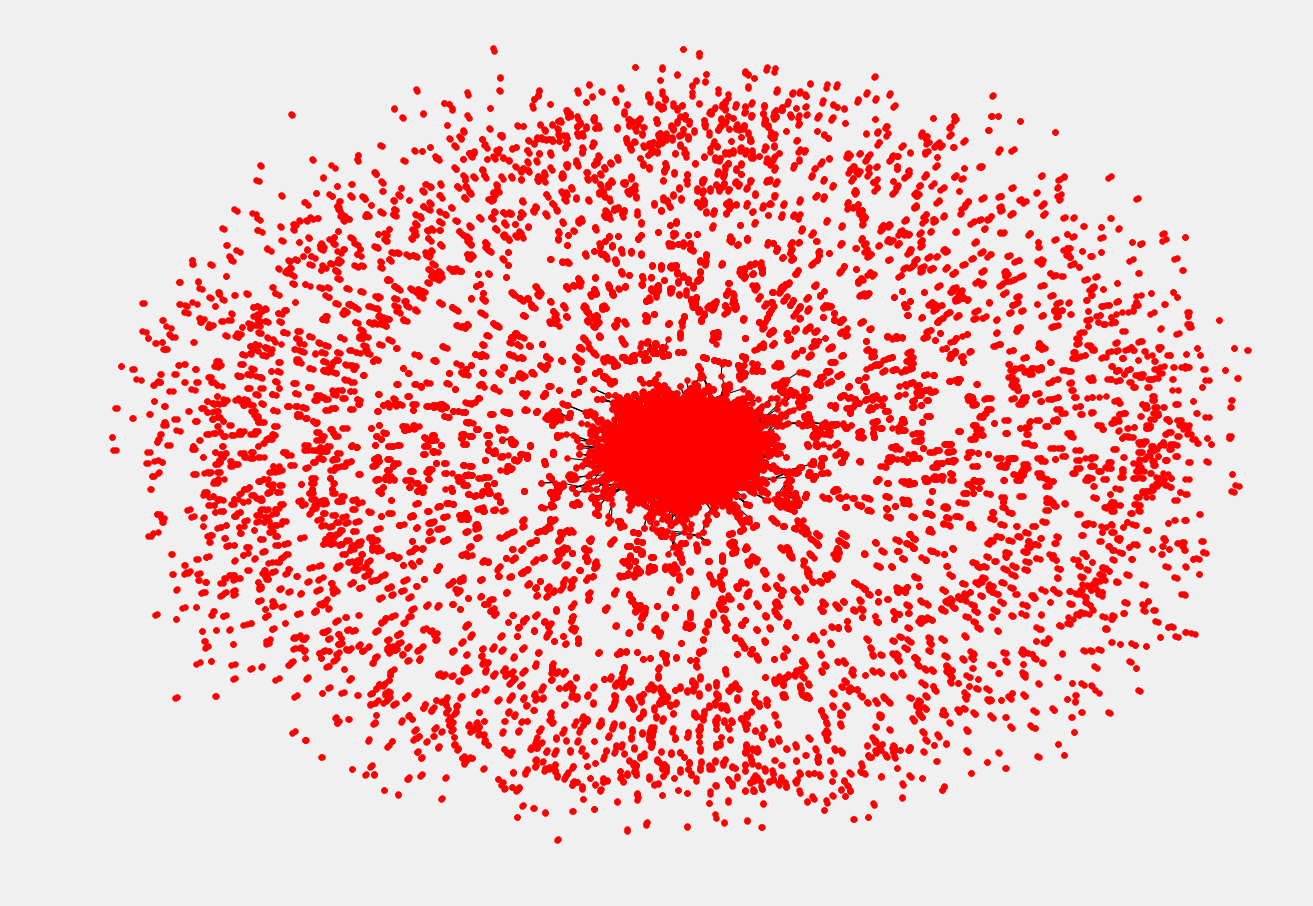
### ∎8.3.1 演员影响力排名列表:





### ∎8.3.2 PageRank因子密度分布图:

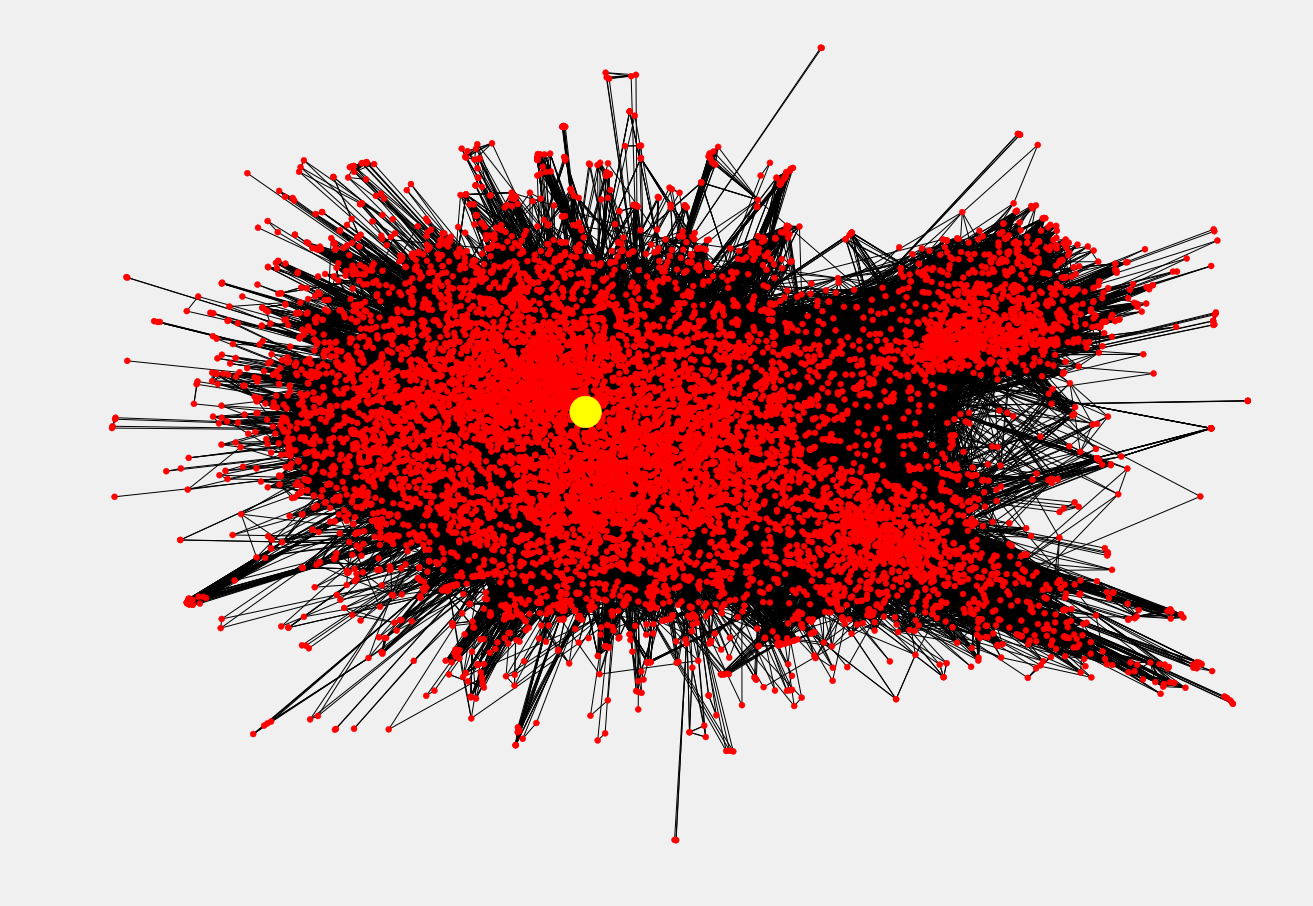
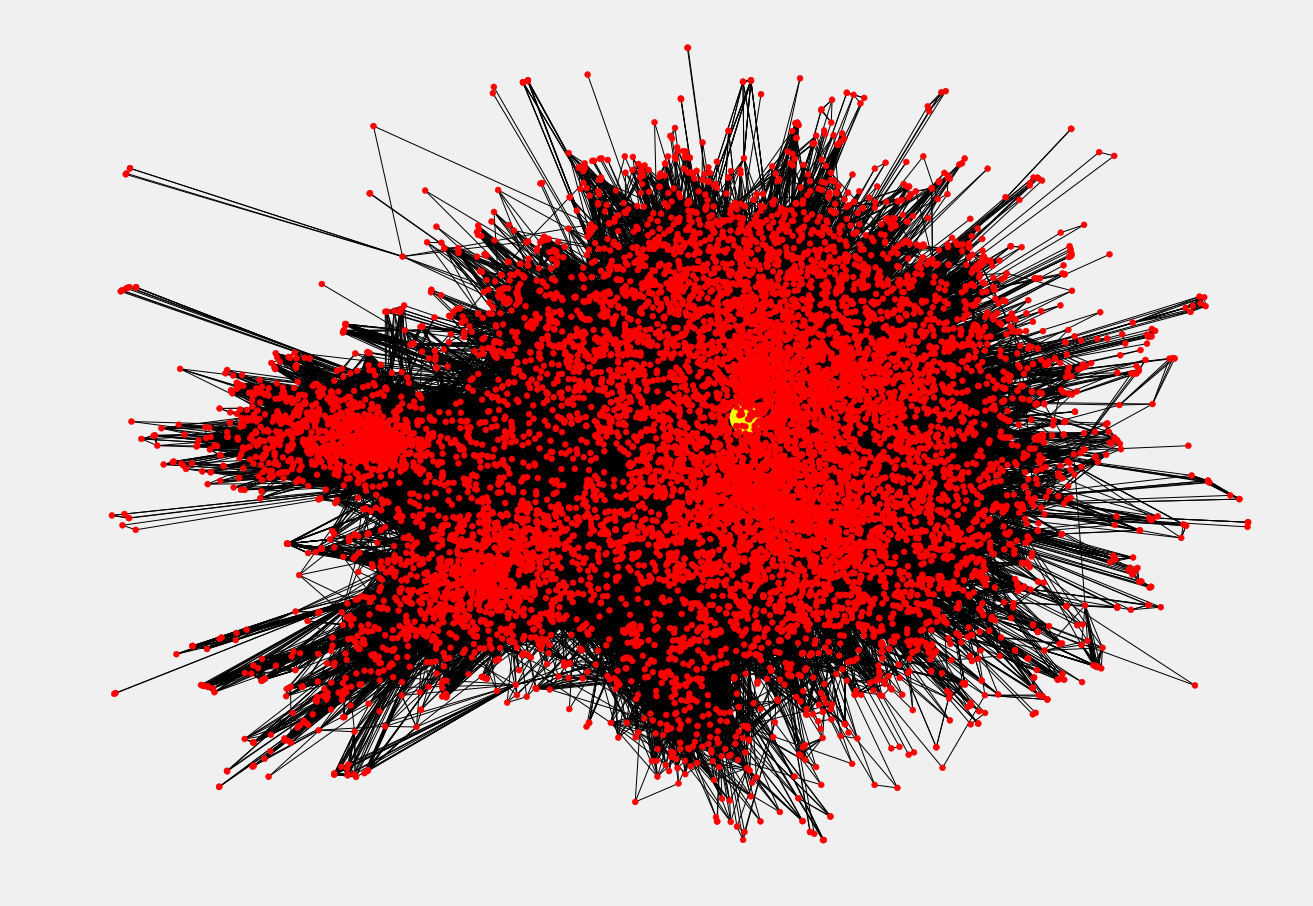




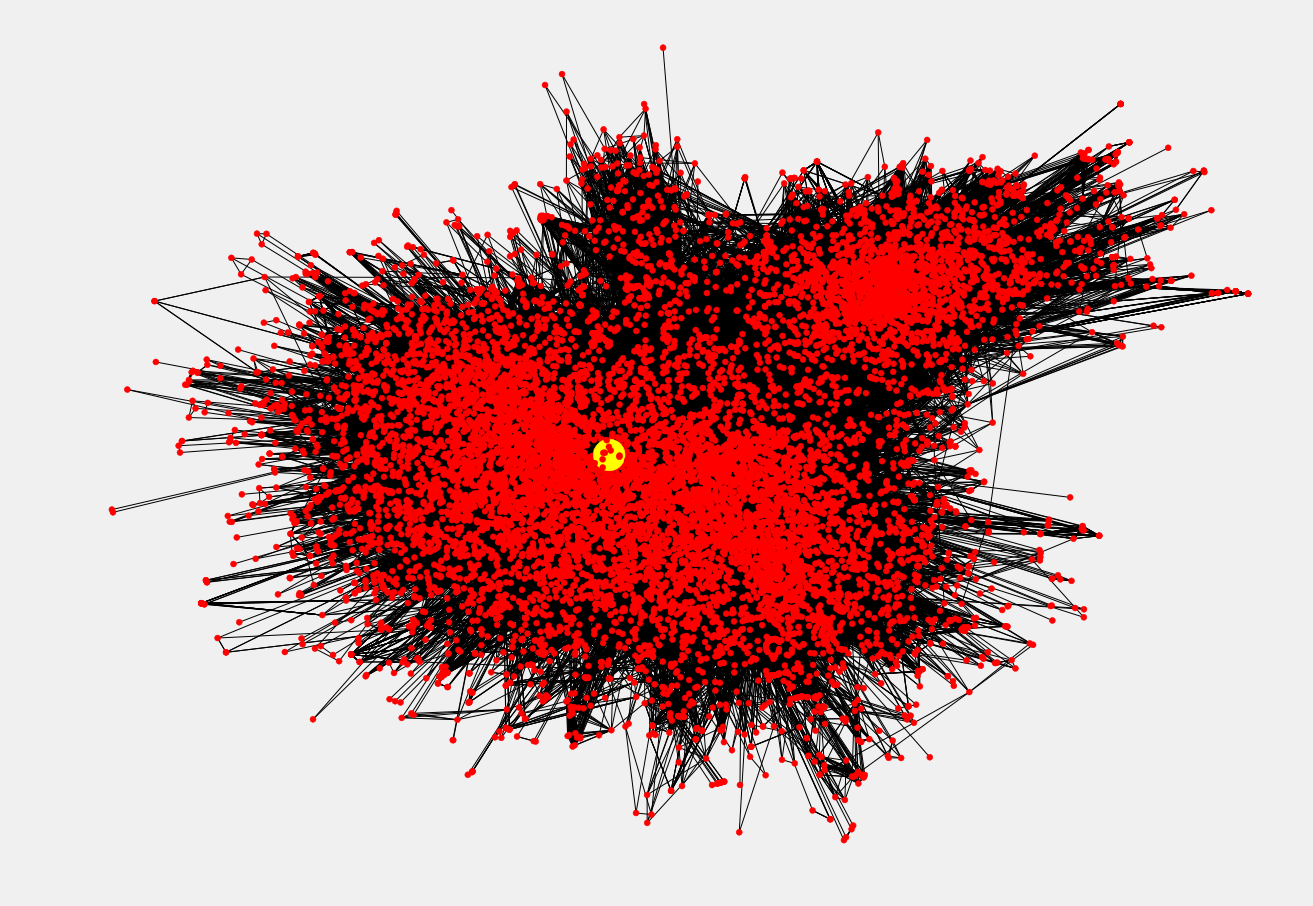
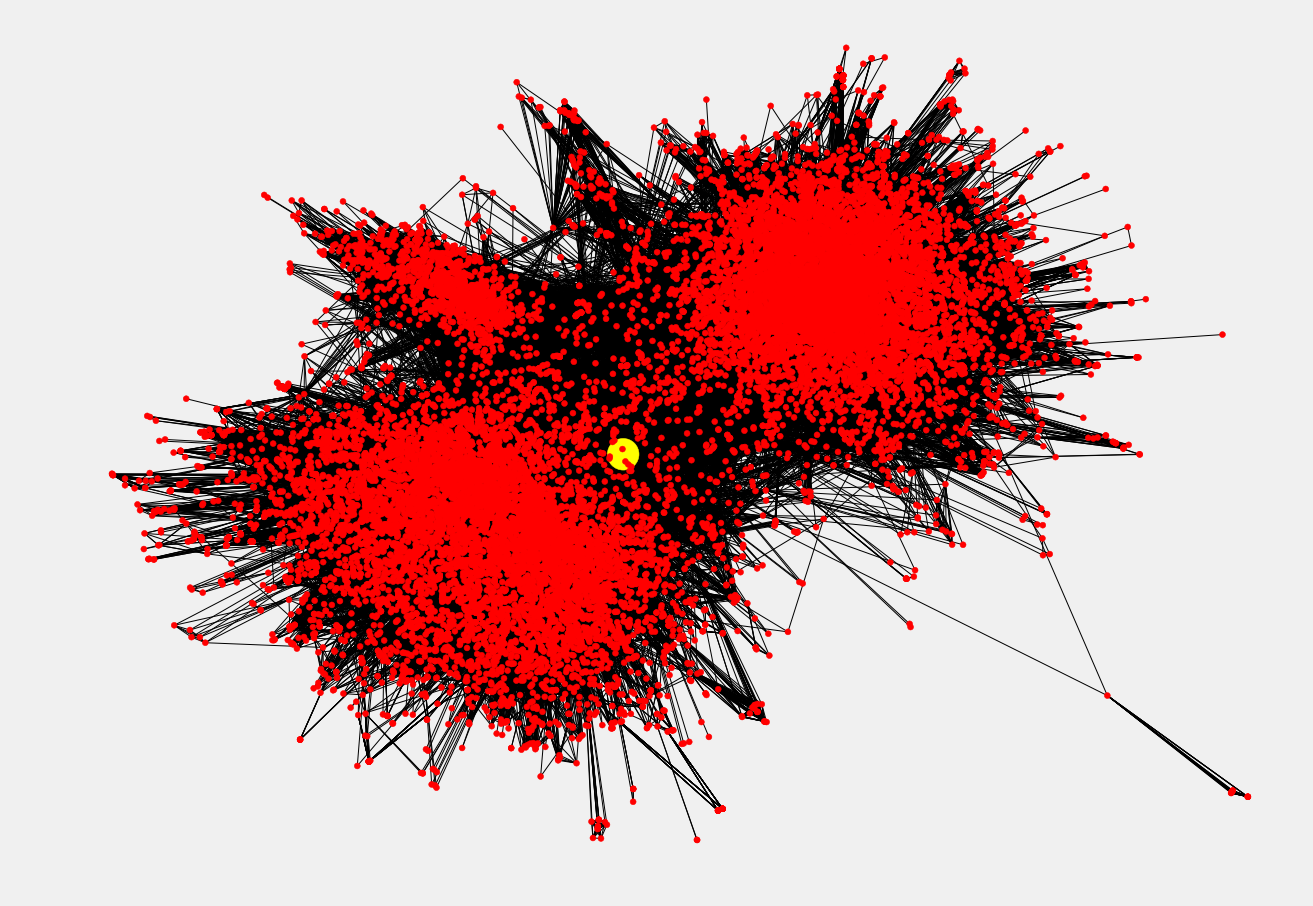
**图8.3.0 PageRank因子相对密度分布图**

### ∎8.3.3 最具影响力前5名演员位置子图:

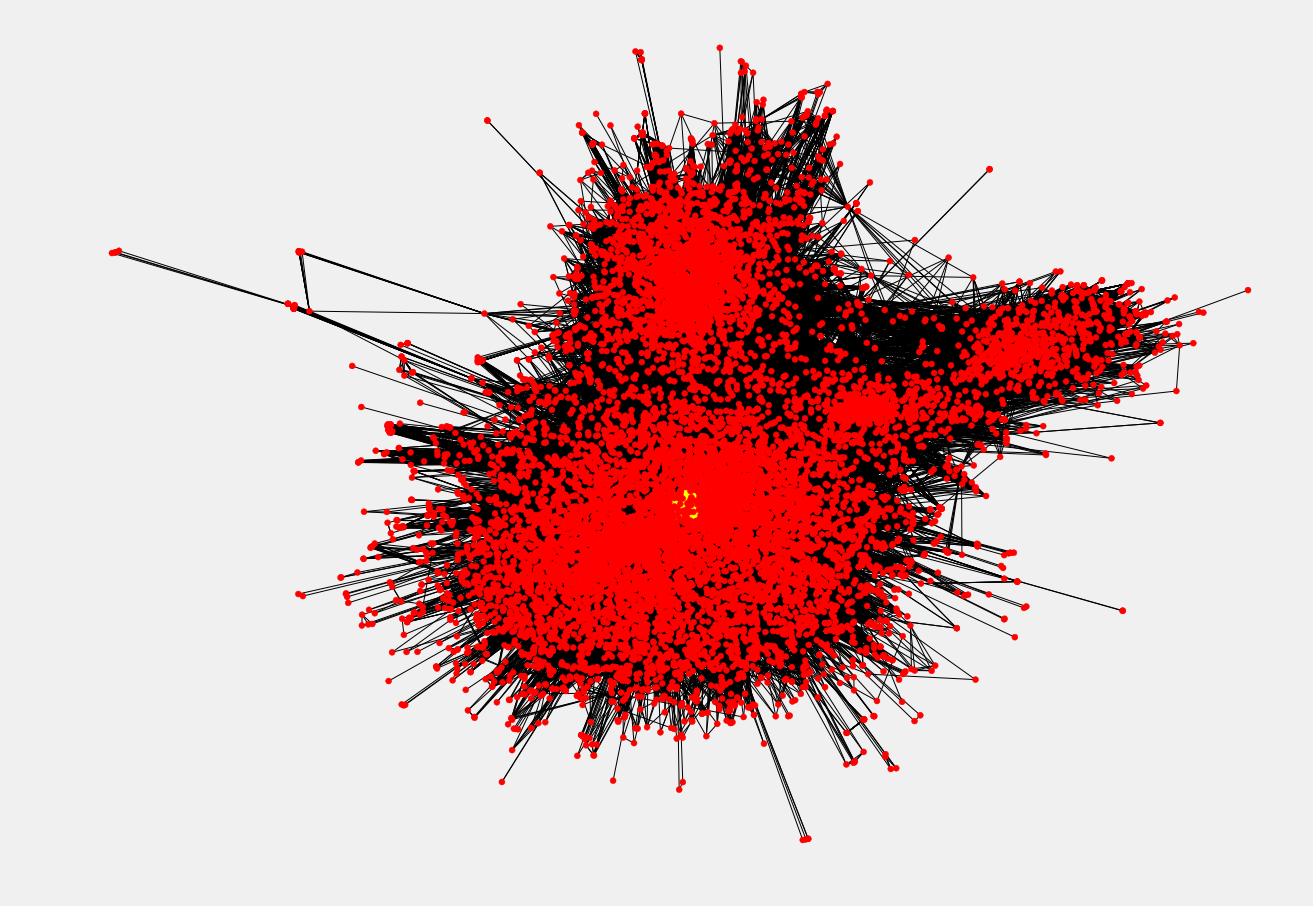




**图 8.3.1 曾志伟 图8.3.2 林雪**



**图8.3.3 成龙 图8.3.4 刘德华**



**图8.3.5 任达华**

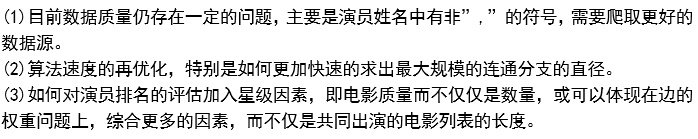
## 

## 8.4 算法发现与分析



# 

# 9.项目展望





# 10.算法各部运行时间简计

# 

# 11.附录

（1）PageRank 算法原理详细介绍：https://blog.csdn.net/leadai/article/details/81230557

（2）PageRank算法简介：<https://www.cnblogs.com/rubinorth/p/5799848.html>



1.  [↑](#footnote-ref-1)
2.  [↑](#footnote-ref-2)
3. 电影类别统计时未考虑个别电影重名的情况，但对种类排名影响不太大。 [↑](#footnote-ref-3)
4.  [↑](#footnote-ref-4)
5.  [↑](#footnote-ref-5)