UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Otimização multinível em predição de links

Vinícius Ferreira da Silva

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Vinícius Ferreira da Silva

Otimização multinível em predição de links

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Alneu de Andrade Lopes

USP – São Carlos
Julho de 2018

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

F383o Ferreira da Silva, Vinícius
 Otimização multinível em predição de links /
 Vinícius Ferreira da Silva; orientador Alneu de
 Andrade Lopes. -- São Carlos, 2018.
 83 p.

 Dissertação (Mestrado - Programa de Pós-Graduação
 em Ciências de Computação e Matemática
 Computacional) -- Instituto de Ciências Matemáticas
 e de Computação, Universidade de São Paulo, 2018.

 1. Otimização multinível. 2. Predição de links. 3.
 Matching. 4. Coarsening. I. de Andrade Lopes,
 Alneu, orient. II. Título.

Vinícius Ferreira da Silva

Multilevel optimization for link prediction

Master dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP, in partial fulfillment of the requirements for the degree of the Master Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Alneu de Andrade Lopes

USP – São Carlos
July 2018

*Dedico este trabalho aos meus pais, Carlos e Izanete, e ao meu irmão Vitor.
Obrigado pelo apoio de sempre.*

AGRADECIMENTOS

Agradeço primeiramente aos meus pais, Carlos e Izanete. Eu não saberia ilustrar, com meu parco talento poético, o quanto devo a vocês. Fazem parte de todas minhas vitórias e eu não poderia acordar todas as manhãs sem ser grato por tudo. Agradeço também ao meu irmão, Vitor, pela descontração e por realmente gostar do seu irmão mais velho. Agradeço também à minha doce Isabela por ser tão ela.

Agradeço aos meus amigos de longa data Adenir, Tuanan, Victor, João Henrique e Eugênio. Não passamos por tempos fáceis mas sempre encontramos conforto em nossa amizade. Apesar da longa distância, nossos breves reencontros sempre enchem meu coração de alegria. Tenho saudades.

Sou grato por todos os amigos feitos no mestrado, mas não poderia deixar de destacar Diego e João Antunes. Pessoas excelentes que fogem totalmente de todos os estereótipos possíveis e imagináveis. Apesar de não terem feito mais do que sua obrigação, espero reencontrá-los muitas vezes.

Agradeço ao meu orientador Alneu por sempre acreditar no meu trabalho, pelo incentivo e por todas as oportunidades. Foi um período de grande engrandecimento.

Agradeço aos colegas do Laboratório de Inteligência Computacional (LABIC) por sua ajuda e pelos bons momentos.

Agradeço à Universidade de São Paulo, especialmente ao Instituto de Ciências Matemáticas e de Computação pela oportunidade. Agradeço ao CNPq (Conselho Nacional de Pesquisa e Desenvolvimento) por ter contribuído financeiramente para a realização desta pesquisa.

*“Eu sô problema de montão
De carnaval a carnaval
Eu vim da selva
Sou leão
Sou demais pro seu quintal”
(Edi Rock e Mano Brown)*

RESUMO

FERREIRA, V. **Otimização multinível em predição de links**. 2018. 83 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

A predição de *links* em redes é uma tarefa com aplicações em diversos cenários. Com a automatização de processos, as redes sociais, redes tecnológicas e outras cresceram muito em número de vértices e arestas. Portanto, a utilização de preditores de *links* em redes com alta complexidade estrutural não é trivial, mesmo considerando algoritmos de baixa complexidade computacional. A grande quantidade de operações necessárias para que os preditores possam escolher quais arestas são promissoras torna o processo de considerar a rede toda inviável na maioria dos casos. As abordagens existentes enfrentam essa característica de diversas formas, sendo que as mais populares são as que limitam o conjunto de pares de vértices que serão considerados para existência de arestas promissoras. Este projeto aborda a criação de uma estratégia que utiliza otimização multinível para contrair as redes, executar os algoritmos de predição de *links* nas redes contraídas e projetar os resultados de predição para a rede original, para reduzir o número de operações necessárias à predição de *links*. Os resultados mostram que a abordagem consegue reduzir o tempo necessário para predição, apesar de perdas esperadas na qualidade na predição.

Palavras-chave: Otimização multinível, predição de links, matching, coarsening.

ABSTRACT

FERREIRA, V. **Multilevel optimization for link prediction**. 2018. 83 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

Link prediction in networks is a task with applications in several scenarios. With the automation of processes, social networks, technological networks, and others have grown considerably in the number of vertices and edges. Therefore, the creation of systems for link prediction in networks of high structural complexity is not a trivial process, even considering low-complexity algorithms. The large number of operations required for predicting which edges are promising makes the considering of the whole network impracticable in many cases. The existing approaches face this characteristic in several ways, and the most popular are those that limit the set of vertex pairs that will be considered for the existence of promising edges. This project addresses a strategy that uses multilevel optimization to coarse networks, execute prediction algorithms on coarsened networks and project the results back to the original network, in order to reduce the number of operations for link prediction. The experiments show that the approach can reduce the time despite some expected losses of accuracy.

Keywords: Multilevel optimization, link prediction, matching, coarsening.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de rede com um conjunto de vértices sendo conectado por um conjunto de arestas.	34
Figura 2 – Exemplo de aplicação de um algoritmo de predição de <i>links</i>	37
Figura 3 – Fases da abordagem multinível para solução de problemas em redes.	45
Figura 4 – Abordagem de otimização multinível para predição de <i>links</i> em redes simples.	50
Figura 5 – Exemplo de aplicação dos algoritmos de <i>matching</i> propostos.	52
Figura 6 – Projeção de predições de <i>links</i> para redes homogêneas.	54
Figura 7 – Exemplo de rede com um conjunto de vértices sendo conectado por um conjunto de arestas.	65

LISTA DE ALGORITMOS

Algoritmo 1 – Predição multinível de <i>links</i> para redes simples	51
--	----

LISTA DE TABELAS

Tabela 1 – Sumário das redes utilizadas nos experimentos.	58
Tabela 2 – Tempos de execução em segundos do processo de <i>coarsening</i> das bases de dados.	62
Tabela 3 – Alterações nos coeficientes estruturais conforme o nível de <i>coarsening</i> da rede aumenta. O símbolo ”+” representa que do nível anterior para o atual houve um acréscimo no valor daquele coeficiente do último nível de <i>coarsening</i> até o atual. O símbolo ”-” indica decréscimos nesses coeficientes.	63
Tabela 4 – Resultados de <i>AUC</i> para as bases de dados. Os dados na tabela foram calculados para $n = 10000$	67
Tabela 5 – Resultados de <i>precisi@L</i> para as bases de dados. Os dados na tabela foram calculados para $L = 1000$	70
Tabela 6 – Resultados de tempos de execução proporcionais ao tempo necessário para predição sem nenhum tipo de predição para as bases de dados.	72
Tabela 7 – Resultados de tempos de execução do algoritmo de predição de <i>links</i> em segundos para o nível 0 de contração.	73

LISTA DE ABREVIATURAS E SIGLAS

AUC	<i>Area Under ROC Curve</i>
CN	<i>Common Neighbors</i>
HEM	<i>Heavy Edge Matching</i>
LEM	<i>Light Edge Matching</i>
LSVM	<i>Least Similar Vertex Matching</i>
MSVM	<i>Most Similar Vertex Matching</i>
PA	<i>Preferential Attachment</i>
RM	<i>Random Matching</i>
SER	<i>Super-Edge Replication</i>
WSER	<i>Weighted Super-Edge Replication</i>

LISTA DE SÍMBOLOS

A — Matriz de adjacência que representa a rede.

A_{ij} — Elemento da matriz de adjacência que representa conexão entre os vértices i e j .

G — Relação que representa uma rede ou grafo.

U — Conjunto universal de todas as arestas possíveis em uma rede.

V — Conjunto de vértices do grafo.

E — Conjunto de pares ordenados que representam arestas da rede.

k — Grau do vértice.

c — Grau médio da rede.

C — Coeficiente de aglomeração global da rede.

r — Coeficiente de assortatividade da rede.

d — Densidade da rede.

SUMÁRIO

1	INTRODUÇÃO	27
1.1	Contexto	27
1.2	Motivação	29
1.3	Objetivos e metodologia	30
1.4	Organização	31
2	FUNDAMENTOS E TRABALHOS RELACIONADOS	33
2.1	Redes complexas	33
2.1.1	<i>Coeficientes estruturais</i>	35
2.2	Predição de <i>links</i>	37
2.2.1	<i>Abordagens</i>	38
2.2.2	<i>Avaliação</i>	40
2.2.3	<i>Aplicações de predição de links</i>	41
2.3	Trabalhos relacionados com escalabilidade em predição de <i>links</i> . .	43
2.4	Otimização multinível	45
3	OTIMIZAÇÃO MULTINÍVEL EM PREDIÇÃO DE LINKS	49
4	AVALIAÇÃO EXPERIMENTAL	57
4.1	Bases de dados utilizadas	57
4.2	Método utilizado	59
4.3	Resultados e análises	60
4.3.1	<i>Análise do processo de coarsening</i>	60
4.3.2	<i>Análise da AUC</i>	64
4.3.3	<i>Análise de precisi@L</i>	66
4.3.4	<i>Análise de tempo de execução</i>	69
5	CONCLUSÕES	75
5.1	Limitações da abordagem	76
5.2	Trabalhos futuros	77
	REFERÊNCIAS	79

INTRODUÇÃO

1.1 Contexto

Redes, ou grafos, são estruturas relacionais que permitem que sistemas do mundo real sejam representados como ligações entre vértices, ou nós, através de arestas, ou *links*. De acordo com (NEWMAN, 2010), redes são formas gerais e ainda poderosas de representar padrões de conexões ou interações entre partes de um mesmo sistema. Embora seja uma forma interessante de representação, redes complexas exigem certos cuidados quanto à complexidade das operações que serão executadas, pois é comum encontrar aplicações que lidam com redes reais de milhares de vértices e milhões de arestas.

Muitos sistemas de informação utilizam redes para fazer a modelagem do objeto de estudo. Alguns exemplos de representações são redes sociais, como redes de interação pessoal, redes de colaboração, plataformas de redes sociais *online*; redes de comunicação, como redes de telefonia, *e-mail* e Internet; economia, como redes de negociação e redes de parcerias empresariais (COSTA *et al.*, 2007). A estrutura da rede permite que características relacionais dos ambientes que são modelados sejam expostas de forma mais direta. Além disso, os avanços no estudo de problemas envolvendo redes trouxeram medidas e algoritmos com diversos objetivos diferentes que permitem um entendimento maior dos objetos de estudo modelados. Alguns exemplos de problemas envolvendo redes são (NEWMAN, 2010): particionamento de redes, detecção de comunidades, buscas, existências de melhores caminhos, predição de *links*, etc.

De acordo com (GETOOR; DIEHL, 2005), o problema de predição de *links*, ou predição de arestas, é um subproblema da área de mineração de *links* que ocorre quando nem todas as arestas possíveis estão presentes em uma rede e existe o interesse em predizer a existência de tais arestas utilizando o conjunto de vértices e arestas existentes. Tendo em vista esta definição generalizada do problema, tais arestas podem ser arestas faltantes, ou seja, que já deveriam estar presentes na rede mas foram perdidas devido à problemas na captação dos dados ou ruídos. Além

de arestas faltantes, o objetivo pode ser também a predição de arestas futuras, ou seja, arestas que podem vir a existir considerando que a rede evolua com o passar do tempo. Descobrir arestas futuras pode ser muito vantajoso em diversos aspectos, tanto comerciais quanto científicos. Diversos algoritmos que operam em sistemas de recomendação em plataformas de redes sociais *online* ou serviços de *e-commerce* fazem a predição de utilidade entre usuários ou entre usuários e objetos para gerar novos relacionamentos ou oportunidades de consumo.

Modelando o serviço como uma rede, algoritmos de predição de *links* poderiam ser utilizados para extrair tais predições. As duas maiores dificuldades do problema estão em garantir a qualidade das arestas preditas e em como lidar com a significativa quantidade de arestas não observadas que podem ser preditas. Em uma rede esparsa com dezenas de milhares de vértices, pequena quando comparada à redes do mundo real que ultrapassam centenas de milhares de vértices, a quantidade de arestas a serem avaliadas ultrapassa centenas de milhões. Mesmo considerando algoritmos de predição que não possuam uma alta complexidade computacional, a quantidade de avaliações necessárias para selecionar arestas promissoras pode tornar o problema inviável quando todas as arestas não observadas são consideradas. É neste contexto que este projeto foi desenvolvido.

Para aumentar a escalabilidade de diversos algoritmos que têm como entrada redes esparsas com dezenas de milhares de vértices, foi desenvolvida uma abordagem de otimização denominada otimização multinível. Dado um algoritmo objetivo que deve ser aplicado em uma rede de larga escala, a otimização multinível objetiva a redução do número de operações necessárias para tal algoritmo e o faz da seguinte forma: A rede original é contraída iterativamente em redes menores, através de um processo de *matching* seguido de um processo de contração, sendo que em cada iteração a rede fica um nível de contração maior começando do nível zero. Quando chega em um nível de contração viável para que o algoritmo objetivo possa ser aplicado, o *coarsening*, ou o processo de contração da rede, termina e o algoritmo objetivo é aplicado nesta versão contraída da rede. A aplicação do algoritmo gera um resultado inicial que, devido ao tamanho reduzido da rede contraída, é obtido muito mais rápido do que seria se o algoritmo objetivo fosse aplicado na rede sem contração. O resultado inicial é então projetado de volta à rede original, gerando ao final do processo uma solução completa.

A contração da rede é feita através do processo de *coarsening*, onde um conjunto de arestas que serão removidas e cujo os vértices adjacentes serão condensados em super-vértices é selecionado. O processo de seleção de arestas é denominado *matching*. O processo de *coarsening*, a execução do algoritmo objetivo na rede contraída e a projeção dos resultados, ou *uncoarsening*, exigem ao todo uma quantidade menor de operações do que o processo de aplicar o algoritmo objetivo diretamente na rede original, ocasionando uma redução de tempo de execução. É comum que, devido à contração e aplicação do algoritmo objetivo em uma rede que não é a original, aconteçam perdas na qualidade dos resultados. Alguns trabalhos ainda realizam um processo de refinamento para diminuir as perdas causadas pela otimização multinível.

Originalmente a otimização multinível era utilizada apenas no problema de particionamento de redes. Porém, devido aos resultados promissores obtidos em diversos estudos e às possibilidades de expansão e fácil adaptabilidade para outros problemas, esta estratégia de otimização começou a ser utilizada também em problemas de detecção de comunidades, problema do caixeiro viajante, clusterização, desenho e visualização de redes, tarefas de aprendizado que utilizam redes, redução de dimensionalidade, etc. Entretanto, durante a pesquisa bibliográfica deste trabalho, não encontramos nenhum trabalho utilizando a otimização multinível para diminuir o tempo de execução de algoritmos de predição de *links*. Os principais trabalhos que propõem soluções para aumentar a escalabilidade de preditores de *links* têm limitações quanto à tecnologia, ou são limitados a algoritmos específicos de predição de *links* ou ainda limitam a predição à apenas um subconjunto de arestas não observadas da rede, fazendo com que arestas promissoras possam ser ignoradas pelos algoritmos. Utilizar a otimização multinível para acelerar o processo de predição de *links* fez com que fosse possível a criação de uma abordagem que não possui nenhuma dessas limitações, uma vez que a abordagem criada é aplicável com qualquer algoritmo de predição de *links*, não depende de tecnologias específicas de processamento paralelo, permitindo que implementação seja feita mais facilmente, e finalmente utiliza a rede toda para extrair as predições.

1.2 Motivação

O surgimento de redes de grande escala, complexas e dinâmicas que modelam cenários sociais, tecnológicos e biológicos provocou a necessidade de estudos e técnicas para a análise de tais redes. Dentre as principais disciplinas de análise e mineração de redes complexas estão os algoritmos de predição de *links*, que são utilizados tanto na recuperação de arestas perdidas quanto na predição de arestas futuras na rede. A predição de *links* é um problema amplamente estudado e com diversas aplicações em várias áreas do conhecimento. Os algoritmos presentes na literatura são utilizados para extrair predições em redes sociais, redes biológicas, aplicações médicas, aplicações comerciais através de sistemas de recomendações e seu uso é bastante diversificado.

Existem diversas abordagens para a predição de *links* sendo que na literatura, na maioria dos casos, o foco dos estudos está na qualidade das arestas preditas. Entretanto, as redes reais em que os algoritmos tendem a ser aplicados têm estruturas cada vez mais complexas, com mais vértices e consequentemente mais arestas não observadas, fazendo com que a escalabilidade dos algoritmos seja crítica. Além disso, diversos trabalhos da área demonstram que os algoritmos mais sofisticados, que exploram aspectos globais da rede, e que consequentemente possuem maior complexidade computacional, apresentam melhores resultados na predição de *links* do que algoritmos mais simples, que exploram apenas informações locais. Porém, devido à sua maior complexidade, a execução de tais algoritmos é impraticável, mesmo em redes com apenas dezenas de milhares de vértices. Alguns poucos trabalhos foram desenvolvidos com o objetivo

aumentar a escalabilidade de preditores de *links*. Dentre estes trabalhos, alguns utilizam características de algoritmos específicos para diminuir a complexidade computacional, outros trabalhos utilizam teorias e tecnologias de processamento paralelo para processar a enorme quantidade de arestas não observadas e, por fim, os trabalhos que propõem a limitação à predição, fazendo com que apenas um subconjunto de pares de vértices da rede seja considerado, diminuindo substancialmente a quantidade de arestas não observadas e correndo o risco da provável perda de arestas promissoras. Embora alguns destes trabalhos apresentem bons resultados, estão limitados pelo tipo de algoritmo de predição, por alguma questão tecnológica ou pagam o preço da perda de possíveis arestas promissoras.

De forma a tentar superar essas limitações, foi desenvolvida neste projeto uma abordagem para acelerar o processo de predição de *links* que não depende de nenhuma característica dos algoritmos de predição, ou de tecnologia, e também não limita a predição em subgrafos da rede original, fazendo com que nenhuma aresta não observada seja desconsiderada. Esta abordagem utiliza o paradigma de otimização multinível. Tendo como base os resultados promissores obtidos em trabalhos sobre particionamento, surgiu a possibilidade de utilizar o paradigma para outros problemas, como detecção de comunidades, coloração de grafos, visualização e também para o problema do caixeiro viajante. Porém, segundo nossa pesquisa bibliográfica, o paradigma nunca havia sido utilizado para o problema de predição de *links*. O paradigma multinível permite que a rede seja contraída, um algoritmo objetivo seja aplicado na versão contraída e os resultados obtidos sejam projetados, e este processo não é dependente do algoritmo objetivo considerado. Essa característica se adapta perfeitamente à necessidade de escalabilidade e também à existência de diversos algoritmos diferentes de predição de *links*.

1.3 Objetivos e metodologia

Este trabalho tem como seu principal objetivo a criação de uma abordagem de otimização multinível que possa ser aplicada com algoritmos de predição de *links* em redes para diminuir o número de operações e, conseqüentemente, o tempo de execução necessário para extrair predições. A partir deste objetivo principal, foram definidos os seguintes objetivos específicos:

- **Levantamento bibliográfico:** Para entender os principais algoritmos, aplicações, formas de avaliação e as principais deficiências das propostas atuais sobre escalabilidade em predição de *links*, realizar um levantamento bibliográfico sobre predição de *links*. Também é necessário realizar um levantamento sobre otimização multinível, com foco nas diferentes aplicações do paradigma de otimização multinível e sua viabilidade;
- **Criação de algoritmos de *matching*:** Os algoritmos de *matching* existentes na literatura não levam em conta questões de similaridade para condensar vértices em um mesmo super-vértice. É um objetivo específico deste projeto criar algoritmos de *matching* que levem em conta a similaridade dos vértices para a seleção dos pares;

- **Criação de algoritmos de *uncoarsening*:** Os algoritmos de *uncoarsening* existentes geralmente retrocedem os níveis de contração até que a rede original seja novamente alcançada, com os resultados devidamente projetados. Visto que não há necessidade para retrocesso na predição de *links*, pois o objetivo do preditor é extrair um *ranking* de arestas promissoras, é então necessário um algoritmo de projeção que consiga extrair as arestas diretamente do nível mais contraído da rede. É um objetivo específico deste projeto desenvolver tal algoritmo;
- **Avaliação da abordagem:** A abordagem desenvolvida deve ser avaliada. A forma mais interessante de se fazer a avaliação é comparar os resultados de medidas de desempenho em predição de *links* e de tempo de execução em diferentes níveis de contração da rede. A comparação de resultados entre os diversos níveis da rede mostra de forma interessante as perdas de qualidade e ganhos de tempo de execução.

1.4 Organização

O restante deste trabalho está organizado da seguinte maneira: O Capítulo 2 apresenta a fundamentação sobre redes complexas, predição de *links* e sobre o paradigma multinível. No Capítulo 3 estão descritos todos os detalhes da abordagem proposta neste projeto. O Capítulo 4 contém a avaliação experimental e as análises dos resultados obtidos. Finalmente, no Capítulo 5 estão as considerações finais, as principais vantagens e limitações do que foi desenvolvido e os trabalhos futuros que serão desenvolvidos.

FUNDAMENTOS E TRABALHOS RELACIONADOS

Neste capítulo são apresentados fundamentos para o entendimento desta dissertação. Inicialmente são apresentados conceitos sobre redes complexas e algumas características importantes. Depois estão conceitos relevantes a este trabalho sobre a tarefa de predição de *links* em redes. Por fim, são apresentados conceitos sobre otimização multinível em redes, que é a abordagem utilizada para a otimização dos preditores de *links* neste trabalho.

2.1 Redes complexas

Uma rede, ou grafo, é um conjunto de vértices, ou nós, ligados por um conjunto de arestas, ou *links*. Tais redes podem ser utilizadas para representar quaisquer sistemas que apresentem características de conectividade como redes de relacionamentos, sistemas biológicos, conexão de locais através de vias de acesso, entre outros (COSTA *et al.*, 2007). De acordo com (BOCCALETTI *et al.*, 2006), o estudo de redes e grafos teve início quando *Leonhard Euler* publicou sua solução para o problema das pontes de *Königsberg*, e teve sua popularização por meio de resultados obtidos de estudos em ciências sociais, mais especificamente de análises de redes sociais que representavam os relacionamentos entre entidades ou indivíduos, entre as décadas de 1920 e 1930. Os primeiros estudos sociais tratavam de redes menores e mais simples, principalmente de relacionamentos, comunicação entre grupos, negociações entre nações e transações entre corporações. A Figura 1 apresenta um exemplo simples de rede com um conjunto de vértices sendo conectado por um conjunto de arestas.

Formalmente, uma rede, ou grafo, é dada pela relação $G = (V, E)$, sendo V um conjunto de vértices e E um conjunto de pares do tipo (i, j) que representam um *link*, ou aresta, entre os vértices i e j . Considerando apenas redes não direcionadas, se todos os pares de vértices da rede que possuem conexões são ligados por apenas uma aresta e não existem arestas do tipo

Figura 1 – Exemplo de rede com um conjunto de vértices sendo conectado por um conjunto de arestas.

Fonte: Elaborada pelo autor.

laço, que ligam um vértice a ele mesmo, a rede é denominada rede simples. Caso exista mais de uma aresta ligando o mesmo par de vértices, ou seja uma aresta múltipla ou paralela, o grafo é então um multigrafo. Se $(i, j) \in E$, então os vértices i e j são tidos como vizinhos e a vizinhança de um vértice i , ou o conjunto de vizinhos do vértice i , é dado por $\Gamma(i)$. As redes podem ser direcionadas ou não direcionadas. Uma rede direcionada, ou digrafo, é uma rede em que as arestas apresentam direções, ou seja, na aresta (i, j) , j é a ponta inicial da aresta e i é a ponta final, sendo esta a definição adotada em (NEWMAN, 2010).

Existem várias formas de representar matematicamente uma rede, sendo a matriz de adjacência a maneira mais simples. A matriz de adjacência A para uma rede simples e não direcionada é uma matriz quadrada e simétrica tal que seus elementos A_{ij} são dados por:

$$A_{ij} = \begin{cases} 1, & \text{se existe uma aresta entre } i \text{ e } j \\ 0, & \text{caso contrário} \end{cases}$$

Considerando apenas vizinhança $\Gamma(i)$ de um vértice, que corresponde ao grupo de vértices que se conectam a i através de uma aresta, é possível extrair a medida de grau do vértice. O grau k_i é dado pelo número de vizinhos de um vértice:

$$k_i = |\Gamma(i)|$$

Além disso, para redes não direcionadas, é possível escrever o grau dos vértices da rede em termos da matriz de adjacência:

$$k_i = \sum_{j=1}^{|V|} A_{ij}$$

Considerando todos os vértices da rede é possível extrair o grau médio dos vértices da rede c (NEWMAN, 2010):

$$c = \frac{1}{|V|} \sum_{i=1}^{|V|} k_i$$

Algumas redes possuem arestas que apresentam certas características como pesos. Redes que apresentam arestas com pesos, ou redes ponderadas, são dadas pela relação $G = (V, E, \omega)$, sendo que ω corresponde a uma função que atribui um peso real para cada aresta da rede. A matriz de adjacência A para uma rede ponderada pode ser derivada da matriz de adjacência vista anteriormente, com a diferença que, quando existe conexão entre i e j , o valor 1 é substituído por $\omega_{i,j}$.

Uma característica importante de redes é a existência de caminhos entre vértices. Um caminho é uma sequência qualquer de arestas que conectam uma sequência de vértices. Um caminho pode ser representado pela sequência de vértices $P = \{v_1, v_2, v_3, \dots, v_n\}$ que formam o caminho. A Figura 1 mostra um exemplo de rede em que é possível observar vários caminhos de diferentes tamanhos entre os vértices 1 e 2. O caminho $P_1 = \{1, 2\}$ que conecta diretamente os vértices com apenas uma aresta é o caminho geodésico entre tais vértices. O caminho geodésico, em uma rede não ponderada, é o caminho que contém o número mínimo de arestas para chegar de um vértice até outro na rede. O caminho geodésico mais longo entre todos os vértices na rede é denominado diâmetro da rede. Em uma rede ponderada, o caminho geodésico é aquele que apresenta o conjunto de arestas com o menor somatório de pesos. O tamanho $\mathcal{L}$ de um caminho é dado pelo número de arestas que formam o caminho, portanto o tamanho $\mathcal{L}(P_1) = 1$, e o tamanho do caminho $P_2 = \{1, 6, 3, 2\}$ é 3. Considerando ainda os vértices 1, 6, 3 e 2, saindo de 1 para 6, 3 e depois 2 é possível chegar novamente no vértice 1. Essa situação configura um caminho fechado, ou *loop*.

2.1.1 Coeficientes estruturais

As redes apresentam certas características que quando extraídas colaboram para um melhor entendimento de sua estrutura. São apresentadas nesta subseção quatro coeficientes que ilustram certas características importantes das redes e que serão avaliados no Capítulo 4. O primeiro coeficiente a ser apresentado é o coeficiente de densidade da rede. Considerando a existência de um conjunto U que corresponde ao conjunto de arestas de uma rede totalmente conectada, temos que a medida de densidade da rede G é dada pela razão entre o número de arestas da rede, ou cardinalidade do conjunto E , pelo número total de arestas possíveis na rede, ou seja, dado pela cardinalidade do conjunto universal U :

$$d_G = \frac{|E|}{|U|}$$

A densidade da rede representa a proporção de arestas possíveis que estão realmente presentes na rede. O valor 0 de densidade remete à uma rede sem quaisquer arestas e o valor 1 a uma rede totalmente conectada. Redes com baixo valor de densidade são também chamadas de redes esparsas.

Uma característica importante das redes é a transitividade. De acordo com (NEWMAN, 2010), a transitividade em uma rede pode ser caracterizada da seguinte forma: se o vértice

v conhece o vértice w e w conhece u , existe um caminho $P = \{v, w, u\}$ tal que $\mathcal{L}(P) = 2$, além disso, se v também conhece u tem-se um caminho fechado, ou triângulo. Quanto maior a quantidade triângulos, maior a transitividade da rede. Utilizando essas informações é possível definir um coeficiente que quantifica a transitividade de uma rede, o coeficiente de *clustering* ou agrupamento, que é dado por:

$$C_G = \frac{\text{Caminhos fechados de tamanho dois (ou triângulos)}}{\text{Total de caminhos de tamanho dois}}$$

O valor de $C = 1$ implica em transitividade perfeita e $C = 0$ implica na não existência de caminhos fechados de tamanho dois. Finalmente, o coeficiente de assortatividade da rede baseia-se na correlação entre propriedades escalares dos vértices adjacentes (NEWMAN, 2002). Esse coeficiente pode ser extraído de qualquer propriedade escalar dos vértices presente na rede, como idade por exemplo. Porém, essas informações nem sempre existem ou estão disponíveis para as redes que estão sendo consideradas. Nesse caso, pode-se utilizar uma propriedade topológica que está sempre disponível para todas as redes, que é o grau dos vértices. Seja p_k a distribuição de grau dos vértices em uma rede não direcionada, ou seja, a probabilidade de que um vértice com o grau k seja aleatoriamente selecionado. Quando uma aresta da rede é aleatoriamente selecionada, o grau dos dois vértices conectados por esta aresta não segue mais a distribuição p_k , pois a seleção da aresta sofre um viés causado pelos vértices de grau mais alto, uma vez que mais arestas conectam vértices de grau alto do que de grau baixo. Os graus dos dois vértices conectados pela aresta selecionada aleatoriamente seguem então uma distribuição que é proporcional a kp_k , que é chamada de q_k . Define-se então a forma normalizada para a distribuição de grau q_k :

$$q_k = \frac{(k+1)p_{k+1}}{\sum_j jp_j}$$

De acordo com (NEWMAN, 2002), O valor 1 é somado pelo que se define por grau remanescente do vértice, que é o número de arestas que incidem no vértice sem considerar a aresta que foi selecionada aleatoriamente e que indiretamente fez com que os vértices fossem selecionados. Considerando e_{jk} como a distribuição de probabilidade conjunta dos graus remanescentes dos dois vértices incidentes à uma aresta aleatoriamente selecionada, o coeficiente de assortatividade da rede, que corresponde ao coeficiente de correlação de *Pearson* dos graus dos vértices incidentes à uma aresta, é dado por:

$$r = \frac{\sum_{jk} jk(e_{jk} - q_j q_k)}{\delta_q^2}$$

O valor δ_q^2 corresponde à variância da distribuição q_k . Este coeficiente é importante pois representa a preferência que vértices de grau alto têm em se conectar com outros vértices de grau alto. Entretanto, diferentemente do modelo de conexão preferencial, ou *Preferential Attachment* (PA), o índice de assortatividade também leva em consideração o grau do vértice de origem da aresta. A próxima seção apresenta os fundamentos do problema de predição de *links*.

2.2 Predição de links

De acordo com (GETOOR; DIEHL, 2005), a tarefa de predição de *links* pode ser classificada como uma tarefa de mineração de *links*. Ainda de acordo com (GETOOR; DIEHL, 2005), a predição de *links* é o problema de predizer a existência de uma conexão entre duas entidades baseado nos atributos entre os objetos e outros *links* já existentes. A Figura 2 apresenta um exemplo de predição de *links*. A rede à esquerda trata de uma rede que possui vértices e arestas que serve de entrada para um algoritmo de predição de *links*. A rede à direita apresenta algumas arestas preditas, representadas por linhas tracejadas, através de um algoritmo de predição de *links*.

Figura 2 – Exemplo de aplicação de um algoritmo de predição de *links*.

Fonte: Elaborada pelo autor.

Existem diversas aplicações de predição de *links* em diversos cenários. De acordo com (MARTÍNEZ; BERZAL; CUBERO, 2016), qualquer domínio que apresente interação entre entidades de uma forma estruturada pode se beneficiar da predição de *links*. Considerando uma rede simples, ou seja, que não possui arestas laço ou arestas múltiplas, e ponderada $G(V, E, \omega)$, temos o conjunto universal U que possui todas as arestas possíveis para aquela rede:

$$|U| = \frac{|V| \times (|V| - 1)}{2}$$

A situação em que $U = E$ não é comum e em algumas aplicações é desejável encontrar arestas promissoras pertencentes ao conjunto $U - E$, seja para completar a rede com informações faltantes ou para predizer quais arestas são mais prováveis de existirem na rede em um momento futuro. O conjunto $U - E$ é denominado de conjunto de arestas não observadas. Para tais tarefas, existem algoritmos de predição de *links*. As arestas $(v, u) \in U - E$ são denotadas por arestas não observadas e o preditor de *links* geralmente atribui, para cada aresta não observada, um valor numérico, ou *score*, que representa a qualidade de tal aresta. Como para cada aresta é atribuído

um valor de *score*, o preditor de *links* tem como saída um *ranking* de arestas preditas em que no topo ficam as arestas mais promissoras de acordo com o preditor.

2.2.1 Abordagens

De acordo com (LU; ZHOU, 2010), os algoritmos mais simples de predição de *links* são os algoritmos baseados em similaridade, em que para cada par de vértices v e u , que não tenham uma aresta conectando ambos, é atribuído um valor que representa a similaridade entre os dois vértices. De acordo com (MARTÍNEZ; BERZAL; CUBERO, 2016), a predição de *links* baseia-se na evidência empírica de que duas entidades são mais susceptíveis de interagir se forem semelhantes. Portanto, para tais algoritmos, arestas não observadas que conectam vértices muito similares têm uma alta probabilidade de serem arestas promissoras.

Algoritmos baseados em similaridade utilizam índices que calculam a similaridade entre vértices por meio de atributos intrínsecos ou através de características estruturais da rede. Apesar de atributos intrínsecos conterem informações importantes de domínio e sua utilização levar a bons desempenhos de predição (GE; ZHANG, 2012; ZHANG *et al.*, 2012), tais atributos nem sempre estão disponíveis, enquanto que as características estruturais podem ser extraídas de qualquer rede. O índice mais simples que utiliza características estruturais da rede para extrair similaridades entre vértices é o *Common Neighbors* (CN), definido pelo número de vizinhos em comum de dois vértices (LORRAIN; WHITE, 1971). Para os vértices v e u , temos que:

$$S_{vu}^{CN} = |\Gamma(v) \cap \Gamma(u)|$$

Índices como *CN* são conhecidos como índices de similaridade local, pois apenas caminhos curtos são considerados. Alguns outros índices locais bastante utilizados são variações normalizadas do índice *CN*, tais como:

- Índice de *Jaccard* (JACCARD, 1901) é um índice bastante utilizado para predição de *links* que normaliza o índice *CN*. Pares de vértices que compartilham muitos em proporção ao seu número total de vizinhos recebem altos valores de similaridade:

$$S_{vu}^{JAC} = \frac{|\Gamma(v) \cap \Gamma(u)|}{|\Gamma(v) \cup \Gamma(u)|}$$

- Índice de *Salton* (SALTON; MCGILL, 1986) é um índice que normaliza *CN* utilizando o grau dos vértices para tal. Este índice é uma medida de similaridade de cosseno para medir a similaridade de um par de vértices. Pares de vértices com maior grau terão um valor mais baixo de similaridade:

$$S_{vu}^{SAL} = \frac{|\Gamma(v) \cap \Gamma(u)|}{\sqrt{k_v \times k_u}}$$

- Índice de *Adamic-Adar* (ADAMIC; ADAR, 2003) atribui valores altos de similaridade para vértices que compartilham vizinhos com baixo valor de grau. O índice utiliza a função log para suavizar os valores de similaridade:

$$S_{vu}^{AA} = \sum_{z \in |\Gamma(v) \cap \Gamma(u)|} \frac{1}{\log(k_z)}$$

Os índices mostrados anteriormente são denominados índices de similaridade local, pois apenas caminhos curtos e vizinhos diretos são considerados. Apesar de apresentarem bons resultados, tais índices consideram apenas a vizinhança mais próxima do vértice. Alguns outros índices consideram caminhos de tamanhos maiores, além de características globais da rede, e são denominados índices de similaridade global. Um dos índices de similaridade global que apresenta os melhores resultados de predição é o índice de similaridade de *Katz* (KATZ, 1953):

$$S_{vu}^{Katz} = \sum_{\mathcal{L}=1}^{\infty} \beta^{\mathcal{L}} \cdot |paths_{vu}^{<\mathcal{L}>}|$$

O índice de similaridade de *Katz* é baseado no conjunto dos caminhos de todos os tamanhos entre os vértices v e u e não apenas os caminhos entre vizinhos diretos, como *CN*. O termo $paths_{vu}^{<\mathcal{L}>}$ representa todos os caminhos de tamanho $\mathcal{L}$ entre os vértices v e u e β é um parâmetro que permite que pesos maiores sejam dados para caminhos menores.

Baseando-se no conceito que dois vértices são similares se seus vizinhos também são similares, os autores de (JEH; WIDOM, 2002) propuseram um índice global recursivo bastante utilizado, o *SimRank*. O índice é definido através de uma equação recursiva em que a similaridade de dois vértices $S_{vu}^{SR} = 1$ se $v = u$ e caso $v \neq u$ tem-se:

$$S_{vu}^{SR} = C \frac{\sum_{a \in \Gamma(v)} \sum_{b \in \Gamma(u)} S_{a,b}^{SR}}{k_v k_u}$$

Em que C é uma constante entre 0 e 1 que define a taxa de decaimento de como a similaridade flui entre as arestas. Índices globais como *Katz* e *SimRank* podem apresentar resultados superiores de predição aos resultados de índices locais como *CN* (LIBEN-NOWELL; KLEINBERG, 2003; LU; ZHOU, 2010). Entretanto, o cálculo de índices globais como *Katz* e *SimRank* é muito mais custoso computacionalmente, tanto em termos de tempo de execução quanto de memória, e geralmente são inviáveis em redes com dezenas de milhares de vértices. Neste trabalho o índice local *CN* e o índice global *Katz* são utilizados na avaliação experimental pelos algoritmos de otimização e predição de *links*, respectivamente. O índice *Katz* foi escolhido ao invés do *SimRank* pois apresenta também uma alta complexidade, uma vez que para extrair as similaridades entre os vértices, a matriz de adjacência da rede precisa ser armazenada e uma matriz inversa precisa ser computada.

Outra tipo de abordagem bastante comum na literatura de predição de *links* é tratar o problema como um problema de classificação binária, sendo que o trabalho (HASAN *et al.*, 2006)

é o trabalho pioneiro com tal abordagem e (BENCHETTARA; KANAWATI; ROUVEIROL, 2010) foi o primeiro trabalho a aplicar aprendizado supervisionado para predição de *links* em redes bipartidas. A abordagem se caracteriza pela utilização de um conjunto de características, ou atributos, que é extraído da rede e utilizado como conjunto de treinamento para a construção de algum modelo supervisionado de classificação. Para construir tais modelos, geralmente são utilizados algoritmos de construção de redes neurais, árvores de decisão e máquinas de vetores de suporte. Uma vez que o modelo é construído, ele é utilizado para classificar arestas não observadas em prováveis de pertencer à rede ou improváveis. Os primeiros atributos extraídos não levavam a conectividade das redes em consideração. Em (HASAN *et al.*, 2006), os atributos utilizados são extraídos de redes de colaboração e, apesar de apresentar bons resultados, são específicos para a predição de novas colaborações em redes de colaborações científicas. Alguns outros trabalhos posteriores como (CUKIERSKI; HAMNER; YANG, 2011), (De Sá; PRUDÊNCIO, 2011) e (CHOUDHURY; UDDIN, 2016) utilizaram índices e medidas provenientes do estudo de redes como atributos para compor o conjunto de treinamento e assim obter melhor aproveitamento da conectividade da rede.

2.2.2 Avaliação

Para avaliar preditores de *links* é necessário utilizar informações existentes na rede e a forma mais simples de fazer tal avaliação é utilizando medidas de desempenho. Para utilizar tais medidas com informações existentes na rede para avaliar o preditor de *links* é necessário separar os conjuntos de arestas em dois conjuntos, um que será utilizado pelo preditor para prever novas arestas e outro ao qual o preditor não terá acesso e será utilizado apenas para avaliação, uma vez que o *ranking* de predições já fora extraído. Dessa forma, o conjunto de arestas E é dividido em dois subconjuntos, E^{tr} e E^{ts} , tal que $E = E^{tr} \cup E^{ts}$ e $E^{tr} \cap E^{ts} = \emptyset$. Todas as arestas pertencentes a E^{ts} são removidas e a rede remanescente $G'(V, E^{tr}, \omega)$ é então utilizada pelo algoritmo como entrada para gerar as predições. O conjunto E^{ts} é utilizado como um conjunto de avaliação com a finalidade de calcular as medidas de desempenho e, assim, avaliar a qualidade das predições feitas. Na realização dos experimentos deste projeto foram utilizadas duas medidas de desempenho. A primeira medida utilizada é *precisi@L*, dada por:

$$precisi@L = \frac{L_r}{L}$$

Preditores de *links* produzem um *ranking* decrescente de arestas preditas ordenado de forma decrescente pelo valor de saída dado pelo preditor para cada aresta, em que as arestas no topo do *ranking* são as arestas que o preditor atribuiu maior *score*. Para calcular a medida *precisi@L*, as primeiras L entradas do *ranking* são extraídas para formar um subconjunto E^f . O termo L_r corresponde ao número de arestas em $E^f \cap E^{ts}$. Se valores altos de *score* forem atribuídos para arestas do conjunto E^{ts} , isso é um indício que o preditor de *links* tem um bom desempenho, pois é capaz de atribuir valores altos para arestas que de fato existiram na rede e foram retiradas. A

outra medida de desempenho utilizada foi a *Area Under ROC Curve* (AUC), dada por:

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

A *AUC* pode ser interpretada como a probabilidade de uma aresta removida, pertencente a E^{ts} , escolhida aleatoriamente, ter um *score* maior do que uma aresta não observada, ou seja, pertencente a $U - E$. Temos que n' corresponde ao número de vezes que arestas removidas que possuem um valor de *score* maior, n'' sendo o número de vezes que o valor do *score* é o mesmo e n como sendo o número total de comparações. Quanto maior o valor da medida de *AUC*, maior a acurácia do algoritmo de predição de *links*.

2.2.3 Aplicações de predição de links

A tarefa de predição de *links* pode ser utilizada para diversas tarefas em vários ambientes. Esta subseção apresenta algumas aplicações de métodos de predição de *links*.

- **Predição em redes sociais:** A aplicação de predição de *links* com maior representatividade na literatura é a predição de novos relacionamentos em redes sociais. A utilização de preditores de *links* para a criação de novos relacionamentos em redes sociais é utilizada em diversos ambientes diferentes, como predição de novas amizades em redes sociais (LIBEN-NOWELL; KLEINBERG, 2003), sugestões de novas parcerias entre autores em redes de co-autoria (PAVLOV; ICHISE, 2007), no auxílio contraterrorismo (ANIL *et al.*, 2015).
- **Predição em redes biológicas:** A predição de *links* também é utilizada em algumas aplicações biológicas, principalmente relacionadas às predições de interação proteína-proteína. De acordo com (IAKOVIDOU; SYMEONIDIS; MANOLOPOULOS, 2010), as redes de interações entre proteínas (*protein-protein interactions*) são as redes mais intensivamente estudadas na biologia e, como as técnicas de biologia molecular para explorar tais interações são caras e demoradas, é preferível aplicar técnicas da teoria dos grafos para auxiliar nos estudos com tais redes. O trabalho (IAKOVIDOU; SYMEONIDIS; MANOLOPOULOS, 2010) apresenta a abordagem *BioSpectral* para predição de interações em tais redes, que extrai os autovetores da matriz de adjacência da rede de proteínas e depois aplica o algoritmo *k-means* para dividir os vértices em *clusters*. A abordagem então calcula a similaridade entre vértices de um mesmo *cluster* e cria um *ranking* de novas possíveis interações com os pares de vértices mais similares. Em (LEI; RUAN, 2013), os autores propõem uma abordagem de predição de *links* para a redução de ruído e esparsidade em dados de redes de interação entre proteínas. A abordagem proposta se baseia em *random walk*, porém com a introdução de viés no algoritmo tradicional, de forma que a caminhada se mantenha mais próxima ao ponto de início.

- **Aplicações médicas:** Alguns trabalhos também utilizam algoritmos de predição de *links* para aplicações médicas. Nos trabalhos (KAYA; POYRAZ, 2014; KAYA; POYRAZ, 2016), os autores utilizam dados médicos com informações sobre resultados de testes laboratoriais para construir uma rede de sintomas. Cada vértice da rede corresponde a um parâmetro anormal (fora do intervalo de referência) que pode indicar a presença de alguma doença. Arestas entre vértices representam a presença simultânea dos parâmetros anormais em um mesmo paciente e a estrutura de relação entre sintomas é então utilizada para prever futuros sintomas. Em (LI *et al.*, 2017) os autores utilizam a predição de *links* para corrigir arestas faltantes e para remover arestas espúrias de redes conectividade funcional, que são redes que modelam o cérebro como uma rede complexa, de forma que regiões do cérebro são representadas por vértices e arestas representam conexões funcionais entre tais regiões. No trabalho (LU; GUO; KORHONEN, 2017), a predição de *links* é utilizada para o reposicionamento de drogas, ou seja, o descobrimento de novos usos para medicamentos existentes. Os autores utilizam índices locais para prever novas interações em uma rede com interações entre medicamentos e usos.
- **Sistemas de recomendação:** A aplicação mais popular e comercial de algoritmos de predição de *links* é em sistemas de recomendação. A tarefa de recomendação consiste em criar um sistema, que atua em um serviço, que seja capaz de prever a utilidade que itens dentro de um catálogo terão para os usuários daquele serviço. Sistemas de recomendação eficientes contribuem para que os usuários consigam lidar com o excesso de itens e também para aumentar as oportunidades de consumo, facilitando o acesso por parte do usuário aos itens do seu gosto. Os trabalhos (HUANG; LI; CHEN, 2005; ESSLIMANI; BRUN; BOYER, 2011) utilizam algoritmos de predição de *links*, como similaridade entre vértices, em redes sociais para construir sistemas de recomendação baseados em filtragem colaborativa. Algumas abordagens mais sofisticadas como a presente em (LI; CHEN, 2013) utilizam algoritmos supervisionados para predição através de atributos extraídos da estrutura da rede.

Algumas outras aplicações interessantes incluem aplicações militares, como a presente em (FAN *et al.*, 2016) em que a predição de *links* é utilizada para remover arestas espúrias e corrigir arestas faltantes em redes de operações. Embora seja aplicável em diversos ambientes para várias aplicações, é preciso levar em conta a escalabilidade nos algoritmos de predição de *links*. Mesmo para algoritmos de similaridade local de baixa complexidade computacional, como CN, algumas redes tornam o processo de predição proibitivo devido à quantidade de vértices, e consequentemente de arestas não observadas. A próxima subseção apresenta abordagens variadas da literatura para melhorar a escalabilidade de algoritmos de predição de *links*.

2.3 Trabalhos relacionados com escalabilidade em predição de links

Poucos trabalhos abordam a questão de escalabilidade em predição de *links* e a maioria das propostas estão relacionadas a considerar apenas subgrafos da rede ou exploram apenas algoritmos locais, devido à sua baixa complexidade computacional, o que pode prejudicar a qualidade das predições. Não foram encontrados trabalhos que utilizam métodos de otimização multinível para problemas de predição de *links*, portanto trabalhos sobre escalabilidade em predição de *links* foram considerados os trabalhos relacionados mais próximos.

Um dos primeiros trabalhos que considera a escalabilidade em predição de *links* é (FIRE *et al.*, 2011). O trabalho propõe alguns atributos que podem ser calculados com um baixo custo computacional para então construir um modelo de aprendizado supervisionado. Os autores propõem atributos estruturais baseados no grau dos vértices, baseados no subgrafo formado pela vizinhança dos vértices e em medidas de similaridade de baixa complexidade. Em (SCELLATO; NOULAS; MASCOLO, 2011) os autores propõem uma forma para redução do espaço de busca de arestas não observadas para também construir um modelo supervisionado. Embora a abordagem proposta seja dependente do ambiente em que é aplicada, recomendação de redes sociais *online* geolocalizadas, os autores conseguiram reduzir o espaço de busca em cerca de 15 vezes. Para tal, um subgrafo é selecionado em uma rede bipartida de usuários e locais, fazendo com que só seja possível prever uma aresta entre um usuário e um local se este local possuir alguma conexão com algum vizinho ou com vizinho de vizinho deste usuário.

A abordagem *FriendLink*, proposta em (PAPADIMITRIOU; SYMEONIDIS; MANOLOPOULOS, 2012), explora caminhos de tamanho maior do que os algoritmos de similaridade local, apresentados na Subseção 2.2.1, porém mantendo certa escalabilidade no método. Limitando o tamanho máximo dos caminhos buscados em $\mathcal{L}$, a abordagem consegue buscar caminhos maiores do que as abordagens de similaridade local tradicionais e não sofre com a proibitividade de algoritmos de similaridade global como *Katz*, que considera caminhos de todos os tamanhos. O trabalho (COHEN; COHEN-TZEMACH, 2013) não apresenta novas abordagens para predição de *links*, entretanto apresenta a implementação de algoritmos tradicionais utilizando diferentes sistemas de bancos de dados e faz comparações de tempo de execução dos algoritmos em cada um dos sistemas. Em (CORBELLINI *et al.*, 2016) os autores propõem um *framework* distribuído para processamento de redes de larga escala para predição de *links*.

Os autores de (SYMEONIDIS; MANTAS, 2013) utilizam uma abordagem de clusterização espectral que, segundo os autores, consegue resultados superiores à abordagens locais e também maior eficiência do que métodos globais tradicionais. O método é baseado nos autovetores e autovalores principais da matriz Laplaciana normalizada extraída de redes sociais. Ao invés de utilizar métodos tradicionais para resolução do problema de autovalores, que têm complexidade de $\mathcal{O}(|V|^3)$ e tornaria a predição proibitiva, os autores aproveitam das caracterís-

ticas de esparsidade e conectividade local das redes sociais, o que consequentemente faz com que apenas os principais autovetores sejam necessários, para empregar métodos mais rápidos e tornar o algoritmo mais escalável para redes maiores. O trabalho (OGATA; SUZUMURA, 2014) não apresenta novas abordagens para predição de *links*, porém apresenta implementações paralelizadas, na linguagem de programação especializada *X10*, de métodos locais, aumentando a escalabilidade dos métodos para redes maiores. Para tal, os autores extraem uma matriz de similaridade S da matriz de adjacência da rede através de índices de similaridade local e decompõem a matriz com aproximação de matrizes com posto reduzido. O trabalho (AHMED *et al.*, 2016) utiliza um método de *random walk* local para criar um subgrafo de vizinhança para um determinado vértice e assim aplicar a medida de similaridade apenas neste subgrafo, reduzindo substancialmente o tempo de execução do preditor de *links*. Os autores conseguem reduzir a complexidade computacional da predição de $\mathcal{O}(|V|^3)$ para $\mathcal{O}(|V|^2)$, além de conseguir também um bom desempenho de predição.

Em (COSKUN; KOYUTURK, 2016), motivados pelo melhor desempenho de métodos que exploram características globais das redes, os autores propõem técnicas de redução de dimensionalidade que levam em conta a modularidade e esparsidade das redes de co-autoria utilizadas no estudo. Os autores extraem, para cada vértice v , um perfil topológico global que é composto pela proximidade de v com todos os outros vértices em V , entretanto como o perfil topológico global é obtido através de medidas globais, seu uso seria proibitivo para redes com dezenas de milhares de vértices. Para tornar o método utilizável em tais redes os autores empregam duas abordagens para redução de dimensionalidade. A primeira utiliza a característica de modularidade através de algoritmos de clusterização, de forma que um perfil topológico global de um *cluster* possa ser utilizado para representar os vértices densamente conectados que a ele pertencem, e o algoritmo utiliza o perfil topológico dos *clusters* para inferir um índice de similaridade. A segunda abordagem utiliza a característica de esparsidade das redes, de forma que nós muito distantes não têm muita influência sobre os outros para a criação de novas arestas. Um parâmetro é utilizado para filtrar os vértices que serão considerados para a predição de acordo com seu perfil topológico global.

Os autores de (CUI *et al.*, 2016) propuseram dois lemas para propor um algoritmo e filtrar pares de vértices da rede utilizando *CN*. O primeiro lema utiliza um parâmetro de limite inferior L de forma que todo vértice v que tenha um grau $k(v) < L$ seja ignorado pelo preditor de *links*. Após a primeira filtragem da rede, os autores utilizam outro lema para reduzir ainda mais o número de pares que não contribuem para a predição: se um vértice aparece em no máximo L adjacências de outros vértices, este vértice também é removido. Além de remover muitas arestas não observadas, a utilização dos dois lemas propiciou a implementação da abordagem de forma paralela, tornando-a ainda mais aplicável para redes de larga escala. Em (MOHAN; VENKATESAN; PRAMOD, 2017) os autores propõem uma abordagem paralela do índice de *Adamic Adar* (ADAMIC; ADAR, 2003) que incorpora informações de comunidade dos vértices para um melhor desempenho de predição.

As abordagens provenientes dos trabalhos apresentados possuem algumas características limitantes. Limitar o conjunto de arestas não observadas pode causar a perda de arestas importantes, uma vez que torna o processo de predição dependente do método de exclusão que limitou quais arestas não observadas seriam consideradas. Além disso, tornar o algoritmo dependente de tecnologias de processamento paralelo pode dificultar a implementação e utilização do mesmo. E por fim, algumas abordagens para aumentar a escalabilidade dos preditores de *links* são totalmente dependentes dos algoritmos em que foram criadas. Em contraste com o que foi apresentado, as abordagens desenvolvidas nesta pesquisa para diminuir o tempo de execução necessário à predição de *links* são independentes do algoritmo de predição e podem ser aplicadas a qualquer método.

2.4 Otimização multinível

Utilizado originalmente de forma a reduzir o número de operações realizadas por algoritmos de particionamento, o paradigma multinível oferece uma forma de contrair a rede, condensando vértices conectados por alguma aresta pela remoção de tais arestas, gerando sucessivas redes contraídas, realizar alguma operação nessa rede reduzida, que geralmente é executar algum algoritmo objetivo em que a aplicação na rede sem contrações seria impedida pela complexidade estrutural da rede, e depois projetar os resultados obtidos pela operação de volta à rede original. A abordagem se divide em três fases, sendo que cada fase é realizada sequencialmente na rede: contração, ou *coarsening*, busca pela solução inicial e projeção, ou *uncoarsening* (KARYPIS; KUMAR, 1998). A Figura 3 apresenta de forma genérica as fases para a resolução de problemas utilizando a abordagem multinível.

Figura 3 – Fases da abordagem multinível para solução de problemas em redes.

Fonte: Adaptada de Valejo *et al.* (2018).

A primeira fase diz respeito à seleção de arestas para composição do *matching* e contração da rede, para gerar uma rede reduzida. O termo *matching* é comumente utilizado em outras áreas de

pesquisa no contexto de redes, para designar conjuntos de pares de arestas que não compartilham vértices comuns. Na literatura de otimização multinível em redes, o termo *matching* refere-se a um conjunto que possui arestas que serão selecionadas para serem removidas da rede e cujos vértices adjacentes serão condensados em super-vértices para realizar uma contração na rede.

Considerando uma rede inicial G_0 , com um conjunto de vértices V_0 e um conjunto de arestas E_0 , a rede é contraída em redes menores $G_1, G_2, \dots, G_n$ de forma que $|V_0| > |V_1| > |V_2| > \dots |V_n|$. A rede G_1 é dita como rede no nível 1 de contração. Algumas arestas pertencentes ao conjunto E_0 são selecionadas para fazerem parte do conjunto de *matching*, conjunto este que será utilizado como entrada para o processo de contração da rede. Os dois vértices unidos pela aresta que foi removida são condensados em um super-vértice. Os super-vértices gerados na rede G_i tornam-se vértices da rede G_{i+1} . Nem todos os vértices da rede G_i são necessariamente combinados com outros em super-vértices. Os que não são unidos a outros vértices são herdados pela rede do próximo nível.

Todas as arestas da rede G_0 que foram removidas e cujo os vértices incidentes foram unidos em super-vértices compõem o conjunto denominado *matching*, ou conjunto de arestas selecionadas. Para gerar o conjunto de *matching* é necessário um processo de seleção que escolhe quais arestas serão removidas da rede no próximo nível e, conseqüentemente, quais vértices incidentes àquelas arestas serão condensados em super-vértices. De acordo com (KARYPIS; KUMAR, 1998), a contração de redes usando *matching* preserva várias propriedades da rede original e, no mesmo trabalho, os autores apresentam os seguintes algoritmos para seleção de arestas, ou construção do *matching* que foram utilizados nesta pesquisa, além dos propostos por nós:

- *Random Matching* (RM): Neste algoritmo, os vértices da rede são visitados aleatoriamente. Se um vértice v não tiver sido contraído ainda, um de seus vizinhos que também não tenha sido contraído é escolhido aleatoriamente. A aresta entre os dois vértices é então selecionada e adicionada ao conjunto *matching*. Caso v não tenha vizinhos que ainda não tenham sido contraídos, v continua a ser um vértice da rede e será herdado pela rede no próximo nível. Este algoritmo apesar de simples apresenta bons resultados e possui complexidade linear $O(|E|)$.
- *Heavy Edge Matching* (HEM): De maneira similar ao RM, os vértices são também visitados em ordem aleatória. Entretanto, diferentemente do algoritmo RM, os vértices v e u são unidos em um super-vértice apenas se a aresta (v, u) possui o peso máximo considerando todas as arestas válidas, ou seja, que ainda não tenham sido selecionadas, incidentes ao vértice v . Embora o algoritmo HEM não garanta que o *matching* obtido tenha o peso máximo devido à sua abordagem gulosa, nos problemas de particionamento o algoritmo gera melhores resultados do que o algoritmo RM tendo a mesma complexidade computacional (KARYPIS; KUMAR, 1998). Embora diminua o somatório do peso das arestas de G_{i+1} ,

esse algoritmo não garante a minimização do mesmo. Em (KARYPIS; KUMAR, 1995), os autores mostram que a escolha das arestas com maior peso gera melhores resultados na aplicação de algoritmos de particionamento. Da mesma forma que o algoritmo anterior, o algoritmo *HEM* apresenta complexidade linear $O(|E|)$.

- *Light Edge Matching* (LEM): Seu funcionamento é inverso ao do algoritmo *HEM* visto anteriormente. Ao invés de selecionar a aresta de maior peso do vértice visitado, este algoritmo seleciona a que possui o menor peso. Isso faz com que o somatório de pesos do grafo G_{i+1} aumente se comparado à seleção aleatória do algoritmo *RM* ou da seleção do algoritmo *HEM*. O algoritmo *LEM* também possui complexidade linear $O(|E|)$.

O conjunto selecionado pelo processo de *matching* é então utilizado como entrada para o processo de contração, que une os vértices em super-vértices, eliminando as arestas entre os mesmos. Caso os vértices unidos tenham algum vizinho em comum, as arestas com este vizinho são contraídas em uma única aresta, criando uma super-aresta, ligando o super-vértice ao vizinho, e o peso desta aresta é dado pelo somatório dos pesos das arestas contraídas. Terminadas as n iterações da fase de contração da rede, obtém-se uma rede reduzida G_n . O algoritmo objetivo é então aplicado na rede reduzida de forma a encontrar uma primeira solução que é projetada para as redes maiores $G_{n-1}, G_{n-2}, \dots, G_0$. De acordo com (WALSHAW, 2004), teoricamente a abordagem multinível pode ser empregada para resolver qualquer problema de otimização combinatória, entretanto na literatura de redes o uso mais comum da abordagem multinível é para o problema de particionamento de redes. Em (KARYPIS; KUMAR, 1998), os autores utilizam o paradigma para executar algoritmos de bipartição espectral e algoritmos heurísticos, que minimizam o valor da função de *cut size* para particionar redes.

Na fase de projeção das soluções, o conjunto de vértices V_n é expandido para V_{n-1} , portanto todos os super-vértices que se formaram no processo de contração da rede V_{n-1} para V_n são decompostos, retornando a vértices ou ainda super-vértices, gerados pela união de vértices em uma rede V_{n-2} . Em trabalhos que utilizam a abordagem multinível para particionamento de grafos, os vértices que formam o super-vértice são adicionados à partição ao qual o super-vértice está inserido (WALSHAW, 2004).

A abordagem multinível tem sido aplicada para diversos problemas e algoritmos objetivos, além das já citadas abordagens clássicas para particionamento de grafos. Alguns exemplos são problemas de coloração de grafos (WALSHAW, 2001), problema do caixeiro viajante (WALSHAW, 2002), desenho e visualização de redes (WALSHAW, 2001), abordagens mais sofisticadas para particionamento de grafos (BICHOT, 2014) e até mesmo na computação de medidas de centralidade em redes (CHERNOSKUTOV; INEICHEN; BEKAS, 2015). Entretanto, até o desenvolvimento desta pesquisa, a abordagem multinível não tinha sido empregada para algoritmos de predição de *links*.

OTIMIZAÇÃO MULTINÍVEL EM PREDIÇÃO DE LINKS

Neste capítulo é apresentada a proposta que foi criada para a redução do número de operações necessárias por preditores de *links*, que foi implementada e avaliada neste projeto, de forma a responder as questões de pesquisa e obter os resultados necessários para as análises. A proposta trata-se de uma abordagem de otimização multinível para predição de *links* em redes simples não-direcionadas.

A abordagem desenvolvida neste trabalho pode ser aplicada juntamente com qualquer algoritmo de predição de *links*, uma vez que a abordagem é independente do algoritmo de predição. O processo de otimização multinível consiste no *coarsening* da rede original, reduzindo-a até um nível satisfatório n , na predição de *links* na rede mais contraída G_n e a projeção, ou *uncoarsening*, dos resultados de predição para os vértices pertencentes à rede original. A Figura 4 ilustra o processo da abordagem proposta.

Para cada quadro que compõe a Figura 4 temos:

1. Uma rede G_0 que tenha $U \neq E$ e que será utilizada como entrada por um algoritmo de predição de *links*;
2. O processo de *coarsening* contraiu iterativamente a rede original G_0 em uma rede G_n , contraída em n níveis por meio de algoritmos de *matching* e do processo de contração. Tal rede possui conjuntos reduzidos $|V_n| < |V_0|$ e $|E_n| < |E_0|$ de vértices e arestas, respectivamente;
3. O preditor de *links* é aplicado na rede G_n e as arestas potenciais, destacadas no quadro pelas linhas tracejadas, recebem seus valores de *score*. O processo de predição na rede G_n é mais rápido do que seria o mesmo processo de predição na rede G_0 devido ao número reduzido de vértices e consequentemente de arestas não observadas;

Figura 4 – Abordagem de otimização multinível para predição de *links* em redes simples.

4. As super-arestas preditas entre os super-vértices da rede contraída G_n são projetadas para os vértices da rede original, que foram condensados em super-vértices, através do algoritmo de *uncoarsening*. Neste ponto do processo, o preditor que se beneficiou da abordagem multinível tem como saída um *ranking* de arestas preditas. Entretanto, este *ranking* não necessariamente é igual ao que seria o *ranking* da aplicação direta do algoritmo de predição de *links* na rede sem contrações. Como as predições foram extraídas de uma versão reduzida da rede, são esperadas algumas perdas de qualidade nas predições.

Uma das fases do processo de otimização multinível é o *uncoarsening*, no qual a solução inicial encontrada na rede contraída é projetada para a rede original. Geralmente, para cada passo do processo de *uncoarsening*, os resultados obtidos na rede G_i são projetados de volta para a rede G_{i-1} até que estejam projetados para a rede G_0 . Entretanto, nas abordagens propostas neste trabalho, os resultados são projetados diretamente da rede mais contraída G_n até a rede original G_0 , sem projeções intermediárias, gerando o *ranking* final de predições. Abaixo temos o Algoritmo 1 que incorpora a otimização multinível em algoritmos de predição de *links*.

Como um processo normal de predição de *links*, a predição multinível recebe como entrada uma rede que possui arestas não observadas e retorna um *ranking* de tais arestas com valores de *score* que indica a qualidade de tais arestas. Da linha 5 até 9 do Algoritmo 1, a rede passa pelo processo de *coarsening* e é reduzida em l níveis, parâmetro este que é passado ao algoritmo e, quanto maior o nível de *coarsening* menores serão os conjuntos V_i e E_i . Os vértices são condensados em super-vértices e arestas são condensadas em super-arestas. Para valores grandes de l , a rede é muito contraída, levando a resultados rápidos porém de baixa qualidade de acordo com os experimentos executados. Dessa forma, é necessário escolher um valor de l que

Algoritmo 1 – Predição multinível de *links* para redes simples

```

1: procedimento PREDICAO-MULTIVEL( $G, l$ )
2:    $G' \leftarrow G$ 
3:    $i \leftarrow 0$ 
4:    $R \leftarrow \emptyset$ 
5:   enquanto  $i \leq l$  faça
6:      $M' \leftarrow \text{MATCHING}(G')$ 
7:      $G' \leftarrow \text{CONTRACAO}(G', M)$ 
8:      $i \leftarrow i + 1$ 
9:   fim enquanto
10:   $R' \leftarrow \text{PREDICAO}(G')$ 
11:  para todo super-aresta predita  $sE = (sV, sU) \in R'$  faça
12:     $R \leftarrow R \cup \text{UNCOARSENING}(sE)$ 
13:  fim para
    retorna  $R$ 
14: fim procedimento

```

leve à perdas aceitáveis de desempenho. Na Linha 5 tem-se o início da execução do processo que vai gerar redes mais contraídas iterativamente. Cada iteração do bloco **enquanto** gera uma versão mais contraída da rede no nível i , através dos algoritmos de *matching* e contração. O processo de *matching* seleciona um conjunto de arestas que, de acordo com algum critério estabelecido no próprio algoritmo, devem deixar de existir na rede no próximo nível. Uma vez que tem-se o conjunto de *matching* M , o algoritmo de contração remove da rede todas as arestas $e_i = (v_1, v_2) \in M$ e une os vértices adjacentes v_1 e v_2 em super-vértices, gerando assim uma rede mais contraída $i + 1$. O processo é repetido até que tenha-se uma rede no nível l de contração.

Para o processo de *matching*, na Linha 6, cinco algoritmos foram utilizados nos experimentos. Os algoritmos *HEM*, *LEM* e *RM* descritos na Seção 2.4 e também dois novos algoritmos propostos e desenvolvidos neste trabalho. Os algoritmos propostos utilizam medidas de similaridade para selecionar os vértices que compõem o *matching*:

- *Most Similar Vertex Matching* (MSVM): um vértice v é aleatoriamente escolhido e, para cada vizinho que não foi selecionado para o *matching*, a similaridade é calculada usando um índice de similaridade que é passado ao algoritmo como parâmetro. O vizinho mais similar é selecionado juntamente com v para compor o *matching*. Através deste algoritmo de *matching*, os super-vértices formados a cada nível de *coarsening* apresentam, em seu interior, vértices bastante parecidos de acordo com a medida de similaridade empregada. O raciocínio por trás deste algoritmo é que o super-vértice consegue melhor representar os vértices que foram condensados nele.
- *Least Similar Vertex Matching* (LSVM): de forma similar, um vértice é aleatoriamente escolhido e para cada vizinho que não foi selecionado para o *matching* a similaridade é calculada. Entretanto, diferentemente do algoritmo *MSVM*, o vizinho que é selecionado

pelo algoritmo é aquele com o menor resultado de similaridade.

A Figura 5 apresenta um exemplo de aplicação dos dois algoritmos de *matching* propostos em uma mesma rede. Para gerar este exemplo, o índice de similaridade *CN* foi utilizado como medida de similaridade para os algoritmos e apenas arestas incidentes no vértice *A* foram consideradas para a composição do conjunto de *matching*. Utilizando o algoritmo de *matching* *MSVM*, *A* é agrupado com *B*, que é o vizinho com maior número de vizinhos em comum (dois vizinhos) e utilizando o algoritmo *LSVM*, *A* é agrupado com *E*, que não possui vizinhos em comum com *A*. Considerando a execução do algoritmo *MSVM*, que compôs um *matching* que condensou os vértices *A* e *B* em um mesmo super-vértice, temos que três arestas (A, B) , (A, F) , (B, F) , (A, C) , (B, C) deixaram de existir na rede contraída. A aresta (A, B) foi selecionada para o *matching*, removida e os vértices *A* e *B* foram condensados em um super-vértice assim que a rede passou para o próximo nível. As arestas (A, F) e (B, F) foram condensadas em uma super-aresta que liga os vértices $\{A, B\}$ e *F*. Os pesos das arestas que foram condensadas são somados e se tornam o peso da super-aresta, portanto $\omega(\{A, B\}, F) = \omega(A, F) + \omega(B, F)$. O mesmo acontece para as arestas (A, C) e (B, C) , portanto $\omega(\{A, B\}, C) = \omega(A, C) + \omega(B, C)$. O algoritmo *LSVM*, por condensar dois vértices que não tinham vizinhos em comum não fez com que se criassem novam super-arestas na rede, neste caso. O *coarsening* através de algoritmos de *matching* apresenta uma forte tendência a reduzir o coeficiente de agrupamento em redes que tenham este coeficiente muito alto. No exemplo da Figura 5 observa-se que após a contração usando os conjuntos de *matching* gerados pelos dois algoritmos, o número de triângulos na rede foi reduzido.

Figura 5 – Exemplo de aplicação dos algoritmos de *matching* propostos.

Fonte: Elaborada pelo autor.

Considerando que apenas similaridades entre vértices vizinhos são calculados, no pior caso, temos uma complexidade no processo de *matching* de $\mathcal{O}(|V| \cdot |E|)$ para compor um conjunto de *matching* utilizando um índice de similaridade com complexidade linear $\mathcal{O}(|E|)$ tal como *CN* (pior caso que nunca acontece ou a rede seria completamente conectada, desconsiderando assim a premissa para predição de *links*). Considerando que a rede possui $|V|$ vértices, no pior

caso temos que a complexidade para a composição de um conjunto de matching é de $\mathcal{O}(|V| \cdot |E|)$, tanto para *MSVM* quanto para *LSVM*, sendo que a única diferença entre os algoritmos é que o primeiro seleciona o vértice mais similar e o segundo o vértice menos similar. Caso outra medida de similaridade seja utilizada, a complexidade para a composição do conjunto de *matching* seria correspondente ao número de vértices do nível atual da rede multiplicado pelo número de operações que tal medida necessita para extrair todas as similaridades entre um vértice e todos os seus vizinhos. Tal implementação pode ser melhorada com técnicas de programação dinâmica considerando apenas redes não direcionadas, pois a similaridade entre v e u é a mesma de u e v . Como as redes em que preditores de *links* são geralmente esparsas, o processo de *matching* utilizando os algoritmos propostos neste trabalho não é custoso e também alcança resultados melhores do que os algoritmos da literatura considerados de acordo com nossos experimentos. Além disso, como o processo é iterativo, a cada nível de contração, a rede se torna menor o processo de *matching* fica ainda mais rápido.

O processo de *matching* aplicado na rede G_i produz um conjunto M que será utilizado como entrada para o processo de contração da rede. Cada par de vértices em M gera um novo super-vértice para G_{i+1} . Os vértices da rede G_i que não são agrupados em pares com outros vértices são herdados pela rede G_{i+1} . Na Linha 7, a rede G_i e o conjunto de *matching* M são utilizados para gerar uma rede mais reduzida G_{i+1} , denotada por G' .

O processo de predição de *links* é mais rápido na rede reduzida do que na rede original, devido ao número reduzido de vértices e, consequentemente, de arestas não observadas entre eles. Na linha 10, o preditor é aplicado na versão mais reduzida da rede, gerando um *ranking* reduzido de arestas não observadas entre super-vértices. Este processo é mais rápido do que aplicar o mesmo preditor na rede original. Entretanto, para que a otimização multinível se torna viável e aplicável em ambiente reais, três condições devem ser satisfeitas: os processos de *coarsening* e *uncoarsening* devem ser rápidos e a perda de qualidade das predições não deve ser grande o suficiente para inviabilizar o processo, isto é, a projeção das super-arestas entre super-vértices para *links* entre vértices da rede original deve resultar em predições de qualidade. Este último processo é comentado à frente.

É importante observar que, graças à separação dos processos de *coarsening*, predição de *links* e *uncoarsening*, independentemente do método de predição de *links* empregado, a abordagem proposta consegue reduzir o número de operações necessárias. Além da redução do número de operações e, consequentemente o tempo de execução necessário, a abordagem proposta também oferece uma redução substancial da quantidade de memória necessária para o processo de predição de *links*, uma vez que todo o processo de predição é feito em uma versão reduzida da rede, não importando a forma de representação de rede escolhida (matriz de adjacência, lista de adjacência, etc) e o *ranking* de predições é extraído desta rede.

O *ranking* reduzido R' é então usado para gerar o *ranking* final R por meio de um algoritmo de *uncoarsening*. O algoritmo de *uncoarsening* recebe um ranking de super-arestas

não observadas preditas e projeta os *scores*, de forma a criar arestas não observadas preditas. Considerando sE como uma super-aresta predita entre os super-vértices sV e sU , o algoritmo de *uncoarsening* projeta o *score* de sE para todos os pares de vértices entre os super-vértices sV e sU . Observa-se que outras possibilidades existem, por exemplo, poderia haver localmente um recálculo dos *scores* das arestas. Tal estratégia, entretanto, encareceria o processo. A Figura 6 ilustra o processo de projeção de *score* da super-aresta não observada sE .

Figura 6 – Projeção de predições de *links* para redes homogêneas.

Na Figura 6, a super-aresta predita $sE \in R'$ é usada para projetar todas as arestas entre vértices que estão condensados nos super-vértices sV e sU . O algoritmo de *uncoarsening* faz a projeção do peso de sE para todas possíveis as arestas entre os vértices que foram condensados em super-vértices durante o processo de *coarsening*. Neste projeto, dois algoritmos de *uncoarsening* foram propostos e desenvolvidos:

- *Super-Edge Replication* (SER): O *score* obtido pelo preditor para sE é replicado para todas as possíveis arestas dentro dos super-vértices. Usando esta projeção no exemplo da Figura 6, todas as arestas não observadas $\{(v_1, u_1), (v_1, u_2), (v_2, u_1), (v_2, u_2), (v_3, u_1), (v_3, u_2)\}$ receberiam do preditor o mesmo valor de *score* da super-aresta sE .
- *Weighted Super-Edge Replication* (WSER): O *score* obtido pelo preditor na rede reduzida é replicado ponderadamente para todas as possíveis arestas não observadas dentro dos super-vértices. O peso é dividido pelo número de arestas não observadas formadas pelos vértices dentro dos super-vértices. O objetivo é que super-vértices massivos, que tem uma probabilidade maior de possuírem arestas incidentes com um peso muito grande, não gerem arestas entre os vértices internos com o mesmo peso massivo. Quanto maior o número de vértices dentro do super-vértice, menor será o peso final para cada aresta não observada predita. Considerando esta projeção, no exemplo da Figura 6, todas as arestas não observadas $\{(v_1, u_1), (v_1, u_2), (v_2, u_1), (v_2, u_2), (v_3, u_1), (v_3, u_2)\}$ receberiam o *score* da super-aresta sE dividido por 6.

Em alguns casos, podem existir arestas não observadas entre os vértices dentro dos super-vértices. Neste casos, ambos os algoritmos de *uncoarsening* consideram que tais arestas

existem, e é atribuído um valor de *score* referente à similaridade entre os dois vértices que são ligados pela aresta não observada. Nesta fase do *uncoarsening* poderia ser executado um processo de refinamento destas arestas que são bastante promissoras, visto que unem vértices bastantes similares. Para este projeto não foi proposto nenhum algoritmo de refinamento, entretanto este assunto é tratado na Seção 5.2 de trabalhos futuros. Após a aplicação do algoritmo de *uncoarsening* no *ranking* reduzido que foi obtido pela aplicação do algoritmo objetivo na versão contraída da rede, tem-se um *ranking* final completo com todas as arestas não observadas ordenadas de acordo com o *score*. Neste trabalho, este *ranking* final é utilizado para extrair os valores das medidas de desempenho.

AVALIAÇÃO EXPERIMENTAL

Neste capítulo são apresentados o método empregado para a realização dos experimentos e também os resultados obtidos pela abordagem proposta. São apresentadas as bases de dados que foram utilizadas, resultados de tempos de execução para o processo de *coarsening*, as mudanças estruturais nas redes intermediárias do processo de *coarsening*, os valores de medidas de desempenho *AUC* e *precis@L* obtidos pelas predições nos vários níveis de contração adotados e os valores de tempos de execução obtidos pelas predições.

4.1 Bases de dados utilizadas

Esta Seção apresenta as bases de dados utilizadas nos experimentos. Para os experimentos deste projeto foram utilizadas sete bases de dados, ou sete redes, em sua maioria redes sociais. Cinco das redes sociais são de co-autoria, uma provém de uma plataforma da rede social *online Facebook* e uma rede biológica de interação proteína-proteína. As bases de dados utilizadas são as seguintes:

- *netscience*¹: Trata-se de uma base que contém uma rede não direcionada e ponderada de co-autoria entre autores que trabalham em teoria de redes. A base foi compilada por M. Newman em maio de 2006, sendo gerada através de bibliografias de artigos de revisão sobre redes (NEWMAN, 2006).
- *yeast*²: Base contendo uma rede biológica de interação proteína-proteína (BU *et al.*, 2003).
- *facebook*³: Esta base contém uma rede não direcionada formada por listas de amigos de usuários anonimizados obtidas da rede social *online Facebook* e foi coletada através do

¹ Disponível em <http://www-personal.umich.edu/mejn/netdata/netscience.zip>

² Disponível em <http://vlado.fmf.uni-lj.si/pub/networks/data/bio/yeast/yeast.htm>

³ Disponível em <https://snap.stanford.edu/data/egonets-Facebook.html>

aplicativo da rede social. A base original inclui atributos do perfil dos usuários e outras informações, porém os experimentos deste projeto utilizam apenas as informações básicas de conexão.

- *geom*⁴: Esta base contém uma rede ponderada e não direcionada formada por autores da área de geometria computacional que colaboraram na produção de trabalhos científicos na área. Dois autores estão conectados se produziram algum trabalho (artigo, livro, etc) em colaboração. O peso das arestas da rede corresponde ao número de trabalhos produzidos em colaboração.
- *hep-th*⁵: Rede ponderada com co-autorias entre pesquisadores publicando pré-impressões sobre *High-energy Theory* em veículos especializados entre 1 de janeiro de 1995 e 31 de dezembro de 1999 (NEWMAN, 2001).
- *astro-ph*⁶: Rede ponderada com co-autorias entre pesquisadores publicando pré-impressões sobre astrofísica em veículos especializados entre 1 de janeiro de 1995 e 31 de dezembro de 1999 (NEWMAN, 2001).
- *condmat*⁷: Rede ponderada com co-autorias entre pesquisadores publicando pré-impressões sobre matéria condensada em veículos especializados entre 1 de janeiro de 1995 e 31 de março de 2005 (NEWMAN, 2001).

As bases de dados utilizadas nos experimentos, assim como algumas características importantes extraídas das mesmas, foram sumarizadas na Tabela 1:

Bases	$ V $	$ E $	c	C	r	d
<i>netscience</i>	1461	2742	3,753	0,693	0,462	0,0025
<i>yeast</i>	2361	6646	5,629	0,102	-0,099	0,0023
<i>facebook</i>	4039	88234	43,69	0,519	0,063	0,0108
<i>geom</i>	6158	11898	3,864	0,240	0,242	0,0006
<i>hep-th</i>	8361	15751	3,768	0,329	0,294	0,0004
<i>astro-ph</i>	16046	121251	15,112	0,426	0,235	0,0009
<i>condmat</i>	23133	93497	8,083	0,264	0,136	0,0003

Tabela 1 – Sumário das redes utilizadas nos experimentos.

Na Tabela 1 temos algumas características que foram extraídas das redes utilizadas nos experimentos para um melhor entendimento do comportamento da abordagem proposta. Temos sequencialmente nas colunas da Tabela 1: $|V|$ como número de vértices da rede, $|E|$ como o número de arestas, c sendo o grau médio dos vértices da rede, C como o coeficiente de

⁴ Disponível em <http://vlado.fmf.uni-lj.si/pub/networks/data/collab/geom.htm>

⁵ Disponível em <http://www-personal.umich.edu/mejn/netdata/hep-th.zip>

⁶ Disponível em <http://www-personal.umich.edu/mejn/netdata/astro-ph.zip>

⁷ Disponível em <http://www-personal.umich.edu/mejn/netdata/cond-mat-2005.zip>

agrupamento da rede, r como o coeficiente de assortatividade da rede e d como a densidade da rede. Para mais informações sobre os coeficientes utilizados, ver Seção 2.1.

4.2 Método utilizado

Nesta Seção estão descritos todos os passos executados para a obtenção dos resultados de predição de *links* utilizando a abordagem de otimização multinível proposta e desenvolvida neste trabalho.

Para testar a abordagem de otimização multinível, um conjunto de experimentos foi executado. Cada experimento consiste da combinação de um algoritmo de *matching* e um algoritmo de *uncoarsening*. Considerando os três algoritmos de *matching* da literatura que foram apresentados na Seção 2.4, *RM*, *HEM* e *LEM* e os dois algoritmos propostos no Capítulo 3, *MSVM* e *LSVM*, cinco algoritmos de *matching* no total foram utilizados nos experimentos. Além disso, apenas os dois algoritmos de *uncoarsening* propostos no Capítulo 3, *SER* e *WSER*, foram utilizados pois na literatura não existem algoritmos de *uncoarsening* para a tarefa de predição de *links*. Visto que foram utilizados sete bases de dados, cinco algoritmos de *matching* e dois algoritmos de *uncoarsening*, foram executados ao todo setenta experimentos. Para obter resultados em diferentes níveis de *coarsening* da rede, os experimentos foram executados em cinco níveis de *coarsening*, portanto considerando também a execução feita com a rede no nível 0 de *coarsening*, cada experimento é composto de seis execuções. Cada execução então é dada por uma combinação de algoritmo de *matching* e *uncoarsening* e um nível de *coarsening* da rede.

Para cada experimento, a estratégia *k-fold cross-validation* (LU; ZHOU, 2010) foi utilizada. Nessa estratégia, o conjunto E de arestas da rede foi aleatoriamente particionado em cinco subconjuntos, pois utilizamos $k = 5$, gerando cinco iterações para cada execução, que, novamente, é composta por um algoritmo de *matching*, um algoritmo de *uncoarsening* e um nível de *coarsening* da rede. Para cada execução, um dos subconjuntos é utilizado como conjunto de teste E^{ts} e o conjunto resultante da união dos outros quatro conjuntos é utilizado como conjunto de treinamento E^{tr} . Após todas as cinco iterações, a média dos valores obtidos de *precis@L*, *AUC* e tempo de execução são calculados para aquela execução. Os valores de parâmetros $L = 1000$ e $n = 10000$ foram escolhidos respectivamente para o cálculo das medidas *precis@L* e *AUC* pois são valores comumente empregados na literatura por não serem pequenos a ponto de apresentar apenas as principais arestas preditas e nem tão grandes, a ponto de levarem arestas desimportantes em consideração. O mesmo conjunto de treino e teste é utilizado em todos os níveis de *coarsening*; ou seja, a rede é contraída um nível, o *ranking* de predições é obtido, as medidas de desempenho são computadas, a rede é contraída novamente, outro *ranking* é obtido e as medidas são computadas novamente até que se alcance o quinto nível de *coarsening*. Só então a rede volta ao nível zero de *coarsening* e outros conjuntos de treino e teste são utilizados. Dessa

forma, é possível extrair de forma mais objetiva as medidas de desempenho pois os algoritmos de predição de *links* e *uncoarsening* têm como entrada os mesmos conjuntos de treinamento e teste em todos os níveis.

Para os gerar os resultados apresentados nesta dissertação, todas as redes das bases de dados utilizadas foram contraídas do nível zero de *coarsening* até o nível cinco. O índice de similaridade de Katz (KATZ, 1953) foi utilizado devido à sua simplicidade, alta complexidade computacional, que o torna impraticável para redes de alta complexidade estrutural, e bom desempenho de predição. Nos algoritmos de *uncoarsening* propostos, *SER* e *WSER*, o índice de similaridade *CN* foi utilizado na criação do conjunto de *matching*. Os índices de similaridade propostos neste trabalho que foram utilizados na composição do *matching* foram apresentados no Capítulo 3.

4.3 Resultados e análises

Esta Seção apresenta os resultados obtidos para cada processo que compõe a abordagem desenvolvida neste trabalho. São apresentados os resultados de tempos de execução de *coarsening* da rede mediante os diferentes algoritmos de *matching*, além dos resultados estruturais da contração em cada rede intermediária e, finalmente, os resultados de medidas de desempenho e tempos de execução dos algoritmos de predição de *links*.

4.3.1 Análise do processo de *coarsening*

O processo de *coarsening*, composto pelas fases de *matching* e contração, é um processo rápido mesmo quando repetido várias vezes, de forma a obter uma rede bastante contraída, e mesmo considerando redes de larga-escala. A Tabela 2 apresenta, para cada base, para cada nível de *coarsening* e para cada algoritmo de *matching* os tempos de execução em segundos necessários para contrair a rede. Cada linha da tabela representa o tempo necessário para o *coarsening* de um nível específico e as colunas representam respectivamente a base, o nível em que a rede estava e o nível para o qual ela vai, e o algoritmo de *matching* utilizado. A última linha de cada base representa o tempo total gasto pelo *coarsening* da base do nível 0 até o nível 5. Os dados foram obtidos de uma média de cem execuções. Mesmo para as maiores redes consideradas, em número de vértices, e levando em conta o algoritmo mais demorado, o tempo de contração da rede do nível 0 ao nível 5 é de pouco mais de um segundo. Para as redes menores, foram obtidos tempos de centésimos e décimos de segundo.

Em todas as bases de dados consideradas, os algoritmos propostos *RM*, *HEM* e *LEM* propostos na literatura foram os mais rápidos, o que era esperado devido às suas complexidades lineares em termos do número de arestas da rede. Entretanto, mesmo com o custo computacional extra para o cálculo de similaridades na composição do *matching*, os algoritmos propostos e desenvolvidos neste trabalho não apresentaram um custo computacional tão elevado, princi-

palmente devido à esparsidade das redes consideradas. Conforme discutido no Capítulo 3, os algoritmos sorteiam vértices que ainda não foram selecionados para o *matching*, calculam as similaridades entre o vértice sorteado e todos os seus vizinhos, de acordo com algum índice de similaridade, e selecionam o mais similar para o algoritmo *MSVM* e o menos similar para *LSVM*. Em uma rede com alta densidade as diferenças se tornam mais evidentes, pois o custo de calcular a similaridade para muitos vizinhos começa a fazer muita diferença no custo total do *matching*. Esperava-se que na rede *facebook*, que é a rede testada com maior densidade, estas diferenças de tempo se tornassem mais evidentes, entretanto isso não aconteceu. O algoritmo de *matching* *HEM*, que obteve o menor tempo para contrair a rede *facebook* do nível 0 até o nível 1, é cerca de 66% mais rápido do que o algoritmo mais lento *LSVM*.

Para examinar a diferença de custo entre os algoritmos em uma rede de densidade maior, foram executados os mesmos testes de *coarsening* em duas redes totalmente conectadas com $|V| = 1000$ e $|V| = 3000$. Na primeira rede, os algoritmos propostos na literatura levaram, em média, apenas 12,2% do tempo gasto pelos algoritmos propostos neste trabalho para contrair a rede do nível 0 até o nível 1. Conforme a rede é contraída em mais níveis, a diferença deixa de ser tão expressiva. Para a segunda rede, com $|V| = 3000$, a diferença é ainda mais significativa para os primeiros níveis de *coarsening*, e os algoritmos da literatura levam apenas cerca de 3,7% em média do tempo necessários pelos algoritmos propostos. Porém, como dito anteriormente, uma rede completamente conectada não necessita de algoritmos de predição de *links*. Para analisar as diferenças entre os tempos dos algoritmos de *matching* em uma rede bastante densa quando comparada às redes reais, mas que ainda possa fazer uso de algoritmos de predição de *links*, foi gerada uma rede randômica seguindo o modelo de *Erdos-Renyi* (ERDŐS; RÉNYI, 1960) com $|V| = 5000$ e probabilidade de conexão $p = 0.25$. Os algoritmos da literatura fizeram o *coarsening* do nível 0 até o nível 1, em média, cerca de 8,4% do tempo gasto pelos algoritmos propostos. Tais diferenças entre os algoritmos indicam que a esparsidade da rede é um fator muito importante para garantir a viabilidade dos algoritmos de *matching* propostos e que, quanto mais densa a rede mais custosos serão os algoritmos. Embora sejam dependentes da esparsidade da rede para serem viáveis, os algoritmos propostos apresentam melhor desempenho nas medidas de qualidade de predição, conforme indicam os resultados apresentados na próxima subseção.

Além dos resultados de tempo de execução do *coarsening*, é importante também observar as mudanças estruturais na rede à medida que vão passando por níveis cada vez maiores de *coarsening*. Portanto, extraímos os valores dos coeficientes para cada base de dados e para cada algoritmo de *matching*, apresentados na subseção 2.1.1, para uma análise dos efeitos do *coarsening* nas redes. A Tabela 3 apresenta os tipos de alterações nos coeficientes estruturais extraídos conforme o nível de *coarsening* das redes utilizadas aumenta. As informações foram extraídas para cada nível, portanto se para um coeficiente estrutural e um algoritmo de *matching* no nível 3 o símbolo “-” estiver preenchido na tabela, significa que o valor daquele coeficiente naquela rede reduziu no nível 3 em relação ao nível 2. O símbolo “+” representa um acréscimo.

Base	Nível	<i>RM</i>	<i>HEM</i>	<i>LEM</i>	<i>MSVM</i>	<i>LSVM</i>
netscience	0 - 1	$3,3 \times 10^{-3}$	$3,9 \times 10^{-3}$	$4,8 \times 10^{-3}$	$9,6 \times 10^{-3}$	$5,6 \times 10^{-3}$
	1 - 2	$1,3 \times 10^{-3}$	$1,9 \times 10^{-3}$	$2,2 \times 10^{-3}$	$2,3 \times 10^{-3}$	$2,3 \times 10^{-3}$
	2 - 3	$7,1 \times 10^{-4}$	$1,0 \times 10^{-3}$	$1,1 \times 10^{-3}$	$1,3 \times 10^{-3}$	$1,4 \times 10^{-3}$
	3 - 4	$4,7 \times 10^{-4}$	$7,9 \times 10^{-4}$	$8,0 \times 10^{-4}$	$1,0 \times 10^{-3}$	$1,0 \times 10^{-3}$
	4 - 5	$4,0 \times 10^{-4}$	$6,1 \times 10^{-4}$	$7,2 \times 10^{-4}$	$1,0 \times 10^{-3}$	$8,0 \times 10^{-4}$
	Total	$6,3 \times 10^{-3}$	8×10^{-3}	$9,7 \times 10^{-3}$	$1,5 \times 10^{-2}$	$1,1 \times 10^{-2}$
yeast	0 - 1	$1,5 \times 10^{-2}$	$1,5 \times 10^{-2}$	$2,3 \times 10^{-2}$	$1,9 \times 10^{-2}$	$2,6 \times 10^{-2}$
	1 - 2	$1,3 \times 10^{-2}$	$1,3 \times 10^{-2}$	$1,6 \times 10^{-2}$	$1,6 \times 10^{-2}$	$2,3 \times 10^{-2}$
	2 - 3	$1,2 \times 10^{-2}$	$1,1 \times 10^{-2}$	$1,3 \times 10^{-2}$	$1,3 \times 10^{-2}$	$2,3 \times 10^{-2}$
	3 - 4	$9,0 \times 10^{-3}$	$8,9 \times 10^{-3}$	$1,1 \times 10^{-2}$	$1,0 \times 10^{-2}$	$2,1 \times 10^{-2}$
	4 - 5	$8,9 \times 10^{-3}$	$8,0 \times 10^{-3}$	$9,9 \times 10^{-3}$	$9,9 \times 10^{-3}$	$1,9 \times 10^{-2}$
	Total	$5,9 \times 10^{-2}$	$5,6 \times 10^{-2}$	$7,4 \times 10^{-2}$	$7,0 \times 10^{-2}$	$1,1 \times 10^{-1}$
facebook	0 - 1	$1,4 \times 10^{-1}$	$1,2 \times 10^{-1}$	$1,3 \times 10^{-1}$	$2,5 \times 10^{-1}$	$3,6 \times 10^{-1}$
	1 - 2	$8,4 \times 10^{-2}$	$7,0 \times 10^{-2}$	$7,9 \times 10^{-2}$	$1,2 \times 10^{-1}$	$2,8 \times 10^{-1}$
	2 - 3	$4,0 \times 10^{-2}$	$3,8 \times 10^{-2}$	$3,9 \times 10^{-2}$	$4,4 \times 10^{-2}$	$1,2 \times 10^{-1}$
	3 - 4	$1,6 \times 10^{-2}$	$1,5 \times 10^{-2}$	$1,7 \times 10^{-2}$	$1,8 \times 10^{-2}$	$5,3 \times 10^{-2}$
	4 - 5	$7,7 \times 10^{-3}$	$8,1 \times 10^{-3}$	$8,2 \times 10^{-3}$	$8,5 \times 10^{-3}$	$1,8 \times 10^{-2}$
	Total	$2,9 \times 10^{-1}$	$2,5 \times 10^{-2}$	$2,7 \times 10^{-1}$	$4,4 \times 10^{-1}$	$8,3 \times 10^{-1}$
geom	0 - 1	$2,1 \times 10^{-2}$	$1,8 \times 10^{-2}$	$3,0 \times 10^{-2}$	$3,1 \times 10^{-2}$	$2,6 \times 10^{-2}$
	1 - 2	$1,0 \times 10^{-2}$	$1,1 \times 10^{-2}$	$2,6 \times 10^{-2}$	$2,1 \times 10^{-2}$	$2,2 \times 10^{-2}$
	2 - 3	$1,5 \times 10^{-2}$	$0,8 \times 10^{-3}$	$1,0 \times 10^{-2}$	$1,1 \times 10^{-2}$	$1,3 \times 10^{-2}$
	3 - 4	$5,8 \times 10^{-3}$	$0,6 \times 10^{-3}$	$1,3 \times 10^{-2}$	$1,0 \times 10^{-2}$	$1,1 \times 10^{-2}$
	4 - 5	$9,7 \times 10^{-3}$	$0,5 \times 10^{-3}$	$8,2 \times 10^{-3}$	$7,5 \times 10^{-3}$	$1,0 \times 10^{-2}$
	Total	$6,2 \times 10^{-2}$	$4,9 \times 10^{-2}$	$8,9 \times 10^{-2}$	$8,2 \times 10^{-2}$	$8,5 \times 10^{-2}$
hep-th	0 - 1	$2,9 \times 10^{-2}$	$2,6 \times 10^{-2}$	$2,5 \times 10^{-2}$	$4,1 \times 10^{-2}$	$4,2 \times 10^{-2}$
	1 - 2	$1,3 \times 10^{-2}$	$2,3 \times 10^{-2}$	$2,2 \times 10^{-2}$	$2,1 \times 10^{-2}$	$2,6 \times 10^{-2}$
	2 - 3	$9,6 \times 10^{-3}$	$1,5 \times 10^{-2}$	$1,1 \times 10^{-2}$	$1,9 \times 10^{-2}$	$2,7 \times 10^{-2}$
	3 - 4	$7,4 \times 10^{-3}$	$9,8 \times 10^{-3}$	$9,0 \times 10^{-3}$	$1,6 \times 10^{-2}$	$2,0 \times 10^{-2}$
	4 - 5	$6,3 \times 10^{-3}$	$8,6 \times 10^{-3}$	$7,4 \times 10^{-3}$	$1,2 \times 10^{-2}$	$1,5 \times 10^{-2}$
	Total	$6,6 \times 10^{-2}$	$8,4 \times 10^{-2}$	$7,6 \times 10^{-2}$	$1,1 \times 10^{-1}$	$1,3 \times 10^{-1}$
astro-ph	0 - 1	$2,1 \times 10^{-1}$	$1,8 \times 10^{-1}$	$2,1 \times 10^{-1}$	$2,8 \times 10^{-1}$	$5,1 \times 10^{-1}$
	1 - 2	$1,9 \times 10^{-1}$	$1,4 \times 10^{-1}$	$1,6 \times 10^{-1}$	$1,7 \times 10^{-1}$	$3,6 \times 10^{-1}$
	2 - 3	$1,0 \times 10^{-1}$	$9,9 \times 10^{-2}$	$1,0 \times 10^{-1}$	$8,7 \times 10^{-2}$	$2,8 \times 10^{-1}$
	3 - 4	$7,7 \times 10^{-2}$	$6,0 \times 10^{-2}$	$7,0 \times 10^{-2}$	$6,6 \times 10^{-2}$	$2,8 \times 10^{-1}$
	4 - 5	$4,9 \times 10^{-2}$	$4,5 \times 10^{-2}$	$6,6 \times 10^{-2}$	$4,8 \times 10^{-2}$	$3,4 \times 10^{-1}$
	Total	$6,4 \times 10^{-1}$	$5,3 \times 10^{-1}$	$6,3 \times 10^{-1}$	$6,5 \times 10^{-1}$	1,7
condmat	0 - 1	$1,9 \times 10^{-1}$	$2,6 \times 10^{-1}$	$1,7 \times 10^{-1}$	$2,2 \times 10^{-1}$	$2,9 \times 10^{-1}$
	1 - 2	$1,4 \times 10^{-1}$	$2,0 \times 10^{-1}$	$1,0 \times 10^{-1}$	$1,6 \times 10^{-1}$	$2,4 \times 10^{-1}$
	2 - 3	$1,0 \times 10^{-1}$	$1,3 \times 10^{-1}$	$1,2 \times 10^{-1}$	$1,3 \times 10^{-1}$	$1,8 \times 10^{-1}$
	3 - 4	$8,4 \times 10^{-2}$	$1,1 \times 10^{-1}$	$8,5 \times 10^{-2}$	$9,3 \times 10^{-2}$	$2,3 \times 10^{-1}$
	4 - 5	$7,3 \times 10^{-2}$	$7,4 \times 10^{-2}$	$5,5 \times 10^{-2}$	$7,1 \times 10^{-2}$	$2,1 \times 10^{-1}$
	Total	$6,1 \times 10^{-1}$	$8,0 \times 10^{-1}$	$5,4 \times 10^{-1}$	$6,8 \times 10^{-1}$	1,1

Tabela 2 – Tempos de execução em segundos do processo de *coarsening* das bases de dados.

		C					r					d				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
netscience	RM	-	-	-	-	-	-	-	-	-	-	+	+	-	-	-
	HEM	-	-	-	-	-	-	-	-	-	-	+	+	-	-	-
	LEM	-	-	-	-	+	-	-	-	-	-	+	+	-	-	-
	MSVM	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-
	LSVM	-	-	-	-	-	-	-	-	-	-	+	+	-	-	-
yeast	RM	+	+	+	+	+	-	-	-	-	-	+	+	+	+	+
	HEM	+	+	+	+	+	-	+	-	+	-	+	+	+	+	+
	LEM	+	+	+	+	+	+	-	+	-	-	+	+	+	+	+
	MSVM	-	-	+	+	-	-	-	-	-	-	+	+	+	-	-
	LSVM	+	+	+	+	+	+	-	-	+	+	+	+	+	+	+
facebook	RM	+	+	-	-	-	+	-	-	-	-	+	+	+	+	+
	HEM	+	+	-	-	-	+	-	-	-	-	+	+	+	+	+
	LEM	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+
	MSVM	+	-	-	-	-	+	-	-	-	-	+	+	+	-	-
	LSVM	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+
geom	RM	-	-	+	+	+	-	-	-	-	-	+	+	+	+	-
	HEM	-	-	+	+	+	-	-	-	-	-	+	+	+	+	+
	LEM	-	-	+	+	+	-	-	-	-	-	+	+	+	+	+
	MSVM	-	-	-	-	-	-	-	-	-	-	+	+	+	-	-
	LSVM	-	-	+	+	+	-	-	-	-	-	+	+	+	+	+
hep-th	RM	-	-	-	+	+	-	-	-	-	-	+	+	+	+	+
	HEM	-	-	-	-	+	-	+	-	-	-	+	+	+	+	+
	LEM	-	-	-	+	+	-	-	-	-	-	+	+	+	+	+
	MSVM	-	-	-	-	+	-	-	-	-	-	+	+	+	+	-
	LSVM	-	-	-	+	+	-	-	-	-	-	+	+	+	+	+
astro-ph	RM	-	-	-	+	+	-	-	-	-	-	+	+	+	+	+
	HEM	-	-	-	-	+	+	-	-	-	-	+	+	+	+	+
	LEM	-	-	-	-	+	+	-	-	-	-	+	+	+	+	+
	MSVM	-	-	-	+	+	-	-	-	-	-	+	+	+	+	-
	LSVM	-	-	-	-	+	+	-	-	-	-	+	+	+	+	+
condmat	RM	-	-	-	-	+	-	-	-	-	-	+	+	+	+	+
	HEM	-	-	-	-	+	+	-	-	-	+	+	+	+	+	+
	LEM	-	-	-	-	+	+	-	-	-	-	+	+	+	+	+
	MSVM	-	-	-	+	+	-	-	-	-	-	+	+	+	+	-
	LSVM	-	-	-	-	+	+	-	-	-	-	+	+	+	+	+

Tabela 3 – Alterações nos coeficientes estruturais conforme o nível de *coarsening* da rede aumenta. O símbolo "+" representa que do nível anterior para o atual houve um acréscimo no valor daquele coeficiente do último nível de *coarsening* até o atual. O símbolo "-" indica decréscimos nesses coeficientes.

É possível observar que alguns coeficientes seguem um padrão para algumas redes específicas. Foi registrado um aumento do coeficiente de agrupamento para todas as redes nos últimos níveis de contração. Nos níveis iniciais, apenas a rede *facebook* e *yeast* apresentaram um aumento. As redes *yeast* e *geom* possuem, respectivamente, os dois mais baixos coeficientes de

agrupamento registrados nas redes estudadas. De todos os algoritmos de *matching*, o algoritmo *MSVM* é o que ocasiona menos aumentos no coeficiente de agrupamento e esses aumentos acontecem apenas no último nível. Em média, os algoritmos de *matching* propostos na literatura ocasionam mais aumentos no coeficiente de agrupamento do que os algoritmos propostos neste trabalho.

A assortatividade da rede decresce na maioria dos casos, exceto nas redes de *yeast* e *facebook* que possuem respectivamente os dois menores coeficientes de assortatividade registrados. Na rede *yeast* os aumentos acontecem com os algoritmos *HEM*, *LEM* e *LSVM*. Na rede *facebook* os aumentos acontecem no primeiro nível de contração e depois seguem acontecendo apenas para os algoritmos que unem vértices com conexões fracas (baixo valor de peso da aresta) e baixa similaridade, *LEM* e *LSVM*. A principal hipótese para tal comportamento é que unindo vértices com conexões e similaridades fracas, lembrando que a medida de similaridade adotada foi *CN*, acaba-se por condensar vértices isolados aos vértices de maior grau. Em redes com baixa assortatividade, ou seja, em que vértices de grau alto tendem a se conectar com vértices de grau baixo, a remoção de arestas que unem vértices com baixa força de conexão ou baixo nível de similaridade elimina arestas que estavam conectando um vértice de grau alto com um vértice de grau baixo, aumentando a assortatividade da rede. Nas redes com maior assortatividade, o fenômeno de aumento é observado em raras ocasiões. Da mesma forma, o aumento é observado apenas uma vez no algoritmo de *matching MSVM*, que naturalmente condensa em super-vértices os vértices com maior grau, contribuindo para uma diminuição na assortatividade da rede.

A densidade da rede também aumenta em quase todos os casos, menos nos últimos níveis de contração da rede *netscience* e alguns outros casos do nível 5 de contração utilizando o algoritmo de *matching MSVM*. A rede *netscience*, embora não tenha uma densidade tão baixa quando as outras redes de colaboração científica, possui um número pequeno de arestas e o maior coeficiente de aglomeração de todas as redes abordadas. Quando ocorre um *matching* entre vértices de um triângulo, além da aresta entre os vértices que estão sendo condensados ser removida, duas outras arestas são unidas em apenas uma (exemplo na Figura 7), diminuindo naturalmente a densidade da rede. Como a rede *netscience* possui um número grande de triângulos formados entre os vértices esta situação é frequente, levando a rede a ter sua densidade diminuída em todos os níveis de *coarsening*. Outro fator que aponta tais indícios é que o algoritmo de *matching MSVM*, que utiliza *CN* para criar selecionar vizinhos mais similares para compor o *matching*, causa a redução da densidade da rede desde o primeiro nível de contração.

4.3.2 Análise da AUC

A Tabela 4 apresenta os resultados de *AUC* para os experimentos executados. Cada linha da tabela apresenta os resultados para um nível de *coarsening* da rede para cada combinação do algoritmo de *matching* e *uncoarsening*. As células da tabela destacadas com um fundo cinza claro mostram o melhor resultado obtido naquela rede e naquele nível de contração da rede. O

Figura 7 – Exemplo de rede com um conjunto de vértices sendo conectado por um conjunto de arestas.

Fonte: Elaborada pelo autor.

nível 0 representa os valores de *AUC* obtidos pelo algoritmo de predição de *links* na rede original, ou seja, na rede que não passou por nenhuma contração.

A abordagem de otimização multinível realiza a contração da rede para que o algoritmo objetivo seja executado mais rapidamente à custa de perdas de qualidade da solução. Nas redes das bases *netscience*, *yeast*, *facebook* e *geom*, as perdas de *AUC* acontecem imediatamente após a contração da rede no primeiro nível e aumentam conforme a rede é contraída. Destas, a base *netscience* apresenta a menor perda média para as combinações de *matching* e *uncoarsening* nos cinco níveis, sendo que, do nível 0 até o nível 5, a *AUC* cai apenas 1,6%, seguida da rede *geom* com uma perda média de 5,9%, *yeast* com 26,8% e *facebook* com 34,6%, sendo esta a maior perda de *AUC* para todas as redes consideradas neste trabalho.

Nas redes *hep-th* e *condmat* ocorre um aumento nos valores de *AUC* nos primeiros dois níveis de contração da rede, o que é bem incomum na utilização de otimização multinível. Entretanto, a partir do nível 2 de *coarsening* é registrado um declínio no valor de *AUC*, e no nível 3 os valores de *AUC* já passam a ser inferiores aos registrados no nível 0. Para as bases *hep-th* e *condmat*, as perdas médias de *AUC*, conforme a contração da rede vai do nível 0 ao nível 5, são de 13,7% e 14,7%, respectivamente. Finalmente, a base de dados *astro-ph* apresenta um resultado bastante inesperado, visto que conforme a rede é contraída, os valores de *AUC* crescem ao invés de diminuir. Para esta base, foram executados testes com níveis mais altos de contração e apenas no nível 7 os valores de *AUC* começam a apresentar declínio. Nesta rede foi registrado um aumento médio de *AUC* de 3% do nível 0 ao nível 5. As redes que apresentam os maiores valores de assortatividade registraram as menores perdas de *AUC* mediante a contração da rede. As redes que obtiveram os melhores resultados de *AUC* foram respectivamente *astro-ph* e *netscience*. Considerando apenas esta medida de desempenho, os resultados indicam que a abordagem proposta é bastante viável.

Comparando as combinações de *matching* e *uncoarsening*, as que obtiveram os melhores resultados foram as combinações do algoritmo *MSVM* quando associado ao algoritmo de *uncoarsening* *SER* e *RM* quando associado ao *uncoarsening* *WSER*. Para cada combinação de *matching* foram extraídas as médias das diferenças entre os valores de *AUC* entre nível 5 e

nível 0 para todas as bases de dados, de forma a obter as médias de perdas associadas a cada combinação de *matching* e *uncoarsening*. Considerando as combinações com o algoritmo de *uncoarsening* *SER*, o algoritmo de *matching* *RM* obteve uma perda de 4,7%, *HEM* obteve 7,3%, *LEM* obteve 8,1%, *MSVM* obteve 3,3% e *LSVM* obteve 6,6%. O algoritmo de *matching* que obteve o pior desempenho foi o *LEM*. Considerando as combinações dos algoritmos de *matching* com o *uncoarsening* *WSER* as perdas de *AUC* foram bem mais significativas. O algoritmo de *matching* que obteve o melhor desempenho foi o *RM* com perdas de 14,7%, *HEM* obteve 25,5%, *LEM* obteve 29,1%, *MSVM* obteve 17% e *LSVM* obteve 20,7%. Novamente o algoritmo *LEM* obteve as maiores perdas. Considerando as combinações com *uncoarsening* *SER*, a base de dados com as maiores perdas foi *yeast*, com a base *facebook* em segundo lugar. Em geral, o algoritmo de *uncoarsening* *SER* obteve um decaimento significativamente mais suave de *AUC* conforme a rede era contraída, quando comparado ao algoritmo *WSER*, que apresentou um decaimento mais brusco.

Quando considerado o *uncoarsening* *WSER* as últimas posições se inverteram e os piores resultados foram obtidos na base *facebook* seguida pela base *yeast*. Desconsiderando a base de dados *astro-ph*, em que ocorreram ganhos de *AUC*, na base *netscience* ocorreram as menores perdas de *AUC* para ambos algoritmos de *uncoarsening*. Levando em consideração a medida de desempenho *AUC* e os testes realizados, o algoritmo de *uncoarsening* *WSER* apresentou resultados piores do que o algoritmo *SER*, embora tenha obtido os melhores resultados nas bases *netscience* com o algoritmo de *matching* *MSVM*, na base *geom* com o *matching* *RM* e na base *condmat* com os algoritmos *MSVM* e *LSVM*. Na rede *astro-ph*, embora não tenha obtido os melhores resultados, obteve resultados superiores à maioria dos resultados obtidos pelos algoritmos de *matching* combinados com o *uncoarsening* *SER*. Considerando *AUC* como base de comparação, o algoritmo de *uncoarsening* *WSER* obteve resultados significativos nas redes com alto valor de assortatividade e resultados inferiores nas redes com baixo valor de assortatividade, o que levanta indícios sobre o desempenho do algoritmo *WSER* em redes com tais características. Nas outras bases de dados e considerando outros algoritmos de *matching*, os valores de *AUC* começam a ter um declínio significativo a partir do nível 3 de *coarsening*, decaimento brusco que não ocorre no algoritmo *SER*.

Os resultados de *AUC* apresentam indícios da superioridade dos algoritmos de *matching* na tarefa de predição de *links* com os algoritmos de *uncoarsening* propostos, frente aos algoritmos de *matching* propostos na literatura que foram utilizados nesses experimentos. A próxima subseção apresenta os resultados levando em consideração outra medida de desempenho de preditores de *links*, a *precisi@L*.

4.3.3 Análise de *precisi@L*

A Tabela 5 apresenta os resultados de *precisi@L* para os experimentos executados. Cada linha da tabela apresenta os resultados para um nível de *coarsening* da rede para cada combinação

Base	N	SER					WSER				
		RM	HEM	LEM	MSVM	LSVM	RM	HEM	LEM	MSVM	LSVM
netscience	0	0,927	0,927	0,927	0,927	0,927	0,927	0,927	0,927	0,927	0,927
	1	0,919	0,910	0,919	0,921	0,924	0,915	0,921	0,922	0,925	0,924
	2	0,918	0,910	0,919	0,921	0,923	0,916	0,922	0,920	0,925	0,923
	3	0,916	0,909	0,915	0,921	0,919	0,914	0,916	0,916	0,925	0,922
	4	0,915	0,907	0,917	0,918	0,916	0,914	0,910	0,910	0,924	0,921
	5	0,912	0,904	0,912	0,915	0,914	0,914	0,908	0,900	0,924	0,922
yeast	0	0,819	0,819	0,819	0,819	0,819	0,819	0,819	0,819	0,819	0,819
	1	0,816	0,815	0,814	0,819	0,817	0,816	0,811	0,802	0,810	0,801
	2	0,793	0,789	0,779	0,799	0,778	0,793	0,716	0,717	0,720	0,719
	3	0,743	0,741	0,722	0,773	0,728	0,743	0,621	0,617	0,618	0,620
	4	0,709	0,687	0,688	0,747	0,679	0,709	0,531	0,549	0,503	0,543
	5	0,684	0,655	0,641	0,743	0,637	0,684	0,503	0,501	0,451	0,496
facebook	0	0,977	0,977	0,977	0,977	0,977	0,977	0,977	0,977	0,977	0,977
	1	0,891	0,894	0,895	0,897	0,888	0,844	0,848	0,846	0,855	0,837
	2	0,894	0,899	0,879	0,927	0,808	0,731	0,730	0,680	0,742	0,591
	3	0,936	0,945	0,895	0,956	0,795	0,619	0,655	0,517	0,655	0,383
	4	0,944	0,950	0,916	0,953	0,857	0,488	0,549	0,344	0,548	0,273
	5	0,924	0,926	0,900	0,927	0,885	0,376	0,458	0,282	0,451	0,252
geom	0	0,878	0,878	0,878	0,878	0,878	0,878	0,878	0,878	0,878	0,878
	1	0,877	0,868	0,874	0,878	0,877	0,875	0,869	0,871	0,871	0,874
	2	0,873	0,865	0,868	0,875	0,875	0,877	0,854	0,854	0,870	0,869
	3	0,868	0,860	0,862	0,870	0,869	0,871	0,824	0,824	0,866	0,865
	4	0,862	0,847	0,849	0,865	0,864	0,866	0,773	0,777	0,860	0,859
	5	0,855	0,831	0,831	0,855	0,856	0,862	0,741	0,730	0,855	0,851
hep-th	0	0,880	0,880	0,880	0,880	0,880	0,880	0,880	0,880	0,880	0,880
	1	0,885	0,881	0,881	0,880	0,885	0,883	0,881	0,879	0,881	0,882
	2	0,886	0,883	0,881	0,884	0,884	0,872	0,872	0,871	0,873	0,865
	3	0,879	0,880	0,878	0,879	0,870	0,840	0,845	0,833	0,842	0,803
	4	0,864	0,865	0,866	0,864	0,838	0,769	0,784	0,768	0,762	0,713
	5	0,837	0,838	0,840	0,842	0,789	0,699	0,709	0,695	0,699	0,646
astro-ph	0	0,982	0,982	0,982	0,982	0,982	0,982	0,982	0,982	0,982	0,982
	1	0,984	0,983	0,983	0,984	0,985	0,984	0,983	0,984	0,983	0,984
	2	0,985	0,985	0,984	0,985	0,987	0,985	0,985	0,985	0,985	0,986
	3	0,986	0,986	0,986	0,986	0,988	0,986	0,986	0,986	0,986	0,987
	4	0,987	0,987	0,987	0,987	0,988	0,987	0,987	0,987	0,987	0,987
	5	0,984	0,984	0,984	0,986	0,988	0,987	0,987	0,986	0,987	0,985
condmat	0	0,958	0,958	0,958	0,958	0,958	0,958	0,958	0,958	0,958	0,958
	1	0,959	0,957	0,958	0,959	0,959	0,959	0,956	0,958	0,960	0,961
	2	0,959	0,956	0,956	0,961	0,961	0,960	0,935	0,931	0,961	0,962
	3	0,954	0,948	0,942	0,960	0,960	0,956	0,850	0,822	0,961	0,956
	4	0,946	0,916	0,895	0,953	0,953	0,949	0,649	0,612	0,954	0,948
	5	0,935	0,824	0,799	0,944	0,944	0,937	0,469	0,443	0,946	0,937

Tabela 4 – Resultados de *AUC* para as bases de dados. Os dados na tabela foram calculados para $n = 10000$.

do algoritmo de *matching* e *uncoarsening*. O nível 0 representa os valores de *precisi@L* obtidos pelo algoritmo de predição de *links* na rede original, ou seja, na rede que não passou por nenhum *coarsening*. As células da tabela destacadas com um fundo cinza claro mostram o melhor

resultado obtido naquela base e naquele nível de contração da rede.

De maneira similar ao que ocorre nos resultados de *AUC*, em algumas redes é presente o fenômeno não usual dos resultados obtidos em redes que passaram por *coarsening* serem melhores do que os obtidos sem nenhuma contração na rede. Novamente este fenômeno acontece nas redes *netscience* e *astro-ph*, porém após o primeiro nível na rede *netscience* os valores já voltam a diminuir e na rede *astro-ph* a redução não chega a acontecer para os algoritmos de *matching MSVM* e *RM* nem mesmo no nível 5. Entretanto, diferentemente da medida *AUC*, não foram observados tais fenômenos nas redes *hep-th* e *condmat*. A hipótese é que este comportamento anômalo acontece pois o super-vértice obtido tende a fazer uma generalização em redes sociais com altos valores de coeficiente de agrupamento e assortatividade, por exemplo, em uma rede sociais de co-autoria, um super-vértice pode representar um grupo de cientistas em termos de colaboração. Quando os *scores* referentes à super-arestas entre super-vértices são projetados para os vértices que foram condensados em super-vértices, o algoritmo de *uncoarsening* projeta arestas entre grupos bastantes similares de pesquisadores, criando predições de alta qualidade. O principal indício que aponta para tal hipótese é que redes que apresentam alto coeficiente de aglomeração e assortatividade apresentam também os menores declínios de qualidade de predição à medida que a rede vai sendo mais contraída no processo de *coarsening*.

Na rede *netscience* o desempenho de acordo com a medida *precisi@L* começa a sofrer mudanças significativas, a ponto de tornar o processo inviável, a partir do nível 3 de *coarsening*, exceto para as combinações de algoritmos *WSER-RM*, *WSER-MSVM* e *WSER-LSVM*, que tiveram perdas do nível 0 ao nível 5 de 26,8%, 21,5% e 14,7% respectivamente. Para as outras combinações, se mantém aceitável até o nível 3 de *coarsening* exceto para as combinações *WSER-HEM* com perda de 81% e *WSER-LEM* com perda de 76,1%. A rede *yeast* apresentou resultados insatisfatórios de *precisi@L* de modo geral. As combinações *WSER-LSVM*, *SER-MSVM* e *SER-HEM*, que são as que apresentaram os melhores resultados para cada nível, obtiveram ainda resultados de baixa qualidade. No nível 3 de contração, para o melhor resultado, a *precisi@L* tinha reduzido em cerca de 77,8% o valor em relação à predição obtida na rede sem contrações. No nível 5 a redução chega a 87,1% no melhor caso. Diferentemente dos resultados de *AUC*, em que os piores resultados tinham sido obtidos para a base *facebook*, os piores resultados de *precisi@L* foram obtidos na base *yeast*.

Na base *facebook* tem-se uma redução média de 65,2% já no primeiro nível de contração da rede. No nível 2 de *coarsening* todas as combinações obtêm uma redução média altíssima de 94,2% dos valores mas algumas combinações conseguem se recuperar logo no nível seguinte, exceto as combinações envolvendo o algoritmo de *uncoarsening WSER* e as combinações envolvendo o algoritmo de *matching HEM*. No nível 3 as combinações envolvendo o algoritmo *SER* conseguem valores mais altos do que no nível 2 de *coarsening*. A combinação *SER-MSVM* obteve novamente os melhores resultados para a base *facebook*, com exceção dos níveis 2 e 5 de *coarsening* da rede. As combinações envolvendo o algoritmo *WSER* mostraram-se altamente

ineficientes para esta base, visto que em algumas combinações no nível 5 os valores de $precisi@L$ chegam a zero. Para as três melhores combinações, obteve-se uma média de perda de 65,7%.

Os resultados obtidos na base *geom* também mostram perdas significativas, sendo também inviáveis a partir do nível 3 de *coarsening* na rede, pois a perda média na rede era de 73% neste nível. As combinações envolvendo o algoritmo de *matching MSVM* obtiveram os melhores resultados em todos os níveis. A base *hep-th* apresenta resultados de $precisi@L$ que se tornam inviáveis a partir do nível 4 de *coarsening* da rede, sendo que a combinação *SER-HEM* obteve os melhores resultados e os resultados obtidos pelas combinações do algoritmo de *uncoarsening WSER* já não são satisfatórios a partir do nível 1 de *coarsening* da rede. Para a base *astro-ph* os melhores resultados de todos os níveis apresentam valores de $precisi@L$ superiores aos resultados obtidos na rede sem contração. Novamente, foram executados testes além do nível 5 de *coarsening*, e no nível 7 os valores deixam de ser superiores. Os melhores resultados em cada nível se deram por combinações diferentes dos algoritmos de *matching* e *uncoarsening*. No nível 5 de *coarsening* o ganho para o melhor resultado foi de 22,3%. Entretanto, os algoritmos *HEM* e *LEM* apresentaram resultados insatisfatórios a partir do nível 3 de contração em combinações com o algoritmo *SER*, com perdas médias de 65% e do a partir do nível 2 nas combinações com o algoritmo *WSER*, com perdas médias de 64%. Finalmente na base *condmat* a melhor combinação em todos os níveis foi *WSER-MSVM*, sendo que no nível 5 a perda era de 70% para esta combinação enquanto que a média obtida por todas as outras chegou a 92,2%. As combinações que envolviam os algoritmos *HEM* e *LEM* obtiveram resultados ruins já a partir do nível 2 de contração chegando a atingir valores zerados em alguns casos com o algoritmo *WSER*.

Os resultados de $precisi@L$ mostram que o algoritmo de *matching MSVM*, para ambas combinações com os algoritmos de *uncoarsening* apresentou melhor desempenho para as bases avaliadas. Além disso, também existem indícios da diferença entre as duas medidas de desempenho empregadas, uma vez que na *AUC* as perdas de qualidade nas predições eram em menor escala dos que as observadas na medida $precisi@L$. Além disso, é possível observar que, para os algoritmos empregados e desenvolvidos de *matching* e *uncoarsening*, o nível 5 de *coarsening* na rede leva a resultados de baixa qualidade. Portanto, na aplicação de tal abordagem, sugere-se que o *coarsening* se mantenha até nível 3.

4.3.4 Análise de tempo de execução

A Tabela 6 apresenta os resultados de tempo de execução para os experimentos executados. Cada linha da tabela apresenta os resultados para um nível de *coarsening* da rede para cada combinação do algoritmo de *matching* e *uncoarsening*. O nível 0 representa os valores de tempo de execução obtidos pelo algoritmo de predição de *links* na rede original, ou seja, na rede que não passou por nenhuma contração e as outras linhas mostram os valores de tempo de execução proporcionais ao tempo necessário para a predição de *links* na rede sem nenhum *coarsening*. As células da tabela destacadas com um fundo cinza claro mostram o menor tempo obtido naquela

Base	N	SER					WSER				
		RM	HEM	LEM	MSVM	LSVM	RM	HEM	LEM	MSVM	LSVM
<i>netscience</i>	0	0,264	0,264	0,264	0,264	0,264	0,264	0,264	0,264	0,264	0,264
	1	0,265	0,254	0,240	0,343	0,270	0,295	0,078	0,107	0,349	0,299
	2	0,202	0,207	0,192	0,191	0,179	0,266	0,050	0,063	0,279	0,243
	3	0,117	0,121	0,112	0,101	0,088	0,230	0,032	0,035	0,229	0,223
	4	0,059	0,061	0,056	0,051	0,042	0,192	0,009	0,010	0,202	0,212
	5	0,027	0,021	0,027	0,014	0,017	0,193	0,002	0,008	0,207	0,225
<i>yeast</i>	0	0,140	0,140	0,140	0,140	0,140	0,140	0,140	0,140	0,140	0,140
	1	0,043	0,042	0,046	0,041	0,045	0,039	0,050	0,045	0,039	0,051
	2	0,015	0,022	0,018	0,031	0,009	0,010	0,013	0,019	0,008	0,016
	3	0,008	0,027	0,008	0,043	0,003	0,002	0,005	0,006	0,004	0,006
	4	0,007	0,030	0,003	0,033	0,002	0,002	0,001	0,002	0,001	0,004
	5	0,005	0,018	0,004	0,015	0,002	0,001	0,000	0,000	0,002	0,002
<i>facebook</i>	0	0,917	0,917	0,917	0,917	0,917	0,917	0,917	0,917	0,917	0,917
	1	0,318	0,327	0,317	0,364	0,286	0,313	0,320	0,321	0,344	0,281
	2	0,052	0,054	0,061	0,044	0,062	0,044	0,048	0,063	0,043	0,062
	3	0,402	0,445	0,283	0,543	0,404	0,010	0,009	0,009	0,009	0,013
	4	0,336	0,419	0,123	0,517	0,130	0,006	0,002	0,003	0,002	0,003
	5	0,209	0,367	0,047	0,365	0,062	0,004	0,000	0,000	0,000	0,000
<i>geom</i>	0	0,184	0,184	0,184	0,184	0,184	0,184	0,184	0,184	0,184	0,184
	1	0,068	0,086	0,072	0,108	0,070	0,070	0,066	0,066	0,098	0,066
	2	0,048	0,063	0,041	0,084	0,039	0,053	0,027	0,026	0,079	0,036
	3	0,024	0,041	0,023	0,054	0,023	0,031	0,014	0,011	0,057	0,024
	4	0,022	0,026	0,018	0,040	0,019	0,026	0,004	0,005	0,045	0,015
	5	0,010	0,012	0,014	0,027	0,013	0,024	0,001	0,002	0,026	0,009
<i>hep-th</i>	0	0,290	0,290	0,290	0,290	0,290	0,290	0,290	0,290	0,290	0,290
	1	0,159	0,176	0,166	0,180	0,142	0,042	0,048	0,048	0,044	0,047
	2	0,157	0,181	0,147	0,179	0,118	0,013	0,011	0,011	0,012	0,013
	3	0,111	0,139	0,100	0,106	0,080	0,004	0,004	0,004	0,004	0,004
	4	0,071	0,083	0,060	0,060	0,044	0,000	0,001	0,001	0,001	0,001
	5	0,023	0,024	0,025	0,020	0,013	0,001	0,000	0,000	0,000	0,000
<i>astro-ph</i>	0	0,215	0,215	0,215	0,215	0,215	0,215	0,215	0,215	0,215	0,215
	1	0,211	0,213	0,223	0,220	0,203	0,209	0,210	0,219	0,209	0,211
	2	0,201	0,206	0,191	0,220	0,201	0,203	0,089	0,063	0,210	0,205
	3	0,228	0,063	0,096	0,216	0,225	0,219	0,000	0,000	0,200	0,212
	4	0,180	0,061	0,064	0,261	0,078	0,254	0,000	0,000	0,267	0,256
	5	0,031	0,001	0,038	0,213	0,015	0,263	0,000	0,000	0,224	0,211
<i>condmat</i>	0	0,410	0,410	0,410	0,410	0,410	0,410	0,410	0,410	0,410	0,410
	1	0,209	0,144	0,159	0,256	0,205	0,225	0,024	0,027	0,251	0,217
	2	0,194	0,173	0,089	0,249	0,192	0,203	0,004	0,006	0,246	0,206
	3	0,110	0,109	0,047	0,139	0,110	0,139	0,002	0,000	0,170	0,132
	4	0,052	0,058	0,027	0,064	0,065	0,096	0,000	0,000	0,123	0,106
	5	0,020	0,022	0,015	0,028	0,036	0,079	0,000	0,000	0,120	0,086

Tabela 5 – Resultados de $precisi@L$ para as bases de dados. Os dados na tabela foram calculados para $L = 1000$.

base e naquele nível de contração da rede.

Conforme os dados apresentados na tabela, quanto maior o grau médio dos vértices da rede, maior a eficiência dos algoritmos de *matching* para contrair a rede e, consequentemente, gerar os resultados de predição de *links* mais rapidamente. As bases *facebook* e *astro-ph* obtiveram os melhores resultados de *coarsening*, ou seja, foram as redes que ficaram proporcionalmente mais reduzidas em relação à rede original e possibilitaram a execução mais rápida dos algoritmos. No nível 5 de contração, os processos de predição de *links* e *uncoarsening* da rede contraída *facebook* levaram, em média, apenas 12,8% do tempo levado pelo mesmo algoritmo de predição na rede original. A segunda rede com maior valor de grau médio dos vértices, *astro-ph*, possui o segundo melhor resultado de tempo de execução com uma média de 33,3% do tempo necessário para execução em uma rede sem nenhum tipo de contração.

Apesar de utilizarem o algoritmo de *matching* mais rápido, nenhuma combinação com o algoritmo *RM* obteve o *ranking* de arestas preditas em menor tempo quando comparadas às outras combinações. Isso acontece pois a rede contraída pelo *RM* exige que mais similaridades sejam extraídas e portanto o processo fica mais demorado. As combinações que envolvem os algoritmos *LSVM* e *LEM* foram as que apresentaram mais vezes os menores tempos por níveis na tabela, sendo *SER-LEM* em cerca de 15% dos casos, *WSER-LEM* com 20% e *WSER-LEM* com cerca de 26%.

Para base de comparação, a Tabela 7 apresenta os valores médios de tempo de execução do índice de *Katz* para predição de *links* para as bases no nível 0 de contração em segundos.

Devido ao tempo disponível para realização dos experimentos e avaliação dos resultados, não foi possível avaliar a abordagem para redes maiores, com centenas de milhares de vértices e consequentemente bilhões de arestas não observadas, com o mesmo rigor estatístico das redes apresentadas anteriormente. Entretanto, alguns experimentos foram realizados em um subgrafo da rede *DBLP*⁸ com 100 mil vértices e 861413 arestas para levantar indícios do funcionamento do algoritmo para redes de maior complexidade estrutural. A rede em questão possui um alto valor de coeficiente de agrupamento e assortatividade, e um baixo valor de grau médio dos vértices. O algoritmo *MSVM* foi utilizado juntamente com o algoritmo de *uncoarsening* *SER* e os resultados foram obtidos para os níveis 0, 1 e 2 de *coarsening* da rede. Como também observado nas redes de menor tamanho, ocorre um ganho de tempo de cerca de 30% no primeiro nível de contração em relação ao nível 0, e o ganho cresce suavemente conforme o nível de *coarsening* aumenta, ficando em cerca de 40% no nível 2 de *coarsening* da rede. No nível 0 de contração, o algoritmo leva cerca de 62 horas para extrair uma predição completa desta rede, no nível 1 de contração o tempo cai para 43 horas e no nível 2 de contração o tempo é reduzido para cerca de 36 horas. Os resultados corroboram a hipótese apresentada anteriormente de que o ganho de tempo é maior quando o grau médio da rede é alto, como observado na rede *facebook*.

⁸ Disponível em <https://snap.stanford.edu/data/com-DBLP.html>

Base	N	SER					WSER				
		RM	HEM	LEM	MSVM	LSVM	RM	HEM	LEM	MSVM	LSVM
<i>netscience</i>	0	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
	1	0,780	0,779	0,772	0,777	0,774	0,781	0,772	0,768	0,782	0,758
	2	0,658	0,659	0,649	0,665	0,652	0,659	0,651	0,649	0,667	0,636
	3	0,611	0,619	0,605	0,626	0,612	0,615	0,607	0,606	0,629	0,597
	4	0,594	0,598	0,588	0,610	0,599	0,598	0,590	0,590	0,612	0,583
	5	0,586	0,591	0,579	0,599	0,591	0,590	0,581	0,582	0,604	0,575
<i>yeast</i>	0	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
	1	0,903	0,916	0,909	0,889	0,904	0,924	0,906	0,911	0,894	0,921
	2	0,721	0,715	0,725	0,754	0,777	0,742	0,725	0,717	0,745	0,767
	3	0,602	0,581	0,584	0,647	0,648	0,630	0,581	0,580	0,671	0,653
	4	0,531	0,508	0,541	0,594	0,575	0,555	0,553	0,523	0,592	0,574
	5	0,481	0,444	0,445	0,545	0,502	0,509	0,442	0,440	0,569	0,503
<i>facebook</i>	0	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
	1	0,425	0,402	0,373	0,353	0,450	0,409	0,410	0,400	0,388	0,441
	2	0,229	0,203	0,192	0,186	0,234	0,220	0,215	0,211	0,212	0,222
	3	0,169	0,149	0,141	0,142	0,161	0,163	0,159	0,154	0,162	0,153
	4	0,149	0,133	0,122	0,128	0,135	0,146	0,144	0,136	0,148	0,130
	5	0,141	0,126	0,114	0,122	0,124	0,139	0,137	0,128	0,142	0,121
<i>geom</i>	0	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
	1	0,834	0,835	0,832	0,828	0,822	0,836	0,829	0,824	0,837	0,833
	2	0,737	0,736	0,731	0,732	0,720	0,737	0,727	0,721	0,743	0,733
	3	0,686	0,676	0,682	0,687	0,674	0,692	0,676	0,668	0,700	0,685
	4	0,663	0,652	0,652	0,668	0,651	0,665	0,645	0,641	0,683	0,659
	5	0,647	0,633	0,635	0,657	0,635	0,649	0,631	0,625	0,672	0,641
<i>hep-th</i>	0	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
	1	0,881	0,885	0,889	0,910	0,888	0,889	0,895	0,878	0,892	0,888
	2	0,741	0,724	0,721	0,788	0,748	0,729	0,741	0,729	0,763	0,744
	3	0,645	0,636	0,627	0,709	0,637	0,646	0,652	0,660	0,663	0,666
	4	0,595	0,568	0,606	0,700	0,617	0,598	0,575	0,615	0,653	0,609
	5	0,552	0,565	0,573	0,681	0,544	0,562	0,563	0,546	0,630	0,587
<i>astro-ph</i>	0	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
	1	0,616	0,637	0,635	0,601	0,644	0,623	0,615	0,631	0,591	0,629
	2	0,443	0,448	0,453	0,449	0,493	0,455	0,424	0,460	0,440	0,476
	3	0,369	0,370	0,376	0,384	0,407	0,382	0,342	0,377	0,375	0,398
	4	0,342	0,338	0,346	0,363	0,370	0,357	0,314	0,346	0,356	0,361
	5	0,326	0,323	0,328	0,351	0,349	0,342	0,300	0,332	0,345	0,344
<i>condmat</i>	0	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
	1	0,732	0,755	0,751	0,674	0,716	0,741	0,739	0,742	0,692	0,683
	2	0,606	0,610	0,609	0,577	0,593	0,606	0,585	0,608	0,582	0,572
	3	0,520	0,510	0,517	0,495	0,498	0,525	0,492	0,514	0,500	0,473
	4	0,484	0,473	0,477	0,470	0,462	0,489	0,455	0,475	0,475	0,437
	5	0,466	0,453	0,456	0,458	0,443	0,474	0,437	0,455	0,464	0,422

Tabela 6 – Resultados de tempos de execução proporcionais ao tempo necessário para predição sem nenhum tipo de predição para as bases de dados.

	SER					WSER				
Base	<i>RM</i>	<i>HEM</i>	<i>LEM</i>	<i>MSVM</i>	<i>LSVM</i>	<i>RM</i>	<i>HEM</i>	<i>LEM</i>	<i>MSVM</i>	<i>LSVM</i>
<i>netscience</i>	11	11	11	11	11	11	1,000	11	11	11
<i>yeast</i>	18	17	17	17	17	17	17	17	17	17
<i>facebook</i>	36	34	34	32	32	37	34	35	32	32
<i>geom</i>	209	211	208	211	209	209	208	209	208	210
<i>hep-th</i>	809	801	799	820	818	810	801	800	822	821
<i>astro-ph</i>	2824	2685	2743	2671	2687	2697	2982	2694	2709	2739
<i>condmat</i>	4050	3967	4057	4373	4324	4054	4133	4003	4322	4516

Tabela 7 – Resultados de tempos de execução do algoritmo de predição de *links* em segundos para o nível 0 de contração.

CONCLUSÕES

Este capítulo apresenta as considerações finais do projeto que foi desenvolvido, principalmente sobre as principais contribuições, aspectos positivos e limitações da abordagem desenvolvida.

A necessidade de abordagem para tornar a predição de *links* mais escalável, juntamente com a falta de aplicações da abordagem multinível nesta tarefa, motivaram as hipóteses e o desenvolvimento deste trabalho. A partir do levantamento bibliográfico realizado, foram observadas certas limitações nas poucas abordagens que tinham como objetivo acelerar o processo de predição de *links* e uma abordagem que não possui nenhuma destas limitações foi desenvolvida. A abordagem utiliza o paradigma de otimização multinível para gerar uma versão contraída da rede, aplicar o algoritmo (qualquer que seja) na rede contraída e então projetar os resultados iniciais obtidos, gerando um *ranking* completo de arestas não observadas. Dentre as contribuições deste projeto destacam-se:

- A utilização do paradigma de otimização multinível em uma tarefa até então inédita, o problema de predição de *links*;
- A criação de novos algoritmos de *matching* que obtiveram bons resultados para o problema, inclusive melhores do que os algoritmos considerados da literatura;
- A criação de dois algoritmos para a projeção dos resultados iniciais, obtidos pela aplicação dos algoritmos de predição de *links* em redes contraídas, para a geração do *ranking* completo de predições.

A estratégia de otimização multinível desenvolvida pode ser utilizada para redes de qualquer domínio, porém os experimentos realizados apresentam indícios de que os melhores resultados são alcançados em redes sociais que apresentem valores elevados nos coeficiente de agrupamento e assortatividade. Além de ser utilizável para qualquer rede, a estratégia é

indicada para situações em que todas as arestas não observadas devam ser avaliadas pelo preditor, ou seja, em situações em nenhum par de vértices possa ser desconsiderado pela predição de *links*. Encontramos também que o nível de contração 3 apresenta o melhor balanço entre perdas aceitáveis de desempenho e o ganho de tempo na predição. Embora resultados satisfatórios tenham sido alcançados, a abordagem desenvolvida apresenta algumas limitações, discutidas a seguir.

5.1 Limitações da abordagem

Embora apresente uma diminuição significativa do tempo de predição, extraindo todos as arestas não observadas possíveis das redes estudadas, a principal limitação da abordagem observada no capítulo de experimentos é que, para algumas bases de dados, os valores de *precisi@L* tornam a abordagem inviável. Em compensação, os valores de *AUC* obtidos mostram que a abordagem é plenamente viável na maioria das bases de dados testadas. A principal hipótese é que os baixos valores de *precisi@L* em algumas redes se deve à simplicidade dos algoritmos de *uncoarsening* desenvolvidos que fazem apenas a replicação dos *scores* obtidos via predição na rede reduzida, não levando em consideração nenhuma característica da estrutura de vizinhança interna dos vértices condensados em super-vértices.

Além disso, observamos que, embora a redução de tempo seja muito significativa para algumas das bases de dados estudadas, o processo de *uncoarsening* ainda é um gargalo que impede que o processo seja mais rápido. Mesmo com a replicação de *scores* de arestas substituindo a maioria dos cálculos de similaridades, o processo ainda exige que muitas operações de replicação sejam realizadas. Em redes massivas, com milhões de vértices, o processo multinível reduziria o número de operações e consequentemente o tempo de execução do processo de predição de *links* mas não o tornaria viável, pois ainda seriam necessárias muitas operações no *uncoarsening* da solução inicial. A abordagem não foi devidamente testada com o mesmo rigor em redes de maior complexidade estrutural, que atinjam centenas de milhares de vértices, de forma que sua viabilidade fosse avaliada em ambientes em que abordagens de escalabilidade são imprescindíveis.

Outra limitação da abordagem proposta é que a mesma não foi projetada para lidar com redes direcionadas. Para tal ampliação da abordagem são necessárias alterações nos algoritmos de *matching* propostos, de forma a considerar apenas vizinhos que possuam arestas de entrada como possíveis candidatos para o *matching*. Outra alteração necessária seria no *uncoarsening*, em que além da projeção da existência das arestas entre super-vértices, seria também necessário projetar a direção de tais arestas, pois se todas as arestas forem consideradas como sendo da mesma direção, tal situação poderia criar redes desconectadas, fazendo com que a qualidade da predição seja prejudicada. Além da questão da direção das arestas, outra limitação do trabalho é que a abordagem não foi testada para a predição de pesos de arestas, apenas para a predição

da existência ou não existência de arestas na rede, e os pesos das arestas das redes só são considerados para a criação do conjunto de *matching* nos algoritmos *HEM* e *LEM*.

5.2 Trabalhos futuros

Vistos os principais aspectos positivos e limitações do trabalho desenvolvido, algumas possibilidades de trabalhos futuros surgiram para aprimorar a abordagem desenvolvida, aumentando ainda mais a viabilidade de aplicação do método. Portanto, na continuação desta pesquisa, os seguintes pontos serão abordados:

- As hipóteses de comportamento dos algoritmos de *matching* propostos devem ser melhor testadas, pois tais informações podem contribuir para o desenvolvimento de abordagens mais sofisticadas que utilizam otimização multinível.
- Aplicar a abordagem multinível para outros métodos de predição de *links*, principalmente para abordagens supervisionadas. Através da contração da rede, os atributos podem ser extraídos da rede $G(V, E^{lr})$ e a classificação produzida ser projetada para as arestas não observadas incidentes aos vértices. Dessa forma, o tempo de processamento necessário para o cálculo de atributos para a composição do conjunto de treinamento é reduzido. Além disso, como os atributos extraídos para um super-vértice podem representar todos os vértices condensados que geraram o super-vértice, o tempo de treinamento e a complexidade do modelo supervisionado gerado também seriam reduzidos.
- A principal hipótese do baixo desempenho de predição em algumas bases de dados é que os algoritmos de *uncoarsening* propostos que replicam os *scores* obtidos não sejam sofisticados, pois apenas fazem a replicação dos *scores* obtidos entre arestas formadas por super-vértices. De forma a melhorar a qualidade da predição, é interessante desenvolver algoritmos de *uncoarsening* mais sofisticados que levem em consideração características dos vértices que foram condensados em super-vértices.
- Como a perda de qualidade devido à contração da rede é comum na área de otimização multinível, existem diversas abordagens para o refinamento, durante a fase de *uncoarsening*, da solução inicial encontrada na fase de busca local. De forma a melhorar o desempenho de predição na abordagem proposta, é interessante desenvolver métodos de refinamento da solução inicial que possam atuar principalmente nas arestas não observadas que existem entre vértices no interior de um mesmo super-vértice.
- Como o processo de *coarsening* da rede condensa vértices da rede, criando super-vértices que estão em vizinhanças de super-vértices, surgiu uma hipótese de utilizar subgrafos de super-vértices para selecionar arestas não observadas. Ao invés de usar caminhos entre vértices, utilizaremos caminhos entre super-vértices, de forma a criar uma redução do

número de arestas não observadas que serão consideradas pelo algoritmo de predição de *links*, porém diminuindo o risco de se perder arestas importantes, já que os super-vértices foram condensados com viés de similaridade.

REFERÊNCIAS

- ADAMIC, L. A.; ADAR, E. Friends and neighbors on the web. **Social networks**, Elsevier, v. 25, n. 3, p. 211–230, 2003. Citado nas páginas 39 e 44.
- AHMED, N. M.; CHEN, L.; WANG, Y.; LI, B.; LI, Y.; LIU, W. Sampling-based algorithm for link prediction in temporal networks. **Information Sciences**, Elsevier Inc., v. 374, p. 1–14, dec 2016. Citado na página 44.
- ANIL, A.; KUMAR, D.; SHARMA, S.; SINGHA, R.; SARMAH, R.; BHATTACHARYA, N.; SINGH, S. R. Link Prediction Using Social Network Analysis over Heterogeneous Terrorist Network. In: **2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)**. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2015. p. 267–272. Citado na página 41.
- BENCHETTARA, N.; KANAWATI, R.; ROUVEIROL, C. Supervised machine learning applied to link prediction in bipartite social networks. In: **Proceedings - 2010 International Conference on Advances in Social Network Analysis and Mining, ASONAM 2010**. [S.l.: s.n.], 2010. p. 326–330. Citado na página 40.
- BICHOT, C.-E. Graph partitioning. In: _____. [S.l.]: ISTE - Wiley, 2014. cap. A Partitioning requiring rapidity and quality: The multilevel method and partitions refinement algorithms, p. 27–63. Citado na página 47.
- BOCCALETTI, S.; LATORA, V.; MORENO, Y.; CHAVEZ, M.; HWANG, D.-U. Complex networks : Structure and dynamics. **Phys. Rep.**, v. 424, n. 4-5, p. 175–308, Fevereiro 2006. Citado na página 33.
- BU, D.; ZHAO, Y.; CAI, L.; XUE, H.; ZHU, X.; LU, H.; ZHANG, J.; SUN, S.; LING, L.; ZHANG, N. *et al.* Topological structure analysis of the protein-protein interaction network in budding yeast. **Nucleic acids research**, Oxford University Press, v. 31, n. 9, p. 2443–2450, 2003. Citado na página 57.
- CHERNOSKUTOV, M.; INEICHEN, Y.; BEKAS, C. Heuristic algorithm for approximation betweenness centrality using graph coarsening. **Procedia Computer Science**, v. 66, p. 83–92, 2015. Citado na página 47.
- CHOUDHURY, N.; UDDIN, S. Time-aware link prediction to explore network effects on temporal knowledge evolution. **Scientometrics**, v. 108, n. 2, p. 745–776, 2016. Citado na página 40.
- COHEN, S.; COHEN-TZEMACH, N. Implementing link-prediction for social networks in a database system. **Proceedings of the ACM SIGMOD Workshop on Databases and Social Networks**, p. 37–42, 2013. Citado na página 43.
- CORBELLINI, A.; GODOY, D.; MATEOS, C.; SCHIAFFINO, S.; ZUNINO, A. **DPM: A novel distributed large-scale social graph processing framework for link prediction algorithms**. [S.l.]: Elsevier B.V., 2016. Citado na página 43.

- COSKUN, M.; KOYUTURK, M. Link Prediction in Large Networks by Comparing the Global View of Nodes in the Network. **Proceedings - 15th IEEE International Conference on Data Mining Workshop, ICDMW 2015**, p. 485–492, 2016. Citado na página 44.
- COSTA, L. d. F.; OLIVEIRA, O. N.; TRAVIESO, G.; RODRIGUES, F. A.; BOAS, P. R. V.; ANTIQUEIRA, L.; VIANA, M. P.; ROCHA, L. E. C. da. Analyzing and Modeling Real-World Phenomena with Complex Networks: A Survey of Applications. **Advances in Physics**, v. 60, p. 1–108, 2007. Citado nas páginas 27 e 33.
- CUI, W.; PU, C.; XU, Z.; CAI, S.; YANG, J.; MICHAELSON, A. Bounded link prediction in very large networks. **Physica A: Statistical Mechanics and its Applications**, Elsevier B.V., v. 457, p. 202–214, sep 2016. Citado na página 44.
- CUKIERSKI, W.; HAMNER, B.; YANG, B. Graph-based features for supervised link prediction. **Proceedings of the International Joint Conference on Neural Networks**, p. 1237–1244, 2011. Citado na página 40.
- De Sá, H. R.; PRUDÊNCIO, R. B. C. Supervised link prediction in weighted networks. **Proceedings of the International Joint Conference on Neural Networks**, p. 2281–2288, 2011. Citado na página 40.
- ERDŐS, P.; RÉNYI, A. On the evolution of random graphs. In: **PUBLICATION OF THE MATHEMATICAL INSTITUTE OF THE HUNGARIAN ACADEMY OF SCIENCES**. [S.l.: s.n.], 1960. p. 17–61. Citado na página 61.
- ESSLIMANI, I.; BRUN, A.; BOYER, A. Densifying a behavioral recommender system by social networks link prediction methods. **Social Network Analysis and Mining**, Springer-Verlag Wien, v. 1, n. 3, p. 159–172, jan 2011. Citado na página 42.
- FAN, C.; XIU, B.; LIU, Z.; CHEN, C.; YANG, Y. Topology reconstruction for target operation network (TON): A link prediction perspective. In: . [S.l.]: University of Plymouth, 2016. p. 55–62. Citado na página 42.
- FIRE, M.; TENENBOIM, L.; LESSER, O.; PUZIS, R.; ROKACH, L.; ELOVICI, Y. Link prediction in social networks using computationally efficient topological features. **Proceedings - 2011 IEEE International Conference on Privacy, Security, Risk and Trust and IEEE International Conference on Social Computing, PASSAT/SocialCom 2011**, n. October 2015, p. 73–80, 2011. Citado na página 43.
- GE, L.; ZHANG, A. Pseudo Cold Start Link Prediction with Multiple Sources in Social Networks. **Sdm**, p. 768–779, 2012. Citado na página 38.
- GETOOR, L.; DIEHL, C. P. Link mining. **ACM SIGKDD Explorations Newsletter**, v. 7, n. 2, p. 3–12, 2005. Citado nas páginas 27 e 37.
- HASAN, M. A.; CHAOJI, V.; SALEM, S.; ZAKI, M. Link prediction using supervised learning. In: **In Proc. of SDM 06 workshop on Link Analysis, Counterterrorism and Security**. [S.l.: s.n.], 2006. Citado nas páginas 39 e 40.
- HUANG, Z.; LI, X.; CHEN, H. Link prediction approach to collaborative filtering. In: **Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries**. [S.l.]: ACM, 2005. (JCDL '05), p. 141–142. Citado na página 42.

IAKOVIDOU, N.; SYMEONIDIS, P.; MANOLOPOULOS, Y. Multiway spectral clustering link prediction in protein-protein interaction networks. **Proceedings of the 10th IEEE International Conference on Information Technology and Applications in Biomedicine**, p. 1–4, 2010. Citado na página [41](#).

JACCARD, P. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. **Bulletin del la Société Vaudoise des Sciences Naturelles**, v. 37, p. 547–579, 1901. Citado na página [38](#).

JEH, G.; WIDOM, J. Simrank: A measure of structural-context similarity. In: **Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: ACM, 2002. p. 538–543. Citado na página [39](#).

KARYPIS, G.; KUMAR, V. Analysis of multilevel graph partitioning. In: **Proceedings of the 1995 ACM/IEEE Conference on Supercomputing**. New York, NY, USA: ACM, 1995. (Supercomputing '95). Citado na página [47](#).

_____. A fast and high quality multilevel scheme for partitioning irregular graphs. **Society for Industrial and Applied Mathematics**, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, v. 20, n. 1, p. 359 – 392, dez. 1998. Citado nas páginas [45](#), [46](#) e [47](#).

KATZ, L. A new status index derived from sociometric analysis. **Psychometrika**, v. 18, n. 1, p. 39–43, March 1953. Citado nas páginas [39](#) e [60](#).

KAYA, B.; POYRAZ, M. Supervised link prediction in symptom networks with evolving case. **Measurement: Journal of the International Measurement Confederation**, Elsevier Ltd, v. 56, p. 231–238, 2014. Citado na página [42](#).

_____. Unsupervised link prediction in evolving abnormal medical parameter networks. **International Journal of Machine Learning and Cybernetics**, Springer Verlag, v. 7, n. 1, p. 145–155, feb 2016. Citado na página [42](#).

LEI, C.; RUAN, J. A novel link prediction algorithm for reconstructing protein-protein interaction networks by topological similarity. **Bioinformatics**, v. 29, n. 3, p. 355–364, 2013. Citado na página [41](#).

LI, W.; MEI, X.; WANG, H.; ZHOU, Y.; HUANG, J. Link prediction boosted psychiatry disorder classification for functional connectivity network. In: . [S.l.]: SPIE, 2017. v. 10225, p. 102252F. Citado na página [42](#).

LI, X.; CHEN, H. Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach. **Decision Support Systems**, v. 54, n. 2, p. 880–890, 2013. Citado na página [42](#).

LIBEN-NOWELL, D.; KLEINBERG, J. The Link Prediction Problem for Social Networks. **Proceedings of the Twelfth Annual ACM International Conference on Information and Knowledge Management (CIKM)**, p. 556–559, 2003. Citado nas páginas [39](#) e [41](#).

LORRAIN, F.; WHITE, H. C. Structural equivalence of individuals in social networks. **The Journal of Mathematical Sociology**, Routledge, v. 1, n. 1, p. 49–80, 1971. Citado na página [38](#).

- LU, L.; ZHOU, T. Link Prediction in Complex Networks: A Survey. **Physica A: Statistical Mechanics and its Applications**, v. 390, n. 6, p. 1150–1170, 2010. Citado nas páginas 38, 39 e 59.
- LU, Y.; GUO, Y.; KORHONEN, A. Link prediction in drug-target interactions network using similarity indices. **BMC bioinformatics**, BioMed Central Ltd., v. 18, n. 1, p. 39, jan 2017. Citado na página 42.
- MARTÍNEZ, V.; BERZAL, F.; CUBERO, J.-C. A Survey of Link Prediction in Complex Networks. **ACM Computing Surveys**, v. 49, n. 4, p. 1–33, 2016. Citado nas páginas 37 e 38.
- MOHAN, A.; VENKATESAN, R.; PRAMOD, K. V. A scalable method for link prediction in large real world networks. **Journal of Parallel and Distributed Computing**, Academic Press Inc., v. 109, p. 89–101, nov 2017. Citado na página 44.
- NEWMAN, M. **Networks: An Introduction**. New York: Oxford University Press, Inc, 2010. Citado nas páginas 27, 34 e 35.
- NEWMAN, M. E. J. The structure of scientific collaboration networks. **Proceedings of the National Academy of Sciences of the United States of America**, Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501, USA. mark@santafe.edu, v. 98, n. 2, p. 404–409, January 2001. Citado na página 58.
- _____. Assortative mixing in networks. **Physical Review Letters**, American Physical Society, v. 89, n. 20, p. 208701, out. 2002. Citado na página 36.
- _____. Finding community structure in networks using the eigenvectors of matrices. **Phys. Rev. E**, American Physical Society, v. 74, p. 036104, Sep 2006. Citado na página 57.
- OGATA, H.; SUZUMURA, T. Towards Scalable X10 Based Link Prediction for Large Scale Social Networks. **Proceedings of the 23rd International Conference on World Wide Web**, p. 1327–1332, 2014. Citado na página 44.
- PAPADIMITRIOU, A.; SYMEONIDIS, P.; MANOLOPOULOS, Y. Scalable link prediction in social networks based on local graph characteristics. **Proceedings of the 9th International Conference on Information Technology, ITNG 2012**, p. 738–743, 2012. Citado na página 43.
- PAVLOV, M.; ICHISE, R. Finding experts by link prediction in co-authorship networks. **CEUR Workshop Proceedings**, v. 290, p. 42–55, 2007. Citado na página 41.
- SALTON, G.; MCGILL, M. J. **Introduction to Modern Information Retrieval**. New York, NY, USA: McGraw-Hill, Inc., 1986. Citado na página 38.
- SCELLATO, S.; NOULAS, A.; MASCOLO, C. Exploiting place features in link prediction on location-based social networks. **Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '11**, n. Section 3, p. 1046, 2011. Citado na página 43.
- SYMEONIDIS, P.; MANTAS, N. Spectral clustering for link prediction in social networks with positive and negative links. **Social Network Analysis and Mining**, v. 3, n. 4, p. 1433–1447, 2013. Citado na página 43.

VALEJO, A.; OLIVEIRA, M. C. F. de; FILHO, G. P.; LOPES, A. de A. Multilevel approach for combinatorial optimization in bipartite network. **Knowledge-Based Systems**, v. 151, p. 45 – 61, 2018. ISSN 0950-7051. Citado na página 45.

WALSHAW, C. A multilevel algorithm for force-directed graph drawing. In: **Proceedings of the Graph Drawing: 8th International Symposium, GD**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001. p. 171–182. Citado na página 47.

_____. A multilevel approach to the travelling salesman problem. **Annals of Operations Research**, v. 50, n. 5, p. 862–877, set. 2002. Citado na página 47.

_____. Multilevel refinement for combinatorial optimisation problems. **Annals of Operations Research**, v. 131, n. 2, 2004. Citado na página 47.

ZHANG, Y.; LI, F.; XU, B.; GAO, K.; YU, G. Using non-topological node attributes to improve results of link prediction in social networks. **Proceedings - 9th Web Information Systems and Applications Conference, WISA 2012**, p. 141–146, 2012. Citado na página 38.