

UMA INTRODUÇÃO À ANÁLISE DE REDES COMPLEXAS

Leandro Chaves Rêgo

Departamento de Estatística e Matemática Aplicada, Universidade Federal do Ceará
Campus do Pici, Bloco 910, Fortaleza, CE, CEP: 60455-760
Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Pernambuco
Av. Prof. Moraes Rego, 1235. Cidade Universitária, Recife - PE CEP: 50670-901
leandro@dema.ufc.br

Jesus Ossian da Cunha Silva

Departamento de Estatística e Matemática Aplicada, Universidade Federal do Ceará
Campus do Pici, Bloco 910, Fortaleza, CE, CEP: 60455-760
jesus.ossian@dema.ufc.br

Pablo Ignacio Fierens

Instituto Tecnológico de Buenos Aires (ITBA)
Iguazú 341, 1437, Ciudad de Buenos Aires, Argentina
Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
Ciudad de Buenos Aires, Argentina
pfierens@itba.edu.ar

RESUMO

O estudo de redes complexas envolve a análise do relacionamento de vários elementos de um sistema. Esse estudo utiliza conceitos de Teoria dos Grafos, que é uma área da Pesquisa Operacional, na qual o relacionamento entre os elementos de um sistema, chamados nós ou vértices, são modelados por arestas que são pares ordenados de vértices. Uma das aplicações de redes complexas tem sido o estudo de redes sociais, em que os vértices correspondem a indivíduos e as relações podem ser relações de amizade, de trabalho, de interesse mútuo, etc. Além disso, as redes complexas também têm sido utilizadas para análise de séries temporais em vários contextos. Neste minicurso, pretendemos discutir as principais aplicações de Redes Complexas, suas propriedades estruturais e apresentar as principais métricas globais e locais utilizadas em Redes Complexas. Os conceitos serão ilustrados através de exemplos computacionais usando a linguagem Python e o pacote NetworkX.

PALAVRAS CHAVE. Redes Sociais, Métricas Globais, Métricas de Centralidade.

Tópicos: Teoria e Algoritmos em Grafos, Big Data e Analytics

ABSTRACT

The study of complex networks involves analyzing the relationships of various elements of a system. This study uses concepts from Graph Theory, which is an area of Operations Research, in which the relationship between the elements of a system, called nodes or vertices, are modeled by edges that are ordered pairs of vertices. One of the applications of complex networks has been the study of social networks, in which the vertices correspond to individuals and the relationships

can be relationships of friendship, work, mutual interest, etc. Furthermore, Complex Networks have also been used for time series analysis in various contexts. In this short course, we intend to discuss the main applications of Complex Networks, their structural properties and present the main global and local metrics used in Complex Networks. The concepts will be illustrated through computational examples using the Python language and the NetworkX package.

KEYWORDS. Social Networks. Global Metrics. Centrality Metrics.

Paper topics: Theory and Algorithms in Graphs. Big Data and Analytics

1. Introdução e motivação

As redes complexas são modelos utilizados para representar relações entre pares de elementos de um certo grupo de interesse. O grupo de interesse está fortemente ligado às aplicações. Por exemplo, a Internet é uma rede em que o grupo de interesse são computadores ligados por conexões de dados; a *World Wide Web* é uma rede em que o grupo de interesse são as páginas da internet e as ligações entre as páginas são feitas por meio dos links que direcionam uma página a outras; em uma rede de comércio internacional, o grupo de interesse são países e as relações entre esses países está ligada às transações comerciais entre eles durante um certo período; em uma rede de amizades no *Facebook* ou de seguidores no *Instagram*, o grupo de interesse são os perfis da plataforma da rede social e as relações entre os perfis se dão por serem amigos (*Facebook*) ou seguidores (*Instagram*) na plataforma; em uma rede de cadeia alimentar ecológica, o grupo de interesse são as espécies de animais e as relações estão relacionadas a quais espécies se alimentam uma das outras; em uma rede neural os elementos de interesse são os neurônios e as ligações estão relacionadas a como os neurônios se conectam entre si.

O estudo de redes complexas consiste no estudo do padrão em que os elementos do grupo de interesse se conectam entre si. As redes complexas se caracterizam pelo conjunto das relações não apresentarem um padrão simples como o que ocorre no caso em que todos os pares de elementos se relacionam entre si, nem também se assemelham a relações que sejam formadas aleatoriamente.

Existem várias aplicações relacionadas ao estudo de redes complexas. O estudo de redes complexas pode ajudar a entender como notícias se espalham, como doenças se propagam, quem são os indivíduos mais importantes em uma rede de criminalidade, para identificar que produtos as pessoas tendem a comprar ou como elas tendem a votar. Existem estudos que também utilizam redes complexas para analisar séries temporais com o objetivo de identificar padrões de anomalias em sinais biológicos [Wang et al., 2016], verificação de assinaturas manuscritas online [Santos et al., 2022] ou estudar como variáveis macroeconômicas podem influenciar na estrutura de redes formadas por séries temporais de preços de ativos em bolsas de valores [de Pontes e Rêgo, 2022].

Existem várias técnicas que podem ser utilizadas para entender o padrão de relações em uma rede complexa. Um dos primeiros passos seria visualizar a rede complexa para identificar padrões que não são fáceis de observar diretamente dos dados brutos. A Figura 1 ilustra um exemplo de uma rede social dos terroristas envolvidos no ataque de 11 de setembro de 2001. As cores identificam o voo em que os terroristas se encontravam e as ligações refletem o fato de terroristas terem tido algum contato prévio, como um treinamento em conjunto, antes do ataque [Lindelauf et al., 2013].

Contudo, a visualização só permite a identificação de padrões se existirem poucos elementos na rede ou poucas relações entre esses elementos. Para análise de redes maiores, existem várias métricas que auxiliam tanto na identificação de padrões da rede como um todo, como informações sobre a posição topológica de elementos individuais da rede. Os nós mais importantes

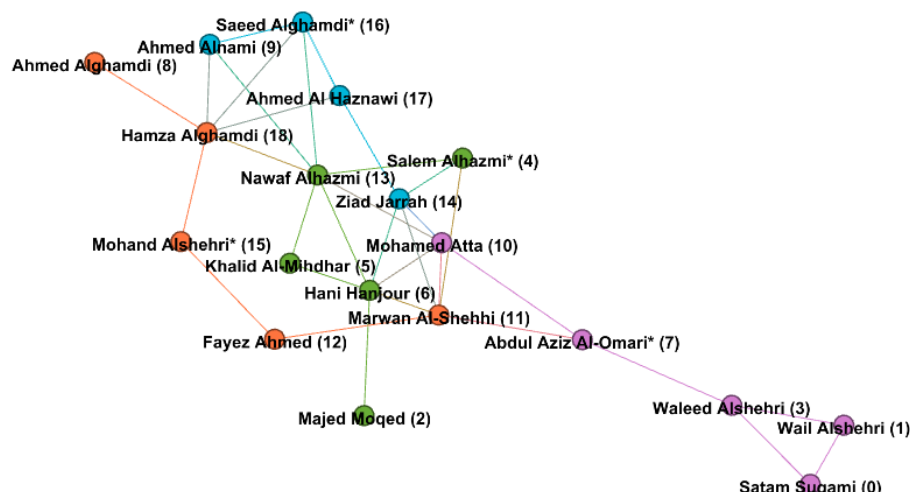


Figura 1: Rede complexa dos terroristas envolvidos no ataque de 11 de setembro de 2001. Fonte: Lindelauf et al. [2013]

de uma rede podem ser identificados com o objetivo, por exemplo, de desestabilizar uma rede de criminalidade, dificultar a transmissão de doenças ou de *fake news* em mídias sociais.

O restante destas notas de aula está organizado da seguinte maneira. Na Seção 2, fazemos uma breve introdução à Teoria dos Grafos que é o modelo matemático que é utilizado para o estudo de redes complexas. Ainda nesta seção, apresentamos o pacote Networkx (<https://networkx.org/>) do Python, que será utilizado para ilustrar computacionalmente os conceitos desenvolvidos neste trabalho. Todos os códigos encontram-se disponíveis no seguinte repositório: https://github.com/jesusossian/minicurso_sbpo2024. Na Seção 3, descrevemos sobre algumas das principais métricas globais, que são utilizadas para caracterizar uma rede complexa como um todo. Na Seção 4, algumas das principais métricas de centralidade, que descrevem a importância dos elementos do grupo de interesse da rede. Na Seção 5, outras métricas que mensuram o grau de coesão e homofilia da rede são apresentadas. Por fim, encerramos este trabalho com considerações finais na Seção 6.

2. Grafos

Através da teoria dos grafos podemos formalizar matematicamente o estudo das redes complexas. Um grafo é um objeto matemático que consiste de um par ordenado de conjuntos $G = (V, E)$, em que V é chamado de conjunto de vértices ou nós do grafo, que representam os elementos que se relacionam em uma rede complexa, e $E \subseteq V \times V$ é um conjunto de pares ordenados de vértices, chamados de arestas, que representam as relações entre os elementos da rede complexa. Dependendo do contexto da rede complexa, os vértices podem receber outros nomes, como atores e sítios, enquanto as arestas podem ser chamadas de arcos, ligações, relações. Usaremos a nomenclatura de grafos e redes para denotar a mesma coisa neste texto.

Se em um grafo $G = (V, E)$, $(u, v) \in E$ implica que $(v, u) \in E$, então dizemos que o grafo é não direcionado ou não dirigido, caso contrário ele é chamado de direcionado, dirigido ou um digrafo. Em grafos não direcionados, (u, v) e (v, u) representam uma só aresta. Grafos não direcionados são úteis para representar redes complexas em que as relações são recíprocas, como por exemplo redes de amigos ou redes de transporte. Por outro lado, grafos direcionados são utilizados para modelar redes complexas em que não necessariamente as relações são recípro-

cas, como por exemplo, a *World Wide Web* ou redes ecológicas de predador-presa. Em um grafo direcionado, a aresta (u, v) é dita ser uma aresta de saída de u e de entrada em v .

Uma aresta que liga um vértice de um grafo a ele mesmo é chamada de laço. A maioria das redes complexas que estudaremos não possuem nem laços nem múltiplas arestas ligando o mesmo par de vértices. Grafos que não possuem laços nem múltiplas arestas são chamados de grafos simples e as redes correspondentes são chamadas de redes simples.

Uma maneira alternativa de representar um grafo simples é através de sua matriz de adjacência, A . A matriz de adjacência de um grafo simples em que $V = \{1, 2, \dots, n\}$ tem seu elemento (i, j) dado por¹:

$$A(i, j) = \begin{cases} 1, & \text{se } (i, j) \in E, \\ 0, & \text{caso contrário.} \end{cases} \quad (1)$$

Deste modo, a matriz de adjacência de grafos não direcionados é simétrica em relação à diagonal principal.

Se em um grafo existir um laço ligando um nó i a ele mesmo, então o valor do elemento (i, i) da matriz de adjacência é igual a 2 se o grafo for não direcionado e igual a 1 se o grafo for direcionado. A justificativa para esta definição é que como em um grafo não direcionado uma aresta envolvendo os nós i e j aparece duas vezes na matriz de adjacência tornando $A(i, j) = A(j, i) = 1$, o laço em um grafo não direcionado deve ser representado em duplicidade.

Em redes complexas, algumas relações possuem uma intensidade maior que outras. Por exemplo, em redes de comércio internacional, a intensidade das relações pode estar relacionada ao volume das transações comerciais entre os países, em uma relação de amizade pode estar relacionada a frequência de encontros entre os amigos, em relação a rede de transportes aéreos pode estar relacionada ao número de passageiros transportados entre dois aeroportos em um certo período de tempo. Para representar essas redes, são usadas redes ponderadas, que possuem pesos nas arestas. Neste caso, utilizaremos uma matriz de adjacência ponderada, W , de modo que $A(i, j) = 0$ se e somente se $W(i, j) = 0$. Neste caso, a rede é não direcionada se $W(i, j) = W(j, i)$, para todo $i, j \in V$. Quanto maior $|W(i, j)|$, maior a intensidade da relação entre os nós que pode ser uma relação positiva ou negativa. Na maioria dos casos, os pesos são positivos, mas existem situações em que podem ser usados pesos negativos, como por exemplo em uma rede social as ligações com pesos positivos entre pessoas representam as amizades e as com pesos negativos as inimizades.

2.1. Passeios, Caminhos e Ciclos

Um passeio (em inglês, *walk*) em uma rede, $G = (V, E)$, consiste de uma sequência ordenada de vértices $(i_0, i_1, i_2, \dots, i_k)$ tal que $(i_{j-1}, i_j) \in E$ para $j = 1, 2, \dots, k$. Neste caso, k é o comprimento do passeio. O número de passeios de comprimento k entre i e j é dado por $A^k(i, j)$.

Se um passeio não contém arestas repetidas, ele é conhecido como um caminho (em inglês, *path*). (Recorde que (i, j) e (j, i) representam uma mesma aresta em uma rede não dirigida.) Um caminho sem vértices repetidos (exceto, possivelmente, pelo primeiro e último da sequência que podem ser iguais) é chamado de caminho simples. Um caminho de menor comprimento entre dois vértices i e j é chamado de caminho geodésico e o comprimento do caminho geodésico é conhecido como distância geodésica, denotada por $d(i, j)$. Na literatura de redes complexas, é comum encontrar o termo caminho (*path*) utilizado como sendo um passeio qualquer também.

¹Utilizamos a convenção de que o elemento (i, j) da matriz de adjacência é igual a 1 quando existe uma aresta saindo i em direção a j . Em algumas referências, como em Newman [2018], a convenção é oposta.

No caso de redes ponderadas com pesos não negativos, como definimos que quanto maior o peso $W(i, j)$ mais intensa é a relação entre i e j , significa que eles são mais próximos. Deste modo, o comprimento de um passeio ou de um caminho é dado pela soma dos recíprocos dos pesos das arestas. Então, a distância geodésica ponderada entre os nós i e j é dada pela menor comprimento de um caminho que liga o nó i ao nó j , $d^w(i, j)$. Em alguns pacotes computacionais, será necessário entrar com o recíproco dos pesos para que as distâncias sejam calculadas corretamente.

Se em um caminho $i_0 = i_k$, este caminho é conhecido como um ciclo. Se o caminho for simples e $i_0 = i_k$, temos um ciclo simples. Note que todo ciclo que não é um ciclo simples, tem algum vértice intermediário igual ao primeiro (que é igual ao último) vértice da sequência, ou possui dois vértices intermediários iguais e, portanto, tem uma subsequência de vértices do ciclo que forma um ciclo simples. Na literatura de redes complexas, as vezes o termo ciclo é utilizado no lugar de ciclo simples.

Redes que não contém ciclos são chamadas de acíclicas. Na literatura de redes complexas, é comum chamar redes que não são acíclicas de cíclicas, embora o termo grafo cíclico seja evitado em teoria dos grafos.

Algumas propriedades sobre redes acíclicas [Newman, 2018]:

1. Toda rede acíclica não contém laços, pois um laço é um ciclo de comprimento 1.
2. Toda rede direcionada e acíclica com uma quantidade finita de vértices tem pelo menos um vértice sem arestas que saem dele. Caso contrário, iniciando um passeio em um vértice qualquer e seguindo as arestas de saída, como existe somente um número finito de vértices, necessariamente será necessário repetir algum vértice no passeio, que portanto conterá um ciclo.
3. Se a quantidade de vértices é finita, é sempre possível rotular os vértices de uma rede direcionada acíclica de tal forma que a matriz de adjacência da rede é triangular com todos os elementos da diagonal principal iguais a zero.
4. Uma rede direcionada é acíclica se e somente se todos os autovalores da matriz de adjacência são nulos, ou seja é uma matriz nilpotente.

2.2. Conectividade e Componentes

Um importante conceito em redes é saber se existe um caminho entre dois vértices quaisquer da rede. Em caso positivo, dizemos que a rede é conectada. No caso de redes direcionadas, existem duas versões possíveis a depender se a direção das arestas é levada ou não em consideração ao observar o caminho. Se a rede for conectada levando em consideração a direção das arestas no caminho entre quaisquer dois vértices, diz-se que a rede é fortemente conectada. Enquanto se as direções das arestas não forem levadas em consideração, temos que a rede é fracamente conectada.

Um subgrafo de um grafo $G = (V, E)$ é qualquer grafo $G' = (V', E')$ tal que $V' \subseteq V$ e $E' \subseteq E$. Ou seja, um subgrafo de um grafo consiste de um subconjunto dos vértices e das arestas do grafo original.

Como nem toda rede é conectada, existem alguns subgrafos que são de interesse. Esses subgrafos são grafos conectados tais que não existe nenhum outro subgrafo do grafo original que seja conectado e cujos conjuntos de vértices e arestas contenham os conjuntos de vértices e arestas da componente, respectivamente. Formalmente, $G' = (V', E')$ é uma componente de $G = (V, E)$ se:

1. G' é um subgrafo de G .

2. G' é conectado (no caso de grafos direcionados, existem duas definições de componente a depender da definição de conectividade utilizada).
3. Para todo subgrafo $G'' = (V'', E'')$ conectado de G , temos que $V'' \cap V' = \emptyset$ ou G'' é um subgrafo de G' .

Um vértice que não possui nenhuma aresta envolvendo ele em uma rede não direcionada é considerado uma componente, de modo que todo vértice da rede pertence a somente uma componente. No caso de grafos direcionados, se um vértice tem apenas uma aresta em uma direção envolvendo ele, ele constitui uma componente isolada.

Observe que qualquer componente com pelo menos dois vértices fortemente conectada de um grafo direcionado não é acíclica. Este fato deriva da definição que deve existir um caminho entre esses dois vértices nas duas direções e a junção desses dois caminhos forma um ciclo.

A componente com maior número de vértices em uma rede é chamada de componente gigante ou principal da rede. Muitos estudos de rede abordam apenas o que acontece na componente gigante da mesma. O diâmetro de uma rede é dado pela maior distância geodésica entre dois vértices pertencentes a mesma componente em uma rede. O diâmetro é uma medida do maior afastamento possível entre vértices que são conectados na rede, representando o quão difícil pode ser a influência de algum vértice atingir outro vértice.

Uma aresta é chamada de ponte se ao se remover esta aresta do grafo, o número de componentes aumenta. Arestas que são pontes tem importante função na conectividade de redes.

2.3. Vizinhança

Em uma rede G , a vizinhança, $N_G(i)$, de um vértice i consiste de todos os vértices que são ligados a i . Formalmente:

$$N_G(i) = \{j : A(i, j) = 1\}.$$

Então, a vizinhança de um nó i contém todos os nós que são alcançados por caminhos de tamanho 1 a partir de i . Podemos definir por indução matemática, a k -ésima vizinhança, $N_G^k(i)$ de um nó i como o conjunto de nós que podem ser alcançados por caminhos de comprimento menor ou igual a k a partir de i , ou seja:

$$N_G^k(i) = \bigcup_{j \in N_G(i)} N_G^{k-1}(j).$$

Finalmente, a vizinhança estendida de um nó i da rede consiste de todos os nós que podem ser alcançados por um caminho a partir de i , ou seja, $N_G^{n-1}(i)$, em que n é o número de nós da rede.

Em redes direcionadas, definem-se dois tipos de vizinhança, predecessores e sucessores, de um vértice i dependendo se a aresta é entrante ou saínte em i . Formalmente:

$$P_G(i) = \{j : A(j, i) = 1\}$$

e

$$S_G(i) = \{j : A(i, j) = 1\}.$$

2.4. Formas especiais

Existem algumas redes que possuem uma topologia regular e assumem formas padrão que descreveremos a seguir e estão ilustradas na Figura 3.

2.4.1. Árvore e Floresta

Uma rede não direcionada é uma árvore se ela for conectada e acíclica. Já uma rede direcionada é uma árvore se a rede não direcionada em que se ignora a direção das arestas da rede direcionada for uma árvore.

Como uma árvore é acíclica, entre dois vértices quaisquer da árvore, existe um único caminho. Uma floresta é uma rede em que cada componente é uma árvore.

Em uma árvore, vértices que só possuem uma aresta envolvendo eles são chamados de folhas. Toda árvore finita com pelo menos dois vértices, possui pelo menos duas folhas. Caso contrário, iniciando em um dos vértices poderíamos ir fazendo um caminho através de arestas distintas, pois cada vértice teria uma aresta para entrar e outra para sair dele. Porém, como o número de vértices é finito, necessariamente teríamos que formar um ciclo, uma contradição já que a árvore é acíclica.

Pode-se provar que uma rede conectada com n vértices é uma árvore se e somente se ela possui $n - 1$ arestas. Algumas árvores recebem nomes especiais. Uma árvore que só possui duas folhas é uma linha. Uma árvore que possui um vértice que toda aresta envolve este vértice é denominada de estrela e este vértice é denominado de centro da estrela.

2.4.2. Círculo

Um círculo é um grafo conectado que possui um único ciclo no qual todos os vértices estão envolvidos em exatamente duas arestas.

2.4.3. Rede regular

Uma rede regular é uma rede em que cada vértice está envolvido no mesmo número de arestas. Como vimos, em um círculo cada vértice está envolvido em exatamente duas arestas. Então, todo círculo é um grafo regular. Como cada aresta envolve dois vértices, se uma rede regular contém uma quantidade ímpar de vértices, cada vértice deve estar envolvido em uma quantidade par de arestas. A seguir, veremos outro tipo de grafo regular.

2.4.4. Rede completa

Uma rede completa é aquela na qual todas as possíveis arestas entre vértices distintos estão presentes, ou seja, que possui uma única componente e tem diâmetro igual a 1.

2.5. Redes aleatórias

Uma das grandes questões no estudo de redes complexas é explicar porque certos padrões ou topologias de redes surgem. Um modelo bem simples de formação de redes que nos ajudará a compreender melhor algumas características das redes complexas são as redes aleatórias, em particular, o conceito de grafo de Erdős-Rényi ou grafos binomiais.

Para obter uma rede não direcionada aleatória segundo o modelo binomial, fixamos o número de vértices n e cada uma das $\binom{n}{2}$ possíveis arestas tem uma probabilidade $0 < p < 1$ fixa e independente de estar presente na rede aleatória. Com este modelo, uma dada rede não ponderada que possua m arestas e n vértices tem probabilidade $p^m(1 - p)^{\binom{n}{2} - m}$ de ser gerada. Como para valores grandes de n e pequenos de p , uma distribuição binomial pode ser aproximada por uma distribuição Poisson, estas redes também são chamadas de redes aleatórias de Poisson.

Apesar deste tipo de rede aleatória não se aproximar das características usualmente encontradas em redes sociais, como a estrutura probabilística que gera a rede é relativamente simples, não é difícil calcular métricas dessas redes e compará-las com redes complexas reais.

2.6. Análise com Networkx

Embora existam muitos pacotes computacionais para o estudo de redes, nesta seção apresentaremos o Networkx (<https://networkx.org/>), uma biblioteca muito útil do Python. Python é uma linguagem de programação muito popular.

Nesta seção, tomaremos exemplos do livro Sayama [2015]. Um exemplo clássico de rede social é um clube de karate constituído por 34 membros Zachary [1977]. As arestas do grafo representam amizades entre os membros. Este exemplo está incluso no pacote Networkx:

```
1 # importa o pacote Networkx com o pseudônimo nx
2 import networkx as nx
3
4 # gera o grafo do clube de karate e armazena ele em g
5 g = nx.karate_club_graph()
6
7 print('=====')
8 print('V:',end=' ')
9 # imprime uma lista com os nós de g
10 print(g.nodes())
11
12 print('=====')
13 print('E:',end=' ')
14 # imprime uma lista com as arestas de g
15 print(g.edges())
16
17 # imprime o grafo g
18 nx.draw(g,with_labels = True)
```

Este fragmento de código produz os seguintes resultados:

```
1 =====
2 V: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
3     21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33]
4 =====
5 E: [(0, 1), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6), (0, 7), (0, 8), (0, 10), (0,
6     11), (0, 12), (0, 13), (0, 17), (0, 19), (0, 21), (0, 31), (1, 2), (1, 3),
7     (1, 7), (1, 13), (1, 17), (1, 19), (1, 21), (1, 30), (2, 3), (2, 7), (2, 8),
8     (2, 9), (2, 13), (2, 27), (2, 28), (2, 32), (3, 7), (3, 12), (3, 13), (4,
9     6), (4, 10), (5, 6), (5, 10), (5, 16), (6, 16), (8, 30), (8, 32), (8, 33),
10    (9, 33), (13, 33), (14, 32), (14, 33), (15, 32), (15, 33), (18, 32), (18,
11    33), (19, 33), (20, 32), (20, 33), (22, 32), (22, 33), (23, 25), (23, 27),
12    (23, 29), (23, 32), (23, 33), (24, 25), (24, 27), (24, 31), (25, 31), (26,
13    29), (26, 33), (27, 33), (28, 31), (28, 33), (29, 32), (29, 33), (30, 32),
14    (30, 33), (31, 32), (31, 33), (32, 33)]
```

Observe-se que, por exemplo, o par (23, 32) corresponde a uma amizade entre o nó 23 e o nó 32. A lista de arestas pode ser, como estrutura de armazenamento, mais eficiente do que a matriz de adjacência nos casos em que esta seja esparsa. No entanto, a matriz de adjacência pode ser achada usando a função `adjacency_matrix()`.

A Figura 2a mostra uma representação gráfica da rede. Esta representação não é muito clara. Há diversas maneiras de mostrar uma rede, algumas das quais estão incluídas no pacote Networkx. Duas alternativas são apresentadas no próximo exemplo: circular (Fig. 2b) e Kamada-Kawai Kamada e Kawai [1989] (Fig. 2c).

```
1 # importa o pacote Networkx com o pseudônimo nx
2 import networkx as nx
```

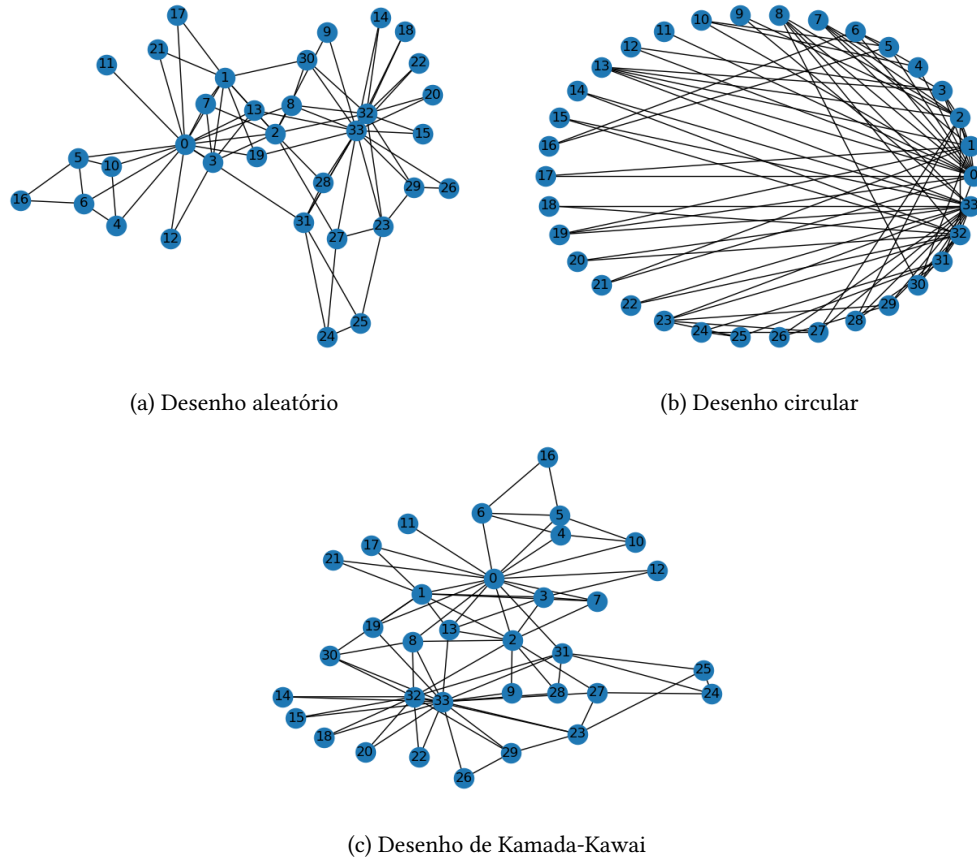



Figura 2: Rede de amizades do clube de karate

```
3
4 nx.draw_circular(g, with_labels = True)
5 nx.draw_kamada_kawai(g, with_labels = True)
6
7 # imprime os vizinhos do nó 12
8 for n in g.neighbors(12):
9     print(n)
```

Este código produz o resultado (cf. Fig. 2a)

```
1 0
2 3
```

Além de exemplos como o karate club, Networkx também pode gerar grafos de diversos tipos. Alguns casos já mencionados: árvore, círculo, grafo regular, grafo completo, grafo aleatório de Erdős-Rényi. O seguinte código mostra a geração destes grafos (observe Fig. 3):

```
1 # importa o pacote Networkx com o pseudônimo nx
2 import networkx as nx
3
4 # gera uma árvore binomial
5 gab = nx.binomial_tree(4)
6
```

```

7 # gera um círculo com 6 nós
8 gci = nx.circulant_graph(6, [1])
9
10 # gera um grafo regular com 6 nós e 3 arestas por nó
11 gre = nx.random_regular_graph(3, 6)
12
13 # gera um grafo completo com 6 nós
14 gco = nx.complete_graph(6)
15
16 # gera um grafo aleatório de Erdős-Rényi com 6 nós e probabilidade 0.25
17 ger = nx.gnp_random_graph(6, 0.25)

```

Networkx também permite construir grafos *ad hoc*. Um exemplo de grafo direcionado:

```

1 # importa o pacote Networkx com o pseudônimo nx
2 import networkx as nx
3
4 # gera um grafo direcionado e armazena ele em gfa
5 gfa = nx.DiGraph()
6
7 # adiciona nós ao grafo gfa
8 gfa.add_nodes_from(['Avó', 'Avô', 'Pai', 'Mãe', 'Filho', 'Filha'])
9
10 # adiciona arestas ao grafo gfa
11 gfa.add_edge('Avó', 'Pai') #Avó -> Pai
12 gfa.add_edge('Avô', 'Pai') #Avô -> Pai
13 gfa.add_edge('Pai', 'Filho') #Pai -> Filho
14 gfa.add_edge('Pai', 'Filha') #Pai -> Filha
15 gfa.add_edge('Mãe', 'Filho') #Mãe -> Filho
16 gfa.add_edge('Mãe', 'Filha') #Mãe -> Filha

```

Às vezes, é necessário criar um grafo partindo de uma lista de arestas em um arquivo de texto. Por exemplo, considere o seguinte arquivo `fami.txt`:

```

1 Avó Pai 5
2 Avô Pai 2
3 Pai Filho 4
4 Pai Filha 6
5 Mãe Filho 9
6 Mãe Filha 7

```

Cada linha do arquivo corresponde a uma aresta. Por exemplo, a linha 1 corresponde a uma aresta da Avó ao Pai. O terceiro elemento na linha é uma propriedade da aresta, como ser a força da ligação. O seguinte código em Python carrega este arquivo e mostra a rede associada (ver Fig. 4).

```

1 # importa o pacote Networkx com o pseudônimo nx
2 import networkx as nx
3
4 # gera um grafo direcionado de acordo com os dados armazenados em fami.txt
5 gfa = nx.read_edgelist("fami.txt",
6                       data= (('peso', float),),
7                       create_using = nx.DiGraph)
8
9 plt.figure(frameon=False)
10 pos=nx.circular_layout(gfa)
11 pos['Avó'] = [-1, 0]
12 pos['Avô'] = [+0, 0]

```

```

13 pos['Pai'] = [-0.5, -0.5]
14 pos['Mãe'] = [+1.5, -0.5]
15 pos['Filho'] = [+0.0, -1.0]
16 pos['Filha'] = [+1.0, -1.0]
17 larg = [(0.5*gfa[u][v]['peso']) for u,v in gfa.edges]
18
19 # imprime o grafo gfa
20 nx.draw_networkx(gfa, pos=pos,
21                  node_size=2000, with_labels=True,
22                  arrows=True, width=larg)
23 plt.box(False)

```

A função `read_edgelist()` carrega o arquivo como uma rede de Networkx. Observe que o parâmetro `create_using` determina o tipo de grafo, neste caso um grafo direcionado `DiGraph()`. O parâmetro `data` nomeia o atributo e indica que é de tipo numérico (ponto flotante). A função `circular_layout()` retorna as posições dos nós em um círculo. Porém, estas posições são alteradas para aprimorar a figura. Finalmente, a função `draw_networkx()` cria a figura. O tamanho dos nós é determinado pelo parâmetro `node_size` e suas posições pelo parâmetro `pos`. O argumento `with_labels` determina se os nomes dos nós são apresentados. Da mesma maneira, `arrows` determina se as flechas são traçadas. A largura das arestas é determinada pelo argumento `width` que, neste exemplo, foi feito proporcional ao atributo.

3. Métricas Globais de Redes

Em redes pequenas, é possível identificar padrões inspecionado visualmente a mesma. Contudo, em redes grandes a visualização não é suficiente para encontrarmos características importantes das mesmas. Para tal, é importante definir métricas que descrevam globalmente esta

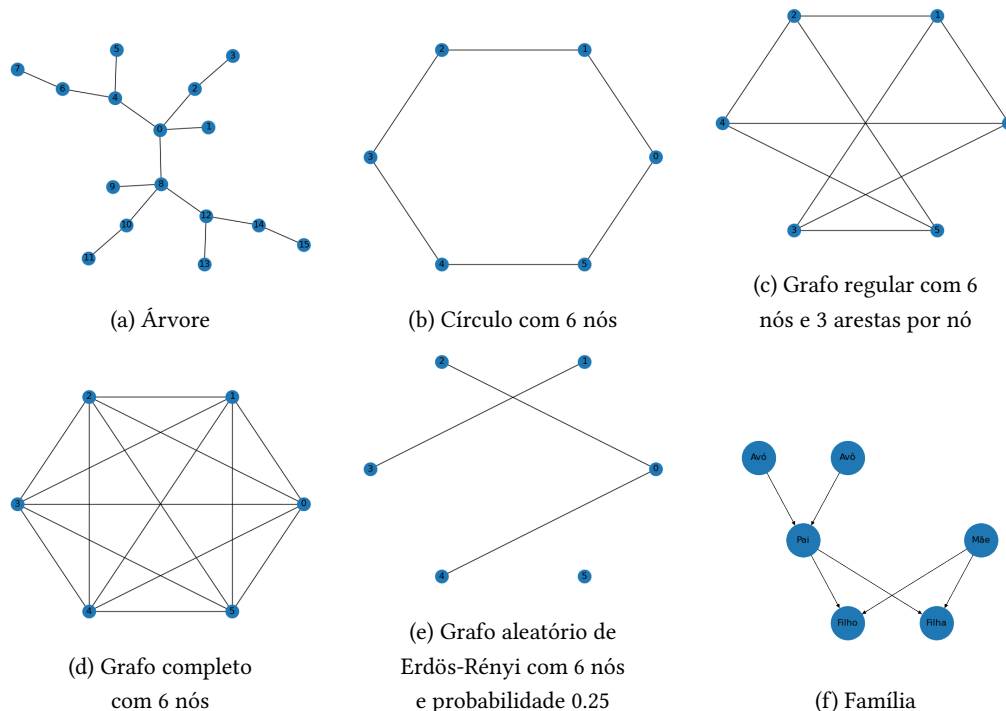


Figura 3: Exemplos de grafos gerados por Networkx

rede. Uma delas nós já vimos que é o diâmetro da rede. Veremos agora outras métricas globais de rede.

3.1. Densidade

Seja uma rede $G = (V, E)$, com $|V| = n$ e m arestas. Então, a densidade da rede é:

$$\text{dens}(G) = \begin{cases} \frac{2m}{n(n-1)}, & \text{se a rede for não direcionada, e} \\ \frac{m}{n(n-1)}, & \text{se a rede for direcionada.} \end{cases}$$

Intuitivamente, a densidade de uma rede revela qual a proporção de potenciais arestas que poderiam estar presentes na rede, sem considerar arestas múltiplas ou laços.

A função `density` do `Networkx` devolve a densidade de uma rede. Por exemplo, o seguinte código

```
1 # importa o pacote Networkx com o pseudônimo nx
2 import networkx as nx
3
4 # gera o grafo do clube karate
5 g = nx.karate_club_graph()
6
7 # gera um grafo regular com 6 nós e 3 arestas por nó
8 gre = nx.random_regular_graph(3, 6)
9
10 # gera um grafo completo com 6 nós
11 gco = nx.complete_graph(6)
12
13 # imprime a densidade dos grafos g, gre e gco
14 print("dens(g)=", nx.density(g))
15 print("dens(gre)=", nx.density(gre))
16 print("dens(gco)=", nx.density(gco))
```

retorna os resultados

```
1 dens(g) = 0.13903743315508021
2 dens(gre) = 0.6
3 dens(gco) = 1.0
```

No caso da rede completa, claramente, a densidade é 1.0. No caso da rede regular com seis nós e três arestas por nó, o número total de arestas é nove (veja Fig. 3). Portanto, a densidade é $(2 \times 9)/(6 \times 5) = 0.6$.

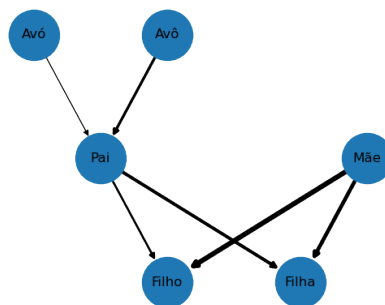


Figura 4: Família: exemplo de um grafo direcionado com arestas de diferentes pesos.

3.2. Comprimento médio dos caminhos

Vimos que o diâmetro de uma rede é igual à maior distância geodésica entre dois nós em uma mesma componente da rede. Similarmente, o comprimento médio dos caminhos de uma rede é dado pela média de todas as distâncias geodésicas entre dois nós que são conectados na rede.

Networkx possui um algoritmo que calcula os caminhos geodésicos entre dois nós. Por exemplo, o caminho mais curto e a distância geodésica entre os nós 3 e 20 na rede de amigos do clube de karate podem ser calculados

```
1 # importa o pacote Networkx com o pseudônimo nx
2 import networkx as nx
3
4 # gera o grafo do clube karate
5 g = nx.karate_club_graph()
6
7 # calcula o menor caminho de 3 a 20
8 caminho = nx.shortest_path(g, 3, 20)
9
10 # calcula o tamanho do menor caminho de 3 a 20
11 longitude = nx.shortest_path_length(g, 3, 20)
12
13 print("menor caminho de 3 a 20 = ", caminho)
14 print("tamanho do menor caminho de 3 a 20 =", longitude)
```

retornando os resultados

```
1 menor caminho de 3 a 20 = [3, 2, 32, 20]
2 tamanho do menor caminho de 3 a 20 = 3
```

O comprimento médio dos caminhos geodésicos e o diâmetro também são calculados por algoritmos do Networkx:

```
1 # importa o pacote Networkx com o pseudônimo nx
2 import networkx as nx
3
4 # Comprimento médio dos caminhos geodesicos de g
5 cm = nx.average_shortest_path_length(g)
6
7 # Diâmetro de g
8 di = nx.diameter(g)
9
10 print("cm =", cm)
11 print("di =", di)
```

O código acima retorna a saída:

```
1 cm = 2.408199643493761
2 di = 5
```

3.3. Distribuição de Graus

O grau de um certo vértice de uma rede não direcionada consiste de quantas arestas envolvem este vértice na rede, ou seja

$$d(i) = \sum_{j \in V} A(i, j).$$

Distribuição de graus

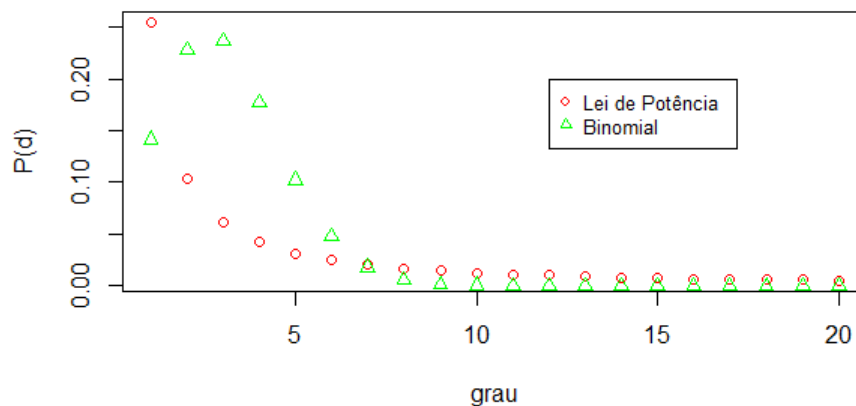


Figura 5: Distribuição dos Graus de uma rede aleatória ($n = 31$ e $p = 0,1$) e de uma rede lei de potência ($\alpha = 1,3$).

Distribuição de graus

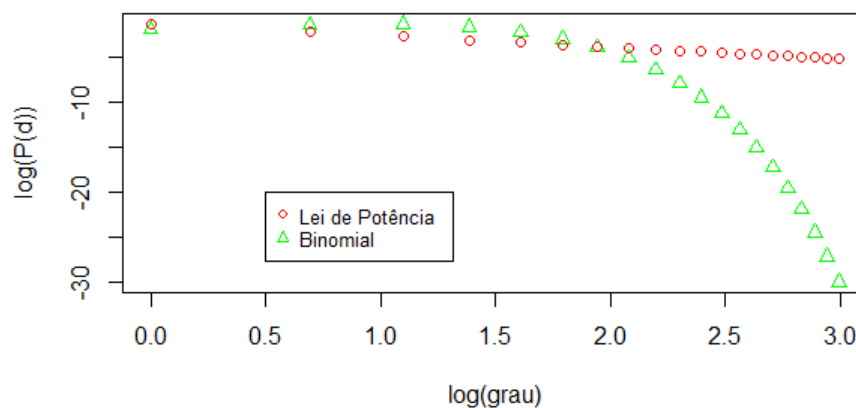


Figura 6: Distribuição dos Graus em escala logarítmica de uma rede aleatória ($n = 31$ e $p = 0,1$) e de uma rede lei de potência ($\alpha = 1,3$).

Então, vértices com alto grau são aqueles que conseguem acessar imediatamente um grande número de vértices. O grau médio da rede é portanto dado por:

$$\bar{d} = \frac{1}{n} \sum_{i \in V} d(i) = \frac{2m}{n} = (n-1) \text{dens}(G).$$

Em redes direcionadas, existem dois tipos de grau que dependem da direção das arestas: o grau de saída (*out-degree*) e o grau de entrada (*in-degree*):

$$d^{out}(i) = \sum_{j \in V} A(i, j)$$

e

$$d^{in}(i) = \sum_{j \in V} A(j, i).$$

Um vértice com grau de saída igual a zero é chamado de sorvedouro. Enquanto, um vértice com grau de entrada nulo, é chamado de fonte. Se em uma rede não direcionada um vértice tiver grau igual a zero, ou se em uma rede direcionada os graus de entrada e saída de um nó forem zero, dizemos que este vértice é isolado e cada vértice isolado constitui uma diferente componente da rede.

Em redes não direcionadas com pesos nas arestas, temos que o grau ponderado de um vértice é dado pela soma dos pesos das arestas que envolvem o vértice, ou seja:

$$d^w(i) = \sum_{j \in V} W(i, j).$$

De forma análoga, os graus de saída e entrada em uma rede direcionada ponderada podem ser definidos.

Uma importante característica das redes é a distribuição de graus dos seus vértices. Para o caso de uma dada rede $G = (V, E)$, a distribuição dos graus é a frequência relativa de ocorrência dos graus nos vértices da rede, ou seja,

$$P(d) = \frac{||V_d||}{n},$$

em que $n = ||V||$, $V_d = \{i \in V : d(i) = d\}$ e $d = 0, 1, \dots, n - 1$.

No caso de uma rede aleatória, a distribuição dos graus é dada pela distribuição de probabilidade de ocorrência dos graus na rede. No caso de uma rede binomial com probabilidade p de ocorrência de cada aresta, a distribuição de graus é dada por:

$$P(d) = \binom{n-1}{d} p^d (1-p)^{n-1-d},$$

em que $n = ||V||$ e $d = 0, 1, \dots, n - 1$.

No caso de uma rede regular, temos que existe algum inteiro k não negativo tal que $P(k) = 1$ e $P(d) = 0$, para todo $d \neq k$.

Em muitas redes complexas, a distribuição dos graus não segue o padrão nem de uma rede regular nem de uma rede aleatória. É comum nessas redes que existam alguns poucos nós com graus bem maiores que a maioria. Neste caso, frequentemente a distribuição dos graus é livre de escala, ou seja, segue uma distribuição de lei de potências:

$$P(d) = cd^{-\alpha},$$

em que c e α são constantes positivas.

A Figura 5 ilustra as distribuições de uma rede aleatória do tipo Binomial e de uma rede livre de escala ou lei de potência. Observe que na lei de potência, a probabilidade de um nó com grau 1 é maior e também as probabilidades dos graus mais elevados são maiores. A cauda da distribuição livre de escala é mais pesada que a binomial. Esta última propriedade pode ser melhor visualizada em um gráfico em escala logarítmica como o da Figura 6. Note que em uma escala

log-log, a distribuição livre de escala se comporta com uma reta com coeficiente angular negativo, que pode portanto, ser estimado por um modelo de regressão linear simples para se determinar o valor de α .

Graus dos nós são devolvidos pela função `degree` do `networkX`:

```
1 # importa o pacote Networkx com o pseudônimo nx
2 import networkx as nx
3
4 # gera o grafo do clube karate
5 g = nx.karate_club_graph()
6
7 # grau do nó 0
8 a = g.degree(0)
9
10 # lista de graus dos nós 0, 1 e 2
11 b = list(g.degree([0,1,2]))
12
13 # lista com as frequências de ocorrência
14 # grau 0, grau 1, etc.
15 c = nx.degree_histogram(g)
16
17 # imprime a, b e c
18 print("a =", a)
19 print("b =", b)
20 print("c =", c)
```

Saída da execução do código:

```
1 a = 16
2 b = [(0, 16), (1, 9), (2, 10)]
3 c = [0, 1, 11, 6, 6, 3, 2, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1]
```

Observe que 11 nós da rede tem grau igual a 2.

3.4. Coesão e Agrupamento

Em geral, redes complexas possuem uma característica de coesão em que nós vizinhos a um certo nó tem uma maior chance de serem vizinhos entre si.

Existem algumas formas de mensurar essa característica. O coeficiente de agrupamento (*clustering* em inglês) total é dado pela proporção de vezes que dois vértices j e k que são vizinhos de um mesmo vértice i também são vizinhos entre si:

$$cl(G) = \frac{\sum_{(i,j,k): j \neq i, k \neq i, k \neq j} A(i,j)A(i,k)A(j,k)}{\sum_{(i,j,k): j \neq i, k \neq i, k \neq j} A(i,j)A(i,k)}.$$

Quando estudarmos métricas de centralidade na Seção 5.1, estudaremos uma métrica local de agrupamento e abordaremos uma outra medida global de agrupamento.

Em redes direcionadas, existem algumas variações de como medir o coeficiente de agrupamento total. A primeira delas, $cl^{d,1}(G)$, é feita ignorando a direção das arestas e calculando este coeficiente como se a rede fosse não direcionada. Uma segunda forma seria vendo a proporção de triplas transitivas que ocorrem. Ou seja, das vezes em que o vértice i está conectado ao j e o j está conectado ao k , quantas vezes o i está ligado ao k diretamente:

$$cl^{d,2}(G) = \frac{\sum_{(i,j,k): j \neq i, k \neq i, k \neq j} A(i,j)A(i,k)A(j,k)}{\sum_{(i,j,k): j \neq i, k \neq i, k \neq j} A(i,j)A(j,k)}.$$

O Exemplo 3.1 ilustra a diferença dos valores das duas formas de calcular o coeficiente de agrupamento total em redes direcionadas.

Exemplo 3.1: Considere a Rede Direcionada ilustrada na Figura 7(a). Para calcular o valor de $cl^{d,1}(G)$ deve-se ignorar a direção das arestas e obter o valor do coeficiente de agrupamento total considerando a Rede Não-Direcionada ilustrada na Figura 7(b). Deste modo, tem-se que

$$cl^{d,1}(G) = \frac{2 + 1 + 2 + 1}{3 + 1 + 3 + 1} = 0.75.$$

Por outro lado, considerando a direção das arestas o valor de $cl^{d,2}(G)$ é:

$$cl^{d,2}(G) = \frac{1 + 0 + 0 + 0}{2 + 1 + 1 + 1} = 0.2.$$

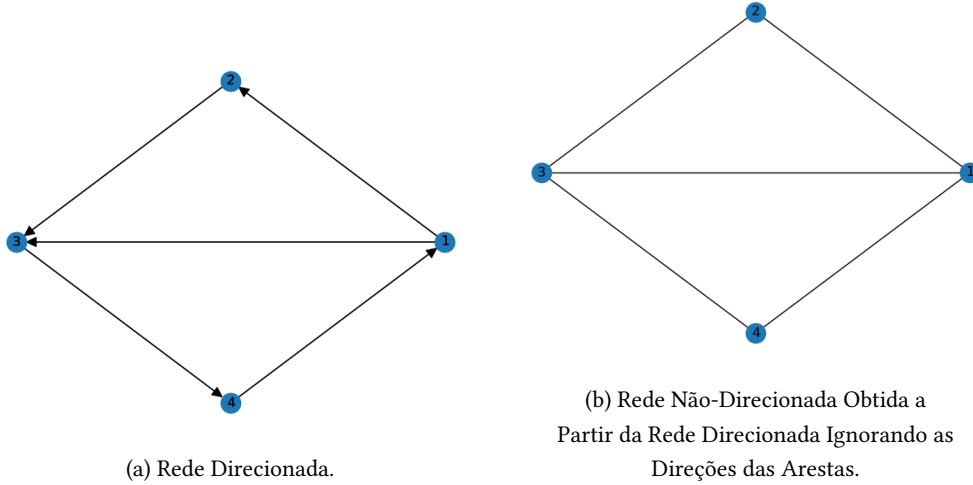


Figura 7: Coeficiente de agrupamento total em redes direcionadas.

Uma das características observadas empiricamente em várias redes complexas é possuir coeficiente de agrupamento maior do que redes aleatórias. Em uma rede aleatória, se o vértice i é ligado aos vértices j e k , como a probabilidade de ocorrências dos links é independente e constante igual a p , temos que o coeficiente de agrupamento deve ser igual a p que também é igual a densidade de rede. Por outro lado, a rede de amigos do clube de Karate de Zachary possui 34 nós e 78 ligações, portanto, uma densidade de 0,139. Já o coeficiente de agrupamento total desta rede é igual a 0,256, que é quase o dobro do esperado em uma rede aleatória de mesma densidade. Em redes maiores, cuja densidade tende a ser bem menor, o coeficiente de agrupamento costuma ser ordens de grandeza maior que a densidade. Este fenômeno é muito comum em redes de coautoria, em que a existência de artigos com 3 ou mais autores já provoca a existência de grupos de nós conectados. Quanto maior o número médio de autores por artigo em uma determinada área científica, maior tende a ser o coeficiente de agrupamento total da rede de coautoria da área.

O coeficiente $cl(\cdot)$ é devolvido pela função `transitivity`:

```

1 # importa o pacote Networkx com o pseudônimo nx
2 import networkx as nx
3
4 # gera o grafo do clube karate
5 g = nx.karate_club_graph()
6
7 # gera o grafo círculo com 6 nós
8 gci = nx.circulant_graph(6, [1])
9
10 # gera o grafo regular com 6 nós e 3 arestas por nó
11 gre = nx.random_regular_graph(3, 6)
12
13 # gera o grafo completo com 6 nós
14 gco = nx.complete_graph(6)
15
16 print("cl(g) =", nx.transitivity(g))
17 print("cl(gci) =", nx.transitivity(gci))
18 print("cl(gre) =", nx.transitivity(gre))
19 print("cl(gco) =", nx.transitivity(gco))

```

Saída da execução do código:

```

1 cl(g) = 0.2556818181818182
2 cl(gci) = 0
3 cl(gre) = 0.3333333333333333
4 cl(gco) = 1.0

```

No caso do círculo, o coeficiente de agrupamento é zero porque o grafo não tem triplas de vizinhos. No caso do grafo completo todos os grupos de três nós estão conectados e, portanto, o coeficiente de agrupamento é 1. O cálculo do coeficiente de agrupamento na rede regular é mais interessante. Considere a rede da Fig. 3c. Dado que cada nó tem três arestas, o número de pares de arestas saindo de um nó também é três. Daí que o denominador do $cl(G)$ na rede regular seja $6 \times 3 = 18$. Por inspeção da Fig. 3c, os grupos de transitivos de três vizinhos (triângulos) são dois, isto é, (0,1,3) e (2,4,5). Cada triângulo é contado três vezes (uma vez por cada nó) e, portanto, o numerador do $cl(G)$ é $2 \times 3 = 6$. Finalmente, $cl(G) = 6/18 = 1/3$.

3.5. Reciprocidade

Em redes direcionadas, uma medida de interesse é saber qual a fração de arestas que ocorrem em ambas as direções, isto é, a reciprocidade é definida por:

$$rc(G) = \frac{\sum_{i,j} A(i,j)A(j,i)}{\sum_{i,j} A(i,j)}.$$

Considere o seguinte código:

```

1 # cria um grafo direcionado gdi
2 gdi = nx.DiGraph()
3
4 # adiciona lista de nós a gdi
5 gdi.add_nodes_from(['A', 'B', 'C'])
6
7 # adiciona arestas a gdi
8 gdi.add_edge('A', 'B') #A -> B
9 gdi.add_edge('B', 'A') #B -> A
10 gdi.add_edge('B', 'C') #B -> C

```

```
11 gdi.add_edge('C', 'A') #C -> A
12
13 # calculo da reciprocidade de gdi
14 re = nx.reciprocity(gdi)
15
16 print("re(gdi) =", re)
```

Saída da execução do código:

```
1 re(gdi) = 0.5
```

A função `reciprocity` do `networkX` devolve a reciprocidade do grafo direcionado. Neste exemplo, o grafo tem 4 arestas e 2 delas são entre os nós A e B. Portanto, a reciprocidade é $2/4 = 0.5$.

3.6. Redes de Mundo Pequeno

Em muitos estudos empíricos envolvendo redes complexas e, em particular, redes sociais, observam-se algumas características típicas de tais estruturas de conexões. Uma das observações típicas diz respeito ao fato de que muitas redes complexas grandes possuem diâmetro e tamanho médio dos caminhos pequenos, o que veio a ser conhecido como redes de mundo pequenos.

Talvez o exemplo de estudo mais famoso seja o de [Milgram \[1967\]](#). Neste experimento, Milgram selecionou pessoas desconhecidas de duas cidades americanas e mandou que certas pessoas mandassem cartas para seus conhecidos próximos, pedindo para que eles repassassem as cartas ou para os destinatários alvo ou para conhecidos que pudessem conhecer os destinatários alvo. Eles apenas sabiam o nome e a região aproximada da localização da residência do destinatário. Como resultado do experimento, aproximadamente $1/4$ das cartas chegaram ao seu destinatário alvo, sendo que o número médio de passos até o destinatário final foi 5 e o maior número de passos observado foi 12. [Jackson \[2010\]](#) observa que este resultado é surpreendente, pois a taxa de resposta em pesquisas voluntárias é de no máximo 30% e o caminho feito pelas cartas não necessariamente seguiu o caminho geodésico da rede de conhecidos.

Supondo que uma pessoa adulta tenha cerca de 100 outras pessoas com as quais ela mantém um contato regular. Então, fazendo uma suposição grosseira de que a rede social de conhecidos seja uma árvore, essa pessoa teria 100^2 conhecidos dos seus conhecidos e 100^3 conhecidos dos conhecidos dos seus conhecidos. Ou seja, em 5 passos atingiria toda a população humana na terra. Se por um lado, a suposição de que a rede é uma árvore gera uma estimativa exagerada para este número, a hipótese de que uma pessoa tem apenas 100 conhecidos está contrabalanceando a estimativa.

As redes de mundo pequeno podem ser observadas por exemplo em redes de coautoria [[Newman, 2004](#)], redes de páginas na internet [[Jin e Bestavros, 2006](#)], redes de voos entre aeroportos [[Guida e Maria, 2007](#)], etc.

3.7. Um exemplo mágico

Com o intuito de ilustrar o cálculo das métricas globais em uma rede complexa real, vamos considerar a rede social dos personagens dos primeiros seis livros de Harry Potter estudada por [Bossaert e Meidert \[2013\]](#). Os dados deste artigo estão disponíveis na Ref. [Snijders](#) e consistem nos seguintes arquivos:

- `hpattributes.txt`: contém atributos de cada personagem, isto é, ano da escola, sexo, e casa de Hogwarts.
- `hpbook1.txt`, `hpbook2.txt`, ..., `hpbook6.txt`: contém as matrizes de adjacência para cada um dos livros. Uma ligação do personagem i com o personagem j indica que i dá apoio (dá suporte emocional) a j . Trata-se de uma rede direcionada.

- `hpnames.txt`: contém os nomes dos personagens.

As primeiras linhas de estes arquivos seguem

[illegible]

Observe que a rede tem laços, isto é, alguns personagens apoiam a si mesmos.

Para trabalhar com arquivos de texto como estes, um pacote muito conveniente é **pandas** [Pandas \[2022\]](#):

```

1 import pandas as pd
2
3 atri = pd.read_csv('data/harrypotter/hpattributes.txt', sep='\t')
4 ares = pd.read_csv('data/harrypotter/hpbook2.txt', sep=' ', header=None)
5 nome = pd.read_csv('data/harrypotter/hpnames.txt', sep='\t')
6
7 print('Palavras chaves de atri:')
8 print(list(atri.columns.values))
9 print('Palavras chaves de nome:')
10 print(list(nome.columns.values))
11
12 print('Valores de atri:')
13 print(atri.values[0:5])
14 print('Valores de nome:')
15 print(nome.values[0:5])
16 print('Valores de ares:')
17 print(ares.values[0:5,0:10])
18
19 print('Casa do terceiro personagem: ', atri['house'][2])
20 print('Nome do terceiro personagem: ', nome['name'][2])

```

Saída da execução do script:

```
1 Palavras chaves de atri:
2 ['id', 'schoolyear', 'gender', 'house']
3 Palavras chaves de nome:
4 ['id', 'name']
```



```

5 Valores de atri:
6 [[ 1 1989 1 4]
7 [ 2 1989 2 1]
8 [ 3 1989 2 1]
9 [ 4 1991 1 3]
10 [ 5 1991 1 4]]
11 Valores de nome:
12 [[1 'Adrian Pucey']
13 [2 'Alicia Spinnet']
14 [3 'Angelina Johnson']
15 [4 'Anthony Goldstein']
16 [5 'Blaise Zabini']]
17 Valores de ares:
18 [[0 0 0 0 0 0 0 0 0]
19 [0 1 1 0 0 0 0 0 0]
20 [0 1 1 0 0 0 0 0 0]
21 [0 0 0 0 0 0 0 0 0]
22 [0 0 0 0 0 0 0 0 0]]
23 Casa do terceiro personagem: 1
24 Nome do terceiro personagem: Angelina Johnson

```

No seguinte fragmento de código, carregamos a informação destes arquivos de texto para construir uma rede direcionada em Networkx. Além disso, produzimos duas figuras da mesma rede, distinguindo nós com cores que representam distintos atributos (ver Fig. 8).

```

1 import networkx as nx
2 import matplotlib.pyplot as plt
3
4 gpotter = nx.DiGraph()
5 n = atri.shape[0]
6
7 # adicionando os nós a rede gpotter
8 for k in range(n):
9     gpotter.add_node(k,nome=nome['name'][k],
10                       ano=atri['schoolyear'][k],
11                       gen=atri['gender'][k],
12                       casa=atri['house'][k])
13
14 # adicionando arestas a rede gpote
15 for k in range(n):
16     for m in range(n):
17         if ares.values[k][m] == 1:
18             gpotter.add_edge(k,m)
19
20 # as casas são nomeadas 1,2,3,4
21 # os gêneros são nomeados 1,2
22 cores = ['red','blue','green','yellow']
23 clh = [cores[gpotter.nodes[i]['casa']-1] for i in range(gpotter.order())]
24 clg = [cores[gpotter.nodes[i]['gen']-1] for i in range(gpotter.order())]
25
26 plt.figure()
27 nx.draw_circular(gpotter,node_color=clh)
28 plt.figure()
29 nx.draw_circular(gpotter,node_color=clg)
30
31 print('Densidade: {:.5f}'.format(nx.density(gpotter)))
32 print('Agrupamento: {:.5f}'.format(nx.transitivity(gpotter)))
33 print('Reciprocidade: {:.5f}'.format(nx.reciprocity(gpotter)))

```

Saída da execução do código:

```
1 Densidade: 0.02728
2 Agrupamento: 0.67611
3 Reciprocidade: 0.65455
```

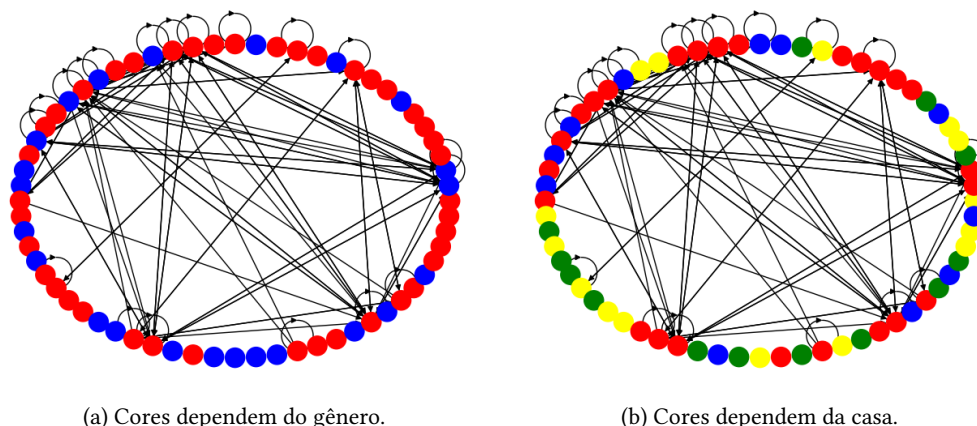


Figura 8: Rede de suporte emocional no segundo livro de Harry Potter.

4. Métricas de Centralidade

Em uma análise de rede complexa, é importante avaliar quem são os principais vértices da rede. Por exemplo, como determinar qual o aeroporto mais importante em uma rede representando a malha aérea de aviões? Como identificar os perfis mais importantes em uma rede social para apresentar um determinado anúncio com o intuito de que ele seja repassado para o maior número de pessoas? Na Seção 3.3, apresentamos uma medida que retrata um certo critério para avaliar a importância ou centralidade de um vértice em uma rede complexa, que foi a centralidade de grau. Como vimos, a centralidade de grau mede quantos nós um dado vértice consegue acessar imediatamente. Como existem vários critérios com os quais podemos avaliar a importância dos nós em uma rede, neste capítulo apresentaremos algumas das principais métricas de centralidade.

As métricas de centralidade fazem parte de um grupo de métricas de redes chamadas métricas locais, que descrevem características individuais dos nós. Além das principais métricas de centralidade, também abordaremos algumas outras métricas locais na próxima seção. Muito do que é discutido aqui, pode ser encontrado em Newman [2018]; Jackson [2010].

4.1. Métricas Geométricas de Centralidade

Um certo grupo de métricas de centralidade que associa a centralidade ou importância de um nó com respeito a distância geodésica deste nó em relação aos demais vértices da rede, de modo que os nós mais centrais sejam os possuem menor distância para os demais. A depender da maneira que sumarmos as distâncias de um nó aos demais, geramos uma medida geométrica de centralidade distinta. Em grafos direcionados, existem duas opções para as definições das métricas geométricas de centralidade que diferem se consideramos as distâncias a partir do nó para o qual estamos calculando a centralidade ou distâncias para chegar nesse nó. Neste texto, iremos descrever de acordo com a primeira opção.

4.1.1. Centralidade de Excentricidade

A excentricidade de um nó é igual à maior distância geodésica do nó aos demais da mesma componente. Formalmente, seja $G_i = (V_i, E_i)$ a componente da rede que contém o nó i . Portanto, a excentricidade do nó i é dada por:

$$ec(i) = \max_{j \in V_i} d(i, j).$$

Para garantir que os nós mais centrais estejam mais próximos dos outros, a centralidade de excentricidade do nó i é dada pelo recíproco da sua excentricidade:

$$C_{ec}(i) = \frac{1}{ec(i)}.$$

Exemplo 4.1.: Em um grafo completo todos os nós possuem excentricidade e centralidade de excentricidade iguais a 1. Em um círculo com $2n$ vértices, todos os nós tem excentricidade igual a n e centralidade de excentricidade igual a $1/n$. Na rede da Figura 3(a), os nós 7 e 15 possuem a maior excentricidade (7) e, conseqüentemente, a menor centralidade de excentricidade que é igual a $1/7$. Por outro lado, os nós 0 e 8 possuem a menor excentricidade (4), e, conseqüentemente, a maior centralidade de excentricidade que é igual a $1/4$, sendo estes os nós mais centrais desta rede, segundo esta medida.

4.1.2. Centralidade de Proximidade

A centralidade de excentricidade considera apenas a maior distância geodésica. Uma outra maneira de sumarizar as diversas distâncias de um nó aos demais da rede é considerar uma média das distâncias geodésicas. A *centralidade de proximidade* é igual ao recíproco da média aritmética das distâncias geodésicas do nó para os demais nós da mesma componente do nó, ou seja:

$$C_c(i) = \frac{|V_i| - 1}{\sum_{j \in V_i - \{i\}} d(i, j)}.$$

Exemplo 4.2: Em um círculo com $2n$ nós, a centralidade de proximidade de qualquer nó i é dada por:

$$\frac{2n - 1}{n + \sum_{j=1}^{n-1} 2j} = \frac{2n - 1}{n^2}.$$

Por outro lado, em um círculo com $2n + 1$ nós, a centralidade de proximidade de qualquer nó i é dada por:

$$\frac{2n}{\sum_{j=1}^n 2j} = \frac{2n}{n(n + 1)} = \frac{2}{n + 1}.$$

Em redes com mais de uma componente, a medida de centralidade de proximidade deve ser usada somente para comparar nós de uma mesma componente. Por exemplo, qualquer nó que pertença a uma componente com dois nós apenas terá uma centralidade de proximidade igual a 1, que é o máximo que pode ser obtido.

4.1.3. Centralidade Harmônica

Para contornar o problema de redes com mais de uma componente em que existem nós que não se conectam, e portanto, tem distâncias geodésicas infinitas, foi proposta a *centralidade harmônica*. A ideia da centralidade harmônica é substituir a média aritmética das distâncias presente na centralidade de proximidade, pela média harmônica das mesmas em que nós pertencentes a componentes distintas possuem distâncias infinitas entre si, ou seja:

$$C_h(i) = \frac{1}{n - 1} \sum_{j \neq i} \frac{1}{d(i, j)},$$

em que n é o número total de nós na rede.

Exemplo 4.3.: Vamos considerar uma rede que é formada por duas componentes sendo uma correspondente a um grafo estrela com n_1 nós e outra correspondente a um círculo com n_2 nós. Então, a centralidade harmônica de um nó periférico da estrela é:

$$\frac{1}{n_1 + n_2 - 1} \left(1 + (n_1 - 2) \frac{1}{2} \right) = \frac{n_1}{2(n_1 + n_2 - 1)}.$$

A centralidade harmônica do nó central da estrela é:

$$\frac{1}{n_1 + n_2 - 1} (n_1 - 1).$$

Finalmente, a centralidade harmônica dos nós no círculo serão:

$$\begin{cases} \frac{1}{n_1 + n_2 - 1} \left(\frac{2}{n_2} + \sum_{k=1}^{\frac{n_2}{2}-1} \frac{2}{k} \right) & \text{se } n_2 \text{ for par;} \\ \frac{1}{n_1 + n_2 - 1} \left(\sum_{k=1}^{\frac{n_2-1}{2}} \frac{2}{k} \right) & \text{se } n_2 \text{ for ímpar.} \end{cases}$$

4.1.4. Centralidade de média- p

Recentemente, foi proposta uma nova medida geométrica de centralidade que generaliza estas três anteriores, é a chamada centralidade de média- p [de Andrade e Rêgo, 2019]. A ideia desta medida de centralidade é usar a noção de média generalizada das distâncias, ou seja:

$$C_p(i) = \begin{cases} \left(\frac{\sum_{j \in V_i - \{i\}} d(i, j)^p}{n-1} \right)^{-\frac{1}{p}} & \text{se } p \neq 0, \\ \left(\prod_{j \in V_i - \{i\}} d(i, j) \right)^{-\frac{1}{n-1}} & \text{se } p = 0. \end{cases}$$

O valor de $C_0(i)$ pode ser obtido através do limite de $C_p(i)$ quando p tende a zero, constituindo este caso da métrica de centralidade que se obtém ao se usar a média geométrica das distâncias. Quando $p = -1$, temos que $C_{-1}(i) = C_h(i)$, ou seja a centralidade harmônica é um caso particular da centralidade de média p para $p = -1$. Quando $p = 1$, temos que $C_1(i) = \frac{n-1}{||V_i||-1} C_c(i)$, ou seja a centralidade de proximidade é um caso particular da centralidade de média p para $p = 1$. Quando $p \rightarrow \infty$, temos que $C_p(i)$ tende a $C_{ec}(i)$.

Exemplo 4.4: Vamos a considerar uma rede aleatória de Erdős-Renyi com 1000 nós e uma probabilidade de conexão $p = 0.01$. Esta rede pode ser criada usando a função `gnp_random_graph()` do pacote Networkx. Este pacote também tem funções para calcular diversas métricas de centralidade:

- `eccentricity()`: retorna um dicionário com a excentricidade de cada nó, sendo os nós as palavras chaves do dicionário.
- `closeness centrality()`: retorna um dicionário com a centralidade de proximidade de cada nó.
- `harmonic centrality()`: retorna as centralidades harmônicas dos vértices sem normalizar, isto é, sem dividir por $n - 1$ em Equação 4.1.3.

No seguinte código em Python, criamos a rede, calculamos as centralidades e geramos figuras das centralidades em função dos graus (ver Fig. 9):

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3 import networkx as nx
4
5 plt.close('all')
6 # gera um grafo aleatório de Erdős-Rényi com 1000 nós e probabilidade 0.01
7 ger = nx.gnp_random_graph(1000,0.01)
8 graus = dict(nx.degree(ger)).values()
9
10 # calculo da centralidade de excentricidade
11 ec = nx.eccentricity(ger)
12 cec = [1.0/ec[n] for n in ec]
13
14 # calculo da centralidade de proximidade
15 # só salvamos os valores do dicionário e transformamos em lista
16 cc = list(nx.closeness centrality(ger).values())
17
18 # calculo da centralidade harmônica
19 # só salvamos os valores do dicionário e transformamos em lista
20 chu= list(nx.harmonic centrality(ger).values())
21 ch = [x/(len(chu)-1) for x in chu] #normalização
22
23 plt.figure()
24 plt.plot(graus,cec,'o')
25 plt.xlabel('grau')
26 plt.ylabel('centralidade')
27
28 plt.figure()
29 plt.plot(graus,cc,'o')
30 plt.xlabel('grau')
31 plt.ylabel('centralidade')
32
33 plt.figure()
34 plt.plot(graus,ch,'o')
35 plt.xlabel('grau')
36 plt.ylabel('centralidade')
37
38 plt.show()

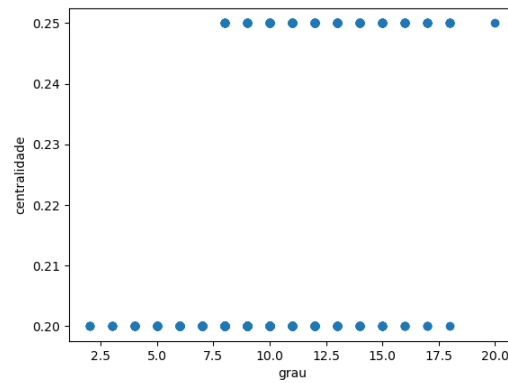
```

Embora o Networkx não tenha funções para calcular a centralidade generalizada, esta pode ser calculada facilmente como a seguir:

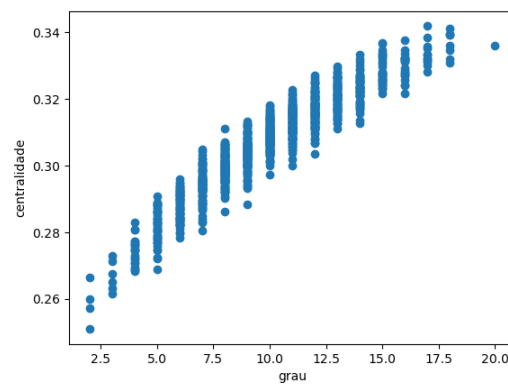
```

1 import matplotlib.pyplot as plt
2 import numpy as np
3 import networkx as nx
4
5 # gera lista com valores de p
6 p = np.linspace(-1,201,1000)
7
8 # calculo do comprimento dos caminhos mais curtos desde nó 1
9 sp= np.array(list(nx.shortest_path_length(ger,1).values()))
10
11 # calculo da centralidade generalizada para o nó 1
12 cp= np.zeros_like(p)
13 for k in range(p.shape[0]):
14     if p[k]:
15         cp[k] = np.mean(sp[1:]**p[k])**(-1.0/p[k])
16     else:
17         cp[k] = np.prod(sp[1:]**(-1.0/(sp.shape[0]-1)))

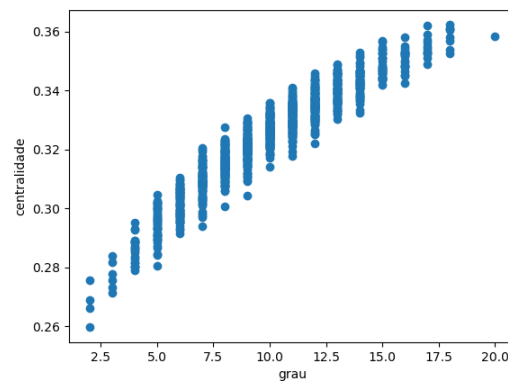
```



(a) Excentricidade



(b) Proximidade



(c) Harmônica

Figura 9: Centralidades em uma rede de Erdős-Renyi.

```

18
19 plt.figure()
20 plt.plot(p, cp, label='Centralidade generalizada')
21 plt.plot(p, np.ones_like(cp) * cec[1], label='Centralidade de excentricidade')
22 plt.plot(p, np.ones_like(cp) * cc[1], label='Centralidade de proximidade')

```



```

23 plt.plot(p, np.ones_like(cp) * ch[1], label='Centralidade harmônica')
24 plt.legend()
25 plt.xlabel('p')
26 plt.ylabel('centralidade')
27
28 plt.figure()
29 plt.plot(p, cp, label='Centralidade generalizada')
30 plt.plot(p, np.ones_like(cp) * cc[1], label='Centralidade de proximidade')
31 plt.plot(p, np.ones_like(cp) * ch[1], label='Centralidade harmônica')
32 plt.legend()
33 plt.xlabel('p')
34 plt.ylabel('centralidade')
35 plt.xlim((-1, 3))
36 plt.ylim((0.31, 0.36))
37
38 plt.show()

```

A Fig. 10 apresenta os resultados deste fragmento de código. Observe que a $C_{-1}(1) = C_h(1)$, $C_1(1) = C_c(1)$ e $C_p(1) \approx C_{ec}(1)$ para grande p .

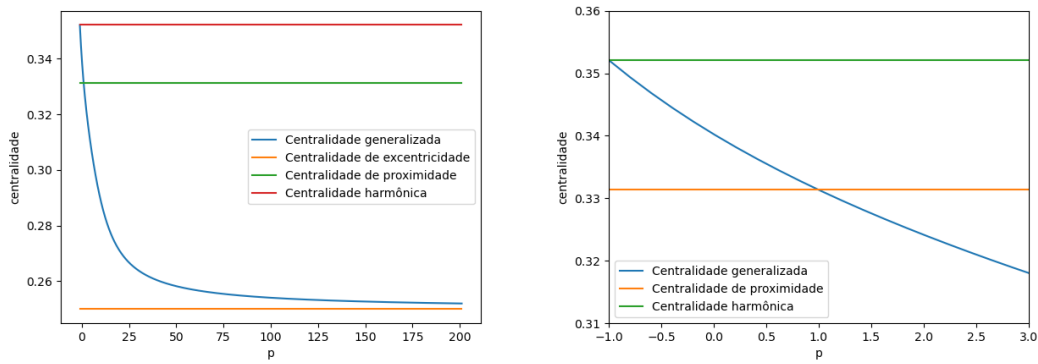


Figura 10: Centralidade generalizada em uma rede de Erdős-Rényi.

4.2. Centralidade de Intermediação

Uma outra maneira de medir a centralidade de um vértice é através de sua capacidade de conectar os demais vértices da rede. Ou seja, essa medida, conhecida como *centralidade de intermediação* (*betweenness* em inglês), quantifica o quanto os vértices são capazes de atuar como intermediários entre outros dois vértices, podendo portanto controlar o fluxo de informação entre eles [Freeman, 1977].

Para definir formalmente esta medida, considere que $Q_{j,k}$ é o número de caminhos geodésicos iniciando no vértice j e terminando no vértice k e que $Q_{j,k}(i)$ é o número de caminhos geodésicos que iniciam em j , terminam em k e passam pelo vértice i . Quanto maior a razão $\frac{Q_{j,k}(i)}{Q_{j,k}}$, maior a importância do vértice i para a conexão entre j e k . Deste modo, a centralidade de intermediação é dada por:

$$C_b(i) = \frac{1}{(n-1)(n-2)} \sum_{(j,k): j \neq k, i \notin \{j,k\}} \frac{Q_{j,k}(i)}{Q_{j,k}}.$$

Exemplo 4.5.: Considere novamente a rede descrita no Exemplo 4.3, mas na qual adicionamos uma ligação entre o nó central da estrela e um nó do círculo. Note que qualquer caminho geodésico

partindo de (resp., terminando em) um nó periférico da estrela passa pelo vértice central da estrela. Deste modo, a centralidade de intermediação do nó central da estrela é:

$$\frac{1}{(n_1 + n_2 - 1)(n_1 + n_2 - 2)} [(n_1 - 1)(n_1 + n_2 - 2) + n_2(n_1 - 1)] = \frac{(n_1 - 1)(n_1 + 2n_2 - 2)}{(n_1 + n_2 - 1)(n_1 + n_2 - 2)}.$$

Além disso, os nós periféricos da estrela possuem centralidade de intermediação nula.

Exemplo 4.6.: Considere novamente uma rede aleatória de Erdős-Renyi com 1000 nós e uma probabilidade de conexão $p = 0.01$. A função `betweenness centrality()` do `Networkx` calcula a centralidade de intermediação (ver Fig. 11):

```
1 import matplotlib.pyplot as plt
2 import networkx as nx
3
4 # calculo da centralidade de intermediação
5 # salvamos os valores do dicionário em uma lista
6 cb = list(nx.betweenness centrality(ger, normalized=True).values())
7
8 plt.figure()
9 plt.plot(graus, cb, 'o')
10 plt.xlabel('grau')
11 plt.ylabel('centralidade')
12 plt.show()
```

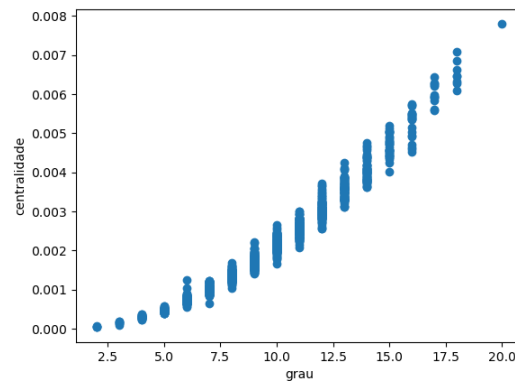


Figura 11: Centralidade de intermediação em uma rede de Erdős-Renyi.

4.3. Centralidade de Autovetor

De acordo com a centralidade de grau, a importância de um nó é igual a quantidade de vizinhos que este nó possui na rede. Deste modo, todos os vizinhos são considerados igualmente importantes. Como em geral os vizinhos ocupam diferentes posições na rede, eles têm níveis distintos de importância. A *centralidade de autovetor* é uma medida que leva em conta não somente a quantidade de vizinhos que um vértice possui, mas também a importância destes vizinhos. Formalmente, tem-se que:

$$C_{eg}(i) = \frac{1}{\lambda_1} \sum_{j=1}^n A(i, j) C_{eg}(j),$$

em que λ_1 é o autovalor com maior valor absoluto. Além disso, é comum normalizar o vetor de centralidades de autovetor de modo que este tenha norma euclidiana 1.

Em redes direcionadas, temos duas opções para a centralidade de autovetor, a depender se um vértice recebe importância dos vértices que estão ligados a ele ou dos vértices aos quais ele está ligado. Em geral, temos a primeira situação. Por exemplo, em uma rede de citações, um artigo se torna importante pelas citações que ele recebe e não pelas que ele faz. Similarmente, um site de internet é importante, pelos outros sites que tem links para ele e não por aqueles que ele exibe um link. Então, nesse caso, a centralidade de autovetor é dada por:

$$\lambda_1 C_{eg}^T = C_{eg}^T A,$$

ou seja, pelo autovetor a esquerda associado com o maior autovalor.

Em redes direcionadas, nós que possuem grau de entrada nulo possuem centralidade de autovetor nula, pois este vértice não recebe a importância de nenhum outro nó. Além disso, se todos os vizinhos que acessam um nó, possuem grau de entrada nulo, todos eles terão centralidade de autovetor nula e, portanto, o nó que só contém estes vizinhos também terá centralidade de autovetor nula. Continuando com este argumento, é possível verificar que todos os nós em uma rede direcionada acíclica possuem centralidade de autovetor nula, o que não é uma característica desejável.

4.4. Centralidade de Katz

Para lidar com o problema de que a centralidade de autovetor não é útil para redes direcionadas e acíclicas, a *centralidade de Katz* resolve o problema adicionando um valor pequeno de centralidade para todos os nós da rede, independente de suas posições na rede [Katz, 1953]. Desta forma, a centralidade de Katz é:

$$C_K(i) = \alpha \sum_{j \neq i} A(i, j) C_K(j) + \beta,$$

em que α e β são constantes positivas. β é o valor de centralidade que os nós de grau nulo possuem e $\alpha < 1/\lambda_1$ em que λ_1 é o maior autovalor de A . Na prática, é comum adotar-se o valor de α próximo a esse valor máximo e com um pequeno valor de β de modo que os valores obtidos são próximos aos da centralidade de autovetor, mas na qual todos os nós tem centralidade de Katz positiva.

Exemplo 4.7: Vamos a considerar novamente uma rede aleatória de Erdős-Renyi com 1000 nós e uma probabilidade de conexão $p = 0.01$. O Networkx calcula a centralidade de autovetor com os algoritmos `eigenvector_centrality()` ou `eigenvector_centrality_numpy()` e a centralidade de Katz com `katz_centrality()` ou `katz_centrality_numpy()`. Enquanto as versões sem `_numpy` usam um algoritmo de potências para calcular autovalores, as outras versões usam uma função do pacote ARPACK (através do Scipy). A Fig. 12 apresenta a centralidade de autovetor e a centralidade de Katz com diferentes valores de α e β . O cálculo foi feito com o seguinte código:

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 import networkx as nx
4
5 plt.close('all')
6 # gera um grafo aleatório de Erdős-Rényi com 1000 nós e probabilidade 0.01
7 ger = nx.gnp_random_graph(1000, 0.01)
```

```
8 graus = dict(nx.degree(ger)).values()
9
10 # calculo da centralidade de autovetor
11 cev = list(nx.eigenvector_centrality_numpy(ger).values())
12
13 # calculo da centralidade de Katz
14 ck1 = list(nx.katz_centrality_numpy(ger, alpha=0.08, beta=1).values())
15 ck2 = list(nx.katz_centrality_numpy(ger, alpha=0.01, beta=1).values())
16 ck3 = list(nx.katz_centrality_numpy(ger, alpha=0.08, beta=2).values())
17
18 plt.figure()
19 plt.plot(graus, cev, 'o')
20 plt.xlabel('grau')
21 plt.ylabel('centralidade')
22 plt.ylim((-0.0025, 0.0725))
23
24 plt.figure()
25 plt.plot(graus, ck1, 'o')
26 plt.xlabel('grau')
27 plt.ylabel('centralidade')
28 plt.ylim((-0.0025, 0.0725))
29
30 plt.figure()
31 plt.plot(graus, ck2, 'o')
32 plt.xlabel('grau')
33 plt.ylabel('centralidade')
34 plt.ylim((-0.0025, 0.0725))
35
36 plt.figure()
37 plt.plot(graus, ck3, 'o')
38 plt.xlabel('grau')
39 plt.ylabel('centralidade')
40 plt.ylim((-0.0025, 0.0725))
41
42 plt.show()
```

4.5. PageRank

De acordo com a centralidade de Katz, o fato de que algum vértice importante está ligado a um dado vértice dá alta importância a este dado vértice, independente de quantas ligações na rede possui aquele vértice importante. Uma métrica de centralidade que lida com este problema é o PageRank que assume que a importância de um nó é dividida igualmente entre os seus vizinhos para os quais ele possui ligação [Brin e Page, 1998]. Deste modo:

$$C_{PR}(i) = (1 - \alpha) + \alpha \sum_{j \neq i} \frac{A(j, i) C_{PR}(j)}{d^{out}(j)},$$

em que $\alpha \in (0, 1)$. O PageRank é utilizado como base para o cálculo pelo Google para classificar a importância de páginas relacionadas a uma certa pergunta.

A importância de páginas é justificada através do modelo do surfista aleatório (*random surfers model*, em inglês). Este modelo considera uma pessoa navegando na internet e considera que dado que ela está em uma certa página, ela pode acessar uma outra página digitando o endereço da página, com probabilidade $1 - \alpha$, ou escolhendo aleatoriamente um link qualquer da página atual para mudar para a próxima página, cada link tendo probabilidade α/d^{out} , em que d^{out} é o

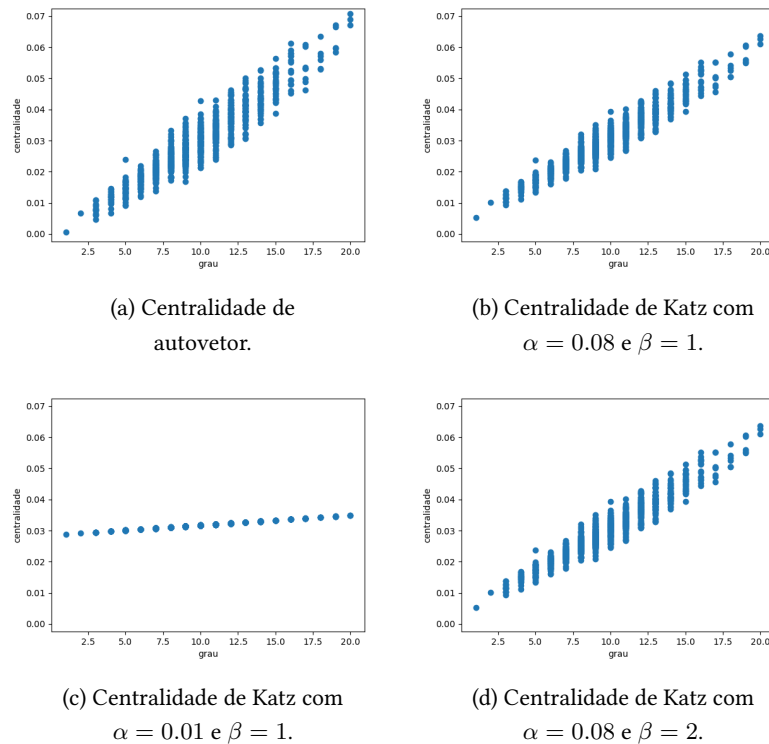


Figura 12: Centralidades de Autovetor e de Katz em uma rede de Erdős-Rényi.

número de links da página atual. De acordo com este modelo, no longo prazo a probabilidade da pessoa estar na página i é dada por $C_{PR}(i)$.

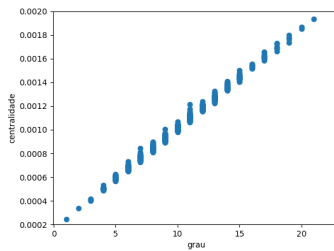
Deste modo, a importância de uma página é dada pela soma das importâncias das páginas que têm links para esta página dividido pelo número de links contidos nestas páginas. O fator α é um fator de amortecimento e pode ser pensado como a probabilidade de um usuário continuar navegando entre páginas através dos links sem parar a navegação. Usualmente, o valor $\alpha = 0,85$ tem sido utilizado, mas não há nenhuma justificativa formal conhecida para isto. No caso de um vértice j com $d^{out}(j)$ nulo, como j não contribui para a importância de nenhum outro nó, adotaremos a convenção de que $d^{out}(j) = 1$, de modo que o termo $A(j, i)/d^{out}(j)$ seja nulo.

Exemplo 4.8: Vamos a considerar novamente uma rede aleatória de Erdős-Rényi com 1000 nós e uma probabilidade de conexão $p = 0.01$. O pacote Networkx pode calcular a centralidade de Pagerank com o algoritmo `pagerank()`. A Fig. 13 apresenta a centralidade PageRank com diferentes valores de α . O cálculo foi feito como o seguinte código:

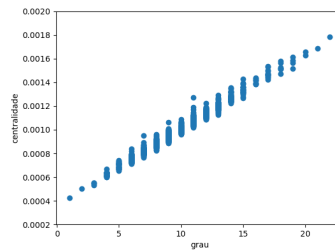
```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 import networkx as nx
4
5 plt.close('all')
6 # gera um grafo aleatório de Erdős-Rényi com 1000 nós e probabilidade 0.01
7 ger = nx.gnp_random_graph(1000, 0.01)
8 graus = dict(nx.degree(ger)).values()
9
10 # calculo da centralidade de PageRank variando o valor de \alpha
```

```

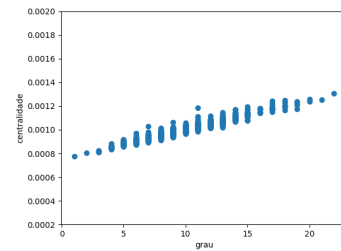
11 cpr1 = list(nx.pagerank(ger, alpha=0.85).values())
12 cpr2 = list(nx.pagerank(ger, alpha=0.65).values())
13 cpr3 = list(nx.pagerank(ger, alpha=0.25).values())
14
15 plt.figure()
16 plt.plot(graus, cpr1, 'o')
17 plt.xlabel('grau')
18 plt.ylabel('centralidade')
19 plt.ylim((0.0002, 0.002))
20
21 plt.figure()
22 plt.plot(graus, cpr2, 'o')
23 plt.xlabel('grau')
24 plt.ylabel('centralidade')
25 plt.ylim((0.0002, 0.002))
26
27 plt.figure()
28 plt.plot(graus, cpr3, 'o')
29 plt.xlabel('grau')
30 plt.ylabel('centralidade')
31 plt.ylim((0.0002, 0.002))
32
33 plt.show()
  
```



(a) $\alpha = 0.85$.



(b) $\alpha = 0.65$.



(c) $\alpha = 0.25$.

Figura 13: PageRank em uma rede de Erdős-Renyi.

5. Outras Métricas

Nesta seção, descreveremos outras métricas locais que descrevem características individuais dos nós que não são relacionadas à importância deles na rede, bem como outras métricas globais que indicam o grau de coesão e homofilia de uma rede.

5.1. Coeficiente de Agrupamento Local

Na Seção 3.4, vimos uma medida global de agrupamento que nos dá uma informação da coesão da rede em termos da conectividade dos vizinhos de um nó. Como também pode ser de interesse avaliar o quanto os nós são capazes de proporcionar interação entre os seus vizinhos, temos uma medida de agrupamento local chamada de *coeficiente de agrupamento local*. O coeficiente de agrupamento local de um nó i é dado por:

$$cl(i) = \frac{\sum_{(j,k): j \neq i, k \neq i, k \neq j} A(i,j)A(i,k)A(j,k)}{\sum_{(j,k): j \neq i, k \neq i, k \neq j} A(i,j)A(i,k)}.$$

Por convenção, nós que possuem grau 0 ou 1, possuem coeficiente de agrupamento local nulo. Baseado neste coeficiente de agrupamento local, pode-se definir uma outra medida de agrupamento global da rede que é o coeficiente de agrupamento local médio:

$$\overline{cl}(G) = \frac{1}{n} \sum_{i=1}^n cl(i).$$

Em comparação com o coeficiente de agrupamento total, o coeficiente de agrupamento local médio dá maior peso aos nós que possuem menor grau.

Exemplo 5.1: Considere uma rede que consiste de um círculo com $n - 1$ nós todos conectados a um nó central. Cada um dos nós no círculo, possui coeficiente de agrupamento local igual a $2/3$, pois eles possuem três vizinhos (dois no círculo e 1 no centro) e dos 3 possíveis pares de vizinhos, existe ligação em dois deles (que são as ligações dos vizinhos no círculo com o nó central). O nó central por sua vez possui coeficiente de agrupamento local igual a $\frac{2(n-1)}{(n-1)(n-2)} = \frac{2}{n-2}$. Deste modo, o coeficiente de agrupamento local médio é dado por:

$$\overline{cl}(G) = \frac{1}{n} \left[\frac{2(n-1)}{3} + \frac{2}{n-2} \right] = \frac{2n^2 - 6n + 10}{3n(n-2)}.$$

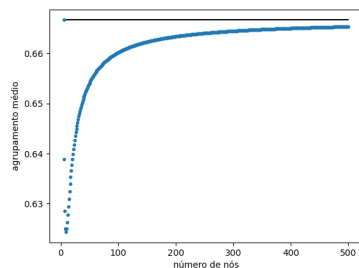
Por outro lado, para o cálculo do coeficiente de agrupamento total, note que cada nó no círculo tem 3 pares de vizinhos dos quais dois são ligados (exceto no caso $n = 4$, em que os três pares de vizinhos são ligados) e o nó central possui $\frac{(n-1)(n-2)}{2}$ pares de vizinhos dos quais $(n-1)$ são ligados. Logo

$$cl(G) = \begin{cases} 1 & \text{se } n = 4, \\ \frac{2(n-1) + (n-1)}{3(n-1) + \frac{(n-1)(n-2)}{2}} = \frac{6}{n+4} & \text{se } n > 4. \end{cases}$$

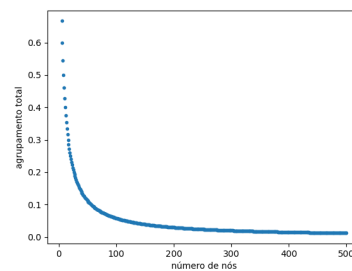
Observe que quando n se torna muito grande, o coeficiente de agrupamento local médio tende a $2/3$, enquanto o coeficiente de agrupamento total tende a zero. Estes fatos são apresentados na Fig. 14 que foi feita com seguinte código de Python. A função `wheel_graph()` do NetworkX constrói uma rede formada por um círculo de $n - 1$ vértices conectados a um nó central. A função `average_clustering()` do NetworkX retorna o coeficiente de agrupamento médio e a função `transitivity()` retorna o coeficiente de agrupamento total como já foi mencionado na Seção 3.4.

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 import networkx as nx
4
5 ccla = []
6 cclt = []
7 n = np.linspace(5, 500, 496)
8 for k in n:
9     gci = nx.wheel_graph(int(k))
10    ccla.append(nx.average_clustering(gci))
11    cclt.append(nx.transitivity(gci))
12
13 plt.figure()
14 plt.plot(n, 2/3*np.ones_like(n), 'k')
15 plt.plot(n, ccla, 'r.')
16 plt.xlabel('Número de nós')
```

```
17 plt.ylabel('agrupamento médio')
18
19 plt.figure()
20 plt.plot(n, cclt, '.')
21 plt.xlabel('número de nós')
22 plt.ylabel('agrupamento total')
23
24 plt.show()
```



(a) Agrupamento médio.
Linha preta em $2/3$.



(b) Agrupamento total.

Figura 14: Coeficiente de agrupamento em um círculo de $n - 1$ nós conectados com um nó central.

Exemplo 5.2.: A Fig. 15 apresenta o coeficiente de agrupamento local para o exemplo de uma rede Erdős-Rényi já tratado. Neste caso, a função de Networkx usada é `clustering()`:

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 import networkx as nx
4
5 # gera grafo aleatório de Erdős-Rényi com 1000 nós e probabilidade 0.01
6 ger = nx.gnp_random_graph(1000, 0.01)
7 graus = dict(nx.degree(ger)).values()
8
9 # calculo do coeficiente de agrupamento local
10 ccl = list(nx.clustering(ger).values())
11
12 plt.figure()
13 plt.plot(graus, ccl, 'o')
14 plt.xlabel('grau')
15 plt.ylabel('agrupamento')
16
17 plt.show()
```

5.2. Homofilia

Uma das informações importantes em redes complexas e, em especial, em redes sociais é saber se os nós possuem alguma tendência a se ligar mais a nós similares a si mesmos, o que é conhecido como homofilia ou assortatividade. Por exemplo, cientistas tendem a ter contato com mais cientistas que, em geral, outras pessoas possuem contatos com cientistas. Pessoas tendem a ter mais relacionamentos intrarraciais que inter-raciais. Nesta subseção, veremos algumas medidas que têm sido usada para mensurar homofilia.

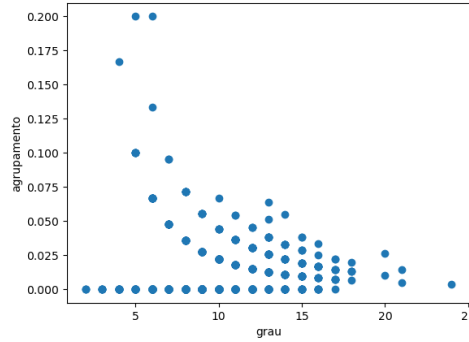


Figura 15: Agrupamento local em uma rede de Erdős-Renyi.

5.2.1. Índice EI

Considere que o conjunto V de vértices de um grafo possa ser particionado em grupos $\{V^1, V^2, \dots, V^a\}$ de acordo com atributos que esses nós possuam. O índice EI mede a homofilia através de uma razão que envolve a quantidade de ligações entre nós de grupos distintos e a quantidade de ligações entre nós do mesmo grupo. Ele pode ser calculado para um subconjunto qualquer de nós da rede, que pode conter desde um único nó ou todos os nós da rede. Para tal, dada a partição dos nós, vamos definir uma função indicadora $\delta : V \times V \rightarrow \{0, 1\}$ tal que:

$$\delta(i, j) = \begin{cases} 1 & \text{se } \exists k \text{ tal que } i, j \in V^k, \\ 0 & \text{caso contrário.} \end{cases}$$

Seja $V' \subseteq V$, então podemos encontrar o número de ligações internas e externas envolvendo os nós em V' , respectivamente, por

$$IL(V') = \sum_{i \in V'} \sum_{j \notin V' \text{ ou } j > i} A(i, j) \delta(i, j)$$

e

$$EL(V') = \sum_{i \in V'} \sum_{j \notin V' \text{ ou } j > i} A(i, j) (1 - \delta(i, j)).$$

Desta forma, o índice EI do conjunto V' é dado por:

$$EI(V') = \frac{EL(V') - IL(V')}{EL(V') + IL(V')}.$$

Observe que o valor do índice EI varia no intervalo $[-1, 1]$, em que um índice EI igual a -1 (resp., 1) indica um subconjunto de nós que só possui ligações internas (resp., externas). A restrição de $j > i$ é para evitar que uma mesma aresta seja contada duas vezes no caso de uma rede não direcionada. No caso de uma rede direcionada, esta restrição deve ser substituída por $j \neq i$.

5.2.2. Assortatividade

O valor do índice EI nos informa se um dado grupo de nós tem mais ligações externas ou internas na rede. Porém, este fato pode estar apenas relacionado de um certo grupo de atributos ser mais ou menos frequente na rede que outros. Por exemplo, suponha que um certo indivíduo faça parte de algum grupo minoritário de uma população. Pelo fato de no grupo minoritário, por definição, haverem poucas pessoas, pode acontecer que este indivíduo possua mais ligações

externas do que internas no grupo e, portanto, tenha um índice EI positivo. Contudo, isso não necessariamente quer dizer que este indivíduo tenha preferência por ter ligações externas, para fazer tal afirmação precisaríamos comparar com o número esperado de ligações que seria obtido em uma rede aleatória com o mesmo conjunto de nós e o mesmo número de arestas escolhidas aleatoriamente. O *coeficiente de assortatividade* nos dá essa informação [Newman, 2002, 2003]. Para tal, veremos três tipos de assortatividade. O primeiro definido para quando os nós estão particionados em uma quantidade finita de grupos de atributos, como no caso do índice EI. O segundo é para quando os atributos dos nós são caracterizados por valores escalares, tais como idade, PIB, índice h, etc. E o último é para quando o atributo utilizado para avaliar a homofilia é a própria centralidade de grau dos nós.

De acordo com os três coeficientes de assortatividade que veremos, valores positivos indicam tendência a homofilia, valores negativos tendência a heterofilia (conectar-se com grupos distintos) e valores próximos de zero indicam que os atributos não influenciam a distribuição das ligações.

5.2.3. Para atributos categóricos

Considere novamente que o conjunto V de vértices de um grafo possa ser particionado em grupos $\{V^1, V^2, \dots, V^a\}$ de acordo com atributos categóricos que esses nós possuam. Em uma rede com m arestas, vamos definir a fração de arestas que ligam um nó no conjunto V^i a um nó no conjunto V^j :

$$e_{ij} = \frac{\sum_{k_1 \in V^i} \sum_{k_2 \in V^j} A(k_1, k_2)}{\sum_{k_1 \in V} \sum_{k_2 \in V} A(k_1, k_2)}.$$

A fração de arestas que iniciam (resp., terminam) em vértices do conjunto V_i é dada, respectivamente, por:

$$a_i = \sum_j e_{ij}$$

e

$$b_i = \sum_j e_{ji}.$$

Em redes não direcionadas, temos que $e_{ij} = e_{ji}$ e $a_i = b_i$. Se as ligações da rede fossem formadas aleatoriamente mantidas as frequências das arestas incidentes em nós de cada grupo, a fração esperada de ligação entre nós pertencentes a um grupo V^i seria igual a $a_i b_i$. Deste modo, o coeficiente de assortatividade é dado por:

$$As^c(G) = \frac{\sum_{i=1}^a e_{ii} - \sum_{i=1}^a a_i b_i}{1 - \sum_{i=1}^a a_i b_i},$$

em que a normalização no denominador é para garantir que no caso de todas as arestas ligarem nós do mesmo grupo, homofilia perfeita, o coeficiente de assortatividade tenha valor igual a 1.

O Exemplo 5.3 ilustra o valor do EI index da rede toda, bem como o valor da assortatividade categórica.

Exemplo 5.3: Considere a rede não direcionada ilustrada na Figura 16. Para calcular o valor do índice EI da rede, temos:

$$EI = \frac{2 - 8}{2 + 8} = -0.6.$$

Já o valor da assortatividade categórica é dado por:

$$As^c(G) = \frac{6/20 + 10/20 - (8/20)^2 - (12/20)^2}{1 - (8/20)^2 - (12/20)^2} = 0.583.$$

Ambas as métricas indicam uma rede com homofilia.

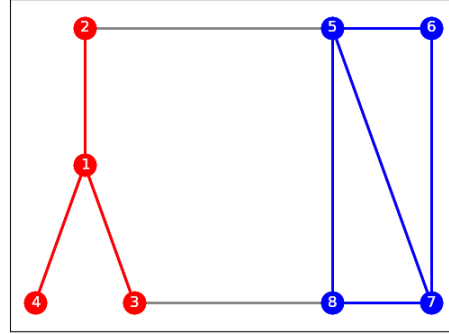


Figura 16: Análise de homofilia em uma rede não direcionada.

5.2.4. Para atributos escalares

No caso em que os atributos são escalares, como idade ou PIB, por exemplo, uma primeira ideia seria calcular a assortatividade particionando os valores do atributo escalar em categorias. Essa medida incorre em erros, pois após a categorização, atributos distintos que caírem na mesma categoria serão tratados como idênticos e atributos próximos que caírem em categorias distintas serão tratados como completamente distintos. Neste caso, o coeficiente de assortatividade para atributos escalares é calculado levando em conta a covariância dos atributos ao longo das arestas da rede. Formalmente, seja x_i o atributo escalar do nó i . Então, como o nó i possui $d(i)$ arestas ligadas a ele, o valor médio dos atributos escalares ao longo das arestas é dado por:

$$\bar{x} = \frac{\sum_i d(i)x_i}{\sum_i d(i)} = \frac{1}{2m} \sum_i d(i)x_i.$$

A covariância ao longo das arestas dos atributos de suas extremidades é:

$$\begin{aligned} \frac{\sum_{ij} A(i,j)(x_i - \bar{x})(x_j - \bar{x})}{\sum_{ij} A(i,j)} &= \frac{1}{2m} \left(\sum_{ij} A(i,j)x_i x_j \right) - \bar{x}^2 \\ &= \frac{1}{2m} \left[\sum_{ij} \left(A(i,j) - \frac{d(i)d(j)}{2m} \right) x_i x_j \right]. \end{aligned}$$

Para garantir que a rede com homofilia perfeita em que só existam ligações entre nós com exatamente o mesmo atributo escalar tenha coeficiente de assortatividade igual a 1, a covariância é normalizada dividindo pelo seu maior valor possível:

$$\begin{aligned}
 As^e(G) &= \frac{\frac{1}{2m} \sum_{ij} (A(i, j) x_i x_j) - \bar{x}^2}{\frac{1}{2m} \sum_{ij} (A(i, j) x_i^2) - \bar{x}^2} \\
 &= \frac{\sum_{ij} \left(A(i, j) - \frac{d(i)d(j)}{2m} \right) x_i x_j}{\sum_i d(i) x_i^2 - \frac{1}{2m} (\sum_i d(i) x_i)^2} \quad (2)
 \end{aligned}$$

5.2.5. Para atributo igual ao grau

Um caso especial de assortatividade com atributo escalar que é bastante utilizado é quando o atributo escalar é a própria centralidade de grau, ou seja, $x_i = d(i)$. Neste caso a assortatividade de grau, $As^d(G)$ é calculada exatamente como a centralidade de atributo escalar fazendo $x_i = d(i)$ e $x_j = d(j)$ na fórmula. Esta assortatividade mede se na rede existe uma tendência de nós muito conectados estarem conectados entre si e nós pouco conectados estarem ligados entre si. Ao contrário dos outros dois casos, para o cálculo da assortatividade de grau não é necessário nenhuma outra informação sobre os nós que não esteja descrita na matriz de adjacência da rede.

Redes com assortatividade de grau positiva possuem nós bastante conectados ligados entre si, formando um núcleo da rede que se destaca da periferia. Essa tendência é bastante observada em redes sociais, onde por exemplo pessoas populares tendem a ter amizades com outras pessoas populares, enquanto pessoas mais introspectivas tendem a ter amizades com outras pessoas introspectivas. Por outro lado, redes com assortatividade de grau negativa têm nós bastante conectados ligados a nós pouco conectados, formando estruturas de rede que lembram estrelas, sendo mais uniforme sem a separação núcleo-periferia observada nas redes com assortatividade de grau positiva.

Exemplo 5.4.: Na Seção 3.7, vimos o exemplo das redes de suporte emocional entre personagens dos livros da saga Harry Potter. Os personagens têm dois atributos categóricos, o gênero do personagem e a casa a que o ele pertence. Se bem o ano da escola pode ser considerado um atributo categórico (o conjunto de anos é discreto e finito) ou numérico, nesta seção tratamos-o como numérico. É interessante estudar a evolução da assortatividade no tempo. A Fig. 17 apresenta a variação da assortatividade com o número de livro. Por exemplo, observe que a assortatividade relacionada como a casa é muito alta no primeiro livro, mas baixa nos seguintes.

As funções de Networkx mais relevantes neste exemplo são:

- `attribute_assortativity_coefficient()`: calcula a assortatividade relacionada com atributos categóricos.
- `numeric_assortativity_coefficient()`: calcula a assortatividade relacionada com atributos numéricos. Dado que a Eq. 2 é para redes não direcionadas, ignoramos as direções das arestas no exemplo.
- `degree_assortativity_coefficient()`: calcula a assortatividade para o atributo igual ao grau. No caso de grafos direcionados, esta função pode calcular quatro valores diferentes, dependendo de se trata de graus saintes, entrantes, saintes e entrantes ou entrantes e saintes. No exemplo dos livros de Harry Potter, consideramos graus totais como se fossem redes não direcionadas.

```

1 import pandas as pd
2 import networkx as nx
3 import matplotlib.pyplot as plt
4
5 def carregarlivro(livro):
6     atri = pd.read_csv('data/harrypotter/hpattributes.txt', sep='\t')
7     ares = pd.read_csv('data/harrypotter/hpbook{1d}.txt'.format(livro), sep=' ',
8     , header=None)
9     nome = pd.read_csv('data/harrypotter/hpnames.txt', sep='\t')
10
11     # grafo dirigido gpotter
12     gpotter = nx.DiGraph()
13     n = atri.shape[0]
14     # adiciona nós a gpotter
15     for k in range(n):
16         gpotter.add_node(k, nome=nome['name'][k],
17         ano=atri['schoolyear'][k],
18         gen=atri['gender'][k],
19         casa=atri['house'][k])
20
21     # adiciona aresta a gpotter
22     for k in range(n):
23         for m in range(n):
24             if ares.values[k][m] == 1:
25                 gpotter.add_edge(k, m)
26
27     agen = nx.attribute_assortativity_coefficient(gpotter, "gen")
28     acasa= nx.attribute_assortativity_coefficient(gpotter, "casa")
29     aano = nx.numeric_assortativity_coefficient(gpotter.to_undirected(), "ano")
30     agrau= nx.degree_assortativity_coefficient(gpotter.to_undirected())
31
32     return agen, acasa, aano, agrau
33
34 # cria listas
35 agen = 6*[0]
36 acasa= 6*[0]
37 aano = 6*[0]
38 agrau= 6*[0]
39 livros = [1,2,3,4,5,6]
40
41 for k in range(len(livros)):
42     agen[k], acasa[k], aano[k], agrau[k]=carregarlivro(livros[k])
43
44 plt.figure()
45 plt.plot(livros, agen, '-o', label='Gênero')
46 plt.plot(livros, acasa, '-s', label='Casa')
47 plt.plot(livros, aano, '-*', label='Ano')
48 plt.plot(livros, agrau, '-^', label='Grau')
49 plt.legend()
50 plt.xlabel('livro')
51 plt.ylabel('assortatividade')
52 plt.show()

```

Embora seja mais comum utilizar a centralidade de grau como atributo escalar para o cálculo da assortatividade, outras métricas de centralidade também podem ser utilizadas. Por exemplo, se utilizarmos a centralidade de intermediação mediremos a tendência de nós intermediários

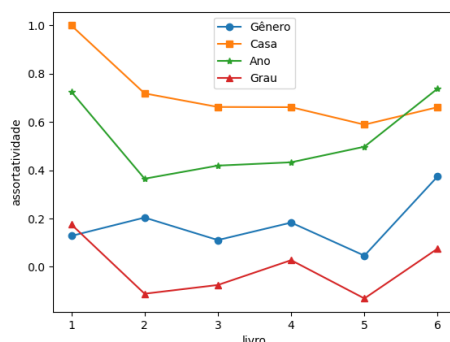


Figura 17: Assortatividades nos livros da saga Harry Potter.

se conectarem entre si. Recentemente, Fierens e Rêgo [2024], mostraram que como a medida de assortatividade escalar de Newman é baseada na correlação de Pearson dos atributos, ela não consegue capturar padrões não lineares de homofilia e também não consegue lidar com atributos vetoriais. Estes autores propuseram uma generalização desta medida que utiliza a noção de similaridade dos atributos para abordar tais situações.

6. Considerações Finais

Os conteúdos descritos nestas notas de aulas visam expor um conteúdo introdutório na área de Redes Complexas. Vimos o que é uma Rede Complexa, que ela pode ser utilizada para modelar fenômenos em diversas áreas do conhecimento e apresentamos as principais métricas globais e locais que são utilizadas descrever uma rede complexa.

Tópicos mais avançados envolvendo redes complexas incluem, por exemplo: modelos de formação de redes complexas, técnicas de detecção de comunidades, modelos de difusão em redes e métodos de predição de novas arestas. Para o leitor interessado em aprofundar seus conhecimentos, sugerimos consultar os seguintes livros textos [Newman, 2018; Jackson, 2010; Wasserman e Faust, 1994; Zinoviev, 2018].

Referências

- Bossaert, G. e Meidert, N. (2013). “We are only as strong as we are united, as weak as we are divided” a dynamic analysis of the peer support networks in the Harry Potter books. *Open Journal of Applied Sciences*, 3(02):174–185.
- Brin, S. e Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117.
- de Andrade, R. L. e Rêgo, L. C. (2019). p-means centrality. *Communications in Nonlinear Science and Numerical Simulation*, 68:41–55. ISSN 1007-5704. URL <https://www.sciencedirect.com/science/article/pii/S1007570418302636>.
- de Pontes, L. S. e Rêgo, L. C. (2022). Impact of macroeconomic variables on the topological structure of the brazilian stock market: A complex network approach. *Physica A: Statistical Mechanics and its Applications*, 604:127660. ISSN 0378-4371. URL <https://www.sciencedirect.com/science/article/pii/S0378437122004447>.
- Fierens, P. I. e Rêgo, L. C. (2024). A similarity-based assortativity measure for complex networks. *Journal of Complex Networks*, 12(2):cnae010.

- Freeman, L. (1977). A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41.
- Guida, M. e Maria, F. (2007). Topology of the italian airport network: A scale-free small-world network with a fractal structure? *Chaos, solitons & fractals*, 31(3):527–536.
- Jackson, M. O. (2010). *Social and Economic Networks*. Princeton University Press.
- Jin, S. e Bestavros, A. (2006). Small-world characteristics of internet topologies and implications on multicast scaling. *Computer Networks*, 50(5):648–666.
- Kamada, T. e Kawai, S. (1989). An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1):7–15.
- Katz, L. (1953). A new status index derived from sociometric analysis. *Psychometrika*, 18:39–43.
- Lindelauf, R., Hamers, H., e Husslage, B. (2013). Cooperative game theoretic centrality analysis of terrorist networks: the cases of jemaah islamiyah and al qaeda. *European Journal of Operational Research*, 229(1):230–238.
- Milgram, S. (1967). The small-world problem. *Psychology Today*, 2:60–67.
- Newman, M. (2018). *Networks*. Oxford University Press.
- Newman, M. E. (2002). Assortative mixing in networks. *Physical Review Letters*, 89(20):208701.
- Newman, M. E. (2003). Mixing patterns in networks. *Physical Review E*, 67(2):026126.
- Newman, M. E. (2004). *Who Is the Best Connected Scientist? A Study of Scientific Coauthorship Networks*, p. 337–370. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN 978-3-540-44485-5.
- Pandas, D. d. (2022). pandas-dev/pandas: Pandas. URL <https://pandas.pydata.org/>. Accessed: 2022-08-10.
- Santos, Y. A. C., Rêgo, L. C., e Ospina, R. (2022). Online handwritten signature verification via network analysis. *Physica A: Statistical Mechanics and its Applications*, 600:127582. ISSN 0378-4371. URL <https://www.sciencedirect.com/science/article/pii/S0378437122004022>.
- Sayama, H. (2015). *Introduction to the modeling and analysis of complex systems*. Open SUNY Textbooks.
- Snijders, T. A. Data files and variables for the Harry Potter support networks of Goele Bossaert and Nadine Meidert . <https://www.stats.ox.ac.uk/~snijders/siena/HarryPotterData.html>. Accessed: 2022-08-10.
- Wang, J., Yang, C., Wang, R., Yu, H., Cao, Y., e Liu, J. (2016). Functional brain networks in alzheimer’s disease: Eeg analysis based on limited penetrable visibility graph and phase space method. *Physica A: Statistical Mechanics and its Applications*, 460:174–187. ISSN 0378-4371. URL <https://www.sciencedirect.com/science/article/pii/S0378437116301972>.
- Wasserman, S. e Faust, K. (1994). *Social Network Analysis: Methods and Applications*. Cambridge University Press.

Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473.

Zinoviev, D. (2018). *Complex network analysis in Python: Recognize-construct-visualize-analyze-interpret*. Pragmatic Bookshelf.