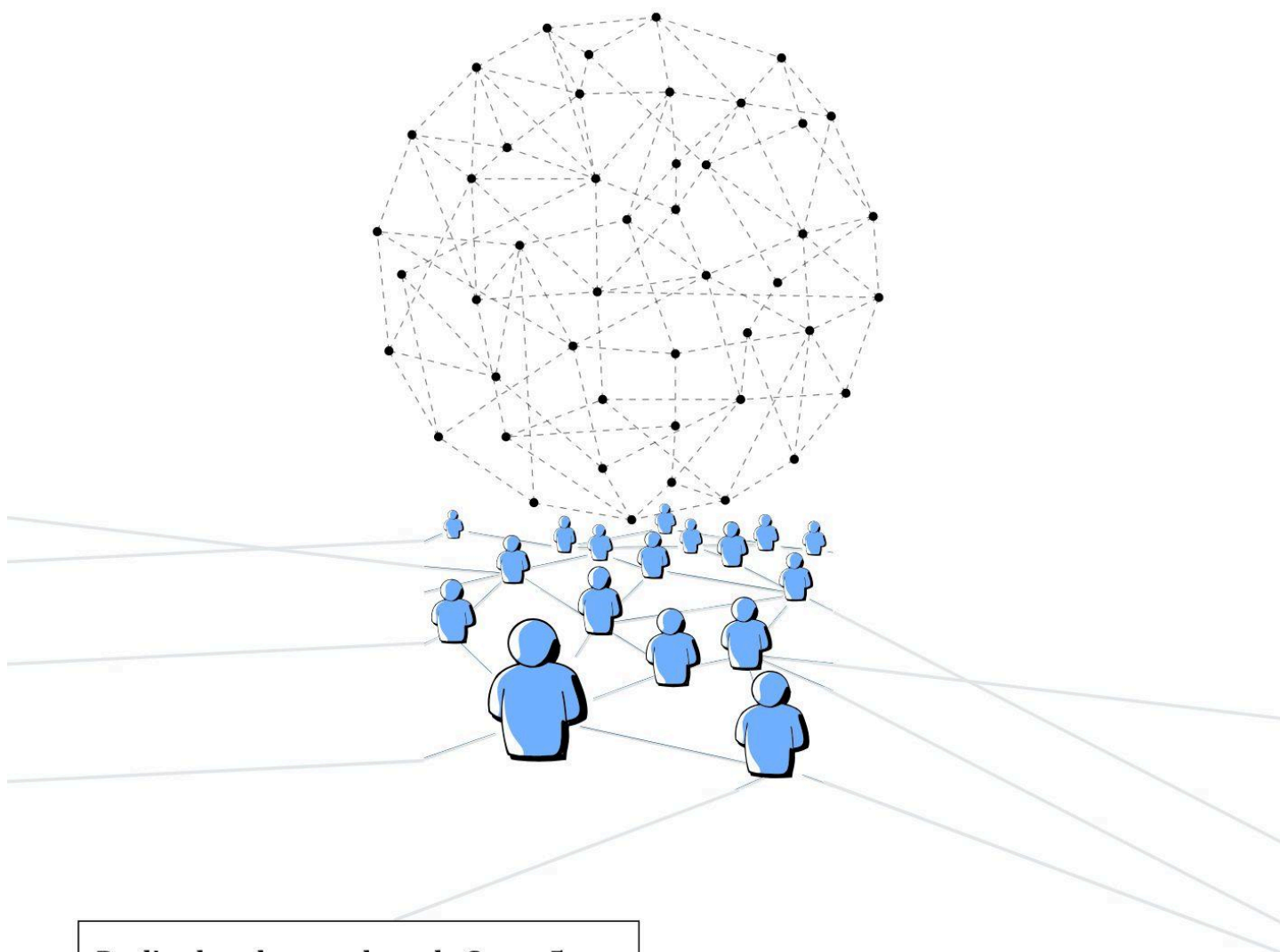


DETEÇÃO DE COMUNIDADES NUMA REDE SOCIAL

No âmbito da UC Análise de Redes



Realizado pelos membros do Grupo 5:

- Rita Guerreiro | nº 112018 | CDC2
- Tomás Monge | nº 98574 | CDC2
- Miguel Celestino | nº 111590 | CDC2

**RELATÓRIO
TRABALHO 2**

Índice

Introdução.....	2
Análise Inicial da Rede.....	3
Q1. Aplicação e análise de quatro métodos de detecção de comunidades.....	5
Q1.1 Método da Remoção de Pontes.....	6
Q1.2 Algoritmo Cluster Fast Greedy.....	9
Q1.3 Algoritmo de Louvain.....	10
Q1.4 Método da Propagação de Etiquetas.....	13
Q2. Comparação e comentário dos resultados obtidos com os quatro métodos.....	17
Comentários Adicionais.....	18
Conclusão.....	19
Referências Bibliográficas.....	20
Apêndice.....	20
Análise inicial das características da Rede.....	20
Secção Q1.....	21
1. Algoritmo de Remoção de Pontes por Girvan e Newman.....	21
2. Algoritmo Cluster Fast Greedy.....	22
3. Algoritmo de Louvain.....	23
4. Algoritmo de Propagação de Etiquetas.....	24
1ª implementação (seed=1):.....	24
2ª implementação (seed=10):.....	25
3ª implementação (seed=23):.....	26
Adicional.....	27
Algoritmo de Louvain com melhor valor de modularidade (seed = 15).....	27
Algoritmo Cluster Optimal.....	28

Introdução

Em Análise de Redes, a deteção de comunidades desempenha um papel importante, uma vez que permite identificar grupos de nodos fortemente conectados entre si. Esta capacidade possibilita a deteção de *clusters* de nodos que numa primeira observação da rede poderiam passar despercebidos. Existem, assim, diversos métodos de deteção de comunidades classificados de acordo com a sua estratégia de as encontrar. No entanto, *como averiguar qual o melhor método aplicado?*

Este projeto foi realizado no âmbito da unidade curricular Análise de Redes e tem como objetivo aplicar os quatro métodos de deteção de comunidades lecionados, realizando, por fim, um estudo comparativo dos mesmos. Para tal, foi fornecido pela docente um conjunto de ligações num ficheiro *.txt* que correspondem à rede onde esses métodos serão aplicados.

Espera-se que a resposta às questões e o estudo comparativo efetuado permitam a obtenção de conclusões relevantes que serão posteriormente enunciadas no final do projeto.

Análise Inicial da Rede

Através da biblioteca *igraph* da linguagem de programação R, foi possível converter o conjunto de ligações contidas no ficheiro “Ligações T2.txt” numa rede. Esta rede foi designada por rede g, e encontra-se representada na figura abaixo (figura 1).

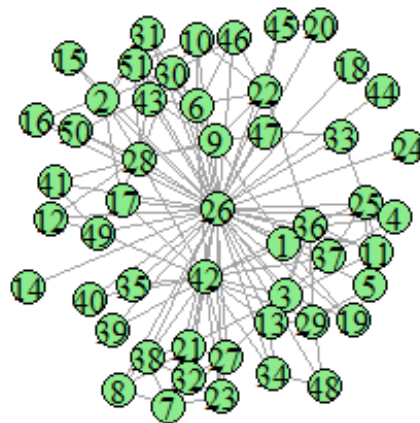


Figura 1: Grafo da Rede g

No contexto do problema, esta é uma rede social ego e não orientada que remete o contacto de pessoas, onde os nodos representam as pessoas e as ligações representam a existência de contacto presencial entre estas. Por ser ego, os indivíduos na rede estabelecem contacto com uma pessoa específica que se situa no centro do grafo, neste caso o nodo 26. Assim, o indivíduo representado pelo nodo 26 conhece todas as restantes pessoas, e cada uma destas além de conhecer o nodo 26, estabelece ou não contacto com mais indivíduos na rede.

Nota: uma rede ego consiste numa subrede onde, de uma rede original, é retirado um nodo aleatório, e com ele todos os nodos adjacentes e as suas ligações.

A tabela seguinte ilustra as propriedades verificadas para uma primeira análise da rede:

N	L	d	k_{min}	k_{max}	$\langle k \rangle$	$\langle l \rangle$	C	K (heterogeneidade de grau)	Nº de cliques
51	155	0.12	1	50	6.08	1.88	0.24	2.3	445

Tabela 1: Tabela referente ao conjunto de características mais relevantes da rede g

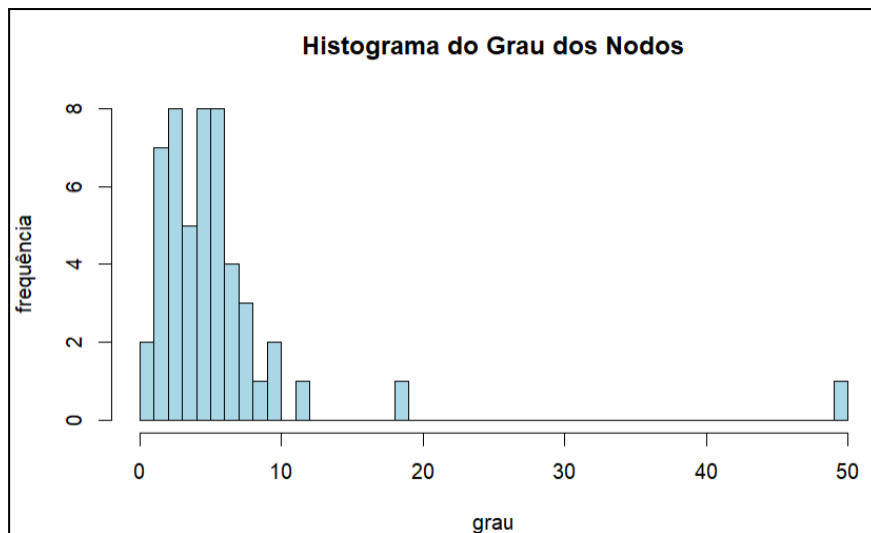


Figura 2: Histograma com a distribuição da frequência dos graus dos nodos da rede g

Inicialmente, verificando a tabela 1, é possível observar que a rede é constituída por 51 pessoas (nº de nodos: $N = 51$), com um total de 155 contactos presenciais entre as mesmas (nº de ligações: $L = 155$). É também possível identificar que as pessoas não tiveram bastante contacto presencial entre si, há exceção do nodo 26, pessoa que teve contacto presencial com toda a gente na rede (densidade consideravelmente pequena: $d = 0.12 < 0.3$). Portanto, a rede é esparsa.

Seguidamente, observa-se na tabela 1 que o menor número de contactos presenciais estabelecidos na rede foi 1, ou seja, há pelo menos uma pessoa na rede que só conhece a pessoa que teve contacto com toda a gente (grau mínimo: $k_{min} = 1$). O grau máximo, como já era de se esperar, é de 50, correspondendo à pessoa identificada pelo nodo 26 que conhece todas as outras na rede sem contar com si própria (grau máximo: $k_{max} = 50$ ($N-1$)). Todas as pessoas que têm contacto com a pessoa central, em média mantêm contacto com 6 pessoas da rede (grau médio da rede: $\langle k \rangle \approx 6.08$).

Também se pode concluir que existe um número muito reduzido de pessoas com bastantes contactos presenciais, face a um número bastante elevado de pessoas que têm poucos conhecidos na rede (o parâmetro de heterogeneidade revela a existência de *hubs*: $K \approx 2.3 \gg 1$). Averigua-se que a rede é de facto heterogénea ao observar o histograma da figura 2.

Verifica-se ainda na tabela 1 que para uma pessoa ter contacto presencial com outra da rede, necessita, provavelmente, de primeiro ter contacto com o indivíduo central, e deste para a pessoa destinada. Em média, o caminho mais curto são cerca de 2 contactos presenciais ($\langle l \rangle \approx 1.88$), e conta-se que essa distância é relativamente pequena ($1.88 \sim \log_{10}(50) \approx 1.69$). Logo, a rede é um *small world*, pois no geral apresenta caminhos curtos ($\langle l \rangle \approx 1.88$) com poucas ligações entre as pessoas ($d = 0.12$).

Outra característica é que, na rede, não é muito comum as pessoas estabelecerem contacto com outras além de com o nodo 26 (o coeficiente de clustering revela que não existe um número significativo de triângulos: $C = 0.24 < 0.3$).

Por último, a rede contém 445 cliques, ou seja, existem 445 subredes completas da rede original com diferentes dimensões onde nelas todas as pessoas se conhecem.

Q1. Aplicação e análise de quatro métodos de detecção de comunidades

Para esta secção serão implementados e estudados 4 métodos de detecção de comunidades na rede g. Esses métodos serão respetivamente:

1. Método da Remoção de Pontes de Girvan e Newman;
2. Algoritmo Clustering Fast Greedy;
3. Algoritmo de Louvain;
4. Método da Propagação de Etiquetas.

Uma boa partição da rede em comunidades remete uma divisão desta em subredes conexas, onde dentro de cada subrede há elevada coesão (muitas ligações internas), e para com outras subredes apresenta separação baixa (poucas ligações externas).

A análise da coesão de comunidades numa partição será baseada nas suas densidades internas (uma comunidade com maior densidade interna é considerada mais coesa, e vice-versa). Já a separação será avaliada pela comparação do número de ligações internas da comunidade com o número de ligações externas (uma comunidade é considerada mais separada das restantes se tiver menos ligações externas do que internas, quanto maior o intervalo de diferença melhor separada a comunidade está).

A medida utilizada para avaliar as partições resultantes dos métodos prévios será a modularidade, a qual mensura a qualidade da divisão da rede em comunidades coesas. A modularidade compara o número de ligações internas de cada *cluster*, com as ligações internas esperadas caso a rede fosse criada por um processo puramente aleatório.

Para ser possível replicar os mesmos resultados, foi decidido o uso de uma semente:

- A semente será 1 para a representação dos grafos (*set.seed(1)*);

- No caso de métodos que seja necessário fazer várias execuções do mesmo código para se obter diferenças nos resultados, foram usadas outras sementes. Contudo, para a visualização dos grafos, será usada apenas a semente 1.

Q1.1 Método da Remoção de Pontes

O Método da Remoção de Pontes de Girvan e Newman é um algoritmo hierárquico e de *clustering* de divisão, utilizado para identificar comunidades numa rede através da remoção de ligações consideradas "pontes" - ligações cuja remoção divide uma rede conexa em duas subredes, i.e. conectam diferentes comunidades. O algoritmo ao longo de sucessivas iterações vai identificando e removendo as ligações com maior centralidade de intermediação, ou seja, as ligações presentes no maior número de caminhos mais curtos entre pares de nodos. Ao remover as pontes, o algoritmo revela as divisões mais significativas na rede, permitindo a formação de comunidades bem definidas. O foco está em melhorar a coesão interna das comunidades, ao mesmo tempo que se minimiza a separação entre elas.

Na rede g, a aplicação do método à sua componente gigante permitiu a identificação de comunidades fortes internamente, mas separadas por pontes. Estas pontes são as conexões restantes entre as diferentes comunidades, e a sua remoção possibilita uma separação mais clara entre os grupos.

Este método de deteção de comunidades foi aplicado na rede g, e foram obtidas as comunidades da seguinte figura 3.

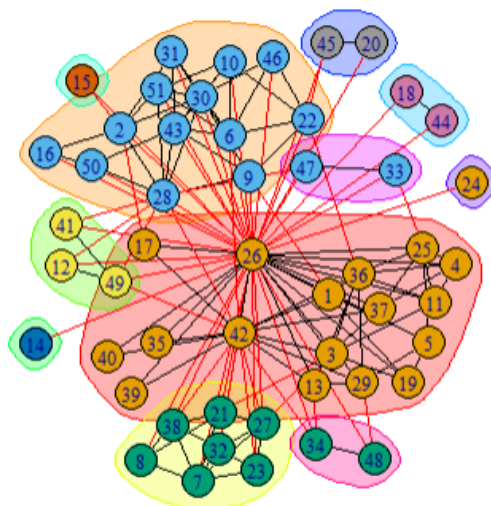


Figura 3: rede g aplicando o método da Remoção de Pontes

Na próxima tabela, encontram-se as características solicitadas nas alíneas i, ii, iii, e iv.

Comunidade	Dimensão (nº nodos na Comunidade)	Nº Ligações Internas	Densidade Interna	Nº Ligações Externas
1	17	47	0.35 ~	50
2	13	32	0.41 ~	24
3	7	14	0.67 ~	13
4	3	2	0.67 ~	7
5	1	0	NA	1
6	1	0	NA	2
7	2	1	1	2
8	2	1	1	3
9	1	0	NA	1
10	2	1	1	4
11	2	1	1	5

Tabela 2: Tabela referente aos resultados obtidos nas alíneas i., ii, iii, iv. para o Algoritmo Remoção de Pontes

Relativamente à tabela 2, podemos notar que existem 11 comunidades, com características distintas em termos de dimensão, número de ligações internas, densidade interna e número de ligações externas.

Destaca-se a comunidade 1, que é a maior em termos de dimensão, com 17 nodos, isto é, 17 indivíduos. No entanto, é também aquela onde as pessoas estabelecem menos contacto entre si, apresentando a densidade interna mais baixa ($d_{c1}^{int} \simeq 0.35$), sendo por isso a menos coesa, apesar de ter o maior número de ligações internas entre essas mesmas pessoas ($L_{c1} = 47$).

A comunidade 2, composta por 13 pessoas, destaca-se por ter uma densidade interna ligeiramente superior ($d_{c2}^{int} \simeq 0.41$) com 32 ligações internas ($L_{c2} = 32$), refletindo uma maior coesão entre os seus membros (estes têm mais contacto entre si do que os da comunidade 1). Relativamente às ligações externas, apresenta 24, o que evidencia uma separação moderada das pessoas desta comunidade com as de outras comunidades ($24 < L_{c2} = 32$).

As comunidades 3 e 4 são mais pequenas, com 7 e 3 pessoas, respetivamente, mas apresentam uma densidade interna mais alta ($d_{c3}^{int} = d_{c4}^{int} \simeq 0.67$). Isto indica uma elevada coesão entre os indivíduos destas comunidades. No entanto, ao contrário da comunidade 3 que tem menos uma ligação externa do que ligações internas ($13 < L_{c3} = 14$), a comunidade 4 tem menos ligações internas ($L_{c4} = 2$) e um número relativamente alto de ligações externas (7), o que demonstra uma pior separação desta em relação às restantes comunidades.

As comunidades 5, 6 e 9 são compostas por apenas 1 pessoa, como tal não apresentam quaisquer ligações internas (sem coesão), e dispõem muito poucas ligações externas (má separação - existem ligações externas para nenhuma ligação interna). Estas são comunidades praticamente isoladas.

Por outro lado, as comunidades 7, 8, 10 e 11 são pequenos grupos de 2 pessoas, o que implica uma ligação interna em cada uma, e por isso apresentam a maior e máxima densidade interna ($d_{c7}^{int} = d_{c8}^{int} = d_{c9}^{int} = d_{c10}^{int} = 1$), sendo as com maior coesão. O número de ligações externas varia entre 2 e 5, e como ultrapassa o número de ligações internas (1), estas comunidades não estão bem separadas das outras.

Relativamente à coesão no geral, as comunidades 7, 8, 10 e 11 são as mais coesas, sendo as que apresentam densidade máxima, isto é, o maior número de contactos entre pessoas da comunidade face ao número total possível. Seguindo-se as comunidades 3 e 4, depois a comunidade 2, e a comunidade 1 com coesão mais baixa. As comunidades 5, 6 e 9 de apenas 1 indivíduo, não têm ligações internas, logo não apresentam coesão.

No que remete à separação, a comunidade 2 é a mais bem separada das restantes devido ao facto de o número de ligações externas ser o mais inferior relativamente ao de ligações internas, e a seguir está a comunidade 3. Já, todas as restantes comunidades apresentam uma pior e má separação, uma vez que o seu número de ligações externas ultrapassa o de internas.

Finalmente, tendo em conta a medida de modularidade, o valor obtido foi relativamente baixo: $Q \simeq 0.346$. Isto indica que as comunidades identificadas podem ter resultado de um processo aleatório e que as partições não refletem uma estrutura significativamente distinta da que seria esperada numa rede puramente aleatória.

Q1.2 Algoritmo Cluster Fast Greedy

Este método de detecção de comunidades consiste num algoritmo de *clustering* hierárquico aglomerativo que permite privilegiar a obtenção de comunidades coesas, maximizando a modularidade.

O algoritmo começa com *clusters* de apenas 1 nodo, pelo que inicialmente a modularidade é negativa. A cada iteração, os *clusters* são agrupados de maneira a maximizar a modularidade entre eles. Dessa forma, a solução ótima deste algoritmo é aquela cuja modularidade tem um valor mais alto. No entanto, como apenas se foca em maximizar a modularidade, pode ficar facilmente preso numa solução com uma modularidade que não é a mais alta possível.

Ao aplicar este método de detecção de comunidades na rede g, foram obtidas as comunidades da componente gigante da figura 4:

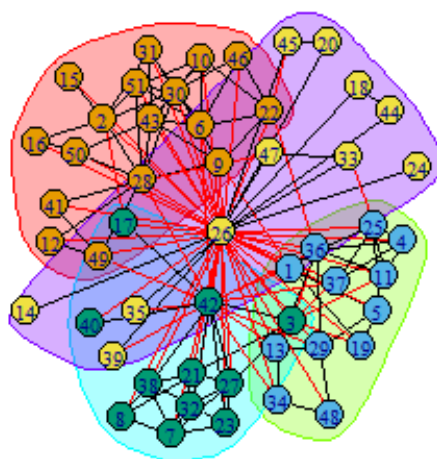


Figura 4: rede g aplicando o método de Clustering Fast Greedy.

Na tabela seguinte estão representadas as características pedidas nas alíneas i, ii, iii, e iv.

Comunidade	Dimensão (nº nodos na comunidade)	Nº Ligações Internas	Densidade Interna	Nº Ligações Externas
1	17	38	0.27 ~	25
2	11	23	0.41 ~	27
3	12	19	0.28 ~	25
4	11	13	0.23 ~	45

Tabela 3: Tabela referente aos resultados obtidos nas alíneas i, ii, iii, iv. para o algoritmo Fast Greedy.

Relativamente à tabela 3, podemos concluir que existem 4 comunidades, todas com uma distribuição relativamente equitativa dos nodos. Destaca-se a comunidade 1, que tem um maior número de pessoas (17 nodos), mas é aquela onde as pessoas conhecem-se menos umas às outras (tem uma menor densidade interna).

As comunidades 2, 3 e 4 têm quase o mesmo número de pessoas na comunidade (11, 12 e 11, respetivamente), no entanto, na comunidade 2 observa-se que as pessoas têm um maior contacto presencial entre elas face às comunidades 3 e 4 (tem uma densidade interna mais elevada: $d_{c2}^{int} \simeq 0.41$).

Relativamente à coesão, a comunidade 2 é aquela que tem uma maior coesão, ou seja, é aquela que se verifica um maior número de ligações internas entre os nodos, face ao número de ligações total possível nessa comunidade (tem uma maior densidade interna). No contexto do problema, significa que na comunidade 2 é onde existe uma maior conexão entre todas as pessoas (as pessoas dentro dessa comunidade conhecem-se relativamente melhor umas às outras). O inverso verifica-se na comunidade 4.

Quanto à separação, a comunidade 1 é a comunidade mais bem separada das restantes porque é a única que tem o número de ligações externas menor que o número de ligações internas. Significa que na comunidade 1 é aquela em que as pessoas têm um contacto maior entre elas do que com as restantes pessoas fora desse círculo.

A comunidade 4 é novamente aquela que tem uma pior separação das restantes comunidades, porque tem das menores ligações internas na rede ($d_{c4}^{int} \simeq 13$), e é a que tem o maior número de ligações externas (45).

Por fim, em relação à medida estudada para avaliar as partições (modularidade), obteve-se um valor de $Q \simeq 0.346$, o que é um valor relativamente baixo. Significa que é mais provável que as comunidades formadas tenham resultado de um processo aleatório, ou seja, as comunidades foram formadas mas não existe um significado específico para o qual foram criadas, apenas se verifica um conjunto de partições com um maior valor de modularidade.

Q1.3 Algoritmo de Louvain

O método de Louvain remete um algoritmo aglomerativo, bastante utilizado por ser rápido, e tal como o Cluster Fast Greedy visa a otimização da modularidade. Embora se trate de um método greedy que costuma gerar partições afastadas da ótima, e dependa

da ordem pela qual os nodos são visitados, podendo por isso retornar soluções diferentes a cada vez que é implementado numa mesma rede.

O algoritmo inicia com cada nodo como a sua própria comunidade (modularidade negativa), e otimiza a modularidade em duas etapas: (1ª) os nodos são atribuídos às comunidades vizinhas que maximizam a modularidade; (2ª) as comunidades formadas são reduzidas a supernodos com pesos que representam as suas ligações internas e externas. O processo termina após a repetição das duas etapas, i.e. até a modularidade estabilizar.

Implementando o algoritmo na rede ego g, foi obtida a partição presente na figura 5.

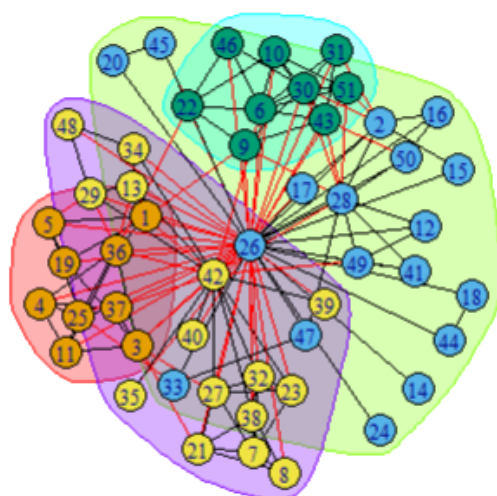


Figura 5: Rede g após aplicação do Algoritmo de Louvain

Na seguinte tabela, encontra-se para cada comunidade as características pedidas nas alíneas i., ii, iii, e iv.

Comunidade	Dimensão (nº nodos na comunidade)	Nº Ligações Internas	Densidade Interna	Nº Ligações Externas
1	9	17	0.47 ~	21
2	18	32	0.21 ~	46
3	9	20	0.56 ~	21
4	15	28	0.27 ~	28

Tabela 4: Tabela referente aos resultados obtidos nas alíneas i., ii, iii, iv. para o algoritmo de Louvain

Analisando a tabela 4, retira-se que no total foram detetadas 4 comunidades, sendo a comunidade 2 a com maior número de nodos (18 nodos), e esta tem apenas mais três nodos que a comunidade 4 (15 nodos). Com menor e igual número de nodos estão as comunidades 1 e 3 (9 nodos). Ou seja, o maior número de indivíduos na rede pertencem

às comunidades 2 e 4, e as comunidades 1 e 3 apresentam ambas metade do número de pessoas da comunidade com mais gente.

Relativamente à coesão e separação das comunidades na partição obtida:

- **Comunidade 1:** comunidade que apresenta menos ligações internas ($L_{c1} = 17$), isto é, aquela em que existem menos contactos entre pessoas dentro desta face às restantes. No entanto, a densidade interna é moderada ($d_{c1}^{int} \simeq 0.47$), revelando que dentro da comunidade muitos indivíduos estabelecem contacto entre si, pelo que a comunidade 1 tem uma coesão significativa. É também uma das comunidades que dispõe de menos ligações externas, significando que dentro desta as pessoas contactam menos com as de fora. Contudo, este número de contactos excede os contactos intrínsecos à comunidade ($21 > L_{c1} = 17$), pelo que esta comunidade está mal separada das restantes.
- **Comunidade 2:** comunidade com mais ligações internas ($L_{c2} = 32$), e por outro lado com a menor densidade interna ($d_{c2}^{int} \simeq 0.21$), significando que apesar de ser a comunidade em que mais pessoas se conhecem em comparação com as outras, é a que apresenta menos conexões entre pessoas dentro da comunidade, ou seja, uma coesão mais fraca. A elevada quantidade de ligações externas indica que esta comunidade se encontra bastante conectada a outras, e ultrapassa por muito as ligações dentro da própria comunidade ($46 > L_{c2} = 32$), sendo a que está pior separada na rede.
- **Comunidade 3:** comunidade com número de ligações internas ligeiramente superior ao da comunidade 1 ($L_{c3} = 20$), e com a maior densidade interna ($d_{c3}^{int} \simeq 0.56$), pelo que esta é a comunidade em que mais indivíduos se conhecem entre si, e por isso é a que tem maior coesão. Além disso, a comunidade 3 apresenta o menor número de ligações externas (igual ao da comunidade 1), significando que as pessoas desta comunidade contactam menos com as de outras, embora, ainda assim, exceda por um as conexões na comunidade em si ($21 > L_{c3} = 20$), estando de igual modo mal separada das outras comunidades.
- **Comunidade 4:** comunidade que contém o mesmo número de ligações internas e externas (28) e uma densidade interna quase moderada ($d_{c4}^{int} \simeq 0.27$). Quanto ao número de ligações internas entre as pessoas este é o segundo valor mais alto, contudo a densidade interna é mais baixa, então as pessoas mantêm um contacto relativo umas com as outras na comunidade, e por isso a coesão é moderada. Comparativamente às restantes comunidades, a 4 tem um número de

ligações externas intermédio, e o número de contactos dentro da comunidade é igual ao número de contactos com pessoas de outras comunidades ($28 = L_{c3} = 28$), logo a comunidade 4 não é bem separada das restantes, apesar de ter o melhor valor de separação da rede.

Resumidamente, as comunidades mais coesas são a 1 e 3, as quais apesar de terem menor número de indivíduos, são aquelas em que dentro da comunidade há mais pessoas que estabelecem contacto presencial umas com as outras. As comunidades 2 e 4 são as menos coesas, sendo aquelas onde há menos, mas ainda assim um número significativo de pessoas que se conhecem dentro da mesma comunidade. Em termos de separação, nenhuma das comunidades apresenta um número inferior de ligações externas face a ligações internas, o que reflete uma má separação em todas as comunidades, significando que todas as comunidades têm maior contacto com pessoas fora do que dentro destas. Ainda assim, as comunidades que estão mais próximas de ser bem separadas são a 3 e a 4, e mais longe encontra-se a 2.

Concluindo, as comunidades estão mal separadas, remetendo que esta partição tem má separação. Por outro lado, a média da soma das densidades internas das comunidades é aproximadamente 0.4, o que indica que, no geral, à partida a partição apresenta coesão moderada.

Para avaliar se a partição gerada é de facto boa ou não, foi calculada a modularidade da rede, e o valor obtido foi de $Q \approx 0.356$. Este valor indica uma modularidade baixa, ou seja, é mais provável que as comunidades formadas tenham resultado de um processo aleatório.

Q1.4 Método da Propagação de Etiquetas

O Método da Propagação de Etiquetas é um algoritmo simples e eficiente de deteção de comunidades, sendo especialmente útil em redes sociais.

O processo inicia-se atribuindo a cada nodo uma etiqueta única, neste caso cada pessoa forma a sua própria comunidade. A cada iteração do algoritmo, cada pessoa (nodo) fica com a etiqueta mais frequente entre os indivíduos com quem interage diretamente (nodos vizinhos). O algoritmo termina quando todas as pessoas (nodos) mantêm o mesmo rótulo, não ocorrendo nenhuma alteração.

As vantagens deste método são a sua rapidez, o não necessitar de parâmetros iniciais nem de cálculos complexos, e a fácil implementação. No entanto, apresenta algumas limitações, como o resultado depender da ordem de atualização dos nodos, e não ser adequado para redes onde um nodo possa pertencer a mais do que uma comunidade.

Com o objetivo de observar as diferentes comunidades detetadas, este método foi aplicado três vezes na rede g, primeiro para a semente 1 (tal como nos métodos anteriores), e seguidamente para as sementes 10 e 23. As figuras 6, 7 e 8 ilustram as comunidades obtidas em cada uma das implementações.

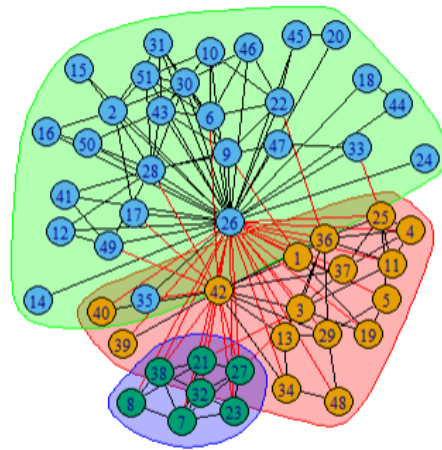


Figura 6: rede g aplicando o Método da Propagação de Etiquetas para a semente 1

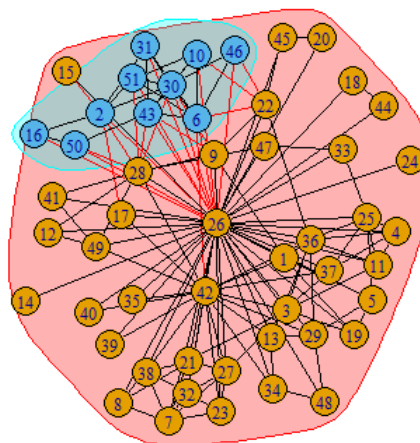


Figura 7: rede g aplicando o Método da Propagação de Etiquetas para a semente 10

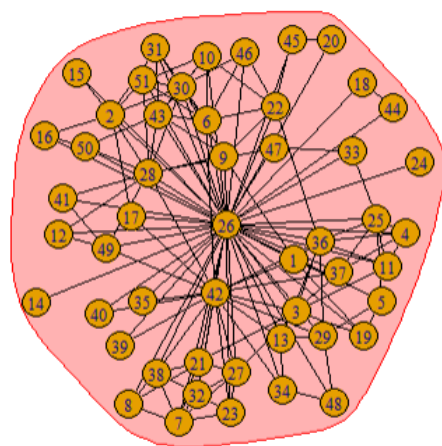


Figura 8: rede g aplicando o Método da Propagação de Etiquetas para semente 23

Na tabela seguinte estão representados as características pedidas nas alíneas i, ii, iii, e iv.

Semente	Comunidade	Dimensão (nº nodos na comunidade)	Nº Ligações Internas	Densidade Interna	Nº Ligações Externas
1	1	16	33	0.28 ~	30
	2	28	71	0.19 ~	31
	3	7	14	0.67 ~	13
10	1	41	112	0.14 ~	23
	2	10	20	0.44 ~	23
23	1	51	155	0.12 ~	0

Tabela 5: Tabela referente aos resultados obtidos nas alíneas i, ii, iii, iv. para o Método da propagação de etiquetas

Relativamente à **semente 1**, foram identificados três grupos de pessoas, respetivamente com 16, 28 e 7 membros. Deteta-se que o grupo de pessoas que estabelece maior contacto presencial entre si corresponde ao mais pequeno, a comunidade 3, que tem uma densidade interna significativamente elevada comparativamente às restantes ($d_{c3}^{int} \simeq 0.67$), refletindo uma comunidade mais coesa e isolada das outras.

Segue-se a comunidade com um grupo intermédio de pessoas, a comunidade 1, onde, para um aumento do número de pessoas no grupo em relação ao da comunidade anterior, nota-se uma diminuição substancial dos contactos presenciais que as pessoas desta comunidade mantêm entre si, pois a densidade interna desta comunidade decresceu ($d_{c1}^{int} = 0.28$).

Em contraste com os restantes grupos, o com maior número de pessoas é aquele onde se observa um menor contacto presencial entre estas, isto é, a comunidade 2, a qual tem a menor densidade interna ($d_{c2}^{int} \simeq 0.19$). Apesar de ser a comunidade que tem maior número de ligações internas ($L_{c2} = 71$), esse número encontra-se afastado do máximo de ligações que podiam existir entre as pessoas da comunidade ($L_{max}(c2) = 378$).

Para cada grupo de pessoas nesta partição, também se observa um maior contacto presencial dentro dos grupos do que destes para fora, mas ligeiramente. Ou seja, verifica-se em todas as comunidades, um maior número de ligações internas face às ligações externas, no entanto, os números não diferem muito, há exceção da comunidade 2, que apresenta a maior diferença e por isso é a que se encontra melhor separada das restantes comunidades.

Com a **semente 10**, foram apenas identificados 2 grupos de pessoas, respetivamente com 41 e 10 pessoas. Novamente, o menor grupo de pessoas é aquele onde as mesmas estabelecem um maior contacto e que corresponde à comunidade 2, sendo a que tem a maior densidade interna ($d_{c2}^{int} \simeq 0.44$). Verifica-se o contrário no restante grupo de pessoas, a comunidade 1, sendo o maior grupo e também aquele que tem o menor contacto presencial entre os seus membros - densidade interna ($d_{c1}^{int} \simeq 0.14$).

Nesta partição também se verifica que ambos os grupos têm um número de contactos presenciais fora do grupo idêntico, pois o número de ligações externas é igual (23). Porém, o grupo com maior número de pessoas (comunidade 1) apresenta um maior número de contactos com membros do grupo do que com pessoas de fora (nesta comunidade o número de ligações internas é bastante superior ao número de ligações externas: $23 < L_{c1} = 112$), pelo que a comunidade 1 está bem separada da comunidade 2. No outro grupo, conclui-se o contrário, a comunidade 2 está mal separada da 1, uma vez que na comunidade 2 o número de ligações externas é superior ao número de ligações internas ($23 > L_{c2} = 20$).

Para a **semente 23**, foi identificado apenas 1 grupo de pessoas, ou seja, uma comunidade com todas as 51 pessoas da rede. Esta comunidade apresenta todas as ligações originais entre indivíduos na rede (número de ligações internas: $L_{c1} = 155$). Todavia, tal como analisado inicialmente na rede antes de aplicar qualquer método de deteção de comunidades, estas pessoas têm pouco contacto presencial face ao contacto máximo que poderia existir (com exceção da pessoa que manteve contacto com todas as outras - nodo 26). Deteta-se assim uma densidade interna baixa equivalente à densidade da rede ($d_{c1}^{int} = d \simeq 0.12$), evidenciando pouca coesão entre as pessoas. Como só foi identificada uma comunidade, significa que o método não detetou nenhum grupo significativo de pessoas que se destacasse, por isso classificou todos os nodos com a mesma etiqueta, não existindo assim quaisquer ligações externas, logo a comunidade não possui separação. O facto de este método ter iterado de forma a haver apenas uma comunidade mostra-nos como esta rede pode ser, por si só, uma comunidade dentro de uma rede bastante maior e mais complexa.

Resumidamente, no que diz respeito à coesão, a comunidade 3 da semente 1 e a comunidade 2 da semente 10 são as mais coesas, com densidades internas de 0.67 e 0.44, respetivamente, ou seja, estes grupos de pessoas apresentam uma ligação mais forte entre os seus membros em comparação com os outros. Por outro lado, a única comunidade da semente 23 é a menos coesa, com a densidade interna mais baixa (0.12), devido ao facto de conter todas as pessoas num único grupo.

Quanto à separação, a comunidade da semente 23 não apresenta separação (pois não possui ligações externas) e a comunidade 2 da semente 10 apresenta ligeiramente mais

ligações externas do que internas, o que revela pior separação. Todas as restantes comunidades obtidas com as sementes 1 e 10, apresentam menos ligações externas do que ligações internas, o que já demonstra uma boa separação das pessoas dentro destas comunidades com as de fora. Sendo a comunidade melhor separada a 1 da semente 10, seguida pela 2 da semente 1.

Por fim, quanto à modularidade, a semente 1 obteve o maior valor ($Q \approx 0.336$), seguido da semente 10 com uma modularidade de aproximadamente $Q \approx 0.175$, e por último a semente 23 com uma modularidade $Q = 0$. Das três sementes, conclui-se que a 1 é aquela que obteve uma melhor partição. Era esperado que a modularidade da semente 23 fosse 0, porque todos os nodos pertencem à mesma comunidade, simbolizando uma concentração total dos indivíduos na rede. Apesar da semente 1 ter a maior modularidade obtida, este valor é baixo, significando que é mais provável que a partição obtida tenha resultado de um processo aleatório, sendo essa aleatoriedade ainda mais observada nas restantes sementes (obtiveram valores de modularidade menores).

Q2. Comparação e comentário dos resultados obtidos com os quatro métodos

A seguinte tabela apresenta para cada um dos métodos de deteção de comunidades implementado, o número de comunidades obtidas e a modularidade da partição:

Método de Deteção	Nº de Comunidades	Modularidade (Q)
Remoção de Pontes	11	0.322
Cluster Fast Greedy	4	0.346
Louvain	4	0.356
Propagação de Etiquetas *semente 1	3	0.336

Tabela 6: Tabela que compara todos os métodos de deteção de comunidades.

Através da tabela 6, é possível observar um contraste do número de comunidades detetadas entre os métodos: o Método de Remoção de Pontes registou o maior número de comunidades (11), indicando que na rede analisada o algoritmo identificou 11 subredes fortemente conexas. É importante referir que dessas 11, 3 são comunidades de

1 nodo e 4 têm 2 nodos, ou seja, 7 das 11 comunidades são apenas nodos isolados ou conectados entre eles, e as restantes 4 detêm os restantes nodos.

Os algoritmos Cluster Fast Greedy e Louvain identificaram ambos 4 comunidades, e por último, o método da Propagação de Etiquetas detetou apenas 3 comunidades.

No que toca à modularidade, observa-se, curiosamente, que os valores são bastante próximos uns dos outros: o melhor valor da modularidade verifica-se no algoritmo de Louvain (0.356), seguido do Algoritmo Cluster Fast Greedy e Método de Propagação de Etiquetas (0.346 e 0.336, respetivamente). Por fim, o valor mais baixo observa-se no Método da Remoção de Pontes (0.322). Face a esta proximidade entre os valores, assume-se que, independentemente do número de comunidades encontradas em cada método, as partições determinadas tenham resultado provavelmente de um processo aleatório, ou seja, as comunidades detetadas em cada método apenas contribuem para o valor da modularidade, sem significado adicional no contexto da rede. Dessa maneira não foi possível encontrar nenhuma comunidade recorrente nos 4 métodos.

Conclui-se que o Algoritmo de Louvain é o melhor método aplicado, visto ser aquele cuja deteção de comunidades obteve o melhor valor de modularidade, e por isso aquele onde a rede apresenta a melhor divisão em comunidades.

Comentários Adicionais

- ★ Para o Método de Louvain, numa tentativa de obter um valor mais elevado de modularidade, foram testadas várias sementes e aquela em que o grafo resultante obteve maior valor de modularidade foi com a *seed* 15: $Q \simeq 0.367$, o que continua a ser um valor baixo de modularidade;
- ★ Para complementar o estudo, também foi implementado o algoritmo Cluster Optimal no R, que permite detetar as comunidades correspondentes ao valor ótimo de modularidade na rede. Observou-se uma modularidade ligeiramente mais elevada que nos outros algoritmos aplicados ($Q \simeq 0.369$), embora ainda seja considerada baixa. Pode-se concluir que devido à estrutura da rede, não é possível obter um valor de modularidade acima deste, demonstrando que é difícil a deteção em comunidades nesta rede.

Conclusão

Relativamente ao número de comunidades detetadas, observou-se um grande contraste entre os métodos, por exemplo, enquanto que o Método de Remoção de Pontes detetou 11 comunidades diferentes, houve uma semente do Método de Propagação de Etiquetas que detetou apenas 1 comunidade (com todos os nodos).

Face à modularidade, o Algoritmo de Louvain é o melhor método aplicado ($Q \approx 0.356$), retornando a melhor divisão de comunidades dentro da rede social. Contudo, nem sempre o melhor valor de modularidade (mais elevado) remete a melhor solução possível.

No geral, os valores de modularidade obtidos em todos os métodos situam-se ligeiramente acima de 0.3, o que indica uma partição que é apenas moderadamente eficaz a identificar comunidades coesas e bem separadas dentro da rede. Embora valores de modularidade mais elevados sejam, em geral, interpretados como uma melhor separação das comunidades, é essencial considerar o contexto da rede estudada para compreender porque razão os valores são semelhantes entre os diferentes métodos.

Sendo esta rede uma subrede ego, a estrutura está naturalmente mais interligada devido à centralidade de uma pessoa (nodo 26). Esta pessoa é um *hub*, que conecta a maioria das comunidades entre si. Como resultado, as ligações externas são inevitáveis e frequentes, o que condiciona a separação exata das comunidades. Mesmo que os valores de modularidade sejam semelhantes, isto reflete mais a estrutura da rede do que a qualidade específica dos algoritmos aplicados.

Adicionalmente, o número reduzido de comunidades em alguns casos, especialmente para a semente 23 do Método de Propagação de Etiquetas, com apenas uma comunidade, reforça a dificuldade em identificar partições claras. Neste contexto, valores de modularidade próximos sugerem que a estrutura densamente conectada da rede influencia fortemente os resultados, independentemente do método utilizado.

Por último, embora valores acima de 0.3 possam ser considerados "aceitáveis" em teoria, os valores de modularidade para redes reais tendem a situar-se entre 0.3 e 0.7 [1]. Dessa forma, a estrutura da rede diminui a separação entre comunidades, tornando os resultados mais homogêneos. Por isso, o recomendado seria ter outras medidas/critérios em consideração ao avaliar uma partição, que permitissem a obtenção de resultados mais conclusivos e claros das partições no contexto da rede.

Referências Bibliográficas

[1] Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). *Fast unfolding of communities in large networks*. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), P10008.

<https://doi.org/10.1088/1742-5468/2008/10/P10008>

Menczer, F., Fortunato, S., & Davis, C. A. (2020). *A first course in network science*. Cambridge University Press.

Apêndice

Importação das Bibliotecas necessárias

```
library(igraph)
```

Importação e Visualização da Rede (contida no ficheiro "Ligações T2.txt")

```
set.seed(1) #para reprodutibilidade

#Definir o grafo "g" da rede importada através do ficheiro .txt com as
#ligações
g <- read_graph('Ligações T2.txt', format=c('edgelist'), directed=F)

#Mostrar o grafo
plot(g, vertex.color = "lightgreen", vertex.label.color = "black",
     vertex.label.cex=0.9, vertex.size = 18)#com caracterização dos nodos
plot(g, vertex.size=7,vertex.label=NA, vertex.color = "lightgreen") #sem
#caracterização dos nodos
```

Análise inicial das características da Rede

```
cat('Nodos: N =',vcount(g),'\n') #dimensão/nº total de nodos

cat('Ligações: L =',ecount(g),'\n') #número total de ligações

cat('Densidade: d =',edge_density(g),'\n') #densidade

cat('Grau médio: <k> =',mean(degree(g)),'\n') #grau médio

cat('Grau mínimo: k_min =',min(degree(g)),'\n') #grau mínimo

cat('Grau máximo: k_max =',max(degree(g)),'\n') #grau máximo
```

```
cat('Distância média: <l> =',mean_distance(g),'\n') #distância média
cat('Diâmetro: l_max =',max(distances(g)),'\n') #diâmetro
cat('Coeficiente de Clustering: C =',transitivity(g),'\n') #coeficiente
de clustering
cat('Nº de cliques:',length(cliques(g)), '\n') #número total de cliques
```

```
#Grau de cada nodo
deg <- degree(g,mode='total')
cat('Grau de cada nodo: ', deg,'\n\n')

#Parâmetro de heterogeneidade
ht <-mean(deg*deg)/mean(deg)^2
cat('Heterogeneidade (de grau): ',ht,'\n\n')

#Distribuição das frequências de grau
cat('Distribuição das frequências de cada grau:')
dg <- table(degree(g,mode='total'))
dg
hist(deg,breaks=0:max(deg), col='lightblue', #representação num
histograma
      main='Histograma do Grau dos Nodos', xlab='grau', ylab='frequência')
```

Secção Q1.

Aplicação de quatro métodos de detecção de comunidades estudados, e para cada um indicadas as seguintes características: **i.** Número de comunidades obtidas; **ii.** Dimensão de cada comunidade; **iii.** Número de ligações internas de cada comunidade e a respectiva densidade interna; **iv.** Número de ligações externas de cada comunidade; **v.** Análise da partição obtida quanto à coesão e a separação; **vi.** Utilização da medida estudada para avaliar as partições.

1. Algoritmo de Remoção de Pontes por Girvan e Newman

```
# Semente
set.seed(1)

# Aplicação do método à rede:
ceb <- cluster_edge_betweenness(g)

# Observar a rede:
plot(ceb, g, vertex.size=15, vertex.label.cex=0.6) #com legendas nos
nodos
plot(ceb, g, vertex.size=7, vertex.label=NA) #sem legendas nos nodos

#Determinar o nº de comunidades e respetiva dimensão, nº de ligações
internas e externas, densidade interna, e modularidade da partição
(alíneas i,ii,iii,iv e vi))
```

```

cat("*Detalhes das Comunidades*\n\n")

cn <- length(communities(ceb)) #número de comunidades
ft <- table(membership(ceb)) # dimensão de cada comunidade
cat(sprintf("N° de comunidades: %d\n\n", cn))
cat("Dimensões das comunidades: ")
ft

memb <- membership(ceb) # vetor com cada comunidade
communities <- unique(memb) # vetor com as comunidades

for (i in communities) {
  # Nodos da comunidade
  nodes <- which(memb == i)
  subgraph <- induced_subgraph(g, nodes)

  # Dimensão da comunidade
  size <- length(nodes)

  # Ligações internas e densidade interna
  internal_links <- ecount(subgraph)
  internal_density <- edge_density(subgraph)

  # Ligações externas
  external_links <- sum(degree(g, nodes)) - 2 * internal_links

  # Exibição
  cat(sprintf("\n Comunidade %d:\n", i))
  cat(sprintf("  - Dimensão: %d nodos\n", size))
  cat(sprintf("  - Ligações internas: %d\n", internal_links))
  cat(sprintf("  - Densidade interna: %.4f\n", internal_density))
  cat(sprintf("  - Ligações externas: %d\n\n", external_links))
}

# Modularidade:
Q <- modularity(ceb)
cat("Modularidade: Q =", Q)

```

2. Algoritmo Cluster Fast Greedy

```

# Fazer o set da seed
set.seed(1)

# Aplicar o algoritmo:
al_FG <- cluster_fast_greedy(g)

# Observar as comunidades
plot(al_FG, g, vertex.size=15, vertex.label.cex=0.6) #com legendas nos
nodos
plot(al_FG, g, vertex.size=7, vertex.label=NA) #sem legendas nos nodos

#Determinar o n° de comunidades e respetiva dimensão, n° de ligações
internas e externas, densidade interna, e modularidade da partição
(alíneas i,ii,iii,iv e vi))

```

```

cat("*Detalhes das Comunidades* \n\n")

cn_fg <- length(communities(al_FG)) #número de comunidades
ft_fg <- table(membership(al_FG)) # tabela da dimensão de cada comunidade
cat(sprintf("N° de comunidades: %d\n\n", cn_fg))
cat("Dimensões das comunidades: ")
ft_fg

membFG <- membership(al_FG) # vetor com cada comunidade
communitiesFG <- sort(unique(membFG)) # vetor com as comunidades

for (i in communitiesFG) {
  # Nodos da comunidade
  nodes <- which(membFG == i)
  subgraph <- induced_subgraph(g, nodes)

  # Dimensão da comunidade
  size <- length(nodes)

  # Ligações internas e densidade interna
  internal_links <- ecount(subgraph)
  internal_density <- edge_density(subgraph)

  # Ligações externas
  external_links <- sum(degree(g, nodes)) - 2 * internal_links

  # Exibição
  cat(sprintf("\n Comunidade %d:\n", i))
  cat(sprintf("  - Dimensão: %d nodos\n", size))
  cat(sprintf("  - Ligações internas: %d\n", internal_links))
  cat(sprintf("  - Densidade interna: %.4f\n", internal_density))
  cat(sprintf("  - Ligações externas: %d\n\n", external_links))
}

# Modularidade:
Q <- modularity(al_FG)
cat("Modularidade: Q =", Q)

```

3. Algoritmo de Louvain

```

#Definir semente
set.seed(1)

#Aplicação do método à rede
gl <- cluster_louvain(g)

#Visualizar o grafo com as comunidades
plot(gl, g, vertex.size=15, vertex.label.cex=0.6) #com legendas nos nodos
plot(gl, g, vertex.size=7, vertex.label=NA) #sem legendas nos nodos

#Determinar o n° de comunidades e respetiva dimensão, n° de ligações
internas e externas, densidade interna, e modularidade da partição
(alíneas i,ii,iii,iv e vi))

```



```

cat("*Detalhes das Comunidades*\n\n")

nc <- length(communities(gl)) #número de comunidades
tf <- table(membership(gl)) #dimensão de cada comunidade
cat(sprintf("N° de comunidades: %d\n\n", nc))
cat("Distribuição de frequências das dimensões das comunidades:")
tf

memb_gl <- membership(gl) # vetor com cada comunidade
communities_gl <- unique(memb_gl) # vetor com as comunidades

for (i in communities_gl) {
  # Nodos da comunidade
  nodes <- which(memb_gl == i)
  subgraph <- induced_subgraph(g, nodes)

  # Dimensão da comunidade
  size <- length(nodes)

  # Ligações internas e densidade interna
  internal_links <- ecount(subgraph)
  internal_density <- edge_density(subgraph)

  # Ligações externas
  external_links <- sum(degree(g, nodes)) - 2 * internal_links

  # Exibição
  cat(sprintf("\n Comunidade %d:\n", i))
  cat(sprintf("  - Dimensão: %d nodos\n", size))
  cat(sprintf("  - Ligações internas: %d\n", internal_links))
  cat(sprintf("  - Densidade interna: %.4f\n", internal_density))
  cat(sprintf("  - Ligações externas: %d\n\n", external_links))
}

# Modularidade:
Q <- modularity(gl)
cat("Modularidade: Q =", Q)

```

4. Algoritmo de Propagação de Etiquetas

1ª implementação (*seed*=1):

```

#Definir semente=1
set.seed(1)

# Aplicar o método
clp <- cluster_label_prop(g)

# Vizualizar as comunidades detetadas
set.seed(1)
plot(clp, g, vertex.size=15, vertex.label.cex=0.6) #com legendas nos
nodos
plot(clp, g, vertex.size=7, vertex.label=NA) #sem legendas nos nodos

```

```

#Determinar o n° de comunidades e respetiva dimensão, n° de ligações
internas e externas, densidade interna, e modularidade da partição
(alíneas i,ii,iii,iv e vi))

cat("*Detalhes das Comunidades*\n\n")

nc_clp <- length(communities(clp)) #número de comunidades
tf_clp <- table(membership(clp)) #dimensão de cada comunidade
cat(sprintf("N° de comunidades: %d\n\n", nc_clp))
cat("Distribuição de frequências das dimensões das comunidades:")
tf_clp

memb_clp <- membership(clp) # vetor com cada comunidade
communities_clp <- unique(memb_clp) # vetor com as comunidades

for (i in communities_clp) {
  # Nodos da comunidade
  nodes <- which(memb_clp == i)
  subgraph <- induced_subgraph(g, nodes)

  # Dimensão da comunidade
  size <- length(nodes)

  # Ligações internas e densidade interna
  internal_links <- ecount(subgraph)
  internal_density <- edge_density(subgraph)

  # Ligações externas
  external_links <- sum(degree(g, nodes)) - 2 * internal_links

  # Exibição
  cat(sprintf("\n Comunidade %d:\n", i))
  cat(sprintf("  - Dimensão: %d nodos\n", size))
  cat(sprintf("  - Ligações internas: %d\n", internal_links))
  cat(sprintf("  - Densidade interna: %.4f\n", internal_density))
  cat(sprintf("  - Ligações externas: %d\n\n", external_links))
}

# Modularidade:
Q <- modularity(clp)
cat("Modularidade: Q =", Q)

```

2ª implementação (seed=10):

```

#Definir semente=10
set.seed(10)

# Aplicar o método
clp2 <- cluster_label_prop(g)

# Vizualizar a Rede
set.seed(1)
plot(clp2, g, vertex.size=15, vertex.label.cex=0.6) #com legendas nos
nodos

```

```

plot(clp2, g, vertex.size=7, vertex.label=NA) #sem legendas nos nodos

#Determinar o n° de comunidades e respetiva dimensão, n° de ligações
internas e externas, densidade interna, e modularidade da partição
(alíneas i,ii,iii,iv e vi))

cat("*Detalhes das Comunidades*\n\n")

nc_clp2 <- length(communities(clp2)) #número de comunidades
tf_clp2 <- table(membership(clp2)) #dimensão de cada comunidade
cat(sprintf("N° de comunidades: %d\n\n", nc_clp2))
cat("Distribuição de frequências das dimensões das comunidades:")
tf_clp2

memb_clp2 <- membership(clp2) # vetor com cada comunidade
communities_clp2 <- unique(memb_clp2) # vetor com as comunidades

for (i in communities_clp2) {
  # Nodos da comunidade
  nodes <- which(memb_clp2 == i)
  subgraph <- induced_subgraph(g, nodes)

  # Dimensão da comunidade
  size <- length(nodes)

  # Ligações internas e densidade interna
  internal_links <- ecount(subgraph)
  internal_density <- edge_density(subgraph)

  # Ligações externas
  external_links <- sum(degree(g, nodes)) - 2 * internal_links

  # Exibição
  cat(sprintf("\n Comunidade %d:\n", i))
  cat(sprintf("  - Dimensão: %d nodos\n", size))
  cat(sprintf("  - Ligações internas: %d\n", internal_links))
  cat(sprintf("  - Densidade interna: %.4f\n", internal_density))
  cat(sprintf("  - Ligações externas: %d\n\n", external_links))
}

# Modularidade:
Q <- modularity(clp2)
cat("Modularidade: Q =", Q)

```

3ª implementação (seed=23):

```

#Definir semente=23
set.seed(23)

# Aplicar o método
clp3 <- cluster_label_prop(g)

# Vizualizar a rede
set.seed(1)

```

```

plot(clp3, g, vertex.size=15, vertex.label.cex=0.6) #com legendas nos
nodos
plot(clp3, g, vertex.size=7, vertex.label=NA) #sem legendas nos nodos

#Determinar o n° de comunidades e respetiva dimensão, n° de ligações
internas e externas, densidade interna, e modularidade da partição
(alíneas i,ii,iii,iv e vi))

cat("*Detalhes das Comunidades*\n\n")

nc_clp3 <- length(communities(clp3)) #número de comunidades
tf_clp3 <- table(membership(clp3)) #dimensão de cada comunidade
cat(sprintf("N° de comunidades: %d\n\n", nc_clp3))
cat("Distribuição de frequências das dimensões das comunidades:")
tf_clp3

memb_clp3 <- membership(clp3) # vetor com cada comunidade
communities_clp3 <- unique(memb_clp3) # vetor com as comunidades

for (i in communities_clp3) {
  # Nodos da comunidade
  nodes <- which(memb_clp3 == i)
  subgraph <- induced_subgraph(g, nodes)

  # Dimensão da comunidade
  size <- length(nodes)

  # Ligações internas e densidade interna
  internal_links <- ecount(subgraph)
  internal_density <- edge_density(subgraph)

  # Ligações externas
  external_links <- sum(degree(g, nodes)) - 2 * internal_links

  # Exibição
  cat(sprintf("\n Comunidade %d:\n", i))
  cat(sprintf("  - Dimensão: %d nodos\n", size))
  cat(sprintf("  - Ligações internas: %d\n", internal_links))
  cat(sprintf("  - Densidade interna: %.4f\n", internal_density))
  cat(sprintf("  - Ligações externas: %d\n\n", external_links))
}

# Modularidade:
Q <- modularity(clp3)
cat("Modularidade: Q =", Q)

```

Adicional

Algoritmo de Louvain com melhor valor de modularidade (seed = 15)

```

#Definir semente
set.seed(15)

```

```

#Aplicação do método à rede
gl <- cluster_louvain(g)

#Visualizar o grafo com as comunidades
plot(gl, g, vertex.size=15, vertex.label.cex=0.6) #com legendas nos nodos
plot(gl, g, vertex.size=7, vertex.label=NA) #sem legendas nos nodos

#Determinar o n° de comunidades e respetiva dimensão, n° de ligações
internas e externas, densidade interna, e modularidade da partição
(alíneas i,ii,iii,iv e vi))

cat("*Detalhes das Comunidades*\n\n")

nc <- length(communities(gl)) #número de comunidades
tf <- table(membership(gl)) #dimensão de cada comunidade
cat(sprintf("N° de comunidades: %d\n\n", nc))
cat("Distribuição de frequências das dimensões das comunidades:")
tf

memb_gl <- membership(gl) # vetor com cada comunidade
communities_gl <- unique(memb_gl) # vetor com as comunidades

for (i in communities_gl) {
  # Nodos da comunidade
  nodes <- which(memb_gl == i)
  subgraph <- induced_subgraph(g, nodes)

  # Dimensão da comunidade
  size <- length(nodes)

  # Ligações internas e densidade interna
  internal_links <- ecount(subgraph)
  internal_density <- edge_density(subgraph)

  # Ligações externas
  external_links <- sum(degree(g, nodes)) - 2 * internal_links

  # Exibição
  cat(sprintf("\n Comunidade %d:\n", i))
  cat(sprintf("  - Dimensão: %d nodos\n", size))
  cat(sprintf("  - Ligações internas: %d\n", internal_links))
  cat(sprintf("  - Densidade interna: %.4f\n", internal_density))
  cat(sprintf("  - Ligações externas: %d\n\n", external_links))
}

# Modularidade:
Q <- modularity(gl)
cat("Modularidade: Q =", Q)

```

Algoritmo Cluster Optimal

```
#Definir semente
```

```

set.seed(1)

#Aplicação do método à rede
gco <- cluster_optimal(g)

#Visualizar o grafo com as comunidades
plot(gco, g, vertex.size=15, vertex.label.cex=0.6) #com legendas nos
nodos
plot(gco, g, vertex.size=7, vertex.label=NA) #sem legendas nos nodos

#Determinar o n° de comunidades e respetiva dimensão, n° de ligações
internas e externas, densidade interna, e modularidade da partição
(alíneas i,ii,iii,iv e vi))

cat("*Detalhes das Comunidades*\n\n")

nco <- length(communities(gco)) #número de comunidades
tfo <- table(membership(gco)) #dimensão de cada comunidade
cat(sprintf("N° de comunidades: %d\n\n", nco))
cat("Distribuição de frequências das dimensões das comunidades:")
tfo

memb_gco <- membership(gco) # vetor com cada comunidade
communities_gco <- unique(memb_gco) # vetor com as comunidades

for (i in communities_gco) {
  # Nodos da comunidade
  nodes <- which(memb_gco == i)
  subgraph <- induced_subgraph(g, nodes)

  # Dimensão da comunidade
  size <- length(nodes)

  # Ligações internas e densidade interna
  internal_links <- ecount(subgraph)
  internal_density <- edge_density(subgraph)

  # Ligações externas
  external_links <- sum(degree(g, nodes)) - 2 * internal_links

  # Exibição
  cat(sprintf("\n Comunidade %d:\n", i))
  cat(sprintf("  - Dimensão: %d nodos\n", size))
  cat(sprintf("  - Ligações internas: %d\n", internal_links))
  cat(sprintf("  - Densidade interna: %.4f\n", internal_density))
  cat(sprintf("  - Ligações externas: %d\n\n", external_links))
}

# Modularidade:
Q <- modularity(gco)
cat("Modularidade: Q =", Q)

```