经济管理学院

课　程　报　告

（复杂网络与社会计算）

题　　目：情绪计算与传播

课程教师：赵吉昌

学院/专业：信息管理与信息系统

学生姓名：郭加璐

学　　号：21377225

作业内容：

1. 阅读本周参考文献。

2. 本周提供的数据文件 weibograph.txt 共 227,122 行， 每行表示一条边，即一条用户连接，每行分为四列，第一列和第二列为有连接关系的两个用户，第三列为两用户之间的转发数，即连接强度。 第四列为一个列表，其中数据为第一列用户的情绪计算结果，列表中的四个数值分别为愤怒、厌恶、高兴、悲伤四种情绪微博的数量。例如，文件第一行为 0　1　12　[151, 97, 385, 135]，含义如下： 用户 0 和 1 之间有连接关系，他们互相转发了 12 条微博数据。其中，用户 0 所发布的愤怒、厌恶、高兴和悲伤四种情绪的微博数量分别为 151, 97, 385, 135。请利用该文件，尝试计算对于特定距离h的用户对之间，不同情绪的相关性，以及该相关性随h的变化趋势。

3.（附加）围绕这一数据，你还能想到哪些分析的思路？尝试实现并对结果进行讨论。

功能实现及分析：

1. 读入数据，构建社交网络

读取 weibograph.txt 中的数据，构建网络。其中，每一行为一条边，第一列和第二列为用户，第三列为转发数（作为边的权重），第四列为用户 1 的情绪计算结果（包含愤怒、厌恶、高兴和悲伤）。

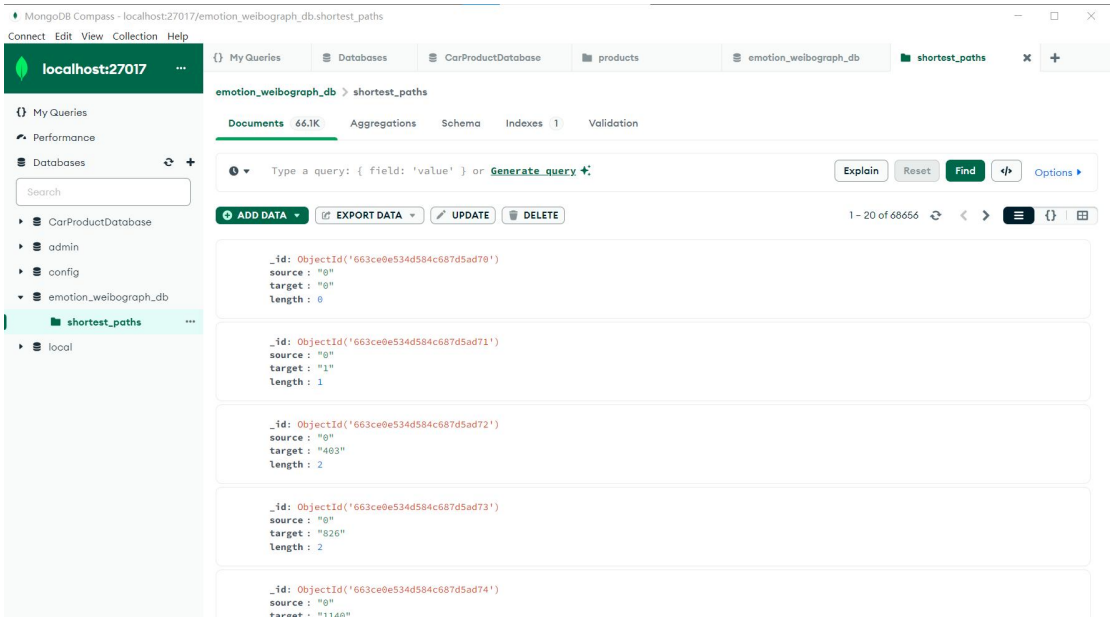
```
# 构建网络
def read_graph(dataFilePath):
    G = nx.Graph()
    with open(dataFilePath, 'r') as file:
        data = [line.strip().split() for line in file]
    for line in data:
        # 每行表示一条边
        user1 = line[0]  # 用户 1
        user2 = line[1]  # 用户 2
        share = int(line[2])  # 转发数（连接强度）
        emotions = list(line[3])  # 用户 1 的情绪列表[愤怒, 厌恶, 高兴, 悲伤]
        G.add_edge(user1, user2, weight=share)
        G.nodes[user1]['emotions'] = emotions
    return G
```

2. 计算最短路径长度

计算最短路径长度并保存到 MongoDB 数据库：

```
# 保存最短路到数据库
def store_shortest_path_lengths_database(G):
    client = MongoClient('mongodb://localhost:27017')
    db = client['emotion_weibograph_db']
    collection = db['shortest_paths']
    collection.drop()
    # 计算并保存最短路
    for source, path_lengths in nx.all_pairs_shortest_path_length(G):
        for target, length in path_lengths.items():
            collection.insert_one({'source': source, 'target': target,
                                   'length': length})
```

保存结果如下：



3. 计算情绪相关性

读取最短路径长度，计算特定距离用户对之间，不同情绪的相关性。其中，设定距离取值为 1-5。

```
# 读取最短路径长度
def get_path_length(source, target):
    client = MongoClient('mongodb://localhost:27017')
    db = client['emotion_weibograph_db']
    collection = db['shortest_paths']
    result = collection.find_one({'source': source, 'target': target})
    if result:
        return result['length']
    else:
```

```
        return None

# 计算 pearsonr 相关性
def correlation(data):
    bootstrap_samples = 10000
    bootstrap_correlations = [pearsonr(*zip(*[random.choice(data) for _ in
                              range(len(data))]))[0] for _ in
                              range(bootstrap_samples)]
    return np.mean(bootstrap_correlations),
           np.std(bootstrap_correlations)

# 计算情绪相关性
def emotion_correlation(G, h):
    anger = []
    disgust = []
    joy = []
    sad = []
    for i in G.node():
        for j in G.node():
            if i<j:
                path_length = get_path_length(i, j)
    if path_length == h:
                    anger.append([G.nodes[i]['emotions'][0],
                          G.nodes[j]['emotions'][0]])
                    disgust.append([G.nodes[i]['emotions'][1],
                          G.nodes[j]['emotions'][1]])
                    joy.append([G.nodes[i]['emotions'][2],
                          G.nodes[j]['emotions'][2]])
                    sad.append([G.nodes[i]['emotions'][3],
                          G.nodes[j]['emotions'][3]])
    anger_correlation, anger_error = correlation(anger)
    disgust_correlation, disgust_error = correlation(disgust)
    joy_correlation, joy_error = correlation(joy)
    sad_correlation, sad_error = correlation(sad)
    return (anger_correlation, anger_error), (disgust_correlation,
              disgust_error), (joy_correlation, joy_error),
              (sad_correlation, sad_error)
```

保存情绪相关性计算结果：

```
if __name__=='__main__':
    dataFilePath = "./data/weibograph.txt"
    G = read_graph(dataFilePath)  # 构建网络
    store_shortest_path_lengths_database(G)  # 保存最短路数据
    hs = [1, 2, 3, 4, 5]  # 特定距离 1~5
    result = {}  # 保存结果
```

```
    for h in hs:
        result[h] = emotion_correlation(G, h)
    # 保存计算结果
    with open('./output/result.csv', 'w', newline='') as file:
        writer = csv.writer(file)
        headers = ['h'] + [f'{emotion} Correlation, {emotion} Error' for
                  emotion in ['Anger', 'Disgust', 'Joy', 'Sad']]
        writer.writerow(headers)
        for h, correlations in result.items():
            row = [h] + [value for stats in correlations.values() for value
                         in stats]
            writer.writerow(row)
```

4. 结果可视化

根据之前保存的计算结果对不同情绪的相关性随距离 h 的变化趋势进行可视化：

```
import pandas as pd
import matplotlib.pyplot as plt
# 读取数据
anger_files = [f"./output/pearson_anger_distance_{i}.csv" for i in range(1,
              6)]
disgust_files = [f"./output/pearson_disgust_distance_{i}.csv" for i in
                 range(1, 6)]
joy_files = [f"./output/pearson_joy_distance_{i}.csv" for i in range(1, 6)]
sad_files = [f"./output/pearson_sad_distance_{i}.csv" for i in range(1, 6)]

anger_data = [pd.read_csv(file) for file in anger_files]
disgust_data = [pd.read_csv(file) for file in disgust_files]
joy_data = [pd.read_csv(file) for file in joy_files]
sad_data = [pd.read_csv(file) for file in sad_files]

anger_correlations = [df['mean_correlation'].mean() for df in anger_data]
disgust_correlations = [df['mean_correlation'].mean() for df in
disgust_data]
joy_correlations = [df['mean_correlation'].mean() for df in joy_data]
sad_correlations = [df['mean_correlation'].mean() for df in sad_data]

# 距离取值 1~5
x_values = range(1, 6)

# 绘制折线图
plt.figure(figsize=(10, 6))
plt.plot(x_values, anger_correlations, marker='o', color='red',
         label='Anger')
```

```
plt.plot(x_values, disgust_correlations, marker='*', color='green',
         label='Disgust')
plt.plot(x_values, joy_correlations, marker='^', color='orange',
         label='Joy')
plt.plot(x_values, sad_correlations, marker='s', color='blue',
         label='Sad')
plt.xlabel('Distance($h$)')
plt.ylabel('Mean Pearson Correlation')
plt.title('Correlation for Emotions over Distance')
plt.legend()
plt.grid(True, linestyle='--')
# 保存图像
plt.savefig("./image/pearson_correlation_distance.png")
plt.show()
```

得到结果如下：可以发现不同情绪的相关性（愤怒、厌恶、高兴和悲伤）整体呈现同样的变化趋势，即随着距离 h 的增加，相关性逐渐下降，说明用户间情绪表达的同步性在距离较近时更强。具体来看，愤怒在 h=1 时具有最高的初始相关性，随着 h 的增加其下降趋势最为显著，表明愤怒情绪的传播在近距离用户之间更为密切；高兴的初始相关性和下降速率仅次于愤怒，对距离也很敏感；二厌恶和悲伤情绪相对具有更低的初始相关性，且随 h 的变化更加平稳，说明厌恶和悲伤情绪的传播更加独立，受到其他用户的影响较小。