离散大作业说明文档

**队员**

骆炳君 2017013573（主要负责Python后端及算法）

王阳 2017013580（主要负责D3.js前端的可视化）

GitHub地址

<https://github.com/LuoBingjun/Concrete-Math-Project>

**成果说明**

1. 实现了界面简洁美观的可视化交互的网页

2、实现了多种不同的力导向图；测试通过网站交互进行

3、实现了最短路、最小生成树、中心度计算等基础算法；测试可通过可视化实例来进行

4、通过适当建模，通过图论算法实现了相似推荐功能，实现了算法与应用结合的拓展算法；测试通过手动进行

5、通过论文阅读，提出了一种可行的电影评分的算法；理论可行，暂未验证

6、通过论文阅读，学习图论算法在基础网络协议中的实际应用——路由扩展算法：分布式的迪杰斯特拉算法

**测试方法**

**1、依赖环境配置：**

由于网页调用了在线D3.js库，故确保在连接网络的情况下使用，同时由于算法的实现依赖Python脚本，故需要Python库eel，具体安装命令如下：

python –m pip install eel

**2、测试方法：**

1.运行web目录下的setup.py脚本搭建本地服务器

2.通过VS2017打开index.html后，右击鼠标选择在浏览器中打开，通过这种方式进入调试模式，之后便可查看相应的网站页面及可视化效果

**数据应用分析**

1. **建模方式：**

本次作业用到了有关豆瓣电影的两个数据集，共采用了两种建模方式：

1. 电影为节点,边权确定方式如下：1.若电影类型不同或评价差距太大则无边，否则通过类型及评分相似度赋予初权值；2.若其他因素(主演、导演等)相同，则适当增加原权值
2. 以人为节点、相似建边确定方式如下：1.若电影类型不同或评价差距太大则无边，否则通过类型及评分相似度赋予初权值；2.若其他因素(主演、导演等)相同，则适当增加原权值；3.对于已有边，若两用户对另外的电影有相似评价，则适当增加原权值，反之适当减少权值

由相似度建边时，我们通过算法控制边权为0到1之间的实数，用以衡量关联程度，同时路径的计算方式转化为边权相乘，较好地模拟了关联程度的递减过程。

对于权值的初始化处理，使用了多线程加速的方式。

1. **最短路算法的实际意义：**
2. 注意到原始的最短路径在这种建图方式下不收敛且无意义，故转求最长路径，路径权值计算方式由建模方式中给出
3. 实际意义：所得值的实际意义代表着两节点之间的最大相似程度，可以用来较精确的衡量节点之间的相似性
4. 算法实现：采用迪杰斯特拉算法
5. 算法评估：在本问题中，权值的更新将采用权值相乘的方式，且边权都是位于0~1之间的小数。在这种意义下，对于**十分稠密**的图，算法可能退化成仅有一层的广度优先搜索
6. **最小生成树算法的实际意义：**
7. 注意到原始的最小生成树在这种建图方式下无意义，故转求最大生成树
8. 实际意义：所得的树实际意义在于可以用来近似求节点间相似程度，经检验效果较好
9. **中心度意义：**
10. 通过节点在图上的度，来反应一个节点的中心度，可以用于衡量节点的普遍程度以及受欢迎程度
11. **推荐算法的产生：**
12. 通过适当的建模方式，人与人、电影与电影之间产生了相似性，通过这种相似性，可以实现较好的相似推荐算法
13. 算法实现：广搜+排序
14. 具体测试：经过一些实例测试以及人工评估，算法产生的结果较为可信合适

**深度扩展**

1. **背景：**

一般的评分系统的实现是通过简单的求大量用户的平均值进行的，无法排除或削弱极端数据或故意伪造数据的影响，为此考虑实现一种新的评分系统机制

1. **相关方法：**

梯度下降法

1. **实现思路(模型)：**

初始化：

设置每个人的初始权值为1.0，计算其加权平均值作为初始总体评价

循环：

根据个人评价与总体评价之间的差值，对每个人的评价权值作出相应修改(梯度下降)

更新：

重新计算总体评价(加权平均值)

判断：

直至模型收敛，即两次更新间电影的评分差足够小

结束：

将收敛结果作为最终评分

1. **算法分析：**

由于缺少TensorFlow框架及其他知识，并未付诸实践，但具有一定可行性。

通过加权权值的模型训练，使得评分系统能够尽量削弱不合理、不专业评价对真实评价的影响，从而取代传统的直接取均值或删极端数据算法。

与此同时，由于可以获得训练的结果，可以对每个用户的评分方式进行相当不错的评估，以用于后续更进一步的用户个性化推荐以及相似推荐。

由于算法要求本身初始的平均值能够较真实客观地反映实际，因此需要本身有一定的数据基数。

**参考文献：**

[1] 吴军. 数学之美[J]. 2012.

[2] 张纪元. 梯度下降法[C]// 中国数学会全国最优化数值方法学术会. 1987.

[3] 张翰林, 关爱薇, 傅珂,等. Dijkstra最短路径算法的堆优化实验研究[J]. 软件, 2017, 38(5):15-21.

[4] 高雨明. 浅析RIP和OSPF协议的工作原理[J]. 中国新通信, 2010(17):35-39.

[5] CSDN系列文档以及d3.js说明文档、layui说明文档