社区发现算法实现与比较

【实验内容】

- 实现 spectral clustering 等几个社区发现算法（共 5 种）
- 比较实验结果
- 评估算法
- 社区结果可视化

【实验环境】

编程语言：Matlab

编程环境、运行环境、使用工具：Windows 10 Matlab R2015b dephi

【实验步骤及方法】

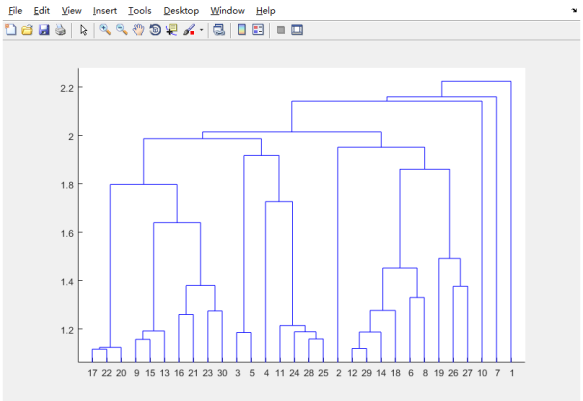
1、 各个函数实现：（基本参考老师的 pdf 再调用函数接口即可实现）

- Jaccard:

$$Jaccard(\mathbf{v}_i, \mathbf{v}_j) = \frac{|N_i \cap N_j|}{|N_i \cup N_j|}$$

```
pdist2(A,A,'jaccard'); % 生成 Jaccard 距离
linkage(Y,'average');  % 按平均距离 linkage
cluster(Z,k); % 聚类计算
dendrogram(Z);        % 生成图像
```

最后一段代码可以生成图像，示例效果如下：（football.mat）



- Radiio Cut:

$$RatioCut(A,\overline{A}) = cut(A,\overline{A})(\frac{1}{|A|}+\frac{1}{|\overline{A}|})$$

这是标准的迹最小化问题，其解为 L 的的前 k 个特征向量所构成的矩阵。最后采用 k-means 方法对该矩阵的行进行聚类，就可以实现对该数据集的 k 聚类。

```
diag(sum(A,2));                    % 求得 D
[X,~]=eig(L);
kmeans(X(:,1:k),k);
```

- Normalized Cut:

$$Ncut(A,B) = cut(A,B)(\frac{1}{vol(A)}+\frac{1}{vol(B)})$$

该问题的解为广义特征值问题 Lh=λDh 的前 k 个特征向量所构成的矩阵。最后采用 k-means 方法对该矩阵的行进行聚类，就可以实现对该数据集的 k 聚类。

```
D=diag(sum(A,2));
L=D-A;
LL=D^(-1/2)*L*D^(-1/2);
[X,~]=eig(LL);
kmeans(X(:,1:k),k);
```

- Modularity:

$$B = A - \mathbf{d}\mathbf{d}^{\top}/2m$$

```
[row,~]=size(A);
d=sum(A,2);
m=nnz(A);   %因为是无向图，所以 m=edge*2
B=A-d*d'/m;
[X,~]=eig(B);
kmeans(X(:,row-k+1:row),k);
```

- Girvan-Newman:

这个算法跑得非常非常地慢……

关键的代码是用 BFS 算出 betweenness，然后移除 betweenness 最高的 edge，直到生成 k 个联通片为止。

代码中用 betweenness()调用 BFS 求得结果。

具体参考了 Matlab BGL 库函数写法和 Girvan-Newman 的 C 语言书写方法。

```
function clustering=girvannewman(A,k)
function [X, path, Y, index] = bfs(A,u, adj_index, n_size)
function calcutBT = betweenness(A)
```

2、 Main.m

简单粗暴地实现了各个聚类算法的运行和 Gephi 输入文件的构建。

3、 Gephi 可视化

就是把构建好的输入文件输入，即可达到可视化的结果。

各个可视化图在文件中 image 文件夹下。

4、 比较 NMI 和 ACC

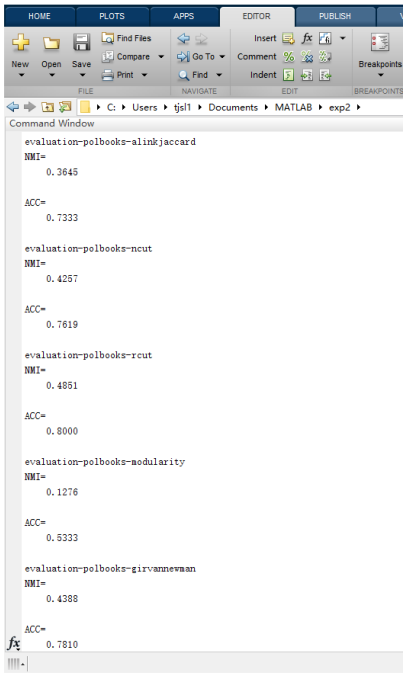
5、 无监督学习评价指标

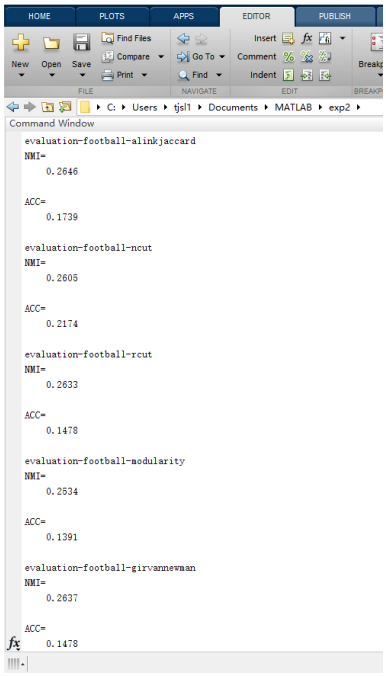
6、 附加讨论

【实验结果说明及演示】

1、 代码运行结果，可见 txt 文件

2、 NMI 和 ACC





	Polbooks		Football	
	NMI	ACC	NMI	ACC
Alink-jaccard	0.3645	0.7333	0.2646	0.1739
Rcut	0.4851	0.8000	0.2633	0.1478
Ncut	0.4267	0.7619	0.2605	0.2174
Modularity	0.1276	0.5333	0.2534	0.1391
Girvan-Newman	0.4388	0.7810	0.2637	0.1478

NMI 评价标准是基于信息熵的评价方法，Accuracy 里可以包含了 precision, recall, f-measure. 从结果来看：

横向比较的话，Polbooks 的 NMI 和 ACC 值较高一些，而 football 结果差很多。我认为是因为 Polbooks 的聚类特征明显一些，而 Football 的聚类特征不那么明显，因此 Football 的结果较为差一些；或者这几种聚类算法没有捕捉到 Football 的聚类特征，因此聚类的结果稍微差许多；或者是由于评价标准的偏向性导致没有专对应于聚类特征的评价；

纵向比较的话，Polbooks 的 NMI、ACC 的 Modularity 值尤其地低，这说明对

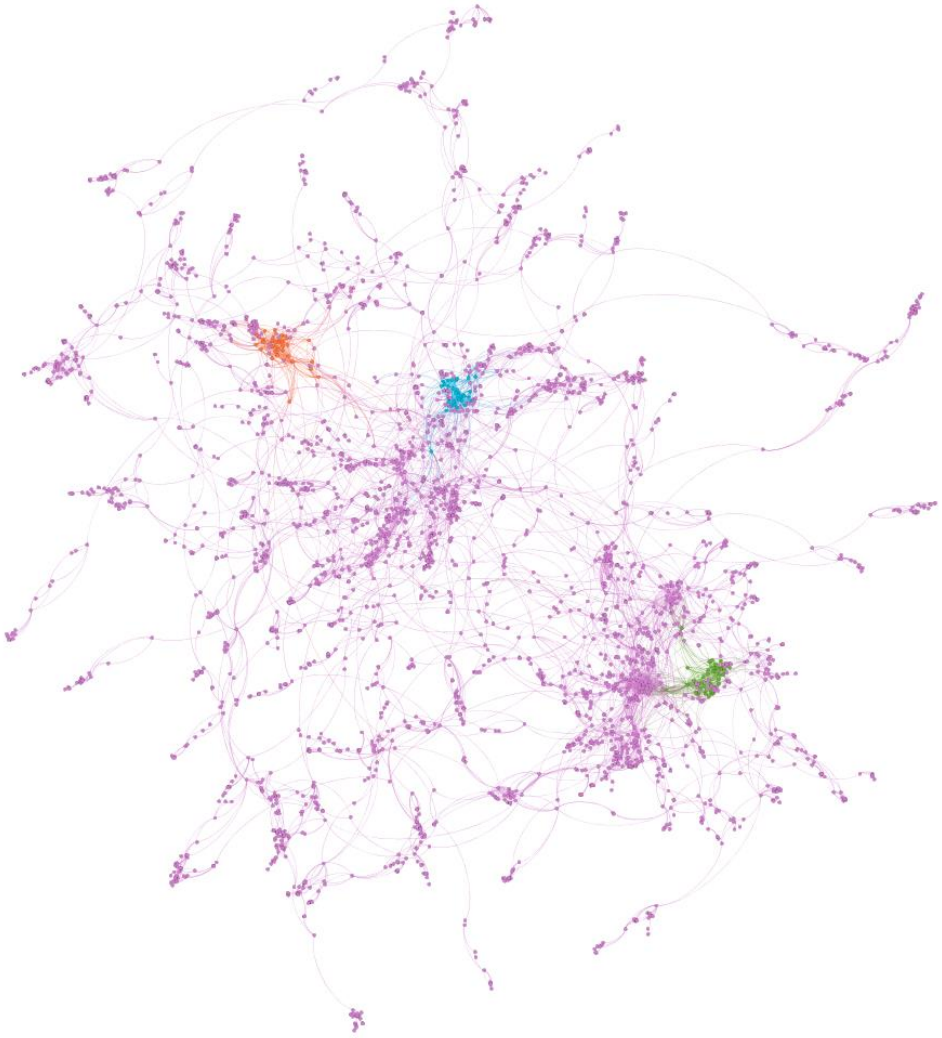
于这个数据集Modularity的聚类方法很不合适,以此思路可以将Modularity的聚类方法使用范围拓展到类似于Polbooks的样本中，都不适合。相比之下Rcut的两个评价标准都表现非常好，以此思路可以将Rcut评价方法适用于类似于Polbooks的样本中。而Football中两个评价标准表现出来的值都比较低，这个评价结果说明得对聚类算法的选择、评价标准的选择做进一步的思考。

3、 可视化

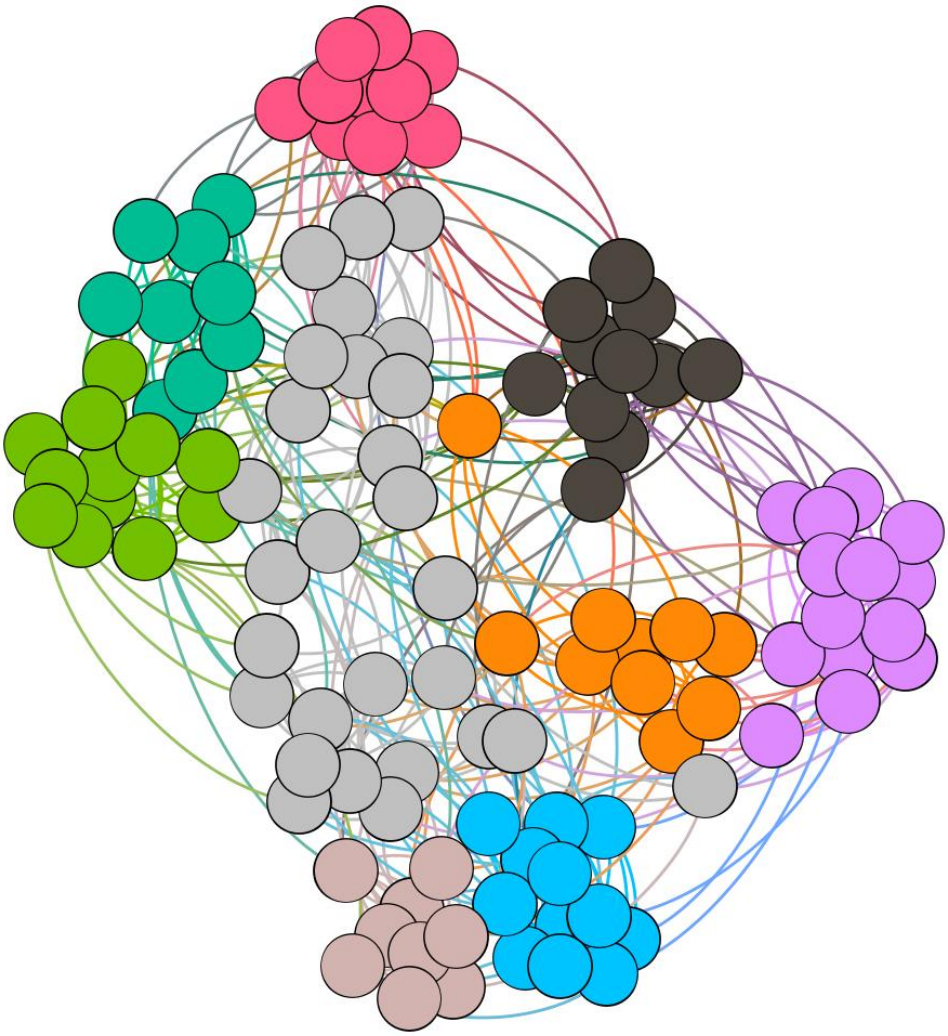
可视化的结果在文件夹image中。

以下拿取几个例子：

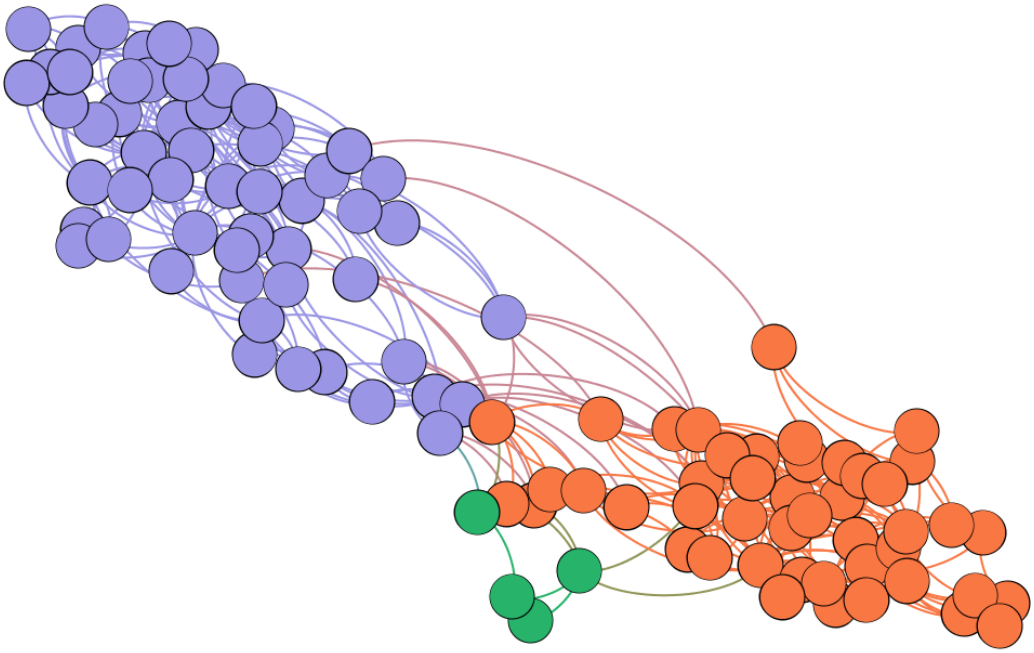
以下是一个DBLP的Molulary结果的可视化：



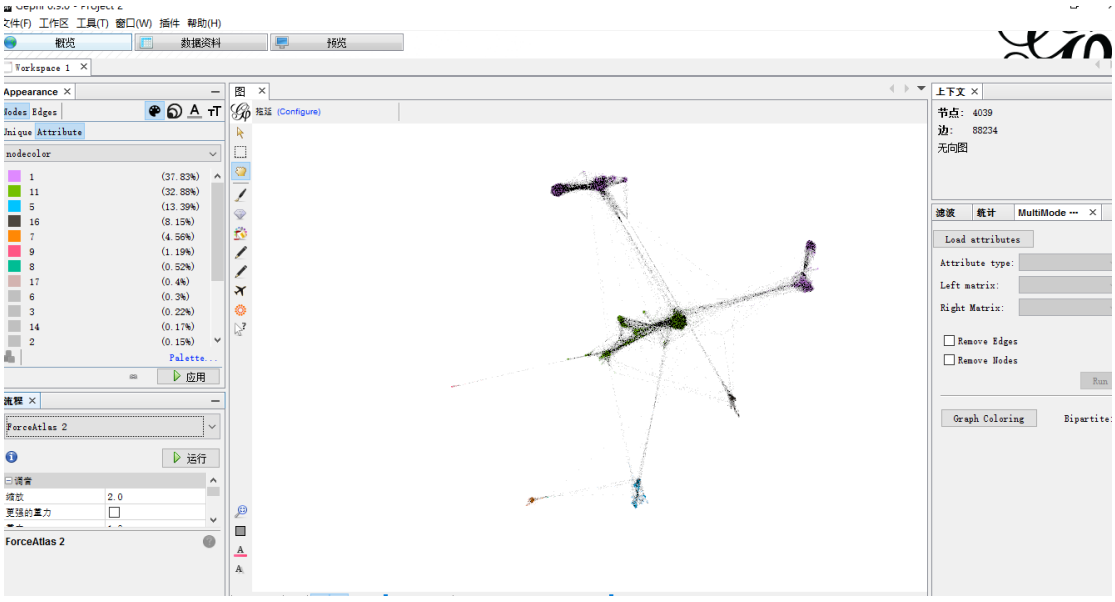
Football 的 rcut 可视化：

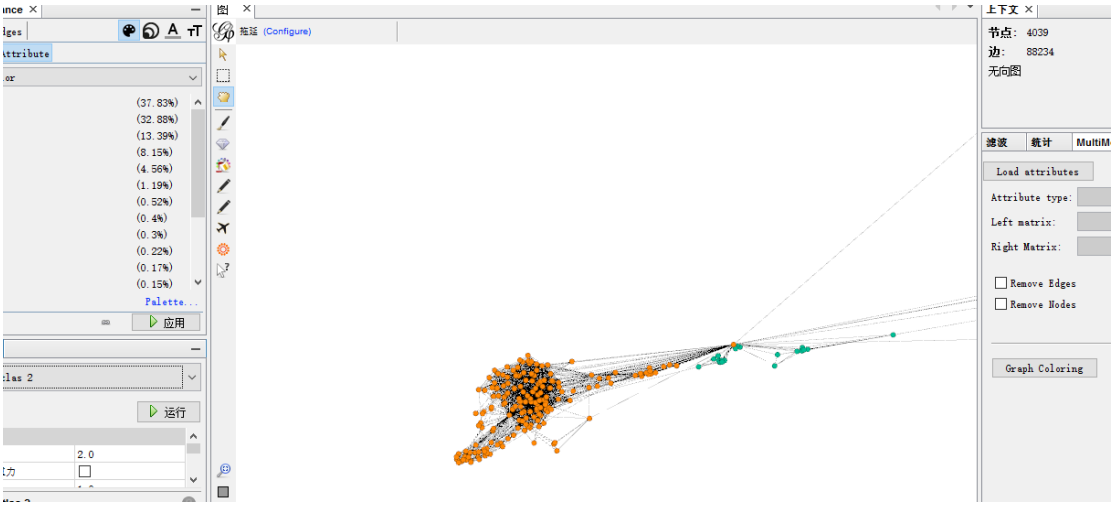


Polbooks 的 rcut 可视化：



Egonet 的 rcut 可视化：





4、 Egonet 的 k 选择

我调研到了几种方法：

A. K-means 聚类数确定 http://www.docin.com/p-1044725223.html

a) 经验法

b) 密度法

c) 逐个归类法

d) 爬山法

B. Modularity Maximization

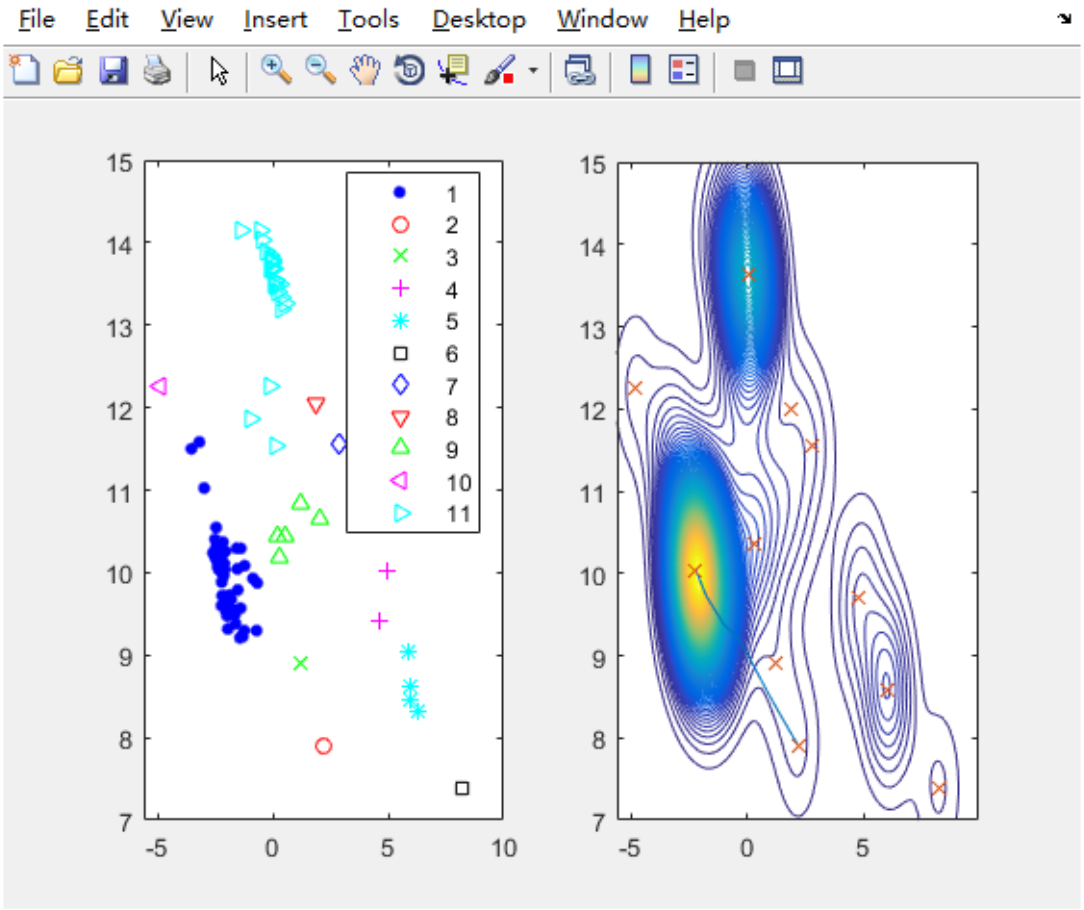
如果数据能转化成用网络(network)来表示,那么用 community detection 的技术便可以把所有节点(node)聚类。主要是因为它们采用了 greedy heuristic 的算法，其中很多方法都不需要提前指定 community 数目，而且也不需要程序自己指定 community 数目。

C. Canopy 算法

可以用于大至估算聚类数，但需事先给定阈值 T1,T2(两个半径)，因此实际是将聚类数量的确定转化为阈值的确定。

我的想法是采用一个不需要 k 值最后靠其它约束参数求出聚类的算法,算出实际分了几个聚类，再使用这个结果用于作为 K。

我还没有明白其中的数学原理,只大概懂了点意思。然后找了个实际的 paper 和算法 http://sites.stat.psu.edu/~jiali/hmac/ 来进行测试和模拟，下面是它的一个 demo 跑出来 11 个聚类：



我用 Egonet 的数据集，然后，有时候是很大的聚类个数，有上千个左右，有时候只有一个，我一直调整它的参数，调出个 18 个数据聚类的结果。然后我就用的 18

刚开始想的就是大部分都是 K-means 算法，找一个算法可以估计 K-means 的 k 值就好，但是看得很多后发现这个算法据说很厉害，然后就认真在调它的参数了，至于结果效果如何，不一定保证用这五个聚类算法就好，我也想强调，这五个聚类算法不一定适合这个数据集。

但是从测试的结果和可视化结果来看，有一些点集和其它点集关系密切但是被分成了两类，可见我觉得还是有点大的，不过我还是使用了 18，因为全跑一遍算法真的很耗时。。。以至于 GN 算法的数据集已经跑了 12h 了并没有跑出结果来：

这是结果为 8 个联通片时耗时 12h

5、 无监督学习评价 DBLP 和 Egonet

在 google scholar 中找到一篇 Understanding of Internal Clustering Validation Measures,但是没有校园网,下载不下来看,Experiment results show that S_Dbw is the only internal validation measure which performs well in all five aspects, while other measures have certain limitations in different application scenarios.

6、 附加讨论

从算法实现上看，这其中三个聚类算法都是依靠 K-means 来求得结果的，K-means 的缺点是显而易见的：

1、在 K-means 算法中，首先需要根据初始聚类中心来确定一个初始划分，然后对初始划分进行优化。这个初始聚类中心的选择对聚类结果有较大的影响，一旦初始值选择的不好，可能无法得到有效的聚类结果，这也成为 K-means 算法的一个主要问题。对于该问题的解决，许多算法采用遗传算法（GA），以内部聚类准则作为评价指标。Canopy 算法可以改进这点。

2、聚类结果是圆形状，对条状和线状支持不好

3、K-means 算法中 k 是事先给定的，这个 k 值的选定是非常难以估计的。很多时候，事先并不知道给定的数据集应该分成多少个类别才最合适。这也是 K-means 算法的一个不足。有的算法是通过类的自动合并和分裂，得到较为合理的类型数目 k，例如 ISODATA 算法。关于 K-means 算法中聚类数目 k 值的确定，有些文献中，是根据方差分析理论，应用混合 F 统计量来确定最佳分类数，并应用了模糊划分熵来验证最佳分类数的正确性，它使用了一种结合全协方差矩阵的 RPCL 算法，并逐步删除那些只包含少量训练数据的类，这是一种称为次胜者受罚的竞争学习规则，来自动决定类的适当数目。它的思想是：对每个输入而言，不仅竞争获胜单元的权值被修正以适应输入值，而且对次胜单元采用惩罚的方法使之远离输入值。

4、从 K-means 算法框架可以看出，该算法需要不断地进行样本分类调整，不断地计算调整后的新的聚类中心，因此当数据量非常大时，算法的时间开销是非常大的。所以需要对算法的时间复杂度进行分析、改进，提高算法应用范围，例如，可以从该算法的时间复杂度进行分析考虑，通过一定的相似性准则来去掉聚类中心的侯选集。在有些文献中，使用的 K-means 算法是对样本数据进行聚类，无论是初始点的选择还是一次迭代完成时对数据的调整，都是建立在随机选取的样本数据的基础之上，这样可以提高算法的收敛速度。

5、K-means 算法对异常数据很敏感。在计算质心的过程中，如果某个数据很异常，在计算均值的时候，会对结果影响非常大

【参考文档】

- Matlab 矩阵运算
  http://yajunok.blog.163.com/blog/static/656576200891794802 57/
- 谱聚类算法详解
  http://blog.csdn.net/jteng/article/details/49590069
- Matlab 中 pdist 函数详解(各种距离的生成)
  http://bulo.hujiang.com/u/6216467/diary/211759/
- 聚类有效性的组合评价方法
- Matlab BGL 库
- 怎样评价聚类结果好坏?
  http://www.cnblogs.com/lifegoesonitself/p/3318643.html
- Mutual information and Normalized Mutual information 互信息和标准化互信息
  http://www.cnblogs.com/ziqiao/archive/2011/12/13/2286273.html

- https://en.wikipedia.org/wiki/Mutual_information
- GAUSSIAN MIXTURE MODELS TUTORIAL https://chrisjmccormick.wordpress.com/2014/08/04/gaussian-mixture-models-tutorial-and-matlab-code/
- Clustering by fast search and find of density peak. Alex Rodriguez, Alessandro Laio
- http://sites.stat.psu.edu/~jiali/hmac/
- https://www.zhihu.com/question/20977382
- https://www.zhihu.com/question/19772767
- Understanding of Internal Clustering Validation Measures