AMOSTRAGEM PARA GRANDES VOLUMES DE DADOS: UMA APLICAÇÃO EM REDES COMPLEXAS

Roberta Carneiro de Souza

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Civil, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Civil.

Orientador: Nelson Francisco Favilla Ebecken

Rio de Janeiro
Junho de 2018

AMOSTRAGEM PARA GRANDES VOLUMES DE DADOS: UMA APLICAÇÃO EM REDES COMPLEXAS

Roberta Carneiro de Souza

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA CIVIL.

Examinada por:

Prof. Nelson Francisco Favilla Ebecken, D.Sc.

Prof. Pedro Luis do Nascimento Silva, D.Sc.

Prof. Beatriz de Souza Leite Pires de Lima, D.Sc.

Prof. Solange Guimarães, D.Sc.

Prof. Elton Fernandes, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
JUNHO DE 2018

Souza, Roberta Carneiro de

Amostragem para grandes volumes de dados: uma aplicação em redes complexas/Roberta Carneiro de Souza. – Rio de Janeiro: UFRJ/COPPE, 2018.

XIV, 63 p.: il.; 29, 7cm.

Orientador: Nelson Francisco Favilla Ebecken

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Civil, 2018.

Referências Bibliográficas: p. 52 – 56.

1. Amostragem. 2. Redes Complexas. 3. Grafos. 4. Mineração de Dados. 5. Centralidade de Intermediação. 6. Agrupamento. I. Ebecken, Nelson Francisco Favilla. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Civil. III. Título.

Aos meus pais,
Clovenildo e Marinete.
Eles são a minha fortaleza.

Agradecimentos

Agradeço aos meu pais, Clovenildo e Marinete, pelo apoio de sempre.

Também agradeço ao meu companheiro André pela parceria incondicional. E aos nossos filhos pela paciência com minha ausência temporária e pela compreensão.

Obrigada Rodrigo, mesmo longe me apoiando e perdoando a minha impossibilidade de te visitar.

Obrigada Mateus, pelos cafezinhos que você levava para mim enquanto eu estudava e pelo carinho que me fazia quando eu estava cansada.

Obrigada Bruno, sempre me transmitindo tranquilidade.

E agradeço a todos os amigos do IBGE, com destaque para Vivi, Sofia, Luiz, Sâmela e Thaís por terem me apoiado desde o início da minha jornada no IBGE.

Um obrigado especial às minhas amigas Ana Mary e Monica Benevides, que mesmo quando eu estava no fundo do poço, acreditaram em mim e permaneceram ao meu lado.

Ao meu orientador Nelson e a todos os chefes que tive nesses longos cinco anos (Jorginho, Andréa, Sônia, Giuseppe, André e Tiago) pela paciência que tiveram comigo e pelo apoio.

Agradeço ao meu mentor Pedro Luis, pelo coleguismo, pela confiança que depositou em mim, pela mão que me estendeu no momento onde mais precisei.

E, finalmente, um agradecimento especial à Andie, que não me deixou cair, leu, revisou, cobrou, tudo com muito carinho.

E a Deus pela força que me deu e me trouxe até aqui.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

AMOSTRAGEM PARA GRANDES VOLUMES DE DADOS: UMA APLICAÇÃO EM REDES COMPLEXAS

Roberta Carneiro de Souza

Junho/2018

Orientador: Nelson Francisco Favilla Ebecken

Programa: Engenharia Civil

Este trabalho tem como objetivo principal implementar e avaliar opções de planos amostrais de algoritmos para cálculo de centralidade de intermediação - uma medida utilizada para identificar vértices importantes e influentes - em redes complexas, visando melhorar a qualidade das estimativas. A avaliação estatística da qualidade dessas estimativas será feita através de indicadores propostos, já utilizados em amostragem mas não em mineração de dados em redes complexas. As técnicas utilizadas de forma combinada para atingir os objetivos e propor um novo algoritmo foram: amostragem, agrupamento (ou detecção de comunidades) e computação paralela. O recurso de amostragem vem sendo utilizado amplamente como ferramenta de redução de dimensionalidade em problemas de mineração de dados para agilizar processos e diminuir custos com armazenagem de dados. As técnicas de agrupamento para detecção de comunidades possuem alta correlação com a medida que se deseja estimar, a centralidade de intermediação. Um dos fatores considerados na escolha dos métodos empregados na implementação dos algoritmos foi a possibilidade de se utilizar computação paralela ou distribuída. Após revisão da literatura e avaliação dos resultados dos experimentos realizados, conclui-se que o algoritmo proposto pelo presente estudo contribui para o estado da arte da utilização de amostragem para estimar centralidade de intermediação em grandes redes complexas, um desafio no cenário atual de *big data*, ao agregar várias técnicas que otimizam a extração de conhecimento de dados. O algoritmo proposto, além de melhorar a qualidade das estimativas, apresentou redução no tempo de processamento mantendo a escalabilidade.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

SAMPLING FOR LARGE DATA VOLUMES: AN APPLICATION ON COMPLEX NETWORKS

Roberta Carneiro de Souza

June/2018

Advisor: Nelson Francisco Favilla Ebecken

Department: Civil Engineering

The main objective of this work is to implement and to evaluate options of sampling plans of algorithms for calculation of betweenness centrality, a measure used to identify important and influential vertices in complex networks aiming to improve the quality of the estimates. For statistical evaluation of variability of the estimates, indicators used in sampling, but not yet in data mining in complex networks, will be proposed. The techniques used in combination to reach the objectives and propose a new algorithm were: sampling, clustering (or community detection) and parallel computing. The sampling feature has been widely used as a tool to reduce dimensionality in data mining problems to streamline processes and reduce costs with data storage. The techniques of grouping for the detection of communities have a high correlation with the measure to be estimated, the betweenness centrality. One of the factors used in choosing the methods used in the implementation of the algorithms was the possibility of using parallel or distributed computing. After the review of the literature and evaluation of the results of the experiments carried out, it is concluded that the proposed algorithm contributes to the state of the art of the use of sampling to estimate betweenness centrality in large complex networks, a challenge in the current scenario of big data, by adding several techniques that optimize the extraction of data knowledge. The proposed algorithm, in addition to improving the quality of the estimates, presented a reduction in the processing time while keeping the scalability.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Símbolos

A	Amostra de r caminhos mínimos, subconjunto de S_G, p. 35
$C(G)$	Agrupamento médio ou coeficiente de agrupamento do grafo G, p. 13
$CI(v)$	Centralidade de intermediação do vértice v, p. 14
$CIp(v)$	Centralidade de intermediação padronizada do vértice v, p. 14
$CM(v)$	Comunidade do vértice v detectada pelo método de Louvain, p. 34
E	Conjunto de arestas, p. 12
G	Grafo, p. 12
H	Quantidade de estratos, p. 9
N	Tamanho da população, p. 8
N_h	Tamanho dos estratos populacionais, p. 9
$Ps(w)$	Conjunto de todos os vértices antecessores diretos de vértice w nos caminhos mais curtos de s para w, p. 27
Q	Modularidade do grafo, p. 18
$SP_{s,t}$	Todos os caminhos mais curtos entre (s,t), p. 29
S_G	Conjunto de todos os caminhos mais curtos do grafo G, p. 35
T_v	Conjunto de todos os caminhos mais curtos onde o vértice v é um vértice dentro do caminho, ou seja, está no meio e não é um dos extremos, p. 35
$V(Y)$	Variância de Y, p. 9
$VC(T_v)$	Dimensão Vapnik-Chervonenkis de T_v, p. 35

$VD(G)$	Número de vértices no caminho do diâmetro de G, p. 13
$V_{AASc}(\overline{y})$	Variância da média amostral sob o plano AASc, p. 9
$Vert$	Conjunto de vértices, p. 12
Y	Variável de interesse, p. 9
$\overline{Y}$	Média populacional, p. 9
$\overline{g}$	Grau médio de um grafo, p. 13
$\overline{y}$	Média amostral, p. 9
$\sigma_{s,t}$	Número de caminhos mais curtos entre os vértices s e t, p. 14
$\sigma_{s,t}(v)$	Número de caminhos mais curtos entre s e t que passam pelo vértice v, p. 14
$\tilde{CIp}(v)$	Estimador não viesado da centralidade de intermediação padronizada do vértice v, p. 37
d	Densidade de um grafo, p. 12
$diam(G)$	Diâmetro do grafo G, maior distância entre dois vértices quaisquer do grafo, p. 13
$dist(s,t)$	Tamanho do caminho mais curto entre os vértices s e t medido em número de arestas entre s e t, p. 13
$g(v)$	Grau do vértice v, p. 12
m	Número de arestas, p. 12
n	Número de vértices, p. 12
$p_{s,t}$	Um caminho mínimo entre s e t, p. 35
r	Tamanho da amostra, p. 8
CV	Coeficiente de variação (desvio padrão sobre a média), p. 8
EPA	Efeito do Plano Amostral de Kish, p. 8
EQM	Erro Quadrático Médio, p. 8

Lista de Abreviaturas

AASc	Amostra Aleatória Simples com reposição, p. 8
AASs	Amostra Aleatória Simples sem reposição, p. 8
ACM	*Association for Computing Machinery*, p. 19
AEp	Amostra Estratificada com alocação proporcional ao tamanho do estrato na população e AASc dentro dos estratos, p. 8
CIAVLCM	Algoritmo baseado em amostragem e agrupamento proposto nesta tese, p. 34
CIA	Algoritmo para estimar centralidade de intermediação baseado em amostragem, p. 29
IBGE	Instituto Brasileiro de Geografia e Estatística, p. 4
INE	Instituto Nacional de Estatística, p. 2
PPT	Amostragem com probabilidade proporcional ao tamanho, p. 10
SIGKDD	*Special Interest Group on Knowledge Discovery and Data Mining*, p. 19
SSSP	*Single Source Shortest Paths* Todos os caminhos mais curtos partindo de um vértice origem, p. 27
UPA	Unidade Primária de Amostragem, p. 10
USA	Unidade Secundária de Amostragem, p. 10

Capítulo 1

Introdução

Atualmente os bancos de dados estão cada vez maiores e a obtenção de informação e conhecimento utilizando os dados nesses bancos torna-se uma tarefa complexa e, às vezes, inviável. Com técnicas de amostragem, é possível obter informações sobre o todo baseando-se no resultado de uma amostra, desde que os dados sejam bem extraídos, ou seja, que se possa generalizar os resultados para o todo (BOLFARINE e BUSSAB, 2004). Estudos recentes mostram a possibilidade de obtenção de estimativas eficientes para várias tarefas de mineração de dados, utilizando amostragem para reduzir custo e tempo de processamento, assim como espaço de armazenagem de dados.

A principal vantagem da utilização de amostragem como estratégia de redução de dimensionalidade na fase de pré-processamento dos dados é a redução de tempo de execução dos algoritmos de mineração de dados (TAN *et al.*, 2009)e, em particular, neste trabalho, no cálculo de centralidade de intermediação dos vértices de redes complexas, que são grafos com características estruturais não triviais. Na era do *big data* e seus V's (velocidade, variedade, volume, valor e veracidade), os grafos representam uma forma de representação de dados complexos (variedade) útil para extrair informação dos mesmos (valor). A amostragem age na redução de volume e consequentemente na redução de tempo de execução (velocidade) e de espaço de armazenamento. E o plano amostral deve ser adequado para se extrair informação para o todo de forma fidedigna (veracidade), ou seja, tem que fornecer estimativas com qualidade.

Um exemplo importante e atual de utilidade da teoria de amostragem está no caso de fluxo de dados (*data stream*), pois são informações em fluxo constante de atualização, e a possibilidade de trabalhar com amostras agiliza a extração de informação desses dados como, por exemplo, milhares de imagens sendo recebidas de satélites para avaliação de uso e cobertura da terra.

O jornal *"The Financial Times"* possui uma coluna chamada *"The big read"* que publica notícias sobre *big data* e novidades no campo da tecnologia. Numa

reportagem com o título "O *big data* pode revolucionar a formulação de políticas pelos governos? Mineração de informações digitais para instantâneos econômicos precisos e atualizados pode ajudar as autoridades a tomar decisões mais rápidas e melhores"[1], de janeiro de 2018, são levantadas questões onde o uso de **amostragem** pode ajudar a evoluir o estado da arte das pesquisas que usam *big data*. Nesta reportagem são apresentados casos de sucesso como, por exemplo, o projeto do MIT chamado "Projeto bilhões de preços", que já consegue estimar inflação diária para vários países utilizando preços retirados de comerciantes digitais, e a ambição de se construir um mapa da economia em tempo real.

Alguns exemplos de amostragem utilizada em *big data* para estatísticas oficiais são os trabalhos realizados pelo grupo de trabalho *Global Working Group on big data for official statistics* criado pela Comissão de Estatística das Nações Unidas (*United Nations Statistical Commission*)[2]. Destacam-se aqui alguns desses exemplos:

- Levantamento de culturas agrícolas: usando sensoriamento remoto por satélite para ajudar a estimar estatísticas agrícolas do Instituto Nacional de Estatística (INE) da China. O trabalho tem como objetivo construir uma estrutura de **amostragem espacial** usando os dados de pesquisas de uso da terra e do censo agropecuário, atualizadas com amostras de imagens por satélite e sensoriamento remoto aéreo. Com as amostras selecionadas pelo método de amostragem espacial, estima-se a área de plantio e a produção a cada temporada.

- Explorando o uso de mensagens da mídia social para indicadores econômicos do INE da Holanda. Pesquisa que explora a usabilidade de mensagens públicas de mídia social para estatísticas oficiais. Verifica se os resultados do índice de confiança do consumidor baseado em pesquisa existente poderiam ser replicados usando apenas essa fonte, reduzindo o tempo de produção e expande o conhecimento metodológico em da **teoria da amostragem**. Essa fonte de dados é amplamente considerada como tendo um enorme potencial para esclarecer uma série de fenômenos sociais. No entanto, o sucesso não significa automaticamente a aplicação às estatísticas oficiais, o que requer uma avaliação que vai além da questão da viabilidade técnica.

- Utilização de dados *online* no Índice Harmonizado de Preços ao Consumidor do INE da Noruega. Este projeto busca identificar áreas onde o comércio na internet é significativo e traz uma discussão sobre como **planejar o processo de amostragem**, analisar dados coletados *online* e avaliar os resultados obtidos.

[1] https://www.ft.com/content/9f0a8838-fa25-11e7-9b32-d7d59aace167
[2] https://unstats.un.org/bigdata/

- Potencial de dados de medição inteligente para detecção de domicílios desocupados do INE do Reino Unido. O INE do Reino Unido adquiriu dados de medidores inteligentes de uso de energia elétrica. Esses dados têm vários usos possíveis dentro das estatísticas oficiais e o foco do trabalho está na ocupação. Saber quais áreas do país têm altos níveis de domicílios desocupados seria benéfico para a logística do trabalho de campo, além de melhorar o **planejamento amostral** das pesquisas domiciliares. O objetivo dessa pesquisa é comparar estimativas derivadas de dados de medidores inteligentes com fontes oficiais alternativas de dados sobre propriedades vagas e verificar se os dados de medidores inteligentes têm alguma vantagem em termos de custo, pontualidade, geografia ou precisão.

Neste estudo, são apresentados alguns algoritmos baseados em amostragem para estimar centralidade de intermediação em redes complexas e um algoritmo retirado da revisão da literatura (estado da arte) é implementado e utilizado como base para novos algoritmos com outras opções de plano amostral. Técnicas de amostragem e medidas de agrupamento formam a base do algoritmo proposto, que busca melhorar a acurácia e a precisão dessas estimativas.

A centralidade de intermediação é uma medida de importância relativa de um vértice num grafo, e seu cálculo é uma das principais tarefas de mineração de dados em grafos para identificação de vértices importantes e influentes. Corresponde à fração de caminhos mais curtos que passam pelo vértice (FREEMAN, 1977), logo, é necessário encontrar todos os caminhos mais curtos entre todos os pares de vértices no grafo. Dentre as aplicações da centralidade de intermediação, pode-se enumerar as seguintes: análise das interações em grafos sociais e em grafos de proteína; para avaliar o tráfego de informações em grafos de comunicação; para identificar interseções importantes em estradas; na identificação de clientes influentes; na identificação de *hubs* em grafos de energia; para avaliar mercado financeiro (DEL-VECCHIO *et al.*, 2009); detecção de comunidades; na identificação de principais transmissores de uma doença e avaliar confiabilidade numa rede (SILVA, 2010). Ao ordenar os vértices conforme esta medida de centralidade, pode-se encontrar o vértice mais importante. Segundo Riondato e Kornaropoulos (2016):

> "Índices de centralidade são métricas fundamentais para análise de redes. Eles expressam a importância relativa de um vértice na rede. Alguns refletem propriedades locais de um grafo, como a centralidade de grau, por exemplo. Enquanto outros fornecem informação sobre a estrutura global da rede, pois são baseados na contagem de caminhos mais curtos, como a centralidade de intermediação. Estamos interessados em estimar

a centralidade de intermediação, que é, para cada vértice do grafo, a soma das frações de caminhos mais curtos que passam por esse vértice."

No estudo mais recente sobre redes e fluxos do território brasileiro divulgado pelo Instituto Brasileiro de Geografia e Estatística (IBGE) , intitulado Logística de Energia 2015 (IBGE, 2016), são utilizadas as centralidades de proximidade e de intermediação para identificar os municípios do Brasil mais importantes com relação a logística de energia. A identificação de vértices que, se desconectados, desconectam parte do grafo é muito importante pois gera o não fornecimento de energia para os vértices do sub-grafo desconectado.

Dois outros estudos da série sobre redes e fluxos do território brasileiro realizados pelo IBGE que podem aplicar estas medidas de centralidade para identificação dos municípios mais importantes são: Gestão do Território (IBGE, 2014a), que consolida as gestões pública e privada dos municípios brasileiros, e Ligações Aéreas (IBGE, 2013), que busca caracterizar os fluxos derivados do transporte aéreo de passageiros e carga.

Uma das medidas utilizadas para avaliar a qualidade das estimativas da centralidade de intermediação foi o coeficiente de variação, utilizada pelos Institutos Nacionais de Estatística como medida de qualidade das estimativas.

A outra medida proposta para avaliar a qualidade das estimativas foi o efeito do plano amostral de Kish. Com o efeito do plano amostral de Kish, a variância de um estimador sob um plano amostral qualquer é comparada com relação a um plano considerado padrão (BOLFARINE e BUSSAB, 2004). Porém esse indicador não ajudou na escolha dos métodos a serem implementados nos algoritmos em função do fato de que, se o vértice não pertence à amostra, sua estimativa de centralidade de intermediação será zero com estimativa nula de variância, além de outras situações que serão discutidas no capítulo de resultados.

No que concerne à mineração em grafos (SAMATOVA *et al.*, 2013), algumas das principais tarefas são a identificação de sub-grafos, detecção de anomalias, cálculo de centralidades, análise de agrupamento, detecção de comunidades, identificação dos top-k vértices ou arestas mais importantes ou influentes. Ao se utilizar dados modelados como grafos é possível otimizar as tarefas de mineração de dados que possuem o objetivo de analisar relacionamentos e fazer buscas. Além da modelagem em grafos, também foi aplicada computação paralela para se obter agilidade do processamento dos dados para estimação da centralidade de intermediação.

Acima foram apresentados o contexto e a importância de amostragem em *big data* e do uso de centralidade de intermediação para identificação de vértices importantes em redes complexas. Também foram apresentados os indicadores de qualidade propostos para avaliação das estimativas de centralidade de intermediação, assim

como a importância da modelagem de dados em grafos para otimizar a extração de informação de redes complexas.

Para atingir os objetivos que serão descritos a seguir, também foi utilizada uma técnica de agrupamento para detecção de comunidades, uma vez que centralidade de intermediação e detecção de comunidades são tarefas com alta correlação. Existem técnicas de detecção de comunidades que excluem os vértices com maior centralidade de intermediação pois estes possuem alta probabilidade de conectar comunidades (SAMATOVA *et al.*, 2013).

O objetivo principal desta tese é implementar e avaliar opções de planos amostrais de algoritmos para cálculo de centralidade de intermediação em redes complexas visando melhorar a qualidade das estimativas. Isto será feito buscando alternativas de planejamento amostral tomando como base o algoritmo apresentado por Riondato e Kornaropoulos (2016) e a análise de agrupamento apresentada por Suppa e Zimeo (2015).

O primeiro algoritmo (RIONDATO e KORNAROPOULOS, 2016) utiliza a dimensão Vapnik-Chervonenkis (FRIEDMAN *et al.*, 2001), da teoria de aprendizado estatístico, para analisar o equilíbrio entre tamanho de amostra e qualidade das estimativas para alguns problemas fundamentais na área de mineração de dados, como regras de associação e centralidade de intermediação em grafos. Além disso, apresenta garantias teóricas sobre a acurácia do estimador para centralidade de intermediação apresentado. Já o segundo algoritmo (SUPPA e ZIMEO, 2015) utiliza análise de agrupamento em grafos e uma classificação dos vértices conforme algumas características topológicas[3] de tal forma que estima com qualidade a centralidade de intermediação, porém sem garantias teóricas, apenas empíricas. Esse algoritmo depende apenas do número de vértices com as mesmas características topológicas enquanto o primeiro depende de alguns caminhos mais curtos sorteados aleatoriamente.

Outro objetivo é de propor indicadores de variabilidade das estimativas para avaliação estatística de qualidade das mesmas que são utilizadas em amostragem, mas não em mineração de dados em redes complexas.

As técnicas utilizadas, de forma combinada, para atingir os objetivos e propor um novo algoritmo foram: amostragem, agrupamento (ou detecção de comunidades) e computação paralela.

Após extensa revisão da literatura e avaliação dos resultados dos experimentos realizados em redes complexas artificiais com propriedades estruturais comuns em redes reais, conclui-se que o algoritmo proposto contribui para o estado da arte da utilização de amostragem para estimar centralidade de intermediação em

[3]As características topológicas são: comunidade a qual pertence o vértice, número de caminhos mínimos e distâncias entre vértices de comunidades diferentes.

grandes redes complexas, um desafio no cenário atual de *big data*, ao agregar várias técnicas que otimizam a extração de conhecimento de dados. O algoritmo proposto, além de melhorar a qualidade das estimativas, apresentou redução no tempo de processamento mantendo a escalabilidade.

Para o desenvolvimento do tema proposto, os capítulos seguintes foram organizados da seguinte forma:

- O Capítulo 2, "Fundamentação teórica", apresenta conceitos sobre amostragem e grafos. Foi feito um resumo com as informações e definições que serão úteis para entendimento da metodologia. Na seção sobre amostragem, existem exemplos de planos amostrais simples e complexos, assim como de utilização do efeito do plano amostral de Kish. Na seção de grafos temos as seguintes subseções: definições (incluindo centralidade), modelos teóricos (modelos matemáticos para gerar grafos com propriedades específicas) e agrupamento (método de Louvain para detecção de comunidades).

- O Capítulo 3, "Revisão da literatura", traz trabalhos relacionados à amostragem aplicada a tarefas de mineração em redes complexas de um modo geral e outros tratando especificamente do cálculo de centralidade de intermediação. Ao final do capítulo existe uma tabela com trabalhos selecionados para ilustrar a evolução do tema no tempo.

- O Capítulo 4, "Metodologia", busca desenhar o processo realizado para atingir os resultados e objetivos desta tese. Ele possui as seguintes seções: dados, algoritmos (Brandes para cálculo exato e outros para cálculos estimados), planos amostrais e implementação.

- O Capítulo 5, "Resultados e discussão", traz o resultado de experimentos realizados com redes complexas com tamanhos variando de 1 mil até 1 milhão de vértices. Também apresenta discussão sobre os mesmos.

- O Capítulo 6, "Considerações finais", além de resumir o trabalho e apresentar os objetivos atingidos, traz sugestões para trabalhos futuros, como por exemplo, adaptar o algoritmo resultado desta tese para estimar centralidade de intermediação em redes complexas dinâmicas.

- No apêndice encontram-se os códigos dos algoritmos de cálculo de centralidade de intermediação, são eles: estimado baseado em amostragem CIA (RIONDATO e KORNAROPOULOS, 2016) usando computação paralela e distribuída, proposto por esta tese CIAVLCM usando computação paralela.

Capítulo 2

Fundamentação teórica

2.1 Amostragem

Em teoria de amostragem temos alguns planos amostrais probabilísticos clássicos: amostragem aleatória simples com ou sem reposição, amostragem estratificada simples, amostragem por conglomerados, dentre outros.

Amostra probabilística é aquela onde cada possível amostra tem uma probabilidade conhecida, a priori, de ocorrer. Desse modo, tem-se toda a teoria de probabilidade e inferência estatística para dar suporte às conclusões. O modo como essas probabilidades são associadas é que irá definir um planejamento amostral (BOLFARINE e BUSSAB, 2004).

Exemplo de amostras não probabilísticas, as que não seguem os critérios acima, são: por conveniência, de voluntários, intencional (ou de corte), por cotas, bola de neve (um indica o outro, seleciona-se os amigos, depois os amigos dos amigos e assim sucessivamente).

Algumas características desejáveis em uma amostra, probabilística ou não, são:

- a capacidade de generalizar estimativas da amostra para a população;
- imparcialidade;
- menor erro amostral possível, dado o custo, tempo e restrições operacionais;
- capacidade de medir a precisão das estimativas;
- simplicidade e possibilidade de se paralelizar a implementação.

O estudo do erro amostral consiste, basicamente, em estudar o comportamento da diferença entre o valor observado na amostra e o parâmetro de interesse na população. Se o valor esperado desta diferença for igual a zero, tem-se um estimador não viesado. Já o valor esperado do quadrado desta diferença, o erro quadrático

médio (EQM), informa sobre a precisão do estimador. Procura-se usualmente estimadores com baixos valores de EQM. Quando o estimador é não viesado, o EQM passa a ser a variância do estimador, calculada em relação à distribuição amostral do estimador. Para recuperar a mesma unidade da variável usa-se o desvio padrão, que é que a raiz quadrada da variância e pode ser visto como indicador do erro médio esperado pelo uso deste estimador e desse plano amostral (BOLFARINE e BUSSAB, 2004).

Para avaliar os estimadores de acordo com o plano amostral adotado tem-se dois indicadores: o Coeficiente de Variação (CV) e o Efeito do Plano Amostral de Kish (EPA) PESSOA e SILVA (1998). O coeficiente de variação é o desvio-padrão sobre a média, ou seja, é uma medida relativa de variabilidade e é um indicador comumente utilizado pelos Institutos Nacionais de Estatística para verificar se os resultados são bons o suficiente para serem divulgados. Um exemplo pode ser visto na Tabela 2.1 a seguir:

Tabela 2.1: Exemplo de classificação quanto a qualidade da estimativa utilizando o coeficiente de variação: *Statistics Canada*.

Nível de qualidade (CV)	Diretriz
0 – 16,5	Aceitável
16,6 – 33,3	Restrito
33,4 ou mais	Inaceitável

Fonte: http://www.statcan.gc.ca/pub/13f0026m/2007001/table/tab5p1-eng.htm

O IBGE ajustou o tamanho da amostra na Pesquisa Nacional por Amostra de Domicílios Contínua para alcançar CV de no máximo 15% para a estimativa de uma determinada variável (IBGE, 2014b).

Já o EPA é a razão entre a variância do estimador sob um plano amostral e a variância do estimador sob outro plano amostral considerado base, ou seja, é utilizado para comparar estratégias amostrais.

Para ilustrar a utilização do EPA vamos comparar duas estratégias amostrais:

- Amostra Aleatória Simples com reposição (AASc) ;
- Amostra Estratificada com alocação proporcional ao tamanho do estrato na população com AASc dentro dos estratos (AEp) .

Numa AASc sorteiam-se, com igual probabilidade, r unidades amostrais de uma lista de N unidades populacionais, retornando essa unidade antes do sorteio da próxima. Quando o elemento sorteado é removido antes do sorteio do próximo, tem-se uma amostra aleatória simples sem reposição (AASs) . Do ponto de vista prático,

uma amostra aleatória simples sem reposição é mais interessante, pois não se ganha mais informação se uma mesma unidade aparece mais de uma vez na amostra. Por outro lado, o plano AASc introduz vantagens estatísticas, como a independência entre as unidades sorteadas, que facilita a determinação das propriedades dos estimadores das quantidades populacionais de interesse. Na AEp, a amostra de tamanho r é distribuída proporcionalmente ao tamanho dos estratos na população N_h com $h = 1, ..., H$, onde H representa a quantidade de estratos.

Suponha que se deseja estimar a média $\overline{Y}$ de uma variável de interesse Y. O estimador usual de $\overline{Y}$ sob o plano AASc é a média amostral $\overline{y}$, cuja variância é dada por (BOLFARINE e BUSSAB, 2004):

$$V_{AASc}(\overline{y}) = \frac{V(Y)}{r} \tag{2.1}$$

Onde r é o tamanho da amostra e $V(Y)$ é a variância da variável original Y.

Sob o plano AEp, a variância de $\overline{y}$ é dada por:

$$V_{AEp}(\overline{y}) = \frac{V_d(Y)}{r} \tag{2.2}$$

É possível provar que (BOLFARINE e BUSSAB, 2004):

$$V_{AASc}(\overline{y}) = V_{AEp}(\overline{y}) + V_e(Y)/r \tag{2.3}$$

Onde:

- $V(Y) = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \overline{Y})^2$, é a variância de Y;
- $V_d(Y) = \frac{1}{N}\sum_{h=1}^{H} N_h V_h(Y)$, é a variância dentro dos estratos;
- $V_e(Y) = \frac{1}{N}\sum_{h=1}^{H} N_h(\overline{Y}_h - \overline{Y})^2$, é a variância entre os estratos;
- $\overline{Y}_h = \frac{1}{N_h}\sum_{i=1}^{N_h} Y_{hi}$, é a média do estrato h;
- $V_h(Y) = \frac{1}{N_h}\sum_{i=1}^{N_h}(Y_{hi} - \overline{Y}_h)^2$, é variância do estrato h.

Desta forma, a variância do estimador da média sob AASc é maior ou igual a variância sob AEp, logo:

$$EPA = \frac{V_{AEp}(\overline{y})}{V_{AASc}(\overline{y})} \leq 1$$

.

Existem ainda os planos amostrais complexos como, por exemplo, os planos utilizados pelo IBGE em suas pesquisas amostrais. Alguns destes planos amostrais complexos apresentados em Bolfarine e Bussab (2004) são: amostragem em 2 ou

mais estágios e amostragem com probabilidade proporcional ao tamanho (PPT) . Um possível plano amostral em dois estágios é:

- 1o estágio: sorteia-se escolas com processo PPT utilizando o número de alunos para medir o tamanho das escolas. Neste caso, as unidades primárias de amostragem (UPA) são as escolas.
- 2o estágio: sorteia-se os alunos por processo AASs. Aqui, as unidades secundárias de amostragem (USA) são os alunos.

Estes planos considerados complexos são os mais comuns na realidade dos INEs.

2.2 Grafos

2.2.1 Definições

Mineração de dados em grafos é uma área de estudo que tem como objetivo o descobrimento de conhecimento e extração de informação em dados representados por grafos(SAMATOVA *et al.*, 2013).

Na Figura 2.1 tem-se o fluxo de um processo de mineração de dados padrão que pode ser aplicado a grafos.

Figura 2.1: Fluxo do processo de mineração de dados.

Nesta parte do trabalho tem-se a definição de alguns conceitos e métricas de grafos e propriedades de redes complexas para facilitar o entendimento dos procedimentos que serão apresentados nos próximos capítulos.

Em sua definição mais geral, uma rede (ou grafo) é uma abstração que permite codificar algum tipo de relacionamento entre pares de objetos. Por exemplo, em redes sociais objetos são geralmente indivíduos e relacionamentos representam algum

tipo de relação social, como amizade ou trabalho em conjunto (FIGUEIREDO, 2011).

Um grafo G é definido por $G = (Vert, E)$, sendo que $Vert$ representa o conjunto de n nós ou vértices e E, o conjunto de m relacionamentos ou arestas (s, t), onde $s, t \in Vert$. Dois vértices s e t são vizinhos se eles estão conectados por uma aresta. Um grafo é dito simples se não possui laços (auto conexões) nem arestas múltiplas (tipos diferentes de relacionamentos entre os vértices s e t).

Alguns exemplos de características estruturais de um grafo são: tamanho, densidade, graus, distâncias, agrupamentos, centralidades, dentre outras. Essas características fornecem uma ideia geral da estrutura do grafo e são importantes porque podem determinar o comportamento geral de algum processo, como epidemias, por exemplo.

Um grafo pode ser direcionado ou não direcionado, conforme pode ser visto na Figura 2.2. Nos grafos direcionados, ou dígrafos, as relações possuem uma direção, como por exemplo, a relação de seguidores no Twitter. Já nos grafos não direcionados, as relações não possuem direção, como por exemplo, amizades no Facebook.

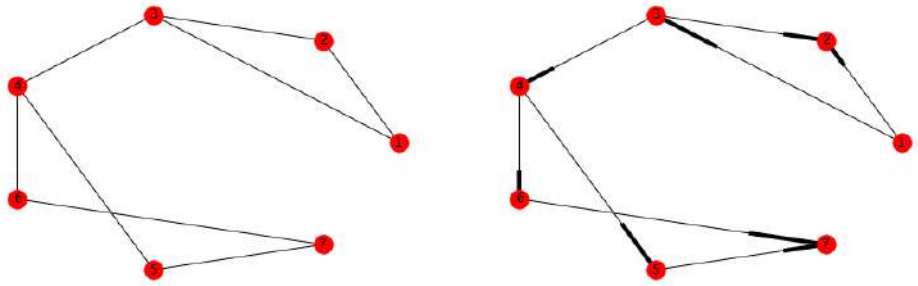

Figura 2.2: Grafos não direcionado e direcionado. As pontas mais escuras indicam a direção da aresta.

Um grafo ponderado é aquele que atribui um peso a cada aresta, representado da forma $w(s, t)$, ou seja, $w(s, t)$ é o peso associado à aresta que une os vértices s e t.

A densidade de um grafo d é fração do número de arestas m sobre o número máximo possível de arestas:

$$d = \frac{m}{\binom{n}{2}} \tag{2.4}$$

O grau de um vértice v, denotado por $g(v)$, é o número de arestas incidentes em v. Um vértice de grau 0 é um vértice isolado. A equação a seguir fornece o grau

médio de um grafo , pois cada aresta aponta para 2 vértices:

$$\overline{g} = \sum_{v \in Vert} \frac{g(v)}{n} = \frac{2m}{n} \tag{2.5}$$

A distribuição de probabilidades dos graus é definida por:

$$P(D = k) = \frac{\textit{número de vértices com grau } k}{n} \tag{2.6}$$

A distância $dist(s, t)$ é o tamanho do caminho mais curto entre os vértices s e t medido em número de arestas entre s e t, ou seja, quantidade de passos necessários para sair da origem s e atingir o vértice destino t.

A excentricidade de um vértice é a maior distância de um vértice v a todos os outros vértices. O diâmetro $diam(G)$ é a maior distância entre dois vértices quaisquer do grafo, ou seja, a maior excentricidade.

A medida $VD(G)$ corresponde ao número de vértices no caminho do diâmetro. Cabe destacar que, para um grafo não ponderado, $VD(G) = diam(G) + 1$.

O agrupamento de um vértice é a tendência de formação de triângulos. O agrupamento em cada vértice $C(v)$ é a probabilidade dos vizinhos do vértice v também serem vizinhos, ou seja, é fração do números de arestas entre os vizinhos de v sobre o número total possível de arestas entre eles.

O agrupamento médio, ou coeficiente de agrupamento, do grafo é dado por:

$$C(G) = \sum_{v \in Vert} \frac{C(v)}{n} \tag{2.7}$$

Outro conceito importante na avaliação de redes complexas, foco deste estudo, é a centralidade, cujo objetivo é ordenar os vértices em relação à sua importância, usando métricas locais ou globais. As centralidades locais dependem apenas da vizinhança do vértice, grau por exemplo, e as globais dependem do grafo inteiro, como por exemplo: intermediação e PageRank™ (PAGE *et al.*, 1999)[1].

Duas destas medidas de centralidade serão descritas em seguida, sendo uma local, a centralidade de grau, e outra global, a centralidade de intermediação (NEWMAN, 2010).

- **Centralidade de Grau**: É definida como o número de arestas incidentes em um vértice. O grau pode ser interpretado como a probabilidade de um vértice receber alguma informação do grafo. O ideal é normalizar esta medida.

  Um vértice com maior grau tem maior número de conexões com outros. Em um processo de comunicação na rede, vértice de grau alto é um canal direto

[1] PageRank™ é um algoritmo utilizado pela ferramenta de busca Google e mede a importância de uma página contabilizando a quantidade e qualidade de links apontando para ela.

de informação, popularidade e influência. São vértices com grande potencial de atividade dentro de uma rede.

Em grafos direcionados é importante diferenciar graus de entrada e saída. Grau de entrada é número de arestas que chegam a um vértice. Grau de saída é o número de arestas que saem de um vértice. Grau é uma medida de centralidade local.

Dois nós com o mesmo grau podem não ter a mesma capacidade de influenciar, por exemplo se o grau é usado para medir influência local, então o poder do vértice depende de quem são os seus vizinhos e do tipo de interação.

- **Centralidade de Intermediação**: Mede o quanto no meio do caminho um vértice está. Foi introduzido por Freeman (1977) como uma medida para quantificar o controle de um ser humano sobre a comunicação entre outros seres humanos numa rede social. Sejam:

  $\sigma_{s,t}(v)$- Número de caminhos mais curtos entre s e t que passam pelo vértice v;

  $\sigma_{s,t}$- Número de caminhos mais curtos entre os vértices s e t

  Então:

$$CI(v) = \sum_{s \neq v \in Vert} \sum_{t \neq v \in Vert} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}} \tag{2.8}$$

  E a centralidade de intermediação padronizada:

$$CIp(v) = \frac{1}{n(n-1)} \sum_{s \neq v \in Vert} \sum_{t \neq v \in Vert} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}} \tag{2.9}$$

  Intermediação mede a frequência relativa com que o vértice aparece no menor caminho entre dois vértices quaisquer. Vértices com alto $CI(v)$ possuem grande potencial de controle do fluxo de informação na rede. Podem ajudar na coordenação de processos dentro de um grupo, influenciar na comunicação da rede, atrasando ou perturbando o fluxo de informação, por exemplo, propagação de doenças em estudos de epidemia.

  Vértices com alta centralidade de intermediação tem potencial para conectar comunidades diferentes. Eliminar vértices de alta intermediação pode ter o efeito de desconectar à rede. Esta propriedade é usada em algoritmos de detecção de comunidades.

Tome como exemplo o grafo da Figura 2.3 abaixo:

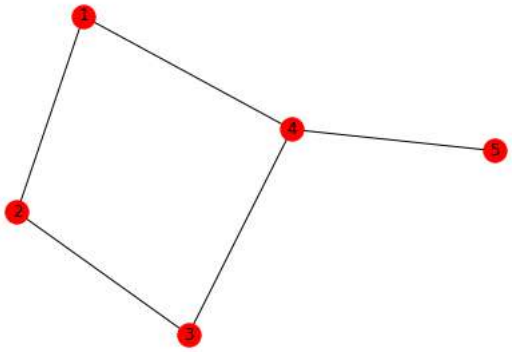

Figura 2.3: Exemplo de cálculo de centralidade $CI(v)$.

A centralidade de intermediação de cada um dos vértices é dada por:

$CI(1) = 1,0$, $CI(2) = 0,5$, $CI(3) = 1,0$, $CI(4) = 3,5$ e $CI(5) = 0,0$.

No caso do vértice 1, por exemplo, existem 2 caminhos mínimos entre os vértices 2 e 5, sendo que um deles passa pelo vértice 1. O mesmo ocorre entre os vértices 2 e 4. Logo $CI(1) = 1/2 + 1/2 = 1$.

Estas métricas de centralidade são as mais utilizadas e intuitivas para se atribuir importância aos vértices de uma rede.

Exemplos das centralidades descritas acima numa rede de co-autoria:

- Grau: autores que possuem muitos co-autores publicando artigos.
- Intermediação: autores que desempenham um papel crucial na ligação de diferentes comunidades.

Na Figura 2.4, o autor com maior centralidade de grau é o A e o de maior centralidade de intermediação é o D.

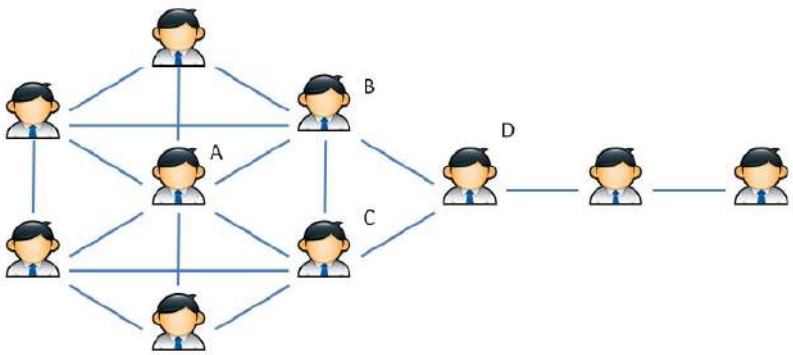

Figura 2.4: Rede de co-autoria. Fonte: http://matteo.rionda.to/centrtutorial/.

Para se escolher uma medida de centralidade, é necessária uma referência externa que depende do contexto e do objetivo da ordenação. Numa rede elétrica, o interesse está nos vértices com maiores centralidades de intermediação pois são os vértices que interligam as comunidades dentro da rede.

Nesta seção, foram descritas as métricas de teoria dos grafos utilizadas nesta tese.

2.2.2 Modelos teóricos e redes complexas

Redes complexas são grafos que apresentam propriedades estruturais não encontradas em grafos simples.

Conforme Albert e Barabási (2002):

> "Redes complexas descrevem uma ampla gama de sistemas na natureza e na sociedade, dois exemplos muito citados são: a célula, uma rede de produtos químicos ligados por reações químicas, e a Internet, uma rede de roteadores e computadores conectados por conexões físicas. É cada vez mais reconhecido que a topologia e a evolução das redes reais são governadas por princípios organizacionais robustos."

As características, ou propriedades, principais de redes complexas são: propriedade mundo pequeno, coeficiente de agrupamento maior que em redes aleatórias e distribuição de graus que segue uma lei de potência. Essas características são comuns em muitas redes reais. Outras duas características que muitas redes reais possuem em comum são o fato de serem esparsas, ou seja, possuem bem menos arestas do que poderiam ter (densidade menor que 1); e o fato de que uma parte do grafo está mais conectada entre si que com o resto do grafo, ou seja, são modulares, formam comunidades.

A seguir, tem-se a descrição de alguns modelos teóricos de redes complexas:

- Aleatório de Erdös-Rényi:

  Erdös e Rényi (1959, 1960) começaram a estudar grafos como objetos estocásticos e não determinísticos, o que os levou à introdução do conceito de grafos aleatórios como um conjunto de vértices conectados aleatoriamente. Estes modelos também são chamados de modelo $G_{n,p}$ ou binomial.

- Mundo pequeno de Watts-Strogatz:

  Milgram (1967), um pesquisador de sociologia em Harvard, investigou a hipótese de "mundo pequeno", ou seja, de que há poucas pessoas separando duas outras pessoas quaisquer no mundo. No seu experimento de investigação, Milgram pediu a várias pessoas de distintos lugares que enviassem cartas a

alguns determinados destinatários. Ele descobriu que a maioria das cartas acabavam chegando, dando origem a teoria dos "6 graus de separação", que seria a quantidade de pessoas necessárias para a carta chegar ao destino. Ou seja, a propriedade mundo pequeno diz que os vértices são separados por distâncias pequenas. Watts e Strogatz (1998) descobriram que o fenômeno de mundo pequeno também é observado em outras redes reais como é o caso da rede elétrica dos Estados Unidos.

- Redes sem escala de Barabási-Albert:

  Barabási e Albert (1999) descobriram que a distribuição de grau dos vértices em redes complexas, como a *World Wide Web (WWW)*, segue uma lei de potência, ou seja, a distribuição de graus é sem escala e é da forma $P(D = k) = K^{-\lambda}$. De acordo com Newman (2003), λ está entre 2 e 3.

2.2.3 Agrupamento

Girvan e Newman (2002) mostraram que a maioria das redes reais possuem estrutura modular, isto é, elas possuem grupos de vértices com mais conexões entre si do que com o restante da rede. Um bom exemplo de redes com esta estrutura são as redes sociais onde os indivíduos com mesma opinião tendem a pertencer à mesma comunidade.

Na Figura 2.5, tem-se um exemplo de rede com 3 comunidades, no qual três partes do grafo têm vértices mais conectados entre si que com o resto do grafo.

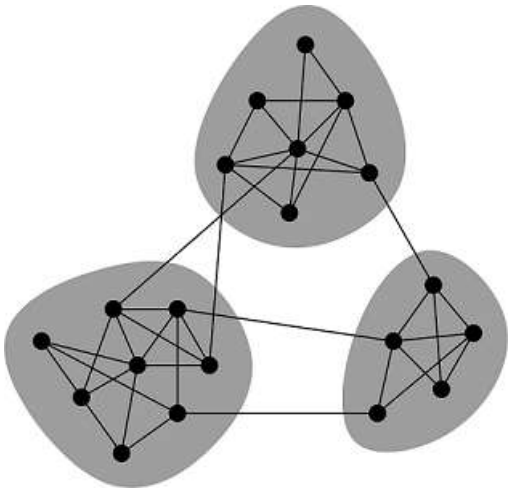

Figura 2.5: Exemplo de rede complexa não direcionada e sem peso com 3 comunidades. Fonte: http://www.pnas.org/content/103/23/8577.full.

O método Louvain (BLONDEL *et al.*, 2008) é um algoritmo heurístico utilizado para detecção de comunidades baseado em otimização da medida de modularidade Q(BLONDEL *et al.*, 2008; CLAUSET *et al.*, 2004):

$$Q = \sum_{k=1}^{K} (e_{kk} - a_k^2) \tag{2.10}$$

Na Equação 2.10, k representa uma das K comunidades, e_{kk} é a fração de arestas que estão inteiramente na comunidade k e a_k é a fração de arestas que possuem pelo menos um extremo na comunidade k. A modularidade assume valores entre -1 e 1, sendo que, quanto mais próximo de 1, melhores são as estruturas de comunidades encontradas.

Também é um método de natureza aglomerativa, assim como o algoritmo de Brandes (2008), o que favorece a implementação de programação paralela. Assumindo que a entrada é uma rede ponderada com n vértices, o funcionamento do método é dividido em duas fases.

Na primeira fase, cada vértice é considerado uma comunidade. Após, para cada vértice v, considera-se cada um de seus vizinhos e avalia-se o ganho de modularidade que ocorreria se o vértice v fosse removido de sua comunidade e colocado na comunidade do vizinho. Ao final da avaliação em todos os vizinhos, o vértice v é colocado na comunidade onde o ganho é máximo, mas apenas se o ganho for positivo. Senão, o vértice permanece em sua comunidade. Esse processo se repete até que nenhum indivíduo possa melhorar a modularidade.

A segunda fase consiste em construir uma nova rede onde os vértices são as comunidades encontradas na primeira fase. Para isto, os pesos das conexões entre os novos vértices são dados pela soma dos pesos das arestas entre vértices nas duas comunidades correspondentes. Conexões entre vértices na mesma comunidade tornam-se laços nesta comunidade na nova rede. Uma vez que a segunda fase esteja concluída, a primeira fase é reaplicada na nova rede.

Os processo prossegue até que não ocorram mais mudanças e a máxima modularidade seja obtida.

Esse método consegue um bom desempenho em redes de tamanhos variados. Além disso, o método obtém melhores valores de modularidade em comparação aos obtidos pelos algoritmos Newman e Girvan (2004).

O método Louvain possui vantagens interessantes sobre os outros métodos, como seu tempo de execução linear e comunidades com maior qualidade, de acordo com os valores de modularidade obtidos em comparação aos demais algoritmos.

2.3 Amostragem em grafos

Nesta tese não se busca uma amostra considerada representativa de todo o grafo, e sim uma amostra de caminhos mínimos que permita estimar a centralidade de intermediação com acurácia, precisão e bom tempo de processamento, principalmente para os top-k vértices mais importantes de acordo com esta medida.

Na tese "Efeito da amostragem nas propriedades topológicas de redes complexas"(BOAS, 2008), existe uma análise sobre amostragem aplicada em redes complexas. Nesse estudo, o autor mostra que o processo de amostragem pode resultar em redes estudadas com estrutura diferente das redes originais, levando a caracterização, classificação e modelagem incorreta. Além disso, os processos dinâmicos como propagação de opiniões e doenças (PASTOR-SATORRAS e VESPIGNANI, 2001), por serem dependentes da estrutura da rede, são afetados pelo processo de amostragem empregado.

Cormode e Duffield (2014), no tutorial *"Sampling for Big Data"* apresentado na conferência internacional anual ACM SIGKDD (*ACM Special Interest Group on Knowledge Discovery and Data Mining*) de 2014, apresentaram a amostragem como uma forma de otimizar a obtenção de resultados em vários problemas envolvendo grandes volumes de dados. O trabalho apresentou a amostragem como uma forma de otimizar a obtenção de resultados em vários problemas envolvendo grandes volumes de dados, com aplicações para grafos e fluxo de dados.

Para fluxo contínuo de dados foram apresentados alguns planos amostrais como amostragem ponderada via *"reservoir"*(TILLÉ, 2006), amostragem com probabilidade de inclusão proporcional ao tamanho com estimador de total não-viesado de Horvitz-Thompson (COHEN *et al.*, 2011), dentre outros.

E para amostragem em grafos, eles destacam objetivos distintos como: estudos locais (vértices e arestas) e globais (quantidade de caminhos mínimos), assim como seleção de subgrafo representativo para que os algoritmos aprendam mais rápido no subgrafo do que usando todo o grafo. No caso em que a amostra deve ser um subgrafo representativo, deve-se verificar quais características do grafo devem ser preservadas como, por exemplo, distribuição de grau. Os autores do tutorial agrupam os modelos de amostragem para grafos em dois grandes grupos:

- Estáticos: são modelos onde se tem acesso a todo o grafo para seleção da amostra.
- Dinâmicos: são modelos que possuem arestas ou vértices chegando arbitrariamente.

Também apresentam algumas heurísticas, algoritmos com plano amostral não probabilístico como, por exemplo:

- **Amostragem bola de neve**: usa busca em largura, ou seja, são sorteados vários vértices iniciais e depois seus vizinhos. Para esse plano amostral não-probabilístico existe viés pois há uma maior probabilidade de vértices com grau alto serem sorteados;

- **Passeios aleatórios**: é uma técnica que se mostra eficiente para cálculo da centralidade de PageRank^{TM};

- **Amostragem baseada em contagem de triângulos e coeficiente de agrupamento**: eficiente mas viesada pois é um tipo de amostragem preferencial, onde os vértices com maiores probabilidades de formarem triângulos, e consequentemente subgrafos conexos, possuem maior probabilidade de serem selecionados. A abordagem baseada em contagem de triângulos pode ser aplicada em grafo de fluxos de dados (*"stream graph"*) (AHMED *et al.*, 2014a,b).

Cormode e Duffield (2014) deixam em aberto a análise de como a amostragem afeta as tarefas de mineração de dados e aprendizado de máquina e sugerem estudos mais específicos, como ao invés de "amostragem+agrupamento", "probabilidade de inclusão proporcional ao tamanho (PPT)+k-médias". Nesta tese foi estudado um caso específico como sugerido pelos autores, onde o plano amostral é o definido em Riondato e Kornaropoulos (2016) e a técnica de agrupamento sugerida por Suppa e Zimeo (2015).

Capítulo 3

Revisão da literatura

As características para se escolher qual algoritmo adotar para estimar centralidade de intermediação em grandes redes complexas envolvem tamanho da amostra, qualidades teóricas e empíricas dos estimadores, simplicidade de implementação, possibilidade de rodar processos em paralelo e rapidez na execução de consultas analíticas. No geral, as amostras probabilísticas apresentam garantias teóricas de qualidade. Porém, existem amostras não probabilísticas que apresentam garantias empíricas, ou seja, podem apresentar menor viés, menor variabilidade, melhor implementação mesmo sem as garantias teóricas. Um exemplo são os métodos apresentados por Riondato e Kornaropoulos (2016), com garantias teóricas, e o de Geisberger, Sanders, e Schultes (2008), maior acurácia sem garantias teóricas, apenas empíricas. Ambos os algoritmos encontram-se implementados no pacote NetworKit (STAUDT *et al.*, 2016). Riondato e Upfal (2015) também apresentam aplicações de amostragem para busca de itens frequentes e regras de associação, uma tarefa de mineração de dados utilizada em sistemas de recomendação como, por exemplo, indicação de filmes para usuários da Netflix.

Um exemplo de utilização de técnicas de amostragem aplicadas a grandes volumes de dados é o estudo para estimar o número de sítios com domínio ".com.br"(SILVA *et al.*, 2014). Nesse estudo, verificou-se um conjunto inicial de aproximadamente 12 mil domínios ".gov.br" e sítios redirecionados que também fossem ".gov.br". Esse censo de sítios ".gov.br" foi coletado em 3 semanas, enquanto que a realização de um estudo similar para o domínio ".com.br" levaria aproximadamente 11 anos, o que ratifica a necessidade da utilização da amostragem.

Outros dois exemplos de utilização de amostragem para tarefas de mineração de dados são da Amazon (LINDEN *et al.*, 2003) e do Facebook (GJOKA *et al.*, 2010). A Amazon utiliza amostragem para desenvolvimento de seus sistemas de recomendação. O Facebook apresenta formas de selecionar usuários uniformemente, como passeio aleatório via Metropolis-Hastings e passeio aleatório reponderado. Tais opções de técnicas de amostragem fornecem estimativas não-viesadas para algumas

propriedades dos usuários como, por exemplo, a centralidade de intermediação.

Para a execução dessa tarefa de mineração em grafos foi utilizado o algoritmo estado da arte de Brandes (2001) para o cálculo exato da centralidade de intermediação de todos os vértices. Os cálculos exatos podem ser feitos com o algoritmo de Brandes, de complexidade $O(nm)$ para grafos não ponderados, e $O(nm + n^2 \log n)$ para grafos ponderados. Foram implementados e avaliados outros algoritmos para estimar a centralidade de intermediação que serão apresentados no capítulo sobre metodologia.

Riondato e Kornaropoulos (2016) provam que seu algoritmo é de três a quatro vezes mais rápido do que outros algoritmos que estimam a centralidade de intermediação(BRANDES e PICH, 2007; JACOB *et al.*, 2005), com menos trabalho computacional e com garantias de eficiência sobre a estimativa. JACOB *et al.* (2005) e Brandes e Pich (2007) apresentaram algoritmos que imitam o exato com a diferença de que não calculam as contribuições de todos os vértices para a centralidade de intermediação considerando apenas a contribuição de alguns vértices sorteados aleatoriamente. Brandes e Pich (2007) escolhem alguns vértices como pivôs para aplicar um algoritmo de busca de caminhos mais curtos, sem resultados significantes.

Suppa e Zimeo (2015) conseguem apresentar um algoritmo baseado em agrupamento pelo método de Louvain para detecção de comunidades, descrito no capítulo anterior, e em classificação dos vértices em redes artificiais que simulam a rede social Facebook, obtendo bons resultados em termos de eficiência e escalabilidade. Suppa e Zimeo (2015) se fazem valer da propriedade de que vértices com maior valor de centralidade de intermediação tendem a conectar comunidades.

Thompson (1998) apresenta amostragem utilizando técnicas adaptativas. De acordo com Thompson (2012):

> "Amostragem adaptativa é aquela na qual o procedimento de seleção das unidades a serem incluídas na amostra pode depender dos valores das variáveis de interesse observados durante a pesquisa, por exemplo, estudos sobre risco de transmissão de HIV, onde planejamento adaptativo de rastreamento através de relacionamentos sociais é a única forma de se obter uma amostra de tamanho suficiente para a realização do estudo."

Thompson (2012) apresenta uma revisão de estratégias amostrais para dados estruturados como grafos que ele chama de rastreamento de relacionamentos (*link-tracing designs*):

- **Amostragem bola de neve** (também mencionada por Cormode e Duffield (2014)): é solicitado para uma amostra inicial de indivíduos que indiquem

uma quantidade determinada indivíduos, e assim repete-se o procedimento sucessivamente para um número fixo de vezes. Alguns autores como por exemplo Frank e Snijders (1994) desenvolvem métodos para estimar o tamanho de populações escondidas ou difíceis de atingir considerando o plano amostral por bola de neve.

- **Amostragem de rede ou multiplicidade**: relacionamentos sociais, de parentesco ou administrativos são utilizados para obter unidades amostrais adicionais. Também é um processo com estimativas viesadas.

- **Passeios aleatórios**: amostra onde apenas um indivíduo é selecionado aleatoriamente para entrar na amostra (THOMPSON, 2006).

- **Amostras intencionais**: se escolhe elementos de populações de difícil acesso através de mapeamento etnográfico usado para estratificação da amostra (THOMPSON, 2006).

- **Conglomerado adaptativo**: é uma classe de planos amostrais onde unidades vizinhas são adicionadas à amostra quando um valor observado satisfaz um critério. Numa amostra espacial, a vizinhança é definida geograficamente, enquanto que, numa amostra em grafos, os vizinhos são determinados pelas conexões sociais. Esta estratégia amostral proporciona estimadores não-viesados quando o procedimento de seleção depende de valores associados tanto ao vértice quanto à aresta, inclusive para grafos direcionados (DRYVER e THOMPSON, 2005).

- **Estratificada com alocação adaptativa**: são amostras onde a alocação num estrato depende da observação, durante a pesquisa, de uma variável de interesse. Estratégias de alocação adaptativa são descritas em Thompson e Seber (1995).

Note que tanto bola de neve quanto passeios aleatórios são técnicas amplamente utilizadas para amostragem em grafos. Por serem técnicas que conduzem a estimadores viesados, deixam o desafio de se utilizar estas amostras com cautela para se obter resultados confiáveis. Zhang e Patone (2017) resumem a teoria existente de amostragem para grafos e desenvolvem uma abordagem geral para o estimador de Horvitz-Thompson para amostragem em bola de neve.

Com relação a amostragem em grafos dinâmicos, importante para análises em tempo real, podemos destacar os seguintes trabalhos:

- AHMED *et al.* (2014a) Nesse artigo é proposto um algoritmo de amostragem para análise de grandes grafos, chamado amostragem com retenção (*Graph*

Sample and Hold ou gSH). O gSH essencialmente mantém uma pequena quantidade de estado e passa por todas as arestas do grafo de maneira contínua. O gSH fornece uma estrutura genérica para a estimativa imparcial das contagens de subgrafos, usando a abordagem de Horvitz-Thompson, na qual a contagem de qualquer objeto amostrado é ponderada pela divisão por sua probabilidade de seleção.

- Ahmed, Neville, e Kompella (2014b) apresentam uma estrutura para o problema geral de amostragem em grafos. Além disso, propõem modelos computacionais para amostragem em grafos, variando do modelo tradicionalmente estudado baseado na suposição de um domínio estático para um modelo mais desafiador que é apropriado para domínios dinâmicos que preservam eficientemente muitas das propriedades topológicas dos grafos de entrada.

- Riondato e Upfal (2016) apresentam o algoritmo ABRA (*Approximating Betweenness Centrality in Static and Dynamic Graphs with Rademacher Averages*). O algoritmo estima a centralidade de intermediação utilizando médias Rademacher, um conceito de teoria do aprendizado estatístico, e amostragem progressiva para garantir boa acurácia para as estimativas. O objetivo da amostragem progressiva é começar com amostras pequenas e progressivamente aumentá-las, desde que a precisão do modelo melhore o suficiente de acordo com uma condição de parada.

- Bergamini e Meyerhenke (2016) apresentam o primeiro algoritmo para estimar centralidade de intermediação com computação realizada em memória, resultando em ganho de velocidade de processamento em grafos com milhões de vértices. A acurácia da estimativa é inferior às garantias teóricas e a classificação dos vértices mais importantes é bem preservada, principalmente para os vértices com maior centralidade de intermediação. Bergamini e Meyerhenke (2016) possuem alguns algoritmos implementados no pacote NetworKit (STAUDT *et al.*, 2016), dentre eles, um para cálculo de centralidade de intermediação de um vértice específico após a inserção de uma nova conexão ou aresta (BERGAMINI *et al.*, 2017).

A Tabela 3.1 apresenta alguns dos principais estudos de amostragem aplicada à tarefa de cálculo de centralidade de intermediação em grafos organizados em ordem cronológica.

Tabela 3.1: Linha do tempo de artigos relevantes sobre a utilização de amostragem para grandes volumes de dados, ênfase em grafos.

Autores	Título
(THOMPSON, 1998)	Adaptive sampling in graphs
(BARABÁSI e ALBERT, 1999)	Emergence of scaling in random networks
(BRANDES, 2001) (algoritmo exato)	A faster algorithm for betweenness centrality
(JACOB *et al.*, 2005)	Algorithms for centrality indices
(BRANDES e PICH, 2007)	Centrality estimation in large networks
(GEISBERGER *et al.*, 2008)	Better approximation of betweenness centrality
(AHMED *et al.*, 2014b)	Network sampling: from static to streaming graphs
(CORMODE e DUFFIELD, 2014)	Sampling for big data
(AHMED *et al.*, 2014a)	Graph sample and hold: a framework for big-graph analytics
(SUPPA e ZIMEO, 2015)	A clustered approach for fast computation of betweenness centrality in social networks
(RIONDATO e UPFAL, 2016)	ABRA: Approximating Betweenness Centrality in Static and Dynamic Graphs with Rademacher Averages
(BERGAMINI e MEYERHENKE, 2016)	Approximating betweenness centrality in fully dynamic networks
(RIONDATO e KORNAROPOULOS, 2016)	Fast approximation of betweenness centrality through sampling
(ZHANG e PATONE, 2017)	Graph sampling
(SOUZA e EBECKEN, 2018)	Sampling for large data volumes: an application on complex networks

Capítulo 4

Metodologia

4.1 Dados

Neste trabalho, as redes artificiais estudadas são do tipo Barabási-Albert (BARABÁSI, 2014), ou sem escala, e foram geradas pelo pacote Networkx (HAGBERG *et al.*, 2018), ou seja, as ligações (arestas) são criadas seguindo o modelo de conexão preferencial, onde as novas arestas se conectam aos vértices de maior grau. O modelo de conexão preferencial também é conhecido como propriedade de redes onde "o rico cada vez fica mais rico".

Estas redes artificiais foram geradas de tal forma que possuíssem as seguinte propriedades de redes complexas reais:

- Mundo pequeno;
- Alta modularidade;
- Sem escala e
- Esparsas.

Com as redes artificiais sendo geradas com as propriedades acima, este estudo pode ser aplicado em várias redes reais. Alguns repositórios de dados para realização deste estudo em redes reais são: projeto de análise de redes de Stanford (*Stanford Network Analysis Project*)[1]; Kaggle[2], uma plataforma de competição mundial em aprendizado de máquina; 10^o desafio DIMACS (*Center for discrete mathematics & theoretical computer science*)[3]; e *Network Repository*[4], que funciona como um repositório de repositórios.

[1] http://snap.stanford.edu/
[2] https://www.kaggle.com/datasets
[3] https://www.cc.gatech.edu/dimacs10/downloads.shtml
[4] http://networkrepository.com/

4.2 Algoritmos

4.2.1 Algoritmo exato de Brandes

O algoritmo estado da arte para cálculo exato da centralidade de intermediação é o algoritmo de Brandes (2001). Este algoritmo baseia sua eficiência em uma técnica de acumulação.

A partir de um vértice fonte de cada vez, são computados todos os caminhos mais curtos partindo deste vértice para todos os outros vértices (*Single Source Shortest Paths*-SSSP). A centralidade de intermediação de um vértice v, $CI(v)$, é obtida a partir da soma das contribuições de todos os caminhos mais curtos do grafo. Dado um vértice fonte (ou pivô) inicial s, um vértice destino t e um vértice genérico v, Brandes (2001) define a dependência de pares de s e t em v da seguinte forma:

$$\delta_{s,t}(v) = \frac{\sigma_{s,t}(v)}{\sigma_{s,t}} \tag{4.1}$$

De acordo com a Equação (2.8), é possível calcular a centralidade de intermediação de v, somando a dependência de cada par de vértices em v. Para reduzir a complexidade, Brandes (2001) também introduz o conceito de dependência de s em v como sendo:

$$\delta_{s.}(v) = \sum_{t \in Vert} \delta_{s,t}(v) \tag{4.2}$$

Logo:

$$CI(v) = \sum_{s \in Vert} \delta_{s.}(v) \tag{4.3}$$

Se um vértice v é antecessor de outro vértice w num caminho mais curto que comece em s, v também é antecessor em qualquer outro caminho mínimo que comece em s e passe por w. Desta forma pode-se reescrever a Equação (4.2) da seguinte forma:

$$\delta_{s.}(v) = \sum_{w,v \in P_s(w)} \frac{\sigma_{s,v}}{\sigma_{s,w}}(1 + \delta_{s.}(w)) \tag{4.4}$$

Onde $Ps(w)$ é o conjunto de antecessores diretos de um vértice w nos caminhos mais curtos de s para w, encontrado com algoritmo de busca em largura (BARABÁSI, 2014), para grafos não ponderados, e com algoritmo de Dijkstra (1959), para grafos ponderados.

Abaixo está o pseudocódigo do algoritmo de Brandes (2001) para cálculo exato da centralidade de intermediação para grafos não ponderados.

Algoritmo 1: Calcula valores exatos da centralidade de intermediação

Entrada: Grafo $G = (Vert, E)$ com $|V| = n$ vértices

Saída: Conjunto de valores exatos $CI(v)$ das centralidades de intermediação

```
1  CI(v) ← 0, v ∈ Vert
2  para cada s ∈ Vert faça
3      S ← []
4      P_s[w] ← [], w ∈ Vert
5      σ[t] ← 0, t ∈ Vert
6      σ[s] ← 1
7      d[t] ← −1, t ∈ Vert
8      d[s] ← 0
9      Queue ← [s]; inicia fila
10     enquanto Queue faça
11         v ← Queue; S ← v
12         para cada vizinho w de v faça
13             // w já foi visitado?
14             se d[w] < 0 então
15                 Queue ← w; d[w] ← d[v] + 1
16             fim
17             // w existe caminho mais curto para w via v?
18             se d[w] = d[v] + 1 então
19                 σ[w] ← σ[w] + σ[v]; P[w] ← v
20             fim
21         fim
22     fim
23     δ[v] ← 0, v ∈ Vert; // S retorna vértices em ordem não-crescente de distância de s
24     enquanto S faça
25         S ← w
26         para v ∈ P_s[w] faça
27             δ[v] ← δ[v] + (σ[v]/σ[w]) * (1 + δ[w])
28         fim
29         se w ≠ s então
30             CI[w] ← CI[w] + δ[w]
31         fim
32     fim
33 fim
```

4.2.2 Algoritmo para estimar centralidade de intermediação baseado em amostragem (CIA)

No algoritmo base deste trabalho (RIONDATO e KORNAROPOULOS, 2016), utilizou-se a dimensão Vapnik-Chervonenkis (VC) (FRIEDMAN *et al.*, 2001) para calcular um tamanho da amostra suficiente para obter uma estimativa de alta qualidade, ou seja, com o tamanho r de caminhos mais curtos tem-se uma estimativa da centralidade de intermediação com uma margem de erro ε do verdadeiro resultado com probabilidade mínima de $1 - \delta$. O plano amostral e teoria presentes neste algoritmo são apresentados na seção sobre planos amostrais complexos deste capítulo.

O algoritmo inicia um vetor de valores de centralidade de intermediação para cada vértice no grafo com o valor zero e calcula o tamanho r de caminhos mais curtos em função do valor estimado do número de vértices no caminho do diâmetro, $VD\,(G)$.

Então, o algoritmo realiza passos descritos a seguir r vezes:

1. Estima $VD(G)$;
2. Sorteia um par de vértices s e t;
3. O algoritmo calcula todos os caminhos mais curtos entre (s, t), $SP_{s,t}$, utilizando o algoritmo de busca em largura (BARABÁSI, 2014), para grafos não ponderados e Dijkstra (1959) para grafos ponderados;
4. Faz a variável auxiliar u=t. Sorteia um vértice antecessor, z, com probabilidade proporcional ao tamanho, ou seja, com probabilidade $\sigma_{s,z}/\sigma_{s,u}$. Este procedimento é realizado enquanto v for diferente de s;
5. Por fim, atualiza a centralidade de intermediação estimada para esse vértice sorteado z adicionando $1/r$, e assim sucessivamente até chegar no início do caminho.

Ao final do algoritmo, a amostra final será composta pelos r caminhos mais curtos encontrados pelos passos descritos anteriormente. Em termos de vértices, a amostra será composta pelos vértices internos a esses r caminhos mais curtos.

O número de vértices no maior caminho mais curto, $VD(G)$, é estimado pois, se o algoritmo fosse calcular o valor exato, poderia calcular a centralidade de intermediação exata também, pois calcularia todos os caminhos mais curtos. Para o estimar $VD(G)$, Riondato e Kornaropoulos (2016) calculam os caminhos mais curtos que partem de um vértice fonte, SSSP, escolhido aleatoriamente, s, e estimam $VD(G)$ pela soma dos comprimentos dos dois maiores caminhos mais curtos para

grafos conectados, não ponderados e não direcionados. Para as classes de grafos restantes (direcionadas e / ou ponderadas), os autores estimam $VD(G)$ com o tamanho da maior componente conexa. Nesta tese, foi usado o método de estimação do diâmetro do pacote NetworKit (STAUDT *et al.*, 2016) que, por sua vez, utiliza o algoritmo ANF do artigo *"A fast and scalable tool for data mining in massive graphs"* (PALMER *et al.*, 2002), pois ele foi mais veloz que o proposto por Riondato e Kornaropoulos (2016) e descrito no parágrafo anterior para estimar $VD(G)$. Para se estimar o diâmetro, o grafo tem que ser conexo, ou seja, para todo par de vértices deve haver um caminho os conectando. Então, para casos onde os grafos não são conexos, utiliza-se a maior componente conexa do mesmo.

Esse algoritmo pode ser aplicado em redes complexas ponderadas (pesos positivos) ou não, direcionadas ou não, sem laços e sem múltiplas arestas. A centralidade de intermediação dos algoritmos apresentados nesta tese está baseada em vértices, ou seja, deseja-se saber qual o vértice mais no meio dos caminhos mais curtos. O algoritmo de Riondato e Kornaropoulos (2016) também pode ser utilizado para estimar a centralidade de intermediação de arestas ao invés de vértices, ou seja, quando se deseja estimar qual a aresta que está mais no meio de todos os caminhos mais curtos.

Abaixo está o algoritmo baseado em amostragem CIA (RIONDATO e KORNAROPOULOS, 2016) para grafos ponderados ou não.

Algoritmo 2: Calcula valores estimados para a centralidade de intermediação utilizando técnicas de amostragem

Entrada: Grafo $G = (Vert, E)$ com $|Vert| = n$ vértices; ε, $\delta \in (0, 1)$

Saída: Conjunto de valores estimados $\tilde{CIp}(v)$ das centralidades de intermediação padronizada para todos os vértices em $Vert$

```
1  C̃Ip(v) ← 0
2  VD(G) ← ṼD(G)
3  r ← 1/(2ε²) [⌊log_2 (VD(G) − 2)⌋ + 1 + ln(1/δ)]
4  para cada i de 1 até r faça
5      //sorteia por AASs (s, t)
6      SP_{s,t} ← todos os caminhos mais curtos entre s e t
7      se SP_{s,t} ≠ ∅ então
8          atualiza as estimativas
9          u ← t
10         enquanto u ≠ s faça
11             sorteia z com probabilidade (σ_{s,z}/σ_{s,u})
12             se z ≠ s então
13                 C̃Ip(v) ← C̃Ip(v) + 1/r
14                 u ← z
15             fim
16         fim
17     fim
18     i ← i + 1
19 fim
```

4.2.3 Algoritmo baseado em amostragem e agrupamento proposto (CIAVLCM)

Como visto na revisão da literatura, os principais algoritmos para estimar centralidade de intermediação em redes complexas são:

- Brandes e Pich (2007) escolhem alguns vértices como pivôs para aplicar um algoritmo de busca de caminhos mais curtos, sem resultados significantes. Estimador não viesado e com garantias sobre o erro absoluto.

- Geisberger, Sanders, e Schultes (2008) usam raciocínio similar ao de Brandes e Pich (2007), porém fazem buscas SSSP executadas para frente ou para trás, melhorando a acurácia ao atribuir menos peso às contribuições dos vértices próximos dos selecionados na amostra.

- Suppa e Zimeo (2015) conseguem apresentar um algoritmo baseado em agrupamento e em classificação dos vértices em redes artificiais, obtendo bons resultados em termos de eficiência e escalabilidade.

- Riondato e Kornaropoulos (2016) provam que seu algoritmo é de três a quatro vezes mais rápido do que o de Brandes e Pich (2007), com menos trabalho computacional e com garantias de eficiência sobre a estimativa. Ao invés de usar amostra de vértices, utilizam amostra de caminhos mais curtos, o que permite provar as garantias teóricas.

Foram avaliados vários algoritmos da literatura para a proposta de um novo algoritmo. Vários experimentos foram realizados com formas diferentes de sorteio dos pares de vértices s e t no algoritmo CIA. Inspirado nos conceitos de agrupamento utilizados no algoritmo de Suppa e Zimeo (2015), foram elaboradas as seguintes estratégias para sorteio de s e t utilizando as comunidades encontradas pelo método Louvain (BLONDEL *et al.*, 2008):

- Vértices limítrofes (CIAVL): sorteio nos vértices limítrofes das comunidades. Vértices limítrofes são vértices que unem comunidades. Estes vértices limítrofes podem pertencer à mesma comunidade;

- Vértices limítrofes em comunidades diferentes (CIAVLCM): sorteio nos vértices limítrofes que pertencem a comunidades diferentes;

- Vértices em comunidades diferentes (CIACM): sorteio de s e t em comunidades diferentes.

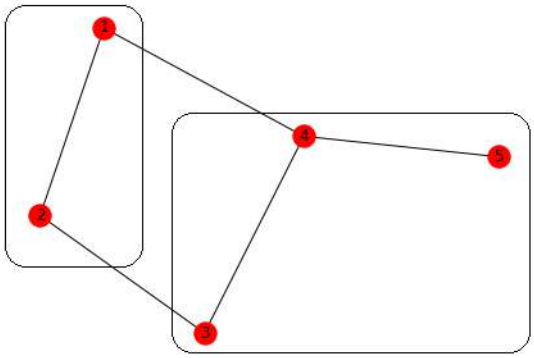

Figura 4.1: Vértices limítrofes.

Para ilustrar os grupos apresentados observe a Figura 4.1. Os vértices 1, 2, 3 e 4 são limítrofes pois unem as duas comunidades, e os vértices 1 e 2 pertencem à mesma comunidade.

Dentre as três propostas acima foi selecionada a que apresentou: menor EQM; menor tempo de processamento mantendo escalabilidade e menor CV. O algoritmo proposto por esta tese buscou combinar formas de otimizar o processo, usando computação paralela e distribuída, e de melhorar acurácia e precisão ao utilizar, associando ao método proposto por Riondato e Kornaropoulos (2016) a divisão do grafo em comunidades, uma vez que centralidade de intermediação tem alta correlação com detecção de comunidades, pois os vértices com maior centralidade tendem a unir comunidades.

Abaixo encontra-se o algoritmo proposto CIAVLVM, que combina conceitos de amostragem e agrupamento.

Algoritmo 3: Calcula valores estimados para a centralidade de intermediação utilizando técnicas de amostragem e agrupamento

Entrada: Grafo $G = (Vert, E)$ com $|Vert| = n$ vértices; ε, $\delta \in (0, 1)$

Saída: Conjunto de valores estimados $\tilde{CIp}(v)$ das centralidades de intermediação padronizada para todos os vértices em $Vert$

```
1  C̃Ip(v) ← 0
2  VD(G) ← ṼD(G)
3  r ← 1/(2ε²) [⌊log₂ (VD(G) − 2)⌋ + 1 + ln(1/δ)]
4  CM(v) ← comunidade de v
5  VL ← vértices limítrofes
6  para cada i de 1 até r faça
7      //sorteia por AASs (s, t) dentre os vértices limítrofes em VL
8      se CM(s) ≠ CM(t) então
9          SP_{s,t} ← todos os caminhos mais curtos entre s e t
10         se SP_{s,t} ≠ ∅ então
11             atualiza as estimativas
12             u ← t
13             enquanto u ≠ s faça
14                 sorteia z com probabilidade (σ_{s,z}/σ_{s,u})
15                 se z ≠ s então
16                     C̃Ip(v) ← C̃Ip(v) + 1/r
17                     u ← z
18                 fim
19             fim
20         fim
21     fim
22     i ← i + 1
23 fim
```

4.3 Planos amostrais

4.3.1 Plano amostral do algoritmo CIA

Deseja-se obter uma amostra aleatória de tamanho r, onde os resultados do estimador da centralidade de intermediação $\tilde{CIp}(v), \forall v \in Vert$ estão a uma margem de erro ε do verdadeiro resultado, $CIp(v), \forall v \in Vert$, com probabilidade mínima de $1 - \delta$. Riondato e Kornaropoulos (2016) conseguem este resultado, Teorema 2, utilizando a teoria da dimensão Vapnik-Chervonenkis (VC) (FRIEDMAN *et al.*, 2001), através da qual pode-se obter relação entre tamanho da amostra, sua complexidade e acurácia dos resultados. A dimensão VC de uma classe de funções é o maior número de pontos que podem ser separados pelos membros dessa classe (FRIEDMAN *et al.*, 2001).

Trazendo para o contexto deste trabalho, no algoritmo base CIA, se utiliza dimensão VC para cálculo do tamanho da amostra suficiente para obter uma estimação de alta qualidade. A dimensão VC é limitada por uma quantidade característica dos dados fácil de se calcular e fornece um número de exemplos suficiente para o aprendizado estatístico. Esse limite permite um algoritmo rápido para mineração de dados. A escolha da dimensão VC é justificada por dois motivos: porque fornece um tamanho de amostra que depende apenas de uma certa quantidade, sendo esta quantidade pequena e independente do número de objetos; e também porque os resultados e técnicas da dimensão VC podem ser utilizados para vários tipos de problemas sem necessitar de premissas com relação à distribuição da medida de interesse, nem com relação à existência de algum conhecimento prévio dos dados (RIONDATO, 2014).

Sejam:

- S_G: domínio, conjunto de todos os caminhos mais curtos do grafo G;
- A: amostra de r caminhos mínimos, subconjunto de S_G;
- T_v: o conjunto de todos os caminhos mais curtos onde o vértice v é um vértice dentro do caminho, ou seja, está no meio e não é um dos extremos;
- $p_{s,t}$: um caminho mínimo entre s e t;
- $VC(T_v)$: dimensão VC do conjunto T_v;
- $\pi(p_{s,t}) = \frac{1}{n(n-1)\sigma_{s,t}}$: distribuição de probabilidade em S_G, ou seja, probabilidade do caminho mais curto $p_{s,t}$ pertencer a amostra A, que é o produto da probabilidade de escolher o par de vértices s e t, vezes a probabilidade de se percorrer um dos caminhos mínimos entre s e t, conforme demonstrado a seguir.

Teorema 1 (Riondato e Kornaropoulos (2016) - Lema 5)**.** *Um caminho construído pelo processo do algoritmo CIA possui probabilidade de seleção dada por* $(1/\sigma_{s,t})$.

Prova(Riondato e Kornaropoulos (2016) - Lema 5).

Suponha $p* = \{s, z_1, z_2, ..., z_{|p*|-2}, t\}$ um caminho sorteado conforme algoritmo CIA. Inicialmente $p* = \{t\}$. Depois um vértice antecessor a t, $z_{|p*|-2}$, é sorteado com probabilidade $\sigma_{s,z_{|p*|-2}}/\sigma_{s,t}$. Posteriormente, outro antecessor é sorteado, $z_{|p*|-3}$, com probabilidade $\sigma_{s,z_{|p*|-3}}/\sigma_{s,z_{|p*|-2}}$, e assim sucessivamente. Logo:

$$P(p*) = \left(\frac{\sigma_{s,z_{|p*|-2}}}{\sigma_{s,t}}\right)\left(\frac{\sigma_{s,z_{|p*|-3}}}{\sigma_{s,z_{|p*|-2}}}\right)...\left(\frac{1}{\sigma_{s,z_2}}\right) = \frac{1}{\sigma_{s,t}}$$

, como queríamos demonstrar.

Os Teoremas e Lemas a seguir também possuem a prova realizada em Riondato e Kornaropoulos (2016):

Teorema 2 (Riondato e Kornaropoulos (2016) - Terorema 1 e Lema 7)**.** *Sejam* $VC(T_v)$, $\varepsilon > 0, \delta < 1$, *e* π *a distribuição de probabilidades em* S_G *conforme acima. Seja ainda* A *um subconjunto de* S_G *amostrados conforme* π *com tamanho mínimo definido abaixo:*

$$|A| \geq \frac{1}{2\varepsilon^2}\left[VC(T_v) + ln\frac{1}{\delta}\right] \tag{4.5}$$

Então, com probabilidade mínima de $1 - \delta$, *todas as estimativas do algoritmo CIA estão a uma distância* ε *do seu real valor:*

$$P(|\tilde{CI}p(v) - CIp(v)| > \varepsilon) < \delta \tag{4.6}$$

Teorema 3 (Riondato e Kornaropoulos (2016) - Corolário 1)**.** *A dimensão VC é limitada por:*

$$VC(T_v) \leq \lfloor log_2(VD(G) - 2)\rfloor + 1 \tag{4.7}$$

Lema 1 (Riondato e Kornaropoulos (2016))**.** *Se* r *caminhos mais curtos são amostrados conforme* π, *então com probabilidade de no mínimo* $(1 - \delta)$, $\tilde{CI}p(v)$ *estão a* ε *de* $CIp(v)$. *Sendo* r *o tamanho da amostra de caminhos mais curtos independente do tamanho do grafo, a fórmula para cálculo do tamanho da amostra* r *é obtida substituindo-se 4.7 em 4.5.*

$$r = \frac{1}{2\varepsilon^2}\left[\lfloor log_2\ (VD\,(G) - 2)\rfloor + 1 + ln(\frac{1}{\delta})\right] \tag{4.8}$$

O estimador não-viesado para o parâmetro centralidade de intermediação

padronizada para todos os vértices é dado por:

$$\tilde{CIp}(v) = \frac{1}{r} \sum_{s \neq v \in A} \sum_{t \neq v \in A} \sigma_{s,t}(v) \tag{4.9}$$

Logo, o número de vértices na amostra vai variar de acordo com os tamanhos dos caminhos mais curtos. Note que, para os vértices que não são internos aos caminhos mais curtos na amostra selecionada, a estimativa da centralidade de intermediação será zero.

Para fins ilustrativos, suponha $r = 3$ e o grafo da Figura 4.2:

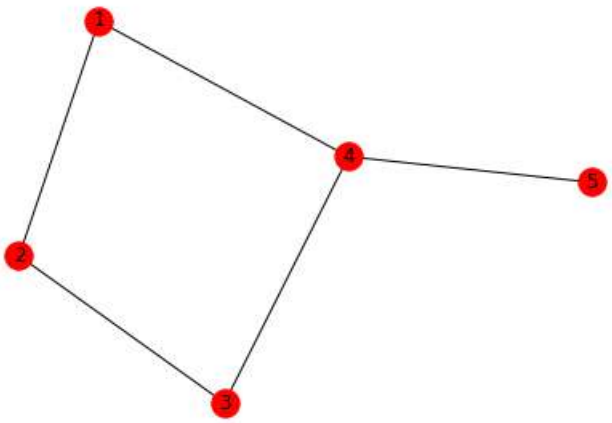

Figura 4.2: Exemplo ilustrativo do algoritmo CIA.

O algoritmo realiza os seguintes passos:

1. As estimativas de centralidade de intermediação padronizada iniciam com valor zero:

   $\tilde{CIp}(1) = 0; \tilde{CIp}(2) = 0; \tilde{CIp}(3) = 0; \tilde{CIp}(4) = 0; \tilde{CIp}(5) = 0.$

2. Suponha que no primeiro passo, $i = 1$, sendo $i = 1, 2, 3$, foram sorteados os vértices 2 e 5. Posteriormente foi sorteado um antecessor de 5, o vértice 4. Neste passo, $\tilde{CIp}(4) = 0 + 1/r = 0,3$. Novamente sorteia-se um vértice antecessor ao 4, o vértice 1. Então faz-se $\tilde{CIp}(1) = 0,3$. Como o próximo vértice antecessor é o início do caminho, $i \leftarrow i + 1$.

3. Neste segundo passo da iteração em i, onde $i = 2$, são sorteados os vértices 2 e 4. Posteriormente sorteia-se o antecessor 1, fazendo $\tilde{CIp}(1) = 0,3+0,3 = 0,6$.

4. Por fim, no último passo da iteração em i, onde $i = 3$, são sorteados os vértices 2 e 4 novamente. Posteriormente sorteia-se o antecessor 3, fazendo $\tilde{CIp}(3) = 0,3$.

Logo, as estimativas finais de $CIp(v)$ são:
$\tilde{CIp}(1) = 0,6; \tilde{CIp}(2) = 0; \tilde{CIp}(3) = 0,3; \tilde{CIp}(4) = 0,3; \tilde{CIp}(5) = 0.$

O plano amostral do algoritmo baseado em amostragem de Riondato e Kornaropoulos (2016) utiliza a seguinte estratégia em dois estágios:

- 1o estágio:

  Sorteia um par de vértices s e t por AASs de $Vert$, que é o conjunto de todos os vértices da rede. Logo, as UPAS são os pares de vértices sorteados.

- 2o estágio:

  Calcula todos os caminhos mínimos entre s e t e faz o caminho de volta de t para s sorteando o vértice antecessor z com probabilidade proporcional ao tamanho, sendo a variável de tamanho dada pela quantidade de caminhos mais curtos que unem s e z, ou seja, $\sigma_{s,z}/\sigma_{s,t}$. Este procedimento é repetido r vezes, onde r é a quantidade de caminhos mais curtos necessária para que o estimador tenha as garantias de qualidade teóricas apresentadas anteriormente. Ou seja, as USAs são os caminhos mínimos sorteados no segundo estágio. A estimativa da centralidade de intermediação será $1/r$ multiplicado pelo número de vezes em que o vértice estava nos caminhos mais curtos amostrados.

Para avaliar as propriedades do estimador 4.9, tem-se os conceitos a seguir. Suponha que se deseja estimar um parâmetro θ usando o seu estimador $\hat{\theta}$.

- Estimador não-viesado: são estimadores onde o valor esperado do estimador é o parâmetro a ser estimado, ou seja, $E(\hat{\theta}) = \theta$.

- Estimador consistente: são estimadores cuja variância tende a zero, ou seja, $V(\hat{\theta}) \to 0$ quando o tamanho da amostra tende a infinito ($r \to \infty$).

Lema 2. *$\tilde{CIp}(v)$ é estimador não-viesado para o parâmetro $CIp(v)$.*

Prova.

$$E[\tilde{CIp}(v)] = E[\frac{1}{r}\sum_{s\neq v\in A}\sum_{t\neq v\in A}\sigma_{s,t}(v)]$$

$$= \frac{1}{r}E[\sum_{s\neq v\in A}\sum_{t\neq v\in A}\sigma_{s,t}(v)]$$

$$= \frac{1}{r}\sum_{s\neq v\in A}\sum_{t\neq v\in A}E[\sigma_{s,t}(v)]$$

Como o teorema 1 diz que a probabilidade de um caminho mínimo construído pelo algoritmo CIA ser selecionado é $1/\sigma_{s,t}$, temos:

$$= \left(\frac{1}{r}\right) r \sum_{s \neq v \in Vert} \sum_{t \neq v \in Vert} \sigma_{s,t}(v)\pi(p_{s,t})$$

$$= \sum_{s \neq v \in Vert} \sum_{t \neq v \in Vert} \sigma_{s,t}(v) \frac{1}{n(n-1)\sigma_{s,t}} = CIp(v),$$

como queríamos demonstrar.

Lema 3. *$\tilde{CIp}(v)$ é estimador consistente para o parâmetro $CIp(v)$.*

Prova.

$$V[\tilde{CIp}(v)] = V\left[\frac{1}{r} \sum_{s \neq v \in A} \sum_{t \neq v \in A} \sigma_{s,t}(v)\right]$$

$$= \frac{1}{r^2} V[\sum_{s \neq v \in A} \sum_{t \neq v \in A} \sigma_{s,t}(v)],$$

que tende a zero quando $r \to \infty$.

4.3.2 Plano amostral do algoritmo proposto CIAVLCM

O algoritmo proposto foi elaborado com experimentos sobre o sorteio do par de vértices, com o objetivo de melhorar a acurácia e precisão das estimativas do parâmetro de interesse centralidade de intermediação para todos os vértices.

O plano amostral do algoritmo proposto por esta tese utiliza a seguinte estratégia em dois estágios:

- 1o estágio:

  Esta fase divide os vértices em dois grupos. O primeiro grupo é formado por vértices que são limítrofes entre as comunidades encontradas por um determinado método de detecção de comunidades e ao mesmo tempo pertencem a comunidades diferentes. Esta estratégia foi escolhida pois estes vértices possuem maior centralidade de intermediação (SUPPA e ZIMEO, 2015). Como a centralidade de intermediação é utilizada para identificação de vértices mais influentes, o interesse maior está nos vértices com maior valor desta centralidade. Podemos chamar este grupo de $Vert_1$. Sorteia-se um par de vértices s e t por AASs de $Vert_1$, um subconjunto de $Vert$, conforme descrito acima. Observe que ao realizar os sorteios em $Vert_1$, os demais vértices não são excluídos da amostra, pois os vértices sorteados s e t serão respectivamente os vértices de origem e destino do caminho selecionado, e os vértices da amostra são os vértices internos a estes caminhos. Logo, as UPAS são os pares de vértices sorteados em $Vert_1$.

- 2o estágio:

  É igual ao segundo estágio do plano amostral do algoritmo CIA. Calcula todos os caminhos mínimos entre s e t e faz o caminho de volta de t para s sorteando o vértice antecessor z com probabilidade proporcional ao tamanho, sendo a variável de tamanho dada pela quantidade de caminhos mais curtos que unem s e z, ou seja, $\sigma_{s,z}/\sigma_{s,t}$. Este procedimento é repetido r vezes, onde r é a quantidade de caminhos mais curtos necessária para que o estimador tenha as garantias de qualidade teóricas apresentadas na seção anterior. Ou seja, as USAs são os caminhos mínimos sorteados no segundo estágio ou os vértices internos aos caminhos mínimos. A estimativa da centralidade de intermediação será $1/r$ multiplicado pelo número de vezes em que o vértice estava nos caminhos mais curtos amostrados.

Esta proposta altera a probabilidade $\pi(p_{s,t}) = \frac{1}{n(n-1)\sigma_{s,t}}$, probabilidade do caminho mais curto $p_{s,t}$ pertencer a amostra A. A nova probabilidade é dada por:

$$\pi_1(p_{s,t}) = \frac{1}{|Vert_1|(|Vert_1| - 1)\sigma_{s,t}} \tag{4.10}$$

O próximo passo a ser realizado em sequencia desta tese será provar teoricamente que o plano amostral proposto satisfaz as mesmas condições do plano amostral anterior, colocando o algoritmo proposto entre o algoritmo de Riondato e Kornaropoulos (2016), o melhor com garantias teóricas, e o de Geisberger, Sanders, e Schultes (2008), o melhor sem garantias teóricas.

4.4 Implementação

O software livre Python foi escolhido porque, além de ser gratuito, apresenta outras vantagens:

- está sempre sendo atualizado pelos usuários;
- possui vasta literatura disponível;
- possui muitos códigos disponibilizados no GitHub (uma plataforma de hospedagem de código-fonte com controle de versão);
- tem uma comunidade de usuários muito ativa no Stack Overflow (sítio que apresenta perguntas e respostas em uma grande quantidade de tópicos de programação de computadores).

Os códigos gerados para esta tese foram elaborados utilizando intensamente as fontes de informação acima, principalmente o GitHub e Stack Overflow.

Na Figura 4.3 tem-se a utilização de códigos abertos em aprendizado de máquina. Observe que o aumento proporcional de usuários em Python foi maior que para a linguagem R.

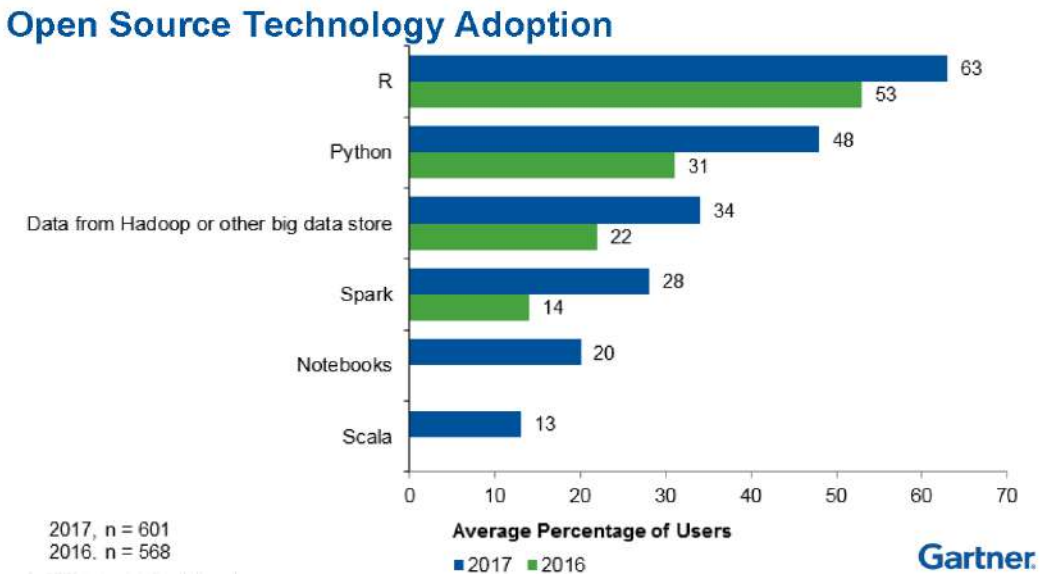

Figura 4.3: Códigos abertos utilizados em 2016 e em 2017.

Os algoritmos foram processados em Python 3.6.4, num servidor de alto desempenho de 20 núcleos (2 processadores Intel Xeon E5-2670v2 de 10 Núcleos cada) e 256GB de memória RAM com processamento paralelo via biblioteca *"multiprocessing"* do Python.

Para avaliar escalabilidade foram criadas redes complexas artificiais de diferentes tamanhos (1 mil a 1 milhão de vértices) usando o modelo Barabási-Albert (BARABÁSI e ALBERT, 1999) implementado por NetworkX(HAGBERG *et al.*, 2018). As redes complexas artificiais deste estudo foram geradas de forma a possuir características de várias redes reais como as redes sociais por exemplo. Elas possuem em comum as seguintes características: são esparsas, possuem a propriedade de mundo pequeno, seguem lei de potência e são modulares.

As bibliotecas para se trabalhar com grafos e redes complexas avaliadas foram:

- **NetworkX**[5]: foi utilizada para gerar as redes artificiais e para análise estrutural das mesmas. Este pacote rodou nos 3 ambientes testados: *notebook*, *cluster* e plataforma Spark na nuvem da Databricks. Os códigos fonte dos pacotes estão disponíveis no GitHub em Python.

- **NetworKit**[6]: foi utilizada para detecção das comunidades, para encontrar a maior componente conexa, para estimar o diâmetro e para análise estrutural

[5]https://networkx.github.io/documentation/stable/tutorial.html
[6]https://networkit.iti.kit.edu/api/modules.html

das redes. Só funcionou no *cluster* com Linux. Os códigos fonte dos pacotes estão disponíveis no GitHub em C++. O algoritmo de Riondato e Kornaropoulos (2016) está implementado neste pacote, o que tornou possível a conferência dos resultados da minha implementação em Python para obtenção dos tempos de processamento dos algoritmos com os mesmos tipos de otimização.

- **Igraph**[7] e **SNAP**[8]: houve dificuldade para instalação destes pacotes para Python com sistema operacional Windows.
- **GraphX**[9]: não foi utlizada pois não é em Python, e sim, em Scala.
- **GraphFrames**[10]: quando testada, esta biblioteca se apresentou instável, gerando problemas na execução.

Para implementação computacional foram testadas duas opções:

1. usando computação paralela com a biblioteca *"multiprocessing"* do Python rodando em um *cluster*;
2. usando plataforma de processamento distribuído com tolerância a falhas Spark sobre HDFS (Hadoop File Systems)[11].

Os resultados apresentados no próximo capítulo são referentes ao processamento usando a biblioteca *"multiprocessing"* do Python no *cluster* descrito anteriormente. Porém os resultados de tempo utilizando Spark no modo autônomo, ou seja, sem ser distribuído de fato, reduziu o tempo de processamento pela metade. Uma opção para trabalho futuro é rodar os algoritmos utilizando uma nuvem. A Databricks possui uma versão grátis, a *community edition*[12], onde são disponibilizados 6GB de memória e roda Spark sobre HDFS.

Foram utilizados os pacotes para linguagem Python: NetworkX(HAGBERG *et al.*, 2018) e NetworKit(STAUDT *et al.*, 2016). A implementação foi realizada para grafos não direcionados, não ponderados, sem laços ou múltiplas arestas. Para grafos não conexos foi utilizada a maior componente conexa.

[7] http://igraph.org/python/
[8] http://snap.stanford.edu/snappy/index.html
[9] https://spark.apache.org/docs/latest/graphx-programming-guide.html
[10] https://graphframes.github.io/user-guide.html
[11] Plataforma de armazenamento distribuído voltada para *clusters* e processamento de grandes volumes de dados, com tolerância a falhas.
[12] https://databricks.com/try-databricks

Capítulo 5

Resultados e discussão

Na Tabela 5.1 tem-se algumas características topológicas das redes artificiais desta tese. Estas redes não são ponderadas nem direcionadas. Também não possuem laços nem múltiplas arestas, uma vez que múltiplas arestas podem facilmente ser transformadas em arestas ponderadas, o que é feito na maioria dos estudos.

Tabela 5.1: Algumas características topológicas das redes artificiais desta tese: número de vértices, número de arestas, grau médio, densidade e diâmetro.

Número de vértices (n)	Número de arestas (m)	Grau médio	Densidade	Diâmetro
1.000	49.705	99,41	0,09951	3
10.000	500.767	100,15	0,01002	3
100.000	5.000.831	100,02	0,00100	3
500.000	12.504.957	50,02	0,00010	4*
1.000.000	4.995.470	9,99	0,00001	7*

*Diâmetro dos dois últimos grafos são estimados.

Estas redes artificiais foram geradas de tal forma que possuíssem as seguinte propriedades de redes complexas reais:

- Mundo pequeno;
- Sem escala e
- Esparsas.

O fato das redes serem esparsas pode ser percebido pela baixa densidade das mesmas, resultados que podem ser verificados na Tabela 5.1. A propriedade mundo pequeno pode ser percebida pelo diâmetro das redes, também na Tabela 5.1, onde o maior valor é 7 para rede de 1 milhão de vértices. E é sem escala pois foi gerada como rede Barabási-Albert(BARABÁSI e ALBERT, 1999).

Os experimentos realizados para análise dos algoritmos têm como objetivo avaliar: acurácia, precisão, tempo de processamento em comparação com o algoritmo tomado como base e escalabilidade em função do tamanho da rede complexa como quantidade de vértices.

Os parâmetros para cálculo do tamanho da amostra r foram $\varepsilon = 0,05$ e $\delta = 0,10$ para permitir a comparação dos resultados do algoritmo proposto com os resultados do algoritmo CIA.

Foi realizado um teste de hipóteses para verificar se os valores de EQM dos algoritmos propostos são diferentes do algoritmo CIA com nível de significância de 5%, e todos os resultados foram significantes, ou seja, obtiveram $p - valor < 0,05$. Neste teste de hipóteses, a hipótese nula é de que o EQM do método alternativo CIVLCM é igual ao EQM do método base CIA. A estatística de teste utilizada foi o $p - valor$.

Nas tabelas a seguir estão os resultados dos indicadores escolhidos para a avaliação da qualidade das estimativas:

- Erro quadrático médio (EQM): utilizado para medir a acurácia da estimativa. Foram realizadas as simulações e, para o resultado de cada simulação, o erro quadrático foi calculado como sendo o quadrado da diferença entre o valor da estimativa e o valor exato da centralidade de intermediação padronizada calculado pelo algoritmo de Brandes (2001). Posteriormente foi calculada a média desses resultados. Suponha que tenham sido realizadas J simulações, então:

$$\text{EQM} = \frac{1}{n} \sum_{v \in Vert} \sum_{j=1}^{J} \frac{[CIp(v) - \tilde{CIp}(v)]^2}{J} \tag{5.1}$$

- Coeficiente de variação médio (CV): para medir a precisão da estimativa.

- Efeito do plano amostral (EPA): para comparar a precisão de dois desenhos amostrais. Neste caso, é dado pela variância da estimativa segundo o plano amostral proposto, que leva em consideração as comunidades às quais pertencem os vértices, e o desenho amostral do algoritmo CIA (V/V_{CIA}).

- Tempo do algoritmo proposto sobre tempo do algoritmo CIA (T/T_{CIA}).

Tabela 5.2: Resultados de 100 simulações para uma rede não ponderada, não direcionada, com 1.000 vértices e 49.705 arestas.

Algoritmo	EQM/EQM$_{CIA}$	CV/CV$_{CIA}$	T/T$_{CIA}$	EPA	p-valor EQM
CIAVL	**0,799**	**0,816**	0,624	0,821	0,0000
CIAVLCM	**0,813**	**0,881**	**0,617**	**0,765**	0,0000
CIACM	0,830	0,921	**0,597**	**0,749**	0,0000

Tabela 5.3: Resultados de 100 simulações para uma rede não ponderada, não direcionada, com 10.000 vértices e 500.767 arestas.

Algoritmo	EQM/EQM$_{CIA}$	CV/CV$_{CIA}$	T/T$_{CIA}$	p-valor EQM
CIAVL	0,690	**0,787**	**0,623**	0,0000
CIAVLCM	**0,588**	**0,836**	0,695	0,0000
CIACM	**0,635**	0,899	**0,589**	0,0000

Tabela 5.4: Resultados de 20 simulações para uma rede não ponderada, não direcionada, com 100.000 vértices e 5.000.831 arestas.

Algoritmo	EQM/EQM$_{CIA}$	CV/CV$_{CIA}$	T/T$_{CIA}$	p-valor EQM
CIAVL	0,639	**0,423**	0,903	0,0000
CIAVLCM	**0,444**	**0,486**	**0,700**	0,0000
CIACM	**0,444**	0,576	**0,696**	0,0000

Com base nos resultados contidos nas Tabelas 5.2, 5.3 e 5.4 o algoritmo escolhido foi o que apresentou melhora significante de acurácia e precisão, além de reduzir o tempo de processamento foi o CIAVLCM, ou seja, baseado em amostragem, levando em consideração a comunidade a qual pertence o vértice e se ele é um vértice que une comunidades. Note que para avaliar precisão foi utilizado o indicador CV/CV$_{CIA}$ e não o EPA sugerido anteriormente. Este fato ocorreu em virtude das redes com mais de mil vértices apresentarem muitos zeros na variância, logo o indicador EPA ficou ineficiente para comparar precisão pois EPA $= V/V_{CIA}$, e o zero no denominador não permite avaliação desta medida.

Os tamanhos de amostras de caminhos mais curtos r se encontram na Tabela 5.5 abaixo, note que ele depende somente da medida $VD(G)$, que é o número de vértices

no caminho do diâmetro do grafo, ou seja, o maior caminho mais curto do grafo. Como os grafos deste estudo são não ponderados, o $VD(G) = diam(G) + 1$. Esta medida é ponto crucial para escalabilidade do algoritmo, pois, graças a propriedade mundo pequeno, os diâmetros das redes complexas tendem a ser valores baixos.

Tabela 5.5: Tamanho das amostras utilizadas nas simulações por tamanho da rede em número de vértices.

Número de vértices (n)	VD(G)	Tamanho das amostras (r)
1.000	4	861
10.000	4	861
100.000	4	861
500.000	5	861
1.000.000	8	1061

Embora a rede com 500 mil vértices não possua o mesmo VD(G) das redes anteriores, o tamanho da amostra foi igual pois a fórmula de r utiliza valores truncados.

O algoritmo proposto melhorou a qualidade das estimativas das centralidades de intermediação obtidas em redes complexas grandes, porém não traz ganhos em redes pequenas. Tome como exemplo um grafo com 100 vértices e diâmetro 4, que resultaria numa amostra de tamanho 861.

Para as redes com 500 mil e 1 milhão de vértices, não foi possível calcular o valor exato da centralidade de intermediação e consequentemente, o erro quadrático médio. Para solucionar este problema, não ter os valores exatos, pode-se usar o método da replicação conforme sugerido por Pessoa e Silva (1998). A ideia é construir uma amostra de como a união de amostras selecionadas de forma independente e usando o mesmo plano amostral nr (número de réplicas) vezes, e usar a média dos estimadores das replicações como estimador do parâmetro de interesse.

Tabela 5.6: Resultados de 2 simulações para uma rede não ponderada, não direcionada, com 500.000 vértices e 12.504.957 arestas e de 1 simulação para uma rede não ponderada, não direcionada, com 1.000.000 vértices e 4.995.470 arestas.

Vértices	T/T_{CIA}
500 mil	0,87
1 milhão	0,92

Tabela 5.7: Resultados de 100 simulações para as redes de 1 mil e de 10 mil vértices; e 20 simulações para uma rede de 100 mil vértices. Estes resultados são para os top-100 vértices com maior centralidade de intermediação.

Vértices	EQM/EQM$_{CIA}$	CV/CV$_{CIA}$
1 mil	1,00	1,02
10 mil	0,52	1,13
100 mil	**0,31**	**0,56**

O tempo e escalabilidade na Figura 5.1 mostram que o algoritmo CIAVLCM é mais rápido mantendo a escalabilidade.

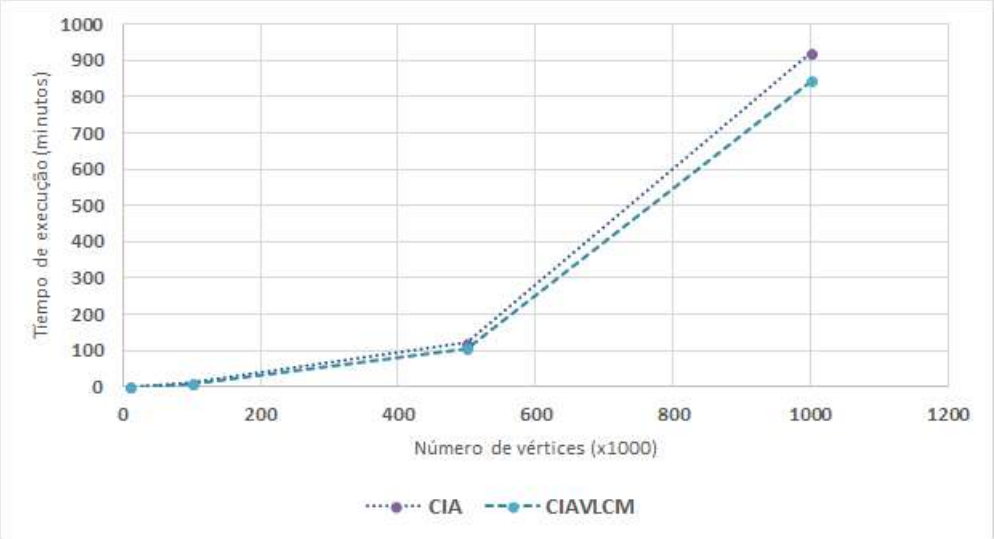

Figura 5.1: Tempo e escalabilidade.

Também foi feita a verificação sobre o Teorema 2, que afirma que uma amostra de caminhos mais curtos de tamanho r fornece estimativas de centralidade com a propriedade abaixo, sendo $\varepsilon = 0,05$ e $\delta = 0,10$:

$$P(|\tilde{CI}p(v) - CIp(v)| > \varepsilon) < \delta \tag{5.2}$$

Para tal verificação, foi feita a contagem de quantas vezes a diferença absoluta entre parâmetro e estimativa ficaram acima de ε. Para todas as redes e todas as simulações, todos os erros absolutos $|CIp(v) - \tilde{CI}p(v)|$ ficaram abaixo de ε.

Resultados esperados e atingidos:

- Ao aplicar os algoritmos nas redes artificiais esperava-se obter estimativas com CV menores que os originais assim como EPA menor que 1, ao comparar os algoritmos. Como não foi possível utilizar o EPA, essa verificação foi realizada com o CV.

- O resultado também melhorou o tempo e as estimativas para os top-100 vértices. Este é um resultado importante pois o objetivo principal do cálculo das medidas de centralidade de intermediação é identificar os vértices mais importantes segundo esta medida.

- Para as redes com 500 mil e 1 milhão de vértices foram rodadas 2 simulações que obtiveram tempos menores de processamento quando comparados ao algoritmo base, 0,87 e 0,92, respectivamente e conforme Tabela 5.6.

Capítulo 6

Considerações finais

Neste trabalho foram estudados planos amostrais aplicados a grandes redes complexas. O campo de estudo ainda permite muitas pesquisas uma vez que os algoritmos com melhores acurácia e precisão não possuem garantias teóricas.

O que levou o resultado ao ganho tanto de tempo quanto de simplicidade no processamento foi utilizar algoritmos paralelizáveis e de natureza aglomerativa como:

- o método de Louvain para detecção de comunidades (BLONDEL *et al.*, 2008);
- a ideia de contribuições parciais do algoritmo de Brandes (2008) com busca em largura e
- o algoritmo de Riondato e Kornaropoulos (2016).

Com base nos resultados obtidos no capítulo anterior, pode-se concluir que os algoritmo baseado em amostragem e em detecção de comunidades CIAVLCM reduziu o erro quadrático médio, logo, melhorou acurácia das estimativas das centralidades de intermediação em comparação com algoritmo CIA, assim como reduziu consideravelmente o tempo de processamento, o que pode ser verificado nas Tabelas 5.2, 5.3 e 5.4.

Ao avaliar a performance do algoritmo proposto nos 100 vértices com maior centralidade de intermediação, uma vez que esta é uma medida para se identificar quais os vértices mais influentes, os resultados foram melhores. Pode-se avaliar este resultado ao comparar a redução no EQM e no CV, como por exemplo para o grafo com 100 mil vértices onde, considerando toda a rede, o algoritmo proposto apresentou EQM/EQM$_{CIA}$ de 0,44 e CV/CV$_{CIA}$ de 0,49 conforme Tabela 5.4; e considerando os top-100 vértices com maior centralidade de intermediação, este indicadores foram 0,31 e 0,56 respectivamente conforme Tabela 5.7.

O algoritmo proposto melhorou a qualidade das estimativas das centralidades de intermediação obtidas em redes complexas grandes, porém, não traz ganhos significativos em redes pequenas.

Como sequência desta tese, será iniciado um projeto de análise em tempo real com dados em *"stream"* da rede social *online* Twitter. Os objetivos deste projeto são: classificar e monitorar mensagens danosas para a imagem da instituição IBGE, posicionamento do IBGE nas redes sociais modeladas em grafo para extração da estrutura onde o IBGE está inserido, e possíveis estudos demográficos sobre migração e movimentos pendulares usando dados georreferenciados, como o realizado pelo INEGI (MUNOZ, 2015) no contexto do grupo de trabalho sobre uso de *big data* para estatísticas oficiais da comissão estatística das Nações Unidas. No projeto de análise em tempo real, a amostragem será utilizada para dar agilidade a todos os modelos do projeto.

Seria interessante continuar este estudo com valores menores de ε e utilizando o Spark. O tempo de processamento dos algoritmos para uma rede com mil vértices caiu pela metade ao se utilizar o Spark no modo *standalone* ao invés da biblioteca *"multiprocessing"* do Python. O código implementado para a plataforma Spark se encontra no Apêndice.

Outros desdobramentos possíveis deste trabalho são:

- Análise de técnicas de amostragem para grafos dinâmicos, uma vez que muitas redes complexas reais são dinâmicas, como por exemplo as redes sociais.

- Amostragem para utilizar em fluxo contínuo de dados (*data stream*), pois avaliar todos os dados e manter os modelos, sejam de previsão ou de classificação ou outro, é complicado. Atualizar os modelos com dados entrantes bem amostrados pode garantir eficiência em qualquer que seja a fase do processo de mineração de dados.

- Rodar mais simulações utilizando Spark sobre HDFS. Uma versão do algoritmo CIA utilizando o Spark reduziu o tempo de processamento pela metade, mesmo tendo como otimizar mais. O código encontra-se no apêndice.

- Elaboração de pacotes ou funções ou painéis de controle (*dashboards*) para estudos posteriores utilizando análise de redes complexas utilizando amostragem com resultados em tempo real ou *nowcasting*. Um trabalho similar foi desenvolvido pelo INE do México (INEGI) para análise de sentimento utilizando dados do Twitter.

- Análise téorica do algoritmo proposto para validar que o novo método de seleção da amostra de caminhos mais curtos melhora acurácia e precisão. O que deixaria este algoritmo como o melhor algoritmo com garantias teóricas. Ou seja, a evolução temporal dos algoritmos para estimar centralidade de intermediação seria:

1. Exato de Brandes (2001).
2. Estimado de Brandes e Pich (2007), que seleciona vértices, apresenta estimador não viesado. Brandes e Pich (2007) sugerem PPT utilizando o grau do vértice como variável de tamanho.
3. Estimado de Geisberger, Sanders, e Schultes (2008), adaptam a parte de busca do algoritmo de Brandes e Pich (2007) conseguindo estimativas com melhor acurácia nos vértices com maior grau.
4. Estimado de Riondato e Kornaropoulos (2016), algoritmo CIA.
5. Estimado pelo algoritmo CIAVLCM.

- Testar o algoritmo novo CIAVLCM em redes ponderadas, direcionadas, dinâmicas, em redes reais, variar ε e δ e avaliar os conjuntos de top-k%.

Referências Bibliográficas

AHMED, N. K., DUFFIELD, N., NEVILLE, J., et al., 2014a, "Graph sample and hold: A framework for big-graph analytics". In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1446–1455. ACM, a.

AHMED, N. K., NEVILLE, J., KOMPELLA, R., 2014b, "Network sampling: From static to streaming graphs", *ACM Transactions on Knowledge Discovery from Data (TKDD)*, v. 8, n. 2, pp. 7.

ALBERT, R., BARABÁSI, A.-L., 2002, "Statistical mechanics of complex networks", *Reviews of modern physics*, v. 74, n. 1, pp. 47.

BARABÁSI, A.-L., 2014, *Network science book*, v. 625. Disponível em: <`http://networksciencebook.com/`>.

BARABÁSI, A.-L., ALBERT, R., 1999, "Emergence of scaling in random networks", *Science*, v. 286, n. 5439, pp. 509–512.

BERGAMINI, E., MEYERHENKE, H., 2016, "Approximating Betweenness Centrality in Fully Dynamic Networks", *Internet Mathematics*, v. 12, n. 5.

BERGAMINI, E., CRESCENZI, P., D'ANGELO, G., et al., 2017, "Improving the betweenness centrality of a node by adding links", *arXiv preprint arXiv:1702.05284*.

BLONDEL, V. D., GUILLAUME, J.-L., LAMBIOTTE, R., et al., 2008, "Fast unfolding of communities in large networks", *Journal of statistical mechanics: theory and experiment*, , n. 10.

BOAS, P. R. V., 2008, *Efeito da amostragem nas propriedades topológicas de redes complexas*. Tese de Doutorado, Universidade de São Paulo.

BOLFARINE, H., BUSSAB, W. O., 2004, *Elementos da Amostragem*. São Paulo, Universidade de São Paulo, Instituto de Matemática.

BRANDES, U., 2008, "On variants of shortest-path betweenness centrality and their generic computation", *Soc. Netw.*, v. 30, n. 2, pp. 136–145.

BRANDES, U., 2001, "A faster algorithm for betweenness centrality", *Journal of mathematical sociology*, v. 25, n. 2, pp. 163–177.

BRANDES, U., PICH, C., 2007, "Centrality estimation in large networks", *International Journal of Bifurcation and Chaos*, v. 17, n. 7, pp. 2303–2318.

CLAUSET, A., NEWMAN, M. E., MOORE, C., 2004, "Finding community structure in very large networks", *Physical review E*, v. 70, n. 6.

COHEN, E., DUFFIELD, N., KAPLAN, H., et al., 2011, "Efficient stream sampling for variance-optimal estimation of subset sums", *SIAM Journal on Computing*, v. 40, n. 5, pp. 1402–1431.

CORMODE, G., DUFFIELD, N., 2014, "Sampling for big data: a tutorial". In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1975–1975. ACM.

DEL-VECCHIO, R. R., GALVÃO, D. J. C., LIMA, L. S., et al., 2009, "Medidas de Centralidade da Teoria dos Grafos aplicada a Fundos de Ações no Brasil", *XLI Simpósio Brasileiro de Pesquisa Operacional*.

DIJKSTRA, E. W., 1959, "A note on two problems in connexion with graphs", *Numerische mathematik*, v. 1, n. 1, pp. 269–271.

DRYVER, A. L., THOMPSON, S. K., 2005, "Improved unbiased estimators in adaptive cluster sampling", *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, v. 67, n. 1, pp. 157–166.

ERDÖS, P., RÉNYI, A., 1959, "On random graphs I", *Publicationes Mathematicae*, v. 6, pp. 290–297.

ERDÖS, P., RÉNYI, A., 1960, "On the evolution of random graphs", *Publications of Mathematical Institute of the Hungarian Academy of Science*, v. 5, pp. 17–61.

FIGUEIREDO, D. R., 2011, "Introdução a redes complexas". pp. 303–358.

FRANK, O., SNIJDERS, T., 1994, "Estimating the size of hidden populations using snowball sampling", *Journal of Official Statistics*, v. 10, n. 1, pp. 53.

FREEMAN, L. C., 1977, "A set of measures of centrality based on betweenness", *Sociometry*, v. 40, pp. 35–41.

FRIEDMAN, J., HASTIE, T., TIBSHIRANI, R., 2001, *The elements of statistical learning - Data Mining,Inference,and Prediction*, v. 1. Springer series in statistics New York.

GEISBERGER, R., SANDERS, P., SCHULTES, D., 2008, "Better approximation of betweenness centrality". In: *Proceedings of the Meeting on Algorithm Engineering & Expermiments*, pp. 90–100. Society for Industrial and Applied Mathematics.

GIRVAN, M., NEWMAN, M. E., 2002, "Community structure in social and biological networks", *Proceedings of the national academy of sciences*, v. 99, n. 12, pp. 7821–7826.

GJOKA, M., KURANT, M., BUTTS, C. T., et al., 2010, "Walking in facebook: A case study of unbiased sampling of OSNS". In: *Infocom, 2010 Proceedings IEEE*, pp. 1–9. IEEE.

HAGBERG, A., SCHULT, D., SWART, P., 2018. "NetworkX Reference". .

IBGE, 2014a, *Redes e fluxos de Território: Gestão do Território 2014*. Relatório técnico, Instituto Brasileiro de Geografia e Estatística, Rio de Janeiro, a.

IBGE, 2013, *Redes e fluxos dos territórios: Ligações aéreas 2010*. Relatório técnico, Instituto Brasileiro de Geografia e Estatística, Rio de Janeiro.

IBGE, 2016, *Redes e fluxos dos territórios: Logística de energia 2015*. Relatório técnico, Instituto Brasileiro de Geografia e Estatística, Rio de Janeiro.

IBGE, 2014b, *Pesquisa Nacional por Amostra de Domicílios Contínua: Notas Metodológicas*. Relatório técnico, Instituto Brasileiro de Geografia e Estatística, Rio de Janeiro, b.

JACOB, R., KOSCHÜTZKI, D., LEHMANN, K., et al., 2005, "Algorithms for Centrality Indices". In: *Network Analysis*, v. 3418, *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 62–82.

LINDEN, G., SMITH, B., YORK, J., 2003, "Amazon. com recommendations: Item-to-item collaborative filtering", *IEEE Internet computing*, v. 7, n. 1, pp. 76–80.

MILGRAM, S., 1967, "The small world problem", *Phychology Today*, v. 1, n. 1, pp. 61–67.

MUNOZ, J., 2015, "Use of data from social networks to obtain statistical and geographical information". In: *Global Conference on Big Data for Official Statistics*. Instituto de Estatística e de Geografia do México - INEGI.

NEWMAN, M. E. J., 2010, *Networks – An Introduction*. Oxford University Press.

NEWMAN, M. E. J., 2003, "The structure and function of complex networks", *SIAM Review*, v. 45, n. 2, pp. 167–256.

NEWMAN, M. E. J., GIRVAN, M., 2004, "Finding and evaluating community structure in networks", *Phys. Rev. E*, v. 69.

PAGE, L., BRIN, S., MOTWANI, R., et al., 1999, *The PageRank citation ranking: Bringing order to the web.* Relatório técnico, Stanford InfoLab.

PALMER, C. R., GIBBONS, P. B., FALOUTSOS, C., 2002, "ANF: A fast and scalable tool for data mining in massive graphs". In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 81–90. ACM.

PASTOR-SATORRAS, R., VESPIGNANI, A., 2001, "Epidemic spreading in scale-free networks", *Physical review letters*, v. 86, n. 14, pp. 3200.

PESSOA, D. G. C., SILVA, P. L. N., 1998, *Análise de Dados Amostrais Complexos*. São Paulo, Associação Brasileira de Estatística.

RIONDATO, M., 2014, *Sampling-based Randomized Algorithms for Big Data Analytics*. Tese de D.Sc., Brown University, Rhode Island, EUA.

RIONDATO, M., KORNAROPOULOS, E. M., 2016, "Fast approximation of betweenness centrality through sampling", *Data Mining and Knowledge Discovery*, v. 30, n. 2, pp. 438–475.

RIONDATO, M., UPFAL, E., 2015, "Mining frequent itemsets through progressive sampling with rademacher averages". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1005–1014. ACM.

RIONDATO, M., UPFAL, E., 2016, "ABRA: Approximating Betweenness Centrality in Static and Dynamic Graphs with Rademacher Averages". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1145–1154. ACM.

SAMATOVA, N. F., HENDRIX, W., JENKINS, J., et al., 2013, *Practical graph mining with R*. Minnesota, CRC Press.

SILVA, P. L. D. N., SANTOS, E. G. D., COELHO, I. B., et al., 2014, "The challenges of producing statistics for the Web: sampling and automated data collection of webpage information in the Brazilian Web", *Statistics Canada Symposium*.

SILVA, T. S. A., 2010, "Um estudo de medidas de centralidade e confibilidade em redes", *CEFET/RJ*.

SOUZA, R. C. D., EBECKEN, N. F. F., 2018, "Sampling for large data volumes: an application on complex networks", *IEEE América Latina - submetido e aguardando revisão*.

STAUDT, C. L., SAZONOVS, A., MEYERHENKE, H., 2016, "NetworKit: A tool suite for large-scale complex network analysis", *Network Science*, v. 4, n. 4, pp. 508–530.

SUPPA, P., ZIMEO, E., 2015, "A clustered approach for fast computation of betweenness centrality in social networks". In: *IEEE International Congress on Big Data*, pp. 47–54. IEEE.

TAN, P.-N., STEINBACH, M., KUMAR, V., 2009, *Introdução ao Data Mining - Mineração de Dados*. Rio de Janeiro, Editora Ciência Moderna.

THOMPSON, S., 2012, *Sampling*. Wiley.

THOMPSON, S., SEBER, G., 1995, "Adaptive sampling". In: *Proceedings of the section on survey research methods of the American Statistical Association*, pp. 784–786.

THOMPSON, S. K., 1998, "Adaptive sampling in graphs". In: *Proceedings of the Section on Survey Methods Research, American Statistical Association*, pp. 13–22.

THOMPSON, S. K., 2006, "Targeted random walk designs", *Survey Methodology*, v. 32, n. 1, pp. 11.

TILLÉ, Y., 2006, *Sampling algorithms*. Springer.

WATTS, D. J., STROGATZ, S. H., 1998, "Collective dynamics of 'small-world' networks", *Nature*, v. 393, n. 6684, pp. 440.

ZHANG, L.-C., PATONE, M., 2017, "Graph sampling", *Metron*, v. 75, n. 3, pp. 277–299.

Apêndice A

Códigos dos algoritmos implementados em Python

A.1 CIA e CIAVLCM

```
# importa pacotes
import networkit as nk
import networkx as nx
import time
import pandas as pd
import random
from numpy.random import choice
import math as m
from heapq import heappush, heappop
from itertools import count
import multiprocessing
from multiprocessing import Pool

# função que calcula SSSP do vértice s para todos os outros via busca em largura
def _single_source_shortest_path_basic(G, s):
    P = {}
    for v in G:
        P[v] = []
    sigma = dict.fromkeys(G, 0.0)
    D = {}
    sigma[s] = 1.0
    D[s] = 0
    Q = [s]
```

```
    while Q:
        v = Q.pop(0)
        Dv = D[v]
        sigmav = sigma[v]
        for w in G[v]:
            if w not in D:
                Q.append(w)
                D[w] = Dv + 1
            if D[w] == Dv + 1:
                sigma[w] += sigmav
                P[w].append(v)
    return P,sigma

# função que excuta o algoritmo CIA
def vc(G):
    cia = dict.fromkeys(G, 0.0)
    u,v = random.sample(G.nodes(), 2)
    if nx.has_path(G,u,v):
        P,sigma = _single_source_shortest_path_basic(G, u)
        t=v
        while t != u:
            pred = P[t]
            d1=sigma[t]
            peso = [sigma[n]/d1 for n in pred]
            z = choice(pred, p=peso)
            if z != u:
                cia[z]+= 1/r
            t=z
    return cia

# executa a função "vc" em paralelo
def ci_am2(G):
    #map
    p1 = Pool(processes=multiprocessing.cpu_count())
    cia_results=p1.map(vc,[G for i in range(r)])

    #reduce
    bt_c = cia_results[0]
    for bt in cia_results[1:]:
```

```
        for n in bt:
            bt_c[n] += bt[n]
    return bt_c

# função que excuta o algoritmo CIAVLCM
def vcbncm(G,bn1,c):
    cibn = dict.fromkeys(G, 0.0)
    u,v = random.sample(bn1, 2)
    if nx.has_path(G,u,v) and c[u]!=c[v]:
        P,sigma = _single_source_shortest_path_basic(G, u)
        t=v
        while t != u:
            pred = P[t]
            d1=sigma[t]
            peso = [sigma[n]/d1 for n in pred]
            z = choice(pred, p=peso)
            if z != u:
                cibn[z]+= 1/r
            t=z
    return cibn

def multirun1(args):
    return vcbncm(*args)

# executa a função "vcbncm" em paralelo
def ci_bncm(G):
    nkG = nk.nxadapter.nx2nk(G)
    vert=nkG.nodes()
    bn=[]
    communities = nk.community.detectCommunities(nkG)
    for i in vert:
        for j in nkG.neighbors(i):
            if not communities.inSameSubset(i,j):
                bn.append(i)
    bn1=list(set(bn))
    c=communities.getVector()

    #map
    p2 = Pool(processes=multiprocessing.cpu_count())
```

```
    cibn_results=p2.map(multirun1,[(G,bn1,c) for i in range(r)])

    #reduce
    bt_c = cibn_results[0]
    for bt in cibn_results[1:]:
        for n in bt:
            bt_c[n] += bt[n]
    return bt_c

# executa as simulações dos algoritmos CIA e CIAVLCM
if __name__ == "__main__":
    G = nx.read_edgelist("grafo.txt",nodetype=int)
    giant = max(nx.connected_component_subgraphs(G), key=len)
    nkG = nk.nxadapter.nx2nk(giant)
    diam=nk.distance.EffectiveDiameterApproximation(nkG).run()
    vd=diam.getEffectiveDiameter()+1
    epsilon=0.05
    delta=0.1
    r=m.ceil((0.5/(epsilon*epsilon))*(m.floor(m.log2(vd-2))+1-m.log(delta)))
    print (vd,r)
    print(nx.info(G))

    cia_simula = []
    s=50    # simulações
    for j in range(s):
        cia=ci_am2(G)
        cia_simula.append(list(cia.values()))
    cia_simula1=pd.DataFrame(cia_simula)
    cia_simula1.T.to_csv("100milcia_simula50a.csv")

    cibncm_simula = []
    s=50    # simulações
    for j in range(s):
        cibncm=ci_bncm(G)
        cibncm_simula.append(list(cibncm.values()))
    cibncm_simula1=pd.DataFrame(cibncm_simula)
    cibncm_simula1.T.to_csv("100milcibncm_simula50a.csv")
```

A.2 CIA-Spark

```
#inicia Spark Context
from pyspark import SparkContext, SparkConf

conf = SparkConf().setAppName('MyFirstStandaloneApp')
sc = SparkContext(conf=conf)
print (sc)

# importa pacotes
import networkit as nk
import networkx as nx
import time
import pandas as pd
import random
from numpy.random import choice
import math as m
from heapq import heappush, heappop
from itertools import count
import pyspark

# função que calcula SSSP do vértice s para todos os outros via busca em largura
def _single_source_shortest_path_basic(G, s):
    P = {}
    for v in G:
        P[v] = []
    sigma = dict.fromkeys(G, 0.0)
    D = {}
    sigma[s] = 1.0
    D[s] = 0
    Q = [s]
    while Q:
        v = Q.pop(0)
        Dv = D[v]
        sigmav = sigma[v]
        for w in G[v]:
            if w not in D:
                Q.append(w)
                D[w] = Dv + 1
```

```
            if D[w] == Dv + 1:
                sigma[w] += sigmav
                P[w].append(v)
    return P,sigma

# função que excuta o algoritmo CIA
def vc(G):
    cia = dict.fromkeys(G, 0.0)
    u,v = random.sample(G.nodes(), 2)
    if nx.has_path(G,u,v):
        P,sigma = _single_source_shortest_path_basic(G, u)
        t=v
        while t != u:
            pred = P[t]
            d1=sigma[t]
            peso = [sigma[n]/d1 for n in pred]
            z = choice(pred, p=peso)
            if z != u:
                cia[z]+= 1/r
            t=z
    return cia

# executa a função "vc" no Spark no modo standalone na nuvem
# da Datbricks sobre HDFS
def ci_am4(G):
    grafos = sc.parallelize([G for i in range(r)],20).cache()
    cia_results=grafos.map(vc).flatMap(lambda x:
    [(k,v) for (k,v) in x.items()]).reduceByKey(lambda x,y: x+y).collect()
    return dict(cia_results)

# simulações de CIA
G = nx.read_edgelist("grafo.txt",nodetype=int)
giant = max(nx.connected_component_subgraphs(G), key=len)
nkG = nk.nxadapter.nx2nk(giant)
diam=nk.distance.EffectiveDiameterApproximation(nkG).run()
vd=diam.getEffectiveDiameter()+1
epsilon=0.05
delta=0.1
r=m.ceil((0.5/(epsilon*epsilon))*(m.floor(m.log2(vd-2))+1-m.log(delta)))
```

```
print (vd,r)
print(nx.info(G))

cia_simula = []
s=50    # simulações
for j in range(s):
    cia=ci_am4(G)
    cia_simula.append(list(cia.values()))
cia_simula1=pd.DataFrame(cia_simula)
cia_simula1.T.to_csv("100milcia_simula50a.csv")
```