Inferência estatística e amostragem de redes complexas

Andrey Luan Gomes Contel

Dissertação de Mestrado do Programa Interinstitucional de
Pós-Graduação em Estatística (PIPGes)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Andrey Luan Gomes Contel

Inferência estatística e amostragem de redes complexas

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP e ao Departamento de Estatística – DEs-UFSCar, como parte dos requisitos para obtenção do título de Mestre em Estatística – Programa Interinstitucional de Pós-Graduação em Estatística. *VERSÃO REVISADA*

Área de Concentração: Estatística

Orientador: Prof. Dr. Adriano Kamimura Suzuki

USP – São Carlos
Julho de 2019

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

C761i Contel, Andrey Luan Gomes
 Inferência estatística e amostragem de redes
 complexas / Andrey Luan Gomes Contel; orientador
 Adriano Kamimura Suzuki. -- São Carlos, 2019.
 78 p.

 Dissertação (Mestrado - Programa
 Interinstitucional de Pós-graduação em Estatística) --
 Instituto de Ciências Matemáticas e de Computação,
 Universidade de São Paulo, 2019.

 1. inferência. 2. amostragem. 3. redes
 complexas. 4. grafos. 5. aprendizado de máquina. I.
 Suzuki, Adriano Kamimura, orient. II. Título.

Andrey Luan Gomes Contel

Statistical inference and sampling of complex networks

Master dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP and to the Department of Statistics – DEs-UFSCar, in partial fulfillment of the requirements for the degree of the Master Interagency Program Graduate in Statistics.
FINAL VERSION

Concentration Area: Statistics

Advisor: Prof. Dr. Adriano Kamimura Suzuki

USP – São Carlos
July 2019

RESUMO

CONTEL, A. L. G. **Inferência estatística e amostragem de redes complexas**. 2019. 77 p. Dissertação (Mestrado em Estatística – Programa Interinstitucional de Pós-Graduação em Estatística) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2019.

Redes complexas são formadas por amostras de dados obtidos a partir do mapeamento da estrutura de sistemas complexos. Geralmente, diferentes métodos de amostragem são considerados para a construção da rede. No entanto, dependendo do método, as amostras podem ser muito diferentes das redes originais. Logo, uma comparação entre os diferentes métodos de amostragem é altamente recomendável, de modo a permitir escolher o método que preserve uma determinada característica. Nesse trabalho, propomos uma comparação de métodos de amostragem de redes e um estudo considerando métodos inferência estatística e técnicas de amostragem em grafos para estimar as principais medidas de caracterização.

Palavras-chave: Inferência, Amostragem, Redes Complexas, Grafos, Aprendizado de Máquina.

ABSTRACT

CONTEL, A. L. G. **Statistical inference and sampling of complex networks**. 2019. 77 p. Dissertação (Mestrado em Estatística – Programa Interinstitucional de Pós-Graduação em Estatística) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2019.

Complex networks are formed by data samples obtained from the mapping of complex system structures. Generally, different sampling methods are considered for the construction of the network. However, depending on the method, the samples may be very different from the original networks. Therefore, a comparison between the different methods of sampling is highly recommended, in order to allow to choose the method that preserves a certain characteristic. In this work, we propose a comparison of network sampling methods and a study considering statistical inference methods and graph sampling techniques to estimate the main characterization measures.

Keywords: Inference, Sampling, Complex Networks, Graphs, Machine Learning.

LISTA DE ILUSTRAÇÕES

Figura 1 – Pontes de Königsberg	22
Figura 2 – Tipos de Grafo	26
Figura 3 – Matriz de Adjacência	27
Figura 4 – Distribuição do grau da rede de <i>e-mails</i> coletada na <i>University Rovira i Virgili</i>	27
Figura 5 – Exemplo de um subgrafo Induzido.	32
Figura 6 – Exemplo do subgrafo induzido com vizinhança.	33
Figura 7 – Exemplo do subgrafo induzido Snowball.	34
Figura 8 – Exemplo do subgrafo Incidente.	36
Figura 9 – Exemplo de um subgrafo por caminhada aleatória.	37
Figura 10 – Redes Reais	40
Figura 11 – Grau Médio	41
Figura 12 – Transitividade	42
Figura 13 – Estatística D - K.S. para distribuição do Grau	43
Figura 14 – Estatística D - K.S. para Autovetor Principal	44
Figura 15 – Valor p - K.S. para distribuição do Grau	45
Figura 16 – Valor p - K.S. para Autovetor Principal	46
Figura 17 – Distribuição do Grau das amostras comparado com a real	47
Figura 18 – Autovetor principal das amostras comparado com o real	48
Figura 19 – Estimadores para o Modelo AL	49
Figura 20 – Rede Neural	51
Figura 21 – Exemplo KNN	53
Figura 22 – Exemplo de Árvore de Decisão	54
Figura 23 – Análise PCA	58
Figura 24 – Importância das Variáveis	60
Figura 25 – Shannon x $\langle Betweenness \rangle^2$	60
Figura 26 – Análise PCA	61
Figura 27 – Importância das Variáveis	63
Figura 28 – Análise PCA	64

LISTA DE CÓDIGOS-FONTE

Código-fonte 1 – Amostragem por Vértice	32
Código-fonte 2 – Amostragem por Vértice com Vizinhança	33
Código-fonte 3 – Amostragem <i>SnowBall</i>	35
Código-fonte 4 – Amostragem por <i>Links</i>	36
Código-fonte 5 – Amostragem por caminhada aleatória	37

LISTA DE TABELAS

Tabela 1 – Redes Reais	39
Tabela 2 – Acurácia Redes Artificiais	59
Tabela 3 – Acurácia Redes Artificiais	62
Tabela 4 – Acurácia	63
Tabela 5 – Acurácia Redes Reais	65
Tabela 6 – Acurácia Seleção <i>Naive Bayes</i>	65
Tabela 7 – Acurácia Seleção Floresta Aleatória	66
Tabela 8 – Acurácia Redes Reais com Amostras	66
Tabela 9 – Acurácia Seleção <i>Naive Bayes</i>	67
Tabela 10 – Acurácia Seleção Floresta Aleatória	67
Tabela 11 – Seleção <i>Naive Bayes</i>	68
Tabela 12 – Seleção Floresta Aleatória	68

LISTA DE ABREVIATURAS E SIGLAS

AASs	Amostragem Aleatória Simples Sem Reposição
ACAV	Amostragem por Caminhada Aleatória com Vizinhaça
AL	Amostragem por <i>Links</i>
AV	Amostragem por Vértice
AVV	Amostragem por Vértice com Vizinhaça
BA	Barabási–Albert
ER	Erdős–Rény
LFR	Lancichinetti-Fortunato-Radicchi
SB	Amostragem <i>SnowBall</i>
WA	Waxman
WS	Watts-Strogatz

LISTA DE SÍMBOLOS

G — Grafo

V — Conjunto de Vértices

E — Conjunto de Arestas (*Edges*)

$e = (u, v)$ — Aresta e composta pelos vértices u e v

$cl()$ — *clustering coefficient*

k — Primeiro Momento Grau

k_2 — Segundo Momento grau

k_3 — Terceiro Momento do grau

ev — *Eigenvalor*

e — Primeiro Momento *Eigenvector*

e_2 — Segundo Momento *Eigenvector*

e_3 — Terceiro Momento *Eigenvector*

c — Primeiro Momento *Closeness*

c_2 — Segundo Momento *Closeness*

c_3 — Terceiro Momento *Closeness*

b — Primeiro Momento *Betweenness*

b_2 — Segundo Momento *Betweenness*

b_3 — Terceiro Momento *Betweenness*

p — Primeiro Momento *Page Rank*

p_2 — Segundo Momento *Page Rank*

p_3 — Terceiro Momento *Page Rank*

S — Entropia de *Shannon*

T — Transitividade

n — Número de vértices

m — Número de arestas

SUMÁRIO

1	INTRODUÇÃO	21
1.1	Revisão Histórica	21
1.2	Objetivo	22
1.3	Organização do Trabalho	23
2	REDES COMPLEXAS	25
2.1	Teoria de Redes	25
2.1.1	<i>Características Básicas</i>	25
2.1.2	<i>Medidas</i>	26
2.1.3	<i>Modelos de Redes</i>	29
2.1.3.1	<i>Grafo Aleatório</i>	29
2.1.3.2	<i>Small-World</i>	30
2.1.3.3	<i>Redes sem escala</i>	30
2.2	Amostragem	30
2.2.1	<i>Amostragem Aleatória Simples Sem Reposição (AASs)</i>	30
2.2.2	<i>Subgrafo Induzido (AV)</i>	31
2.2.3	<i>Subgrafo Induzido com Vizinhança(AVV)</i>	33
2.2.4	<i>Subgrafo Induzido Snowball (SB)</i>	34
2.2.5	<i>Subgrafo Incidente (AL)</i>	35
2.2.6	<i>Subgrafo por Caminhada Aleatória com Vizinhança (ACAV)</i>	36
3	AMOSTRAS E ANÁLISES	39
3.1	Dados	39
3.2	Amostras	41
3.3	Modelo AV	49
4	APRENDIZADO DE MÁQUINA	51
4.1	Redes Neurais	51
4.2	Naive Bayes	52
4.3	KNN	53
4.4	Florestas Aleatórias	54
4.5	Métricas de Avaliação	55
5	CLASSIFICAÇÃO DE REDES	57

5.1	Estrutura	57
5.1.1	<i>Dados</i>	57
5.1.2	<i>Medidas</i>	57
5.2	Redes Artificiais	58
5.2.1	<i>Primeira Parte</i>	58
5.2.1.1	<i>Análise de Componentes Principais</i>	58
5.2.1.2	<i>Machine Learning</i>	59
5.2.2	<i>Segunda Parte</i>	61
5.2.2.1	<i>Análise de Componentes Principais</i>	61
5.2.2.2	<i>Machine Learning</i>	62
5.3	Redes Reais	62
5.3.1	<i>Primeira Parte</i>	63
5.3.1.1	<i>Análise de Componentes Principais</i>	63
5.3.1.2	<i>Machine Learning</i>	65
5.3.2	<i>Amostras</i>	66
6	CONSIDERAÇÕES FINAIS	69
	REFERÊNCIAS	71
APÊNDICE A	REDES UTILIZADAS	75
A.1	Redes Reais	75

INTRODUÇÃO

Neste capítulo, será feita uma revisão histórica sobre redes complexas, assim como os objetivos com esse trabalho.

1.1 Revisão Histórica

“Vivemos em um mundo conectado” por mais repetida que possa parecer, essa frase explica a explosão de pesquisas relacionadas a redes complexas atualmente, segundo [Kolaczyk \(2009\)](#). Embora isso pareça natural quando pensamos em redes sociais, podemos encontrar diversos exemplos de redes complexas em diversas áreas, desde a interação entre animais em cadeias alimentares até redes de roteadores na Internet. A Teoria das Redes Complexas é uma área multidisciplinar, que inicialmente utiliza da Teoria de Grafos para modelar problemas de praticamente todas as áreas do conhecimento ([COSTA *et al.*, 2011](#)). A Teoria dos Grafos teve seu primeiro trabalho atribuído a Euler em 1736, que pode ser visto em ([BIGGS; LLOYD; WILSON, 1976](#)), com a resolução do famoso problema das sete pontes de Königsberg, o qual se resume em atravessar todas as sete pontes da cidade sem repetir nenhuma. Nas Figuras [1a](#) e [1b](#), temos a representação do problema e sua forma de grafo respectivamente.

O estudo das redes foi matematicamente consolidado apenas no final do século XX, quando bases de diversas redes reais começaram a ser disponibilizadas ([COSTA *et al.*, 2011](#)). Ademais, o surgimento da Teoria das Redes Complexas foi impulsionado pelo aumento do poder computacional de computadores convencionais, facilitando a análise de grandes quantidades de dados. A partir do início desse século, o estudo das redes ganhou cada vez mais interesse na comunidade científica e hoje abrange áreas desde a Biologia ([GRIGORIEV, 2003](#)), até Engenharia ([ŠUBELJ; BAJEC, 2011](#)) e Comunicação ([GUIMERA *et al.*, 2003](#)).

No entanto, apesar dos avanços obtidos, há ainda muitos desafios a serem superados no estudo das redes. Dentre eles, há uma limitação no fato da maioria das redes reais serem amostras

Figura 1 – Pontes de Königsberg

Fonte: Elaborada pelo autor.

de redes maiores. Por exemplo, no caso da interação de proteínas, ainda não é possível estimar o número de interações que ocorrem na natureza, sendo grande parte das conexões conhecidas apenas em experimentos *in vitro* (GRIGORIEV, 2003). Redes complexas obtidas de redes reais são frequentemente caracterizadas por falta de dados e ruído, isso se dá através de amostragens limitadas e do processo de aquisição (BOAS *et al.*, 2010).

Ademais, muitas redes são grandes demais para serem analisadas tanto do ponto de vista de armazenamento quanto de processamento computacional. Desse modo, faz-se necessário o uso de métodos de inferência estatística e amostragem de redes. Com esses métodos, podemos estimar propriedades das redes a partir das amostras, bem como determinar como se comportam amostras obtidas por diferentes métodos.

1.2 Objetivo

Esse trabalho tem como objetivo principal estudar os efeitos da amostragem em redes complexas, comparar os diferentes métodos de amostragem em redes reais, e com diferentes tamanhos de amostras entender como os mesmos se comportam na caracterização das medidas das redes. Além disso propomos também um estudo de classificação das redes com base em suas medidas e também utilizando amostras para assim verificar se há melhora nos resultados obtidos.

Esse estudo será importante para determinar o efeito da amostragem em redes complexas, bem como permitir o desenvolvimento futuro de estimadores não-viesados e consistentes para caracterizar a estrutura de redes complexas, visto que as medidas atuais são todas dependentes do tamanho e número de conexões das redes.

Uma vez que ao se comparar duas redes de tamanhos diferentes não há paralelos, mas se mantido as características da mesma em uma amostra, para a igualdade de dimensão, a comparação tornar-se-á possível.

1.3 Organização do Trabalho

Esse trabalho será organizado da seguinte maneira:

- No [Capítulo 2](#) iniciamos com a parte básica de redes complexas. Algumas definições de medidas de caracterização, assim como modelos de redes artificiais são apresentados nessa parte do trabalho, bem como os quatro métodos de amostragens utilizados no trabalho.
- O [Capítulo 3](#) introduz as redes reais escolhidas para serem estudadas, assim como as análises das amostras obtidas com os modelos antes apresentados.
- O [Capítulo 4](#) apresenta as técnicas de aprendizado de máquina utilizadas posteriormente na classificação de redes.
- O [Capítulo 5](#) é focado na classificação dos tipos de rede, Trabalhamos primeiramente com redes artificiais e, por fim, redes reais e amostras das mesmas.
- Finalmente, no [Capítulo 6](#) concluímos o trabalho apresentando os principais resultados e propomos futuros possíveis trabalhos.

REDES COMPLEXAS

Neste capítulo serão listadas todas as competências necessárias para a elaboração desse trabalho. Na [Seção 2.1](#) são introduzidos conceitos básicos sobre redes e sua ligação com teoria dos grafos, apresentando medidas e métodos para a obtenção de redes artificiais. Na [Seção 2.2](#) estão expostas as técnicas de amostragem que serão utilizadas no [Capítulo 3](#).

2.1 Teoria de Redes

2.1.1 Características Básicas

Inicialmente, vamos relacionar as redes complexas com a teoria de grafos. Existem diversos tipos de redes complexas, mas o que todas representam são interações entre elementos discretos. Assim, podemos representá-la como um grafo, em que os indivíduos são os vértices e as relações entre eles as arestas. O grafo pode ser expresso como o seguinte par ordenado $G = (V, E)$, em que V e E são conjuntos que compreendem os vértices e arestas, respectivamente. Os elementos do conjunto E são expressos também como um par ordenado da seguinte forma $e = (u, v)$, i.e. aresta (*edge*) entre os vértices u e v , com $e \in E$ e $u, v \in V$, ou seja, a aresta e conecta o vértice u ao v e são chamados de vizinhos ou adjacentes.

Construído o conjunto de arestas precisamos saber qual a natureza das mesmas, definindo se são direcionadas ou não. Se a direção da conexão é necessária, temos $e_1 = (u, v) \neq e_2 = (v, u)$ como no caso de redes de *e-mails*. Em se tratando de não direcionada, temos $e_1 = (u, v) = e_2 = (v, u)$, como no caso de redes de amizade.

Outra variação das arestas são quando adicionamos pesos, incluindo assim um vetor P no par ordenado do Grafo $G = (V, E, P)$, que representam a força da conexão entre pares de vértices. Neste trabalho vamos utilizar apenas grafos simples, com arestas não direcionadas, que não possui em múltiplas arestas entre dois vértices e nem *selfloops*, ou seja, uma aresta que conecta o vértice a ele mesmo. Na [Figura 2](#) são mostrados alguns exemplos de grafos.

Figura 2 – Tipos de Grafo

Fonte: Elaborada pelo autor.

Nota – Um Multi-grafo apresenta múltiplas conexões mas não *selfloop*.

A conectividade de um grafo G pode ser expressa por uma matriz binária $N \times N$ em que $N = |V|$, e os elementos A_{ij} da matriz são da seguinte forma:

$$A_{ij} = \begin{cases} 1, & \text{se } \{i, j\} \in E \\ 0, & \text{caso contrário.} \end{cases}$$

em que os nomes de linhas e colunas $1, 2, \dots, N$ são os vértices em V e os valores das entradas da matriz indicam a existência de aresta entre os vértices. Essa estrutura é chamada matriz de adjacência. Para um grafo ser não direcionado temos que $A_{ij} = A_{ji}$, exigindo assim simetria como pode ser vista na [Figura 3](#), que mostra dois grafos com suas respectivas matrizes, sendo um chamado de completo pelo fato de todos os vértices possuírem todas as arestas possíveis entre si.

2.1.2 Medidas

Com o intuito de caracterizar os diferentes tipos de grafos, foram desenvolvidos diversos tipos de medidas de redes ([COSTA et al., 2007](#); [KOLACZYK, 2009](#)). Nessa seção serão apresentadas algumas medidas importantes que vamos considerar em nosso trabalho.

A medida mais simples para caracterizar uma rede é o grau de cada vértice i , k_i , $i = 1, \dots, N$. Ele é caracterizado como a quantidade de arestas que incidem no vértice i . Com essa medida pode-se ver quão importante esse vértice é para a rede local e globalmente, pois quando

Figura 3 – Matriz de Adjacência

Fonte: Elaborada pelo autor.

Nota – O nome grafo completo se dá pelo fato de todos os vértices se conectarem entre si, completando assim a matriz de adjacência com exceção da diagonal principal, pois não estamos interessados em autoconexões.

temos que k_i é muito alto com relação aos outros vértices, ele é chamado de *HUB*. Uma vez calculado o grau de todos os vértices da rede, a média amostral entre eles $\langle k \rangle$ (Subseção 2.1.2) dá a noção de quão densa e conectada é a rede,

$$\langle k \rangle = \sum_{i=1}^N \frac{k_i}{N}.$$

Com isso também pode-se construir a distribuição do grau, ou seja $P(k)$, que é a probabilidade de retirar o grau k aleatória e uniformemente. Como exemplo, na Figura 4 temos a distribuição do grau da rede de *e-mails* na *University Rovira i Virgili*, Espanha, estudada por Guimera *et al.* (2003).

Figura 4 – Distribuição do grau da rede de *e-mails* coletada na *University Rovira i Virgili*.

Fonte: Elaborada pelo autor.

Outra medida importante relacionada ao número de conexões é o segundo momento da

distribuição do grau $\langle k^2 \rangle$ (Equação 2.1), que nos dá a noção de variabilidade do grau, podendo diferenciar duas redes com mesmo $\langle k \rangle$ e auxiliando também a identificação de *HUBS*,

$$\langle k^2 \rangle = \sum_k k^2 \times P(k). \quad (2.1)$$

Clustering coefficient é uma medida de aglomeração que é calculada por meio de fração de triplas de um grafo que estão realmente conectadas. Por exemplo, uma tripla conectada no caso de amizades seria o amigo de seu amigo também é seu amigo formando assim um triângulo (KOLACZYK, 2009). Nesse trabalho vamos utilizar duas variantes dessa medida. A primeira, proposta por Watts e Strogatz (1998), é uma medida local a qual pode ser calculada da seguinte forma:

$$cl(v) = \frac{\tau_{\Delta}(v)}{\tau_3(v)},$$

em que $\tau_{\Delta}(v)$ é o total de triângulos formados no grafo G ao qual o vértice $v \in V$ faz parte e $\tau_3(v)$ o número de triplas às quais v está, ou seja, $\tau_3(v) = \binom{k_v}{2}$, a mesma medida tem sua versão global como a média dos $cl(v)$:

$$cl(G) = \frac{1}{|V'|} \sum_{v \in V'} \frac{\tau_{\Delta}(v)}{\tau_3(v)}$$

com $|V'|$ o tamanho de $V' \subset V$ tal que $v \in V'$ se e somente se $k_v > 2$.

A segunda variante da medida é a transitividade (AMARAL; OTTINO, 2004):

$$T(G) = \frac{3\tau_{\Delta}(G)}{\tau_3(G)},$$

em que $\tau_{\Delta}(G)$ é o total de triângulos formados no grafo G e $\tau_3(G)$ o número de triplas.

Eigenvector centrality ou centralidade de autovetor, mensura a centralidade de um vértice levando em consideração se está ou não conectado a outros importantes vértices. Inicialmente para todos os vértices são atribuídos $X_0(i) = 1$ e, a partir daí, realizamos a iteração:

$$X_{t+1}(i) = \sum_{j=1}^N A_{ij} X_t(j) = \sum_{j \text{ é vizinho de } i}^N X_t(j),$$

de forma que $t \rightarrow \infty$ os valores (normalizados) de $X(i)$ são considerados centrais à medida que se aproximem de 1.

Betweenness centrality (FREEMAN, 1978), mede a centralidade de um vértice pelo número de menores caminhos entre um par de vértices que passam por ele.

$$b(v) = \sum_{s \neq v \neq t} \frac{\delta_{st}(v)}{\delta_{st}},$$

em que, δ_{st} é número de menores caminhos entre os vértices s e t , já o $\delta_{st}(v)$ não os menores caminhos entre s e t que passam por v .

Entropia de Shannon (SHANNON, 1948).

$$S = - \sum P(k) * \log(P(k)),$$

$P(k)$ é a probabilidade do grau k na distribuição do grau.

Page Rank (BRIN; PAGE, 1998), é um algoritmo patentado pela *Google*, que basicamente representa a probabilidade de uma pessoa chegar a um determinado site clicando aleatoriamente em links.

$$p(v) = \sum_{s \in A_v} \frac{p(s)}{B(s)},$$

$p(s)$ é o PageRank de uma página s com $s \in A_v$, A_v é o conjunto de páginas que referenciam v e por fim $B(s)$ número de reverencias de s .

Closeness centrality (FREEMAN, 1978), mede a centralidade do vértice através da soma dos menores caminhos dele com todos os outros sendo assim quem obtiver a menor soma mais central será.

$$c(v) = \frac{1}{\sum_s (d(s, v))},$$

Em que, $d(s, v)$ é a distância entre s e v .

2.1.3 Modelos de Redes

Diversos modelos foram desenvolvidos para gerar redes complexas com determinadas propriedades. Vamos discutir os modelos mais famosos e historicamente relevantes para a área.

2.1.3.1 Grafo Aleatório

Erdős e Rényi (1959) trataram grafos como objetos estritamente estocásticos, criando assim grafos aleatórios. Para a construção do Grafo, iniciamos com um conjunto V de vértices sem conexão e que gradativamente dois a dois são escolhidos sem repetição e ligados ou não com uma probabilidade p , criando assim o conjunto E de arestas. Através desse método obtemos um grafo com sua distribuição do grau como binomial e, de modo que $|V| \rightarrow \infty$ e $p \rightarrow 0$, temos que a mesma segue a distribuição Poisson:

$$P(k) = \frac{(|V|p)^k e^{-|V|p}}{k!},$$

$$\langle k \rangle \approx (|V| - 1)p.$$

Por Costa *et al.* (2007), o grafo gerado tem as seguintes características, distribuição do grau homogênea, propriedade *small-world* (pequena distância entre os vértices), mas uma baixa média do coeficiente de aglomeração.

2.1.3.2 Small-World

O modelo de *small-world* de Watts e Strogatz (1998) veio suprir uma deficiência dos grafos aleatórios, que seria uma maior presença de triângulos nas conexões, ou seja, caminhos fechados de ordem três encontrados com frequência em redes reais. Para gerar a referida rede inicia-se com um grafo regular de $|V| = N$ vértices e cada um deles com k ligações não direcionadas sendo $k = 2j$ e $j \in \mathbb{N}$ respeitando sempre a seguinte desigualdade $N \gg k \gg \ln(N) \gg 1$. Com isso, as arestas são reconectadas ou não com uma probabilidade p , o que traz a característica aleatória à rede.

2.1.3.3 Redes sem escala

Mais recentemente no trabalho de Barabási e Albert (1999), os pesquisadores fizeram uma importante descoberta ao estudarem a Rede Mundial de Computadores (*World Wide Web*). Notaram que a distribuição do grau era sem escala (*scale-free*),

$$P(k) \propto k^{-\alpha}. \quad (2.2)$$

Segundo (COSTA *et al.*, 2007), o valor de α para redes reais giram em torno de 2 e 3. Com a constatação de que essas redes seguiam essa distribuição foi desenvolvido o modelo *scale-free* BA (BARABÁSI; ALBERT, 1999), que se estrutura da seguinte forma: um pequeno grupo com N_0 vértices é criado e a ele adicionado um vértice com $m < N_0$ arestas e cada aresta é conectada aos vértices já existentes levando em consideração a conexão deles, ou seja, prioriza vértices mais conectados. Por exemplo, a probabilidade de i se conectar com j é:

$$P(ij) = \frac{k_j}{\sum_{u \in V} k_u}. \quad (2.3)$$

Como pode-se ver esse método tende a priorizar a conexão com os vértices adicionados inicialmente, aumentando seus graus com as iterações gerando assim poucos vértices muito conectados (*HUBS*) e em sua maioria com poucas conexões.

2.2 Amostragem

Definidas as redes, podemos dar o enfoque no objetivo do trabalho que são métodos de amostragens para a obtenção das sub-redes. Como visto anteriormente, uma rede pode ser representada por um grafo $G = (V, E)$, da mesma forma ao se retirar uma amostra, obtemos um subgrafo $G' = (V', E')$ em que $V' \subset V$ e $E' \subset E$. Todas as rotinas foram desenvolvidas em linguagem R (R Core Team, 2018) com o auxílio do pacote iGraph (CSARDI; NEPUSZ, 2006).

2.2.1 Amostragem Aleatória Simples Sem Reposição (AASs)

Três dos métodos de amostragem em redes introduzidos a seguir utilizam a ideia de amostragem aleatória simples sem reposição. Então, com o intuito de melhorar a compreensão

futura o mesmo será introduzido. Essa técnica segue o algoritmo que também pode ser visto em (BUSSAB; BOLFARINE, 2005).

- i. A população é numerada de 1 a N , de acordo com o sistema de referências, $U = 1, \dots, N$;
- ii. Utilizando o R ou outro *software* é sorteado um dentre os N elementos da população aleatoriamente, com a mesma probabilidade;
- iii. Sem repor o item amostrado, repete-se o procedimento até que n unidades tenham sido sorteadas.

Definido, isso temos os seguintes resultados:

Teorema 1. Com a AASs temos que o fato de um elemento i da população aparecer ou não na amostra pode ser expresso pela variável aleatória Z_i , em que Z_i assume os valores 1 ou 0. Essa variável segue a distribuição Bernoulli, com probabilidade de sucesso n/N , ou seja $Z_i \sim B(1, n/N)$, e satisfaz as seguintes relações:

$$\mathbb{P}(Z_i = 1) = \frac{n}{N} \quad \text{e} \quad \mathbb{P}(Z_i = 0) = 1 - \frac{n}{N},$$

Assim,

$$\begin{aligned} \mathbb{E}[Z_i] &= \frac{n}{N}, \\ \text{Var}[Z_i] &= \frac{n}{N} \left(1 - \frac{n}{N}\right), \\ \pi_i &= \frac{n}{N}, \\ \pi_{ij} &= \frac{n(n-1)}{N(N-1)} \end{aligned}$$

Também temos que a média amostral

$$\bar{Z} = \frac{1}{N} \sum_{i=1}^N z_i = \hat{\mu} \tag{2.4}$$

é um estimador não-viesado para a média populacional μ .

2.2.2 Subgrafo Induzido (AV)

O primeiro método de amostragem é o subgrafo Induzido (KOLACZYK, 2009), que para abreviação chamaremos Amostragem por Vértice (AV), o qual é baseado na AASs. Ou seja, amostramos n vértices tais que $n \leq N$ elementos do conjunto de vértices V do grafo original, obtendo assim V' . Após isso, construímos então o conjunto de arestas E' com base nas ligações

Figura 5 – Exemplo de um subgrafo Induzido.

Fonte: Elaborada pelo autor.

do grafo original presentes no conjunto E , ligando os vértices amostrados como mostrado na Figura 5.

Assim como na AASs, temos que a probabilidade de retirar o vértice i do grafo é:

$$\pi_i = \frac{n}{N},$$

em que $n = |V'|$ e $N = |V|$. A probabilidade de retirar uma aresta $e = (i, j)$ com $i, j \in V$ é dada por:

$$\pi_{ij} = \pi_i(P(j = 1 | i = 1)) = \frac{n(n-1)}{N(N-1)}. \quad (2.5)$$

Uma vez obtida essa relação com AASs, o comum é utilizar a teoria já consolidada dessa técnica como visto em (BUSSAB; BOLFARINE, 2005), mas o cuidado a ser tomado aqui é que, para o grau, por exemplo, uma vez amostrados os vértices e construído o conjunto de arestas, obtemos $k_i^* \leq k_i$. Em outras palavras, não amostramos os reais valores de grau da rede, sendo assim o estimador apresentado na Equação 2.4 é viesado como será visto no Capítulo 3. Uma alternativa é o usar o estimador de Horvitz-Thompson para a inferência do total das arestas, que é dada por:

$$2\hat{N}_e = \sum_{i \in V'} \frac{k_i^*}{\pi_i}, \quad \langle \hat{k} \rangle = \frac{2\hat{N}_e}{N} \quad (2.6)$$

Como pode ser visto, o pré-requisito para essa amostragem é conhecer previamente o valor de N . A seguir, no Código-fonte 1, disponibilizamos a rotina em linguagem de programação R.

Código-fonte 1 – Amostragem por Vértice

```
1: require(igraph)
2: AV<-function(n,G){
3:   #n tamanho da amostra
4:   #G grafo pode ser substituído com a lista de nomes dos vértices
   rtices
```

```

5:  V=names(V(G))
6:  V0<-sample(V,n,replace=F)
7:  return(V0)
8: }
```

Na saída da função temos o vetor V' com todos os vértices amostrados, então são utilizados ele e G o grafo original na função `induced_subgraph(G,V0)` já implementada no pacote `iGraph`, a fim de obter o subgrafo induzido.

2.2.3 Subgrafo Induzido com Vizinhaça(AVV)

Da mesma forma que o subgrafo induzido, esse método, ou como vamos chamá-lo Amostragem por Vértice com Vizinhaça (AVV), amostra os vértices da forma AASs da população V , mas com a diferença de que após amostrado o vértice i incluímos na amostra também seus k_i vizinhos, sendo assim preservado o grau do vértice i como pode ser visto na [Figura 6](#). É amostrado então mais um vértice dessa vez de $V - \{i\}$ desta forma até o valor de n desejado.

Figura 6 – Exemplo do subgrafo induzido com vizinhaça.

Fonte: Elaborada pelo autor.

Na [Figura 6](#) os vértices vermelhos são os amostrados da forma AASs e os amarelos as vizinhanças. Como pode-se ver existe um vértice vermelho sem toda a vizinhaça. Isso se dá pelo fato do tamanho da amostra desejada, limitando assim o número de vértices amostrados.

Código-fonte 2 – Amostragem por Vértice com Vizinhaça

```

1: require(igraph)
2: AVV<-function(n,G){
3:   V<-names(V(G))
4:   aux=sample(V,length(V),replace=F)
5:   o=1
6:   V0=c()
```

```

7:  p=c()
8:  repeat{
9:    if(length(unique(V0))>=n){break}
10:   u=aux[o]
11:   p<-c(p,u)
12:   Nu=names(neighbors(G,u))
13:   V0=c(V0,u,Nu)
14:   o=o+1
15: }
16: return(list(unique(V0)[1:n],p))
17: }

```

Como saída do [Código-fonte 2](#) temos uma lista com duas entradas, na primeira temos o vetor total de vértices amostrado V' e na segunda entrada os vértices amostrados aleatoriamente. Essa segunda saída tem a finalidade de facilitar a construção da [Figura 6](#) (vértices vermelhos).

2.2.4 Subgrafo Induzido Snowball (SB)

Da mesma forma que a técnica anterior ([Subseção 2.2.3](#)), o método de Amostragem *SnowBall* (SB) ([KOLACZYK, 2009](#)) amostra vértices da forma AASs da população V , formando assim o conjunto $|V_0| = n$. Então, os próximos passos são no formato de ondas, amostrando as vizinhanças dos vértices em V_0 e então vizinhanças dos vizinhos e assim por diante como pode ser visto na [Figura 7](#). Para simplificar vamos utilizar o caso particular, em que $n=1$, ou seja, é amostrado apenas 1 vértice no primeiro passo.

Figura 7 – Exemplo do subgrafo induzido Snowball.

Fonte: Elaborada pelo autor.

Na [Figura 7](#) o vértice vermelho é o amostrado da forma AASs e os amarelos sua vizinhança, os vértices rosas a segunda onda de vizinhos. Alguma vizinhança pode não ser amostrada por completa, pelo fato do tamanho da amostra desejada, limitando assim o número de vértices amostrados.

Código-fonte 3 – Amostragem *SnowBall*

```

1: require(igraph)
2: SNOW<-function(L,N,G){
3:   n0<-sample(names(V(G)),1)
4:   u<-n0
5:   V0<-c(n0)
6:   count<-1
7:   while (length(unique(V0))<N) {
8:     Nu<-names(L(G,V0[count]))
9:     V0<-unique(c(V0,Nu))
10:    count<-count+1
11:  }
12:  if(length(V0)>N){
13:    dif=length(V0)-N
14:    V0=V0[-c((length(V0)-dif+1):length(V0))]
15:  }
16:
17:  return(list(unique(V0[1:N]),n0))
18: }

```

Como saída do [Código-fonte 3](#) temos uma lista com duas entradas, na primeira temos o vetor total de vértices amostrado V' e na segunda entrada o vértice amostrado aleatoriamente.

2.2.5 Subgrafo Incidente (AL)

Uma outra forma de aplicar a AASs é usando um subgrafo incidente, que também pode ser visto em ([KOLACZYK, 2009](#)) (chamaremos esse método de Amostragem por *Links* (AL)). Diferente dos métodos anteriores, o enfoque deste são as arestas. Consistindo em n retiradas sem reposição do conjunto E , obtemos o conjunto E' e, conseqüentemente, os extremos (vértices) das arestas construindo o subconjunto V' ([Figura 8](#)).

Como a amostra segue AASs, temos que a probabilidade de amostrar uma aresta é:

$$\pi_{ij} = \frac{n}{N_e}, \quad (2.7)$$

em que $N_e = |E|$, e por sua vez a probabilidade de se amostrar um vértice se torna menos trivial,

$$\begin{aligned}
 \pi_i &= \mathbb{P}(\text{Amostrar vértice } i) \\
 &= 1 - \mathbb{P}(\text{não amostrar arestas de } i) \\
 &= \begin{cases} 1 - \frac{\binom{N_e - d_i}{n}}{\binom{N_e}{n}}, & \text{se } n \leq N_e - d_i, \\ 1, & \text{se } n > N_e - d_i. \end{cases} \quad (2.8)
 \end{aligned}$$

Figura 8 – Exemplo do subgrafo Incidente.

Fonte: Elaborada pelo autor.

Como se pode notar, nesse caso necessitamos ter acesso ao grau real do vértice i assim com o total de arestas.

Código-fonte 4 – Amostragem por *Links*

```

1: require(igraph)
2: AL<-function(n,E){
3:   #n tamanho da amostra
4:   #G grafo original
5:   #E lista com ligações
6:
7:   a<-sample((1:length(E[,1])),n)
8:   E0<-E[a,]
9:   return(E0)
10: }
```

O [Código-fonte 4](#) nos devolve uma matriz $n \times 2$, em que cada linha representa uma aresta amostrada. Com isso, podemos construir o subgrafo com a utilização da função `graph_from_data_frame(E0,directed=FALSE)` do pacote `igraph`.

2.2.6 Subgrafo por Caminhada Aleatória com Vizinhança (ACAV)

A amostragem por caminhadas aleatórias (Amostragem por Caminhada Aleatória com Vizinhança (ACAV)) inicia com um vértice i escolhido aleatoriamente dentre os possíveis da rede. A seguir, dentre os seus vizinhos, é escolhido um vértice de forma aleatória. O processo é executado até completarmos o tamanho desejado da amostra (desconsiderando a reincidência de vértices já amostrados). Na [Figura 9](#) temos um exemplo da amostragem em que o vértice vermelho é o primeiro a ser amostrado no início da caminhada aleatória.

Figura 9 – Exemplo de um subgrafo por caminhada aleatória.

Fonte: Elaborada pelo autor.

Uma desvantagem desse método é uma tendência a amostrar *HUBS* ou seja vértices muito conectados, viesando assim as estimativas. Isso ocorre porque a probabilidade estacionária de cada vértice é proporcional ao seu grau, como pode ser visto em [Stutzbach et al. \(2009\)](#).

Código-fonte 5 – Amostragem por caminhada aleatória

```

1: require(igraph)
2: ACAV<-function(n,G){
3:   #L lista com vizinhança
4:   #N total de vertices
5:   #G grafo
6:   #n0 vertice inicial da caminhada
7:   n0<-sample(names(V(G)),1)
8:   u<-n0
9:   V0<-c(n0)
10:  while (length(unique(V0))<n) {
11:    Nu<-names(neighbors(G,u))
12:    V0<-unique(c(V0,Nu))
13:    u<-sample(Nu,1)
14:  }
15:  return(list(unique(V0[1:n]),n0))
16: }
```

A saída do [Código-fonte 5](#), como os métodos baseados em vértices, libera uma lista com duas saídas, na primeira saída encontra-se o vetor V' com vértices amostrados e na segunda saída o vértice inicial (em vermelho na [Figura 9](#)). Assim como os outros métodos, é utilizada a função `induced_subgraph` do pacote `igraph`.

AMOSTRAS E ANÁLISES

Esse capítulo tem como finalidade por em prática a teoria apresentada no [Capítulo 2](#) utilizando redes reais obtidas em banco de dados na Internet, com o intuito de observar os efeitos causados pelos métodos de amostragem aos subgrafos.

3.1 Dados

Inicialmente, foram selecionadas quatro redes reais, listadas na [Tabela 1](#). Esse número pode aumentar futuramente para manter o cuidado em selecionar redes com diferentes características para tornar o resultado mais abrangente possível.

Tabela 1 – Redes Reais

Redes	Vértices	Arestas	$\langle k \rangle$	Fonte
Facebook	4039	88234	43,69101	(LESKOVEC; KREVL, 2014)
Emails	1133	5451	9,622242	(KUNEGIS, 2017)
Rodovias	1174	1417	2,413969	(KUNEGIS, 2017)
R Brasil - Programadores	1479	4024	5,441515	-

Fonte: Dados da pesquisa.

A rede *Facebook* foi coletada e estudada por [Leskovec e Mcauley \(2012\)](#), a qual tem como base as amizades no *Facebook* em que os vértices são usuários que concordaram em divulgar seus círculos de amizade na rede social. Esses dados foram obtidos por meio de um aplicativo para o *Facebook*.

A segunda rede é de comunicação de *e-mails* na *University Rovira i Virgili* em *Tarragona*, no sul da Catalunha na Espanha. Sua análise e coleta se dá a [Guimera et al. \(2003\)](#). Nela, os vértices são usuários e cada aresta representa que, pelo menos, um *e-mail* foi enviado entre dois usuários, a direção ou o número de *e-mails* não foram armazenados.

Rodovias foi estudada por Šubelj e Bajec (2011), e se baseia na malha viária da Europa em que os vértices são as cidades e as rodovias que ligam duas cidades as arestas.

A última rede é a baseada nas publicações do **GRUPO R BRASIL - PROGRAMADORES** (2017), a qual foi possível a mineração graças ao aplicativo para *Facebook* Netvizz desenvolvido por Rieder (2013). Este disponibiliza para *download* o conjunto de arestas referente às interações (curtidas e comentários) da quantidade desejada de publicações, que no caso foram escolhidas as últimas 200 no dia 8 de agosto de 2017. Por sua vez, o conjunto de arestas continha *selfloops* e *parallels*, após a simplificação foi montada a rede de interação entre os usuários.

Figura 10 – Redes Reais

Fonte: Elaborada pelo autor.

Na [Figura 10](#) é possível ver a representação das quatro redes apresentadas, a construção dessas imagens se deu ao *software* Gephi desenvolvido por [Bastian *et al.* \(2009\)](#).

3.2 Amostras

Para cada rede foram feitas amostras de 10 tamanhos diferentes, começando em 10% até o censo (100%) igualmente espaçadas. Para cada uma dessas configurações foram retiradas 1000 amostras, por sua vez cada uma destas dá origem a um subgrafo, que então é tratado como uma rede nova, foram obtidas as seguintes medidas: grau médio, transitividade, distribuição do grau e autovetor principal (*Eigenvector Centrality*). Uma vez calculados o grau médio e a transitividade foram montados então intervalos de 95% de confiança para cada porcentagem de amostra.

Figura 11 – Grau Médio

Fonte: Elaborada pelo autor.

A [Figura 11](#) mostra os intervalos de confiança para o grau médio para as quatro abordagens, sendo possível observar que ambos os métodos baseados em vizinhança, AVV e ACAV,

tendem a obter maiores valores que o desejado na maioria dos casos, com exceção da rede Rodovias Europeias, que apenas ACAV superestima acima de 30%. Como dito anteriormente, a maior probabilidade de obter um *HUBS* desses métodos tende a superestimar o grau médio. Como a rede Rodovias Europeias não contém vértices discrepantes com relação ao grau, isso pode explicar o comportamento.

Figura 12 – Transitividade

Fonte: Elaborada pelo autor.

Por sua vez, a [Figura 12](#) nos dá os intervalos de confiança da medida transitividade, a qual mostra um melhor resultado para o método AV em todas as redes, o método ACAV para o Facebook ([Figura 12a](#)) e as Rodovias Europeias ([Figura 12d](#)) também se aproxima do valor, ainda assim não compreendendo as bandas de confiança. O que se nota é que a abordagem AL subestima pelo fato da baixa probabilidade de se obter triângulos na amostragem, enquanto os baseados em vizinhança superestimam.

Figura 13 – Estatística D - K.S. para distribuição do Grau

Fonte: Elaborada pelo autor.

A Figura 13 e a Figura 14 são baseadas na estatística D proposta, que pode ser vista em Conover (1972), e que basicamente é a maior diferença em módulo entre o gráfico da distribuição acumulada real com a amostrada. A Figura 13 é baseada na distribuição do grau, observando a mesma pode-se notar que, com exceção para a rede "R Brasil - Programadores", o método que representa menor estatística D é o ACAV seguido pelo AVV, já para a exceção os mais eficientes são AL e AV.

Figura 14 – Estatística D - K.S. para Autovetor Principal

Fonte: Elaborada pelo autor.

Com Autovetor Principal na [Figura 14](#) vemos que para a rede *Facebook* essa diferença tem um menor resultado para AL e seu pior para ACAV mantendo a diferença maior que o valor de 10% da AL até 70%, e assim perpetua pelas próximas redes até a rede *Rodovias Europeias*, na qual há uma inversão onde o melhor método é ACAV (ainda assim bem falho), chamando a atenção para os métodos AL e AV que iniciam no valor 1 (maior distância possível).

Embora os resultados da Estatística D pareçam estar próximos de zero, assim seguindo a mesma distribuição da rede completa, nas [Figuras 15 e 16](#) vemos os valores p para o teste Kolmogorov-Smirnov indicando a rejeição da hipótese nula, ou seja, que sigam a mesma distribuição.

Figura 15 – Valor p - K.S. para distribuição do Grau

Figura 16 – Valor p - K.S. para Autovetor Principal

Fonte: Elaborada pelo autor.

Para critério de comparação foi escolhida a 50ª amostra de cada modelo, com o equivalente a 40% da rede real para assim, juntamente com as distribuições reais, identificar as principais divergências com as obtidas com os subgrafos.

Figura 17 – Distribuição do Grau das amostras comparado com a real

Fonte: Elaborada pelo autor.

Observando a [Figura 17](#), a qual se refere à distribuição do grau, os modelos AL e AV parecem ser os mais próximos da original para essa amostra, ainda que pela natureza de amostragem se obtenha menos vértices muito conectados.

Figura 18 – Autovetor principal das amostras comparado com o real

Fonte: Elaborada pelo autor.

A Figura 18 representa densidade aproximada do autovetor principal. Para as três primeiras figuras, o método AL parece se aproximar da real, porém cabe ressaltar que o método AL pode ser utilizado em casos específicos, quando temos acesso a todo o conjunto de arestas por exemplo ligações, *emails*, etc. E por se tratar de amostragem por arestas, a quantidade de vértices varia podendo assim compreender toda o conjunto V sem antes completar E , por exemplo, as amostras escolhidas acima de 40% da rede Rodovias Europeias é aproximadamente 415 vértices, com o método AL são amostrados 713.

3.3 Modelo AV

Até aqui foram apenas consideradas as próprias medidas como estimadores, nessa seção será considerado outro estimador para o grau médio no modelo AV, mostrando assim uma possível continuação para este trabalho.

O estimador $\langle k_2 \rangle$ se difere do já visto por utilizar o valor real do grau de cada vértice, podendo assim ser usado como na AASs. Basicamente, à medida que os vértices são amostrados guardamos a dimensão de sua vizinhança, esses vértices estando ou não na amostra. Por exemplo no *Facebook* amostramos uma pessoa i e sabemos que ela tem k_i amigos. Sua fraqueza é que essa abordagem serve apenas para a estimação de medidas relacionadas ao grau e a necessidade de conhecer a vizinhança do vértice amostrado, o que às vezes não é possível.

Figura 19 – Estimadores para o Modelo AL

Fonte: Elaborada pelo autor.

Na Figura 19 o boxplot $\langle k_1 \rangle$ é referente ao já mostrado anteriormente na Figura 11. Para o modelo AV e o $\langle k_2 \rangle$, a melhora é significativa porém apresentando uma maior variância nos

valores obtidos com as amostras.

APRENDIZADO DE MÁQUINA

Aqui, vamos dar uma breve introdução aos métodos de aprendizado de máquina que serão utilizados no próximo capítulo. O uso deles se dá a classificação de redes artificiais e reais por seus algoritmos de geração e tipos respectivamente.

4.1 Redes Neurais

Em Redes Neurais, o *Multilayer Perceptron* é a técnica utilizada quando tratamos de casos não linearmente separáveis. Trata-se de um algoritmo de aprendizado supervisionado que funciona como na [Figura 20](#): é dividido em camadas, entre elas, de entrada (*Input*), camada escondida (*Hidden*) e a de saída (*Output*).

Figura 20 – Rede Neural

Fonte: Elaborada pelo autor.

Uma vez iniciado o processo, o conjunto de treinamento é adicionado à camada de entrada, segregado pelas variáveis $X = (x_1, x_2, \dots, x_m)$, e então cada ligação é ponderada como

pode ser visto na [Equação 4.1](#). Em cada camada, além dos neurônios utilizados com os dados, é adicionado um neurônio com o valor 1 chamado de *bias* com seu próprio peso θ_i . Nos neurônios ocorre um processo simples que é a ativação s_i em relação a a seguinte entrada net_i :

$$net_i = \sum_{j \in pred(i)} w_{ij} \times s_j - \theta_i \quad (4.1)$$

Na equação acima $pred(i)$ denota o conjunto que neurônios que precedem a camada atual, w_{ij} é o peso da conexão entre o neurônio j com o i , que pode ser atribuído de diversas formas influenciando assim os resultados, mas isso é contornado utilizando o método *Backpropagation* ([RIEDMILLER, 1994b](#)), que atualiza os pesos de acordo com os resultados obtidos no conjunto de treinamento. Geralmente, a ativação da unidade i em s_i é dada pela função logística:

$$s_i = f_{log}(net_i) = \frac{1}{1 + e^{-net_i}} \quad (4.2)$$

Tudo isso pode ser visto com mais detalhes em ([RIEDMILLER, 1994a](#)). A função usada foi `neuralnet` que faz parte do pacote `neuralnet` ([FRITSCH; GUENTHER, 2016](#)). A configuração usada foi de duas camadas escondidas, a primeira com 5 neurônios e segunda com 3.

4.2 Naive Bayes

O algoritmo de *Naive Bayes* é baseado, como o próprio nome diz, no teorema de Bayes com a suposição de que cada par de variáveis é independente condicionado ao valor da classe. Ele é considerado um ótimo algoritmo para aprendizado de máquina, mesmo os exemplos reais raramente satisfazendo essa suposição, como pode ser visto em [Zhang \(2004\)](#). A seguir, forma de como são calculadas as probabilidades condicionais, com $x_1, \dots, x_n$ sendo as variáveis explicativas e y a variável resposta.

$$P(y|x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n|y)}{P(x_1, \dots, x_n)} \quad (4.3)$$

Da suposição de independência, segue:

$$P(x_i|y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i|y) \quad (4.4)$$

E, assim, para todos os i 's,

$$P(y|x_1, \dots, x_n) = \frac{P(y) \prod_{i=1}^n P(x_i|y)}{P(x_1, \dots, x_n)} \quad (4.5)$$

O pacote usado com as função utilizada foi o `e1071` ([MEYER et al., 2018](#)).

4.3 KNN

O Knn (*k-Nearest Neighbors*) é um algoritmo de classificação supervisionada, i.e, o mesmo classifica as observações a partir de um banco de treinamento com a variável resposta devidamente preenchida. A forma com que ele classifica essas observações é representando-as como pontos no espaço definidos pelos atributos; dado isso, é definida uma métrica, no caso Euclidiana, e basicamente o algoritmo calcula a distância entre o ponto não classificado e todos os outros no conjunto de treinamento. Assim, a maioria dos k mais próximos pontos ditam sua classificação. Na [Figura 21](#) vemos como o algoritmo funciona, para classificar o ponto de interrogação seleciona os elementos mais próximos conhecidos. No caso com $k = 3$ seria classificado como quadrado.

Figura 21 – Exemplo KNN

Fonte: Elaborada pelo autor.

A função na qual usamos é implementada a partir de [Ripley \(1996\)](#), essa com o nome de `knn` que pode ser encontrada no pacote `class` ([VENABLES; RIPLEY, 2002](#)). O valor de k escolhido foi $k = 3$, o qual se mostrou com melhores resultados entre as classificações.

4.4 Florestas Aleatórias

Para entender a Floresta Aleatória precisamos aprender árvores de decisão. Esse método classifica suas observações analisando cada variável, dividindo assim o problema maior em problemas menores. Criando assim um roteiro baseado nos valores das variáveis, como exemplo temos a [Figura 22](#), uma árvore de decisão para classificar os tipos de redes, que utiliza das medidas de caracterização para isso.

Figura 22 – Exemplo de Árvore de Decisão

Medidas: **Shannon** = Entropia de Shannon, **eigen** = *Eigenvetor*, **transiti** = Transitividade, **ki** = i-ésimo momento do grau, **m** = Número de arestas da rede, **eigenval** = *Eigenvalor*.

Classes (tipos de redes artificiais): **BA** = "Barabási-Albert", **ER** = "Erdős-Rény", **WS** = "Watts-Strogatz" (p=0.01 e p=0.1), **WA** = "Waxman", **LFR** = "Lancichinetti-Fortunato-Radicchi".

Fonte: Elaborada pelo autor.

Como o nome diz o método de Florestas Aleatórias é um conjunto grande de Árvores de Decisão, em que cada árvore é constituída com uma amostra com reposição do mesmo tamanho que o conjunto de treinamento. Então a árvore é construída, não mais sobre o conjunto de treinamento, e sim em uma amostra aumentando ligeiramente seu viés com relação a árvore sem aleatoriedade, mas a partir do momento que o resultado final é uma média, sua variância diminui tornando um melhor modelo global. A função `randomForest` (LIAW; WIENER, 2002)

foi a utilizada para esse trabalho e segue a implementação de (BREIMAN, 2001). O número de árvores usadas foi o *default* da função que é 500 árvores.

4.5 Métricas de Avaliação

Quando usamos um modelo para prever classes, ele atribui probabilidades as observações, assim temos uma ideia como classificá-la. Uma vez classificadas (predito) podemos montar a matriz de confusão (com uma base de teste):

		Predito	
		Positivo	Negativo
Verdadeiro	Positivo	VP	FN
	Negativo	FP	VN

Em que,

- VP = Verdadeiro Positivo - positivos classificados corretamente
- VN = Verdadeiro Negativo - negativos classificados corretamente
- FP = Falso Positivo (Erro do tipo I) - negativos classificados como positivos
- FN = Falso Negativo (Erro do tipo II) - positivos classificados como negativos.

Através desses valores, podemos calcular algumas medidas com o intuito de mensurar a predição do modelo. A mais usada é a Acurácia, e foia a que escolhemos nesse trabalho.

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN};$$

Indica a performance geral do modelo, dá ideia de todas as possibilidades quantas classificou corretamente;

$$\text{Precisão} = \frac{VP}{VP + FP};$$

Dos valores positivos preditos quais são realmente positivos;

$$\text{Recall} = \frac{VP}{VP + FN};$$

Da classe verdadeira positiva quantos foram marcados como positivo;

$$F1 = \frac{2 * \text{Precisão} * \text{Recall}}{\text{Precisão} + \text{Recall}};$$

Média harmônica entre Precisão e Recall.

CLASSIFICAÇÃO DE REDES

Esse capítulo abordará a classificação de redes com base em suas características utilizando algoritmos de *Machine Learning*.

5.1 Estrutura

5.1.1 Dados

Inicialmente geramos 6 tipos de redes artificiais, utilizando os seguintes modelos Erdős–Rény (ER) (ERDÖS; RÉNYI, 1959), Barabási–Albert (BA) (BARABÁSI; ALBERT, 1999), Waxman (WA) (WAXMAN, 1988), Watts–Strogatz (WS) (WATTS; STROGATZ, 1998) ($p=0,01$ e $p=0,1$), Lancichinetti–Fortunato–Radicchi (LFR) (LANCICHINETTI; FORTUNATO; RADICCHI, 2008). Com 50 redes para cada modelo, variamos os números de vértices para 300, 500, 800, 1000, 1500, 2000 e o grau médio 10 e 20, Gerando assim 3600 redes.

Então, para redes reais utilizamos 74 redes, obtidas no site *Network Repository* (ROSSI; AHMED, 2015), divididas em 3 categorias: Biológica, Social, *WEB*, essas nomenclaturas se dão à origem das redes.

5.1.2 Medidas

Uma vez selecionadas as redes foram extraídas algumas de suas medidas, dentre elas Grau (k), *Eigenvector Centrality* (e), *Closeness* (c), *Betweeness* (b), *Page Rank* (p) e foram calculados os três primeiros momentos de todas essas distribuições ($k_2, k_3, e_2, e_3, c_2, c_3, b_2, b_3, p_2, p_3$) e também Transitividade (T), entropia de *Shannon* (S), número de vértices (n) e número de arestas (m).

Obtemos assim um banco de dados com 21 colunas, sendo a última a classificação da rede, para assim podermos alimentar os algoritmos de classificação.

5.2 Redes Artificiais

Separamos as redes artificiais em 2 partes, inicialmente com apenas as quais foram geradas com 1000 vértices e 10 de grau médio, logo após todas as 3600 redes artificiais com todas as variações citadas em [Subseção 5.1.1](#).

Após calculadas as medidas de caracterização de cada rede, aplicamos escala ao banco de dados para que todos os valores obtidos sejam de mesma grandeza e então utilizamos da Análise de Componentes Principais (PCA) para melhor visualização.

5.2.1 Primeira Parte

5.2.1.1 Análise de Componentes Principais

Figura 23 – Análise PCA

Fonte: Elaborada pelo autor.

Observando a [Figura 23a](#) podemos notar que os tipos de redes estão bem separados com a exceção WA e ER. Nas Figuras [23b](#) e [23c](#) podemos ver a contribuição de cada variável aos componentes principais, em que a linha vermelha na [Figura 23b](#) é a contribuição média esperada e na [Figura 23c](#) a seta quanto mais próxima da circunferência maior a contribuição da variável para aquela direção. A [Figura 23d](#) mostra que a Dimensão 1 do PCA explica 66,4% dos dados originais, seguido da Dimensão 2 com 13%.

Para o cálculo do PCA foi utilizada a função PCA do pacote FactoMineR ([LÊ; JOSSE; HUSSON, 2008](#)) para o *software* R.

5.2.1.2 Machine Learning

Utilizando Validação Cruzada com a técnica *k-fold*, com 10 dobras, treinamos os algoritmos *Naive Bayes*, KNN, Redes Neurais e Florestas Aleatórias para assim obter a seguinte tabela de Acurácia ([Tabela 2](#)):

Tabela 2 – Acurácia Redes Artificiais

<i>Fold</i>	<i>Naive Bayes</i>	KNN	Redes Neurais	Floresta Aleatória
1	1,000	1,000	1,000	1,000
2	1,000	1,000	1,000	1,000
3	1,000	1,000	1,000	1,000
4	1,000	1,000	1,000	1,000
5	1,000	1,000	1,000	1,000
6	1,000	0,967	1,000	1,000
7	1,000	1,000	1,000	1,000
8	1,000	1,000	1,000	1,000
9	1,000	1,000	1,000	1,000
10	1,000	1,000	1,000	1,000
Média	1,000	0,997	1,000	1,000

Podemos perceber que os algoritmos estimam perfeitamente as redes, isso se dá pelo fato desse banco de dados conter menos dados destoantes dentro de cada categoria. Através destas simulações o algoritmo Floresta Aleatória nos dá duas medidas discriminando a importância das variáveis, as quais podem ser vistas na [Figura 24](#).

Uma vez que as médias foram obtidas com Validação Cruzada, os valores nos gráficos são resultado da média entre os 10 *folds*. O que se pode observar nas Figuras [24a](#) e [24b](#), é que em relação à importância ao modelo, não há discordância, ambos colocam *PageRank* e o número de arestas como pouca relevância para a classificação.

Em seguida, ainda com o algoritmo Floresta Aleatória e agora juntamente com a técnica RFE (*Recursive Feature Elimination*), foram selecionadas as seguintes variáveis: S e b2 (segundo momento do *Betweenness*), contrariando assim o PCA pois suas contribuições foram abaixo da média esperada.

Figura 24 – Importância das Variáveis

Como duas variáveis foram selecionadas, é possível visualizar a relação das mesmas no $\mathbb{R}^2$. A Figura 25 mostra essa relação e como podemos ver os grupos estão bem separados facilitando assim a classificação.

Figura 25 – Shannon x $\langle Betweenness \rangle^2$

5.2.2 Segunda Parte

Agora, como dito anteriormente, vamos trabalhar com todas as redes artificiais geradas. Iniciamos com o estudo dos gráficos fornecidos pelo PCA (Figura 26a).

5.2.2.1 Análise de Componentes Principais

Figura 26 – Análise PCA

Fonte: Elaborada pelo autor.

É possível ver que assim que as redes de diferentes tamanhos são inseridas no banco de dados, aumenta a dificuldade de separá-las por *clusters* com base em seu tipo (Figura 26a). Na Figura 26b vemos que as contribuições diferem em direção das observadas na Figura 23b, aparecendo menos concentradas em uma direção. O que pode ser observado também é que as duas principais dimensões do PCA explicam menos dos nossos dados originais divididos entre 36,2% e 25,2%, respectivamente, mesmo somadas (61,4%) são menos explicativas que a

primeira dimensão no caso anterior (66,4%) mostrando essa maior dificuldade de representar no $\mathbb{R}^2$.

5.2.2.2 Machine Learning

Damos prosseguimento com a parte de classificação com os algoritmos de *Machine Learning* utilizando todas as variáveis.

Tabela 3 – Acurácia Redes Artificiais

<i>Fold</i>	<i>Naive Bayes</i>	KNN	Redes Neurais	Floresta Aleatória
1	0,814	1,000	1,000	1,000
2	0,819	0,997	1,000	1,000
3	0,836	1,000	1,000	1,000
4	0,794	1,000	1,000	1,000
5	0,814	1,000	1,000	1,000
6	0,797	1,000	1,000	1,000
7	0,822	1,000	1,000	1,000
8	0,811	1,000	1,000	1,000
9	0,839	1,000	1,000	1,000
10	0,814	1,000	0,997	1,000
Média	0,816	1,000	1,000	1,000

A partir do método de validação cruzada obtemos 10 valores de acurácia (Tabela 3), dos quais apenas o algoritmo de *Naive Bayes* não classificou corretamente todos os casos.

A seguir vemos que os gráficos de Gini e Acurácia entram em consenso em relação a importância das variáveis deixando por último *PageRank* e o número de vértices.

Após essa análise foi feita a seleção de variáveis baseado em *Naive Bayes* RFE utilizando da validação cruzada com 10 *folds* repetida 10 vezes, obtivemos que 8 variáveis são importantes para o modelo, e elas são *Eigenvector* (Três primeiros momentos), *Betweenness* (segundo e terceiro momentos), Entropia de Shannon, Transitividade e terceiro momento do Grau.

Então com essas 8 variáveis foi realizado novamente o treinamento dos algoritmos e como pode ser visto na Tabela 4, houve uma melhora em *Naive Bayes* mas também uma pequena piora no KNN e Florestas Aleatórias.

5.3 Redes Reais

Nessa seção trabalharemos com redes reais como dito na Subseção 5.1.1. Inicialmente, trabalharemos com as 74 redes retiradas de bancos de dados na Internet, e em seguida utilizaremos de amostras para melhorar o resultado de classificação.

Figura 27 – Importância das Variáveis

Fonte: Elaborada pelo autor.

Tabela 4 – Acurácia

<i>Fold</i>	<i>Naive Bayes</i>	KNN	Redes Neurais	Floresta Aleatória
1	0,839	0,997	1,000	1,000
2	0,833	1,000	1,000	0,997
3	0,842	0,997	1,000	1,000
4	0,814	0,997	1,000	1,000
5	0,808	1,000	1,000	0,997
6	0,819	1,000	1,000	0,997
7	0,836	1,000	1,000	1,000
8	0,825	0,997	1,000	0,994
9	0,842	1,000	1,000	1,000
10	0,831	1,000	1,000	0,997
Média	0,829	0,999	1,000	0,998

5.3.1 Primeira Parte

5.3.1.1 Análise de Componentes Principais

O que vemos aqui é uma maior dificuldade de visualizar os *clusters* na [Figura 28a](#). 12 variáveis ultrapassam a média de contribuição para Dimensão 1 e 2 na [Figura 28b](#), mas vemos na [Figura 28c](#) que as maiores contribuições estão concentradas em uma única direção positiva para dimensão 1, diminuindo assim a dispersão na [Figura 28a](#).

Figura 28 – Análise PCA

Fonte: Elaborada pelo autor.

Pelas Figuras 28d e 28e pode-se ver quais variáveis contribuem para cada dimensão, inclusive as duas primeiras dimensões somam 69,4% de explicação do banco de dados.

5.3.1.2 Machine Learning

Na Tabela 5, temos as acurácias divididas em 5 *folds* pelo fato de nosso banco de dados ser menor que os anteriores. Pode-se notar que, agora que estamos trabalhando com redes reais, a dificuldade para classificação se torna mais visível.

Tabela 5 – Acurácia Redes Reais

<i>Fold</i>	<i>Naive Bayes</i>	KNN	Redes Neurais	Floresta Aleatória
1	0,533	0,600	0,667	0,800
2	0,800	0,667	0,867	0,933
3	0,643	0,714	0,786	0,714
4	0,733	0,667	0,733	0,867
5	0,533	0,600	0,600	0,667
Média	0,649	0,650	0,730	0,796

Com o intuito de melhorar a acurácia, foram feitas duas seleções de variáveis, usando os algoritmos *Naive Bayes* e Floresta Aleatória, com Validação Cruzada 5 *folds* e 20 repetições.

Tabela 6 – Acurácia Seleção *Naive Bayes*

	<i>Fold</i>	<i>Naive Bayes</i>	KNN	Redes Neurais	Floresta Aleatória
(1)	1	0,533	0,600	0,733	0,800
	2	0,733	0,867	0,733	0,933
	3	0,643	0,571	0,857	0,714
	4	0,667	0,667	0,733	0,867
	5	0,533	0,600	0,667	0,667
	Média	0,622	0,661	0,745	0,796
(2)	1	0,600	0,667	0,667	0,800
	2	0,933	0,667	0,467	0,733
	3	0,714	0,500	0,643	0,714
	4	0,800	0,733	0,800	0,867
	5	0,600	0,467	0,400	0,667
	Média	0,730	0,607	0,595	0,756

(1) 17 variáveis selecionadas pela técnica: *n, p, e3, c, b3, e2, c2, b, b2, e, c3, p2, m, p3, k3, T, ev*.

(2) Top 5 Variáveis da técnica: *n, p, e3, c, b3*.

Tabela 7 – Acurácia Seleção Floresta Aleatória

	<i>Fold</i>	<i>Naive Bayes</i>	KNN	Redes Neurais	Floresta Aleatória
(1)	1	0,533	0,600	0,600	0,800
	2	0,800	0,667	0,667	1,000
	3	0,643	0,714	0,857	0,714
	4	0,733	0,733	0,800	0,867
	5	0,600	0,667	0,600	0,667
	Média	0,662	0,676	0,705	0,810
(2)	1	0,667	0,667	0,867	0,867
	2	0,533	0,733	0,800	0,933
	3	0,643	0,571	0,643	0,786
	4	0,800	0,733	0,800	0,867
	5	0,533	0,667	0,667	0,733
	Média	0,635	0,674	0,755	0,837

(1) 18 variáveis selecionadas pela técnica: *b2, b3, b, e2, p, n, e3, e, c, c2, m, c3, k3, k2, k1, S, T, ev*.

(2) Top 5 variáveis da técnica: *b2, b3, b, e2, p*.

Nas quatro variações obtivemos a melhor acurácia para o algoritmo *Naive Bayes* é na [Tabela 6](#) (2) com 0,730, para KNN na [Tabela 7](#) (1) com 0,676, Redes Neurais na [Tabela 7](#) (2) com 0,755 e por fim Floresta Aleatória com a melhor acurácia [Tabela 7](#) (2) com 0,837. Todas superiores ao modelo utilizando todas as variáveis.

5.3.2 Amostras

Com o intuito de melhorar a classificação das redes reais, utilizamos a técnica de amostragem *SnowBall* para gerar 50 amostras de cada rede, com 10% dos vértices. Depois de geradas foram calculadas as mesmas 20 medidas utilizadas para as classificações anteriormente.

Tabela 8 – Acurácia Redes Reais com Amostras

	<i>Fold</i>	<i>Naive Bayes</i>	KNN	Redes Neurais	Floresta Aleatória
	1	0,600	0,733	0,667	0,867
	2	0,667	0,800	0,867	0,800
	3	0,786	0,714	0,786	0,857
	4	0,867	0,733	0,733	0,933
	5	0,533	0,867	0,800	0,867
	Média	0,690	0,770	0,770	0,865

Como pode ser visto, ao adicionar ao conjunto de treinamento das redes reais as medidas geradas das amostras, temos um ganho significativo nessa primeira simulação. Utilizando as variáveis selecionadas na seção anterior treinamos os modelos novamente adicionando o banco de dados das redes amostrados para auxiliar na classificação.

Tabela 9 – Acurácia Seleção *Naive Bayes*

	<i>Fold</i>	<i>Naive Bayes</i>	KNN	Redes Neurais	Floresta Aleatória
(1)	1	0,600	0,733	0,867	0,933
	2	0,667	0,867	1,000	0,800
	3	0,714	0,571	0,643	0,857
	4	0,867	0,600	0,933	0,933
	5	0,533	0,667	0,667	0,867
	Média	0,676	0,688	0,822	0,878
(2)	1	0,733	0,800	-	0,867
	2	0,800	0,733	-	0,733
	3	0,786	0,929	-	0,857
	4	0,933	0,733	-	0,867
	5	0,667	0,733	-	0,800
	6	0,784	0,786	-	0,825
	Média	0,784	0,786	-	0,825

(1) 17 variáveis selecionadas pela técnica: $n, p, e3, c, b3, e2, c2, b, b2, e, c3, p2, m, p3, k3, T, ev$.

(2) Top 5 variáveis da técnica: $n, p, e3, c, b3$.

Tabela 10 – Acurácia Seleção Floresta Aleatória

	<i>Fold</i>	<i>Naive Bayes</i>	KNN	Redes Neurais	Floresta Aleatória
(1)	1	0,600	0,733	0,667	0,800
	2	0,800	0,800	0,867	0,800
	3	0,786	0,714	0,714	0,857
	4	0,867	0,667	0,800	0,933
	5	0,600	0,867	0,867	0,867
	Média	0,730	0,756	0,783	0,851
(2)	1	0,533	0,733	-	0,933
	2	0,800	0,800	-	0,800
	3	0,786	0,714	-	0,857
	4	0,933	0,867	-	0,933
	5	0,667	0,800	-	0,800
	Média	0,744	0,783	-	0,865

(1) 18 variáveis selecionadas pela técnica: $b2, b3, b, e2, p, n, e3, e, c, c2, m, c3, k3, k2, k1, S, T, ev$.

(2) Top 5 variáveis da técnica: $b2, b3, b, e2, p$.

Com as variáveis escolhidas utilizando o algoritmo *Naive Bayes* (Tabela 9), vemos uma melhora nas acurácias de Redes Neurais e Floresta Aleatória com 17 variáveis. Já para as top 5 variáveis, os algoritmos *Naive Bayes* e KNN que melhoraram e com essa configuração não foi possível para que o Redes Neurais convergisse.

Por sua Vez, a Tabela 10 mostra que não houve melhora na primeira configuração (18 variáveis). Mas para configuração com as top 5 variáveis, vemos melhora significativa para o algoritmo *Naive Bayes* e KNN. Para Floresta Aleatória a mesma acurácia de antes da seleção de

variáveis.

Por fim, calculamos Acurácia, Precisão, Recall e F1 para as configurações de 17 e 18 variáveis escolhidas, as acurácias diferem das anteriores, pois se tratam de acurácia global e as medidas abaixo (Tabelas 11 e 12) acurácia média entre as classes.

Tabela 11 – Seleção *Naive Bayes*

	Naive Bayes	KNN	Redes Neurais	Floresta Aleatória
Acurácia	0.748	0.774	0.830	0.864
Precisão	0.597	0.628	0.769	0.751
Recall	0.548	0.628	0.772	0.810
F1	0.694	0.726	0.894	0.879

17 variáveis selecionadas pela técnica: $n, p, e3, c, b3, e2, c2, b, b2, e, c3, p2, m, p3, k3, T, ev$.

Tabela 12 – Seleção Floresta Aleatória

	Naive Bayes	KNN	Redes Neurais	Floresta Aleatória
Acurácia	0.775	0.784	0.803	0.864
Precisão	0.616	0.627	0.665	0.776
Recall	0.582	0.652	0.676	0.810
F1	0.723	0.751	0.776	0.904

18 variáveis selecionadas pela técnica: $b2, b3, b, e2, p, n, e3, e, c, c2, m, c3, k3, k2, k1, S, T, ev$.

Pode-se ver que, como anteriormente o algoritmo de Floresta Aleatória se destaca em todas as medidas. E também não apresenta muita diferença nas duas configurações.

CONSIDERAÇÕES FINAIS

Conforme discutimos anteriormente, o estudo de métodos para amostrar redes é de importância fundamental no estudo de redes complexas, pois a estrutura de muitas redes não está disponível atualmente, como ocorre com as redes de interação de proteínas, ou as redes são grandes demais para serem processadas e armazenadas pelos computadores atuais. Nossas análises feitas nos subgrafos amostrados mostram que, conforme aumentamos o tamanho da amostra, as estimativas tendem ao valor real da medida estudada. No entanto, dependendo do método e da rede, as estimativas ficam muito diferentes dos valores observados nos dados originais. Algumas dificuldades foram encontradas no caminho, algumas relacionadas a processamento de redes muito grandes e também o elevado tempo de amostragem da técnica de caminhada aleatória. Assim sendo, um possível futuro trabalho é comparar os métodos de amostragem de redes em diferentes tipos de redes, sejam elas reais ou sintéticas, com objetivo de determinar o método mais adequado para um tipo de rede. Por exemplo, é esperado que a amostragem aleatória simples seja adequada para grafos aleatórios, nos quais os vértices apresentam número similar de conexões, mas não seja válida para redes sem escala, que apresentam *HUBS*.

Finalmente, analisamos as redes artificiais e reais com o intuito de classificá-las através de suas medidas. Para as artificiais, a classificação foi de baixa dificuldade, uma vez que, por se tratarem de algoritmos, não apresentam fortes discrepâncias em suas medidas. Já para as redes reais notamos uma dificuldade para a classificação. Aqui a proposta de amostrar as redes surtiu o efeito desejado ultrapassando as seleções de variáveis com apenas as redes reais, e atingindo seu máximo de acurácia com as 17 variáveis selecionadas pelo algoritmo RFE com *Naive Bayes* e utilizando Floresta Aleatória.

REFERÊNCIAS

- AMARAL, L. A.; OTTINO, J. M. Complex networks. **The European Physical Journal B-Condensed Matter and Complex Systems**, Springer, v. 38, n. 2, p. 147–162, 2004. Citado na página [28](#).
- BARABÁSI, A.-L.; ALBERT, R. Emergence of scaling in random networks. **science**, American Association for the Advancement of Science, v. 286, n. 5439, p. 509–512, 1999. Citado nas páginas [30](#) e [57](#).
- BASTIAN, M.; HEYMANN, S.; JACOMY, M. *et al.* Gephi: an open source software for exploring and manipulating networks. **Icwsn**, v. 8, p. 361–362, 2009. Citado na página [41](#).
- BIGGS, N.; LLOYD, E. K.; WILSON, R. J. **Graph Theory, 1736-1936**. [S.l.]: Oxford University Press, 1976. Citado na página [21](#).
- BOAS, P. V.; RODRIGUES, F.; TRAVIESO, G.; COSTA, L. da F. Sensitivity of complex networks measurements. **Journal of Statistical Mechanics: Theory and Experiment**, IOP Publishing, v. 2010, n. 03, p. P03009, 2010. Citado na página [22](#).
- BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, n. 1, p. 5–32, 2001. Citado na página [55](#).
- BRIN, S.; PAGE, L. The anatomy of a large-scale hypertextual web search engine. **Computer networks and ISDN systems**, Elsevier, v. 30, n. 1-7, p. 107–117, 1998. Citado na página [29](#).
- BUSSAB, W. D. O.; BOLFARINE, H. **Elementos de amostragem**. [S.l.: s.n.], 2005. Citado nas páginas [31](#) e [32](#).
- CONOVER, W. J. A kolmogorov goodness-of-fit test for discontinuous distributions. **Journal of the American Statistical Association**, Taylor & Francis, v. 67, n. 339, p. 591–596, 1972. Citado na página [43](#).
- COSTA, L. D. F.; JR, O. N. O.; TRAVIESO, G.; RODRIGUES, F. A.; BOAS, P. R. V.; ANTIQUEIRA, L.; VIANA, M. P.; ROCHA, L. E. C. Analyzing and modeling real-world phenomena with complex networks: a survey of applications. **Advances in Physics**, Taylor & Francis, v. 60, n. 3, p. 329–412, 2011. Citado na página [21](#).
- COSTA, L. D. F.; RODRIGUES, F. A.; TRAVIESO, G.; BOAS, P. R. V. Characterization of complex networks: A survey of measurements. **Advances in physics**, Taylor & Francis, v. 56, n. 1, p. 167–242, 2007. Citado nas páginas [26](#), [29](#) e [30](#).
- CSARDI, G.; NEPUSZ, T. The igraph software package for complex network research. *inter-journal complex systems* 1695. **Available at igraph.org/**. Accessed November, v. 30, p. 2015, 2006. Citado na página [30](#).
- ERDÖS, P.; RÉNYI, A. On random graphs i. **Publ. Math. Debrecen**, v. 6, p. 290–297, 1959. Citado nas páginas [29](#) e [57](#).

- FREEMAN, L. C. Centrality in social networks conceptual clarification. **Social networks**, North-Holland, v. 1, n. 3, p. 215–239, 1978. Citado nas páginas 28 e 29.
- FRITSCH, S.; GUENTHER, F. **neuralnet: Training of Neural Networks**. [S.l.], 2016. R package version 1.33. Disponível em: <<https://CRAN.R-project.org/package=neuralnet>>. Citado na página 52.
- GRIGORIEV, A. On the number of protein–protein interactions in the yeast proteome. **Nucleic acids research**, Oxford University Press, v. 31, n. 14, p. 4157–4161, 2003. Citado nas páginas 21 e 22.
- GRUPO R BRASIL - PROGRAMADORES. **R Brasil - Programadores**. 2017. Disponível em: <<https://www.facebook.com/groups/1410023525939155/>>. Citado na página 40.
- GUIMERA, R.; DANON, L.; DIAZ-GUILERA, A.; GIRALT, F.; ARENAS, A. Self-similar community structure in a network of human interactions. **Physical review E**, APS, v. 68, n. 6, p. 065103, 2003. Citado nas páginas 21, 27 e 39.
- KOLACZYK, E. D. **Statistical Analysis of Network Data: Methods and Models**. [S.l.]: Springer, 2009. Citado nas páginas 21, 26, 28, 31, 34 e 35.
- KUNEGIS, J. **konect network dataset – KONECT**. 2017. <<http://konect.uni-koblenz.de>>. Citado na página 39.
- LANCICHINETTI, A.; FORTUNATO, S.; RADICCHI, F. Benchmark graphs for testing community detection algorithms. **Physical review E**, APS, v. 78, n. 4, p. 046110, 2008. Citado na página 57.
- LÊ, S.; JOSSE, J.; HUSSON, F. FactoMineR: A package for multivariate analysis. **Journal of Statistical Software**, v. 25, n. 1, p. 1–18, 2008. Citado na página 59.
- LESKOVEC, J.; KREVL, A. **SNAP Datasets: Stanford Large Network Dataset Collection**. 2014. <<http://snap.stanford.edu/data>>. Citado na página 39.
- LESKOVEC, J.; MCAULEY, J. J. Learning to discover social circles in ego networks. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2012. p. 539–547. Citado na página 39.
- LIAW, A.; WIENER, M. Classification and regression by randomforest. **R News**, v. 2, n. 3, p. 18–22, 2002. Disponível em: <<https://CRAN.R-project.org/doc/Rnews/>>. Citado na página 54.
- MEYER, D.; DIMITRIADOU, E.; HORNIK, K.; WEINGESSEL, A.; LEISCH, F. **e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien**. [S.l.], 2018. R package version 1.7-0. Disponível em: <<https://CRAN.R-project.org/package=e1071>>. Citado na página 52.
- R Core Team. **R: A Language and Environment for Statistical Computing**. Vienna, Austria, 2018. Disponível em: <<https://www.R-project.org/>>. Citado na página 30.
- RIEDER, B. Studying facebook via data extraction: the netvizz application. In: **ACM. Proceedings of the 5th annual ACM web science conference**. [S.l.], 2013. p. 346–355. Citado na página 40.

RIEDMILLER, M. Advanced supervised learning in multi-layer perceptrons-from backpropagation to adaptive learning algorithms. **Computer standards and interfaces**, Amsterdam, The Netherlands: North-Holland, c1986-, v. 16, n. 3, p. 265–278, 1994. Citado na página 52.

_____. **Rprop-Description and implementation details: technical report**. [S.l.]: Inst. f. Logik, Komplexität u. Deduktionssysteme, 1994. Citado na página 52.

RIPLEY, B. D. Pattern recognition via neural networks. **a volume of Oxford Graduate Lectures on Neural Networks, title to be decided**. Oxford University Press.[See <http://www.stats.ox.ac.uk/ripley/papers.html>.], 1996. Citado na página 53.

ROSSI, R. A.; AHMED, N. K. The network data repository with interactive graph analytics and visualization. In: **Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence**. [s.n.], 2015. Disponível em: <<http://networkrepository.com>>. Citado nas páginas 57, 75, 76 e 77.

SHANNON, C. E. A mathematical theory of communication. **Bell system technical journal**, Wiley Online Library, v. 27, n. 3, p. 379–423, 1948. Citado na página 29.

STUTZBACH, D.; REJAIE, R.; DUFFIELD, N.; SEN, S.; WILLINGER, W. On unbiased sampling for unstructured peer-to-peer networks. **IEEE/ACM Transactions on Networking (TON)**, IEEE Press, v. 17, n. 2, p. 377–390, 2009. Citado na página 37.

ŠUBELJ, L.; BAJEC, M. Robust network community detection using balanced propagation. **The European Physical Journal B-Condensed Matter and Complex Systems**, Springer, v. 81, n. 3, p. 353–362, 2011. Citado nas páginas 21 e 40.

VENABLES, W. N.; RIPLEY, B. D. **Modern Applied Statistics with S**. Fourth. New York: Springer, 2002. ISBN 0-387-95457-0. Disponível em: <<http://www.stats.ox.ac.uk/pub/MASS4>>. Citado na página 53.

WATTS, D. J.; STROGATZ, S. H. Collective dynamics of 'small-world' networks. **nature**, Nature Publishing Group, v. 393, n. 6684, p. 440, 1998. Citado nas páginas 28, 30 e 57.

WAXMAN, B. M. Routing of multipoint connections. **IEEE journal on selected areas in communications**, IEEE, v. 6, n. 9, p. 1617–1622, 1988. Citado na página 57.

ZHANG, H. The optimality of naive bayes. **AA**, v. 1, n. 2, p. 3, 2004. Citado na página 52.

REDES UTILIZADAS

A.1 Redes Reais

	Redes	$\langle k \rangle$	Vértices	Arestas	Fonte
1	bio-CE-CX	32.64	15063	245862	(ROSSI; AHMED, 2015)
2	bio-CE-GN	48.47	2215	53680	(ROSSI; AHMED, 2015)
3	bio-CE-GT	7.25	878	3181	(ROSSI; AHMED, 2015)
4	bio-CE-HT	2.45	2194	2688	(ROSSI; AHMED, 2015)
5	bio-CE-LC	2.62	993	1300	(ROSSI; AHMED, 2015)
6	bio-CE-PG	55.92	1692	47309	(ROSSI; AHMED, 2015)
7	bio-celegans-dir	8.94	453	2025	(ROSSI; AHMED, 2015)
8	bio-celegans	8.94	453	2025	(ROSSI; AHMED, 2015)
9	bio-diseasome	4.60	516	1188	(ROSSI; AHMED, 2015)
10	bio-DM-CX	38.05	4032	76713	(ROSSI; AHMED, 2015)
11	bio-DM-HT	3.22	2831	4562	(ROSSI; AHMED, 2015)
12	bio-DM-LC	4.13	483	997	(ROSSI; AHMED, 2015)
13	bio-dmela	6.92	7393	25569	(ROSSI; AHMED, 2015)
14	bio-DR-CX	51.68	3287	84939	(ROSSI; AHMED, 2015)
15	bio-grid-fission-yeast	12.78	1972	12601	(ROSSI; AHMED, 2015)
16	bio-grid-fruitfly	6.93	7163	24835	(ROSSI; AHMED, 2015)
17	bio-grid-human	6.76	9186	31038	(ROSSI; AHMED, 2015)
18	bio-grid-mouse	2.78	791	1098	(ROSSI; AHMED, 2015)
19	bio-grid-plant	4.29	1272	2726	(ROSSI; AHMED, 2015)
20	bio-grid-worm	3.85	3343	6437	(ROSSI; AHMED, 2015)
21	bio-HS-CX	49.36	4409	108816	(ROSSI; AHMED, 2015)
22	bio-HS-HT	10.92	2499	13650	(ROSSI; AHMED, 2015)
23	bio-HS-LC	18.73	4215	39477	(ROSSI; AHMED, 2015)
24	bio-SC-CC	31.38	2223	34879	(ROSSI; AHMED, 2015)
25	bio-SC-GT	39.79	1708	33982	(ROSSI; AHMED, 2015)
26	bio-SC-HT	60.69	2077	63023	(ROSSI; AHMED, 2015)
27	bio-SC-LC	20.46	1999	20448	(ROSSI; AHMED, 2015)

	Redes	$\langle k \rangle$	Vértices	Arestas	Fonte
28	bio-SC-TS	66.00	67	2211	(ROSSI; AHMED, 2015)
29	bio-WormNet-v3-benchmark	68.89	2274	78328	(ROSSI; AHMED, 2015)
30	bio-yeast-protein-inter	2.67	1458	1948	(ROSSI; AHMED, 2015)
31	bio-yeast	2.67	1458	1948	(ROSSI; AHMED, 2015)
32	soc-academia	10.22	200167	1022440	(ROSSI; AHMED, 2015)
33	soc-advogado	15.58	5054	39374	(ROSSI; AHMED, 2015)
34	soc-anybeat	7.77	12645	49132	(ROSSI; AHMED, 2015)
35	soc-BlogCatalog	47.15	88784	2093195	(ROSSI; AHMED, 2015)
36	soc-brightkite	7.51	56739	212945	(ROSSI; AHMED, 2015)
37	soc-buzznet	54.63	101163	2763066	(ROSSI; AHMED, 2015)
38	soc-douban	4.22	154908	327162	(ROSSI; AHMED, 2015)
39	soc-epinions	7.53	26588	100120	(ROSSI; AHMED, 2015)
40	soc-flickr	12.41	513969	3190452	(ROSSI; AHMED, 2015)
41	soc-google-plus	11.23	201949	1133956	(ROSSI; AHMED, 2015)
42	soc-gowalla	9.67	196591	950327	(ROSSI; AHMED, 2015)
43	soc-gplus	3.32	23613	39182	(ROSSI; AHMED, 2015)
44	soc-hamsterster	16.10	2000	16097	(ROSSI; AHMED, 2015)
45	soc-LiveMocha	42.13	104103	2193083	(ROSSI; AHMED, 2015)
46	soc-loc-brightkite	7.51	56739	212945	(ROSSI; AHMED, 2015)
47	soc-sign-Slashdot081106	12.12	77350	468554	(ROSSI; AHMED, 2015)
48	soc-sign-Slashdot090216	12.16	81867	497672	(ROSSI; AHMED, 2015)
49	soc-sign-Slashdot090221	12.19	82140	500481	(ROSSI; AHMED, 2015)
50	soc-slashdot-trust-all	11.83	79120	467869	(ROSSI; AHMED, 2015)
51	soc-slashdot	10.24	70068	358647	(ROSSI; AHMED, 2015)
52	soc-Slashdot0811	12.13	77360	469180	(ROSSI; AHMED, 2015)
53	soc-Slashdot0902	12.27	82168	504230	(ROSSI; AHMED, 2015)
54	soc-twitter-follows-mun	3.58	465017	833541	(ROSSI; AHMED, 2015)
55	soc-twitter-follows	3.53	404719	713319	(ROSSI; AHMED, 2015)
56	soc-wiki-elec	28.51	7066	100727	(ROSSI; AHMED, 2015)
57	soc-wiki-Vote	6.56	889	2914	(ROSSI; AHMED, 2015)
58	web-arabic-2005	21.36	163598	1747269	(ROSSI; AHMED, 2015)
59	web-baidu-baike-related	12.43	372840	2318025	(ROSSI; AHMED, 2015)
60	web-BerkStan	3.17	12305	19500	(ROSSI; AHMED, 2015)
61	web-edu	4.27	3031	6474	(ROSSI; AHMED, 2015)
62	web-EPA	4.18	4253	8897	(ROSSI; AHMED, 2015)
63	web-frwikinews-user-edits	5.48	25042	68675	(ROSSI; AHMED, 2015)
64	web-google-dir	10.03	855802	4291352	(ROSSI; AHMED, 2015)
65	web-google	4.27	1299	2773	(ROSSI; AHMED, 2015)
66	web-indochina-2004	8.38	11358	47606	(ROSSI; AHMED, 2015)
67	web-italycnr-2000	16.83	325557	2738969	(ROSSI; AHMED, 2015)
68	web-NotreDame	6.69	325729	1090108	(ROSSI; AHMED, 2015)
69	web-polblogs	7.09	643	2280	(ROSSI; AHMED, 2015)

	Redes	$\langle k \rangle$	Vértices	Arestas	Fonte
70	web-sk-2005	5.51	121422	334419	(ROSSI; AHMED, 2015)
71	web-spam	15.68	4767	37375	(ROSSI; AHMED, 2015)
72	web-Stanford	15.21	255265	1941926	(ROSSI; AHMED, 2015)
73	web-webbase-2001	3.19	16062	25593	(ROSSI; AHMED, 2015)
74	web-wikipedia2009	4.84	1864433	4507315	(ROSSI; AHMED, 2015)