

第 4 周. 包和模块

题干

图是非常重要的—种数据结构，常用于描述社交网络。本次作业提供了 twitch_gamers 数据集（见在线平台课程资料 twitch_gamers.zip），希望基于该数据，构建一个 python 程序包，并在对应模块中实现：读取并存储节点信息，建立无向的社交网络，以及实现相关统计和可视化功能。

1. 该数据中，large_twitch_features.csv 中每一行为一个节点的相关属性，large_twitch_edges.csv 中第一行为一条边。具体详细信息请阅读 README.txt。
2. 建立包 GraphStat，实现网络的构建、分析与可视化。其中
 1. 包 Graph，用以实现点和图结构的创建，以及相关的基础统计功能
 1. 实现 node.py 模块
 1. 实现函数 init_node()，从数据文件中加载所有节点及其属性；
 2. 实现函数 print_node()，利用 format 函数或 f-string，输出某节点的属性
 2. 实现 graph.py 模块，实现图结构的序列化存储和加载。
 3. 实现 stat.py 模块，进行基础的统计分析
 1. 计算网络的节点数、边数、平均度等并返回
 2. 统计某个节点属性的分布
 2. 包 Visualization，基于上述构建的图和节点结构，利用 pyecharts 或 matplotlib 绘制相关的统计结果
 1. 实现 plotgraph.py 模块
绘制网络的局部结构（如某个节点及其所有邻居所组成的 ego 网络）
 2. 实现 plotnodes.py 模块
绘制节点的属性分布，并提供结果的输出或文件保存（图片）

结构示例（仅为参考，可以自行设计其他结构）

GraphStat/

__init__.py

NetworkBuilder/

__init__.py

node.py

```
def init_node()
```

返回字典，key 为节点的 ID，值为该节点对应的各属性值（可以同样设计为字典或列表）

```
def get_xxx(node)...
```

获取节点 node 的 xxx 属性，如度，views，状态等。

```
def print_node(node)
```

显示节点全部信息（利用 format 或者 f 函数）

stat.py

```
def get_node_number(graph)
    计算节点数
def get_edge_number(graph)
    计算边数
def cal_average_dgree(graph)
    计算网络中的平均度
def cal_dgree_distribution(graph)
    计算网络的度分布
def cal_views_distribution(graph)
    计算 views 属性的分布
```

graph.py

```
def init_graph() (可以考虑用 networkx 中的 Graph 等。)
    构建网络
def save_graph(graph):
    序列化图信息
def load_graph(file):
    将网络加载至内存
```

Visualization/

__init__.py

plotgraph.py

```
def plot_ego(graph,node) (附加 :使用 networkx 库中的布局算法可视化结构,
    注意避免结构太大, 复杂可能导致绘制失败, 或者杂乱。)
    绘制节点的局部网络 (找一些度大小合适的节点尝试。)
def plotdgree_distribution(graph) (观察度分布的形态)
    度的分布图
```

plotnodes.py

绘制图中节点属性的统计结果 (附加 : 尝试 matplotlib, seaborn 等对应绘图库的文件输出, 尤其是矢量图 eps 等格式的输出)

```
def plot_nodes_attr(graph,属性) (观察属性的分布形态)
```

3. (附加) 观察所构建的网络在平均度, 度分布, 甚至局部结构上的结果和形态, 讨论如何用这些数据来对节点进行排序或者挑选, 比如假设你想在这个网络上营销一个新游戏, 应该找那些节点来 “试用”, 以快速地产生口碑? 了解一些节点重要性的一些常见指标。

4. (附加) 了解并使用 Gephi 工具, 尝试网络结构的可视化与社团分析等。