实验报告

一、图形界面说明

网站： https://jianyutang.github.io/Graph-Theory/ （如果提交的版本出现问题，可以直接访问发布的版本）

- 环境依赖：保持提交的src目录下的结构即可，无需额外配置
- 功能：主页上有两个 [View Details] 入口进入 “电影数据集”页面 和 “论文数据集”页面；两个页面均包含五个算法的展示，其中四个为基本算法，以及一个提高算法（电影数据集的提高算法是图聚类，论文数据集的提高算法是社区发现）；每个算法都在相应的选项卡下有交互按钮（或输入框），点击相应按钮即可在页面右下方的d3的画布上展示相应算法的结果；画布上的结点可拖动、点击（点击结点后会在左侧显示结点对应电影或论文的信息）。

二、豆瓣电影数据集部分

环境：Win10 VS2019(C++17) Python3.7(NumPy计算库)

分析目标

希望通过对电影数据集的建模，分析近年来国际电影界的合作、合拍的情况。

建模方法

以电影为结点，有共同导演或演员的的电影间存在边，边的权重在不同场景下采用了三种建模方式

- $w_{ij}=5d+a$ ，其中 d 是共同导演数，a 是共同演员数；
- $w_{ij}=\frac{1}{5d+a}$ ；
- 无权图。

第一种权重的理由主要是考虑到边的权重代表两部电影在电影人这个群体中联系的紧密程度，因为导演往往决定了影片的灵魂和艺术风格，因此赋予了较大的权重，之所以选择 5 ，是因为在实验中发现共同演员数超过5个的电影极少，因此可以体现出导演这一特殊角色的主导地位和作用；第二种度量距离，很自然的选择了倒数；第三种是关注连通性的场景，因此直接采用无权图。

结果分析

1.连通性

首先我试图通过深度优先搜索来确定图的连通性，结果很遗憾（但也情理之中），即整个图是不连通的，其中最大的一个联通支大小为3828个。因此，可以认为有这么多电影是处在这个国际电影的圈子当中的，经过随机挑选查看的结果，这些基本上都是商业电影，而剩下的很多是艺术电影或独立电影，也由此可以明确，国际商业电影的确是一个大圈子，而接下来的分析也以这个最大的联通支为主。

2.最短路径

最短路径的分析中，对边权采用了倒数的建模方式，两部电影对应结点的最短路径衡量了制作两步电影的片方或导演之间如果想要合作，中间介绍、撮合所需的人脉成本。通过在Floyd算法中添加计数，我发现在任意两个电影之间的最短路径长度分布大致如下：

距离	百分比
[0,3]	33%
(3,6]	56%
(6,9]	11%

可以发现几乎任意两点距离都在9以内，而小于等于6的也高达九成。这也说明了，在这个主流圈子内，任意两家制作方想要合作，介数人（简单将人脉成本大致等于）的中位数也仅为4左右，这对于国际合作是非常利好的。

3.最小生成树

同样采用倒数权重建模方式。通过最小生成树，可以知道整个电影圈子是怎样最短地被联通的，即找到重要的边，通过追溯这些树枝边代表的共同导演或演员，就可以了解哪些导演或演员在电影界的合作中扮演了较为重要的角色，拥有较广泛的人脉。

4.介数中心度

这里采用了无权图的建模方法，因为关注的是最短路径的条数而非距离本身。根据介数中心度的定义，这一中心度反应了一部电影在圈子中的重要程度，因为介数中心度大代表经过它的最短路径多，也即这部电影的相关人员、片方在电影圈子里的人脉交集最广，具有桥梁的作用。如果电影界要评选“国际合作奖”或“国际交流奖”，无疑是选择介数中心度高的结点。这样的结点确实非常稀少，大约只占了总量的1%。例如张艺谋导演的《长城》，在图形界面上所反应出来的位置就是中美两支的桥梁，事实上，这部电影也的确是中美合拍的典范作品，无论是导演还是演员，都是具有较高知名度和声誉的电影圈资深选手，这与计算出来的结果高度吻合。

算法：通过深度搜索和定义结合计算。

5.紧密中心度

采用了倒数的权重。与介数中心度不同，紧密中心度的分布就显得平均的多，这也是圈子良好连通性的体现。因为紧密中心度所关注的是总距离的平均，和介数中心度的结果有部分重叠（尤其是数值大的那些结点），它更能反应一个电影在大圈子中所处的小社区的大小，例如在这个数据集中，国产电影还是主流，因此在国产电影这一社区的电影就明显比小一些的国家的电影要有优势。

算法：利用Floyd算法的结果直接按定义计算。

提高算法

采用了第一种权重建模的方法实现了聚类算法中的标签传播算法。因为对图有关的机器学习感兴趣，但是感觉做GNN或者GCN训练数据不够，因此最后选定了了弱监督的聚类。在分析电影界国际合作情况这一大目标下，聚类的目标很直接的就是通过电影间的共同导演、艺人的特征对电影的生产国进行分类。

算法步骤：

1. 计算通过任意两点间紧密程度计算标签转移概率矩阵
2. 大部分通过噪声生成伪标签，少量数据真实标注得到标签矩阵
3. 迭代计算概率矩阵与标签矩阵的乘积

作为弱监督的机器学习方法，人工标注的标签数很少，约只有百分之十五，而体现出来的效果虽然远没有达到精确的程度，但也相当可以，至少有了大致的界限。一开始的实验效果总是很差，后来发现是因为我总是选择前一两百个结点当作有标签的，这样产生了有标注数据分布的不均匀，如果放到神经网络中就是一个有偏差的训练集，这的确会对结果有较大的影响。

最终通过均匀地选取标注数据，达到了百分之70左右的准确率。这个准确率提升的瓶颈在于首先信息量有限，没办法将电影内容、情节等重要参考因素考虑进去，其次还是有标注数据量的不足，从约15%的标注数据到70%的最终正确率，也是比较大的提升了。

可视化

可视化部分选用了三百个结点进行展示。将本地计算的结果直接存入js文件，前端不进行计算，只根据已有数据完成相应展示任务。

三、论文部分

实验环境：WIN10，VS2017，Anaconda 5.1.0

1. 建模

以论文为节点，以共同的关键字构建边，以共同关键字的多少为权重。

2. 基本算法

最短路径：

采用floyd算法，得到任意两个节点间的最短路径。

最小生成树：

Prim算法。

节点中心度：

（1） 介数中心度：定义为某一个节点出现在其他节点对间的最短路径上的次数。利用floyd求得的各点最短路径，对每条最短路径上出现过的点进行次数相加，即可得到。依赖于已求得的最短路径。

（2） 紧密中心度：定义为某一个节点，和其它所有节点间最短路径距离之和；节点越中心，则该指数越小。利用floyd求得的各点最短路径，对关联某一结点的最短路径长度相加，即可得到对于某一节点的紧密中心度。依赖于已求得的最短路径。

连通分量：

该算法主要用在对数据的筛选上。比如，我想对一部分的论文数据做可视化，那么需要一个联通的图，这样展示效果会比较好；否则 ，对于个算法的生成都需要单独考虑各连通分支，比较麻烦，且没什么必要。实现方法为利用floyd求得的各点最短路径，从节点0出发的最短路径开始，对于最短路径小于∞的终点，都可以和起点归为一类；如此直到分类所有点即可。

3. 提高算法：GN算法

该算法用于社区发现。在本模型中，相当于对文章内容进行归类，分成不同的“社区”。下面对GN算法做出一个较为详细的说明。

定义社区：

设图$G = G(V, E)$, 所谓社区发现是指在图G中确定 n(>=1)个社区 $C = C_1, C_2, \ldots, C_n$,使得各社区的顶点集合构成V的一个覆盖。

定义边介数中心数：

某一条边出现在其他节点对见的最短路径上的次数。

基本思想：

算法步骤为：1）计算网络中所有边的边介数中心数 2）找到介数最高的边并从网络中移除 3）重复直至边为空

问题是：如何定义一个好的社区划分？一个好的社区划分应该为多少个社区？因此，引入了模块度的概念。

定义模块度（Q值）：

$Q = \frac{1}{2m}\Sigma_{i \neq j}(A_{ij} - \frac{k_i k_j}{2m})\delta(C_i, C_j)$ 其中，A_{ij} 是整个网络对应的邻接矩阵的任意元素（这里只考虑无向无权图的情况）。若i与j有连接的话，则 A_{ij}，否则$A_{ij} = 0$。$\frac{1}{2m}\Sigma_{i \neq j}A_{ij}$是图的总边数。任意节点的度数为$k_i = \Sigma j A_{ij}$。节点所在的社区为$C_i \in 1, 2, \ldots q$。当$u = v$时，$\delta(u, v)$，否则为0。当所有节点都归属同一个社区的时候，$Q = 0$。

Q的值越大，说明社区的结构越清晰。每当删除一条边后，模块度比原先最优模块度大，就令当前划分为最优划分，当前模块度为最优模块度。如此循环往复，直到边为空。此时，最优化分即为最终结果。

4. 图论分析

1. 对于一堆论文集，已知它们各自的关键字集合（每篇论文都有多篇关键字），我们如何可以给这些论文分到指定数量的类中呢?

可利用GN算法，进行社区划分。一个社区里的就是一类划分在一起的论文集。GN算法本身是依靠模块度达到最优化分的，如果有指定数量的社区限制，那么可以在运行过程中中途"切断"，即到达一定数量的社区集合即停止，令目前最优模块度的划分为最优解。

2. 对于一堆论文集，我已经了解了一片论文的内容，我希望了解另外一篇的内容，但是两篇内容关键字看似毫不重合，通过阅读其他论文的方式，我如何能最快地了解到那篇论文的内容呢?

最短路径的模拟。对于两篇论文之间的最短路径上节点的论文，我依次进行阅读，即可从目前已知 领域"有层次"地进入我目前不了解的领域。（这个看起来不是那么真实x）

3. 对于一堆论文集，我想大概了解所有论文的内容，从任意一篇论文开始，我该如何实现"轻松""有层次"的阅读?

最小生成树的模拟。

5. 可视化实现

基础算法部分和电影部分一致。选用了155个节点，构成了一个联通图，进行分析。将节点一系列信息，通过算法部分输出到.js文件中，作为可视化资源。

下面只说明GN算法的可视化实现。

该算法部分是采用python编写的，利用了networkx库。因为该算法的实现有比较多繁复的步骤，比如计算边的介数中心度，删除图中边，计算各点度数等等。这些步骤在每次迭代中都要进行，比较麻烦。而networkx库中封装好了一系列可供复杂网络分析的函数，可调用，比较方便。编写过程中，就按照前面说明的算法步骤进行迭代，直到边为空，得到最优社区划分。

同时，库中还有直接绘图的函数，可在每次生成一个"最优解"的时候都进行一次图的输出，就可以很明显的看到社区划分的每个阶段。如下图。同时，也利用网页做了d3版的可视化，如第二张图。对同一社区的节点，采用相同颜色标记。效果比较明显。


BestG


BestG0


BestG1


BestG2


BestG3


BestG4


BestG5


BestG6

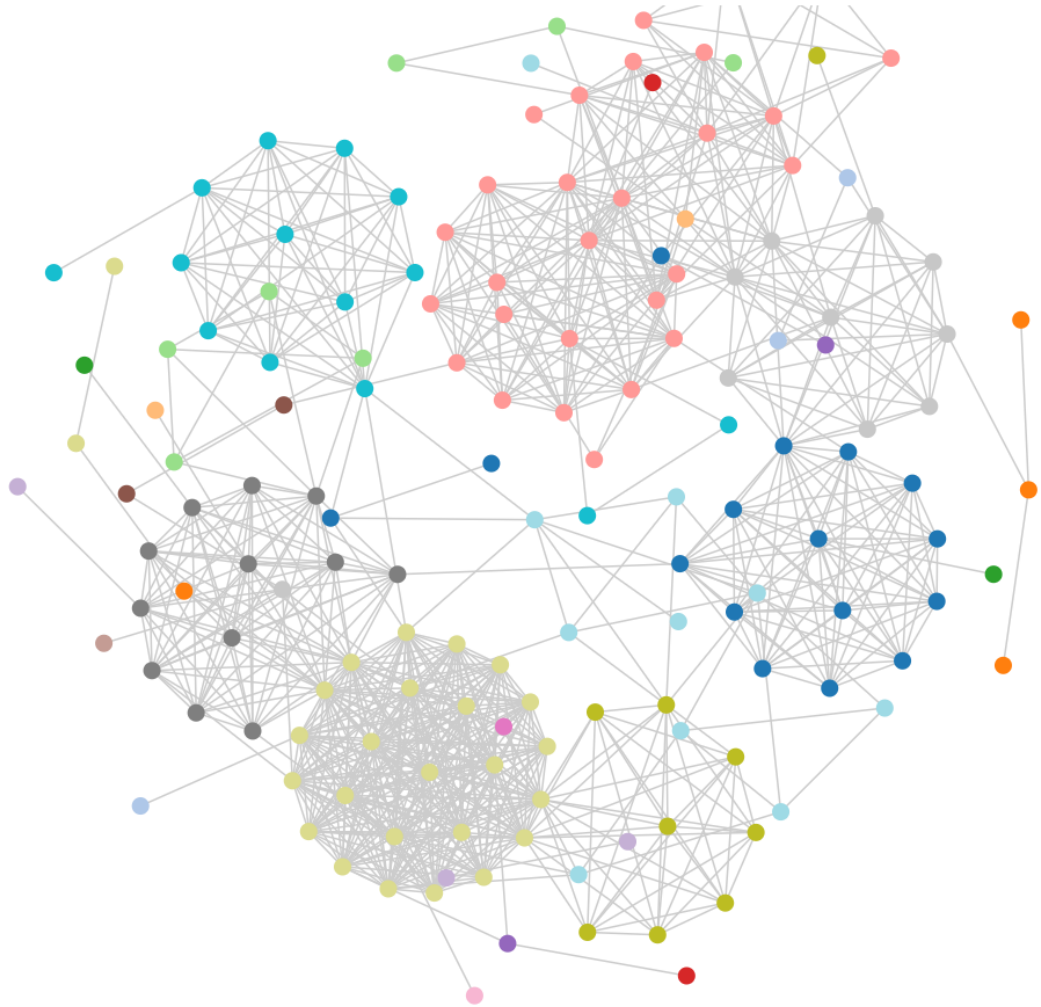

BestG7


BestG8


BestG9


BestG10


BestG11



6. 问题与改进

代码源文件不够整齐。一开始写的时候，还会比较注意类的定义，各函数接口等等，写到后面的时候，却因为急躁急于完成，就会杂乱地在代码中添加函数，因此导致可复用能力比较差。而且，在输出节点信息供可视化的过程中，由于各数据要统一输出（包括python实现的GN算法中的结果），在源文件中写了比较多仅仅供本次作业处理用的代码，并没有可扩展性。下次如果再进行类似的大作业，会考虑在前面先做好规划，不要在最后的时候狂写代码，而忽略了结构。

7. 实验感想

感谢和唐建宇的合作！实名吹！可视化结果很满意！

四、分工信息

- 唐建宇：电影数据建模、代码、分析、gui
- 汪颖祺：论文数据建模、代码、分析、gui
- 共用了读取csv文件、建图等部分代码和简单算法以及可视化部分