MINISTÉRIO DA DEFESA
EXÉRCITO BRASILEIRO
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
CURSO DE MESTRADO EM SISTEMAS E COMPUTAÇÃO

ARGUS ANTONIO BARBOSA CAVALCANTE

PREDIÇÃO DE *LINKS* EM REDES SOCIAIS: RESGATANDO INFORMAÇÕES CONTEXTUAIS E TOPOLÓGICAS NO HISTÓRICO DE EVOLUÇÃO DO GRAFO

Rio de Janeiro
2020

ARGUS ANTONIO BARBOSA CAVALCANTE

PREDIÇÃO DE *LINKS* EM REDES SOCIAIS: RESGATANDO INFORMAÇÕES CONTEXTUAIS E TOPOLÓGICAS NO HISTÓRICO DE EVOLUÇÃO DO GRAFO

Dissertação apresentada ao Programa de Pós-graduação em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Mestre em Ciência em Sistemas e Computação.

Orientador: Ronaldo Ribeiro Goldschmidt - D.Sc.
Coorientador: Claudia Marcela Justel - D.Sc.

Rio de Janeiro

2020

©2020
INSTITUTO MILITAR DE ENGENHARIA
Praça General Tibúrcio, 80 – Praia Vermelha
Rio de Janeiro – RJ CEP: 22290-270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmar ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es) e do(s) orientador(es).

Cavalcante, Argus Antonio Barbosa
Predição de *Links* em Redes Sociais: Resgatando Informações Contextuais e Topológicas no Histórico de Evolução do Grafo / Argus Antonio Barbosa Cavalcante. – Rio de Janeiro, 2020.
92 f.

Orientador: Ronaldo Ribeiro Goldschmidt.
Coorientador: Claudia Marcela Justel.

Dissertação (mestrado) – Instituto Militar de Engenharia, Sistemas e Computação, 2020.

1. redes sociais virtuais. 2. mineração de dados. 3. mineração de *links*. 4. predição de *links*. I. Goldschmidt, Ronaldo Ribeiro, orient. II. Justel, Claudia Marcela, coorient. III.Título

ARGUS ANTONIO BARBOSA CAVALCANTE

Predição de Links em Redes Sociais: Resgatando Informações Contextuais e Topológicas no Histórico de Evolução do Grafo

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Mestre em Ciências de Sistemas e Computação.

Orientador: Ronaldo Ribeiro Goldschmidt
Coorientador: Claudia Marcela Justel

Aprovado em Rio de Janeiro, 30 de Janeiro de 2020, pela seguinte banca examinadora:

Prof. **Ronaldo Ribeiro Goldschmidt** - D.Sc. do IME - Presidente

Prof. **Claudia Marcela Justel** - D.Sc. do IME

Prof. **Julio Cesar Duarte** - D.Sc. do IME

Prof. **Eugênio da Silva - D.Sc.** - D.Sc. da UEZO

Rio de Janeiro
2020

Ao Instituto Militar de Engenharia,
alicerce da minha formação e aperfeiçoamento.

AGRADECIMENTOS

Agradeço à minha esposa Letícia pelo suporte, paciência e incentivo em todos os momentos dessa jornada, e também aos meus filhos Beatriz e Rafael pela motivação.

Agradeço também aos amigos Hamilton Lima e Paulo Motta, além dos amigos e companheiros de jornada Airan Camargo, Gabriel Machado e Paulo Marcio pela força e incentivo.

Agradeço de forma especial aos meus professores orientadores Ronaldo Ribeiro Goldschimidt e Cláudia Marcela Justel por sua atenção, disponibilidade e exemplo.

"Todas as coisas são números."

(Pitágoras)

RESUMO

A predição de *links* em redes sociais virtuais é uma tarefa de mineração de grafos. O objetivo desta tarefa é identificar pares de nós não conectados que possuem alta probabilidade de se conectar no futuro. De maneira geral, o estado da arte dos métodos de predição de *links* considera apenas a estrutura mais completa e recente da rede, não levando em consideração as informações sobre o estado da rede quando novas arestas foram adicionadas à sua estrutura. Os poucos trabalhos que o fizeram, utilizaram exclusivamente informações topológicas sem considerar as importantes informações contextuais disponíveis na rede. Este trabalho levanta a hipótese de que resgatar tanto informações topológicas quanto contextuais pode contribuir na criação de modelos preditivos com maior qualidade do que os existentes no estado da arte, tendo em vista que esses dados combinados podem vir a enriquecer a descrição do problema com exemplos que representam justamente o tipo de evento que se pretende prever: o surgimento de novas ligações. A fim de cumprir esse objetivo, é proposto um método de predição de *links* baseado em informações topológicas e contextuais do histórico de estados da rede social analisada. Os resultados dos experimentos com redes sociais de coautoria, frequentemente empregadas na avaliação de métodos de predição de *links* apontam para a validade da hipótese levantada.

Palavras-chave: *redes sociais virtuais. mineração de dados. mineração de links. predição de links.*

ABSTRACT

Link prediction in social networks is a graph mining task. This task's objective is to identify pairs of non-connected vertices that have a high probability to be connected in the near future. In general, the state-of-the-art link prediction methods only consider data from the most complete and recent structure of the network and do not take into account the information about the network state when new edges were added to its structure. The few works that have considered this approach have exclusively used topological information without using the meaningful contextual information available at the network. This study raises the hypothesis that recovering both topological and contextual data may contribute for building more accurate predictive models than the available ones, since those data enrich the description of the network with examples that represent exactly the kind of event to be foreseen: the appearance of new connections. For this purpose, this work proposes a link prediction method based on topological and contextual information from the history of the social network. Results from experiments with co-authorship social networks, frequently employed in link prediction methods evaluation, reveal the validity of the proposed method and the confirmation of the raised hypothesis.

Palavras-chave: online social network. data mining. link mining. link prediction.

LISTA DE ILUSTRAÇÕES

Figura 1 – Grafo representando uma RSV das Nações Unidas (GRANDJEAN, 2014) no qual os vértices representam pessoas e as arestas documentos. 16
Figura 2 – Predição de *links* - Ilustração gráfica de exemplo hipotético em uma rede com apenas 3 vértices. 18
Figura 3 – Exemplo de um grafo. 23
Figura 4 – Exemplo de um multigrafo. 23
Figura 5 – Exemplos de subgrafos. 24
Figura 6 – Exemplo de um grafo com atributos no qual os vértices representam capitais de estados brasileiros e as arestas a distância entre essas cidades. 24
Figura 7 – Exemplos de informações de uma rede de coautoria. 25
Figura 8 – Exemplos de informações topológicas representadas em um grafo . . . 26
Figura 9 – Exemplos de informações contextuais representadas em um grafo. . . . 26
Figura 10 – Exemplo da abordagem supervisionada na predição de *links*. 29
Figura 11 – Exemplo da abordagem não supervisionada na predição de *links* 30
Figura 12 – Dados estruturados e não estruturados de uma RSV 35
Figura 13 – Exemplo de modelo *Doc2Vec* com a matriz de parágrafo D e as matrizes de palavras $W_1, W_2, ..., W_n$ agrupados por média, ou concatenação em um vetor que é utilizado para prever uma próxima palavra *on*. (LE; MIKOLOV, 2014) 38
Figura 14 – (a) Dados não *clusterizados* e (b) dados *clusterizados*. 39
Figura 15 – Estado único 42
Figura 16 – Múltiplos estados 43
Figura 17 – Visão Geral do Método 50
Figura 18 – Preparação dos Dados 51
Figura 19 – Predição de Links 54
Figura 20 – Cálculo de Métricas 54
Figura 21 – (a) Representação dos pares de vértices conectados (elementos de $H_{(d_1,d_2,...,d_q)}$ representados como "1") para duas métricas arbitrárias (d_1, d_2) representadas no espaço 2-dimensional; (b) Exemplo de resultado da *clusterização* de pares de vértices conectados e os centroides (x) de cada um destes *clusters*. 56
Figura 22 – Exemplo de alocação de pares de vértices não conectados (elementos de $P_{(d_1,d_2,...,d_q)}$ representados como "0") aos *clusters*. 57
Figura 23 – Avaliação do Resultado 58

Figura 24 – Representação da lista ordenada com as predições L_r, das novas ligações de vértices do $Core$ que surgiram na rede E_{New}, além das predições corretas $E_{Correct}$ e incorretas $E_{Incorrect}$, ambas constantes da lista L_r e avaliadas em E_{New}. 59
Figura 25 – Exemplo de dados estruturados e não estruturados da rede de coautoria gr-qc. 74
Figura 26 – Particionamento do Grafo. 75
Figura 27 – Comparação entre o método proposto $H_{TopCont}$ (métrica de similaridade contextual utilizada: DC) e o método H_{Top} de (FLORENTINO; GOLDSCHMIDT, 2017) com base na medida de avaliação AUC média entre todos os *datasets*. 82
Figura 29 – Comparação entre o método proposto $H_{TopCont}$ (métrica de similaridade contextual utilizada: DC) e o método H_{Top} de (FLORENTINO; GOLDSCHMIDT, 2017) com base na medida de avaliação precisão média entre todos os *datasets*. 82
Figura 28 – Comparação entre o método proposto $H_{TopCont}$ (métrica de similaridade contextual utilizada: DC) e o método H_{Top} de (FLORENTINO; GOLDSCHMIDT, 2017) com base na medida de avaliação precisão por *dataset*. 83

LISTA DE TABELAS

Tabela 1 – Quadro resumo dos trabalhos do estado da arte analisados. 43
Tabela 2 – Lista de algoritmos e sua complexidade na qual $|V|$ representa a quantidade de vértices, $|E|$ quantidade de arestas, $|S|$ quantidade de estados da rede, $|D|$ quantidade de métricas de similaridade e $|X|$ quantidade de palavras a serem representadas. 71
Tabela 3 – Dados Estatísticos - 1992 a 1998 73
Tabela 4 – Exemplo de dados formatados obtidos da RSV. 74
Tabela 5 – Comparação entre combinações de métricas de similaridade contextuais e métricas topológicas clássicas do método proposto $H_{TopCont}$. 80
Tabela 6 – Comparação entre o método proposto $H_{TopCont}$ (métrica de similaridade contextual utilizada: DC) e o método H_{Top} de (FLORENTINO; GOLDSCHMIDT, 2017) com base na medida de avaliação cálculo do *improvement*. 81
Tabela 7 – Comparação entre o método proposto $H_{TopCont}$ (métrica de similaridade contextual utilizada: DC) e o método H_{Top} de (FLORENTINO; GOLDSCHMIDT, 2017) com base na medida de avaliação AUC. 81
Tabela 8 – Tempo de execução do experimento com o método proposto $H_{TopCont}$. . 84

LISTA DE ABREVIATURAS E SIGLAS

RSV	Rede social virtual
CN	Vizinhos Comuns
JC	Coeficiente de Jaccard
AA	Adamic-Adar
XML	eXtensible Markup Language
API	Application Programming Interface

SUMÁRIO

1 INTRODUÇÃO . . . 15
1.1 Contextualização e motivação . . . 15
1.2 Problema . . . 17
1.3 Objetivos do trabalho . . . 19
1.4 Método . . . 20
1.5 Contribuições esperadas . . . 20
1.6 Organização do texto . . . 20

2 FUNDAMENTAÇÃO TEÓRICA . . . 22
2.1 Considerações iniciais . . . 22
2.2 Conceitos em teoria de grafos . . . 22
2.3 Tipos de informação em RSVs . . . 24
2.4 Predição de links . . . 26
2.4.1 Abordagens . . . 28
2.4.2 Métricas de similaridade . . . 30
2.5 Mineração de dados . . . 34
2.5.1 Tipos e representações de dados . . . 34
2.5.2 Clusterização de dados . . . 38

3 TRABALHOS RELACIONADOS . . . 42
3.1 Proposta de modelo comparativo . . . 42
3.2 Comparação e análise dos trabalhos . . . 43
3.2.1 Informações topológicas e estado único . . . 44
3.2.2 Informações topológicas, contextuais e estado único . . . 45
3.2.3 Informações topológicas e múltiplos estados . . . 47

4 ABORDAGEM PROPOSTA . . . 50
4.1 Considerações preliminares . . . 50
4.2 Descrição conceitual . . . 50
4.2.1 Preparação dos Dados . . . 50
4.2.2 Predição de Links . . . 53
4.2.3 Avaliação do Resultado . . . 58
4.3 Análise da complexidade . . . 60

5 EXPERIMENTOS E RESULTADOS . . . 72
5.1 Considerações iniciais . . . 72

5.2 Protótipo e ambiente de apoio à experimentação 72
5.3 Configurações dos experimentos 73
5.4 Combinação entre métricas e representações de dados 76
5.4.1 Métrica de similaridade contextual DC (Doc2Vec Cosine) 77
5.4.2 Métrica de similaridade contextual BC (BoW Cosine) 78
5.4.3 Métrica de similaridade contextual TC (TF-IDF Cosine) 78
5.4.4 Métrica de similaridade contextual BJ (BoW Jaccard) 79
5.5 Resultados e avaliação 79
5.5.1 Primeira execução 79
5.5.2 Segunda execução 80
5.5.3 Tempo de execução 84

6 CONSIDERAÇÕES FINAIS 85

REFERÊNCIAS 87

1 INTRODUÇÃO

1.1 Contextualização e motivação

Nas últimas décadas houve um importante avanço na popularização do uso de sistemas de informação através da internet (KUMAR, 2019). Dentre as opções disponíveis destes sistemas, como sites de comércio eletrônico, *blogs* e portais de notícias, entre outros, destacam-se as redes sociais virtuais (RSVs), que hoje são utilizadas por milhões de pessoas em todo o mundo (KUMAR; NOVAK; TOMKINS, 2010). Uma RSV é uma estrutura composta de pessoas e outras entidades passíveis de representação em um ambiente virtual, como empresas, organizações não governamentais, órgãos do governo e outras instituições. Estas entidades possuem algum tipo de interação mútua, como a troca de mensagens, a colaboração e diversos outros tipos de relacionamento ou influência (MISLOVE et al., 2007).

As redes sociais virtuais mudam e sofrem alterações com o passar do tempo. Isto ocorre por conta da inclusão de novos usuários, que tipicamente se juntam a uma dessas redes através da publicação de um perfil ou de conteúdo relacionado à temática em questão, mas, principalmente, devido ao estabelecimento de novas conexões entre os usuários dessas redes. Essas novas conexões estão relacionadas a eventos como a indicação de uma amizade baseada em algum tipo de afinidade por parte da RSV, encontros aleatórios que ocorrem na exploração das informações disponíveis na rede, a sugestão de amizades por amigos em comum, entre outros.

Ao observar os dados estatísticos de uma das mais populares RSVs em todo o mundo, é possível ratificar a relevância destas redes para a sociedade, justificando o crescente interesse da academia no estudo da dinâmica de evolução da estrutura das mesmas. Em Junho de 2019, um número médio de 1,59 bilhão de usuários utilizaram o Facebook diariamente (FACEBOOK, 2019). No âmbito de seu impacto econômico, considerando números de Julho de 2019, o Facebook é uma empresa avaliada em mais de 562 bilhões de dólares americanos segundo o (NASDAQ, 2019).

Conforme ilustrado na Figura 1, uma RSV pode ser representada matematicamente através de um grafo altamente interligado. Nesse grafo, os vértices, ou nós, representam os usuários, enquanto as arestas, as conexões que ocorrem entre estes usuários.

Figura 1 – Grafo representando uma RSV das Nações Unidas (GRANDJEAN, 2014) no qual os vértices representam pessoas e as arestas documentos.

Dentre os diversos tipos de análise aplicáveis às RSVs, destacam-se:

- Detecção de comunidades (FORTUNATO, 2010)
- Análise do fluxo de informações (*Cascade behavior*) (CHENG et al., 2014)
- Predição de *links* (LIBEN-NOWELL; KLEINBERG, 2007a)

O objetivo da detecção de comunidades em grafos (FORTUNATO, 2010) é identificar grupos de vértices relacionados e também, a estrutura hierárquica eventualmente existente entre eles, utilizando apenas as informações disponíveis na estrutura do grafo. Os membros destes grupos possuem maior probabilidade de estarem ligados uns aos outros do que membros de outros grupos. O agrupamento destes vértices (ou *clusterização*) é realizado através da execução de algoritmo dedicado a este objetivo.

Considerando que o re-compartilhamento ou re-postagem de informações é uma das atividades mais comuns realizada pelos usuários das RSVs (CHENG et al., 2014), a análise do fluxo (ou *cascade*) de informações se propõe ao estudo da ocorrência e do crescimento em massa destes fluxos. Nesta tarefa, são analisados os fatores particulares

de cada RSV que podem contribuir para seu crescimento, como as características do conteúdo recebido e compartilhado, informações sobre a postagem original e informações estruturais e temporais dos próprios *cascades*. A partir destas informações uma abordagem de aprendizado de máquina é então utilizada para a predição.

Objeto de estudo do presente trabalho, a tarefa predição de links diz respeito a prever quais novas ligações surgirão em uma rede, por meio da análise das características de seus elementos e das interações que já ocorreram (LIBEN-NOWELL; KLEINBERG, 2007a). A predição de links é uma tarefa não trivial da análise de RSVs e mineração de dados considerada útil e aplicável em diversos domínios, tendo em vista que as ligações representam as interações entre seus elementos, característica principal e intrínseca das RSVs.

Um exemplo popular de aplicação da predição de links em RSVs é a recomendação, por parte da RSV, de novas conexões entre usuários (LU; ZHOU, 2011). De acordo com a temática particular da RSV em questão, estas novas ligações recomendadas podem ir além da esfera social, tocando questões econômicas, como no caso da sugestão de vagas de emprego ou da indicação de produtos e serviços, nos quais um usuário pode se interessar. Esta recomendação tem comumente objetivos comerciais, e aponta para o crescimento das próprias RSVs.

Outro tipo de aplicação está relacionada à segurança e trata da identificação, por parte de órgãos de segurança pública, de interações entre indivíduos suspeitos de atividades ilícitas, como o terrorismo e outros atos de violência. As interações entre esses indivíduos podem ocorrer através da troca de mensagens de cunho ilegal.

1.2 Problema

Inicialmente descrita no trabalho seminal de (LIBEN-NOWELL; KLEINBERG, 2007a), a predição de links (Figura 2) pode ser definida por meio da seguinte pergunta: levando-se em conta o estado de uma rede no tempo t, é possível inferir quais novas interações entre seus elementos (nunca ligados anteriormente), são mais prováveis de ocorrer em um futuro próximo $t'(t < t')$? O referido estado da rede diz respeito à totalidade de informações dos vértices e arestas disponíveis em uma RSV.

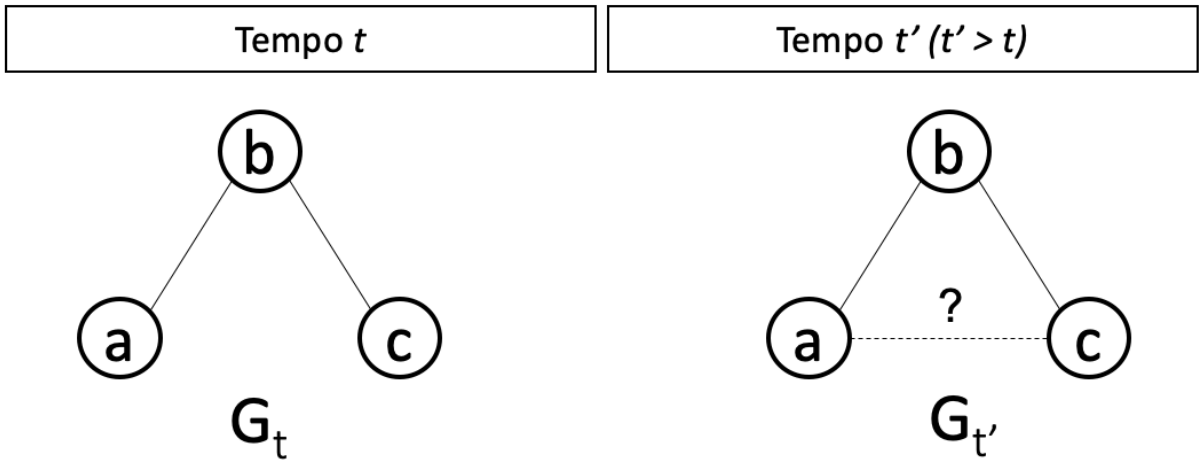


Figura 2 – Predição de *links* - Ilustração gráfica de exemplo hipotético em uma rede com apenas 3 vértices.

Existem duas abordagens comuns para o tratamento do problema da predição de *links*, a abordagem não supervisionada (LIBEN-NOWELL; KLEINBERG, 2007a), mais comum no estado da arte e a abordagem supervisionada (HASAN et al., 2006). A diferença entre essas abordagens é caracterizada pela transformação do problema da predição de links em uma tarefa de classificação binária com exemplos de elementos conectados e não conectados (supervisionada), versus o cômputo de métricas de similaridade (também chamadas funções de similaridade) que buscam expressar o grau de compatibilidade entre pares de vértices não conectados, a fim de ranqueá-los para posterior seleção daqueles com maiores chances de se conectarem (não supervisionada).

Um dos principais desafios da tarefa de predição de links consiste em determinar quais informações existentes nas redes, devem ser levadas em consideração para o cálculo de métricas de similaridade (PECLI; CAVALCANTI; GOLDSCHMIDT, 2018; MUNIZ; GOLDSCHMIDT; CHOREN, 2018). Essas informações são caracterizadas por sua natureza e seus principais tipos são: o topológico e o contextual. As informações topológicas representam a estrutura do grafo no contexto da aplicação em que a RSV se encontra inserida e não possuem significado, como por exemplo, o número de vizinhos que determinado vértice possui. Já no caso das informações contextuais, há semântica que representa explicitamente características e informações dos usuários das RSVs, bem como de suas interações com outros usuários. Por exemplo, a idade de um usuário, seu local de residência e os tipos de filmes e músicas favoritos.

Outro aspecto importante da predição de links diz respeito ao tratamento do estado da rede analisada. A maioria dos trabalhos do estado da arte incorpora todas as informações dos vértices e arestas para um dado tempo t como sendo parte de um único estado. Outra possibilidade é dividir a rede, e, portanto, as informações nela disponíveis,

em unidades de tempo segundo algum critério de separação temporal, criando assim múltiplos estados da rede.

Dentre os principais trabalhos relacionados que utilizam múltiplos estados da rede, destacam-se (SOARES; PRUDÊNCIO, 2012a) e (FLORENTINO; GOLDSCHMIDT, 2017). Ambos desenvolveram à sua maneira, abordagens que consideram múltiplos estados da rede. No entanto, uma análise mais detalhada do estado da arte mostra que a utilização dos múltiplos estados da rede ainda foi pouco explorada. Além disso, quando esta foi utilizada pelos referidos autores, apenas informações topológicas foram levadas em conta na tarefa da predição de *links*, deixando de lado as informações contextuais encontradas nas RSVs.

1.3 Objetivos do trabalho

Diante do exposto, a presente dissertação levanta a hipótese segundo a qual é possível melhorar os resultados da predição de links com múltiplos estados, combinando informações topológicas e contextuais. Esta hipótese se justifica com base na premissa de que o resgate de informações topológicas e contextuais sobre como era a rede no momento em que novas arestas foram incluídas no grafo pode levar a melhores predições, uma vez que tais informações enriquecem o conjunto de dados a ser analisado justamente com o tipo de evento que se deseja prever: o surgimento de arestas.

Dessa forma, o objetivo geral deste trabalho é demonstrar que o resgate de informações topológicas mais informações contextuais correspondentes ao momento de surgimento de novas arestas, poderá melhorar o desempenho na predição de links baseada em múltiplos estados do grafo.

Para cumprir esse objetivo geral, esta dissertação possui ainda os seguintes objetivos específicos:

- Formular um método que permita combinar informações topológicas e contextuais utilizando dados obtidos no momento de surgimento das arestas.
- Desenvolver um protótipo com base no método proposto para apoio à experimentação do objetivo geral deste trabalho.
- Realizar experimentos que permitam avaliar o método proposto em comparação ao estado da arte, em busca de evidências que possam suportar a hipótese levantada.

1.4 Método

De forma a atender os objetivos desta dissertação, foi desenvolvido um método não supervisionado de predição de links, que resgata informações topológicas e contextuais que representam o estado da rede no momento em que cada nova aresta foi incluída na mesma, formando pares de nós conectados. A partir deste resgate, ambos os tipos de informações são agregados e representados para então servirem de entrada para a aplicação de métricas de similaridade. Os resultados dessas métricas apresentados em um espaço q-dimensional passam por um processo de *clusterização* para representar os padrões de similaridade encontrados na RSV. Ao calcular essas mesmas métricas de similaridade para pares de nós, que não estão conectados, busca-se associar esses nós aos *clusters* nos quais esses melhor se ajustam. Dessa forma é possível, utilizando o critério do cálculo de *score* para cada um dos pares de nós não ligados, identificar aqueles pares mais passíveis de estabelecer uma ligação.

1.5 Contribuições esperadas

As contribuições esperadas como resultado deste trabalho são as seguintes:

- Melhoria nos resultados da tarefa de predição de links baseada em múltiplos estados quando comparada a outros métodos que também utilizam a mesma abordagem;
- Implementação de protótipo para realização de experimentos e validação da hipótese levantada;
- Modelo comparativo de trabalhos relacionados com base nos tipos de informações utilizados e do tratamento conferido ao estado da rede.

1.6 Organização do texto

Além da introdução, esta dissertação possui outros seis capítulos adicionais que são descritos brevemente a seguir:

No Capítulo 2 são apresentados os conceitos fundamentais relacionados a este trabalho no contexto da tarefa da predição de links, os tipos de informações utilizados e as estratégias para tratamento do estado da rede. Além disso, são introduzidas informações sobre aprendizado de máquina e sobre técnicas de clusterização de dados. No Capítulo 3, são discutidos os trabalhos relacionados por meio da sua descrição detalhada, com destaque para o tipo de informações, o tratamento dos estados da rede utilizado, e a comparação destes com a abordagem introduzida por esta dissertação. Na sequência, no Capítulo 4, a abordagem utilizada é descrita de forma a detalhar cada uma de suas etapas. Aqui será

discutida também, a complexidade dos algoritmos que representam detalhadamente os passos das etapas da abordagem proposta. Em seguida, no Capítulo 5, são registradas as informações sobre as características de execução do experimento desenvolvido, para validar a hipótese levantada pelo presente trabalho. Além disso, os resultados obtidos são apresentados e discutidos. Finalmente, no Capítulo 6, é feita uma retrospectiva do trabalho realizado, desde seu objetivo, passando pela comparação e apresentação de seus diferenciais em relação ao estado da arte, discutindo também as conclusões sobre os resultados obtidos nos experimentos e apontando direções para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Considerações iniciais

Nesta seção são introduzidos os conceitos fundamentais relacionados ao presente trabalho de forma a estabelecer a base de conhecimento necessária ao seu entendimento. Aqui são apresentadas informações e conceitos sobre teoria de grafos, principal estrutura utilizada para representar as RSVs. São discutidos também, os tipos de informações encontrados nas redes sociais virtuais e ainda, a tarefa de análise de RSVs, alvo de estudo desta dissertação, a predição de links. De forma similar, é feita uma introdução ao tema da mineração de dados, com informações sobre os principais tipos e representações de dados. Além disso, embora existam diversas outras tarefas, a *clusterização* de dados é discutida no contexto do método proposto no Capítulo 4, devido ao interesse existente como parte deste trabalho.

2.2 Conceitos em teoria de grafos

Os grafos são as estruturas utilizadas para representar os elementos de uma rede e suas ligações, e são comumente usados em tarefas de análise de RSVs (NETTLETON, 2013), como por exemplo na predição de *links*.

Um grafo G é composto basicamente por dois conjuntos (TRUDEAU, 1993): um conjunto não vazio e finito V de vértices (ou nós) e um segundo conjunto, também finito, E de arestas. Cada aresta $e \in E$ contém pares distintos e não ordenados de elementos do conjunto do vértices. Os grafos são ditos homogêneos quando possuem somente um tipo de vértice e um tipo de aresta. Já os grafos heterogêneos se caracterizam por possuírem mais de um tipo de vértice ou de arestas (KONG; YU, 2017).

A Figura 3 apresenta um grafo G cujo conjunto de vértices $V = \{u, v, w, z\}$ e o conjunto de arestas é $E = \{(u, v), (u, w), (v, w), (w, z)\}$. Dois vértices u e v são ditos adjacentes, ou vizinhos, se (u, v) é uma aresta do conjunto E, ou seja, dados $G = \left\langle V, E \right\rangle$ e $u, v \in V$, u é vizinho de v se existe $(u, v) \in E$.

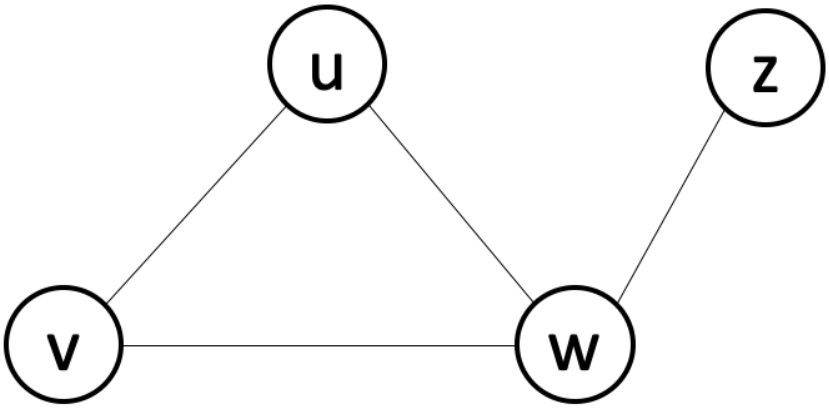


Figura 3 – Exemplo de um grafo.

Uma generalização possível a partir de um grafo diz respeito a permitir múltiplas arestas adjacentes ao mesmo par de vértices. Apresentado na Figura 4, este grafo é chamado de multigrafo (SHAFIE, 2015).

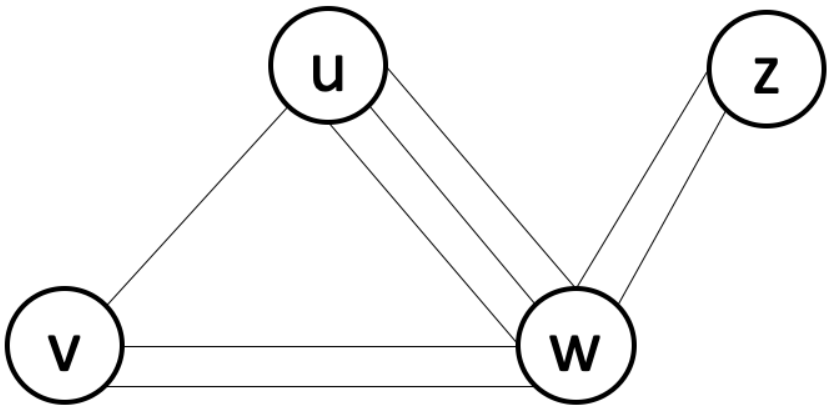


Figura 4 – Exemplo de um multigrafo.

Um subgrafo $G' = (V', E')$ de um grafo $G = (V, E)$ é um grafo no qual V' é um subconjunto de V, e E' é um subconjunto de E. Desta forma, o grafo apresentado na Figura 5 (b) é um subgrafo do grafo (a), mas (c) não é um subgrafo de (a). É possível obter um subgrafo de um grafo apenas removendo vértices e arestas (TRUDEAU, 1993).

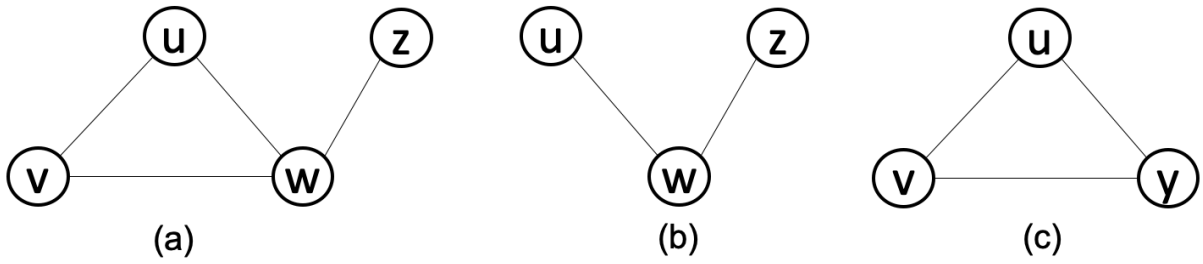


Figura 5 – Exemplos de subgrafos.

Os elementos de um grafo podem possuir características que são representadas através de atributos existentes nos vértices e arestas (XU et al., 2012). Esses atributos ampliam a capacidade de um dado grafo em generalizar outros modelos de dados, além de possibilitar que os referidos atributos sejam utilizados em tarefas de análise de RSVs. Um exemplo de um grafo com atributos é apresentado na Figura 6.

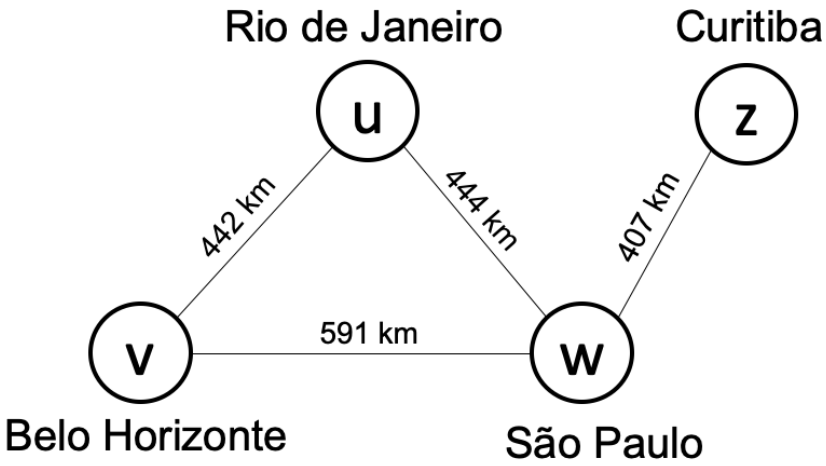


Figura 6 – Exemplo de um grafo com atributos no qual os vértices representam capitais de estados brasileiros e as arestas a distância entre essas cidades.

2.3 Tipos de informação em RSVs

Existem dois tipos distintos de informação disponíveis nas redes sociais virtuais: as informações topológicas e as contextuais. As informações topológicas são aquelas que descrevem a organização e estrutura do grafo, sem levar em conta qualquer semântica inerente ao contexto de aplicação da rede social (MUNIZ; GOLDSCHMIDT; CHOREN, 2018). As informações contextuais são aquelas que expressam características dos usuários das redes e de suas relações. Este tipo de informação possui significado intrínseco ao contexto de aplicação da rede social a qual pertencem. Exemplos: as palavras-chave e as áreas de conhecimento de artigos informados em uma rede social de coautoria de publicações

científicas, o momento de inserção de um pesquisador representado por um vértice no grafo e o ano de publicação de um artigo escrito em parceria por dois pesquisadores, entre outros exemplos possíveis. Tomando como exemplo um conjunto de dados de uma rede de coautoria (Figura 7), é possível distinguir os diferentes tipos de informação disponíveis em uma RSV. Nesse tipo de rede de colaboração estão disponíveis informações sobre as publicações científicas de diversos autores. Uma dada publicação é definida como uma publicação realizada em coautoria, quando dois ou mais autores colaboraram para seu desenvolvimento e publicação. As informações da Figura 7, são usadas nos exemplos representados em formato de grafo (Figuras 8 e 9) na sequência.

Id_Autor	Autor	Id_Autor	Autor	Id_Publicacao	Título	Ano	Resumo	Palavras-chave
a	V. T. Toth	**b**	J. W. Moffat	**1005.2685**	Can Modified Gravity (MOG) explain the speeding Bullet (Cluster)?	2016	We apply our scalar-tensor-vector (STVG) modified gravity theory (MOG) to calculate the infall velocities of the two clusters constituting the Bullet Cluster 1E0657-06.	STVG, 1E0657-06
a	V. T. Toth	**b**	J. W. Moffat	**1404.1845**	Karlhede's invariant and the black hole firewall proposal	2017	The Karlhede invariant is formed from the contraction of the covariant derivative of the Riemann tensor. It is a coordinate invariant that vanishes at the Schwarzschild event horizon r=2m.	Karlhede, Riemann
b	J. W. Moffat	**c**	S. Rahvar	**1807.07424**	Propagation of Electromagnetic Waves in MOG: Gravitational Lensing	2018	We investigate the solution of Maxwell's equations in curved spacetime within the framework of Modified Gravity (MOG).	Maxwell, MOG

Figura 7 – Exemplos de informações de uma rede de coautoria.

Conforme a Figura 8, no âmbito das informações topológicas, temos por exemplo o grau de um vértice (número de arestas incidentes a um vértice) ou a quantidade de vizinhos comuns a um par de vértices, são informações de natureza tipicamente topológica e, como tal, retratam aspectos estruturais da rede.

Vale ressaltar que na Figura 8, os rótulos de identificação (a, b e c) foram incluídos a título de ilustração, visto que como estes identificadores possuem significado para a rede em questão, não são desta forma estritamente topológicos, mas contextuais.

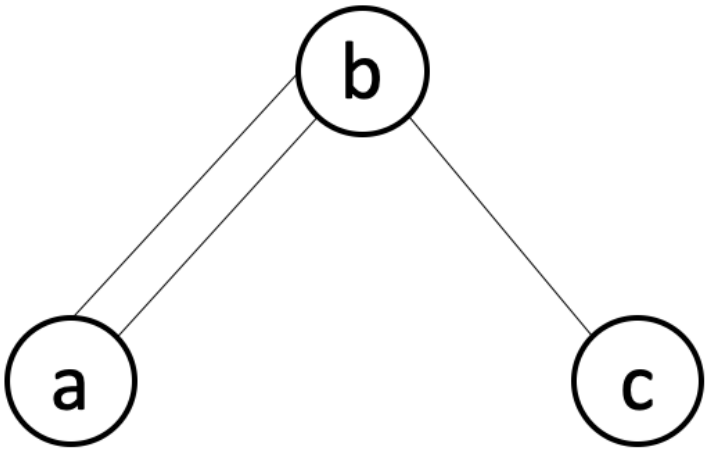


Figura 8 – Exemplos de informações topológicas representadas em um grafo

Em relação às informações contextuais, a Figura 9 mostra um exemplo de uma RSV correspondente a uma rede de coautoria; os vértices a, b, c representam pesquisadores, as arestas representam publicações conjuntas entre dois pesquisadores, e associadas com as arestas existem informações contextuais como ano de publicação e palavras-chave do artigo. Esta informação de cunho temporal (o ano de publicação neste caso) pode também ser entendida como um tipo particular de informação contextual, conforme introduzido por (MUNIZ; GOLDSCHMIDT; CHOREN, 2018) e caracterizada em função de seu aspecto cronológico.

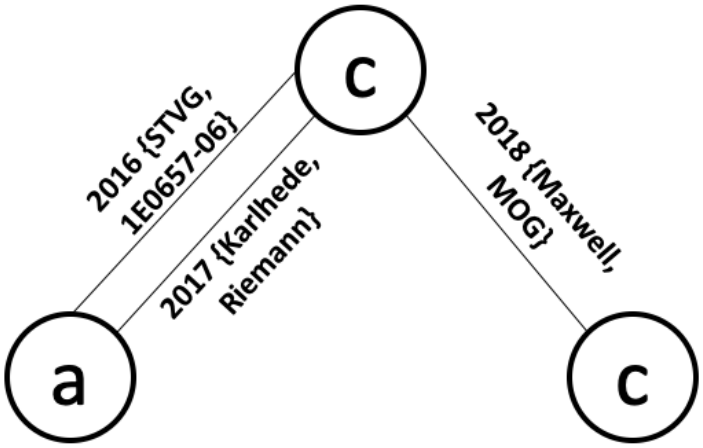


Figura 9 – Exemplos de informações contextuais representadas em um grafo.

2.4 Predição de links

Conforme surgiram e se estabeleceram diversas redes sociais virtuais, como Facebook, Instagram, Linkedin, entre outras, o conceito de rede social passou a ocupar um lugar de relevância em nossa sociedade. Anteriormente à consolidação desse conceito, no

entanto, as redes sociais já eram objeto de estudo de pesquisadores em todo o mundo. Um dos temas mais relevantes nesse campo de pesquisa é o fenômeno *small-world* (TRAVERS; MILGRAM, 1969), cujo experimento mais amplamente difundido tratou de recrutar um grupo de cem indivíduos em Omaha, uma cidade nos Estados Unidos da América. Para cada um destes indivíduos (u) foi enviado um pacote com instruções e uma carta que deveria ser enviada a um outro indivíduo (v), incluindo informações adicionais sobre o destinatário, como seu o endereço, sua ocupação e os lugares onde estudou. Caso u conhecesse v, poderia enviar diretamente a carta para este, porém caso não o conhecesse (o que ocorreu na maioria dos casos) deveria fazer com que a carta chegasse ao seu destino através de pessoas que ele conhecia e que, com base nas informações repassadas sobre o destino, poderiam possuir algum tipo de relação com este. A definição de amizade neste contexto era representada pela ideia de que u conhecia v pelo seu primeiro nome. Entre as 100 cartas enviadas (de fato 96, já que 4 indivíduos não iniciaram o processo), 18 chegaram ao seu destino. O tamanho médio do caminho percorrido foi de seis pessoas, nos fornecendo a noção hoje popular dos chamados seis graus de separação, que se sustenta na ideia de que são necessários no máximo os referidos seis passos, para que quaisquer duas pessoas estejam ligadas.

Outra tema em destaque é a homofilia (MCPHERSON; SMITH-LOVIN; COOK, 2001), que representa o princípio segundo o qual o contato entre indivíduos similares, ou seja, que compartilham características e tem perfis parecidos, ocorre mais frequentemente do que com pessoas dissimilares. Como resultado desse princípio é possível concluir que as redes de contato mais próximos das pessoas são homogêneas em relação a características interpessoais, de comportamento e sócio demográficas. A homofilia pode também ser vista como um elemento limitante, tendo em vista que uma fronteira entre as redes sociais das quais cada indivíduo faz parte e tende a influenciar nas informações que uma pessoa recebe, suas atitudes e mesmo nas interações que esta pessoa experimenta ao longo de sua vida. A homofilia é, portanto, um princípio básico de organização dos grupos sociais.

De maneira geral, entender os mecanismos através dos quais uma rede social do mundo real se desenvolve serve como fundamento para a análise e estudo das RSVs. As RSVs são modelos altamente dinâmicos, crescendo e mudando rapidamente com o passar do tempo através do surgimento de novas ligações entre seus elementos. Ao estudo do surgimento dessas novas ligações, dá-se o nome de predição de links, problema cuja representação computacional é subjacente ao tema da evolução das próprias RSVs. Na prática, o problema da predição de links se apresenta com os seguintes questionamentos: O que uma rede social pode dizer a respeito de seu próprio futuro? Até que ponto a evolução de uma rede social pode ser modelada usando apenas suas próprias características?. Ao modelarmos uma rede social como um grafo, é possível observar a noção de proximidade de seus componentes, conforme o fenômeno *small-world*, visto que elementos conectados tendem a estar localizados mais próximos uns dos outros na estrutura desta rede. Sob o

aspecto da similaridade, como introduzido pela homofilia, temos a noção de que indivíduos conectados possuem algum tipo de semelhança ou interesse comum que os torna conectados.

A seguir, estes conceitos são exemplificados através de alguns cenários de RSVs reais: No caso de um blog de esportes, as pessoas que acessam e comentam o conteúdo publicado por seus autores são muito provavelmente interessados no tema. Ao utilizar um aplicativo de mensagens, possivelmente o fazemos com pessoas do nosso círculo social, como familiares, colegas de trabalho, amigos de escola ou faculdade com os quais estudamos ou com quem temos qualquer outro tipo de laço de amizade. Em uma rede social para compartilhamento de música, teremos pessoas que consomem ou produzem este tipo de conteúdo, entre diversos outros exemplos possíveis. Na perspectiva do problema da predição de links, todos estes diferentes tipos de interações servem como insumo para a inferência de futuras novas ligações.

Além da predição de links futuros, outra vertente possível diz respeito ao problema da predição de links a inferência de links faltantes. Considerando os elementos de uma rede particular no tempo t, seu objetivo é verificar em um tempo futuro t' quais das ligações existentes em t deixarão de existir. É aplicável principalmente a outros tipos de redes, não somente as RSVs, como por exemplo as redes de computadores e equipamentos.

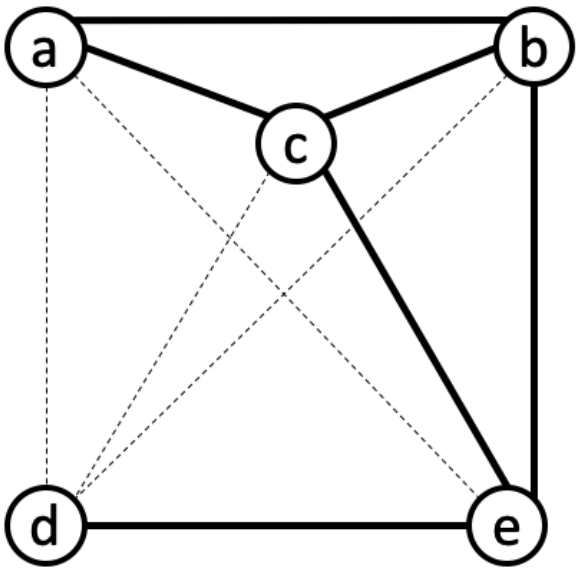
Voltando ao objeto de estudo do presente trabalho, na predição de links futuros procede-se tipicamente (LU; ZHOU, 2011) a representar os elementos de uma rede através de um grafo. Este grafo $G = \left\langle V, E \right\rangle$ possui um conjunto V de vértices v que representam os indivíduos e um outro conjunto E cujos elementos e são arestas no grafo, ligando um par de vértices de V. Para que seja possível utilizar os dados disponíveis em uma rede para avaliar os métodos de predição de links, a abordagem mais comumente utilizada passa por particionar ou dividir este grafo G em dois períodos distintos com seus respectivos subgrafos. No primeiro deles chamado de treino ou G_{Treino}, temos a estrutura que será utilizada para alimentar a predição. Um segundo subgrafo, G_{Teste}, é utilizado para verificar a efetividade do método empregado, ou seja, verificar se de fato, a inferência de novas ligações se concretizou.

2.4.1 Abordagens

As abordagens utilizadas para resolução do problema da predição de links se dividem em supervisionada e não supervisionada.

A abordagem supervisionada (HASAN et al., 2006) trata da criação de um modelo para análise de exemplos, cuja resposta esperada é definida por um supervisor externo. Este modelo deve ser capaz de avaliar hipóteses induzidas a partir de um conjunto inicial de dados, quando colocado diante de novos exemplos. As técnicas da abordagem supervisionada incluem algoritmos de classificação, tais como redes neurais e árvores de

decisão, cujo objetivo é gerar um modelo de classificação que será aplicado na predição. Dessa forma, a tarefa de predição de links é geralmente transformada em um problema de classificação binária na qual cada par de vértices é avaliado diante das classes positiva, nós ligados, e negativa, não ligados. Estes conjuntos são enriquecidos com informações extraídas da rede utilizadas para o cálculo similaridade. No exemplo da Figura 10 tem-se um conjunto de vértices e suas ligações, representados por meio de um grafo no qual as linhas contínuas representam uma ligação entre dois vértices e as linhas tracejadas, ligações não existentes. Nesse exemplo, as informações sobre a quantidade de vizinhos compartilhados simultaneamente por dois vértices é calculada através da métrica de similaridade *Vizinhos comuns*. Ambos os vértices a e b, por exemplo, possuem uma ligação com c. A ligação de b com o vértice e, no entanto não é compartilhada por a, resultando em um total de apenas um vizinho em comum. A classificação binária e o resultado das métricas são utilizadas então para treinar um modelo de classificação capaz de avaliar novos exemplos em relação a esse modelo de aprendizado. Calculando uma ou mais métricas de similaridade e tendo a classe desses elementos definida (Ligação = Sim para os vértices a e b), é possível realizar o mesmo processo para pares de vértices não ligados, inferindo sua ligação futura, como no caso dos pares de vértices não conectados (a, d), (a, e) e etc.

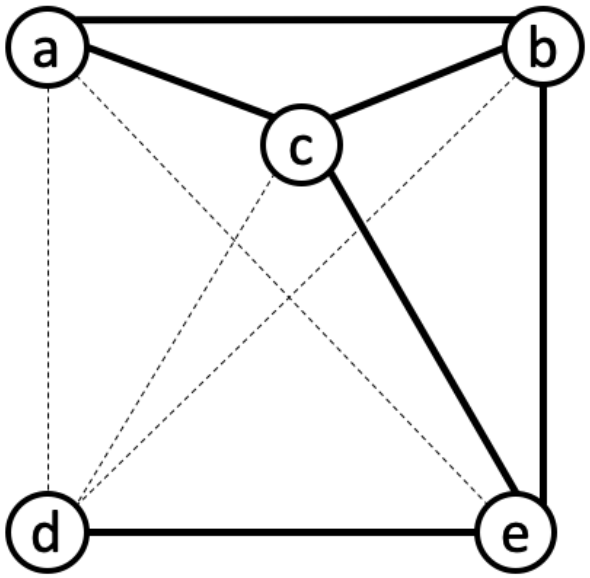


Nó *u*	Nó *v*	Ligação
a	b	**Sim**
a	c	**Sim**
a	d	**Não**
a	e	**Não**
b	c	**Sim**
b	d	**Não**
b	e	**Sim**
c	d	**Não**
c	e	**Sim**
d	e	**Sim**

Figura 10 – Exemplo da abordagem supervisionada na predição de *links*.

Na abordagem não supervisionada (LIBEN-NOWELL; KLEINBERG, 2007a), mais comumente adotada na tarefa de predição de links (HASAN; ZAKI, 2011), é realizado um cálculo nos pares de vértices não ligados que resulta em um grau de similaridade,

ou compatibilidade entre esses elementos. Conforme exemplo da Figura 11, a métrica de similaridade *Vizinhos Comuns* foi utilizada para avaliação da similaridade entre os pares de vértices não conectados, indicados através das linhas tracejadas. O resultado dessas métricas é então ordenado de forma decrescente e assume-se que quanto maior for esse grau de similaridade, maiores as chances desses vértices se tornarem conectados no futuro.



Nó *u*	Nó *v*	CN(u,v)
a	d	0
a	e	2
b	d	1
c	d	1

Nó *u*	Nó *v*	CN(u,v)
a	**e**	**2**
b	**d**	**1**
c	d	1
a	d	0

Figura 11 – Exemplo da abordagem não supervisionada na predição de *links*

2.4.2 Métricas de similaridade

Tanto na abordagem supervisionada, quanto na abordagem não supervisionada para a predição de links, as informações disponíveis na rede são utilizadas para que seja verificado o grau de similaridade entre pares de vértices. Esse processo ocorre com o uso de métricas criadas para executar o cálculo dessa similaridade. Dentre as mais populares métricas de similaridade topológicas, ou seja, métricas que utilizam esse tipo de informação, destacam-se no estado da arte (LU; ZHOU, 2011; HASAN; ZAKI, 2011; NETTLETON, 2013): *Vizinhos em Comum*, *Adamic-Adar* e *Conexão Preferencial*, entre outras que são detalhadas na sequência.

Vizinhos em Comum (CN): Representa a noção de que dois desconhecidos que

compartilham um amigo em comum podem ser apresentados um ao outro por esse amigo (NEWMAN, 2001). Essa apresentação sugere o efeito do fechamento de um triângulo entre esses três vértices em um grafo, que funciona de maneira similar a um comportamento social comum no mundo real. A Equação 2.1 define formalmente a métrica, representando a interseção entre o conjunto dos vértices vizinhos de u e o conjunto dos vértices vizinhos de v.

$$CN(u,v) = |\Gamma(u) \cap \Gamma(v)| \tag{2.1}$$

Adamic-Adar (AA): A similaridade de Adamic-Adar (ADAMIC; ADAR, 2003) é um refinamento da função *Vizinhos Comuns* que atribui maior peso aos vizinhos que possuem uma quantidade menor de vizinhos. Essa métrica, apresentada na Equação 2.2, formaliza a noção intuitiva de que um nó que não possui um número tão grande de conexões (ou de amigos, por exemplo) possui uma probabilidade maior de apresentar dois de seus amigos um ao outro, do que um nó com muitas conexões.

$$AA(u,v) = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{1}{log|\Gamma(z)|} \tag{2.2}$$

Katz (KA): representa a soma do conjunto de caminhos entre u e v de comprimento l, exponencialmente amortecida para dar maior peso a caminhos mais curtos (KATZ, 1953). A Equação 2.3 apresenta a fórmula dessa métrica.

$$KA(u,v) = \sum_{\ell=1}^{\infty} \delta^{\ell} \cdot |paths_{u,v}^{<\ell>}| \tag{2.3}$$

Alocação de Recurso (RA): Avaliação da quantidade de características (ou recursos) que podem ser transferidas de um nó para outro. Esta função (apresentada na Equação 2.4) considera que quanto menor for a quantidade de vizinhos de um vizinho comum z, menor a possibilidade de dispersão destas características (ZHOU; LÜ; ZHANG, 2009).

$$RA(u,v) = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{1}{|\Gamma(z)|} \tag{2.4}$$

Conexão Preferencial (PA): Uma ideia bastante difundida na análise de redes sociais tem relação com a noção de que indivíduos com muitas ligações tendem a criar um

número maior de novas ligações no futuro, ou seja, indivíduos populares ou conhecidos por sua capacidade de influenciar um maior número de pessoas, terão potencializada a sua capacidade de estabelecer novas conexões (NEWMAN, 2001). A estimativa dessa popularidade ou influência se dá através da multiplicação entre o número de vizinhos (amigos, ou seguidores no contexto das RSVs) que cada um dos elementos do par de vértices possui. Esta métrica está representada na Equação 2.5.

$$PA(u, v) = |\Gamma(u)| \cdot |\Gamma(v)| \tag{2.5}$$

Além de métricas exclusivamente topológicas, houve tentativas de combinar simultaneamente informações topológicas e contextuais em uma mesma métrica de similaridade, como por exemplo, na métrica clássica *Similaridade do Cosseno* (SALTON, 1989) e também nas métricas *Time Score* e *Context-based Time Score*:

Time Score (TS): A métrica *Time Score* de (MUNASINGHE; ICHISE, 2012), apresentada na Equação 2.6, considera a média harmônica do número de interações entre nós com cada vizinho em comum multiplicada por um fator de amortecimento β, que é utilizado para conferir maior peso às interações mais recentes. O denominador desta fração usa a diferença entre t_1 e t_2, as datas mais recentes das interações entre os vizinhos em comum e os nós u e v. Ainda, k é a diferença entre o tempo atual e o *timestamp* mais recente de t_1 e t_2, para cada uma das n ligações entre u e v. É importante ressaltar que, além das informações contextuais, esta métrica considera também informações topológicas disponíveis na rede.

$$TS(u, v) = \sum_n \frac{Hm_n(\beta^{k_n})}{|t_{1_n} - \ t_{2_n}| + 1} \tag{2.6}$$

Context-based Time Score (CTS): (MUNIZ; GOLDSCHMIDT; CHOREN, 2018) estendeu o comportamento da métrica *Time Score*, incorporando nesta métrica, além do fator tempo, informações sobre palavras-chave das publicações dos autores de uma base de coautoria. Segue sua formulação na Equação 2.7.

$$CTS(u, v) = TS(u, v) \times \alpha^{(1-Cos(u,v))} \tag{2.7}$$

Similaridade do Cosseno (Cos): A similaridade do cosseno (ou índice de Salton) (SALTON, 1989) é usada para determinar a similaridade entre dois elementos representados

como vetores. Seu cálculo é realizado com base no produto escalar normalizado de dois vetores com o uso da norma Euclidiana. Considerando dois vetores A e B de dimensão n, a similaridade do cosseno entre esses vetores é calculada conforme descrição da Equação 2.8.

$$Cos(A, B) = \frac{A.B}{\|A\| \, \|B\|} = \frac{\sum_{i=1}^{n} A_i.B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{2.8}$$

Uma outra métrica popular é o *Coeficiente de Jaccard* que possui flexibilidade para ser aplicada em informações exclusivamente topológicas ou ainda, em informações contextuais, como por exemplo nas palavras-chave de uma publicação.

Coeficiente de Jaccard (JC): Extremamente comum em outros campos além da predição de links (LIBEN-NOWELL; KLEINBERG, 2007b), o coeficiente de *Jaccard* representa a ideia de que a quantidade de indivíduos que interagiram com u e que também interagiram com v (e vice versa) se apresentam como uma boa maneira de medir a similaridade de u e v. O coeficiente de *Jaccard* resolve também uma deficiência da métrica *Vizinhos em Comum* devido ao fato de que em algumas situações o número total de vizinhos pode ser grande, mas não representativo em relação à similaridade como no caso de celebridades e influenciadores digitais, por exemplo. Na prática essa métrica representa uma normalização da métrica *Vizinhos em Comum*. Segue abaixo sua Equação 2.9.

$$JC(x, y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|} \tag{2.9}$$

Considerando a existência de múltiplas ocorrências de um mesmo tipo de dado nos vértices e arestas de uma RSV, as métricas de similaridade topológicas ou contextuais invariavelmente dependem de algum tipo de consolidação para que seu cálculo seja realizado de forma correta. Por exemplo, ao analisar a similaridade entre os esportes preferidos por um usuário, em relação aos esportes preferidos por um segundo usuário, não faz sentido escolher apenas um entre esses esportes para realizar qualquer tipo de comparação, sendo necessário dessa forma, agregar esses diversos esportes preferidos por ambos os usuários. O estado da arte apresenta algumas possibilidades para agregar ambos os tipos de dados, estruturados e não estruturados. Uma função de agregação pode ser definida como $fa(o, (u, v))$, onde o é um operador de agregação e (u, v) um par de vértices sobre o qual será aplicada a função de agregação.

Em (HASAN et al., 2006), foi utilizada a soma como função de agregação dos dados de informações topológicas e contextuais, que no contexto de uma rede de coautoria pode incluir a soma da quantidade de publicações de um determinado autor, a soma dos

vizinhos (coautores com quem um autor publicou) ou vizinhos dos vizinhos de um autor, a soma das palavras-chave das publicações e dos códigos de classificação.

Em (POPESCUL; POPESCUL; UNGAR, 2003), foram utilizadas funções de agregação típicas de bancos de dados relacionais para combinar características de vértices e arestas para a predição de *links* baseada em um modelo estatístico. As funções utilizadas no referido trabalho foram *count*, *avg*, *max*, *min*, *mode*, e *empty*, variando conforme o resultado de agregação esperado.

(MUNIZ; GOLDSCHMIDT; CHOREN, 2018) utilizou as operações básicas da teoria dos conjuntos como união e interseção para combinar palavras-chave das publicações de autores em uma rede de coautoria.

2.5 Mineração de dados

2.5.1 Tipos e representações de dados

É possível encontrar informações contextuais de indivíduos (vértices) e de interações (arestas) representadas como dados estruturados e não estruturados em redes sociais virtuais (MARTÍNEZ; BERZAL; CUBERO, 2016). Mesmo deixando de lado imagens e vídeos, que exigem tratamento e técnicas específicas, exemplos de dados estruturados e não estruturados são encontrados em grande variedade em diversas RSVs.

Segundo (GOLDSCHMIDT; PASSOS, 2005) os dados estruturados utilizados para a aplicação de técnicas de mineração de dados podem ser: qualitativos ou quantitativos. Os dados qualitativos contêm os valores nominais e ordinais. Já os dados quantitativos são representados por valores numéricos, sendo discretos ou contínuos. Esse esquema de organização está representado na Figura 12.

Os dados qualitativos nominais se caracterizam por indicar nomes ou atribuir rótulos a objetos. Não sendo aplicável nesse tipo de atributo o uso de operadores aritméticos. Também não existe ordem entre os valores, que são comumente representados por valores alfanuméricos (ex. estado civil: solteiro, casado, etc.). Os atributos qualitativos ordinais são similares aos nominais em relação à sua função primária (dar nomes a objetos) e quanto à impossibilidade de aplicação de operadores aritméticos. Porém, diferentemente dos nominais, no caso dos ordinais existe ordem entre os valores (ex. dias da semana: domingo, segunda-feira, terça-feira, etc. ou renda: de R$ 1.000,00 até R$ 1.999,00, de R$ 2.000,00 até R$ 2.999,00, etc.).

Os dados quantitativos discretos por sua vez, possuem um número contável de valores (finito ou infinito) e nele são aplicáveis os operadores aritméticos. Além disso, existe um sentido de ordem entre os valores que utilizam dados numéricos (ex. idade, quantidade de itens de determinado produto em estoque, etc.). No caso dos atributos contínuos, a

principal característica que o diferencia dos quantitativos discretos é a possibilidade de assumir infinitos valores (ex. renda, altura, peso, temperatura e etc.).

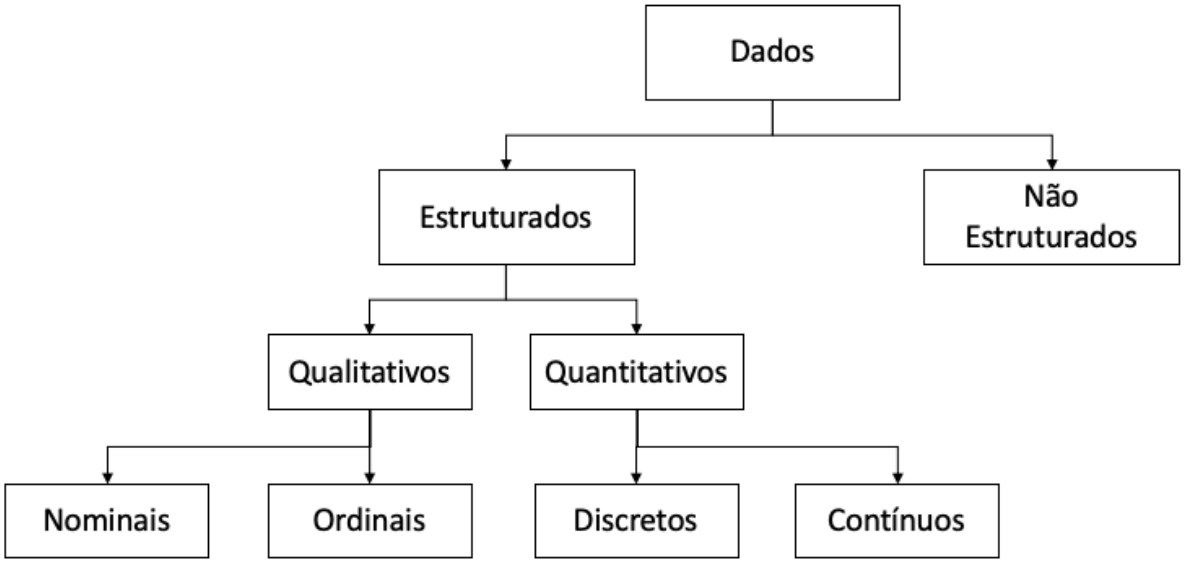


Figura 12 – Dados estruturados e não estruturados de uma RSV

No caso dos dados não estruturados, (GAMA et al., 2011) apresenta uma distinção entre as técnicas tradicionais de mineração de dados e a mineração de texto. Enquanto a mineração de dados tradicional, se baseia em dados gerados por sistemas de informação que já se encontram em um formato de dados estruturados (tipicamente de atributo-valor), os dados de texto não estruturado precisam passar por um processo de pré-processamento para que se alcance o formato estruturado, sendo a partir de então possível, a aplicação de técnicas e algoritmos clássicos de mineração de dados.

A mineração de texto pode ser definida como um conjunto de métodos utilizados para analisar dados não estruturados e identificar padrões que eram até então desconhecidos (SINOARA; ANTUNES; REZENDE, 2017), como por exemplo o padrão de similaridade entre dois textos. Durante sua análise, esses métodos devem lidar com a linguagem em formato natural (também chamado na literatura como Processamento de Linguagem Natural ou NLP), ou seja, a linguagem que as próprias pessoas utilizam para se comunicar umas com as outras. Processar linguagem humana em formato natural impõe diversos desafios, dada a ambiguidade intrínseca e variável das linguagens, além do fato de que estas se alteram e evoluem com certa frequência com o passar do tempo.

De forma geral, os métodos de mineração de texto são aplicados em três níveis distintos de texto: documento, parágrafo e sentença (KOWSARI et al., 2019). Quando extraídos de sua fonte de dados original, os dados não estruturados em formato de texto usualmente não estão representados em um formato que permita a aplicação dos métodos

de extração padrão, sendo dessa forma necessário fazê-lo por meio de uma etapa de pré-processamento.

Uma das formas mais comuns de representação de dados não estruturados é a *bag-of-words* (ou BoW) (HARRIS, 1954) na qual um texto é representado por um vetor de palavras. BoW é uma representação simplificada de um texto baseada em um critério de frequência das palavras dentro desse mesmo texto, presente em qualquer um dos níveis anteriormente citados (documento, parágrafo e sentença), que pode incluir apenas palavras distintas (removendo palavras duplicadas) ou ainda incluir todas as palavras, mesmo as repetidas. Para que isso ocorra é gerada uma lista de palavras que não possuem qualquer tipo de relacionamento semântico entre si ou mesmo estrutura gramatical. Em seguida, cada ocorrência de cada uma das palavras é contada numericamente, conforme exemplo apresentado na sequência (LU; HENCHION; NAMEE, 2019):

Texto = ['Manually labelling large collections of text data is a time-consuming, expensive, and laborious task, but one that is necessary to support machine learning based on text datasets. Active learning has been shown to be an effective way to alleviate some of the effort required in utilising large collections of unlabeled data for machine learning tasks without needing to fully label them. ']

Palavras = ['active', 'alleviate', 'an', 'and', 'based', 'be', 'been', 'but', 'collections', 'consuming', 'data', 'datasets', 'effective', 'effort', 'expensive', 'for', 'fully', 'has', 'in', 'is', 'label', 'labelling', 'laborious', 'large', 'learning', 'machine', 'manually', 'necessary', 'needing', 'of', 'on', 'one', 'required', 'shown', 'some', 'support', 'task', 'tasks', 'text', 'that', 'the', 'them', 'time', 'to', 'unlabelled', 'utilising', 'way', 'without']

Ocorrências por palavra = [1 1 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 2 1 1 1 2 3 2 1 1 1 3 1 1 1 1 1 1 1 1 2 1 1 1 1 4 1 1 1 1]

Uma versão mais simples do *bag-of-words* é aplicada quando não se permitem palavras duplicadas na lista, gerando desta forma uma lista de ocorrências mais simples com as palavras que estão ou não estão incluídas na lista.

Uma limitação imposta pela utilização da frequência de termos diz respeito a privilegiar palavras bastante utilizadas, mas implicitamente comuns em um conjunto de documentos. (JONES, 1988) desenvolveu um método de representação que resolve esta limitação penalizando termos que são frequentes em vários documentos chamado de *TF-IDF*, ou *Term Frequency - Inverted Document Frequency*. No *TF-IDF*, o *score* de uma palavra aumenta à medida que esse vocábulo aparece no documento, porém diminui, na medida em que muitos documentos contém esse termo. A representação matemática do *TF-IDF* é apresentada na Equação 2.10. Nessa equação tf é a contagem (frequência) da ocorrência de um termo em um documento. Enquanto *df* é a medida

calculada com no número total de documentos dividido pelo número de arquivos nos quais um determinado termo aparece, indicando se essa palavra é comum ou rara entre um conjunto de documentos.

$$W_{x,y} = tf_{x,y} \times \log \frac{N}{df_x} \tag{2.10}$$

Embora tanto o *BoW* quanto *TF-IDF* sejam caracterizados por sua simplicidade, ambos apresentam uma importante desvantagem. A ordem das palavras é totalmente perdida, assim sentenças completamente diferentes podem resultar em uma mesma representação quando as mesmas palavras são utilizadas. Além disso, sem ordem há perda semântica do texto analisado. Visando tratar essa restrição, (LE; MIKOLOV, 2014) propôs um modelo de representação chamado vetor de parágrafo, ou *Doc2Vec*, que é capaz de criar uma representação vetorial distribuída a partir de porções variáveis de texto, desde uma sentença, até mesmo um documento completo.

O *Doc2Vec* é inspirado na abordagem que utiliza a representação de vetores de aprendizado (BENGIO et al., 2006; COLLOBERT; WESTON, 2008; MIKOLOV et al., 2013). Nesse modelo, cada palavra é mapeada para um único vetor, representado por uma coluna em uma matriz. Essa coluna é indexada com base na posição da palavra no vocabulário. A soma ou concatenação dos vetores resultantes é então utilizada como entrada para prever a próxima palavra em uma sentença. Ou seja, palavras com significado similar são mapeadas para posições similares no espaço vetorial. Diferentemente da representação *BoW*, na qual as palavras são dimensões de um vetor, no *Doc2Vec* as dimensões são obtidas com base em uma rede neural (MIKOLOV et al., 2013).

Para tratar o problema de forma mais abrangente, o *Doc2Vec* incorpora um vetor no qual são armazenadas as características de um conjunto de palavras (o parágrafo). Um exemplo desse modelo aplicado na identificação de uma próxima palavra em uma frase a partir de um conjunto inicial de palavras conhecidas é apresentado na Figura 13.

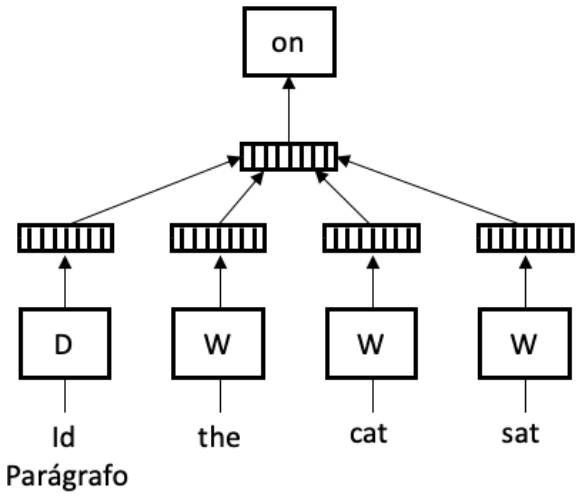


Figura 13 – Exemplo de modelo *Doc2Vec* com a matriz de parágrafo D e as matrizes de palavras $W_1, W_2, ..., W_n$ agrupados por média, ou concatenação em um vetor que é utilizado para prever uma próxima palavra *on*. (LE; MIKOLOV, 2014)

2.5.2 Clusterização de dados

O aprendizado de máquina é um campo da inteligência artificial cujo objetivo é desenvolver conhecimento de forma automática, a partir de algum tipo de exemplo e com a utilização de técnicas computacionais voltadas a esse fim. O conhecimento desenvolvido para executar determinada tarefa, será continuamente aperfeiçoado. As técnicas computacionais utilizadas envolvem algoritmos capazes de tomar decisões baseadas na experiência prévia com a resolução de algum tipo de problema (JAIN; MURTY; FLYNN, 1999). Os exemplos que geram a experiência de aprendizado devem suportar a descrição do problema em questão e o tipo de resolução aplicada em determinado contexto. O aprendizado de máquina pode ser dividido em duas vertentes: o aprendizado supervisionado e o não supervisionado. No aprendizado supervisionado os exemplos disponíveis devem conter o resultado esperado que será produzido ao final da tarefa, enquanto no aprendizado não supervisionado esse resultado esperado não existe.

Tarefa de aprendizado de máquina não supervisionado de grande relevância, a *clusterização* de dados tem por objetivo organizar automaticamente um conjunto de dados em k grupos, chamados *clusters* (JAIN; MURTY; FLYNN, 1999). Nessa técnica não existem exemplos pré-definidos que indiquem o resultado esperado, ou seja, não é realizado nenhum tipo de aprendizado indutivo a partir de rótulos dos exemplos.

A ideia central da *clusterização* é que os elementos que fazem parte de um mesmo *cluster* devem apresentar algum tipo de similaridade entre si, além de serem dissimilares de elementos de outros *clusters*. Ou seja, os grupos de dados de um mesmo *cluster* são tão homogêneos quanto possível, mas heterogêneos na comparação com elementos de outros

clusters. A Figura 14 apresenta um exemplo do processo de *clusterização* de dados.

Embora não exista um consenso claro sobre a definição de *clusterização*, algumas regras gerais podem ser utilizadas (XU; TIAN, 2015):

- Objetos em um mesmo *cluster* devem ser tão similares quanto for possível;
- Objetos em *clusters* diferentes devem ser tão diferentes quanto for possível;
- As medidas de similaridade e dissimilaridade devem ser claras e possuir significado prático.

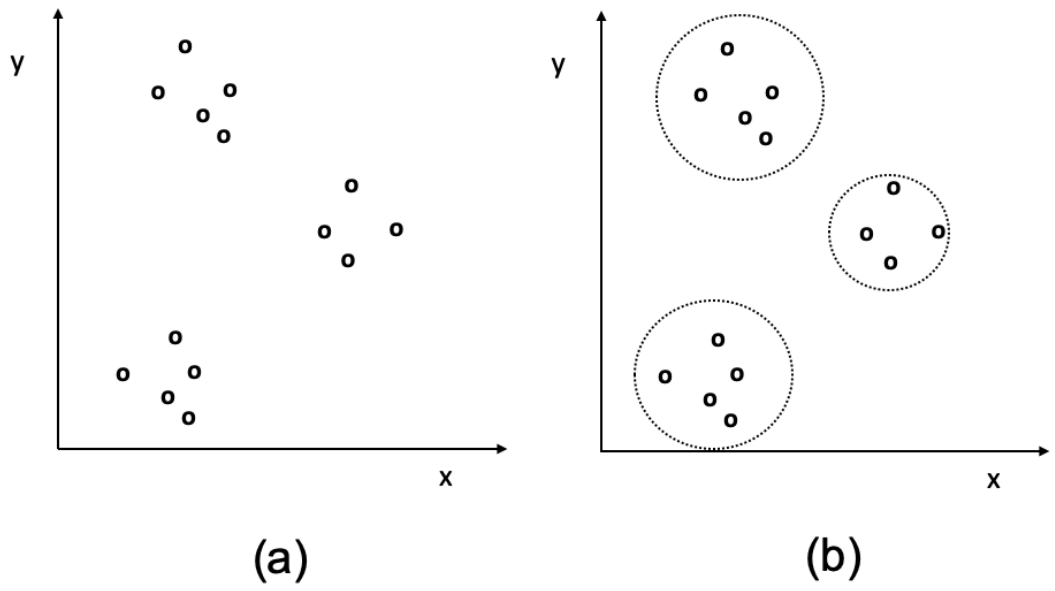


Figura 14 – (a) Dados não *clusterizados* e (b) dados *clusterizados*.

Os algoritmos de *clusterização* podem ser utilizados para gerar ideias a respeito do conjunto de dados analisado, com base nas características similares que os dados agrupados em *cluster* possuem, sendo uma maneira de descrever esses dados e ressaltar suas características.

Formalizando o conceito de *clusterização* (HRUSCHKA; EBECKEN, 2003), temos um conjunto de n objetos $X = \{X_1, X_2, ..., X_n\}$ onde, $X_i \in X$ deve ser representado por um vetor de dimensão q, que representa a quantidade de medidas que dimensionam as características do objeto X_i, sendo os dados agrupados em k *clusters* distintos $C = \{C_1, C_2, ..., C_k\}$.

A construção dos algoritmos de *clusterização* tem como base a relação entre os conceitos de distância e similaridade, na qual quanto menor a distância entre seus elementos, maior a sua similaridade. No caso de atributos quantitativos, é preferível reconhecer o

relacionamento entre diferentes objetos através do cálculo de distância. As métricas mais comuns para esse tipo de cálculo são a distância *Euclidiana* e a correlação de distância de *Pearson* (XU; TIAN, 2015). Ao lidar com dados qualitativos, a tendência é que seja utilizado o cálculo de similaridade, assim como parte das métricas utilizadas na própria tarefa da predição de links. Exemplos deste tipo de cálculo são o *coeficiente de Jaccard*, *similaridade de Hamming*, etc. (XU; TIAN, 2015)

Segundo (XU; TIAN, 2015), existem nove categorias que são utilizadas para classificar os algoritmos de *clusterização*, os algoritmos baseados em partição, hierárquicos, baseados em densidade, *fuzzy*, teoria dos grafos, baseados em *grid*, distribuição, teoria fractal ou baseados em modelos. Descreve-se a seguir um dos tipos mais comuns de algoritmos de *clusterização*: os algoritmos baseados em partição.

A ideia central dos algoritmos baseados em partição tem relação com centro dos pontos que representam os dados em cada um dos *clusters*. O algoritmo *k-means* é um dos mais populares nessa categoria (XU; TIAN, 2015) e está formalmente descrito pelo Algoritmo 1. Esse algoritmo atualiza o centro (chamado de centroide) dos *clusters* e atribui a esses *clusters* os elementos analisados. Esse processo é realizado de forma iterativa e continua até que algum critério de parada seja atingido. O algoritmo *k-means* tem como entrada o número k de *clusters* que serão construídos ao final de sua execução. Na sequência, procede-se a seleção randômica dos centroides dos k *clusters* escolhidos. Supõe-se que seja dado como entrada $k = 2$, os centroides destes 2 *clusters* serão aleatoriamente selecionados.

A seguir, a distância (ex. distância euclidiana) de cada um dos elementos a serem *clusterizados* será calculada em relação aos *clusters* randomicamente escolhidos e assim esses elementos serão associados aos centroides mais próximos. Em seguida, os centroides são atualizados, porém agora não aleatoriamente mas em função dos elementos atribuídos a um referido *cluster*. Esse processo é repetido até que uma condição de parada seja atendida. Essas condições incluem a estabilização dos valores dos centroides (que não mudam em uma nova iteração) e a não modificação dos elementos atribuídos a cada *cluster*. É possível também, determinar adicionalmente um número máximo de iterações para o algoritmo.

Algoritmo 1: Clusterizar dados

Entrada: Lista de elementos que serão *clusterizados* P, número de *clusters* k

Saída: Lista de elementos clusterizados *ElementosClusterizados*

```
1  início
2      Lista de Clusters de tamanho k recebe valores aleatórios definidos como os
         centroides de cada um dos k clusters
3      repita
4          para cada elemento p de P faça
5              para cada cluster da Lista de Clusters faça
6                  Calcula a distância entre p e o centroide de cada cluster
7                  ElementosClusterizados recebe cluster e centroide mais próximo
                     de p.
8              fim
9          fim
10         para cada cluster da Lista de Clusters faça
11             Recalcula o centroide
12         fim
13     até Não haver mais alterações na associação dos elementos aos clusters;
14 fim
15 retorna
```

3 TRABALHOS RELACIONADOS

Com o objetivo de analisar o estado da arte do tema da predição de links, foi realizada a revisão da bibliografia relacionada ao tema de pesquisa. Definida a questão de busca, a pesquisa foi feita tendo como referencial os anos de 2015 a 2018, mas não se limitando a esse período em casos específicos de trabalhos que possuem relação direta com a abordagem aqui introduzida. A análise dos trabalhos foi feita tendo como referência o modelo comparativo proposto que é apresentado na sequência.

3.1 Proposta de modelo comparativo

O primeiro critério utilizado para a análise do estado da arte foi o tipo de informação extraído das RSVs e utilizado como entrada para a aplicação de métricas de similaridade. Conforme detalhado na Seção 2, essas informações podem ser do tipo topológico ou contextual.

Além do tipo de informação, foi avaliado o tipo de tratamento desses trabalhos em relação ao estado da rede e que serve de base para a execução das abordagens supervisionada e não supervisionada na tarefa da predição de *links*. Considerando a visão do particionamento do grafo G que representa a estrutura da rede em dois subgrafos distintos G_{Treino} e G_{Teste}, existe uma diferença básica entre as possibilidades de representação e resgate das informações que são utilizadas como entrada para a predição de links e que portanto fazem parte de G_{Treino}. A primeira possibilidade é considerar todos os elementos do período de treino como um só estado, conforme representado na Figura 15.

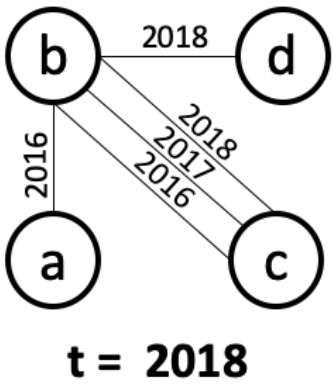


Figura 15 – Estado único

Uma outra forma de manipular os dados do grafo de treino G_{Treino}, é particionar

este grafo em períodos de tempo distintos, criando uma visão atualizada para cada um desses períodos e que retratam de forma fiel o estado da rede nesses períodos. Nessa visão, têm-se múltiplos estados da rede. A visão com múltiplos estados tem relação intrínseca com a predição de links, haja visto que ela retrata exatamente aquilo que se deseja prever, ou seja, o surgimento de novas ligações na rede. A Figura 16 apresenta os estados nesse formato.

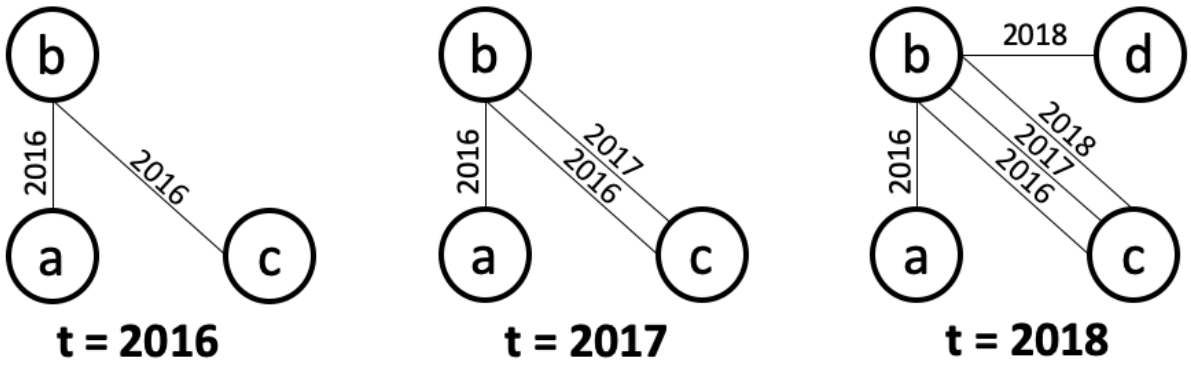


Figura 16 – Múltiplos estados

3.2 Comparação e análise dos trabalhos

A Tabela 3.2 apresenta um quadro resumo dos trabalhos selecionados com base nos critérios constantes no modelo comparativo proposto.

	Tipos de Informações		**Estado da Rede**	
Referência	**Topológicas**	**Contextuais**	**Estado Único**	**Múltiplos Estados**
(HE et al., 2015)	x		x	
(AOUAY; JAMOUSSI; GARGOURI, 2015)	x		x	
(NEGI; CHAUDHURY, 2016)	x		x	
(LAISHRAM; MEHROTRA; MOHAN, 2016)	x		x	
(WU; ZHANG; REN, 2017)	x		x	
(YUAN et al., 2015)	x		x	
(SHARMA; RATHORE; PARK, 2017)	x	x	x	
(SHALFOROUSHAN; JALALI, 2015)	x	x	x	
(RUMMELE; ICHISE; WERTHNER, 2015)	x	x	x	
(AHMED; CHEN, 2016)	x	x	x	
(YAO et al., 2016)	x	x	x	
(ZENG; DING; XIA, 2016)	x	x	x	
(HAN et al., 2015)	x	x	x	
(TASNADI; BEREND, 2015)	x	x	x	
(DEYLAMI; ASADPOUR, 2015)	x	x	x	
(MUNIZ; GOLDSCHMIDT; CHOREN, 2018)	x	x	x	
(SOARES; PRUDÊNCIO, 2012b)	x			x
(FLORENTINO; GOLDSCHMIDT, 2017)	x			x

Tabela 1: Quadro resumo dos trabalhos do estado da arte analisados.

3.2.1 Informações topológicas e estado único

Dentre os trabalhos selecionados, nesta subseção são apresentados aqueles que, de forma distinta do método proposto no presente trabalho, utilizaram apenas o estado único da rede, além de não terem levado em consideração informações contextuais disponíveis nas redes analisadas. Ao deixar de considerar múltiplos estados, esses trabalhos não consideraram o estado da rede no momento do surgimento de novas arestas. Deixando de lado as informações contextuais, não foram avaliadas importantes características dos usuários das RSVs.

Considerando a premissa de que as informações topológicas oriundas da rede podem variar intensamente e impactar os algoritmos utilizados na tarefa de predição de links, os autores de (HE et al., 2015) desenvolveram um mecanismo de predição baseado no conceito de OWA (*Ordered Weighted Averaging*) para algoritmos combinados. A ideia central dessa abordagem consiste em implementar vários algoritmos individuais e então organizar esses algoritmos de maneira correta, visando aumentar a performance na tarefa de predição. Aplicado a tarefa de predição, esse conceito trata de métricas de similaridade clássicas do estado da arte como *Vizinhos Comuns*, *Adamic Adar* e *Coeficiente de Jaccard*, entre outras. O resultado do cálculo das métricas é então combinado com uma abordagem que privilegia com peso seus elementos, resultando na melhor combinação possível desses resultados. Embora tenham obtido resultados relevantes, seus autores não avaliaram a aplicabilidade dessa abordagem em redes sociais de grandes dimensões.

Os autores de (AOUAY; JAMOUSSI; GARGOURI, 2015) se inspiraram no algoritmo PSA (*Particle Swarm Optimization*) combinado com uma abordagem de aprendizado de máquina supervisionado para tratar o problema da predição de links. A estratégia PSA é uma meta heurística bio inspirada, cujo objetivo é modelar a dinâmica de interação entre pessoas com base em como partículas próximas umas das outras se associam e cooperam entre elas se comportando como uma unidade (ex. enxame, matilha, cardume, bando e etc.), de forma similar ao que ocorre na vida real ao tratar de como uma pessoa é influenciada pelo comportamento de outras pessoas em seu círculo social. Utilizando a abordagem supervisionada, seus autores estabelecem comparações com as métricas de similaridade clássicas como *Vizinhos Comuns*, *Adamic-Adar* e *Conexão Preferencial* que se justifica diante de sua abordagem exclusivamente topológica.

A análise de redes sociais que possuem diferentes tipos de interação entre seus participantes apresenta um desafio bastante específico na tarefa de predição de links, visto que uma abordagem tradicional aplicada a uma rede homogênea não pode ser utilizada da mesma forma nesse tipo de situação. O trabalho de (NEGI; CHAUDHURY, 2016) se baseia na hipótese de que tratar da mesma forma relações heterogêneas implica em ignorar padrões de dependência entre os diferentes tipos de ligações. O objetivo dos autores é o aprendizado de medidas de distância para cada tipo de ligação e seu uso para a predição.

Para aplicar esse conceito a uma rede heterogênea, seus autores identificaram os tipos de ligações presentes na rede, decompondo a rede original em redes individuais homogêneas, a partir das quais são calculadas métricas de similaridade.

Como tipicamente os trabalhos de predição de links assumem que as conexões formadas em uma rede são permanentes, (LAISHRAM; MEHROTRA; MOHAN, 2016) se propõe a investigar redes que, ao contrário desta premissa, consideram que essas conexões não são permanentes, ou seja, poderão ser criados ou removidos. O modelo criado com esse objetivo, se baseia na definição de um peso para as conexões existentes entre dois vértices, que pode também resultar em um valor negativo para os casos em que uma referida conexão deixa de existir. As métricas de similaridade *Vizinhos Comuns*, *Adamic Adar*, *Conexão Preferencial* e outras baseadas em informações topológicas, foram adaptadas para refletir o peso das conexões, e seu resultado é utilizado como entrada para um algoritmo de aprendizado supervisionado que é então utilizado para prever novas conexões.

Os autores de (WU; ZHANG; REN, 2017) trataram a predição de links com uma abordagem baseada em comunidades. Essa abordagem busca incorporar informações de uma estrutura de comunidade de forma a aumentar a acurácia na predição. A performance de métodos baseados em comunidade tipicamente é, no entanto, sensível ao algoritmo de detecção selecionado, que em sua maioria falham em capturar a relação entre a criação de uma ligação e a evolução das comunidades. O método proposto por seus autores inclui a criação de um modelo baseado na maximização da modularidade, que objetiva tratar essa lacuna. A ideia central desse método é integrar a formulação de dois tipos de links em um modelo generativo de uma rede particionada. Sua abordagem foi experimentalmente comparada com modelos de classificação e outras abordagens baseadas em comunidade. Esse trabalho utiliza apenas informações sobre a estrutura da rede.

Em (YUAN et al., 2015) é introduzido um método que usa um algoritmo de *clusterização* para agrupar as informações dos usuários das RSVs depois da execução de uma técnica de redução de dados chamada de *MapReduce*. O algoritmo de *clusterização* é chamado *ROCK* e as métricas utilizadas foram métricas clássicas como *Adamic-Adar* e *Resource Allocation*.

3.2.2 Informações topológicas, contextuais e estado único

Os trabalhos deste grupo exploraram a combinação de diferentes tipos de informações, porém, diferentemente do método proposto neste trabalho, não consideraram múltiplos estados da rede, utilizando dessa forma uma visão estática da rede.

Uma rede *bayesiana* é um modelo confiável para entender as relações entre variáveis que tem sido usado em diversas tarefas de predição. Aplicando esse conceito na tarefa da predição de links, (SHALFOROUSHAN; JALALI, 2015) desenvolveram um método

que encontra os atributos que possuem maior efeito na criação de novas amizades em RSVs, para em seguida sugerir novas amizades com outras pessoas com maior grau de similaridade, ou seja, verificam-se os atributos que possuem maior efeito na probabilidade de uma nova amizade ocorrer. Além de informações topológicas, informações contextuais como a região onde determinada pessoa vive, sua idade, nível educacional, seus *hobbies*, a empresa onde ele ou ela trabalham, além de seu estado civil e sexo são utilizados. A referida abordagem supervisionada considera apenas o estado atual da rede e foi comparada com sucesso com outras abordagens (*Friend of Friend* e *FriendLink*) e também com as métricas de similaridade *Coeficiente de Jaccard* e, *Adamic Adar*.

(SHARMA; RATHORE; PARK, 2017) concentra seus esforços em um subconjunto de participantes das RSVs, representado pelos novos usuários que ainda não estabeleceram nenhum tipo de ligação ou usuários existentes que estabeleceram poucas ligações sociais. Dessa forma, o objetivo da predição passa por identificar com quais usuários já existentes, esses novos usuários mais provavelmente se ligarão. Seus experimentos foram aplicados também à afinidade de usuários a produtos em uma relação comercial, que extrapola o aspecto social. O método aplicado envolve a aplicação de técnicas de aprendizado profundo.

Os autores de (RUMMELE; ICHISE; WERTHNER, 2015) utilizaram a abordagem supervisionada da predição de links. Seu método inclui a contagem de sub grafos com três elementos, a partir do qual é identificado um tipo de padrão (ou tríade) formada com seus vizinhos. Essas tríades passam a incorporar um esquema de pesos que privilegia alguns nós e tipos de interação. Do ponto de vista da abordagem, foram utilizadas técnicas de *Logistic Regression*, *Condicional Inference Trees* e *Random Forests*. Sua base de informações é principalmente topológica, mas foi estendida para incorporar tempo como tipo de informação contextual.

Em (AHMED; CHEN, 2016) é abordada a incerteza inerente a alguns tipos de redes, que contêm dados incorretos, incompletos ou corrompidos em suas estruturas. Considerando que a similaridade dos elementos é parte importante do processo de predição em redes sociais, seu entendimento é de que a incerteza encontrada nestas redes vai afetar de forma determinante a eficácia da predição. Sua abordagem utiliza a técnica de *Random Walks*, com informações contextuais e topológicas disponíveis na rede estudada.

Em (YAO et al., 2016) é proposto um método de predição de links centrado na métrica *Vizinhos Comuns*. Diferentemente de grande parte dos outros diversos métodos que também utilizam essa popular métrica, nesse trabalho seu resultado é combinado com informações contextuais (o peso de conexões que variam com o tempo) e outras que representam a topologia da rede, como a frequência com que ocorrem mudanças em novos links e a intimidade entre vizinhos comuns.

No trabalho de (ZENG; DING; XIA, 2016) tem-se a apresentação de uma abordagem que constrói uma visão dinâmica da rede com peso para vértices e arestas. Baseado na

abordagem supervisionada para a predição de links, seus autores fazem uso de técnicas de classificação dos elementos, com o uso de informações contextuais disponíveis na rede.

Em (HAN et al., 2015) os autores desenvolveram o conceito de *single platform* versus *cross platform* na predição de links. Ao considerar que a questão da privacidade pode levar usuários de RSVs a não compartilhar os dados de suas interações (como por exemplo mensagens trocadas com outros usuários), são utilizados apenas os dados do perfil dos usuários das RSVs, cadastrados no momento da criação de uma nova conta. A partir daí são estabelecidas classificações para esses dados que serão utilizados e as métricas de similaridade calculadas servem de insumo para redes *SVM*.

(TASNADI; BEREND, 2015) desenvolveram um método baseado em informações implícitas disponíveis nas redes, incluindo a linguagem utilizada pelos membros das redes e as classificações concedidas por estes em revisões de restaurantes (caso tratado pelo *dataset* utilizado yelp.com). As métricas de similaridade criadas por seus autores tratam do vocabulário utilizado pelos usuários combinadas com informações estruturais da rede em relação à sua proximidade. Foi utilizada a abordagem supervisionada na predição de links.

No caso de (DEYLAMI; ASADPOUR, 2015) foi desenvolvido um algoritmo de detecção de comunidades. Considerando a hierarquia gerada por esse método, são identificados os elementos originadores e recebedores de novas ligações. Sua apresentação se dá de forma similar a uma extensão da métrica *Vizinhos Comuns* para um generalização aplicada ao conceito de comunidade. Sua abordagem é então comparada com métricas de similaridade clássicas.

(MUNIZ; GOLDSCHMIDT; CHOREN, 2018) explorou a combinação entre informações topológicas e contextuais disponíveis nas redes, com foco na execução da abordagem não supervisionada da predição de links. Seus autores propuseram 3 tipos de combinação: informações topológicas com informações contextuais de natureza temporal, informações contextuais de natureza temporal com informações contextuais de outra natureza, e os três tipos de combinados (informações topológicas, informações contextuais de natureza temporal e informações contextuais de outra natureza).

3.2.3 Informações topológicas e múltiplos estados

Os trabalhos desta seção consideraram o uso dos múltiplos estados da rede, mas de forma diferente do presente trabalho, deixaram de fora as importantes informações contextuais disponíveis na rede, desconsiderando dessa forma informações que descrevem características dos usuários das RSVs.

O trabalho de (SOARES; PRUDÊNCIO, 2012a) procura transformar o problema de predição de ligações em um problema de classificação a partir de séries temporais. Para

tanto, dividem a rede em *frames* (intervalos) de tamanho fixo e calculam, para o subgrafo associado a cada intervalo, diversas métricas de similaridade sobre pares de vértices não conectados. Assim, cada métrica de similaridade dá origem a uma série temporal cujos valores da dimensão cronológica são os extremos dos sucessivos intervalos. Os autores avaliaram diferentes modelos de classificação baseados nas múltiplas séries temporais construídas. Nesse cenário, uma importante limitação do método proposto em (SOARES; PRUDÊNCIO, 2012a) em relação ao método proposto pelo presente trabalho tem relação ao resgate de informações históricas acerca de vértices não conectados. Assim sendo, como a escolha do comprimento do intervalo de cálculo das métricas de similaridade é arbitrário, não há como garantir que o cálculo dessas informações seja no momento exato em que ocorram eventos de surgimento de arestas. Outro problema é que diferentes intervalos consecutivos podem apresentar valores idênticos de medidas de similaridade para vértices conectados. Isso pode ocorrer sempre que não tenham surgido novas arestas durante os referidos intervalos.

No método proposto no presente trabalho, o resgate das informações históricas é guiado pelo surgimento de novas arestas, ou seja, o cálculo das medidas de similaridade é feito considerando única e exclusivamente os exatos momentos em que ocorreram alterações na estrutura do grafo. Dessa forma, a informação recuperada é aquela que pode, de fato, conter indícios das circunstâncias que tenham levado ao surgimento de arestas e que, portanto, pode ser útil na construção de modelos preditivos mais precisos do que os existentes.

Outro aspecto importante observado em relação ao trabalho de (SOARES; PRUDÊNCIO, 2012a) é que a complexidade computacional de seu processo é da ordem de $max\{O(n^2), F\}$, na qual n corresponde ao número de vértices da rede e F a complexidade do modelo de séries temporais escolhido para cada experimento. Assumindo que F é menor que $O(n^2)$, a complexidade dos algoritmos é dominada pelo primeiro passo no qual as métricas de similaridade são calculadas para pares de vértices não conectados. $O(n^2)$ reflete o pior caso, no qual o grafo não contém nenhum vértice. Embora o número de *frames* definido pelo usuário dependa de características da rede analisada, este tende a ser baixo quando comparado ao número de vértices em um grafo.

Em (FLORENTINO; GOLDSCHMIDT, 2017), os autores se basearam em informações históricas que descrevem como era a topologia de uma RSV no momento em que cada aresta foi incluída na rede, gerando, portanto, para sua análise, múltiplos estados da rede. A partir do resgate das informações e da aplicação de métricas de similaridade exclusivamente topológicas, seguiu-se um processo de *clusterização* desses dados com o objetivo de identificar padrões de similaridade de topologia entre os eventos de surgimento das arestas. Na sequência, as densidades dos *clusters* identificados foram utilizadas no cálculo dos elementos não conectados e, finalmente, aplicada a avaliação do desempenho

da predição. O método apresentado pelo presente trabalho foi adaptado a partir da ideia concebida por (FLORENTINO; GOLDSCHMIDT, 2017) e busca sanar sua principal limitação: não levar em conta a riqueza das informações contextuais disponíveis em uma dada RSVs.

A complexidade computacional do trabalho de (FLORENTINO; GOLDSCHMIDT, 2017) é de $max\{O(|D| * |S| * |V|^2), O_C\}$. Considerando seu uso do histórico da rede, existe um limite em função da quantidade de métricas $|D|$, o número de estados na rede $|S|$ e o número de vértices na rede. Como seu método possibilita o uso de diferentes tipos de algoritmos de *clusterização*, a outra possibilidade é que a complexidade deste algoritmo, O_C, defina a complexidade geral do método.

É importante notar que a complexidade geral do método proposto como parte do presente trabalho, que é detalhada no Capítulo 4, não ultrapassa a complexidade dos métodos de (SOARES; PRUDÊNCIO, 2012a) e (FLORENTINO; GOLDSCHMIDT, 2017).

4 ABORDAGEM PROPOSTA

4.1 Considerações preliminares

Este capítulo descreve a abordagem proposta no presente trabalho. O método não supervisionado utilizado é composto de 3 etapas, conforme Figura 17 cujo detalhamento conceitual é apresentado na sequência.

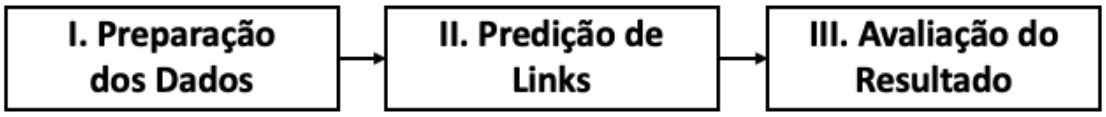


Figura 17 – Visão Geral do Método

Uma questão importante que merece destaque e desambiguação: o presente trabalho faz uso da abordagem não supervisionada na predição de links que, em sua essência original, não envolve nenhum tipo de aprendizado de fato. No entanto, no método proposto neste trabalho, de forma análoga ao utilizado por (FLORENTINO; GOLDSCHMIDT, 2017), foi introduzido um passo que inclui, por meio da clusterização de dados, uma tarefa de aprendizado não supervisionado (GAMA et al., 2011).

4.2 Descrição conceitual

4.2.1 Preparação dos Dados

A Etapa I é composta por 6 passos (conforme apresentado na Figura 18) e trata da preparação dos dados para a tarefa de predição de *links*. Nesta etapa, são obtidos os dados da rede a ser analisada e na sequência são gerados: o grafo G que contêm todas as informações da RSV e a partição em dois conjuntos, um de treinamento G_{Treino} e outro teste G_{Teste}, com base em configurações definidas pelo usuário. A partir do grafo G_{Treino}, são gerados os grafos que representam os múltiplos estados da rede e contém o histórico da rede no momento do surgimento de novas ligações, incluindo não somente informações topológicas da estrutura do grafo, mas também informações contextuais. Na sequência é

apresentado o detalhamento de cada um dos passos dessa etapa.

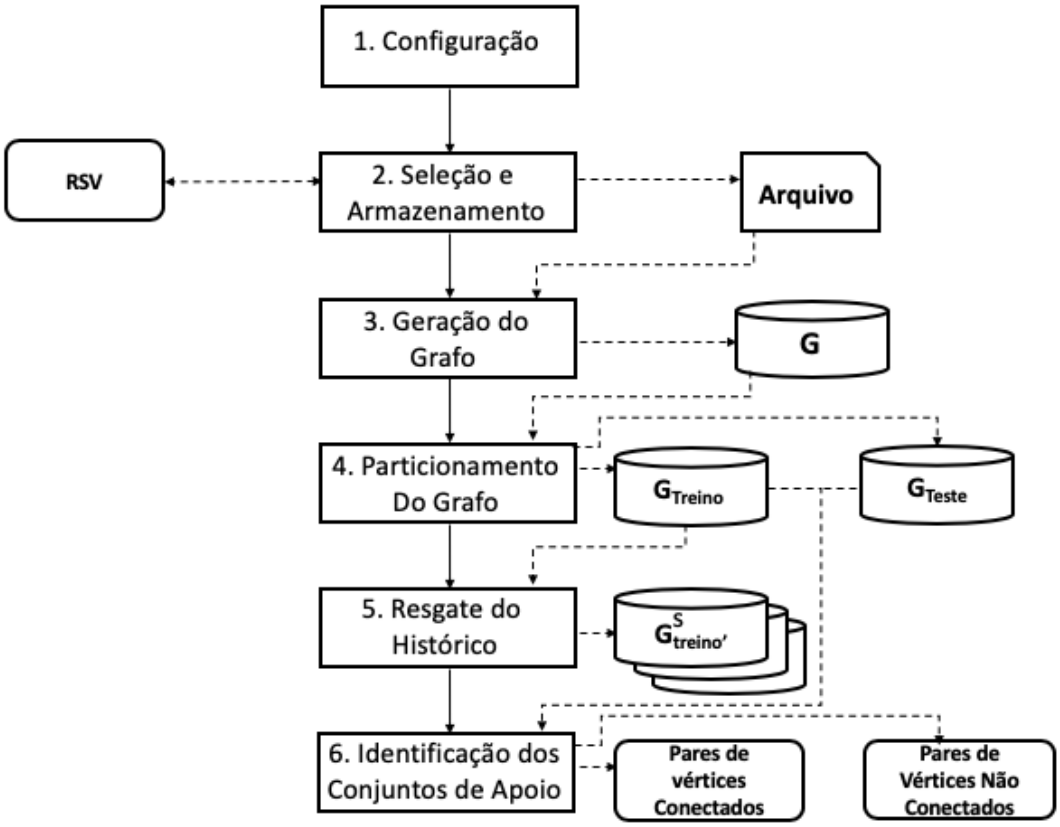


Figura 18 – Preparação dos Dados

Passo 1. Configuração

Para que seja possível realizar a preparação dos dados que serão utilizados como base para a tarefa de predição de links, é necessário um conjunto inicial de configurações. Essa atividade é tipicamente desempenhada por um analista de dados que possui experiência na RSV a ser analisada, assim como o entendimento de suas principais características. Em situações nas quais o analista não possui conhecimento da RSV que será analisada, a etapa de configuração pode ser precedida pela extração de amostras de dados da RSV. A extração dessas amostras tem como objetivo oferecer suporte (subsídios) para o analista.

Dentre estas configurações, merecem destaque os seguintes parâmetros que serão utilizados nos passos *2*, *3*, *4* e *5*: o período de observação da rede a ser considerado na análise da RSV $[t_i, t_f]$ e a unidade de tempo utilizada como critério para a definição dos múltiplos estados da rede, isto é, o formato das informações temporais existentes na rede. Por exemplo, pode-se considerar apenas os dados da RSV referentes aos anos de 1997 até 2000, ou seja $t_i = 1997$ e $t_f = 2000$, excluindo da análise dados anteriores ao limite inicial e posteriores ao limite final. A unidade de tempo é representada por um *timestamp* cujo

formato pode variar em função da unidade de tempo escolhida, por exemplo: *yyyy-mm-dd h:mm:ss* para informações com base em ano, mês, dia, hora, minuto e segundos, ou *yyyy* para somente anos.

Considerando a dinâmica particular de evolução da RSV, deve também ser definido o critério para seleção dos vértices observados como sendo os mais frequentes e ativos, utilizado no Passo *6*. Este critério leva em conta um número mínimo de arestas que indica a frequência de interações de um determinado vértice.

Base fundamental da tarefa de predição de links, é definido o conjunto D que contém as métricas de similaridade que serão utilizadas na Etapa II. Essas métricas podem ser topológicas, contextuais ou a combinação de ambas. Outra escolha relevante relacionada à Etapa II corresponde ao algoritmo de clusterização C que será aplicado, além de eventuais parâmetros específicos desse algoritmo.

Passo 2. Seleção e Armazenamento

Utilizando t_i e t_f, definidos no Passo *1*, os dados da RSV são lidos e armazenados em um arquivo. Esse processo se dá por meio de algum tipo de facilidade oferecida pela RSV (como por exemplo um serviço de integração de dados, oi API), com o qual são realizadas consultas das informações da RSV que serão representadas em vértices e arestas no passo seguinte. Este passo poderá ser omitido caso os referidos dados já estejam disponíveis em arquivo de texto.

Passo 3. Geração do Grafo

Com base nos dados obtidos da RSV é gerado um multigrafo homogêneo com atributos [1] $G = \left\langle V, E \right\rangle$, onde V é um conjunto de vértices v, E é um conjunto de arestas e que contêm pelo menos um atributo temporal ($e.t$) indicando o momento de surgimento de e, compreendido dentro do período entre $t_i \leq e.t \leq t_f$, definidos no Passo *1*.

Passo 4. Particionamento do Grafo

Neste passo, $G = \left\langle V, E \right\rangle$ é dividido nos subgrafos de treino $G_{Treino} = \left\langle V, E_{Treino} \right\rangle$ e de teste $G_{Teste} = \left\langle V, E_{Teste} \right\rangle$. O particionamento ocorre com base em t_c (também definido no Passo *1*) onde $t_i < t_c < t_f$. G_{Treino} e G_{Teste} contêm as arestas surgidas nos respectivos intervalos, ou seja $t_i \leq e.t \leq t_c$ para treino e $t_c < e.t \leq t_f$ para teste.

Passo 5. Resgate do Histórico

Este passo é responsável por obter de $G_{Treino}\left\langle V, E_{Treino} \right\rangle$ (ou seja, do subgrafo

[1] Um grafo G é dito: (a) homogêneo se, e somente se, G possui apenas um tipo de vértice e um tipo de aresta; e (b) possuir atributos se, e só se, G contiver atributos em seus vértices e/ou em suas arestas.

de treino) as informações topológicas e contextuais que descrevem os estados da rede nos momentos nos quais surgiram novas arestas entre pares de vértices nunca ligados anteriormente. Para que isso seja feito, é utilizado o formato da unidade de tempo (*timestamp*) definida pelo usuário que é aplicado ao atributo temporal ($e.t$) existente em cada aresta e, gerando um conjunto S de instantes de tempos s, no qual $S = \{t'/\exists e'(e' \in E_{Treino} \wedge e' = (u,v), u,v \in V \wedge e'.t = t' \wedge \nexists e(e \in E_{Treino} \wedge e = (u,v) \wedge e.t < e'.t))\}$. Assim, para cada $s \in S$, este passo obtém o subgrafo $G^s_{Treino}\left\langle V, E^s_{Treino}\right\rangle$, onde $\forall e \in E_{Treino}$ $(e.t \leq s) \rightarrow (e \in E^s_{Treino})$. Este conjunto de subgrafos retrata o histórico evolutivo da rede $H = (G^{s_1}_{Treino}, G^{s_2}_{Treino}, ..., G^{s_m}_{Treino})$, onde $s_i \in S$ para $i = 1, 2, ..., m$ tal que $s_1 < s_2 < ... < s_m$, sendo m o número de elementos em S.

Passo 6. Identificação dos Conjuntos de Apoio

De forma a dar prosseguimento à execução do processo, são criados os conjuntos de apoio necessários para os próximos passos. Procede-se então inicialmente para a identificação do conjunto $Core$. Esta atividade consiste em determinar o conjunto dos vértices da RSV que tenham se comportado de forma ativa na rede, ou seja, vértices cujo número de interações com outros tenha uma certa frequência mínima. Tal procedimento visa assegurar que vértices espúrios, ou com poucas interações, não sejam levados em conta no processo preditivo, evitando distorções na avaliação dos resultados (LIBEN-NOWELL; KLEINBERG, 2007a). Para tanto, aplica-se um filtro nos grafos de treino e teste, de forma a obter os vértices que possuem pelo menos ϕ arestas em G_{Treino} e pelo menos ψ arestas em G_{Teste} criando o chamado conjunto $Core$. Os parâmetros ϕ e ψ são definidos pelo analista de dados no passo de configuração (Passo 1) e podem variar conforme a média das interações observadas em cada RSV. As arestas que surgiram entre elementos do conjunto $Core$ em G_{Treino} são denominadas E_{Old}, enquanto aquelas (ainda de elementos do $Core$) surgidas em G_{Teste} são chamadas de E_{New}.

O conjunto de pares de vértices conectados a serem considerados pelos próximos passos do método ora descrito é obtido com base nos vértices que originaram as ligações de E_{Old} existentes em G_{Treino}. Já os pares de vértices não conectados, são aqueles que não estão conectados entre si em G_{Treino} $((Core \times Core) - E_{old})$. Esse conjunto contém os pares de vértices cujo potencial de ligação deve ser avaliado ao final do processo de predição.

4.2.2 Predição de Links

Na segunda etapa (Figura 19), a predição de ligações é de fato executada. Aqui são calculadas métricas de similaridade que serão utilizadas por um algoritmo de *clusterização*. O resultado do processo de *clusterização* é por sua vez utilizado para representar os padrões de similaridade da rede analisada. Ao calcular estas mesmas métricas para pares de vértices

que não estão conectados, busca-se associar estes elementos aos *clusters* obtidos com os vértices conectados visando identificar aqueles elementos mais passíveis de estabelecer uma nova ligação. A seguir são descritos os passos que implementam esta etapa.

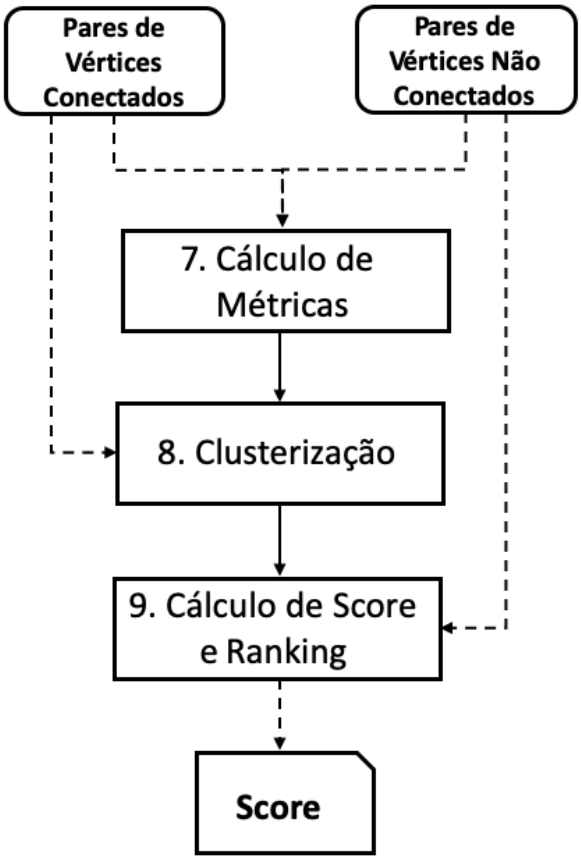


Figura 19 – Predição de Links

Passo 7. Cálculo de Métricas

O cálculo de métricas de similaridade topológicas e contextuais se dá em duas fases: a agregação e representação dos dados e a verificação da similaridade entre os pares de vértices conectados e não conectados, conforme descrito na Figura 20.

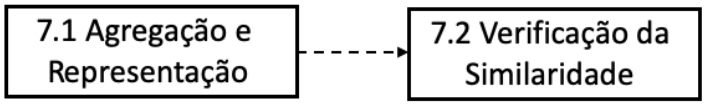


Figura 20 – Cálculo de Métricas

7.1 Agregação e Representação

Independente do tipo de informação (topológico ou contextual) disponível nos vértices e arestas que representam a RSV e dos dados estarem em formato estruturado ou não estruturado, pode ser necessário realizar a agregação destes dados para que seja possível

efetuar o cálculo da similaridade entre pares de vértices. Por exemplo, pode ser necessário combinar todos os vizinhos de um determinado vértice e compará-los aos vizinhos de um segundo vértice (informação topológica em formato estruturado) visando identificar os vizinhos em comum entre esses vértices. De forma semelhante, pode-se combinar todas os títulos das publicações de um autor em uma rede de coautoria e comparar seu resultado com o de outro autor (informação contextual em formato não estruturado) de forma a verificar a similaridade entre os assuntos tratados por esses autores. Essa combinação é realizada com o uso de funções de agregação (introduzidas no Capítulo 2), cujo operador indica a soma da quantidade de elementos, a união entre palavras-chave, entre diversos outros exemplos possíveis.

No caso de dados não estruturados, além da aplicação de uma função de agregação, é necessário representá-los de forma a tornar possível o cálculo de sua similaridade. Dentre as opções de representação (introduzidas no Capítulo 2), têm-se as representações *bag-of-words*, *TF-IDF*, *Doc2Vec*, entre outras. No caso dos dados estruturados também existe algum tipo de representação dos dados, mas esta é trivial, como o dado em formato numérico ou textual.

Vale notar que a relação de ordem entre as atividades de agregação dos dados e a sua representação pode variar. Ao utilizar a representação *Doc2Vec*, por exemplo, agregar dados não estruturados antes de inferir sua relação com um modelo provocaria perda semântica dos dados. Neste caso portanto, a agregação ocorre após a inferência, visando combinar os vetores que representam a comparação desses dados com o modelo. Em contrapartida, para usar uma estrutura *BoW*, é necessário agrupar os dados antecipadamente.

7.2 Verificação da Similaridade

Sobre todos os elementos dos conjuntos de pares de vértices conectados e não conectados são aplicadas cada uma das métricas de similaridade topológicas e contextuais, do conjunto de métricas D, definidas pelo analista de dados. Essas métricas estão ligadas diretamente com a representação dos dados utilizada. No caso dos pares de vértices ligados, seu resultado descreve de forma sumarizada as informações sobre a topologia e também sobre o domínio contextual do grafo no momento em que surgiram as ligações entre seus vértices. Caracteriza-se, desse modo, o evento de surgimento de cada aresta e. Como resultado, é produzido um conjunto de elementos no espaço q-dimensional definido a partir do número de métricas de similaridade topológicas e contextuais definidas em D. O resultado é o conjunto $H_{(d_1,d_2,...,d_q)}$, ou seja, uma representação simplificada do histórico de evolução da rede $H = (G^{s_1}_{Treino}, G^{s_2}_{Treino}, ..., G^{s_m}_{Treino})$. Este conjunto contém os registros de dados extraídos a partir de G^{s}_{treino} $\forall s \in S$ e que retratam todos os momentos em que surgiram arestas durante o intervalo $[t_i, t_c]$.

De forma similar, estas mesmas métricas de similaridade topológicas e contextuais do conjunto de métricas D são aplicadas aos elementos do conjunto de pares de vértices

não conectados, gerando o conjunto $P_{(d_1,d_2,...,d_q)}$. Esse conjunto, no entanto, ao invés de representar o histórico da rede, objetiva identificar o grau de compatibilidade entre os elementos dos pares de nós não ligados, cujo potencial de ligação será posteriormente calculado.

Passo 8. *Clusterização*

Neste passo, aplica-se o algoritmo de *clusterização* de dados C sobre o conjunto resultante do cálculo de métricas para os pares de vértices conectados $H_{(d_1,d_2,...,d_q)}$, gerando um agrupamento com k *clusters* $\{c_1,\ c_2,\ ...,\ c_k\}$. Cada *cluster* c_j delimita uma região no espaço q-dimensional caracterizada pelo surgimento de arestas durante a evolução de G_{Treino}. Quanto mais densa e populosa for essa região, mais representativo é o *cluster* correspondente e maior é a expectativa de novas arestas que surjam no grafo possam ser associadas a ele. O resultado do processo está exemplificado na Figura 21 (b).

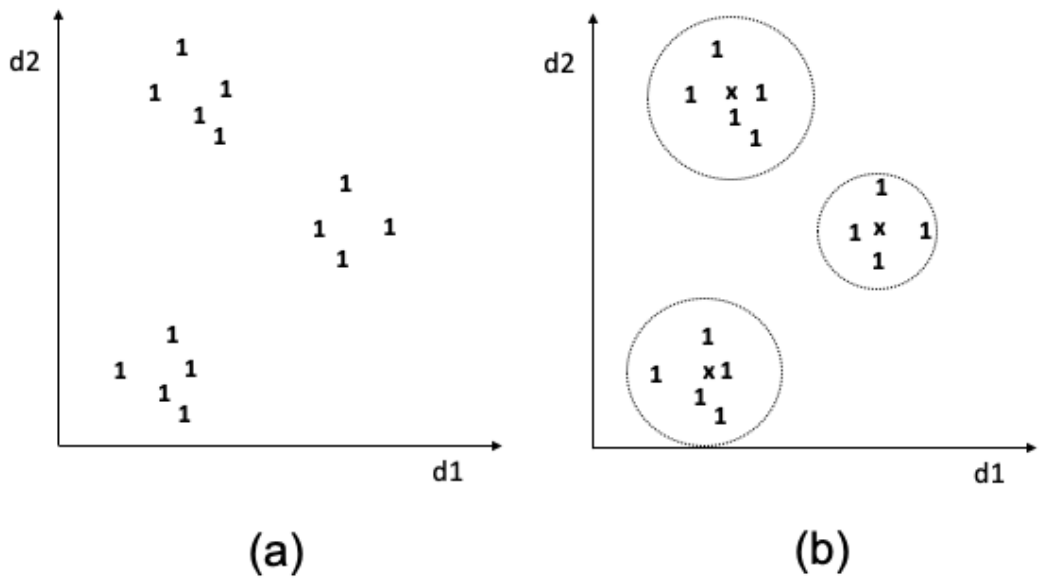


Figura 21 – (a) Representação dos pares de vértices conectados (elementos de $H_{(d_1,d_2,...,d_q)}$ representados como "1") para duas métricas arbitrárias (d_1, d_2) representadas no espaço 2-dimensional; (b) Exemplo de resultado da *clusterização* de pares de vértices conectados e os centroides (x) de cada um destes *clusters*.

Passo 9. Cálculo de Score e Ranking

Neste passo é calculado o *score* de cada par (u, v) de vértices não conectados representado como p em $P_{(d_1,d_2,...,d_q)}$, que é obtido a partir da Equação 4.1. Desta forma, este cálculo resulta no score associado a cada par de vértices (u, v) avaliado em relação ao

cluster c ao qual o referido par pertence.

$$Score(u,v) = Cos(p, Centroide(c)) + \frac{|H_c|}{\max\limits_{1 \leq j \leq k}(|H_{c_j}|)} \cdot \frac{|H_c|}{|P_c|} \tag{4.1}$$

Utiliza-se como entrada, o registro de dados p referente ao par (u,v) armazenado em $P_{(d_1,d_2,...,d_q)}$, topologicamente e contextualmente representados por meio do resultado do cálculo das métricas do conjunto D e os c_j $(1 \leq j \leq k)$ *clusters* gerados na etapa de clusterização resultante da análise do histórico da rede. Para cada elemento p representado pelos resultados da aplicação das métricas do conjunto de métricas D ao par de vértices não conectados (u,v) correspondentes, verifica-se em qual *cluster* esse p deve ser alocado (Equação 4.2). Para tanto, a distância de p ao centroide de cada *cluster* c_j $(1 \leq j \leq k)$ é calculada. O ponto p é alocado ao *cluster* c cuja distância de p ao centroide é a menor dentre todos os *clusters*. Dessa forma, obtém-se o *cluster* que contém elementos semelhantes a p.

$$c = \underset{1 \leq j \leq k}{\operatorname{argmin}}\, Cos(p, Centroide(c_j)) \tag{4.2}$$

Nos exemplos da Figura 22, tem-se cada um dos pontos $p \in P_{(d_1,d_2,...,d_q)}$ representados como "0"e uma seta que aponta para o centroide do *cluster* mais próximo do par de vértices avaliado representado por p.

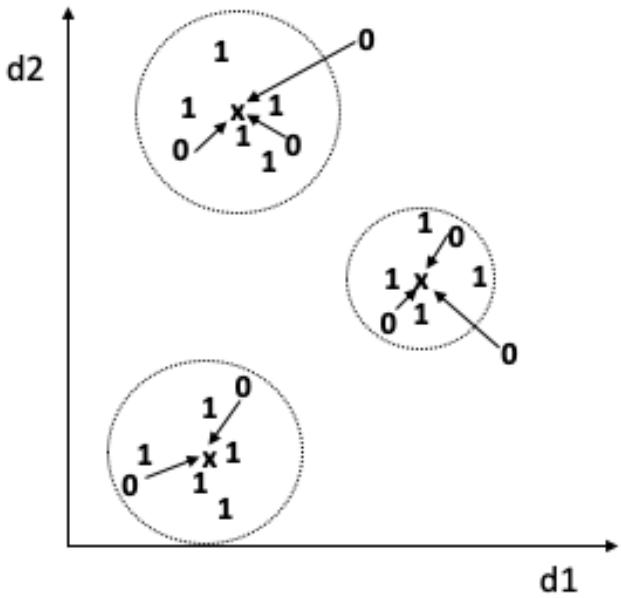


Figura 22 – Exemplo de alocação de pares de vértices não conectados (elementos de $P_{(d_1,d_2,...,d_q)}$ representados como "0") aos *clusters*.

A seguir é identificada a frequência com que são encontrados pares de nós ligados do conjunto que representa o histórico da rede $H_{(d_1,d_2,...,d_q)}$ no *cluster* c em relação à maior quantidade total de pares de nós ligados entre todos os *clusters*. Esse resultado é ponderado pela quantidade de pares de vértices ligados de $H_{(d_1,d_2,...,d_q)}$ em c divididos pela quantidade de pares de vértices não ligados do conjunto $P_{(d_1,d_2,...,d_q)}$, ambos identificados dentro do *cluster* c ao qual o par de nós avaliado mais se assemelha. Este processo visa dar maior score aos pares em *clusters* com muitos vértices ligados e poucos vértices não ligados.

Finalmente, a lista com todos os valores dos *scores* calculados para todos os pares de nós não conectados é então ordenada de forma decrescente, gerando a lista L_r. Assume-se que quanto maior for o *score*, maior a probabilidade dos vértices correspondentes se conectarem no futuro.

4.2.3 Avaliação do Resultado

Na terceira etapa (Figura 23), é realizada a avaliação dos resultados com o objetivo de identificar quais, dos pares de nós não ligados apontados como mais prováveis de se conectarem, de fato se ligaram. A avaliação do resultado é realizada com base em uma medida de avaliação. Esta etapa gera um relatório final de todo o processo.

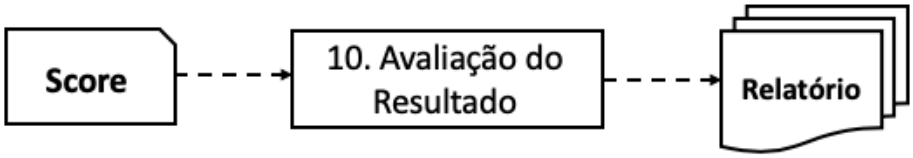


Figura 23 – Avaliação do Resultado

Passo 10. Avaliação do Resultado

A partir da lista de pares de vértices ordenados produzida na etapa anterior, o método proposto faz uso da abordagem de predição de links não supervisionada, sendo assim, seleciona-se os N primeiros elementos (os Top_N) a fim de constatar quantos deles, de fato, se conectaram em G_{Teste}. N é encontrado por meio da Equação 4.3.

$$N = |E_{New} \cap (Core \times Core)| \tag{4.3}$$

Onde E_{New} é composto por arestas $e = (u, v)$ que pertençam a G_{Teste} e seus vértices u e v pertençam ao conjunto *Core*. Posteriormente, são analisadas as arestas pertencentes a E_{New} e os primeiros N pares ordenados de $Core \times Core$, identificando $E_{Correct}$, ou seja, o conjunto de predições corretas.

Dentre as medidas de avaliação mais comuns na abordagem não supervisionada, destacam-se o cálculo do *improvement* (LIBEN-NOWELL; KLEINBERG, 2007a), a área sob a curva e a precisão (LU; ZHOU, 2011). A Figura 24 apresenta em forma de conjuntos os principais elementos utilizados por essas medidas.

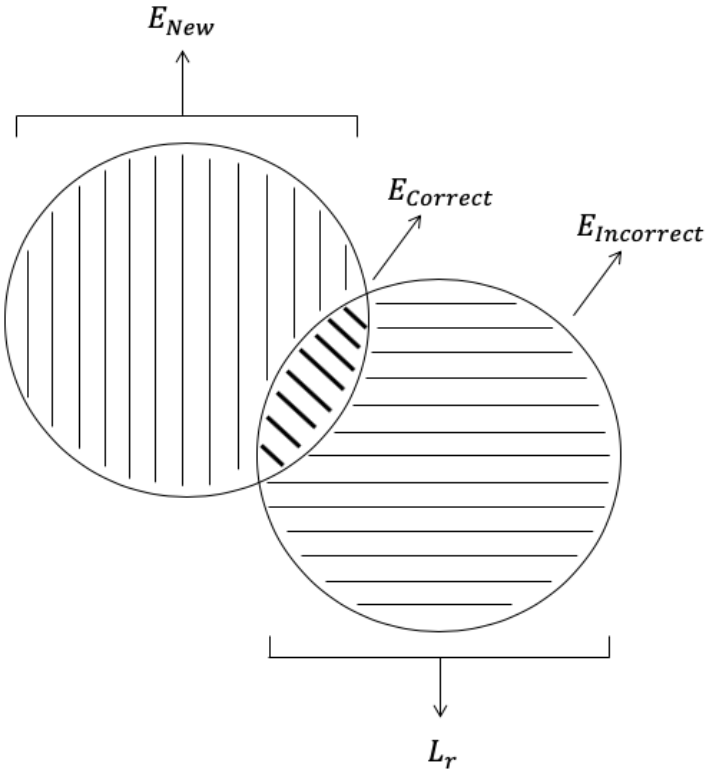


Figura 24 – Representação da lista ordenada com as predições L_r, das novas ligações de vértices do *Core* que surgiram na rede E_{New}, além das predições corretas $E_{Correct}$ e incorretas $E_{Incorrect}$, ambas constantes da lista L_r e avaliadas em E_{New}.

Para a avaliação do *improvement*, inicialmente, calcula-se o preditor randômico (PR), que se trata do número de possíveis predições corretas (Equação 4.4).

$$PR = \frac{|E_{new}|}{|(Core \times Core)| - |E_{old}|} \tag{4.4}$$

Posteriormente, calcula-se o *improvement* que expressa em quantas vezes o desempenho do método avaliado é superior ao desempenho do preditor randômico PR, conforme demonstrado na Equação 4.5.

$$Improvement = \frac{|E_{Correct}|/|E_{New}|}{PR} \tag{4.5}$$

A medida de avaliação área sob a curva (*area under the receiver operating characteristic curve* ou *AUC*) avalia o desempenho de um método em função da probabilidade

de que uma ligação que se concretizou tenha um *score* maior do que uma ligação que não se concretizou. Sua fórmula é apresentada na Equação 4.6. Para seu cálculo é necessário avaliar quantas vezes o *score* de uma predição correta é maior do que o *score* de uma predição incorreta $n' = |\{e \in E_{Correct} \mid score(e) > score(\hat{e}) \forall \hat{e} \in E_{Incorrect}\}|$ e também, quantas vezes o *score* de ambas as predições (corretas e incorretas) foi igual, ou seja $n'' = |\{e \in E_{Correct} \mid score(e) = score(\hat{e}) \forall \hat{e} \in E_{Incorrect}\}|$.

$$AUC = \frac{n' + 0.5n''}{n} \tag{4.6}$$

A avaliação da precisão é definida como a razão entre o número de predições corretas e o número total de predições (corretas e incorretas). A fórmula da precisão é apresentada na Equação 4.7.

$$Precision = \frac{|E_{Correct}|}{|E_{Correct}| + |E_{Incorrect}|} \tag{4.7}$$

4.3 Análise da complexidade

A análise da complexidade de um algoritmo tem por objetivo avaliar a sua eficiência em termos do tempo de sua execução e dos recursos de memória utilizados. Esse tipo de estudo tem como fundamento o número de operações consideradas relevantes que são realizadas por um algoritmo. O tempo de execução de um algoritmo pode variar em função da entrada de dados, ou parâmetros, utilizados por determinada função para que esta execute. Com foco no objetivo específico do presente trabalho, optou-se por considerar a complexidade do pior caso, ou seja, o maior número de operações utilizado para processar uma entrada qualquer de tamanho n. Conforme a estrutura geral de organização do método proposto, apresentada na Figura 17, são aqui representados os algoritmos mais importantes para a execução do protótipo desenvolvido. Ao final da seção, a complexidade geral do método é discutida.

Etapa I - Preparação dos Dados:

A Etapa I conta com um passo inicial de leitura das configurações para execução do processo que tem complexidade constante $O(1)$. Dada sua representação trivial, optou-se por não descrever o algoritmo referente a esse passo.

A complexidade do Algoritmo 2 tem relação com dois fatores. Em primeiro lugar, o período de tempo de t_i até t_f que será utilizado como critério para obtenção dos dados. Além desse fator, tem-se a quantidade de registros de cada ano (considera-se n o número de indivíduos da rede e o número de registros limitado por $n.(n-1)/2$). Multiplicando estes

dois fatores se obtêm como resultado uma complexidade quadrática $O(n^2)$, ou simplificando $O(|V| + |E|)$ (vértices e arestas, como por exemplo autores e publicações em um *dataset* de coautoria).

Algoritmo 2: OBTER DADOS DA RSV

Entrada: Rede Social Virtual RSV, quantidade máxima de registros por consulta *quantidadeMáximaRegistros*, limite inicial do período de tempo t_i e limite final do período de tempo t_f

Saída: Lista de *Registros*

```
1 início
2     para cada unidade de tempo > t_i e < t_f faça
3         Consultar a quantidade total de registros disponíveis na RSV e atribuir a
           totalRegistrosAno
4         enquanto tamanho da lista de Registros < totalRegistrosAno faça
5             Obter os registros de dados da RSV limitados por
               quantidadeMáximaRegistros e atribuir a lista de Registros
6         fim
7     fim
8 fim
9 retorna
```

A complexidade de geração do grafo $G = \langle V, E \rangle$ relacionada ao Algoritmo 3 está limitada pelo número de registros obtidos da RSV, que serão transformados em vértices e arestas de G, resultando em $O(|V| + |E|)$.

Algoritmo 3: Gerar grafo contendo os vértices e arestas da *RSV*

Entrada: Lista *Registros* de registros da *RSV*

Saída: Arquivo com vértices e arestas do grafo G

```
1  início
2      para cada registro da lista de Registros faça
3          Identificar os indivíduos relacionados com cada registro e atribuir à lista de
             Indivíduos.
4      fim
5      se tamanho da lista de Indivíduos > 1 então
6          Gerar todas as combinações possíveis entre os indivíduos e atribuir à lista
             CombinaçãoIndivíduos
7          para cada combinação da lista de CombinaçãoIndivíduos faça
8              Adicionar uma nova aresta e ao grafo G na qual os elementos da combinação
                 serão os vértices u e v da aresta e
9          fim
10     fim
11 fim
12 retorna
```

Nos Algoritmos 4 e 5, são realizadas iterações sobre todos os elementos do grafo G de maneira a selecionar aqueles que devem fazer parte dos grafos de treino G_{Treino} e teste G_{Teste}. Assim sendo, a complexidade é dada pelo número total de vértices e arestas da rede, com uma complexidade de $O(|V| + |E|)$ que é o limite superior para os pares formados entre os vértices.

Algoritmo 4: Gerar grafo de treino G_{Treino}

Entrada: Arquivo com vértices e arestas do grafo G, limite de corte t_c do período de tempo

Saída: Arquivo com vértices e arestas do grafo G_{Treino}

```
1  início
2      Criar uma cópia G_Treino do grafo G
3      para cada aresta e de G_Treino faça
4          se o atributo t da aresta e > t_c então
5              Remover a aresta e.
6          fim
7      fim
8      para cada vértice v de G_Treino faça
9          se o vértice v não contém nenhuma aresta (isolado) então
10             Remover o vértice v
11         fim
12     fim
13 fim
14 retorna
```

Algoritmo 5: Gerar grafo de teste G_{Teste}

Entrada: Arquivo com vértices e arestas do grafo G, limite de corte t_c do período de tempo

Saída: Arquivo com vértices e arestas do grafo G_{Teste}

```
1  início
2      Criar uma cópia G_Teste do grafo G
3      para cada aresta e de G_Teste faça
4          se o atributo t da aresta e < t_c então
5              Remover a aresta e.
6          fim
7      fim
8      para cada vértice v de G_Teste faça
9          se o vértice v não contém nenhuma aresta (isolado) então
10             Remover o vértice v
11         fim
12     fim
13 fim
14 retorna
```

Embora dependa primariamente apenas das arestas e vértices existentes no grafo de G_{Treino}, o Algoritmo 6 possui relação direta com o *timestamp* utilizado como critério

para seleção dos estados da rede e depende do atributo temporal t disponível em cada aresta e: quanto mais fina for a granularidade deste atributo temporal, maior o número de estados a serem criados. Considerando um intervalo entre os anos de 2001 e 2019, caso o formato *yyyy-mm-dd h:mm:ss* fosse utilizado, teriam-se no máximo 599.184,000 estados. Com um formato *yyyy* são obtidos no máximo 19 estados, uma evidente diferença entre o número de estados possíveis. A complexidade é $O(|S| * |V|^2)$ na qual $|S|$ é o número de estados obtidos a partir do formato do *timestamp* utilizado o número $|V|$ de vértices da rede.

Algoritmo 6: Resgatar histórico evolutivo da rede

Entrada: Arquivo com vértices e arestas do grafo G_{Treino}, unidade de tempo *timestamp*
Saída: Lista de arquivos contendo os grafos G^s_{Treino}

```
1  início
2      Lista de instantes de tempo S = ∅
3      para cada aresta e de G_Treino faça
4          S recebe o atributo temporal t de cada aresta e com base no formato da unidade
             de tempo timestamp
5      fim
6      para cada instante de tempo s ∈ S faça
7          G^s_Treino recebe a cópia (ou subgrafo) de G_Treino contendo todos elementos
             criados até o instante de tempo s
8      fim
9  fim
10 retorna
```

A complexidade do Algoritmo 7 é limitada pelo número de arestas do período de treinamento, tendo como complexidade $O(|E|)$.

Algoritmo 7: Identificar pares de nós ligados

Entrada: Arquivo com vértices e arestas do grafo G_{Treino}
Saída: Lista de pares de vértices conectados $ParesConectados$

```
1  início
2      para cada aresta e de G_Treino faça
3          ParesConectados recebe o par de vértices da aresta e
4      fim
5  fim
6  retorna
```

No caso do Algoritmo 8, a complexidade é da ordem de $O(2|E|) = O(|E|)$, considerando que a soma dos graus para todos vértices do grafo é igual a duas vezes a soma de

todos os graus dos vértices.

Algoritmo 8: Identificar conjunto Core

Entrada: Arquivo com vértices e arestas do grafo G_{Treino}, número mínimo de arestas em treino ϕ, Arquivo com vértices e arestas do grafo G_{Teste}, número mínimo de arestas em testes ψ

Saída: Lista de vértices do conjunto $Core$

```
1  início
2      para cada vértice u de G_Treino faça
3          Identificar as arestas e ligadas ao vértice u e atribuir a grauVerticeTreino
4          se grauVerticeTreino >= φ então
5              para cada vértice v de G_Teste faça
6                  Identificar as arestas e ligadas ao vértice v e atribuir a grauVerticeTeste
7                  se grauVerticeTeste >= ψ então
8                      Incluir vértice na lista Core
9                      grauVerticeTreino = 0
10                     grauVerticeTeste = 0
11                 fim
12             fim
13         fim
14     fim
15 fim
16 retorna
```

O Algoritmo 9 tem como base os elementos do conjunto $Core$ que são utilizados para gerar uma lista com todas as combinações possíveis, o número de arestas em treino e uma operação de diferença entre conjuntos, resultando em $O(|Core|^2 + |E| + |E_{Treino}|)$.

Algoritmo 9: Identificar pares de nós não ligados

Entrada: Arquivo com vértices e arestas do grafo G_{Treino}, Lista de vértices do conjunto $Core$

Saída: Lista de pares de vértices não conectados $ParesNaoConectados$

1 **início**
2 Gerar uma lista com todas as combinações possíveis entre os elementos do conjunto $Core$ e atribui a $Core \times Core$.
3 **para cada** *aresta e de* G_{Treino} **faça**
4 E_{Old} recebe o par de vértices da aresta e
5 **fim**
6 $ParesNaoConectados$ recebe a diferença entre $Core \times Core$ e E_{Old}
7 **fim**
8 **retorna**

Etapa II - Predição de Links

O Algoritmo 10 é o primeiro da etapa de predição de *links* e está limitado pelos elementos da lista de pares de nós ligados (no pior caso, quando todos os vértices estão ligados, temos $|Core|^2$) e a lista com os elementos do conjunto D contendo as métricas que serão executadas de complexidade é $O(|Core|^2 * |D|)$.

Em relação ao cálculo de métricas, há dois elementos a serem considerados. Em primeiro lugar tem-se a função de agregação e a representação utilizadas. As funções de agregação são guiadas pelo quantidade de informações topológicas e contextuais que serão executadas. No caso das informações topológicas os limites estão relacionados ao número de vértices, $|V|$, e ao número de arestas, $|E|$, existentes no grafo. A agregação de informações contextuais em formato de texto, está limitada pelo número de palavras existentes no atributo escolhido que na maioria dos casos deve estar limitado a uma complexidade linear. De maneira similar, as representações *bag-of-words* e *TF-IDF* também estão limitadas segundo os mesmos critérios. Um caso particular de complexidade de representação está relacionada ao *Doc2Vec*. Segundo (PARIKH et al., 2018), sua complexidade varia quadraticamente em função da quantidade de palavras, $O(|X|^2)$. Em resumo, na agregação e representação tem-se $O(|V| + |E| + |X|^2)$, ou somente $O(|X|^2)$.

O segundo elemento relacionado ao cálculo de métricas é a verificação de similaridade a partir de uma métrica voltada para esse fim. Tanto as métricas topológicas quanto as contextuais estão limitadas pelo número de vértices $|V|$ e arestas $|E|$ no grafo. Nos casos de métricas contextuais que utilizam atributos do grafo, pode haver variação em função da quantidade de cada atributo utilizado, que tipicamente deve ser linear. A complexidade resultante é portando O($|V|$ + $|E|$).

Finalmente, a complexidade geral do Algoritmo 10 é $O(|Core|^2 * |D| * O(|X|^2) + O(|V| + |E|))$.

Algoritmo 10: Calcular métricas - Pares de nós ligados no $Core \times Core$

Entrada: Lista de arquivos contendo os grafos G^s_{Treino}, Lista de métricas do conjunto D, Lista de pares de nós ligados $ParesConectados$

Saída: Lista contendo o resultado de cada uma das métricas para cada par de nós ligados $H_{(d_1,d_2,...,d_q)}$

```
1  início
2      para cada par (u, v) de ParesConectados faça
3          Carrega G^s_Treino correspondente a (u, v)
4          para cada métrica d ∈ D faça
5              Calcular d_i(u, v)
6          fim
7          Determinar h_(d_1,d_2,...,d_q) de (u, v)
8      fim
9  fim
10 retorna
```

A diferença entre o Algoritmo 11 e o Algoritmo 10 está nos elementos da lista de pares de nós não ligados ($|V|^2$ no pior caso, quando nenhum vértice está ligado $O(|V|^2 * |D|)$ ou $O(|V|^2 * |D| * O(|X|^2) + O(|V| + |E|))$.

Algoritmo 11: Calcular métricas - Pares de nós não ligados

Entrada: Arquivos contendo o grafo G_{Treino}, Lista de métricas do conjunto D, Lista de pares de nós ligados $ParesNaoConectados$

Saída: Lista contendo o resultado de cada uma das métricas para cada par de nós ligados $H_{(d_1,d_2,...,d_q)}$

```
1  início
2      para cada par (u, v) de ParesNaoConectados faça
3          Carrega G_Treino
4          para cada métrica d ∈ D faça
5              Calcular d_i(u, v)
6          fim
7          Determinar P_(d_1,d_2,...,d_q) de (u, v)
8      fim
9  fim
10 retorna
```

A complexidade do Algoritmo 12 depende do algoritmo de *clusterização* utilizado.

Um exemplo desse tipo de algoritmo é o *k-means* que foi utilizado nos experimentos descritos no Capítulo 5. No caso desse algoritmo (*k-means*), tem-se uma complexidade linear (JAIN; MURTY; FLYNN, 1999; CELEBI; KINGRAVI; VELA, 2013) que é limitada pela quantidade de dimensões, o número de *clusters* e pelo número de elementos a serem *clusterizados*, item que define sua complexidade, ou ($O(|V|)$ no caso do método proposto.

Algoritmo 12: Clusterizar pares de nós ligados

Entrada: Lista contendo o resultado de cada uma das métricas para cada par de nós ligados $H_{(d_1,d_2,...,d_q)}$, número de *clusters* k

Saída: Lista $ParesConectadosClusterizados$ contendo os pares de nós ligados, o cluster ao qual cada par pertence e seu centroide

```
1 início
2 |  Clusterizar dados
3 fim
4 retorna
```

O Algoritmo 13 está limitado pelo número de pares de nós ligados (resultado da *clusterização*) e a lista com o resultado das métricas para os pares de nós não ligados, tendo uma complexidade $O(|V| + |E|)$.

Algoritmo 13: Calcular score e criar ranking

Entrada: Lista de pares de nós ligados $ParesConectadosClusterizados$, Lista com o resultado das métricas de pares não ligados $P(d_1, d_2, ..., d_q)$, Lista com os centroides de cada cluster $CentroidesClusters$, Lista de Pares de nós não ligados por cluster $ParesNaoConectadosPorCluster$

Saída: Lista em ordem decrescente baseada no score obtido para cada par de nós não ligados $RankingScore$

1 **início**
2 **para cada** *par* (u, v) *em* $ParesConectadosClusterizados$ **faça**
3 Atribuir a $CentroidesClusters$ o centroide de cada cluster existente na lista de pares de nós ligados por cluster
4 **fim**
5 **para cada** *par* $[u, v]$ *em* $P(d_1, d_2, ..., d_q)$ **faça**
6 **para cada** *centroide* k *em* $CentroidesClusters$ **faça**
7 Verifica a menor distância do par $[u, v]$ até $ParesConectadosPorCluster$ no centroide k e atribui a $ClusterPar$
8 **fim**
9 $|TotalParesCluster| = |ParesConectadosClusterizados| + |ParesNaoConectadosPorCluster|$ referentes ao cluster $ClusterPar$
10 $TotalParesTodosClusters$ é obtido através da contagem de elementos $ParesConectadosClusterizados$ e $ParesNaoConectadosPorCluster$
11 $Frequencia$ é obtida por meio da divisão de $TotalParesCluster$ por $TotalParesTodosClusters$
12 $TotalParesConectadosCluster$ é obtido através da contagem de elementos $ParesConectadosClusterizados$ referentes ao cluster $ClusterPar$
13 $TotalParesNaoConectadosCluster$ é obtido através da contagem de elementos $ParesNaoConectadosPorCluster$ referentes ao cluster $ClusterPar$
14 Atribui a $ScoreCluster$ a Multiplicação de $Frequencia$ pelo resultado de $TotalParesConectadosCluster$ divididos por $TotalParesNaoConectadosCluster$
15 Calcula a similaridade dos resultado das métricas do (u, v) em relação co centroide de $ClusterPar$ e atribui a $ScorePar$
16 Atribui a $RankingScore$ a soma de $ScoreCluster$ e $ScorePar$
17 **fim**
18 Ordena $RankingScore$
19 **fim**
20 **retorna**

Etapa III - Avaliação do Resultado

O Algoritmo 14 de avaliação do resultado é limitado pela lista ranqueada obtida

no Algoritmo 13, as arestas do grafo de treino e de teste. A operação de interseção entre a lista de predições corretas e a lista de ligações criadas em treino guia a complexidade para $O(|V| + |E|)$.

Algoritmo 14: Avaliação do Resultado

Entrada: Lista em ordem decrescente baseada no score obtido para cada par de nós não ligados *RankingScore*, Um grafo G_{Teste}, Lista de vértices do conjunto *Core*, Um grafo G_{Treino}

Saída: Fator de melhoria *Improvement* em relação a um dado preditor randômico

1 **início**
2 **para cada** *aresta e de* G_{Teste} **faça**
3 $E_{New} = E_{New} \cup e$
4 **fim**
5 $TamanhoTop_N = |E_{New} \cap Core|$
6 **enquanto** *tamanho da lista de RankingScore* $<= TamanhoTop_N$ **faça**
7 $Top_N = Top_N \cup e$
8 Top_N recebe elemento de *RankingScore*
9 **fim**
10 $E_{Correct} = E_{New} \cap Top_N$
11 **para cada** *aresta e de* G_{Treino} **faça**
12 $E_{Old} = E_{Old} \cup e$
13 **fim**
14 $PreditorRandomico$ recebe cardinalidade de E_{New} divididos por: cardinalidade de *Core* tendo subtraído a cardinalidade de E_{Old}
15 *Improvement* recebe cardinalidade de $E_{Correct}$ dividido por: Cardinalidade de E_{New} dovidivo pela cardinalidade do $PreditorRandomico$
16 **fim**
17 **retorna**

A complexidade de pior caso geral da abordagem proposta é $max\{O(|S|*|V|^2), O(|V|^2 * |D| * O(|X|^2) + O(|V| + |E|)), O(|V|)\}$ relacionada aos algoritmos 6 (Resgatar histórico evolutivo da rede), 11 (Calcular métricas - Pares de nós não ligados) e 12 (Clusterizar pares de nós ligados).

É importante observar que ao usar métricas de similaridade contextual, o tamanho do texto obtido do *dataset* vai influenciar a complexidade do algoritmo de cálculo das métricas (Algoritmo 11). De forma similar, a granularidade do *timestamp* influenciará o algoritmo de resgate do histórico evolutivo da rede (Algoritmo 6).

Para a definição da complexidade apresentada, a representação de dados não estruturados considerada foi *Doc2Vec* (relacionada ao Algoritmo 11) e o algoritmo de

clusterização k-means (relacionado ao Algoritmo 12).

Na tabela 2 são apresentadas as complexidades de pior caso de todos os algoritmos, com destaque para os algoritmos de que guiam a complexidade geral do método.

Etapa	**Passo**	**Algoritmo**	**Complexidade**
I. Preparação dos Dados	1. Configuração		$O(1)$
	2. Seleção e Armazenamento	Obter dados RSV	$O(\|V\| + \|E\|)$
	3. Geração do Grafo	Gerar grafo contendo os registros da RSV	$O(\|V\| + \|E\|)$
	4. Particionamento do Grafo	Gerar grafo de treino G_{Treino}	$O(\|V\| + \|E\|)$
		Gerar grafo de treino G_{Teste}	$O(\|V\| + \|E\|)$
	5. Resgate do Histórico	**Resgatar histórico evolutivo da rede**	$\boldsymbol{O(\|S\| * \|V\|^2)}$
	6. Identificação dos Conjuntos de Apoio	Identificar pares de nós ligados	$O(\|E\|)$
		Identificar conjunto *Core*	$O(\|E\|)$
		Identificar pares de nós não ligados	$O(\|Core\|^2 + \|E\| + \|E_{treino}\|)$
II. Predição de Links	7. Cálculo de Métricas	Calcular métricas - Pares de nós ligados no $Core \times Core$	$O(\|Core\|^2 * \|D\| * O(\|X\|^2) + O(\|V\| + \|E\|))$
		Calcular métricas - Pares de nós não ligados	$\boldsymbol{O(\|V\|^2 * \|D\| * O(\|X\|^2) + O(\|V\| + \|E\|))}$
	8. Clusterização	**Clusterizar pares de nós ligados** (*k-means*)	$\boldsymbol{O(\|V\|)}$
	9. Cálculo de Score e Ranking	Calcular score e criar ranking	$O(\|V\| + \|E\|)$
III. Avaliação do Resultado	10. Avaliação do Resultado	Avaliação do Resultado	$O(\|V\| + \|E\|)$

Tabela 2: Lista de algoritmos e sua complexidade na qual $|V|$ representa a quantidade de vértices, $|E|$ quantidade de arestas, $|S|$ quantidade de estados da rede, $|D|$ quantidade de métricas de similaridade e $|X|$ quantidade de palavras a serem representadas.

5 EXPERIMENTOS E RESULTADOS

5.1 Considerações iniciais

Neste capítulo são apresentadas as características dos experimentos e do ambiente computacional utilizados para conduzir a avaliação do método proposto, além de discutir e analisar os resultados obtidos nos referidos experimentos. Principal insumo para os experimentos, os dados foram carregados a partir das seguintes redes de coautoria disponíveis no repositório Arxiv:

- Astro-ph (Astrofísica)
- Cond-mat (Matéria Condensada)
- Gr-qc (Relatividade geral e cosmologia quântica)
- Hep-ph (Física em energia alta - Fenomenologia)
- Hep-th (Física em energia alta - Teoria)

Populares na tarefa da predição de links (LU; ZHOU, 2011), essas redes têm servido como base para os experimentos de diversos trabalhos de pesquisa. Com destaque para o trabalho de (LIBEN-NOWELL; KLEINBERG, 2007a) e também os trabalhos de (FLORENTINO; GOLDSCHMIDT, 2017) e (SOARES; PRUDÊNCIO, 2012a), que, de forma análoga ao presente trabalho, estudaram uma visão com múltiplos estados da rede (conforme discutido no Capítulo 3), entre outros trabalhos de pesquisa nesta área.

5.2 Protótipo e ambiente de apoio à experimentação

Com o objetivo de validar o método proposto por este trabalho, foi desenvolvido um protótipo na linguagem Python (versão 2.7). A escolha desta linguagem de programação se deve à sua popularidade em projetos relacionados ao tema da ciência de dados, além da disponibilidade de bibliotecas que suportaram a implementação do protótipo, com destaque para:

- Networkx: Biblioteca voltada para a criação, manipulação e estudo da estrutura, dinâmica e funções de redes complexas, incluindo o grafo;

- scikit-learn: Algoritmos de aprendizado de máquina, em especial o algoritmo de clusterização utilizado no presente trabalho;
- gensim: Biblioteca de mineração de texto.

O código do protótipo [1] desenvolvido foi criado de forma a priorizar a separação de responsabilidades entre seus componentes, oferecendo a possibilidade de que as etapas e passos do processo fossem executadas de forma independente, além de ser configurável.

A execução dos experimentos e do protótipo criado para este fim ocorreu em um ambiente computacional contendo o sistema operacional Ubuntu 18.04.2 LTS, com 4 núcleos de processador Intel Xeon (R) e 4 Gigabytes de memória RAM, utilizando uma nuvem pública *AWS* como infraestrutura.

5.3 Configurações dos experimentos

Seguindo o método introduzido no Capítulo 4, em cada execução dos experimentos foi definido um conjunto de parâmetros. Em todos os experimentos foram considerados os valores de $t_i = 1992$ até $t_f = 1998$, com 6 anos de treinamento e 1 ano para teste ($t_c = 1997$). A escolha deste período se justifica na análise do conjunto de dados, cujo atributo temporal disponível (*timestamp*) é na base anual, visto que um período maior de treinamento pode ser útil para melhor avaliar o uso do histórico de evolução da rede.

Os dados estatísticos das redes de coautoria utilizadas nos experimentos são apresentados na Tabela 3.

	G_{Treino}		$Core$		
Rede	**Vértices**	**Arestas**	**Autores**	$\|E_{new}\|$	$\|E_{old}\|$
astro-ph	8024	88964	1400	2064	9262
cond-mat	8278	31708	1030	343	2366
gr-qc	2594	9366	294	67	586
hep-ph	6692	54592	1382	317	6903
hep-th	6158	24321	1056	421	2647

Tabela 3: Dados Estatísticos - 1992 a 1998

A Figura 25 traz um exemplo de um arquivo XML retornado via API da rede gr-qc no repositório Arxiv. Neste arquivo estão disponíveis informações relacionadas às publicações (arestas), bem como aos seus autores (vértices).

[1] O código do protótipo encontra-se disponível em: https://gitlab.com/arguscavalcante/link_pred

```
<entry>
  <id>http://arxiv.org/abs/gr-qc/9401005v1</id>
  <updated>1994-01-07T16:50:32Z</updated>
  <published>1994-01-07T16:50:32Z</published>
  <title>Geometrical Constraints on the Cosmological Constant</title>
  <summary>  The cosmological constant problem is examined under the assumption that the extrinsic curvature
      of the space-time contributes to the vacuum. A compensation mechanism based on a variable cosmological
      term is proposed. Under a suitable hypothesis on the behavior of the extrinsic curvature, we find that an initially
      large $\Lambda(t)$ rolls down rapidly to zero during the early stages of the universe. Using perturbation
      analysis, it is shown that such vacuum behaves essentially as a spin-2 field which is independent of the metric.
</summary>
  <author>
   <name>M. D. Maia</name>
   <arxiv:affiliation xmlns:arxiv="http://arxiv.org/schemas/atom">Universidade de Brasilia and NASA</arxiv:affiliation>
  </author>
  <author>
   <name>G. S. Silva</name>
   <arxiv:affiliation xmlns:arxiv="http://arxiv.org/schemas/atom">Universidade de Brasilia and NASA</arxiv:affiliation>
  </author>
  <arxiv:doi xmlns:arxiv="http://arxiv.org/schemas/atom">10.1103/PhysRevD.50.7233</arxiv:doi>
  <arxiv:comment xmlns:arxiv="http://arxiv.org/schemas/atom">Marcos@FNALV.FNAL.GOV, 17 pages, Latex, 2 figures obtained by
request</arxiv:comment>
  <arxiv:journal_ref xmlns:arxiv="http://arxiv.org/schemas/atom">Phys.Rev.D50:7233-7238,1994</arxiv:journal_ref>
  <arxiv:primary_category xmlns:arxiv="http://arxiv.org/schemas/atom" term="gr-qc" scheme="http://arxiv.org/schemas/atom"/>
 </entry>
```


Figura 25 – Exemplo de dados estruturados e não estruturados da rede de coautoria gr-qc.

Após carregados da RSV no passo de seleção e armazenamento (Passo 2 da Figura 18 no Capítulo 4), o arquivo contendo os dados assume em cada uma de suas linhas o formato apresentado na Tabela 4.

Atributo	**Valor**
Identificador:	astro-ph/9204001v1
Título da publicação:	Gamma-Ray Bursts as the Death Throes of Massive Binary Stars
Data de publicação:	1992
Rede de coautoria:	set(['astro-ph'])
Lista de autores:	set(['Ramesh Narayan', 'Tsvi Pira', 'Bohdan Paczyski'])
Resumo:	'It is proposed that gamma-ray bursts are created in the mergers of double neutron star binaries and black hole neutron star binaries at cosmological distances. Bursts with complex profiles and relatively long durations are the result of magnetic flares generated by the Parker instability in a post-merger differentially-rotating disk. Some bursts may also be produced through neutrino-antineutrino annihilation into electrons and positrons. In both cases, an optically thick fireball of size sles 100 km is initially created, which expands ultrarelativistically to large radii before radiating. Several previous objections to the cosmological merger model are eliminated. It is predicted that gamma-ray bursts will be accompanied by a burst of gravitational radiation from the spiraling-in binary which could be detected by LIGO.'

Tabela 4: Exemplo de dados formatados obtidos da RSV.

Sob a ótica da dinâmica temporal das redes, optou-se pela escolha dos valores dos parâmetros $\phi = 6$ e $\psi = 1$ utilizados para filtrar o conjunto mais frequente de vértices em G_{Treino} e G_{Teste}, respectivamente. Essa escolha foi fundamentada na obtenção de uma frequência de publicações para cada um dos autores classificados como mais ativos de forma a coincidir com, no mínimo, uma publicação em cada um dos anos de treinamento e teste.

O algoritmo de *clusterização k-means* foi utilizado nos experimentos devido à sua popularidade (XU; TIAN, 2015). Anteriormente à execução dos experimentos propriamente dita, a escolha dos parâmetros relacionados ao algoritmo de clusterização utilizado, passou por um processo de validação com o objetivo de identificar os valores ótimos para a base de coautoria analisada. Este processo de escolha de parâmetros ocorreu com base na abordagem de (KOHAVI, 1995), no qual após a obtenção do conjunto de dados a ser analisado (Passo 3 da Figura 18 no Capítulo 4), a base de treino G_{Treino} é separada em duas outras bases $G_{Pré-Treino}$ e $G_{Validação}$ usando como referência o período $t_i = 1992$ até $t_f = 1995$ (3 anos de treinamento e 1 de teste), conforme Figura 26, de forma idêntica ao que ocorre em Passo 3, Particionamento do Grafo. Na sequência todos os passos do processo são executados com uma variação nos valores destes parâmetros. Este processo resultou na escolha de um $k = 3$ para o parâmetro de entrada do algoritmo *k-means*. Após esse processo de validação, todos os outros experimentos foram executados com base no particionamento entre os períodos de treino e testes.

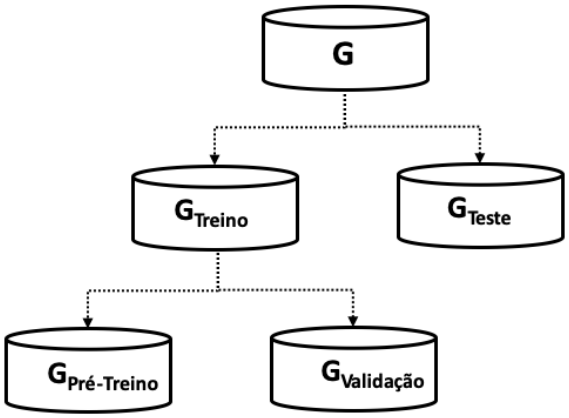


Figura 26 – Particionamento do Grafo.

Algumas heurísticas foram também utilizadas para lidar com problemas ocasionais nos dados. Os nomes dos autores por exemplo, foram construídos a partir da inicial do primeiro nome e último nome. Este processo pode incluir alguns itens incorretos devido à existência de autores diferentes com nomes nessa mesma estrutura. Outro problema identificado nos dados, foi a existência de publicações de um único autor e muito embora fosse possível representar este tipo de publicação como um *loop* (aresta que começa e

termina em um mesmo vértice), essa aresta não teria relação com o problema da predição de *links*, visto que não haveria ligação entre dois pares de vértices distintos. De qualquer forma, ambos problemas pareceram não impactar a lógica do processo e foram desta forma mantidos.

Com relação aos métodos de predição de ligações, foram avaliados:

- O método de (FLORENTINO; GOLDSCHMIDT, 2017) H_{Top} que utiliza exclusivamente informações topológicas, representados através das métricas *AA: Adamic-Adar*, *CN: Vizinhos Comuns* e *JC: Coeficiente de Jaccard*, que foram combinadas em conjuntos de 2 e 3 métricas. A escolha destas métricas se deve a sua popularidade no estado da arte (LU; ZHOU, 2011).
- O método proposto $H_{TopCont}$ combinando informações topológicas e contextuais. Com o objetivo de habilitar a comparação entre o método proposto e o método de (FLORENTINO; GOLDSCHMIDT, 2017), as mesmas métricas topológicas foram utilizadas. Em relação às métricas contextuais, foram analisados 4 diferentes tipos de combinações entre métricas e representações. Detalhadas na seção seguinte, as referidas combinações serão denominadas neste trabalho, para fins de simplificação do texto, de métricas de similaridade contextuais.

5.4 Combinação entre métricas e representações de dados

Considerando a hipótese a ser validada no presente trabalho, que combinar informações topológicas e contextuais no resgate do histórico do grafo pode levar a melhores resultados na tarefa de predição de links e considerando a questão fundamental da predição de links: prever o surgimento de novas ligações (ou arestas), optou-se por utilizar informações contextuais relacionadas às publicações, ou seja, às arestas dessas redes de coautoria. Além disso, tendo em vista que o título das publicações deve representar o objetivo primário destes trabalhos, são aqui apresentadas quatro diferentes combinações entre métricas de similaridade contextuais e representações de dados que foram utilizadas nos experimentos. De maneira a explorar essas combinações, foram utilizadas por conta de sua popularidade no estado da arte (LU; ZHOU, 2011; HASAN; ZAKI, 2011; NETTLETON, 2013; WANG et al., 2015; MARTÍNEZ; BERZAL; CUBERO, 2016) as métricas similaridade do cosseno e coeficiente de *Jaccard*, além das representações *BoW*, *TF-IDF* e *Doc2Vec*.

Na sequência são ilustradas as combinações dos 3 tipos de representação da informação contextual para a métrica de similaridade cosseno, e a representação *BoW* para a métrica de similaridade coeficiente de *Jaccard*, com base nas publicações fictícias de dois autores arbitrários u e v. Para acompanhar os exemplos, considere-se a título ilustrativo,

as seguintes informações sobre os títulos de 2 publicações de dois autores diferentes (u e v).

Título da publicação 1 do autor u = ['p1 p2 p3']

Título da publicação 2 do autor u = ['p1 p3 p4']

Título da publicação 1 do autor v = ['p1 p4 p5']

Título da publicação 2 do autor v = [p1 p5 p5']

5.4.1 Métrica de similaridade contextual DC (Doc2Vec Cosine)

Na primeira métrica, a representação dos dados é feita com base em *Doc2Vec* (LE; MIKOLOV, 2014). Inicialmente, é gerado um modelo contendo os títulos de todas as publicações de G_{Treino}, baseado em um vetor de 300 posições. Na sequência, cada um dos títulos de um determinado autor é submetido ao modelo previamente construído, gerando uma saída em formato vetorial. Aplicando esse processo para cada um dos títulos individualmente, obtêm-se um conjunto de vetores cujo cálculo da média é a agregação de todos os títulos de determinado autor. Tendo os dois vetores sido gerados para os dois autores, procede-se ao cálculo da similaridade do cosseno (SALTON, 1989). Esta combinação é exemplificada na sequência.

Representação *Doc2Vec* da publicação 1 do autor u = [1.1208358e-03 -1.5608722e-04 1.3389485e-05 -4.6276793e-04 4.2649574e-04 3.7302950e-04 -1.1391718e-03 ...]

Representação *Doc2Vec* da publicação 2 do autor v = [-6.81917067e-04 -7.98018591e-04 1.02653657e-03 1.27843325e-03 -8.54002155e-06 -1.61439937e-04 3.10837087e-04 ...]

Representação *Doc2Vec* da publicação 1 do autor v = [-8.64784124e-06 1.57604599e-03 -1.46874855e-03 1.07545266e-03 1.54794205e-03 -1.21643953e-03 1.24430144e-03 ...]

Representação *Doc2Vec* da publicação 2 do autor v = [-1.2115438e-03 -4.8288709e-04 8.1494864e-04 1.6353980e-03 1.0725727e-03 5.6104892e-04 -1.5668343e-03 ...]

Agregação das representações *Doc2Vec* das publicações do autor u = [2.19459354e-04 -4.77052905e-04 5.19963040e-04 4.07832675e-04 2.08977857e-04 1.05794781e-04 -4.14167356e-04 ...]

Agregação das representações *Doc2Vec* das publicações do autor v = [-6.10095798e-04 5.46579482e-04 -3.26899957e-04 1.35542534e-03 1.31025736e-03 -3.27695307e-04 -1.61266420e-04 ...]

Similaridade do cosseno entre as publicações de u **e** v = 0.048033364

5.4.2 Métrica de similaridade contextual BC (BoW Cosine)

Nesta métrica foi aplicado um pré-processamento nos títulos com o suporte de uma técnica de *stemming*, na qual as palavras foram encurtadas para sua raiz (LOVINS, 1968). Foram também removidas *stopwords*, que são palavras sem significado. Após o pré-processamento, os títulos das publicações dos pares de vértices foram agregados com um operador de concatenação, juntando todos os títulos de um autor em um grande parágrafo. Uma segunda agregação é realizada para identificar a união das palavras de ambos os parágrafos. Em seguida, cada uma das palavras é contada gerando um vetor de ocorrência *bag-of-words* (HARRIS, 1954). Finalmente a similaridade destes dois vetores é calculada através da similaridade do cosseno (SALTON, 1989). Usando como base as mesmas publicações dos autores u e v do exemplo anterior, a seguir é demonstrada a representação e métrica combinadas:

Agregação das publicações do autor u = ['p1', 'p2', 'p3', 'p1', 'p3', 'p4']

Agregação das publicações do autor v = ['p1', 'p4', 'p5', 'p1', 'p5', 'p5']

Agregação das publicações dos autores u **e** v = ['p1', 'p2', 'p3', 'p4', 'p5']

Representação *bag-of-words* das publicações do autor u = ['2', '1', ' 2', '1', '0']

Representação *bag-of-words* das publicações do autor v = ['2', '0', '0', '1', '3']

Similaridade do cosseno entre as publicações de u **e** v = 0.42257712736425834

5.4.3 Métrica de similaridade contextual TC (TF-IDF Cosine)

De forma similar a métrica anterior, foi realizado um pré-processamento com *stemming* e remoção de *stopwords*, além de efetuar as mesmas agregações. A diferença neste caso é a escolha da representação através do vetor *TF-IDF* (destalhada no Capítulo 2), que objetiva refletir a importância de cada palavra diante de um conjunto de palavras (ZHAI; LAFFERTY, 2004). Este conjunto de palavras, também chamado de *corpus* (KOWSARI et al., 2019), neste caso é composto da agregação das publicações dos autores u e v. Da mesma forma, a similaridade do cosseno (SALTON, 1989) foi utilizada para se obter o resultado final. O exemplo a seguir detalha essa diferente representação e o resultado obtido.

Agregação das publicações do autor u = ['p1', 'p2', 'p3', 'p1', 'p3', 'p4']

Agregação das publicações do autor v = ['p1', 'p4', 'p5', 'p1', 'p5', 'p5']

Agregação das publicações dos autores u **e** v = ['p1', 'p2', 'p3', 'p4', 'p5']

Representação *TF-IDF* das publicações do autor u = ['0.518534', '0.364391', ' 0.728781', '0.259267', '0.000000']

Representação *TF-IDF* das publicações do autor v = ['0.419056', '0.000000', '0.000000', '0.209528', '0.883453']

Similaridade do cosseno entre as publicações de u **e** v = 0.27161864519299295

5.4.4 Métrica de similaridade contextual BJ (BoW Jaccard)

Na métrica BJ, os títulos das publicações dos pares de vértices foram agregados com um operador de concatenação, juntando todos os títulos de um autor em uma grande sentença. Neste caso porém, não é utilizado um formato *bag-of-words* com um vetor de ocorrências, assim as próprias palavras são utilizadas. Na métrica de similaridade coeficiente de *Jaccard* a interseção das palavras do conjunto de palavras dos títulos de cada autor é normalizada pelo número de palavras da união de ambos os conjuntos, conforme exemplo a seguir.

Agregação das publicações do autor u = ['p1', 'p2', 'p3', 'p1', 'p3', 'p4']

Agregação das publicações do autor v = ['p1', 'p4', 'p5', 'p1', 'p5', 'p5']

Similaridade com o coeficiente de *Jaccard* entre as publicações de u **e** v = $\frac{2}{5} = 0.4$

5.5 Resultados e avaliação

Os resultados dos experimentos são apresentados nesta seção por meio de tabelas que indicam a medida de avaliação utilizada, a rede analisada e demais informações sobre as métricas e métodos analisados, conforme objetivo particular de cada execução.

5.5.1 Primeira execução

Na primeira execução foi avaliado o método proposto pelo presente trabalho $H_{TopCont}$ em relação ao cálculo do *improvement*. O objetivo da primeira execução foi identificar no método proposto, a métrica com melhor resultado, considerando as métricas de similaridade contextuais adotadas nos experimentos (DC, BC, TC e BJ) combinadas com métricas topológicas clássicas (AA, CN, JC). Os dados referentes aos resultados são apresentados na Tabela 5.

Rede	PR	Métrica de Similaridade	AA, CN	AA, JC	CN, JC	AA, CN, JC
astro-ph	0,0021	DC	**26,18**	**44,85**	**31,87**	**25,73**
		BC	9,79	6,14	8,42	10,24
		TC	8,19	7,51	7,51	9,33
		BJ	10,70	5,92	9,33	10,24
cond-mat	0,0006	DC	**40,35**	26,90	13,45	**40,35**
		BC	26,90	26,90	26,90	26,90
		TC	31,38	**31,38**	8,96	26,90
		BJ	22,42	**31,38**	**31,38**	26,90
gr-qc	0,0015	DC	**18,92**	-	9,46	**18,92**
		BC	9,46	-	**28,39**	9,46
		TC	9,46	-	**28,39**	9,46
		BJ	9,46	-	-	-
hep-ph	0,0006	DC	**50,03**	**20,38**	9,26	**50,03**
		BC	22,23	14,82	**14,82**	24,09
		TC	22,23	9,26	11,11	20,38
		BJ	14,82	12,97	**14,82**	16,67
hep-th	0,0007	DC	**46,92**	62,56	56,30	**43,79**
		BC	40,66	**68,81**	**59,43**	40,66
		TC	40,66	65,68	56,30	40,66
		BJ	14,82	12,97	14,82	16,67

Tabela 5: Comparação entre combinações de métricas de similaridade contextuais e métricas topológicas clássicas do método proposto $H_{TopCont}$.

A métrica DC obteve o maior número absoluto de vitórias (13), enquanto BC obteve 2 vitórias, BJ obteve 1 vitória e TC nenhuma vitória. Houve também três situações de empate, sendo 1 para TC e BJ, 1 para BC e TC e 1 para BC e BJ.

5.5.2 Segunda execução

De forma a verificar a questão fundamental deste trabalho, que busca validar a hipótese de que a combinação de informações topológicas e contextuais no histórico do grafo pode levar a melhores resultados, foi avaliado o método proposto $H_{TopCont}$ configurado com a melhor métrica de similaridade contextual (DC conforme resultado da primeira execução) e o comparamos com o método H_{Top} proposto em (FLORENTINO; GOLDSCHMIDT, 2017). Para essa análise foram utilizadas as medidas de avaliação: cálculo do *improvement* (LIBEN-NOWELL; KLEINBERG, 2007a), área sob a curva e precisão (LU; ZHOU, 2011).

Os dados referentes aos resultados dessa execução com base no cálculo do *improvement* são apresentados na Tabela 6. Em resumo, o método proposto $H_{TopCont}$ foi superior em 14 ocasiões, contra 3 de H_{Top}. Houve ainda 3 empates.

Rede	PR	Método	AA, CN	AA, JC	CN, JC	AA, CN, JC
astro-ph	0,0021	H_{Top}	10,01	7,51	8,88	9,79
		$H_{TopCont}$	**26,18**	**44,85**	**31,87**	**25,73**
cond-mat	0,0006	H_{Top}	35,87	22,42	**22,42**	35,87
		$H_{TopCont}$	**40,35**	**26,90**	13,45	**40,35**
gr-qc	0,0015	H_{Top}	9,46	**9,46**	-	9,46
		$H_{TopCont}$	**18,92**	-	**9,46**	**18,92**
hep-ph	0,0006	H_{Top}	18,53	9,26	**29,65**	22,23
		$H_{TopCont}$	**50,03**	**20,38**	9,26	**50,03**
hep-th	0,0007	H_{Top}	**46,92**	**62,56**	53,17	**43,79**
		$H_{TopCont}$	**46,92**	**62,56**	**56,30**	**43,79**

Tabela 6: Comparação entre o método proposto $H_{TopCont}$ (métrica de similaridade contextual utilizada: DC) e o método H_{Top} de (FLORENTINO; GOLDSCHMIDT, 2017) com base na medida de avaliação cálculo do *improvement*.

Os resultados da segunda execução com base na medida de avaliação AUC são apresentados em duas visões distintas: na primeira visão, têm-se os resultados por *dataset* (Tabela 7) e na segunda pela média entre os resultados de todos os *datasets* (Figura 27). Os resultados mostram que o método proposto $H_{TopCont}$ foi superior em 9 ocasiões, contra 7 de H_{Top}, além de 4 empates na visão por *dataset*. Na visão média entre todos os *datasets*, o método proposto $H_{TopCont}$ obteve resultado superior em todas as comparações com H_{Top}.

Rede	Método	AA, CN	AA, JC	CN, JC	AA, CN, JC
astro-ph	H_{Top}	0.7360	0.7349	0.7350	0.7361
	$H_{TopCont}$	**0.7370**	**0.7376**	**0.7372**	**0.7375**
cond-mat	H_{Top}	0.6469	0.6468	**0.6468**	0.6469
	$H_{TopCont}$	**0.6470**	**0.6470**	**0.6468**	**0.6470**
gr-qc	H_{Top}	**0.5682**	0.5679	**0.5681**	**0.5682**
	$H_{TopCont}$	**0.5682**	**0.5681**	0.5677	**0.5682**
hep-ph	H_{Top}	**0.7163**	0.7157	**0.7160**	**0.7163**
	$H_{TopCont}$	0.7161	**0.7159**	0.7152	0.7161
hep-th	H_{Top}	**0.6820**	**0.6821**	**0.6820**	**0.6821**
	$H_{TopCont}$	**0.6820**	0.6819	0.6819	0.6820

Tabela 7: Comparação entre o método proposto $H_{TopCont}$ (métrica de similaridade contextual utilizada: DC) e o método H_{Top} de (FLORENTINO; GOLDSCHMIDT, 2017) com base na medida de avaliação AUC.

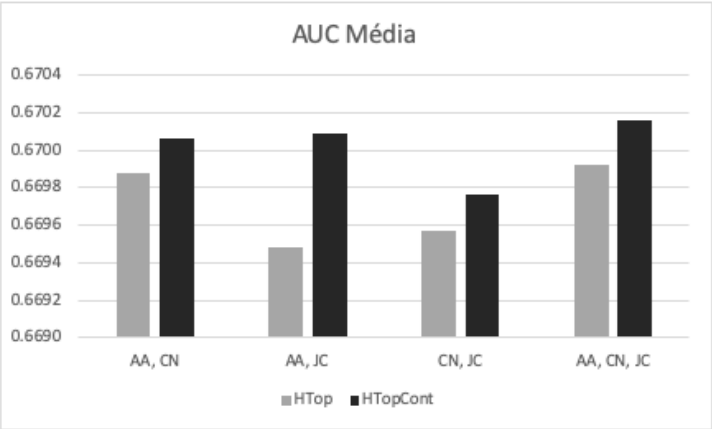


Figura 27 – Comparação entre o método proposto $H_{TopCont}$ (métrica de similaridade contextual utilizada: DC) e o método H_{Top} de (FLORENTINO; GOLDSCHMIDT, 2017) com base na medida de avaliação AUC média entre todos os *datasets*.

Na medida de avaliação precisão também são apresentadas duas visões distintas. Na primeira visão, têm-se os resultados por *dataset* (Figura 28) e na segunda pela média entre os resultados de todos os *datasets* (Figura 29). Os resultados mostram que o método proposto $H_{TopCont}$ foi superior em 13 ocasiões, contra 3 de H_{Top}, além de 4 empates na visão por *dataset*. Na visão média entre todos os *datasets*, o método proposto $H_{TopCont}$ obteve resultado superior em todas as comparações com H_{Top}.

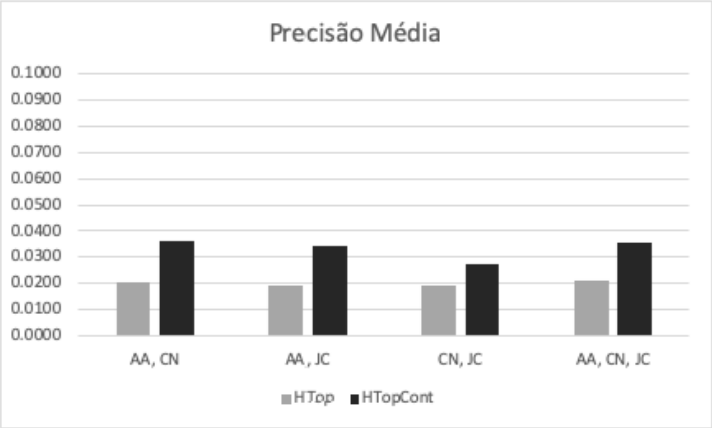


Figura 29 – Comparação entre o método proposto $H_{TopCont}$ (métrica de similaridade contextual utilizada: DC) e o método H_{Top} de (FLORENTINO; GOLDSCHMIDT, 2017) com base na medida de avaliação precisão média entre todos os *datasets*.

Com o resultado dos experimentos é possível chegar a algumas conclusões. Em primeiro lugar, tanto o método proposto (em todas as suas quatro combinações de métricas e representações) e o método de (FLORENTINO; GOLDSCHMIDT, 2017) obtiveram resultados superiores aos resultados do preditor randômico. Este resultado aponta que os

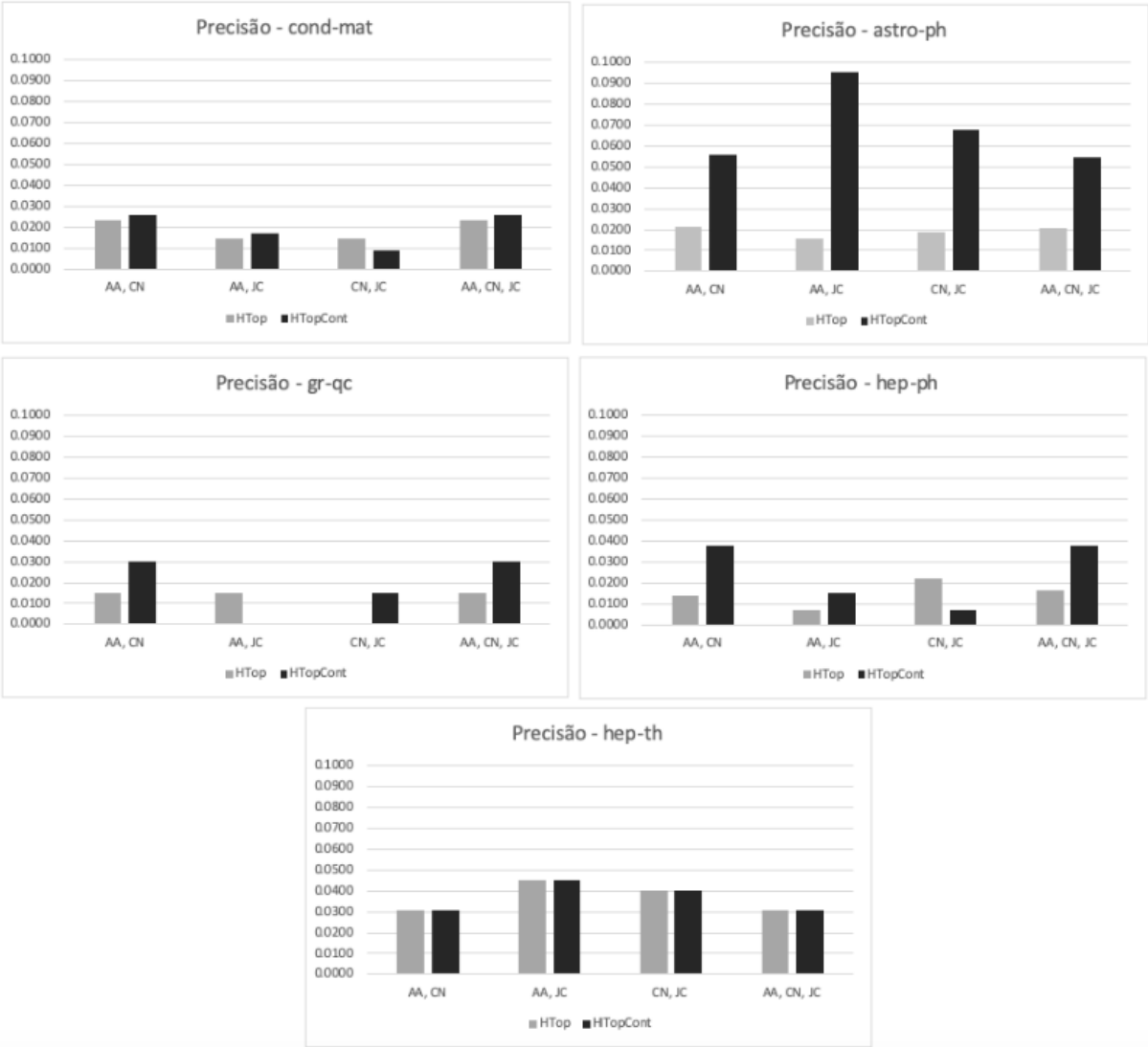


Figura 28 – Comparação entre o método proposto $H_{TopCont}$ (métrica de similaridade contextual utilizada: DC) e o método H_{Top} de (FLORENTINO; GOLDSCHMIDT, 2017) com base na medida de avaliação precisão por *dataset*.

métodos avaliados tem melhores chances de realizar predições corretamente, do que com base em uma decisão de escolha aleatória.

No método proposto por este trabalho, obteve-se melhores resultados da métrica de similaridade contextual DC em relação às demais, indicando que o uso da representação *Doc2Vec* tem melhor desempenho do que as outras representações (BoW e *Vetor TF-IDF*) para o cálculo usando a similaridade do cosseno.

O resultado mais importante obtido, diz respeito à superioridade do método proposto $H_{TopCont}$ (na métrica de similaridade contextual DC) em relação ao método H_{Top} de (FLORENTINO; GOLDSCHMIDT, 2017) que utiliza apenas informações topológicas. Os resultados superiores obtidos nestes experimentos com todas as medidas de avaliação

confirmam de forma expressiva a validade da hipótese da combinação das informações contextuais com informações topológicas disponíveis no histórico evolutivo da rede.

5.5.3 Tempo de execução

Visando complementar a análise da complexidade apresentada e discutida no Capítulo 4, o tempo de execução do método proposto foi registrado para uma execução completa do experimento em todos os *datasets*. Em um modelo de execução similar aos demais experimentos, foram utilizados: a unidade de particionamento da rede (*timestamp*) em base anual, as métricas de similaridade topológicas (AA, CN, JC) e contextual (DC), além do algoritmo de *clusterização k-means*.

Etapa	**Passo**	**Algoritmo**	**astro-ph**	**cond-mat**	**gr-qc**	**hep-ph**	**hep-th**
I. Preparação dos Dados	1. Configuração						
	2. Seleção e Armazenamento	Obter dados RSV	00:03:46	00:03:57	00:01:48	00:06:20	00:04:51
	3. Geração do Grafo	Gerar grafo contendo os registros da RSV	**00:02:28**	**00:01:10**	**00:00:15**	**00:01:25**	**00:00:46**
	4. Particionamento do Grafo	Gerar grafo de treino G_{Treino}					
		Gerar grafo de treino G_{Teste}					
	5. Resgate do Histórico	**Resgatar histórico evolutivo da rede**					
	6. Identificação dos Conjuntos de Apoio	Identificar pares de nós ligados	00:55:51	00:15:59	00:00:43	00:18:10	00:05:37
		Identificar conjunto *Core*					
		Identificar pares de nós não ligados					
II. Predição de Links	7. Cálculo de Métricas	Calcular métricas - Pares de nós ligados	**00:00:59 AA** **00:01:06 CN** **00:01:31 JC** **08:34:02 DC**	**00:00:23 AA** **00:00:19 CN** **00:00:23 JC** **01:23:29 DC**	**00:00:04 AA** **00:00:03 CN** **00:00:03 JC** **00:10:17 DC**	**00:00:49 AA** **00:00:36 CN** **00:00:43 JC** **02:57:39 DC**	**00:00:23 AA** **00:00:19 CN** **00:00:23 JC** **01:10:44 DC**
		Calcular métricas - Pares de nós não ligados					
	8. Clusterização	**Clusterizar pares de nós ligados**	**00:12:32**	**00:05:06**	**00:00:30**	**00:09:06**	**00:05:10**
	9. Cálculo de Score e Ranking	Calcular score e criar ranking	00:12:25	00:02:45	00:00:12	00:06:33	00:02:57
III. Avaliação do Resultado	10. Avaliação do Resultado	Avaliação do Resultado					
		Total	10:04:40	01:53:31	00:13:55	03:50:41	01:31:10

Tabela 8: Tempo de execução do experimento com o método proposto $H_{TopCont}$.

É possível verificar que o tempo de execução do cálculo de métricas de similaridade (especialmente a métrica de similaridade contextual) foi maior em todas as situações, indicando ser este um fator importante a ser levado em consideração na aplicação do método proposto.

6 CONSIDERAÇÕES FINAIS

As RSVs desempenham atualmente um papel de relevância em nossa sociedade. Diariamente, milhões de usuários acessam essas redes para interagir com outros indivíduos, empresas e organizações. Com base na sua importância e popularidade, academia e indústria tem se dedicado a analisar seu desenvolvimento e a evolução dessas redes.

Nas tarefas de análise das RSVs utiliza-se com frequência um grafo para representar matematicamente essas redes. Nesse grafo, os vértices representam os indivíduos e as arestas, algum tipo de conexão entre esses indivíduos. Prever se um par de vértices vai se conectar no futuro é uma tarefa relevante na análise de RSVs conhecida como o problema da predição de *links*. Essa tarefa se baseia nas informações disponíveis na rede, que podem ser de natureza estritamente topológica (que diz respeito à estrutura da rede) ou ainda contextual, que incorpora características dos usuários das redes e de suas interações.

A maioria dos trabalhos de pesquisa no estado da arte da tarefa de predição de *links*, utiliza uma versão estática do estado da rede, sem levar em conta os instantes de tempo nos quais surgiram novas arestas na rede. Uma visão com múltiplos estados, que utiliza esses instantes de tempo, ainda foi pouco explorada e os autores que o fizeram levaram em conta exclusivamente informações topológicas disponíveis nas RSVs. Assim sendo, o presente trabalho se propôs a investigar o efeito da combinação de informações topológicas com informações contextuais. O objetivo desta investigação é melhorar o desempenho da tarefa de predição de links que utiliza múltiplos estados da rede.

Utilizando a abordagem não supervisionada na predição de *links*, o presente trabalho avaliou diferentes métricas de similaridade contextual em cinco redes de coautoria. A avaliação foi realizada com a combinação das representações de dados não estruturados em formato de texto *BoW*, *TF-IDF* e *Doc2Vec* para a similaridade do cosseno e da representação *BoW* para o coeficiente de *Jaccard*. A comparação experimental da abordagem proposta pelo presente trabalho com o trabalho de (FLORENTINO; GOLDSCHMIDT, 2017) indicou resultados superiores em todas as avaliações e a confirmação da hipótese levantada.

Em relação ao tempo de execução do método proposto, vale ressaltar que o uso de métricas de similaridade contextual aumenta o tempo de execução do método quando comparado ao estado da arte. Essa ressalva deve ser levada em consideração de acordo com as particularidades de outras RSVs nas quais o método aqui proposto seja aplicado.

Além dos resultados obtidos e da confirmação da hipótese levantada, são apresen-

tadas ainda as seguintes contribuições:

- Análise do estado da arte da predição de links e sua relação com as informações utilizadas e o modo de tratamento do estado da rede;
- Criação de um protótipo para avaliação do método proposto;
- Recriação do método proposto por (FLORENTINO; GOLDSCHMIDT, 2017) para efeitos de comparação com o método proposto;
- Estudo da análise de complexidade dos algoritmos implementados.

Como parte dos possíveis trabalhos futuros deve ser verificado o comportamento do método proposto em outras redes sociais virtuais. O principal objetivo desta avaliação é testar o método em um cenário no qual os instantes de tempo do histórico evolutivo da rede variam em uma unidade de tempo diferente da rede de coautoria (de base anual). Outra possibilidade de pesquisa, diz respeito a utilizar as técnicas da abordagem supervisionada na predição de links considerando o histórico evolutivo da rede, tópico ainda não explorado no estado da arte.

Uma outra linha de investigação seria aplicar diferentes algoritmos de clusterização, verificando sua performance no mesmo método e demais implicações da complexidade de tempo de execução destes algoritmos, discutindo sua viabilidade na análise de RSVs. Ainda é possível explorar outras informações contextuais, além do título da publicação, na mesma rede de coautoria. Finalmente, o uso de outras métricas de similaridade contextual podem ser analisado tendo como base o método proposto. Adicionalmente, é possível utilizar o método de (FLORENTINO; GOLDSCHMIDT, 2017), apenas com métricas de similaridade contextuais, ao invés das topológicas exploradas por seus autores tendo em vista que o método aqui proposto, permite essa abordagem.

REFERÊNCIAS

ADAMIC, L. A.; ADAR, E. Friends and neighbors on the web. *Social Networks*, v. 25, n. 3, p. 211 – 230, 2003. ISSN 0378-8733. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0378873303000091>. 31

AHMED, N. M.; CHEN, L. An efficient algorithm for link prediction in temporal uncertain social networks. *Inf. Sci.*, Elsevier Science Inc., New York, NY, USA, v. 331, n. C, p. 120–136, fev. 2016. ISSN 0020-0255. Disponível em: <https://doi.org/10.1016/j.ins.2015.10.036>. 43, 46

AOUAY, S.; JAMOUSSI, S.; GARGOURI, F. Hybrid swarm based method for link prediction in social networks. In: *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*. [S.l.: s.n.], 2015. p. 974–981. ISSN 1082-3409. 43, 44

BENGIO, Y. et al. Neural probabilistic language models. In: _____. *Innovations in Machine Learning: Theory and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 137–186. Disponível em: <https://doi.org/10.1007/3-540-33486-6_6>. 37

CELEBI, M. E.; KINGRAVI, H. A.; VELA, P. A. A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Systems with Applications*, v. 40, n. 1, p. 200 – 210, 2013. ISSN 0957-4174. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0957417412008767>. 68

CHENG, J. et al. Can cascades be predicted? In: *Proceedings of the 23rd International Conference on World Wide Web*. New York, NY, USA: ACM, 2014. (WWW '14), p. 925–936. ISBN 978-1-4503-2744-2. Disponível em: <http://doi.acm.org/10.1145/2566486.2567997>. 16

COLLOBERT, R.; WESTON, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In: *Proceedings of the 25th International Conference on Machine Learning*. New York, NY, USA: ACM, 2008. (ICML '08), p. 160–167. ISBN 978-1-60558-205-4. Disponível em: <http://doi.acm.org/10.1145/1390156.1390177>. 37

DEYLAMI, H. A.; ASADPOUR, M. Link prediction in social networks using hierarchical community detection. In: *2015 7th Conference on Information and Knowledge Technology (IKT)*. [S.l.: s.n.], 2015. p. 1–5. 43, 47

FACEBOOK. *Facebook Newsroom*. 2019. 30 jul. de 2019. Disponível em: <https://newsroom.fb.com/company-info>. 15

FLORENTINO, E.; GOLDSCHMIDT, R. Um método para predição de ligações em redes complexas baseado em históricos da topologia de grafos. 2017. 10, 11, 19, 43, 48, 49, 50, 72, 76, 80, 81, 82, 83, 85, 86

FORTUNATO, S. Community detection in graphs. *Physics Reports*, v. 486, n. 3, p. 75 – 174, 2010. ISSN 0370-1573. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0370157309002841>. 16

GAMA, J. et al. *Inteligência artificial: uma abordagem de aprendizado de máquina*. Grupo Gen - LTC, 2011. ISBN 9788521618805. Disponível em: <https://books.google.com.br/books?id=4DwelAEACAAJ>. 35, 50

GOLDSCHMIDT, R.; PASSOS, E. *Data Mining: Um Guia PráTico*. [S.l.]: Campus, 2005. ISBN 8535218777. 34

GRANDJEAN, M. La connaissance est un réseau: Perspective sur l'organisation archivistique et encyclopédique. 2014. 9, 16

HAN, X. et al. Link prediction for new users in social networks. In: *2015 IEEE International Conference on Communications (ICC)*. [S.l.: s.n.], 2015. p. 1250–1255. ISSN 1550-3607. 43, 47

HARRIS, Z. S. Distributional structure. *WORD*, Routledge, v. 10, n. 2-3, p. 146–162, 1954. 36, 78

HASAN, M. et al. Link prediction using supervised learning. 01 2006. 18, 28, 33

HASAN, M. A.; ZAKI, M. J. A survey of link prediction in social networks. In: _____. *Social Network Data Analytics*. Boston, MA: Springer US, 2011. p. 243–275. ISBN 978-1-4419-8462-3. Disponível em: <https://doi.org/10.1007/978-1-4419-8462-3_9>. 29, 30, 76

HE, Y.-l. et al. Owa operator based link prediction ensemble for social network. *Expert Syst. Appl.*, Pergamon Press, Inc., Tarrytown, NY, USA, v. 42, n. 1, p. 21–50, jan. 2015. ISSN 0957-4174. Disponível em: <http://dx.doi.org/10.1016/j.eswa.2014.07.018>. 43, 44

HRUSCHKA, E.; EBECKEN, N. A genetic algorithm for cluster analysis. *Intell. Data Anal.*, v. 7, p. 15–25, 02 2003. 39

JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data clustering: A review. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 31, n. 3, p. 264–323, 1999. ISSN 0360-0300. Disponível em: <http://doi.acm.org/10.1145/331499.331504>. 38, 68

JONES, K. S. Document retrieval systems. In: WILLETT, P. (Ed.). London, UK, UK: Taylor Graham Publishing, 1988. cap. A Statistical Interpretation of Term Specificity and Its Application in Retrieval, p. 132–142. ISBN 0-947568-21-2. Disponível em: <http://dl.acm.org/citation.cfm?id=106765.106782>. 36

KATZ, L. A new status index derived from sociometric analysis. *Psychometrika*, v. 18, n. 1, p. 39–43, Mar 1953. ISSN 1860-0980. Disponível em: <https://doi.org/10.1007/BF02289026>. 31

KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995. (IJCAI'95), p. 1137–1143. ISBN 1-55860-363-8. Disponível em: <http://dl.acm.org/citation.cfm?id=1643031.1643047>. 75

KONG, X.; YU, P. S. Graph classification in heterogeneous networks. In: _____. *Encyclopedia of Social Network Analysis and Mining*. New York, NY: Springer New York, 2017. p. 1–7. ISBN 978-1-4614-7163-9. Disponível em: <https://doi.org/10.1007/978-1-4614-7163-9_176-1>. 22

KOWSARI, K. et al. Text classification algorithms: A survey. *CoRR*, abs/1904.08067, 2019. Disponível em: <http://arxiv.org/abs/1904.08067>. 35, 78

KUMAR, A. *Web Technology*. USA: New York: Chapman and Hall/CRC., 2019. 15

KUMAR, R.; NOVAK, J.; TOMKINS, A. Structure and evolution of online social networks. In: _____. *Link Mining: Models, Algorithms, and Applications*. New York, NY: Springer New York, 2010. p. 337–357. ISBN 978-1-4419-6515-8. Disponível em: <https://doi.org/10.1007/978-1-4419-6515-8_13>. 15

LAISHRAM, R.; MEHROTRA, K.; MOHAN, C. K. Link prediction in social networks with edge aging. In: *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*. [S.l.: s.n.], 2016. p. 606–613. ISSN 2375-0197. 43, 45

LE, Q.; MIKOLOV, T. Distributed representations of sentences and documents. In: *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*. JMLR.org, 2014. (ICML'14), p. II–1188–II–1196. Disponível em: <http://dl.acm.org/citation.cfm?id=3044805.3045025>. 9, 37, 38, 77

LIBEN-NOWELL, D.; KLEINBERG, J. The link-prediction problem for social networks. *J. Am. Soc. Inf. Sci. Technol.*, John Wiley & Sons, Inc., New York, NY, USA, v. 58, n. 7, p. 1019–1031, maio 2007. ISSN 1532-2882. Disponível em: <http://dx.doi.org/10.1002/asi.v58:7>. 16, 17, 18, 29, 53, 59, 72, 80

LIBEN-NOWELL, D.; KLEINBERG, J. The link-prediction problem for social networks. *J. Am. Soc. Inf. Sci. Technol.*, John Wiley & Sons, Inc., New York, NY, USA, v. 58, n. 7, p. 1019–1031, maio 2007. ISSN 1532-2882. Disponível em: <http://dx.doi.org/10.1002/asi.v58:7>. 33

LOVINS, J. B. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, v. 11, p. 22–31, 1968. 78

LU, J.; HENCHION, M.; NAMEE, B. M. *Investigating the Effectiveness of Representations Based on Word-Embeddings in Active Learning for Labelling Text Datasets*. 2019. 36

LU, L.; ZHOU, T. Link prediction in complex networks: A survey. *Physica A*, v. 390, n. 6, p. 11501170, 2011. 17, 28, 30, 59, 72, 76, 80

MARTÍNEZ, V.; BERZAL, F.; CUBERO, J.-C. A survey of link prediction in complex networks. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 49, n. 4, p. 69:1–69:33, dez. 2016. ISSN 0360-0300. Disponível em: <http://doi.acm.org/10.1145/3012704>. 34, 76

MCPHERSON, M.; SMITH-LOVIN, L.; COOK, J. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, v. 27, p. 415–, 01 2001. 27

MIKOLOV, T. et al. Distributed representations of words and phrases and their compositionality. In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. USA: Curran Associates Inc., 2013. (NIPS'13), p. 3111–3119. Disponível em: <http://dl.acm.org/citation.cfm?id=2999792.2999959>. 37

MISLOVE, A. et al. Measurement and analysis of online social networks. In: *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*. New York, NY, USA: ACM, 2007. (IMC '07), p. 29–42. ISBN 978-1-59593-908-1. Disponível em: <http://doi.acm.org/10.1145/1298306.1298311>. 15

MUNASINGHE, L.; ICHISE, R. Time score: A new feature for link prediction in social networks. E95.D, p. 821–828, 03 2012. 32

MUNIZ, C. P.; GOLDSCHMIDT, R.; CHOREN, R. Combining contextual, temporal and topological information for unsupervised link prediction in social networks. *Knowledge-Based Systems*, v. 156, p. 129 – 137, 2018. ISSN 0950-7051. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0950705118302569>. 18, 24, 26, 32, 34, 43, 47

NASDAQ. *Nasdaq*. 2019. 30 jul. de 2019. Disponível em: <https://www.nasdaq.com/symbol/fb>. 15

NEGI, S.; CHAUDHURY, S. Link prediction in heterogeneous social networks. In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. New York, NY, USA: ACM, 2016. (CIKM '16), p. 609–617. ISBN 978-1-4503-4073-1. Disponível em: <http://doi.acm.org/10.1145/2983323.2983722>. 43, 44

NETTLETON, D. F. Data mining of social networks represented as graphs. *Computer Science Review*, v. 7, n. Supplement C, p. 1 – 34, 2013. ISSN 1574-0137. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1574013712000445>. 22, 30, 76

NEWMAN, M. Clustering and preferential attachment in growing networks. v. 64, p. 025102, 01 2001. 31, 32

PARIKH, Y. et al. Efficient word2vec vectors for sentiment analysis to improve commercial movie success. In: BHATTACHARYYA, S. et al. (Ed.). *Advanced Computational and Communication Paradigms*. Singapore: Springer Singapore, 2018. p. 269–279. ISBN 978-981-10-8240-5. 66

PECLI, A.; CAVALCANTI, M. C.; GOLDSCHMIDT, R. Automatic feature selection for supervised learning in link prediction applications: a comparative study. *Knowledge and Information Systems*, v. 56, n. 1, p. 85–121, Jul 2018. ISSN 0219-3116. Disponível em: <https://doi.org/10.1007/s10115-017-1121-6>. 18

POPESCUL, A.; POPESCUL, R.; UNGAR, L. H. Statistical relational learning for link prediction. 2003. 34

RUMMELE, N.; ICHISE, R.; WERTHNER, H. Exploring supervised methods for temporal link prediction in heterogeneous social networks. In: *Proceedings of the 24th International Conference on World Wide Web*. New York, NY, USA: ACM, 2015. (WWW '15 Companion), p. 1363–1368. ISBN 978-1-4503-3473-0. Disponível em: <http://doi.acm.org/10.1145/2740908.2741697>. 43, 46

SALTON, G. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989. ISBN 0-201-12227-8. 32, 77, 78

SHAFIE, T. A multigraph approach to social network analysis. *Journal of Social Structure*, Carnegie Mellon University, v. 16, 6 2015. ISSN 1529-1227. 23

SHALFOROUSHAN, S. H.; JALALI, M. Link prediction in social networks using bayesian networks. In: *2015 The International Symposium on Artificial Intelligence and Signal Processing (AISP)*. [S.l.: s.n.], 2015. p. 246–250. 43, 45

SHARMA, P. K.; RATHORE, S.; PARK, J. H. Multilevel learning based modeling for link prediction and users' consumption preference in online social networks. *Future Generation Computer Systems*, 2017. ISSN 0167-739X. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167739X17308014>. 43, 46

SINOARA, R. A.; ANTUNES, J.; REZENDE, S. O. Text mining and semantics: a systematic mapping study. *Journal of the Brazilian Computer Society*, v. 23, n. 1, p. 9, Jun 2017. ISSN 1678-4804. Disponível em: <https://doi.org/10.1186/s13173-017-0058-7>. 35

SOARES, P. R. d. S.; PRUDÊNCIO, R. B. C. Time series based link prediction. In: *The 2012 International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2012. p. 1–7. ISSN 2161-4407. 19, 47, 48, 49, 72

SOARES, P. R. d. S.; PRUDÊNCIO, R. B. C. Time series based link prediction. In: *The 2012 International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2012. p. 1–7. ISSN 2161-4407. 43

TASNADI, E.; BEREND, G. Supervised prediction of social network links using implicit sources of information. In: *Proceedings of the 24th International Conference on World Wide Web*. New York, NY, USA: ACM, 2015. (WWW '15 Companion), p. 1117–1122. ISBN 978-1-4503-3473-0. Disponível em: <http://doi.acm.org/10.1145/2740908.2743037>. 43, 47

TRAVERS, J.; MILGRAM, S. An experimental study of the small world problem. *Sociometry*, American Sociological Association, v. 32, n. 4, p. 425–443, dez. 1969. ISSN 0038-0431. Disponível em: <http://links.jstor.org/sici?sici=0038-0431%28196912%2932%3A4%3C425%3AAESOTS%3E2.0.CO%3B2-W>. 27

TRUDEAU, R. *Introduction to Graph Theory*. Dover Pub., 1993. (Dover Books on Mathematics). ISBN 9780486678702. Disponível em: <https://books.google.com.br/books?id=h6sS_mrJDaEC>. 22, 23

WANG, P. et al. Link prediction in social networks: the state-of-the-art. *Science China Information Sciences*, v. 58, n. 1, p. 1–38, Jan 2015. ISSN 1869-1919. Disponível em: <https://doi.org/10.1007/s11432-014-5237-y>. 76

WU, J.; ZHANG, G.; REN, Y. A balanced modularity maximization link prediction model in social networks. *Information Processing & Management*, v. 53, n. 1, p. 295 – 307, 2017. ISSN 0306-4573. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0306457316304757>. 43, 45

XU, D.; TIAN, Y. A comprehensive survey of clustering algorithms. *Annals of Data Science*, v. 2, n. 2, p. 165–193, Jun 2015. ISSN 2198-5812. Disponível em: <https://doi.org/10.1007/s40745-015-0040-1>. 39, 40, 75

XU, Z. et al. A model-based approach to attributed graph clustering. In: *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. New York, NY, USA: ACM, 2012. (SIGMOD '12), p. 505–516. ISBN 978-1-4503-1247-9. Disponível em: <http://doi.acm.org/10.1145/2213836.2213894>. 24

YAO, L. et al. Link prediction based on common-neighbors for dynamic social network. *Procedia Computer Science*, v. 83, p. 82 – 89, 2016. ISSN 1877-0509. The 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016) / The 6th International Conference on Sustainable Energy Information Technology (SEIT-2016) / Affiliated Workshops. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1877050916301259>. 43, 46

YUAN, H. et al. A distributed link prediction algorithm based on clustering in dynamic social networks. In: *2015 IEEE International Conference on Systems, Man, and Cybernetics*. [S.l.: s.n.], 2015. p. 1341–1345. 43, 45

ZENG, R.; DING, Y.; XIA, X. Link prediction based on dynamic weighted social attribute network. In: *2016 International Conference on Machine Learning and Cybernetics (ICMLC)*. [S.l.: s.n.], 2016. v. 1, p. 183–188. ISSN 2160-1348. 43, 46

ZHAI, C.; LAFFERTY, J. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, ACM, New York, NY, USA, v. 22, n. 2, p. 179–214, abr. 2004. ISSN 1046-8188. Disponível em: <http://doi.acm.org/10.1145/984321.984322>. 78

ZHOU, T.; LÜ, L.; ZHANG, Y.-C. Predicting missing links via local information. *The European Physical Journal B*, v. 71, n. 4, p. 623–630, Oct 2009. ISSN 1434-6036. Disponível em: <https://doi.org/10.1140/epjb/e2009-00335-8>. 31