

Instituto Superior Técnico



Mestrado em Engenharia Informática e de Computadores

Ciências das Redes Complexas

Responsável da disciplina: Francisco Correia dos Santos

Projeto 1

Grupo 15

89905, João Gomes

98649, Pedro Guerra

98668, Ricardo Gonçalves

Lisboa, 2020

Introdução

O presente trabalho analisa, através dos parâmetros abordados na disciplina, duas redes bastante distintas. Uma social que diz respeito a uma aos membros de gangues da Itália, e uma de dependências de software que ilustra as dependências do JUnit.

Escolhemos redes com conceitos bastante diferentes para percebermos qual o tema que mais nos interessava desenvolver no próximo projeto da disciplina.

Neste projeto recorreremos à programação para calcular algumas das métricas apresentadas. Achámos que esta era a melhor forma de perceber e consolidar conceitos de análise de redes.

Resultados Obtidos

	Italian Gangs	JUnit 4.8.1 Dependencies
Nodes	67	128
Edges	114	311
Average Degree, <k>	3.403	2.4297 (in) 2.4297 (out)
Average Shortest Path Length	3.012 (giant component), 1.0 (other component)	-
Average Clustering Coefficient	0.435	0.17
Centrality	Betweenness Centrality { 18 (Nacionalidade 3): 0.556, 62 (Nac. 4): 0.363, 17 (Nac. 5): 0.288, 46 (Nac. 3): 0.14, 3 (Nac. 1): 0.131, 10 (Nac. 3): 0.107, ...}	Degree Centrality { 129: 0.323, 137: 0.244, 138: 0.205, 148: 0.197, 179: 0.197, 144: 0.134}

Discussão

Italian Gangs

A *label* de cada nó representa a nacionalidade do membro do gangue, é gerada uma ligação quando dois membros trabalham juntos no mesmo gangue.

Analisando a representação da *network*, conseguimos observar a ligação de indivíduos de diferentes nacionalidades nos gangues italianos, a partir de trabalhos que fizeram em conjunto.

Existe uma grande discrepância entre o *degree* dos nós, sendo possível perceber que existem poucos nós com *degree* elevado e bastantes com *degree* reduzido. Isto mostra que embora haja alguns elementos que tenham algum tipo de conexão com grande parte dos membros, a maioria tende a trabalhar com um pequeno número de membros. Isto pode ser concluído através da análise da topologia da *network* e da *degree distribution*. Observando

a última podemos ver que é possível aproximá-la através de uma *power-law*, o que faz desta *network* uma *Scale-Free Network*. Também é possível observar um *high-degree cutoff*.

O *average path length*, tal como esperado devido às dimensões da rede, é pequeno. Existem 2 componentes, um *giant component* que tem um APL de 3.012 e um componente desconectado com um APL de 1, uma vez que apenas tem 2 nós. Fazendo uma análise mais cuidadosa percebemos que, apesar de a maioria dos nós terem *degree* baixo, existem *hubs* que estão fortemente ligados com toda a rede, criando assim caminhos mais curtos entre todos os membros do gangue.

Dentro dos gangues, existe uma ligeira tendência para os nós se agruparem. Existindo os membros “A” e “C” com um vizinho comum “B”, há a possibilidade de “A” e “C” terem uma ligação, como é possível ver através do *average clustering coefficient*, 0.43.

Utilizando o “*Louvain method*”, é possível analisar as diferentes comunidades formadas pelos membros do gangue italiano. Analisando a nacionalidade das comunidades, é perceptível que os indivíduos de nacionalidade 4 têm uma grande tendência para estabelecerem uma conexão com indivíduos da mesma nacionalidade. Por outro lado, os membros do gangue com as restantes nacionalidades tendem a estabelecer conexão com membros de qualquer nacionalidade. A partir destes dados podemos especular que a razão para tal acontecer é a incompatibilidade de costumes das diversas nacionalidades.

Decidimos utilizar a *Betweenness Centrality* para inferir a importância de cada nó da *network*. Utilizando este método é facilmente observado que existem múltiplos membros que têm uma grande quantidade de conexões. Devido às conexões destes membros é possível ter a *network* maioritariamente conectada. Mas, caso alguns destes nós, que podem ser considerados *hubs*, fossem eliminados, como existem nós que ligam secções que estão ligadas com apenas um ou dois nós, a *network* seria fragmentada, dividindo assim o *giant component* em novos componentes.

Após analisar a *network*, formulámos alguns princípios que podem ter levado à criação e organização da mesma. A criação do *giant component* pode ter ocorrido da junção de duas redes diferentes. Observando a Figura 1, é possível distinguir duas zonas (com nós fortemente ligados) ligadas por apenas 4 caminhos distintos. Outra possibilidade para a criação do *giant component* é os membros com o maior *degree* terem aparecido primeiro e a *network* ter se formado a partir destes, nesta sugestão o *degree* dos nós é diretamente proporcional à antiguidade dos mesmos. Em relação ao pequeno componente, sugerimos que seja o início de um novo gangue, que teria tendência a criar mais ligações e juntar-se ao *giant component*. Outra sugestão para o componente pequeno é esta ter estado ligada ao gangue maior por nós ou ligações que deixaram de existir.

JUnit 4.8.1 Dependencies

Cada nó é um *Software Package* e, sendo um grafo orientado cada ligação corresponde a uma dependência, e.g., Nó A para Nó B significa que o Nó A tem dependência do Nó B.

O *average degree* é baixo pois, embora existam *Software Packages* com muitas dependências, a maioria tem 3 ou menos. Observando as *scatter plots* do *in* e *out degree distribution*, é possível aproximar ambas as distribuições através de *power-laws*. Também é visível na *in degree distribution* um *high-degree cutoff*, devido a esta *network* ser finita e cada *Software Package* ter um número limitado de *packages* dependentes dele.

Dos métodos disponíveis para inferir a centralidade de um nó na *network* utilizámos o *degree centrality*. Consideramos que este é o método mais adequado porque, através do *in-degree* conseguimos concluir quais os *Software Packages* que têm mais *packages* a depender deles, por exemplo, SP 129 com 41 dependentes, SP 137 com 28 dependentes. Através do *out-degree* conseguimos descobrir quais os *Software Packages* com mais dependências, por exemplo, SP 15, SP 43 e SP 170 com 9 dependências. Ao analisar o gráfico da *network* verificamos nós com uma extrema importância pois, tendo um grande *in-degree* se estes fossem removidos, uma grande parte dos *Software Packages* existentes na *network* deixariam de funcionar corretamente.

Não foi possível tirar resultados calculando o *average shortest path length*, pois esta *network* é direcionada e também não é *weakly connected*.

O Clustering Coefficient é muito pequeno, querendo isto dizer que existem poucos *Software Packages* que estão dependentes de outros *Software Packages* e estes estão dependentes uns dos outros. Existindo um SP "A" que dependa dos SPs "B" e "C", é incomum que os SPs "B" e "C" tenham alguma dependência entre si. O único cenário em que faz sentido existir uma dependência entre os SPs dos quais "A" depende é, quando "B", embora dependa de "C" não utilize todas as suas funções ou caso utilize, não as disponibiliza a "A". Neste caso, o SP "A" precisa de criar uma dependência diretamente a "C" para poder utilizar as suas funções não utilizadas pelo "B".

Tendo em conta o tipo de *network*, não conseguimos formular uma boa sugestão para a sua criação. No entanto, pareceu-nos que a *network* começou com os nós que têm maior *in-degree* e que os nós com menor *in-degree* foram criados posteriormente. Os nós que têm *in-degree* nulo podem ser, por exemplo, SPs para teste do próprio JUnit ou SPs de interfaces para o utilizador.

Referências

- J. Coutinho, "Italian Gangs." (2016), <https://sites.google.com/site/ucinetsoftware/datasets/covert-networks/italiangangs>
- L. Šubelj, & M. Bajec, "Community structure of complex software systems: Analysis and applications." *Physica A* 390(16), 2968-2975 (2011), <https://www.lovre.appspot.com/support.jsp>, <https://www.lovre.appspot.com/resources/research/networks/ssc/junit> (dataset file), <https://www.lovre.appspot.com/publications.jsp?show=ssc#ssc>
- Repositório Github, https://github.com/ricardo-martins-goncalves/CRC_Project1

Anexo

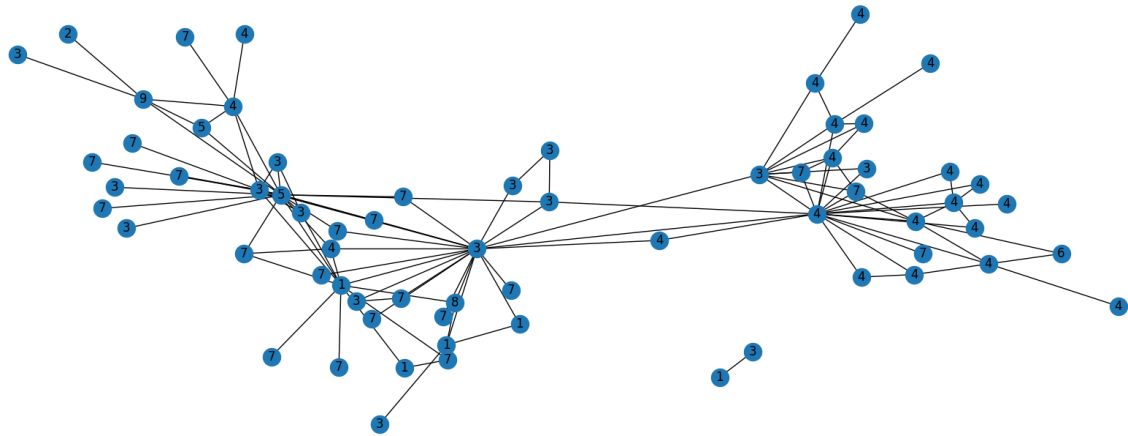


Figura 1 – “Italian Gangs” Network Topology

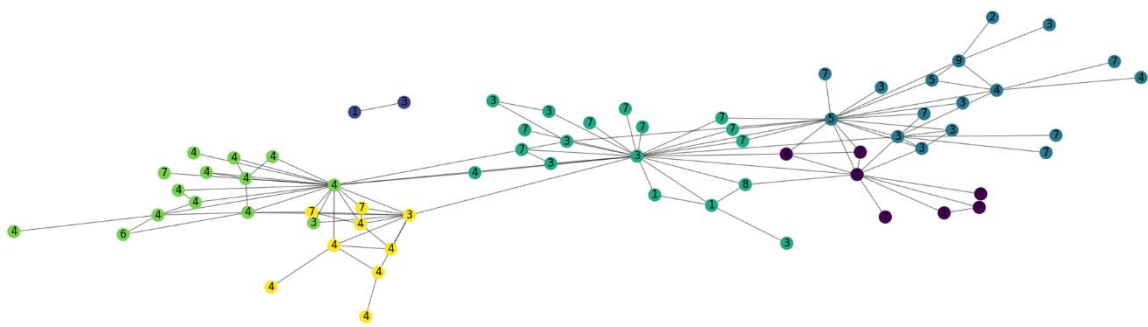


Figura 2 – “Italian Gangs” Network Louvain Communities

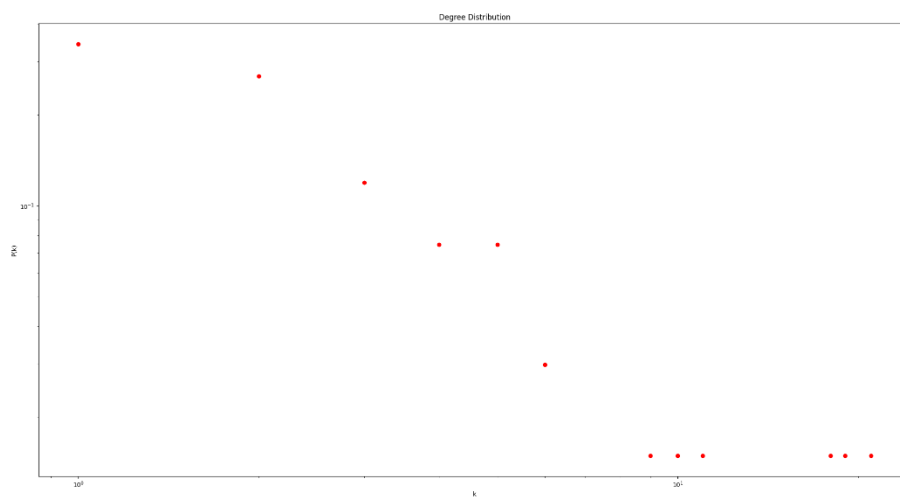


Figura 3 – “Italian Gangs” Network Degree Distribution

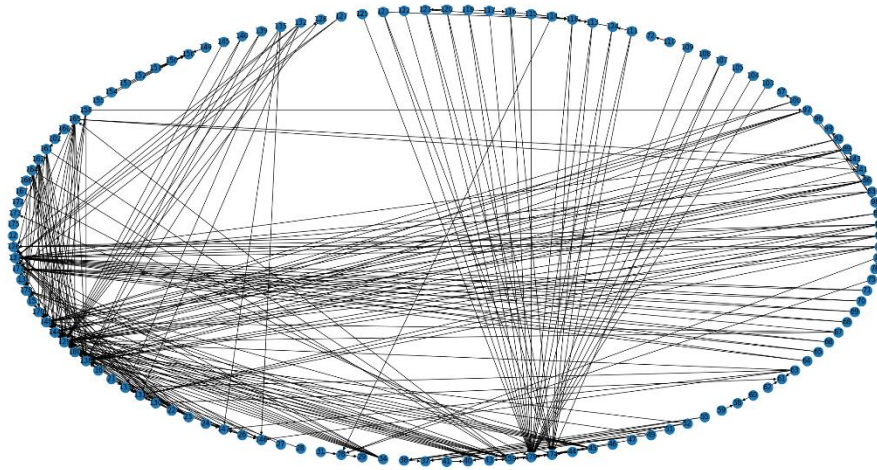


Figura 4 – "JUnit 4.8.1 Dependencies" Network Topology

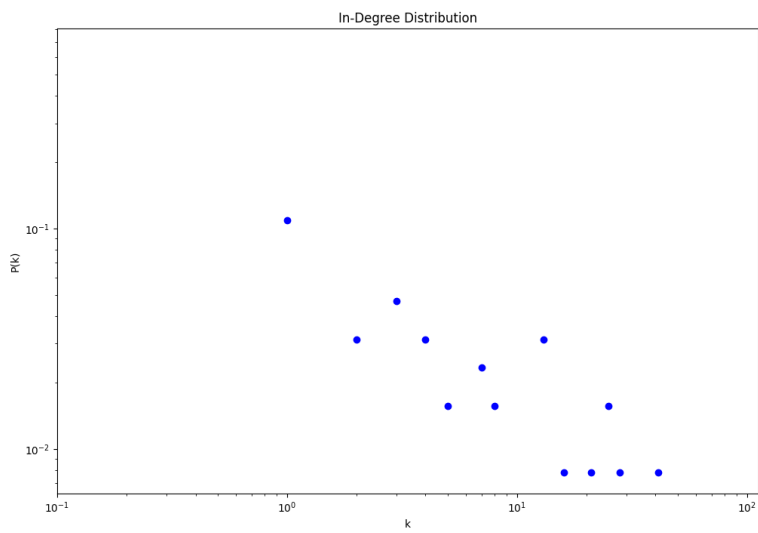


Figura 5 – "JUnit 4.8.1 Dependencies" Network In Degree Distribution

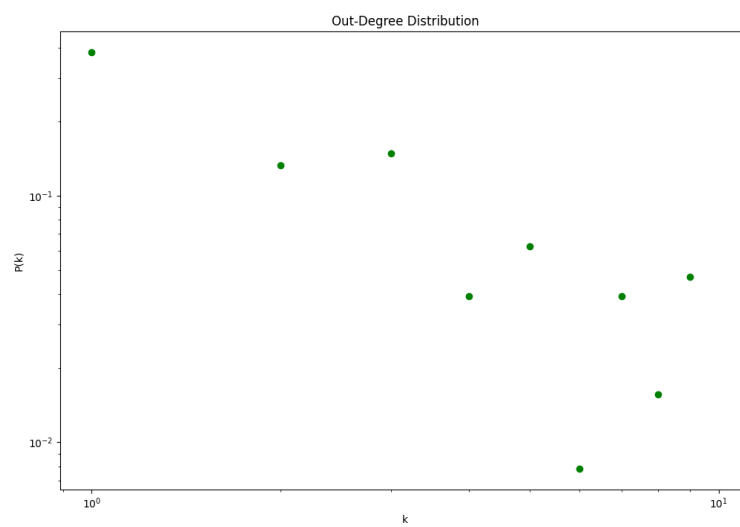


Figura 6 – "JUnit 4.8.1 Dependencies" Network Out Degree Distribution