neo4j算法包调研

neo4j算法文档

- Section 5.3, “Community detection algorithms”社区检测算法
- Section 5.4, “Similarity algorithms”相似度算法

社区检测算法

- Louvain分层聚类算法，该算法将社区递归合并到单个节点中，并在压缩图上执行模块化聚类
- Label Propagation标签传播
- Weakly Connected Components弱连接组件
- ~~Triangle Count三角数~~
- ~~Local Clustering Coefficient局部聚类系数：返回一个节点相邻的节点之间联系的紧密程度~~
- ~~K-1 Coloring着色问题：相邻节点颜色不同，用尽量少的颜色~~
- Modularity Optimization模块化优化：度量模块或社区内连接的密度。社区内具有许多联系，社区间联系较少
- ~~Strongly Connected Components强连接组件：两个节点之间有互通的路径称为强连接~~

相似性算法

- Node Similarity节点相似度：共享邻居

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

- ~~Jaccard Similarity同节点相似度~~
- ~~Cosine Similarity余弦相似性，当不考虑权重的情况，退化为jaccard~~

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

- ~~Pearson Similarity可解释性稍差,运行慢~~

$$similarity(A,B) = \frac{\mathrm{cov}(A,B)}{\sigma_A \sigma_B} = \frac{\sum_{i=1}^{n}(A_i - \overline{A})(B_i - \overline{B})}{\sqrt{\sum_{i=1}^{n}(A_i - \overline{A})^2 (B_i - \overline{B})^2}}$$

- ~~Euclidean Similarity欧式距离，不合适~~
- Overlap Similarity重叠相似性

$$O(A,B) = \frac{|A \cap B|}{\min(|A|, |B|)}$$

- ~~Approximate Nearest Neighbors相似性根据雅克卡德相似性、余弦、欧式距离，皮尔逊相似性~~

结论

1. 4种社区检测算法，在2w个节点的图谱上聚类后，每个社区的实体融合结果一致

	nodeId	person	communityId
16881	7174	刘兰芳	20400
16883	10707	刘兰芳	20400
14283	4595	刘兵	19549
14279	1559	刘兵	19549
1933	13558	刘志刚	15432
...	...	...	...
21021	11201	陈浩	21678
20289	13036	陈磊	21457
20288	1026	陈磊	21457
6587	4411	黄志勇	16935
6584	805	黄志勇	16935

176 rows × 3 columns

2. 2种相似性算法，结果也一致，但overlap更符合条件，且不会出现重复的情况（jaccard一对节点会重复）

	fromID	from	toID	to	count1	count2	intersection	similarity
9850	7174	刘兰芳	10707	刘兰芳	2	2	2	1.0
2384	1559	刘兵	4595	刘兵	2	2	2	1.0
8335	5956	刘志刚	13558	刘志刚	3	3	3	1.0
9081	6540	刘振华	14288	刘振华	2	2	2	1.0
9823	7153	刘斌	13612	刘斌	5	5	5	1.0
...	...	...	...	...	...	...	...	...
10139	7418	陈彬	14148	陈彬	2	2	2	1.0
7252	5084	陈林	9054	陈林	2	2	2	1.0
9111	6559	陈浩	11201	陈浩	2	2	2	1.0
1555	1026	陈磊	13036	陈磊	2	2	2	1.0
1230	805	黄志勇	4411	黄志勇	2	2	2	1.0

91 rows × 8 columns

3. 相似性算法和社区检测算法结果一致

In [234]:
```
1 overlap[overlap['from']=='李勇']
```

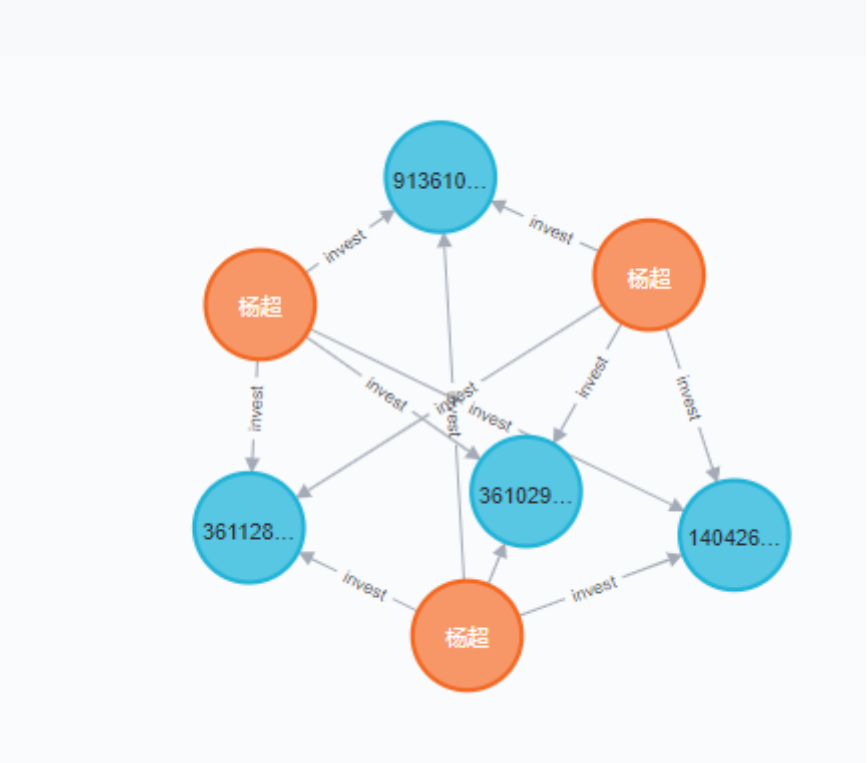
Out[234]:

	fromID	from	toID	to	count1	count2	intersection	similarity
1565	1032	李勇	10672	李勇	5	5	5	1.0
1566	1032	李勇	12815	李勇	5	5	5	1.0
13907	10672	李勇	12815	李勇	5	5	5	1.0

In [235]:
```
1 overlap[overlap['from']=='杨超']
```

Out[235]:

	fromID	from	toID	to	count1	count2	intersection	similarity
3748	2479	杨超	8675	杨超	4	4	4	1.0
3749	2479	杨超	10579	杨超	4	4	4	1.0
11567	8675	杨超	10579	杨超	4	4	4	1.0



六种图算法.ipynb
38.27KB