图分析大作业报告书

软件61 崔殷庇 2015080118, 软件61 李相赫 2016080042

1.编程环境：

操所系统：Windows 10

编程语言：JavaScript，C++

2.核心算法：

最短路程：

使用了Dijkstra算法。

最小生成树：

使用了prim算法。

中心度：

我们在求最短路程的过程中记录下了所有节点之间的最短路程，然后直接在计算closeness，betweenness中心都的时候利用了之前计算的两点之间的最短路程的数据，可以说也是利用了Dijkstra算法。

连通分量：

使用DFS，求一个连通支的所有结点。

如果存在结点不在此连通支内，新建一个连通支再用DFS计算连通支。

直到所有结点在连通支内，反复操作。

3. 数据提取及网络构建：

数据提取：

此次大作业的数据利用了助教提供的user.csv，对此数据进行了一点优化。

我们把一个电影当作一个节点，如果一个人看过不同的几个电影，那么这几个电影之间就产生了一条边，权是该节点的重复出现次数。举例子，如果一个人看过电影A，B，C，那么电影A与电影B，电影C之间就产生一条边。为了进行优化，我们把user.csv中的控制用户看过的电影数为2-6。最终剩下一千多个节点和两万个边，在可视化过程中数据太多太卡了，于是利用了其中的一部分数据。

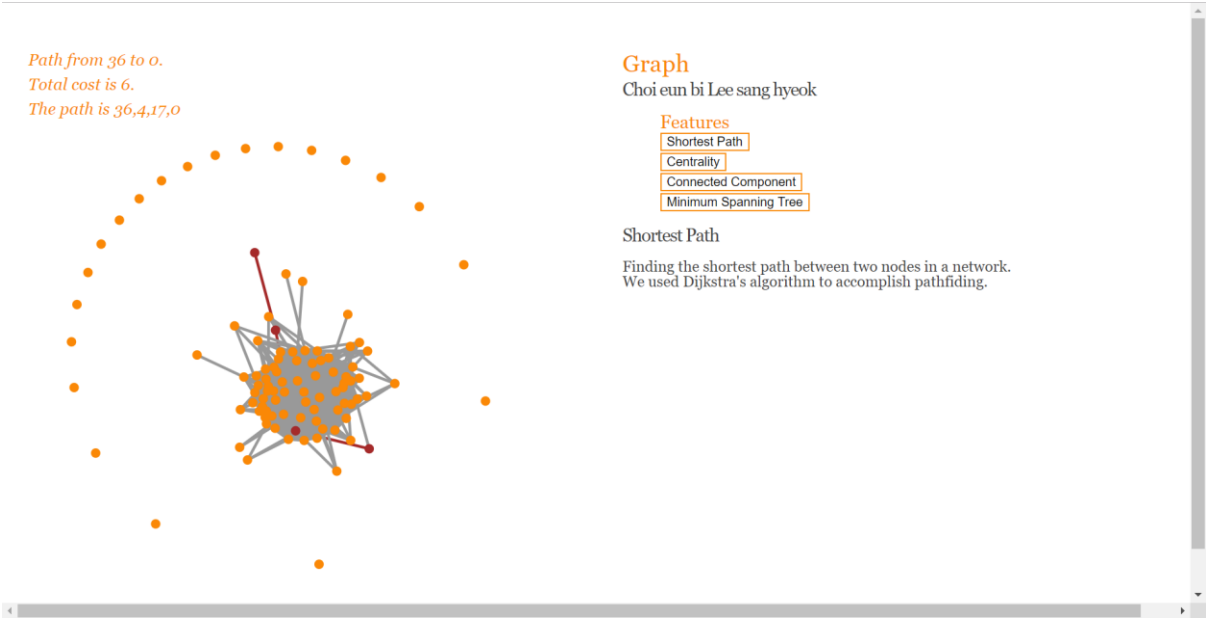
网络构建：

使用了json格式，里面有links和nodes，分别表示节点之间的边和节点的编号，links中有source，target和weight,cost，分别表示两个边的端点,weight和cost指该边的权值。 LinksAll.json和movie_id.json是数据优化之后的网络构建，但可视化过程中使用的部分数据在data.json里，而且movie_id.json里写了编号对应的电影名。

4. GUI使用说明：

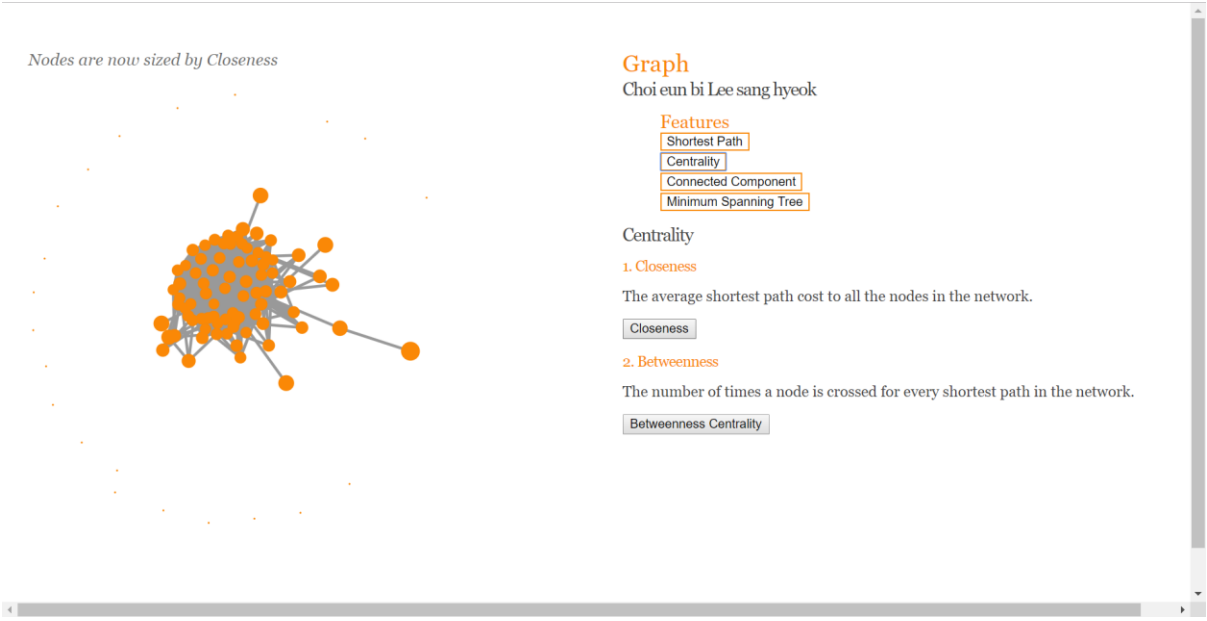
界面中有四个按钮，

1. shortest path是最短路程，使用方法为，点击shortest path按钮之后，再点击左图中的两个节点，那么根据搜索结果，图中的边和节点和边会变色。

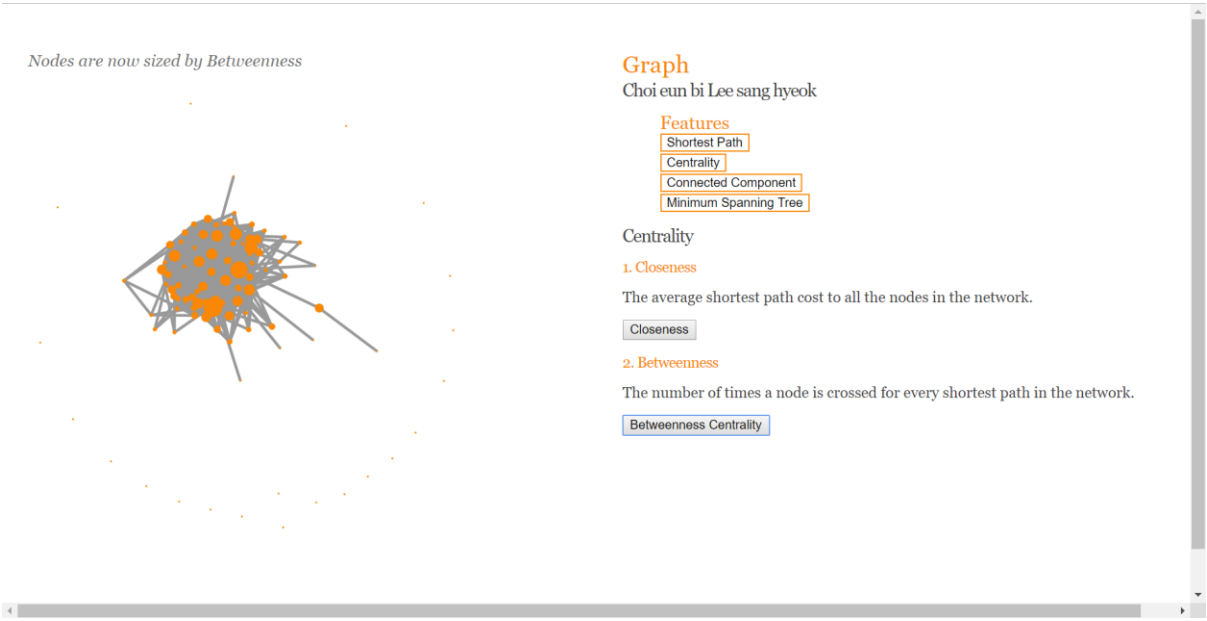


2. Centrality是中心度，点击该按钮之后下面会出现closeness和betweenness按钮，

点击即可。左图中的节点大小会根据要求变化。（计算需要一点时间）。

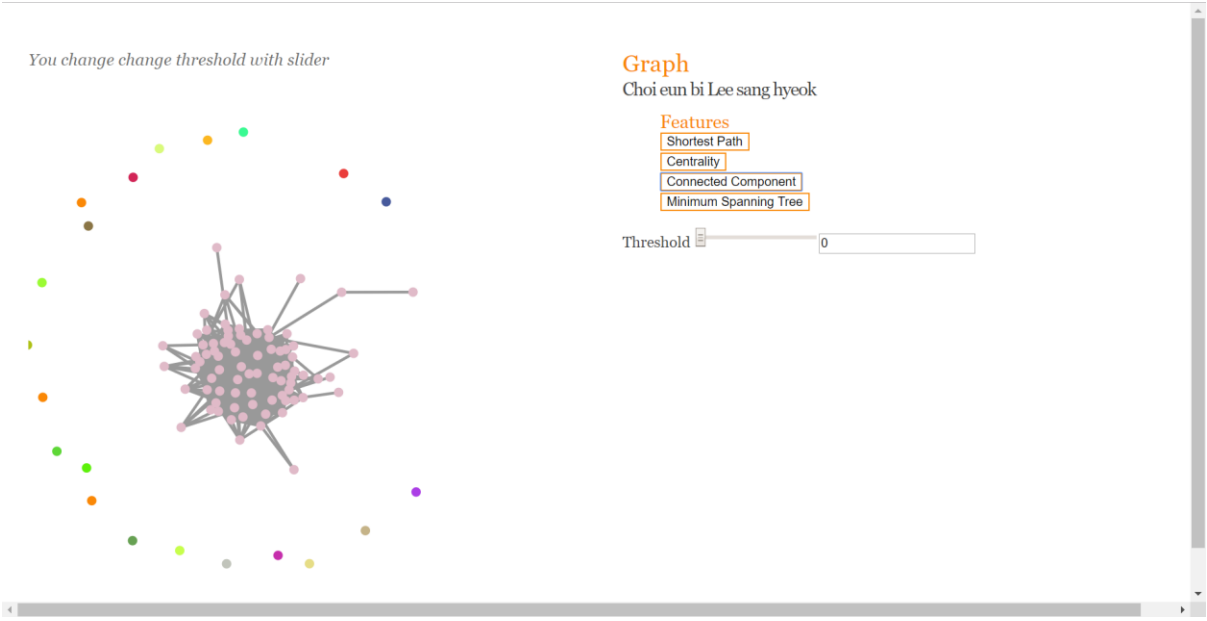


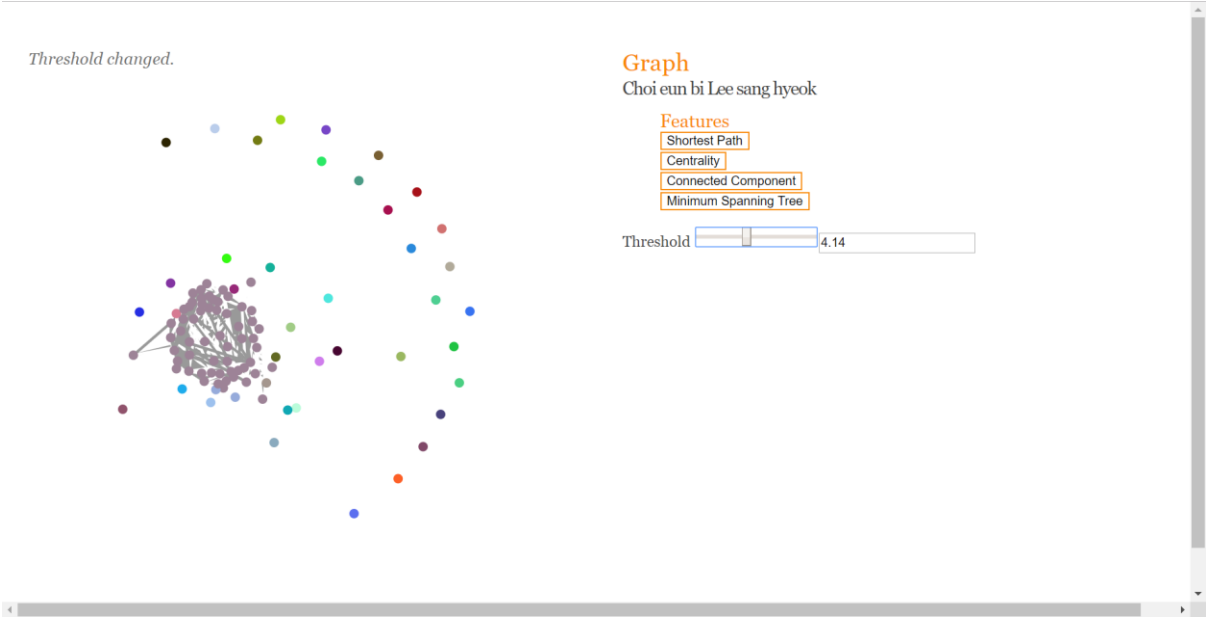
closeness中若结点大，表示此结点到各个节点的平均距离大。



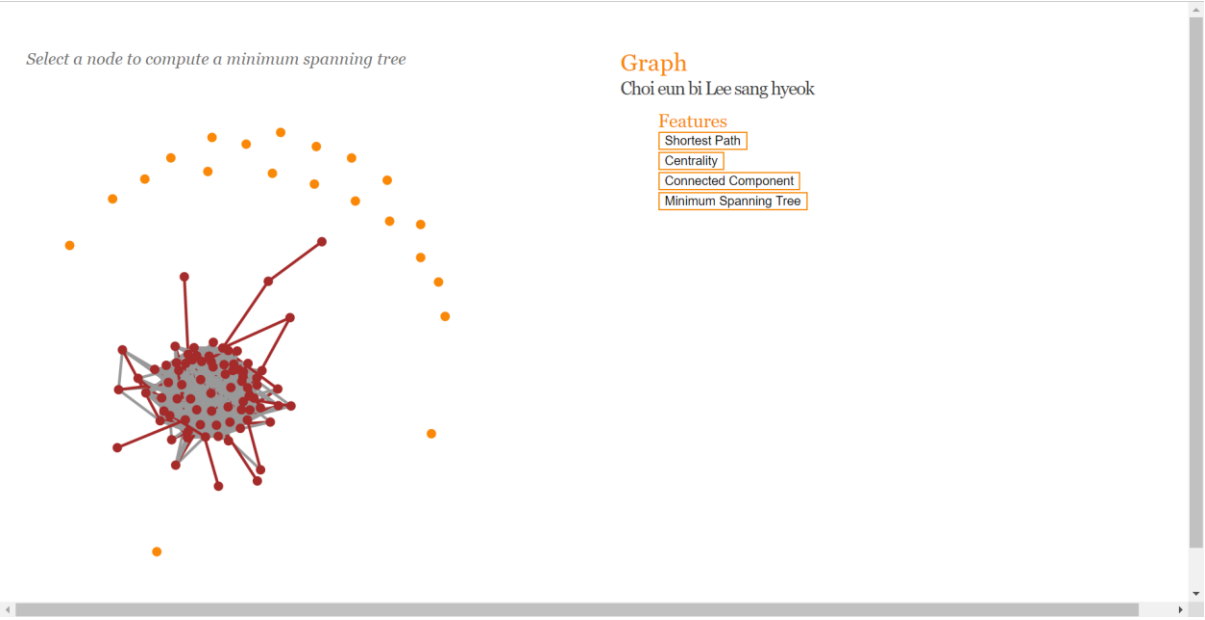
betweenness中若结点大，表示经过此结点的最短路径数多。

3.Connected Component是连通分量，点击之后可以调节下面的滚动条该边阈值。





4.Minimum Spanning Tree是最小生成树，点击按钮之后，再点击左图中的节点，最小生成树的边会变色（计算需要一点时间）。



以上是利用我们提取的数据进行一系列操作得到的图片。但是因为点和点之间的关系太紧密，不美观，不容易直接看出结果，不能体现我们的可视化技术，于是提交的代码里用了随机生成的点200个。

5.运行方法：

请访问网址

网址：www.kidmobile.com