团体紧密程度算法说明文档

互联网大数据分析中心实习生　　周建宇　　2017/8/14

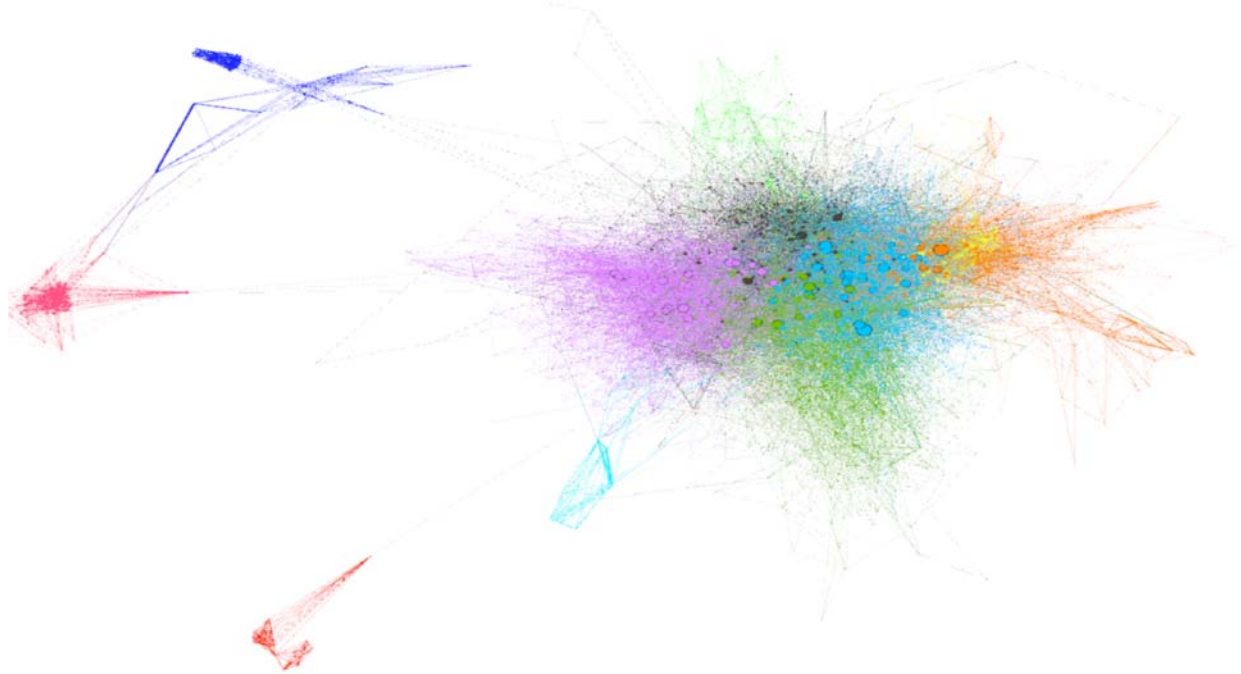
需求说明

给定一个简单图G，我们赋予G中每个节点一个类别，我们需要根据G中的节点的度、边权等信息计算G中任意两个类别（或团体）的紧密程度。其中紧密程度可以理解为一种描述两个类之间关联大小的一种量化表示。本次研究的目标如下所示：

- 给出团体紧密程度的形式化定义
- 实现团体紧密度算法

数据描述

算法的输入是一张简单图G，G以xml标准进行表达，以gexf格式存储。G中每个node具有自己的id，node所属的类别用modularity_class属性标明。同样地，每条边均有一个id，并人为规定了每条边的源source，和汇target，以及edge的权重weight。数据描述与可视化情况如下图所示：



算法描述

1. 对于G中每一个节点， 计算每个节点对自身类的重要程度和对其他类别的重要程度。对于第i个节点, 其与各类节点的重要程度计算公式如下：

$$a_{ik} = \frac{\sum_{m=0}^{M_i} w_m}{\sum_{n=1}^{N_i} w_n}$$

其中a_{ik}代表第i个节点与第k类的关联程度，M_i、N_i分表表示第i个节点的邻居节点中属于第k类的个数与第i个节点的邻居节点总数。

2. 计算第m个类与第n个类的未归一化关联度：

$$S_{mn} = \sum_{n=1}^{k} a_{im} a_{in} a_{im} a_{jn} w_{ij} A_{ij}$$

$$A_{ij} = \begin{cases} 1, \text{if } i \in m \text{ and } j \in n \\ 1, \text{if } i \in n \text{ and } j \in m \\ 0, \text{other} \end{cases}$$

其中w_{ij}代表第i节点与第j个节点形成的边e_{ij}的权重。

3. 关联度归一化：

$$S'_{mn} = softmax(S_{mn})$$

注：softmax是一种常见的多分类函数，常用来做归一化。

复杂度分析

设图G中有N个节点，M条边。我们从两个角度分析复杂度，节点数量N和边数M。

- 在仅考虑N的情况下：
  - 在未建立索引的情况下
    - 计算节点重要程度的复杂度上限$\Theta_1(N) = (N-1)N^2$
    - 关联度加总的复杂度上限$\Theta_2(N) = (N-1)N + \frac{(N-1)N}{2}$
    - 总复杂度$\Theta(N) = N(N-1)(N^2 - N + 1.5) \approx O(N^3)$
  - 若考虑建立索引
    - 建立索引所需的时间为$\Theta_1(N) \approx O(N)$(同时给节点和边建索引)
    - 计算节点重要程度$\Theta_2(N) \approx O(N)$
    - 关联度加总$\Theta_3(N) \approx O(N^2)$
    - 总复杂度$\Theta(N) \approx O(N^2)$
      -
- 在M和N均考虑的情况下：
  - 为简单起见我们仅考虑建立索引的情况，同理通过上述分析可得出$\Theta M, N \approx O(N+M)$
- 实际运行过程中，我们对有2708个节点，28079条边，总共十二个类别的图两两计算紧密程度（总共$C_{12}^2 = 64$次计算），总共5.37秒，能够满足实际业务对效率的要求。

脚本使用说明

- 程序默认gexf文件均与脚本在同一目录下，若要更改数据路径，则在脚本中dataPaths即可
- 原始gexf文件为发行版本，为使networkx能够读取，需要手动将gexf文件改为草稿版，具体更改方法如下图所示：

```
<?xml version="1.0" encoding="UTF-8"?>
<gexf xmlns="http://www.gexf.net/1.3" version="1.3" xm
www.gexf.net/1.3/gexf.xsd">
```

```
<?xml version="1.0" encoding="UTF-8"?>
<gexf xmlns="http://www.gexf.net/1.2draft" version="1.3"
www.gexf.net/1.3/gexf.xsd">
```

- 程序执行完毕后会生成相应的同名csv文件

实现环境

- 实现语言：python3.5
- 第三方包：pandas, networkx1.1
- 操作系统：windows 10, 64位
- 内存：16.0GB
- 处理器：Intel(R) Core(TM) i7-7700K CPU@4.20GHz 4.20GHz

问题总结

- 在实际计算中，可能会存在输入图不是简单图的问题，对此我们首先要去除自环和重边。另外可能有部分边没有权重，此时可以将该边忽略或统一设置为有权重边的平均值，从而减少数据质量不高对计算结果的影响。

- 在计算节点重要程度的过程中，我们选择了比较简单的方法，其他方法例如考虑每个节点邻居节点连接关系等方法也可以尝试使用，但是会是计算开销增加，个人认为在当前的数据规模和团体连接情况下若不需要十分精确的紧密度计算无需采用跟复杂的方法。