# 大数据原理与技术 Lab8

20337025 崔璨明

## 1、实验要求

- 下载并运行GenLouvain算法
- 从SNAP网站下载相应的数据集，测试GenLouvain算法
- 通过NMI指标评估算法结果。
- 通过modularity指标评估算法结果。
- 通过实验结果分析NMI和modularity两个指标之间的关系。

## 2、数据集

选取了https://snap.stanford.edu/data/index.html上的email-Eu-core数据集，该数据集为无向图，有1005个结点25571条边，带有真实社区标签，社区数目为42。并在数据集上做了一些处理，删掉了描述文字。

## 3、实验代码

下载GenLouvain算法，编写调用程序：

```
% 读取文本文件
data = dlmread('email-Eu-core.txt');
data = data + 1;
% 顶点数
num_vertices = max(max(data));
% 稀疏矩阵
adj_matrix = sparse(data(:, 1), data(:, 2), 1, num_vertices, num_vertices);
adj_matrix = adj_matrix + adj_matrix.'; % 对称化，将其表示为无向图
% 计算模块化矩阵
gamma = 2; % 设置 gamma 值
%twom为网络图的加权边数的两倍
[modularity_matrix, twom] = modularity(adj_matrix, gamma);
% 读取真实的社区标签文件
ground_truth = dlmread('email-Eu-core-department-labels.txt');
ground_truth(:, 1:2) = ground_truth(:, 1:2) + 1;
% 调用genlouvain函数进行社区检测
[S, Q] = genlouvain(modularity_matrix,42);
% 计算NMI
nmi_value = compute_nmi(ground_truth(:, 2), S);
% 计算模块度
modularity_value = Q/twom;
% 显示结果
fprintf('community num: %d\n',max(S));
fprintf('NMI: %f\n', nmi_value);
fprintf('Modularity: %f\n', modularity_value);
% 将社区结果写入文本文件
fileID = fopen('community_results.txt', 'w');
fprintf(fileID, 'NodeID\tCommunityID\n');
for i = 1:length(S)
```

```
    fprintf(fileID, '%d\t%d\n', i, S(i));
end
fclose(fileID);
```

`compute_nmi()`为自定义的计算nmi指标的函数，具体代码就不在此赘述，见`compute_nmi.m`文件。

# 4、实验结果

运行程序，得到结果如下，**社区划分的数目为44**，原数据集的真实社区数为42，非常接近；**NMI为0.67，modularity指标为0.3399**，具体的划分结果见result目录下的`community_results.txt`文件。

```
community num: 44
NMI: 0.670023
Modularity: 0.339907
>>
```

# 5、分析

NMI和模块度（Modularity）都是用于评估社区检测算法的常见指标。它们提供了关于社区结构质量和相似度的信息，但从不同的角度进行衡量：

- NMI是一种基于信息论的指标，用于度量两个社区分配结果之间的相似度。它考虑了社区中节点的互信息和熵，通过归一化计算得到一个在 [0, 1] 范围内的值。NMI值越接近1，表示两个社区分配结果越相似，而值越接近0，表示两个社区分配结果越不相似。
- Modularity是一种基于图结构的指标，用于评估社区划分的质量。它度量了实际社区内部连接的紧密程度与预期随机网络内部连接的差异。Modularity值介于 [-1, 1] 范围内，值越接近1表示社区划分更好，而值越接近0或负值表示社区划分较差。

实验结果显示，NMI为0.646072，表示两个社区分配结果之间的相似度较高。而Modularity为0.340029，表示社区划分的质量一般，可能还存在改进的空间。