



Universidade do Minho
Mestrado Integrado em Engenharia Informática
4ºano - 1º Semestre

Inteligência Ambiente : Tecnologias e Aplicações

Questão de Aula nº 2



a83732 – Gonçalo Rodrigues Pinto

Janeiro de 2021

Conteúdo

1	Introdução	3
2	Descrição do Problema	3
3	Descrição do Trabalho	4
3.1	Criação do gráfico	4
3.2	Análise Desenvolvida	5
3.2.1	Distribuição de grau	5
3.2.2	Diâmetro e distância	6
3.2.3	Medidas de centralidade	9
4	Conclusão	11

Lista de Figuras

1	Grafo criado	4
2	Grafo considerado	5
3	Distribuição de grau.	6
4	Diâmetro obtido.	7
5	Centro do grafo considerado.	7
6	Grau do centro do grafo considerado.	7
7	Caminho mais curto como respectivo tamanho.	8
8	<i>Eccentricity</i>	8
9	Centralidade do grau	9
10	<i>Eigenvector centrality</i>	9
11	<i>Closeness centrality</i>	10
12	<i>Betweenness centrality</i>	10
13	Centralidade dos nós	11

1 Introdução

No 1º semestre do 4º ano do Curso de Engenharia Informática da Universidade do Minho, existe uma unidade curricular complementar denominada por Inteligência Ambiente Tecnologias e Aplicações, que tem como objectivo introduzir aos estudantes conhecimentos sobre as potencialidades da Inteligência Ambiente, sua caracterização, bem como das tecnologias e metodologias de resolução problemas, mais comuns nesta área, sendo estes capazes de as contextualizar, aplicar e analisar.

O presente trabalho teve como principal objectivo sensibilizar e motivar os alunos para a concepção, manipulação e aplicação de conhecimento em Redes Sociais utilizando a linguagem Python e a biblioteca NetworkX.

2 Descrição do Problema

Neste trabalho foi disponibilizado um ficheiro texto, *lista_arestas.txt*, que serviu de base para a construção da rede social que se pretendeu estudar. Este ficheiro tem a seguinte estrutura (na ordem a seguir apresentada) :

1. Nó de origem ;
2. Nó de destino ;
3. Força da ligação, isto é, 5 equivale a uma forte ligação enquanto 1 equivale a uma fraca ligação ;
4. Nível que cada ligação foi verificada, ou seja, 1 corresponde a um contacto próximo confirmado, 2 a várias interacções registadas e por fim 3 demonstra uma ligação potencial ou planeada que necessita de confirmação.

Com base nos seguintes dados pretendeu-se criar um grafo com os mesmos e desenvolver as seguintes análises sobre:

- a Distribuição de grau
- b Diâmetro e distância
- c Medidas de centralidade

tal medida recorreu-se à função `nx.degrees` e de forma ilustrar esses valores representou-se na forma de um histograma.

```
degs = [i[1] for i in component.degree()]
fig, ax = plt.subplots(figsize=(13, 9))
ax.get_xaxis().tick_bottom()
ax.get_yaxis().tick_left()
ax.hist(degs, color="#3F5D7D", bins='auto')
plt.show()
```

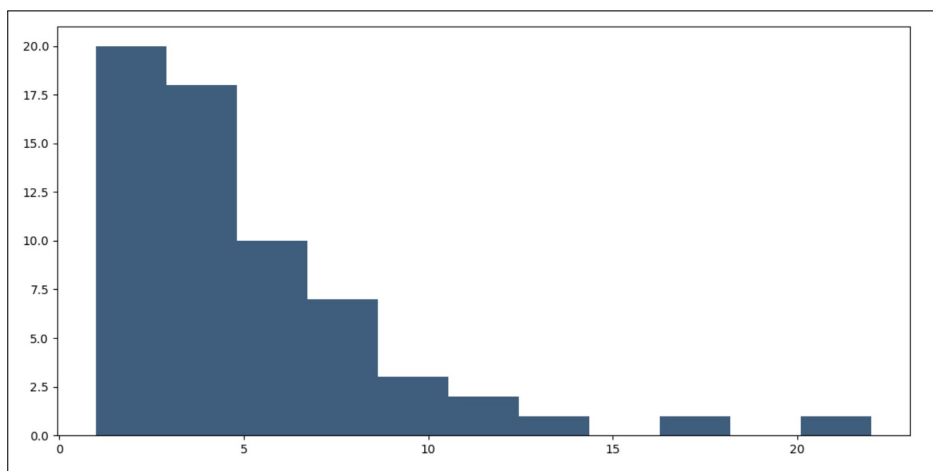


Figura 3: Distribuição de grau.

Como podemos observar, existem apenas alguns vértices com graus altos e muitos com graus relativamente pequenos - um resultado típico para redes sociais.

3.2.2 Diâmetro e distância

Outro dos parâmetros analisados foi o diâmetro, este valor num grafo traduz a maior distância entre dois vértices. É fornecido pela função `nx.diameter`, é de realçar que esta função apenas é calculada quando o grafo é conectado.

```
if nx.is_connected(component):
    print('Diameter = ', nx.diameter(component))
else:
    print('Found infinite path length because the graph is not connected ')
```

```
Diameter = 6
```

Figura 4: Diâmetro obtido.

Tal como a figura 4 indica a maior distancia obtida neste sub-grafo foi de 6 unidades.

No mesmo seguimento foram calculados algumas unidades de distância, tais como:

- **Centro do grafo** - O centro é o conjunto de nós com excentricidade igual ao raio, através da função `nx.center` obteve-se esse nó.

```
center = nx.center(component)
print('Center = ', center)
```

```
Center = ['Mohamed Atta']
```

Figura 5: Centro do grafo considerado.

Tal como a figura 5 indica o nó central é denominado de Mohamed Atta.

- **Grau do centro do grafo** - tal como foi apresentado previamente o grau de um nó numa rede é o número de conexões que esse nó tem com outros nós, podemos calcular individualmente através da chamada à `nx.degree`. Para exemplificação calculou-se o grau do centro.

```
print(nx.degree(component, center))
```

```
[('Mohamed Atta', 22)]
```

Figura 6: Grau do centro do grafo considerado.

Tal como a figura 6 indica o grau do nó central, denominado de Mohamed Atta, é 22, ou seja, este nó possui 22 conexões com outros nós. Retornando à figura 3 aquele vértice que possui o maior grau é este apresentado.

- **Caminho mais curto como respectivo tamanho** - uma outra unidade de distância analisada foi o caminho mais curto e respectivamente tamanho para tal recorreu-se à função `nx.shortest_path` e `nx.shortest_path_length`, respectivamente. Para exemplificação calculou-se caminho mais curto e tamanho do mesmo entre dois nós que previamente viu-se na visualização visual (figura 2) que estavam distantes.

```
print('Shortest path between Abu Zubeida and Nabil al-Marabh is',
      nx.shortest_path(component, 'Abu Zubeida', 'Nabil al-Marabh'))
print('Length of shortest path between Abu Zubeida and Nabil al-Marabh is',
      nx.shortest_path_length(component, 'Abu Zubeida', 'Nabil al-Marabh'))
```

```
Shortest path between Abu Zubeida and Nabil al-Marabh is
['Abu Zubeida', 'Djamal Beghal', 'Zacarias Moussaoui', 'Mohamed Atta', 'Satam Suqami', 'Nabil al-Marabh']
Length of shortest path between Abu Zubeida and Nabil al-Marabh is 5
```

Figura 7: Caminho mais curto como respectivo tamanho.

Tal como a figura 7 indica o caminho mais curto entre o nó Abu Zubeida and Nabil al-Marabh é ['Abu Zubeida', 'Djamal Beghal', 'Zacarias Moussaoui', 'Mohamed Atta', 'Satam Suqami', 'Nabil al-Marabh'], cujo comprimento é 5. Tal como era espectável existiu a necessidade de percorrer o nó central.

- ***Eccentricity*** - a excentricidade de um nó é a distância máxima desse para todos os outros nós no grafo. Para exemplificação o nó considerado foi o central utilizando a função `nx.eccentricity`.

```
print('Eccentricity = ', nx.eccentricity(component, 'Mohamed Atta'))
```

```
Eccentricity = 3
```

Figura 8: *Eccentricity*

O valor apresentado na figura 8 traduz que do nó central para qualquer outro no máximo tem que se percorrer apenas 3 nós.

3.2.3 Medidas de centralidade

O último parâmetro considerado foram medidas de centralidade no seguimento de algumas análises apresentadas anteriormente é importante realçar que o nó central é denominado de Mohamed Atta cujo grau é 22. Portanto as medidas analisadas foram:

- **Centralidade do grau** - o número de conexões de um nodo para todos os outros, o valor 1 representa que está ligado a todos. A função utilizada foi `nx.degree_centrality`.

```
print(nx.degree_centrality(component))
```

```
{'Hani Hanjour': 0.2096774193548387, 'Majed Moqed': 0.06451612903225806, 'Nawaf Alhazmi': 0.1774193548387097,
```

Figura 9: Centralidade do grau

Dada a extensão do resultado obtido não é possível apresentar o resultado na totalidade pois esta medida analisa todo o gráfico.

- ***Eigenvector centrality*** - representa o quão importante é um nodo em função de quão bem conectado está. A função utilizada foi `nx.eigenvector_centrality`.

```
print(nx.eigenvector_centrality(component))
```

```
{'Hani Hanjour': 0.24855477543754168, 'Majed Moqed': 0.07494537631193433, 'Nawaf Alhazmi': 0.14414554325478365,
```

Figura 10: *Eigenvector centrality*

Dada a extensão do resultado obtido não é possível apresentar a resposta na totalidade pois esta medida analisa todo o gráfico.

- ***Closeness centrality*** - traduz a importância de um nodo em função da sua proximidade com os outros da rede. A função utilizada foi `nx.closeness_centrality`.

```
print(nx.closeness_centrality(component))
```

```
{'Hani Hanjour': 0.4397163120567376, 'Majed Moqed': 0.3263157894736842, 'Nawaf Alhazmi': 0.4397163120567376,
```

Figura 11: *Closeness centrality*

Dada a extensão do resultado obtido não é possível apresentar a resposta na totalidade pois esta medida analisa todo o gráfico.

- ***Betweenness centrality*** - quantifica quantas vezes um nodo aparece nos caminhos mais curtos entre dois nodos. A função utilizada foi `nx.betweenness_centrality`.

```
print(nx.closeness_centrality(component))
```

```
{'Hani Hanjour': 0.1236167484316612, 'Majed Moqed': 0.0, 'Nawaf Alhazmi': 0.15207834940780468, 'Khalid AL-Mihdhar': 0.005429226158998767,
```

Figura 12: *Betweenness centrality*

Dada a extensão do resultado obtido não é possível apresentar a resposta na totalidade pois esta medida analisa todo o gráfico.

Por fim, demonstrou-se visualmente com recurso a cores e à saliência do tamanho quais são os nós mais centrais na rede para tal, através do seguinte algoritmo:

```
pos = nx.spring_layout(component)
betCent = nx.betweenness_centrality(component,
    normalized=True, endpoints=True)
node_color = [20000.0 * component.degree(v) for v in component]
node_size = [v * 10000 for v in betCent.values()]
plt.figure(figsize=(10, 10))
nx.draw_networkx(component, pos=pos, with_labels=True,
    node_color=node_color, node_size=node_size)
plt.axis('off')
plt.show()
```

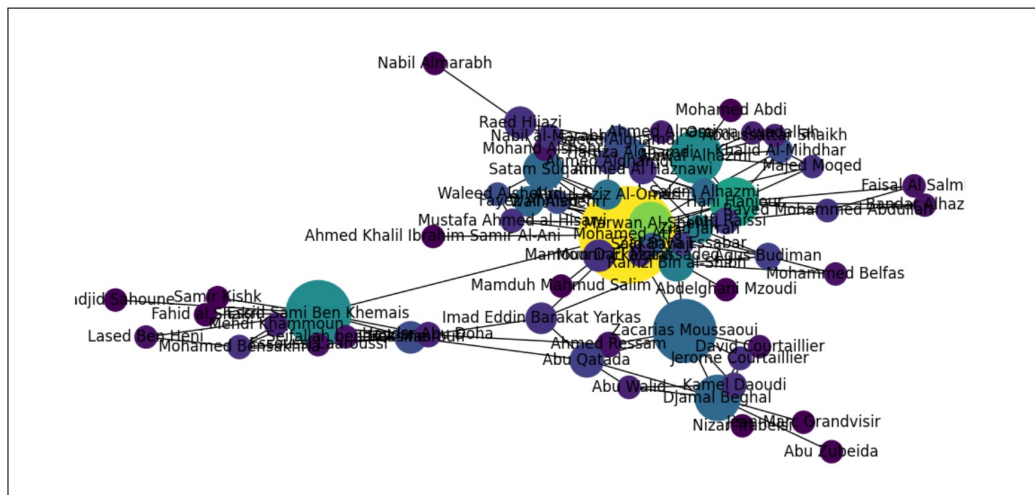


Figura 13: Centralidade dos nós

Como pode ser observado o nó a amarelo representa o nó central apresentado previamente.

4 Conclusão

O presente relatório descreveu, de forma sucinta, o desenvolvimento do processo de análise de uma rede através de um ficheiro fornecido utilizando a linguagem Python e a biblioteca NetworkX.

Após a realização deste trabalho, fiquei consciente das potencialidades que a concepção, manipulação e aplicação de conhecimento em Redes Sociais pode trazer no processo de tomadas de decisões, estudo social, etc.

Considero que os principais objectivos foram cumpridos.

Senti também que a realização deste trabalho prático consolidou os meus conhecimentos nas tecnologias e aplicações no âmbito da Inteligência Ambiente.