Adalberto Shindy Hassobe, Bruno de Santana Braga Contreras, Lucas Mendes Sales, Luisa Dipierri Landert, Victor Ferreira Lopes

Detecção de Comunidades em Redes Complexas Algoritmos e Testes

São Paulo, SP

Junho, 2022

Adalberto Shindy Hassobe, Bruno de Santana Braga Contreras, Lucas Mendes Sales, Luisa Dipierri Landert, Victor Ferreira Lopes

Detecção de Comunidades em Redes Complexas
Algoritmos e Testes

Detecção de comunidades em redes complexas, algoritmos e testes. Trabalho realizado na disciplina de Introdução a Redes Complexas. Professor: Masayuki Oka Hase

Universidade de São Paulo (USP)

Escola de Artes Ciências e Humanidades (EACH)

São Paulo, SP

Junho, 2022

Resumo

Este trabalho tem o intuito de apresentar conceitos que envolvem o problema de detecção de comunidades. A princípio é analisado o que são comunidades, tendo essa base, exemplificamos abordagens na tentativa de fazer o agrupamento dessas comunidades, mas muitas delas são inviáveis na aplicação de redes mais complexas, por conta de sua performance. Para isso, também passamos para modelos estudados na literatura, como os algoritmos divisivos e aglomerativos. Ao final, fazemos testes de algoritmos (benchmark) entre dois modelos para a comparação dos resultados dos mesmos.

Palavras-chaves: 1. Redes Complexas. 2. Detecção de Comunidades. 3. Algoritmos divisivos e aglomerativos. 4. Testes de Algoritmos .

Sumário

Introdução

A detecção de comunidades em grafos é um tema bastante abordado na ciência moderna de redes. Comunidades, clusters, módulos ou agrupamentos, como são chamados, são estruturas nas redes complexas amplamente investigadas. Elas representam propriedades de uma rede, de modo que os vértices dentro da comunidade sejam densamente conectados e que também tenha grande probabilidade de possuir comportamentos ou funções semelhantes. Tendo isso em mente, a detecção de comunidade é o processo de determinar uma comunidade em uma rede. Na Figura 1, temos um exemplo de uma estrutura de comunidade, onde os vértices são os cientistas e eles estão dispostos de forma que cada cor representa um grupo distinto de pesquisa.

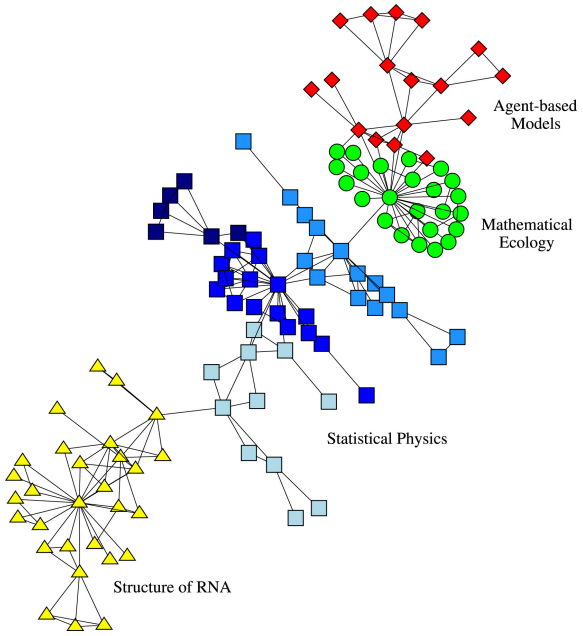


Figura 1 – Estrutura de comunidades na rede de colaboração de cientistas

Uma comunidade pode permitir que a cada nó do grafo esteja apenas contido no mesmo, ou também pode permitir que existam nós compartilhados com outras comunidades, neste caso denominadas comunidades sobrepostas, onde existe compartilhamento de vértices de fronteiras. Existem vários exemplos dessas organizações em grafos no mundo real, como no âmbito social, biológico, econômico, tecnológico (por exemplo na ciência da computação, Web), entre outros. Comunidades em si, também possuem exemplos concretos no mundo real, como em redes ad-hoc (PERKINS, 2001), balanceamento de carga da Web

(KRISHNAMURTHY; WANG, 2000), recomendações de produtos a clientes (REDDY et al., 2002).

Identificar uma comunidade, nos permite entender como a rede está organizada, focando apenas em algumas regiões com certo grau de autonomia. Isso ajuda na classificação dos vértices, este que pode ser distinguido em um que se encontra dentro do agrupamento, fora do agrupamento ou que fica na fronteira do agrupamento. A técnica de detecção de comunidades possibilita identificar os grupos apenas baseando-se em informações na topologia do grafo, sendo um problema estudado já a bastante tempo, onde os primeiros trabalhos ocorreram em torno de mais de 50 anos no passado.

Durante anos, vários cientistas estudaram o problema de detecção de comunidades, e criaram modelos/algoritmos em vista desses estudos. Girvan e Newman (GIRVAN; NEWMAN, 2002) apresentaram seu trabalho, com um novo algoritmo de detecção de comunidade, isso impulsionou as atividades na área, e novos métodos foram surgindo com o tempo.

A função de modularidade de Girvan e Newman (NEWMAN; GIRVAN, 2004) teve bastante destaque, pois resumia informações das estruturas de comunidades em uma função que se comparava a um grafo aleatório. Sendo assim, ela ganhou bastante popularidade e foi adotada como ferramenta de otimização para vários outros algoritmos que foram surgindo com o tempo, desde otimizações gulosas até as espectrais. Newman (NEWMAN, 2004), Clauset et al. (CLAUSET; NEWMAN; MOORE, 2004), Blondel et al. (BLONDEL et al., 2008), Shah & Zaman (PARTHASARATHY; SHAH; ZAMAN, 2010) são exemplos de autores que desenvolveram algoritmos tendo a modularidade como base para grafos grandes.

Newman (NEWMAN, 2004) propôs o primeiro algoritmo de detecção de comunidades em grafos grandes, sendo um algoritmo guloso que obtém um bom particionamento por meio do agrupamento de comunidades repetidas vezes. Clauset et al. (CLAUSET; NEWMAN; MOORE, 2004) reformulou o algoritmo apresentado por Newman, modificando as estruturas de dados do mesmo, a fim de poder ser utilizado em grafos ainda maiores como também obter otimização, o objetivo era deixar o algoritmo mais eficiente e melhorar a qualidade da modularidade.

Blondel et al. (BLONDEL et al., 2008) desenvolveu um outro algoritmo (conhecido como algoritmo de Louvain, por causa da universidade de Louvain) para detecção de comunidades, que também possui uma abordagem baseada em modularidade, além disso também é um algoritmo guloso para grandes redes. Ele é um algoritmo que tenta maximizar

a diferença entre o número real de uma aresta com o número esperado da aresta em uma comunidade de rede ponderada. No geral, é um algoritmo que apresenta valores de modularidade melhores que os de Newman e Clauset et al., sendo um algoritmo bastante eficiente, além disso é bastante popular pela facilidade de sua implementação e alta velocidade, seu ponto negativo é a limitação pelo uso da memória principal. Na Figura 2, temos uma tabela que compara os métodos de otimização dos autores citados acima para cada tipo de problema diferente, incluindo também Pons e Latapy (PONS PASCAL; LATAPY, 2006), Wakita e Tsurumi (WAKITA KEN; TSURUMI, 2007). Os valores são dados em modularidade/tempo, quanto maior a modularidade melhor, pois representa uma comunidade mais bem definida, e quanto maior a velocidade mais eficiente é o algoritmo (roda em um tempo menor).

	Karatê	**Arxiv**	**Internet**	**Web nd.edu**	**Telefone**	**Web uk-2005**	**Web WebBase 2001**
Nós / links	34/77	9k / 24k	70k / 351k	325k / 1M	2,6M / 6,3M	39M / 783M	118M / 1B
Clauset, Newman e Moore	.38 / 0s	.772 / 3.6s	.692 / 799s	.927 / 5034s	- / -	- / -	- / -
Pons e Latapy	.42 / 0s	.757 / 3.3s	.729 / 575s	.895 / 6666s	- / -	- / -	- / -
Wakita e Tsurumi	.42 / 0s	.761 / 0.7s	.667 / 62s	.898 / 248s	.56 / 464s	- / -	- / -
Método Louvain	.42 / 0s	.813 / 0s	.781 / 1s	.935 / 3s	.769 / 134s	.979 / 738s	0,984 / 152mn

Figura 2 – Comparação de otimização de modularidade. -/- refere-se aos métodos que demoraram mais de 24h para ser executado

Para esse trabalho será focado mais nos métodos de Girvan e Newman, Ravasz e Lancichinetti-Fortunato-Radicchi para os testes de benchmark. Nas próximas seções serão abordados com mais detalhes a explicação dos modelos e seus algoritmos, como também alguns conceitos a respeito do problema de identificação de comunidades, visando também suas implicações em redes maiores, a qual se tem muita importância a performance.

Parte I

Algoritmos de detecção de comunidades

1 Conceitos básicos de comunidades

1.1 O que são comunidades

Redes complexas são formadas por diversas estruturas modulares que podemos chamar de comunidades. Essa estrutura está unicamente codificada no seu diagrama de conexões e através do estudo dessas conexões podemos estudar e identificar essas comunidades.

O entendimento do que são comunidades está fortemente ligado a dois importantes conceitos aplicados à redes: conectividade e densidade. Uma comunidade é definida como um subgrafo densamente conectado internamente e com poucas conexões externas.

Mais especificamente, todos os membros de uma comunidade devem ser capazes de alcançar qualquer outro membro da mesma comunidade (relacionado à conectividade do subgrafo) e um membro deve ter mais chances de se conectar com outros membros da comunidade que com outros nós da rede (relacionado à densidade do subgrafo).

É importante ressaltar que esses subgrafos que definem uma comunidade não precisam ser grafos completos (com número máximo de arestas). No mundo real uma comunidade representada por um grafo completo seria aquela em que todos os membros se conhecem (ou seja, estão conectados), o que não representa a maioria dos casos de redes complexas.

Com essa consideração podemos então classificar comunidades olhando para o número de conexões que cada nó da comunidade faz com nós internos (dentro da comunidade) e externos (fora da comunidade), classificando-a como forte ou fraca.

Dado uma comunidade C com N_C nós, o grau interno k_i^{int} do nó i será o número de conexões de i com outros nós de C. O grau externo k_i^{ext} de i será o número de conexões que i faz com o resto da rede. Se $k_i^{ext} = 0$, todos os vizinhos de i são pertencentes à comunidade C, logo essa comunidade é uma boa escolha para i. Por outro lado, se $k_i^{int} = 0$, o nó i deve ser colocado em outra comunidade.

Utilizando os dados de grau interno e externo de cada nó temos que se o grau interno é maior que o grau externo em cada nó da comunidade esta é então uma comunidade forte. Se a somatória do grau interno de todos os nós da comunidade for maior que a somatória do grau externo de todos os nós da comunidade, esta é classificada como fraca.

Pela definição todo subgrafo completo é uma comunidade forte e toda comunidade forte também é uma comunidade fraca. Não podemos dizer, porém, que toda comunidade fraca também é classificada como forte. Sendo a definição de comunidade fraca uma suavização das condições de classificação da comunidade forte, permitindo que alguns nós violem a condição $k_i^{int} > k_i^{ext}$.

Assim, dado que redes são compostas por comunidades, um problema importante e recorrente é a detecção de comunidades, como já insinuado pelas definições apresentadas acima a respeito dos graus interno e externo dos nós. Podemos seguir diversas abordagens para essa detecção, desde as mais simples porém custosas às mais eficientes e aprimoradas.

1.2 Força bruta e cut size

Uma primeira abordagem para pensar no problema de identificar as comunidades pode ser o que chamamos de bisseção de grafos, que consiste em separar a rede em 2 subgrafos sem interseções, objetivando ter 2 grupos com o menor número possível de conexões entre eles, chamados de *cutsize*. Esse caso segue a linha dos problemas de partição de grafos, que embora sejam muito parecidos com os de detecção de comunidades, diferem pelo fato de que as comunidades não têm número nem tamanho específico, ocorrendo naturalmente na rede e sendo codificados pelas conexões da rede como mencionado mais acima. Esse modelo de detecção é, portanto, uma simplificação do problema real.

Para resolver este problema devemos fazer todas as partições possíveis em dois grupos e escolher aquela com menor *cutsize*, ou seja, menor número de conexões com o outro grupo. Dessa forma estaríamos garantindo que esses dois grupos são os que têm maior número de conexões entre o próprio grupo e somente algumas conexões com o outro, sendo assim, as melhores opções de separação entre comunidades, levando em consideração os parâmetros já apresentados. Esse seria um algoritmo guloso, que busca todas as soluções possíveis e escolhe a melhor, usando de força bruta para encontrá-la. Sendo assim, é um método bastante custoso. O número de partições distintas em uma rede com N nós separados em grupos N_1 e N_2 seria:

$$\frac{N!}{N_1!N_2!} \tag{1.1}$$

Para simplificar o cálculo podemos utilizar a aproximação de Sterling:

$$n! \simeq \sqrt{2\pi n}(n/e)^n$$

Substituindo na equação 1.1:

$$\frac{N!}{N_1!N_2!} \simeq \frac{\sqrt{2\pi N}(N/e)^N}{\sqrt{2\pi N_1}(N_1/e)^{N_1}\sqrt{2\pi N_2}(N_2/e)^{N_2}} \sim \frac{N^{N+1/2}}{N_1^{N_1+1/2}N_2^{N_2+1/2}} \tag{1.2}$$

Podemos ainda assumir que N_1 e N_2 tem o mesmo tamanho $N/2$ e chegar no resultado abaixo:

$$\frac{2^{N+1}}{\sqrt{N}} = e^{(N+1)\ln 2 - \frac{1}{2}\ln N} \tag{1.3}$$

Com esse resultado percebemos que o número de partições que devemos fazer para chegar na solução ótima aumenta exponencialmente conforme N cresce, o que torna essa solução extremamente custosa e impraticável em situações reais com redes de N grande.

Para uma rede de 10 nós, para $N_1 = N_2 = 5$, aplicando os cálculos acima seria necessário checar 252 bissecções para encontrar a com menor *cutsize*. Por outro lado, em uma rede de 100 nós, com $N_1 = N_2 = 50$ precisaríamos realizar 1029 partições, o que com um tempo de um milissegundo para cada partição levaria 10^{16} anos (BARABáSI; PóSFAI, 2016).

Como foi mencionado, o problema de detectar comunidades pode parecer similar ao de bisseção de grafos, porém, no caso de comunidades o número de partições assim como o tamanho de cada partição é variado, além dos dois serem parâmetros desconhecidos. O número necessário de partições possíveis em uma rede com os parâmetros descritos acima seria então:

$$B_N = \frac{1}{e}\sum_{j=0}^{\infty}\frac{j^N}{j!} \tag{1.4}$$

Se plotarmos um gráfico das expressões 1.3 e 1.4 e com N variando, percebemos que a expressão que representa o número de partições considerando a definição de comunidade segue a Série de Bell, crescendo ainda mais rápido que a curva exponencial (Figura 3).

Fica claro então, que, dado a definição de comunidades, com inúmeras partições possíveis, além de características variáveis como cliques (grafo completo), comunidades fracas e fortes, é inviável buscar soluções para o problema de detecção de comunidades através de algoritmos de força bruta, sendo necessário assim buscar outras abordagens.

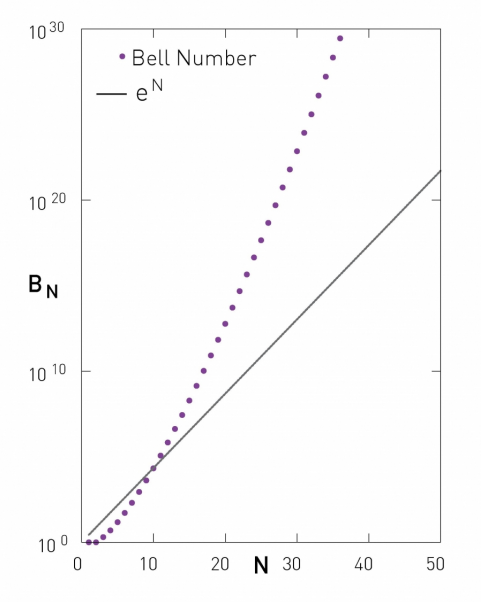


Figura 3 – Número de partições em uma rede, comparando expressão com Série de Bell e exponencial
(BARABáSI; PóSFAI, 2016)

1.3 Clustering hierárquico

Para detectar comunidades em redes reais é necessário usar uma abordagem diferente da apresentada anteriormente, com tempos de execução crescendo segundo número de Bell. Para ser viável, a solução adotada deve ter seu tempo de execução crescendo em uma proporção polinomial em relação ao N, existindo algumas opções para tal.

O ponto de partida para algoritmos de clustering hierárquico é a matriz de similaridade, que utiliza a representação x_{ij} para indicar a distância do nó i ao nó j. Essa distância pode ser calculada de diversas maneiras como similaridade por cosseno ou distância euclidiana, e essa escolha impacta no resultado final, sendo assim mais uma variável que pode trazer benefícios para o algoritmo ou dificultar sua efetividade.

Após calcular a similaridade de todos os pares de nós na rede, os nós com maior similaridade são agrupados na mesma comunidade. É importante notar que nem sempre essa escolha é tão direta, se pegarmos como exemplo três nós A, B e C cujas similaridades são fortes para o par A e B, para o par B e C, e fraco para o par A e C, conforme descrito na Figura 4.

Poderíamos entender que por B possuir similaridade forte com A assim como com C, tanto A, B e C devem ficar na mesma comunidade, apesar de A e C terem

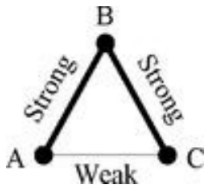


Figura 4 – similaridade entre nós A, B e C
(NEWMAN, 2010)

pouca similaridade. Essa escolha, porém, pode se mostrar errada já que A e C tem pouca similaridade.

Para implementar o algoritmo podemos seguir dois caminhos diferentes, o dos algoritmos aglomerativos, que juntam nós com similaridade alta em uma mesma comunidade e o dos algoritmos divisivos, que separam as comunidades ao remover os nós com baixa similaridade conectados na comunidade. Em ambos casos o resultado final é uma árvore hierárquica, chamada de dendrograma, que mostra as possíveis separações em comunidades.

1.3.1 Algoritmos aglomerativos

Para ilustrar os procedimentos dessa categoria de algoritmos, a seguir serão apresentados os passos do algoritmo de Ravasz.

O primeiro passo é definir a matriz de similaridade.

Nos algoritmos aglomerativos os pares de nós com similaridade alta serão mantidos na mesma comunidade e os com similaridade baixa devem ficar em comunidades diferentes. Se analisarmos os nós do ponto de vista de vizinhos (olhar então para a matriz de adjacência) podemos concluir que nós que estão conectados e que possuem vizinhos em comum provavelmente devem estar na mesma comunidade, sendo assim, o valor de x_{ij} deve ser alto.

Podemos utilizar o cálculo a seguir para encontrar os valores de x^o_{ij} (que serão a medida de overlap topológico):

$$x^o_{ij} = \frac{J(i,j)}{\min(k_i, k_j) + 1 - \Theta(A_{ij})} \tag{1.5}$$

Na equação, $\Theta(x)$ é a função de Heaviside, cujo valor é zero para $x \leq 0$ e um para $x > 0$, $J(i,j)$ é o número de vizinhos em comum entre i e j, na fórmula adicionamos $+1$ a

esse valor caso exista uma conexão direta entre i e j. O $\min(k_i, k_j)$ vai ser o menor valor do grau de i e j.

Podemos tirar algumas conclusões sobre os resultados da equação:

1. Se a medida de overlap topológico para i e j for igual a 1 significa que os nós possuem os mesmos vizinhos e estão conectados diretamente.
2. Se a medida de overlap topológico for zero para i e j, os nós não possuem vizinhos em comum e nem estão conectados.
3. Se os nós forem vizinhos a medida de overlap topológico será alta.

O segundo passo é comparar a similaridade dos grupos encontrados com os outros grupos. Para tal podemos seguir três abordagens diferentes: similaridade individual (single similarity), completa (complete similarity) ou média (average similarity). A intenção com esse passo é definir a matriz de similaridade.

Na similaridade individual a similaridade entre dois grupos é igual a similaridade entre os elementos com menor distância (similaridade). Os dois nós comparados devem ser de comunidades diferentes.

A similaridade completa utiliza a mesma ideia da similaridade individual porém utiliza como referência os elementos com distância(similaridade) mais próximas entre os clusters.

Na similaridade média é considerada a distância média de cada combinação de dois nós entre os dois clusters analisados.

O diagrama ilustra o uso das diferentes abordagens possíveis nesse segundo passo, sendo x_{ij} nesse caso uma medida de distância entre os nós. Na imagem b, onde utiliza-se a similaridade individual é escolhida a menor distância entre nós, 1.59 entre os nós C e E. Na imagem c, com o uso na similaridade completa escolhem-se os nós mais distantes, B e F, com distância de 3.97. Por fim, no caso da imagem d, com o uso da similaridade média, é feita a média de todas as distâncias entre os pares de nós, obtendo o valor de 2.84. O algoritmo de Ravasz usado como exemplificação, utiliza a similaridade média.

O terceiro passo é aplicar o clustering hierárquico. O algoritmo irá designar uma comunidade para cada nó da rede e avaliar o valor de x_{ij} para todos os pares da rede, descobrindo assim as similaridades. Os nós serão unidos aos outros nós ou comunidades com maior similaridade. A cada junção as similaridades serão recalculadas e os dois passos

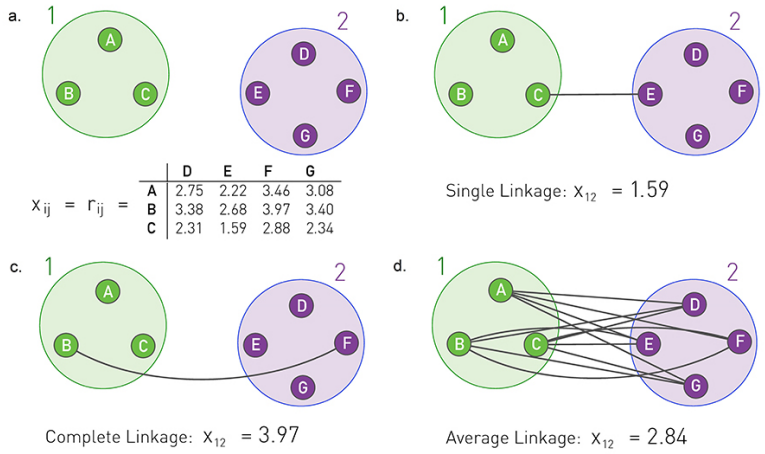


Figura 5 – exemplo da aplicação de similaridade individual, completa e média (BARABáSI; PóSFAI, 2016)

descritos anteriormente serão repetidos até todos os nós serem agrupados em uma única comunidade. Então, nesse tipo de algoritmo, uma rede com N nós começa a execução com N comunidades e ao final de todas as iterações fica com uma única comunidade.

O último passo é analisar como as comunidades foram sendo formadas durante as iterações. Para tal podemos utilizar um dendrograma, que mostra a ordem com que os grupos se formaram e como foi feita a junção dos nós. Ao cortar o gráfico em diferentes pontos, temos diversas possibilidades de divisões em comunidades. É importante ressaltar que o algoritmo não sabe informar qual é o melhor número de divisões, onde cortar o gráfico, o usuário deve escolher a melhor formação para o problema analisado. Podemos melhorar o algoritmo para trazer uma resposta mais assertiva sobre a melhor divisão aplicando o conceito de modularidade.

1.3.1.1 Complexidade

Como foi descrito no começo da sessão, é crucial conseguir manter a complexidade na ordem polinomial para termos um algoritmo praticável para redes reais. Como o algoritmo é separado em quatro passos, devemos analisar a complexidade de cada um deles separadamente.

1. Passo 1: nesse passo devemos calcular a similaridade entre todos os pares de nós da rede, Assim faremos N^2 comparações, obtendo uma complexidade da ordem $O(N^2)$
2. Passo 2: para analisar a similaridade entre grupos precisamos comparar em cada passo a distância entre o novo grupo e todos os outros, sendo a complexidade $O(N^2)$

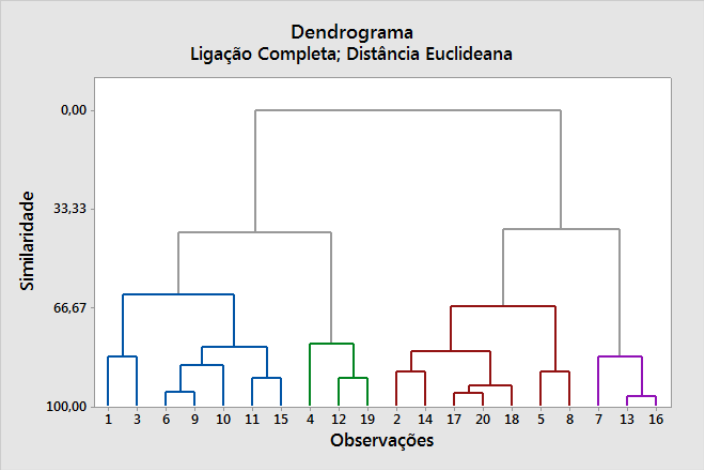


Figura 6 – exemplo de dendrograma
(DENDROGRAMA,)

3. Passos 3 e 4: a construção do dendrograma pode ser feita em $O(N \log N)$

A complexidade final do algoritmo é a soma de todas as complexidades $O(N^2) + O(N^2) + O(N \log N)$, prevalecendo a complexidade $O(N^2)$, muito mais eficaz que o do algoritmo de força bruta, com complexidade de $O(e^N)$.

1.3.2 Algoritmos divisivos

Algoritmos divisivos funcionam removendo conexões que pertencem a comunidades diferentes, iniciando o algoritmos com uma única comunidade e iterando até obter N comunidades de um único nó, fazendo o oposto dos algoritmos aglomerativos.

Novamente, a exemplificação de como esse tipo de algoritmo funciona será feita usando um algoritmo específico, o de Girvan-Newman.

Passo 1: similarmente ao algoritmo anterior é necessário calcular os valores de x_{ij}, que nesse caso será chamado de centralidade. Aqui pares de nós que pertencem a comunidades diferentes serão os nós selecionados. Um x_{ij} alto (ou baixo dependendo da abordagem) será o desejado, representando dois nós i e j que pertencem a comunidades diferentes. Consequentemente, nós presentes na mesma comunidade tem x_{ij} baixo. Dois tipos de centralidade são os mais comuns:

1. Betweenness das conexões: considera a influência de cada conexão na forma como as informações fluem na rede. x_{ij} é proporcional aos caminhos mais curtos entre os

pares. As conexões que fazem o link entre comunidades vão ser mais utilizados e consequentemente terão valor de betweenness alto.

2. Random-Walk Betweenness: para encontrar essa medida dois nós são escolhidos aleatoriamente m e n, é traçado um caminho começando em n passando por conexões adjacentes com as mesmas probabilidades até chegar no nó m. A medida de centralidade será então a probabilidade da conexão i, j ter sido escolhida nesse caminho de n até m.

No Girvan-Newman é utilizada a abordagem de betweenness.

Passo 2: esse passo é similar ao último passo dos algoritmos aglomerativos, consistindo da aplicação do clustering hierárquico, através dos passos:

1. Computar a centralidade x_{ij} de cada conexão
2. Remover a conexão com maior centralidade, escolhendo aleatoriamente em caso de empate
3. Recalcular as centralidades de cada conexão na rede alterada
4. Repetir os passos 2 e 3 até todas as conexões serem removidas e existirem N comunidades com 1 elemento cada

A execução também gera um dendrograma com as separações das comunidades.

1.3.2.1 Complexidade

A complexidade desse tipo de algoritmo vai depender da medida de centralidade escolhida, a complexidade para o betweenness é $O(LN)$. A complexidade final do algoritmo seria então $O(L^2N)$ (como o passo 3 adiciona o fator L) ou $O(N^3)$ em redes esparsas.

2 Avaliação e otimização de comunidades

Dada a natureza do problema da detecção de comunidades, é perceptível que boa parte do desafio se encontra na necessidade de tornar os algoritmos mais assertivos, identificando comunidades mais significativas, e ao mesmo tempo computacionalmente viáveis, para que possam ser utilizados em redes de larga escala. No caso das possíveis comunidades que surgem através do agrupamento hierárquico, por exemplo, uma medida de qualidade das comunidades pode responder a questão de onde ou quando cortar o dendrograma para se obter a melhor partição para a rede em questão.

Existem diversas funções usadas para realizar esta medição e quantificar a qualidade e a estabilidade de comunidades em redes complexas, estas podem ser divididas entre duas categorias, as que utilizam medidas internas e as que utilizam medidas externas (PONVENI; VISUMATHI, 2021).

A seguir, exemplificamos algumas funções de medidas de qualidade de comunidades comumente utilizadas.

2.1 Medidas internas

As medidas internas avaliam o resultado do agrupamento sem nenhuma informação externa e são baseados apenas em informações intrínsecas aos dados da rede.

2.1.1 Modularidade

A modularidade desempenha um papel importante no entendimento da estrutura de comunidade de uma rede, sendo a medida mais comum de avaliação entre os algoritmos de detecção de comunidades e oferecendo entendimento sobre a estrutura de comunidade de uma rede.

Baseada na hipótese de que redes que tem conexões geradas de forma aleatória não possuem estrutura de comunidade, a modularidade compara a densidade de ligações de uma comunidade com a densidade de ligações obtida para o mesmo grupo de nós em uma rede gerada com base na rede que está sendo avaliada, na qual o grau de cada nó se mantém, mas as ligações são refeitas aleatoriamente, indicando se a comunidade

original corresponde a um subgrafo denso ou se o padrão de suas ligações surgiu por acaso (BARABáSI; PóSFAI, 2016).

A modularidade pode ser obtida da seguinte forma (NEWMAN; GIRVAN, 2004):

Considere uma rede com N nós e L links e uma partição em n_c comunidades, cada comunidade tendo N_c nós conectados entre si por L_c links, onde c=1,...,nc. Se L_c for maior que o número esperado de links entre os mesmos N_c nós obtido na rede aleatória, então os nós do subgrafo C_c poderiam de fato fazer parte de uma comunidade verdadeira.

Com isso, primeiramente calcula-se o número de arestas presentes em cada comunidade ($\boldsymbol{H}$), sendo o somatório do produto dos valores dos vértices i e j na matriz de adjacência (A_{i_j}) pelo delta de Kronecker obtido das comunidades a que pertencem ($\delta(c_i, c_j)$). Multiplica-se o resultado do somatório por 1/2 para eliminar a dupla contagem das ligações.

$$H = 1/2 \sum_{i=1}^{N} \sum_{j=1}^{N} A_{i_j} \delta(c_i, c_j) \tag{2.1}$$

Em seguida, calcula-se o número esperado de conexões entre vértices da mesma comunidade na rede com conexões geradas aleatoriamente, sendo o número de conexões esperadas entre os vértices i e j (E_{i_i}) dado pelo produto do grau do vértice i (k_i) com a probabilidade de ligação com o vértice j ($k_j/2L$).

$$E_{i_j} = k_i \frac{k_j}{2L} \tag{2.2}$$

Então, o número de conexões esperadas entre nós da mesma comunidade em toda a rede ($\boldsymbol{R}$) será o somatório do produto de E_{i_i} pelo delta de Kronecker de ci, cj. Novamente multiplicando o resultado por 1/2 para eliminar a dupla contagem de ligações.

$$R = 1/2 \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{k_i k_j}{2L} \delta(c_i, c_j) \tag{2.3}$$

A modularidade é dada pela diferença entre H e R, e quantifica a tendência dos nós de uma mesma classe estarem conectados em uma rede.

$$M = (H - R)\frac{1}{2L} = 1/2 \sum_{i=1}^{N} \sum_{j=1}^{N} \left[A_{i_j} - \frac{k_i k_j}{2L} \right] \delta(c_i, c_j) \tag{2.4}$$

A Modularidade pode variar de -1 a 1, sendo que:

- Quanto maior o valor de M para uma partição, melhor é sua correspondência a uma estrutura de comunidade.
- A modularidade será zero se toda a rede for tomada como uma única comunidade.
- A modularidade será negativa se cada nó pertencer a uma comunidade diferente.

Se o valor da modularidade for maior do que 0.3, indica-se uma rede com estrutura modular. Este resultado nunca foi provado, mas é observado na prática.

A modularidade é então usada para decidir qual das muitas partições previstas por um método hierárquico oferece a melhor estrutura de comunidade, selecionando aquela para a qual M é máximo.

2.1.1.1 Exemplos de algoritimos baseados em modularidade

The Greedy Algorithm: une iterativamente pares de comunidades se tal movimento aumentar a modularidade da partição. O algoritmo segue os seguintes passos:

- Atribua cada nó a uma comunidade própria, começando com N comunidades de nós únicos.
- Inspecione cada par de comunidade conectadas por pelo menos uma aresta e calcule a diferença de modularidade ΔM obtida se as unirmos. Identifique o par de comunidades para o qual ΔM é o maior e funda-as. Observe que a modularidade é sempre calculada para a rede completa.
- Repita a Etapa 2 até que todos os nós se unam em uma única comunidade, registrando M para cada etapa.
- Selecione a partição para a qual M é máximo.

Note que o algoritmo segue uma abordagem aglomerativa e sua complexidade de tempo é O(N^2). Este algoritmo possui uma otimização chamada *Fast Greed Algorithm* com complexidade de tempo $O(nlog^2n)$ em redes esparsas.

***Girvan-Newman* com modularidade**: já citado anteriormente, com a aplicação da modularidade para otimização este algoritmo segue os seguintes passos:

- Calcule a centralidade de todas as arestas.

- Remova a aresta com a maior centralidade.
- Calcule a modularidade e guarde seu valor em um vetor.
- Recalcule a centralidade de todas as arestas
- Repita o ciclo a partir do passo passo 2
- Depois de remover todas as arestas, retorne ao grafo original e remova as arestas até atingir a modularidade máxima.

Seguindo uma abordagem divisiva, é um algoritmo fácil de implementar e de entender, porém, sua complexidade de tempo é O(NL^2), o que o torna inviável para redes de grande escala.

2.1.2 Outras medidas internas

- **Taxa de participação de triângulo (TPR = $\frac{T_c}{N_c}$)**: dada pela quantidade de nós que pertencem a um triângulo. Onde T_c é a quantidade de nós que formam o triângulo na comunidade c e N_c é o número de nós em c.
- **Condutância (*conductance*)**: para grafos não ponderados, quantifica a porção do número absoluto de arestas que apontam para fora da comunidade. Para um grafo ponderado, mede a parte do peso total de tais arestas. O valor mais baixo de Condutância indica maior qualidade da comunidade.
- **Índice Calinski-Harabasz**: além compactação interna dos grupos, este índice também considera a destacabilidade dos grupos. Fundamentalmente similar ao *F-test* em ANOVA.
- **Índice de silhueta**: mede de maneira quantitativa o quão bem cada nó reside em determinada comunidade em comparação com outras.
- **Densidade de partição**: é a quantidade ponderada da densidade de dois segmentos vizinhos. Pode ser usada para encontrar comunidades úteis avaliadas pela semelhança dos metadados dos nós.
- **Precisão**: considera todos os pares possíveis de nós verificando se residem na mesma comunidade.
- **Cobertura**: avalia a fração de nós atribuídos a pelo menos uma comunidade.

- **Expansão**: estima o número médio de arestas (por vértice) que apontam para fora da comunidade ou do peso médio por nó de tais arestas. A qualidade da comunidade é decidida pelo valor de Expansão. Quanto menor o valor de expansão maior a qualidade da comunidade.

2.2 Medidas externas

De um modo geral, as medidas externas avaliam o resultado do agrupamento comparando-o com um resultado externo que forneça valores verdadeiros, a estrutura de comunidade que está sendo avaliada é comparada com uma estrutura referência estabelecida como verdadeira.

2.2.1 Informação Mútua Normalizada

Informação Mútua Normalizada, ou *Normalized Mutual Information (NMI)* é um método amplamente utilizado para avaliar algoritmos de detecção de comunidade e clusterização, ele é definido da seguinte forma (PONVENI; VISUMATHI, 2021):

$$NMI(C_i : C_j) = \frac{H(C_i) + H(C_j) - H(Ci, Cj)}{(H(C_i) + H(C_j)/2)} \tag{2.5}$$

Onde, C_i e C_j são considerados como comunidades de duas redes diferentes $N1$ e $N2$ e $H(C_i)$ e $H(C_j)$ são chamados valores de entropia de NMI, variando entre 0 e 1. Onde $NMI = 0$ indica que não há correspondência entre as comunidades, ou seja, terá valor 0 se o agrupamento for aleatório e não houver estrutura de comunidade, e $NMI = 1$ indica uma recriação perfeita entre da comunidade.

Apesar de bastante utilizada, existem questionamentos levantados em relação a NMI como medida de avaliação de comunidades em redes, sendo estes relacionados a possibilidade da mesma levar a erros sistemáticos devido ao tamanho finito das redes ou resultar em estimativas errôneas da performance de algoritmos (ZHANG, 2015).

2.2.2 Outras medidas externas

- **Pureza**: calcula a porcentagem de elementos posicionados corretamente nos agrupamentos comparando os obtidos com os selecionados como verdadeiros. A pureza penaliza itens de diferentes categorias em um mesmo agrupamento, mas não recom-

pensa os itens de uma mesma categoria agrupadas em um mesmo agrupamento, portanto pode ser máxima quando cada elemento pertencer a um único grupo.

- **Variação da informação (VI)**: uma medida da distância entre dois agrupamentos. Está intimamente relacionado à informação mútua.
- **Fração de nós classificados corretamente (FCC)**: esta medida diz que um nó foi classificado corretamente se sua comunidade estimada é a mesma que para a maioria dos nós presentes em sua comunidade de referência.

Parte II

Testes de algoritmos de detecção de comunidades

3 Testes de algoritmos para detecção de comunidades

Testes de algoritmos para detecção de comunidades nascem com a proposta de responder a seguinte pergunta: Como comparar os algoritmos de detecção de comunidades?

Quando construímos um algorítimo que detecta comunidades, também é necessário testar o seu desempenho. Nas seções anteriores, discutimos brevemente ferramentas que podem ser utilizadas para este fim, nesta seção iremos nos aprofundar mais em tal discussão levantando a prática dos métodos de comparação e como podemos controlar esses *benchmarks*.

3.1 Avaliação do algoritmo

Ao trazer o conceito de avaliação de um algoritmo para a detecção de comunidades, é possível levantar inicialmente em duas medidas, o primeiro será discutido ao abordar o *Girvan Newman (GN) Benchmark* na seção 3.2. Nesse método a avaliação ocorre mediante a taxa de acerto do algorítimo, ou seja após o algorítimo executar a detecção de comunidades, é medido uma razão entre quantos vértices da rede foram categorizados corretamente dentro da mesma comunidade pelo número total de vértices na rede (FCC 2.2.2). A segunda forma de avaliação já foi discutido em medidas externas na seção Informação Mútua Normalizada 2.2.1, essa medida externa será utilizado no método de *Lancichinetti-Fortunato-Radicchi (LFR) Benchmark* 3.3.

3.2 Girvan-Newman (GN) Benchmark

Ao falar de testes para algorítimos de detecção de comunidades, podemos começar pelo método de comparação descrito por Girvan e Newman (GIRVAN; NEWMAN, 2002). Para realizar o GN Benchmark é necessário gerar uma rede com comunidades, porém com parâmetros e métricas para regular as mesmas. Com isso GN cria o primeiro método que funciona da seguinte forma:

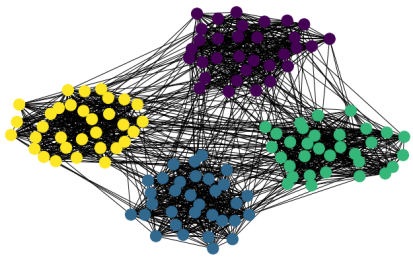
- Cria-se uma rede com 128 vértices

- Com 4 comunidades presentes
- Com 32 vértices por comunidade
- Cada vértice possui 16 ligações

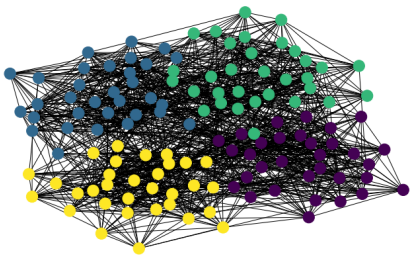
Estas ligações dos vértices, podem ser feitas internamente, ou seja, dentro da própria comunidade do vértice, quanto externamente, com vértices fora da comunidade. Sendo P_{in} a probabilidade de realizar uma ligação interna e P_{out} com $P_{out} < P_{in}$. Como cada vértice possui 16 ligações somando ligações internas e externas, podemos chamar de K o número total de ligações de um vértice, e K_{in} número de ligações internas e K_{out} número de ligações externas, assim chegando em $K = K_{in} + K_{out}$ ou nesse caso $K_{in} + K_{out} = 16$. Ao aumentar o número de ligações internas, o número de ligações externas diminui e vice e versa.

(a) $K_{out} = 0$

(b) $K_{out} = 2$



(c) $K_{out} = 4$



(d) $K_{out} = 8$

Figura 7 – Figura recriada a partir da referencia (FORTUNATO; CASTELLANO, 2008)

Dessa forma, podemos criar redes com comunidades utilizando os parâmetros K_{in} e K_{out} para determinar o nível de modularidade da rede. Utilizando o $K_{out} = 0$ ou seja $P_{out} = 0$ e $P_{in} = 1$, Teremos uma rede com 4 comunidades desconexa, nenhum vértice de uma comunidade faz conexão com um vértice externo. na Figura 7a podemos ver graficamente como é seu comportamento. Ao aumentarmos o K_{out} até chegar em $K_{out} = 8$, o número de ligações internas e externas é o mesmo, dificultando o processo de detecção

de comunidades. Criando assim uma forma de testar e comparar algorítimos de detecção de comunidades.

Na Figura 8 temos graficamente o desempenho do algorítimo *The Greedy Algorithm* descrito na seção 2.1.1.1. É possível notar que ao aumentar a probabilidade P_{out} da rede, diminui a fração de vértices classificados corretamente (FCC) ao detectar comunidades.

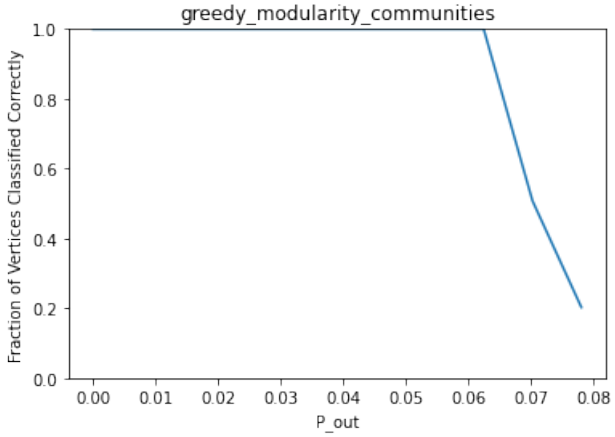


Figura 8 – Imagem gerada a partir da biblioteca Networkx

Este método de testes de algorítimos se torna o primeiro a ser abordado na literatura, por ser o mais fácil de entender. Porém ele traz algumas limitações, a primeira e a maior delas é o fato de que no momento de criar a rede com suas comunidades para executar os testes, é criado redes regulares, o número de comunidades e o tamanho da comunidade é fixo, seguindo assim uma distribuição de grau homogênea. Porém as redes reais não seguem uma distribuição homogênea, elas seguem como discutido em (BARABáSI; ALBERT, 1999) uma distribuição de lei de potência. Em uma rede real temos muitas comunidades com poucos vértices, enquanto algumas comunidades são muito grandes, ou seja seguindo uma irregularidade. Para resolver esse problema não é possível gerar uma rede com comunidades regulares, e esse tópico será abordado no algorítimo *Lancichinetti-Fortunato-Radicchi (LFR)* na seção 3.3.

3.3 Lancichinetti-Fortunato-Radicchi (LFR) Benchmark

A proposta do método LFR para testes de algorítimos para detecção de comunidades surge com a discussão levantadas pelos seus autores (LANCICHINETTI; FORTUNATO; RADICCHI, 2008), no intuito de gerar redes que seguem uma lei de potência, ou seja redes sem escala, assim se aproximando de redes reais. Nesse modelo tanto a distribuição de grau

quanto o tamanho das comunidades da rede seguem uma lei de potência. Para a realização dos testes faz-se necessário obter o controle da rede, o parâmetro μ é utilizado desta forma, controlando a fração de conexões entre as comunidades, quanto maior o parâmetro, mais difícil é a identificação das comunidades dentro da rede, uma vez que as redes serão mais conectadas. O comportamento do parâmetro μ se assemelha ao parâmetro K_{out} no método de *Girvan-Newman* 3.2. Para gerar a rede proposta no método LFR, é seguido os 5 passos a seguir:

1. Gere uma distribuição seguindo uma lei de potência com o expoente γ. Crie uma rede e atribua a cada vértice da rede um grau obtido da distribuição. São escolhidos k_{min} como o menor grau da rede, k_{max} o maior grau da rede e $\langle k \rangle$ como grau médio. É possível criar esta rede utilizando o Modelo de Configuração (*Configuration Model*), discutido em (BARABáSI; PóSFAI, 2016).

2. Cada vértice compartilha uma fração $(1 - \mu)$ de suas conexões com os outros vértices de sua comunidade e uma fração μ com os demais vértices da rede. Sendo assim μ um parâmetro de mistura entre as comunidades.

3. O tamanho da comunidade também é retirado de uma lei de potência com o parâmetro β. E a soma de vértices de todas as comunidades é igual a N, número total de vértices da rede. Igualmente ao gerar o grau de vértices é parametrizado a comunidade com tamanho mínimo e tamanho máximo, sendo s_{min} e s_{max} respectivamente, tal qual $s_{min} > k_{min}$ e $s_{max} > k_{max}$.

4. No início, todos os vértices estão desabrigados (*homeless*), ou seja não pertencem a nenhuma comunidade. Na primeira iteração, um vértice é atribuído aleatoriamente a uma comunidade, porém o vértice apenas entra na comunidade se o tamanho da comunidade for maior que o grau interno do vértice. Caso contrário o vértice continua desabrigado. Essa iteração se repete sucessivamente, atribuindo vértices a comunidades aleatórias, e verificando a condição de entrada. Depois de sucessivas iterações, pode chegar no caso de uma comunidade ficar lotada, nesse caso é escolhido um vértice aleatório para ser expulso da comunidade, tornando-o um vértice desabrigado, o processo de atribuição se repete. Esse passo termina quando não sobrar nenhum vértice desabrigado.

5. Por último, o passo para impor a fração, expressa pelo parâmetro μ. Nesse passo são feitas varias etapas de religação, de modo que os graus de todos os vértices

permanecem os mesmos, apenas o grau interno é afetado. Dessa forma, a razão entre o grau externo e interno de cada vértice é definido pelo parâmetro μ desejado.

Para a validação de desempenho de um algorítimo utilizando o método LFR é utilizado a medida externa Informação Mútua Normalizada 2.2.1. Nessa medida levamos em consideração a entropia de Shennon (SHANNON, 1948). Com ela é possível medir a quantidade de informação que temos de uma variável aleatória.

$$H(\{C_1\}) := -\sum_{C_1} p(C_1) log\, p(C_1) \tag{3.1}$$

Sendo C_1 a comunidade que em análise, $p(C_1)$ a razão entre a quantidade de vértices que pertencem a C_1 com o número total de vértices na rede. Ao utilizar a entropia de Shennon, é possível definir a Informação Mutua 3.2. E por fim a Informação Mutua Normalizada definida na Seção (2.2.1).

$$I(\{C_1\}, \{C_2\}) = \sum_{C_1} \sum_{C_2} p(C_1, C_2) log\, \frac{p(C_1, C_2)}{p(C_1)p(C_2)} \tag{3.2}$$

$$p(C_1, C_2) = \frac{\text{Quantidade de vértices que estão em } C_1 \text{ e também estão em } C_2}{\text{Soma de todos os possíveis pares de } C_1 \text{ e } C_2} \tag{3.3}$$

$$\mu = \frac{k_{out}}{k_{in} + k_{out}} \tag{3.4}$$

Na Figura 9 retirada do exemplo do (BARABáSI; PóSFAI, 2016) é possível ver como essa medida se comporta, utilizando o Algorítimo de Detecção de Comunidades Ravasz (DERITEI et al., 2014), nela é utilizado o parâmetro μ descrito na equação 3.4. É possível notar que ao chegar na razão $0,4$ até $0,5$, ocorre uma queda abrupta no desempenho do algorítimo. Nesse exemplo o algorítimo de teste é o *Girvan-Newman* 3.2.

Agora na Figura 10 temos uma comparação de dois métodos de comparação de Algorítimos de Detecção de Comunidades. Também retirado do livro (BARABáSI; PóSFAI, 2016). Nessa Figura é possível notar como os dois algorítimos de comparação geram informações diferentes. Um exemplo disso é que enquanto no *NG Benchmark* o algorítimo de detecção *Greedy Modularity* aparenta ter um desempenho muito parecido com os demais algorítimos, enquanto ao analisar o *LFR Benchmark* é possível notar que esse algorítimo não alcança um bom desempenho quando comparado aos demais.

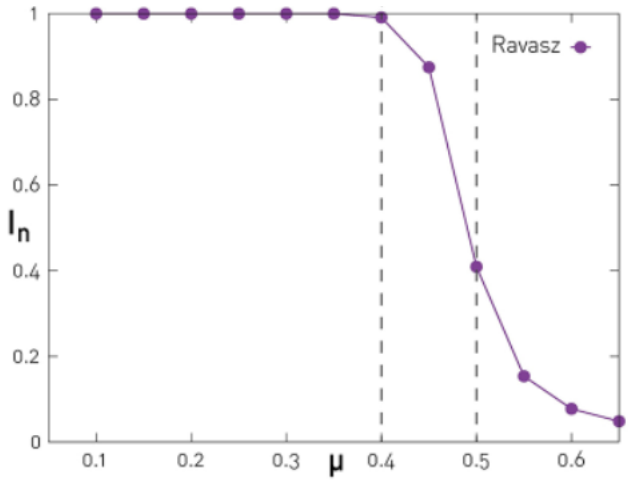


Figura 9 – Figura retirada do livro (BARABáSI; PóSFAI, 2016)

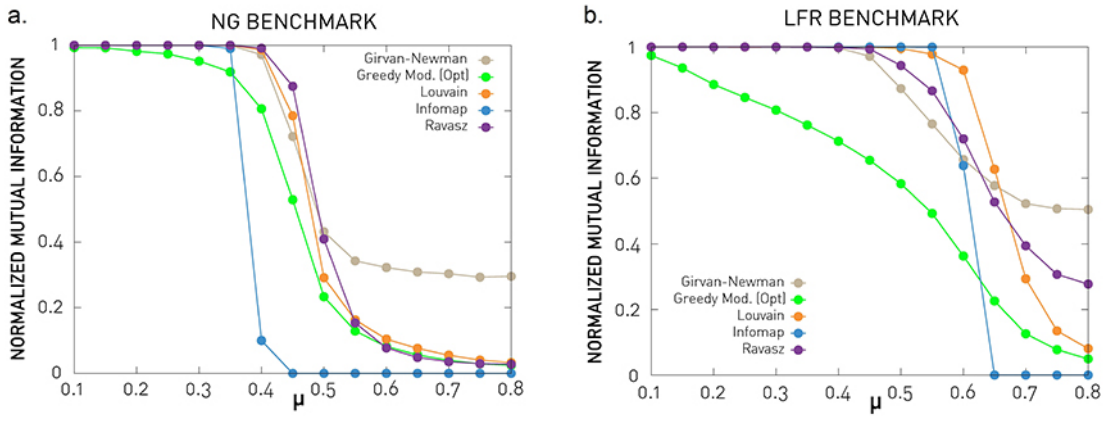


Figura 10 – Figura retirada do livro (BARABáSI; PóSFAI, 2016)

4 Conclusão

Comunidades desempenham um papel muito importante em várias áreas de estudo, enquanto existir uma rede, haverá algum tipo de comunidade na mesma. Vários estudos se utilizam de métodos como a de Louvain para problemas reais, um exemplo são as redes sociais, que tiveram um notável aumento em escala e alcance nos últimos anos, o que gerou muitos desafios em termos de projeto e manutenção do sistema (podemos citar o Twitter), esse é um caso que envolve o problema de particionamento (PUJOL; ERRAMILLI; RODRIGUEZ, 2009).

Em redes sociais, comunidades possuem um papel particularmente importante, pois são mais simples de identificar. Colaboradores de uma empresa são mais propensos a interagirem com seus colegas de trabalho do que com os funcionários de outra empresa, tornando uma comunidade densamente conectada. Isso se aplica a vários outros aspectos, não apenas no ambiente corporativo, mas também em amizades, gostos, localização, entre outros.

Na área médica, comunidades também têm bastante influência, pois colaboram para o entendimento das doenças humanas. Através das comunidades, foi possível descobrir que proteínas envolvidas em uma mesma doença tendem a interagirem entre si. Isso possibilitou a inspiração da hipótese do módulo da doença, que diz que cada doença pode estar ligada a uma vizinhança bem definida na rede celular. Além disso, também foi possível compreender como as funções biológicas são codificadas em redes celulares. Ravaz e seus colaboradores (RAVASZ et al., 2002) estudaram e testaram como grupos de moléculas formam módulos funcionais para realizar funções específicas em redes metabólicas (HARTWELL et al., 1999), e isso foi feito por meio de um algoritmo que identifica clusters de moléculas que formam comunidades localmente densas.

A detecção de comunidades tem grandes aplicações para o estudo, entendimento e avaliação em redes grandes e complexas. Ela usa conceitos como arestas em grafos, sendo bastante adequada para análise de redes. Vários algoritmos de detecção de comunidades foram estudados e propostos no decorrer do tempo, onde cada qual possui suas vantagens e desvantagens, isso dependendo bastante do problema e da natureza da rede.

Referências

BARABáSI, A.-L.; ALBERT, R. Emergence of scaling in random networks. *Science*, v. 286, n. 5439, p. 509–512, 1999. Disponível em: <https://www.science.org/doi/abs/10.1126/science.286.5439.509>. 27

BARABáSI, A.-L.; PóSFAI, M. *Network science*. Cambridge: Cambridge University Press, 2016. ISBN 9781107076266 1107076269. Disponível em: <http://barabasi.com/networksciencebook/>. 11, 12, 15, 19, 28, 29, 30

BLONDEL, V. et al. Fast unfolding of communities in large networks. *J. Stat. Mech*, p. P10008, 2008. 6

CLAUSET, A.; NEWMAN, M. E. J.; MOORE, C. Finding community structure in very large networks. *Physical Review E*, American Physical Society (APS), v. 70, n. 6, dec 2004. Disponível em: <https://doi.org/10.1103%2Fphysreve.70.066111>. 6

DENDROGRAMA. <https://support.minitab.com/pt-br/minitab/18/help-and-how-to/modeling-statistics/multivariate/how-to/cluster-observations/interpret-the-results/all-statistics-and-graphs/dendrogram/>. 16

DERITEI, D. et al. Community detection by graph voronoi diagrams. *New Journal of Physics*, IOP Publishing, v. 16, n. 6, p. 063007, jun 2014. Disponível em: <https://doi.org/10.1088/1367-2630/16/6/063007>. 29

FORTUNATO, S.; CASTELLANO, C. Community structure in graphs. *Encyclopedia of Complexity and System Science*, 01 2008. 26

GIRVAN, M.; NEWMAN, M. E. J. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, v. 99, n. 12, p. 7821–7826, 2002. Disponível em: <https://www.pnas.org/doi/abs/10.1073/pnas.122653799>. 6, 25

HARTWELL, L. H. et al. From molecular to modular cell biology. *Nature*, v. 402, p. C47–C52, 1999. 31

KRISHNAMURTHY, B.; WANG, J. On network-aware clustering of web clients. *SIGCOMM Comput. Commun. Rev.*, Association for Computing Machinery, New York, NY, USA, v. 30, n. 4, p. 97–110, aug 2000. ISSN 0146-4833. Disponível em: <https://doi.org/10.1145/347057.347412>. 6

LANCICHINETTI, A.; FORTUNATO, S.; RADICCHI, F. Benchmark graphs for testing community detection algorithms. *Phys. Rev. E*, American Physical Society, v. 78, p. 046110, Oct 2008. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevE.78.046110>. 27

LIU, X.; CHENG, H.-M.; ZHANG, Z.-Y. Evaluation of community detection methods. *IEEE Transactions on Knowledge and Data Engineering*, v. 32, n. 9, p. 1736–1746, 2020.

NEWMAN, M. E. J. Fast algorithm for detecting community structure in networks. *Physical Review E*, American Physical Society (APS), v. 69, n. 6, jun 2004. Disponível em: <https://doi.org/10.1103%2Fphysreve.69.066133>. 6

NEWMAN, M. E. J. *Networks: an introduction*. Oxford University Press, 2010. ISBN 9780199206650 0199206651. Disponível em: <http://www.amazon.com/Networks-An-Introduction-Mark-Newman/dp/0199206651/ref=sr_1_5?ie=UTF8&qid=1352896678&sr=8-5&keywords=complex+networks>. 13

NEWMAN, M. E. J.; GIRVAN, M. Finding and evaluating community structure in networks. *Phys. Rev. E*, American Physical Society, v. 69, p. 026113, Feb 2004. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevE.69.026113>. 6, 19

PARTHASARATHY, D.; SHAH, D.; ZAMAN, T. *Leaders, Followers, and Community Detection*. arXiv, 2010. Disponível em: <https://arxiv.org/abs/1011.0774>. 6

PERKINS, C. E. Ad hoc networking. capítulo ad hoc networking: an introduction. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA., p. 1–28, 2001. 5

PONS PASCAL; LATAPY, M. Comunidades de computação em grandes redes usando caminhadas aleatórias"(pdf). *Journal of Graph Algorithms and Applications*, p. 191–218, 2006. 7

PONVENI, P.; VISUMATHI, J. A review on community detection algorithms and evaluation measures in social networks. *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, v. 1, p. 1892–1899, 2021. 18, 22

PUJOL, J. M.; ERRAMILLI, V.; RODRIGUEZ, P. *Divide and Conquer: Partitioning Online Social Networks*. arXiv, 2009. Disponível em: <https://arxiv.org/abs/0905.4918>. 31

RAVASZ, E. et al. Hierarchical organization of modularity in metabolic networks. *Science*, American Association for the Advancement of Science (AAAS), v. 297, n. 5586, p. 1551–1555, aug 2002. Disponível em: <https://doi.org/10.1126%2Fscience.1073374>. 31

REDDY, P. K. et al. A graph based approach to extract a neighborhood customer community for collaborative filtering. In: BHALLA, S. (Ed.). *Databases in Networked Information Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002. p. 188–200. ISBN 978-3-540-36233-3. 6

SHANNON, C. E. A mathematical theory of communication. *The Bell System Technical Journal*, v. 27, n. 3, p. 379–423, 1948. 29

WAKITA KEN; TSURUMI, T. Encontrando estrutura comunitária em redes sociais em mega-escala. 2007. 7

ZHANG, J. et al. Research review on algorithms of community detection in complex networks. *Journal of Physics: Conference Series*, IOP Publishing, v. 1069, p. 012124, aug 2018. Disponível em: <https://doi.org/10.1088/1742-6596/1069/1/012124>.

ZHANG, P. Evaluating accuracy of community detection using the relative normalized mutual information. *Journal of Statistical Mechanics: Theory and Experiment*, IOP Publishing, v. 2015, n. 11, p. P11006, nov 2015. Disponível em: <https://doi.org/10.1088%2F1742-5468%2F2015%2F11%2Fp11006>. 22