ARTHUR PATRICIO GRAVA

Sistema de recomendação de artigos científicos utilizando dados sociais

São Paulo 2016

ARTHUR PATRICIO GRAVA

Sistema de recomendação de artigos científicos utilizando dados sociais

Dissertação apresentada à Escola de Artes, Ciências e Humanidades da Universidade de São Paulo para obtenção do título de Mestre em Ciências pelo Programa de Pós-graduação em Sistemas de Informação.

Área de concentração: Inteligência Computacional

Versão corrigida contendo as alterações solicitadas pela comissão julgadora em 21 de junho de 2016. A versão original encontra-se em acervo reservado na Biblioteca da EACH-USP e na Biblioteca Digital de Teses e Dissertações da USP (BDTD), de acordo com a Resolução CoPGr 6018, de 13 de outubro de 2011.

Orientador: Prof. Dr. Luciano Antonio Digiampietri

São Paulo 2016 Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

CATALOGAÇÃO-NA-PUBLICAÇÃO

(Universidade de São Paulo. Escola de Artes, Ciências e Humanidades. Biblioteca)

Grava, Arthur Patricio

Sistema de recomendação de artigos científicos utilizando dados sociais / Arthur Patricio Grava ; orientador, Luciano Antonio Digiampietri. – São Paulo, 2016 93 f. : il.

Dissertação (Mestrado em Ciências) - Programa de Pós-Graduação em Sistemas de Informação, Escola de Artes, Ciências e Humanidades, Universidade de São Paulo Versão corrigida

Inteligência computacional.
 Sistemas de recomendação.
 Artigo científico - Recomendação.
 Produção científica - Recomendação.
 Redes sociais.
 Bases de dados.
 Bibliometria.
 Digiampietri, Luciano Antonio, orient.
 Título

CDD 22.ed.- 006.3

Dissertação de autoria de Arthur Patricio Grava, sob o título "Sistema de recomendação de artigos científicos utilizando dados sociais", apresentada à Escola de Artes, Ciências e Humanidades da Universidade de São Paulo, para obtenção do título de Mestre em Ciências pelo Programa de Pós-graduação em Sistemas de Informação, na área de concentração Inteligência Computacional, aprovada em 21 de Junho de 2016 pela comissão julgadora constituída pelos doutores:

Prof. Dr. Luciano Antonio Digiampietri

Presidente Instituição: Universidade de São Paulo (USP)

Prof. Dr. Sarajane Marques Peres Instituição: Universidade de São Paulo (USP)

Prof. Dr. Jesús Pascual Mena Chalco Instituição: Universidade Federal do ABC (UFABC)

Agradecimentos

Agradeço aos meus familiares por estarem sempre ao meu lado durante este percurso sempre demonstrando total apoio e confiança.

À minha companheira Samira que me deu todo o suporte e ouvidos nos últimos meses, além de me ajudar de todas as maneiras possíveis e a seu alcance.

Ao meu orientador Luciano que me ensinou muito, demostrou total parceria e me guiou durante esses anos tão especiais.

Aos meus amigos Sionenses que mesmo em meus períodos de aflição e dificuldade souberam transformar angústias em ótimas ridasas, além de sua participação fundamental na formação da pessoa que sou hoje.

Aos meus amigos e líderes do Luizalabs que, em todo o momento, demonstraram total suporte e tranquilidade nesta trajetória e a empresa em si, que financiou minha participação no RecSys'14.

Resumo

GRAVA, Arthur Patricio. Sistema de recomendação de artigos científicos utilizando dados sociais. 2016. 93 f. Dissertação (Mestrado em Ciências) – Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, São Paulo, 2016.

Sistemas de recomendação estão se tornando ferramentas indispensáveis para diversos websites, que buscam oferecer ao seu usuário uma experiência personalizada e simplificada, e sua adoção se deve principalmente devido ao grande volume de dados disponíveis, advindos de diferentes fontes e contendo informações diversificadas, aumentando a necessidade e a complexidade de se extrair valor desses dados. Com o surgimento de redes sociais online os usuários passaram a expressar seus gostos e preferências além de estabelecer relações com outros usuários, podendo estes serem seus amigos, parentes, ídolos, etc. Estas possibilidades encontradas em redes sociais motivou o presente trabalho a interpretar a comunidade científica como uma rede social, utilizando relações de coautoria, colaboração em projetos, orientações, além de citações de trabalhos e, consequentemente, citações aos respectivos autores. O objetivo deste projeto foi propor um sistema de recomendação de trabalhos científicos combinando informações sociais e informações bibliométricas, no que diz respeito a artigos citados em publicações, caraterizando-se como um facilitador para auxiliar os pesquisadores a responderem perguntas como: Quais artigos interessantes da minha área eu ainda não tenho conhecimento? e Quais artigos podem auxiliar em trabalhos que tenho em desenvolvimento? Para atingir o objetivo proposto foram desenvolvidas duas abordagens de recomendação. A primeira abordagem teve como premissa que o tempo em que as relações entre os autores foi estabelecida é determinante para selecionar os autores mais próximos (ou similares), ou seja, as relações mais recentes tendem a ser mais relevantes que as relações mais antigas. Já a segunda técnica combinou o resultados das diferentes técnicas implementadas (tanto a proposta quanto técnicas da literatura correlata) para gerar novas recomendações de maneira híbrida. Os resultados mostraram que a solução baseada no tempo apresentou resultados superiores às estratégias correlatas quando se possui mais informações sobre o autor, ou seja, autores que possuem diversas relações de coautoria e um conjunto de artigos citados elevado tendem a obter resultados melhores quando comparados aos autores que possuem poucas relações e citaram poucos artigos. Já a solução híbrida, que combina os resultados dos diversos recomendadores, apresentou uma cobertura de recomendações superior às demais, pelo fato de combinar os pontos fortes de cada uma das técnicas, encontrando recomendações relevantes no conjunto de testes em mais de 57% dos casos.

Palavras-chaves: Sistema de recomendação, Filtro social, Redes sociais, Recomendação social, Contextualização temporal

Abstract

GRAVA, Arthur Patricio. Papers recommender system using social information. 2016. 93 p. Dissertation (Master of Science) – School of Arts, Sciences and Humanities, University of São Paulo, São Paulo, 2016.

Recommender systems are becoming indispensable tools on websites, in order to offer a simplified and personalized experience to their users, and its adoption is due to the fact that the volume of data available has increased and also comes from different sources with different types of information. Thus, it is challenge and necessary tools for helping to extract more valuable information from these data. The arise of online social networks allowed users to express their tastes and preferences and establish relationships with other users, such as friends, relatives, idols, etc. Those possibilities found in social networks motivated this work to interpret the scientific community as a social network, providing the ability to use co-authorship relations, collaboration in projects, tutoring relations, as well as paper citations and thus citations from their authors. The goal of this project was to propose a papers recommender system combining social and bibliometric information, regarding cited articles on published papers, being characterized as a facilitator to help researchers to answer questions such as: "What interesting articles in my area I still have no knowledge of?" and "Which articles can assist in the project I am developing?". The first algorithm proposed used the time when the coauthorship relations among authors were established as a determining parameter to choose which authors are more similar, meaning that relations established in recent time are more relevant than those that are older. The second algorithm combines the results from different implemented algorithms to determine which would be the ideal weight of each algorithm on the recommendation result, using a linear regression on the recommendations scores. The results showed that the time based solution achieved a better performance for the authors with higher amount of information available, i.e., if the author has many coauthorship relations and cited many papers, the results are better when compared with authors that does not have many relations and cited articles. On the other hand, the hybrid solution which combines the results from different recommendations approaches presented a higher coverage compared with others, due to the fact that it combines the strengths of each one of the algorithms, finding recommendation for users on 57% of the cases.

Keywords: Recommender system, Social filtering, Social network, Social recommendation, Time context

Lista de figuras

Figura 1 –	Representação de uma rede de confiança a partir de grafos	24
Figura 2 –	Distribuição anual dos artigos publicados.	35
Figura 3 –	Técnicas empregadas durante os anos.	36
Figura 4 –	Métricas utilizadas para comparar as abordagens desenvolvidas	37
Figura 5 –	Distribuição de autores por sua quantidade de publicações	44
Figura 6 –	Distribuição de autores por suas citações, limitados até 200 citações	45
Figura 7 –	Distribuição das publicações através dos anos	46
Figura 8 -	Representação do arquivo de citações em memória para o cálculo da	
	similaridade entre os autores	48
Figura 9 –	Ilustração dos resultados para todas as métricas e para os melhores	
	resultados encontrados em cada uma das execuções dos algoritmos para	
	cada K, utilizando-se os autores de W_{teste}	82
Figura 10 –	Ilustração dos resultados para todas as métricas e para os melhores	
	resultados encontrados em cada uma das execuções dos algoritmos para	
	cada K, utilizando-se os autores de R_n	83

Lista de algoritmos

Algoritmo 1 – Algoritmo para calcular a correlação de <i>Pearson</i> entre dois vetores de	
citações, utilizando as listas de citações pré-existente de dois autores	48
Algoritmo 2 – Algoritmo para paralelizar a execução da correlação <i>Pearon</i>	49
Algoritmo 3 – Cálculo de filtro colaborativo baseado em usuário	51
Algoritmo 4 – Paralelismo obtido atravé de threads que calculam a predição dos itens para	
um determinado autor	51
Algoritmo 5 – Definição de confiança entre coautores	52
Algoritmo 6 – Cálculo do $score$ para as possíveis citações para um autor a utilizando os	
autores em que a confia	53
Algoritmo 7 – Cálculo do SRS para um autor a qualquer	54
Algoritmo 8 – Cálculo do SRS* para um autor a qualquer	54
Algoritmo 9 – Cálculo do TSRS para um autor a qualquer	55
Algoritmo 10 – Cálculo do TSRS* para um autor a qualquer	56
Algoritmo 11 – Cálculo do Time-TSRS para um autor a qualquer	58

Lista de tabelas

Tabela I –	Sumário das Informações Extraídas na Revisão: Técnicas implementadas	
	e dados utilizados	33
Tabela 2 –	Sumário das Informações Extraídas na Revisão - continuação	34
Tabela 3 –	Métricas da base de dados Epinions	37
Tabela 4 –	Exemplo de arquivo extraído a partir do conjunto da WoS	47
Tabela 5 –	Modelo em formato de tabela do arquivo resultante do cálculo de	
	similaridade entre os autores	49
Tabela 6 –	Modelo em formato de tabela do arquivo resultante do cálculo de	
	recomendações do algoritmo UCF	52
Tabela 7 –	Formato do arquivo que contém todos os $scores$ dos algoritmos utilizados.	58
Tabela 8 –	Métricas coletadas a partir da execução do algoritmo UCF verificando	
	todos os autores e artigos no conjunto de testes W_{teste}	61
Tabela 9 –	Métricas coletadas a partir da execução do algoritmo UCF verificando	
	todos os autores do conjunto de recomendações	62
Tabela 10 –	Métricas coletadas a partir da execução do algoritmo TRS verificando	
	todos os autores no conjunto de testes W_{teste}	64
Tabela 11 –	Métricas coletadas a partir da execução do algoritmo TRS verificando	
	todos os autores do conjunto de recomendações	65
Tabela 12 –	Métricas coletadas a partir da execução do algoritmo SRS verificando	
	todos os autores no conjunto de testes W_{teste}	66
Tabela 13 –	Métricas coletadas a partir da execução do algoritmo SRS verificando	
	todos os autores do conjunto de recomendações	67
Tabela 14 –	Métricas coletadas a partir da execução do algoritmo SRS* verificando	
	todos os autores no conjunto de testes W_{teste}	68
Tabela 15 –	Comparação entre SRS* e SRS utilizando 5 vizinhos e as primeiras 25	
	recomendações a partir de W_{teste}	69
Tabela 16 –	Métricas coletadas a partir da execução do algoritmo SRS* verificando	
	todos os autores do conjunto de recomendações	70
Tabela 17 –	Comparação entre SRS* e SRS utilizando 5 vizinhos e as primeiras 25	
	recomendações a partir de R_n	70

Tabela 18 –	Métricas coletadas a partir da execução do algoritmo TSRS verificando	
	todos os autores no conjunto de testes W_{teste}	71
Tabela 19 –	Métricas coletadas a partir da execução do algoritmo TSRS* verificando	
	todos os autores do conjunto de recomendações	72
Tabela 20 –	Métricas coletadas a partir da execução do algoritmo TSRS* verificando	
	todos os autores no conjunto de testes W_{teste}	74
Tabela 21 –	Comparação entre TSRS* e TSRS utilizando 5 vizinhos e as primeiras	
	25 recomendações a partir de W_{teste}	74
Tabela 22 –	Métricas coletadas a partir da execução do algoritmo TSRS* verificando	
	todos os autores do conjunto de recomendações	75
Tabela 23 –	Comparação entre TSRS* e TSRS utilizando 5 vizinhos e as primeiras	
	25 recomendações a partir de R_5	75
Tabela 24 –	Métricas coletadas a partir da execução do algoritmo Time-TSRS	
	verificando todos os autores no conjunto de testes W_{teste}	76
Tabela 25 –	Métricas coletadas a partir da execução do algoritmo Time-TSRS	
	verificando todos os autores do conjunto de recomendações	78
Tabela 26 –	Pesos encontrados para cada um dos algoritmos variando o número	
	mínimo de scores necessários	79
Tabela 27 –	Métricas coletadas a partir da hibridização utilizando regressão verifi-	
	cando todos os autores no conjunto de testes W_{teste}	79
Tabela 28 –	Métricas coletadas a partir a partir da hibridização utilizando regressão	
	verificando todos os autores do conjunto de recomendações	80

Lista de abreviaturas e siglas

AUC Área sobre a curva

BD Banco de dados

BF Filtro colaborativo baseado em redes de comportamento

BIB Recomendação baseada em redes Bayesianas

CF Filtro colaborativo

CIF Recomendação baseada em círculos sociais

CxF Filtro colaborativo contextual

ER Estratégia de recomendação

GBA Algoritmo genético baseado em grafos

LBSR Sistema de recomendação baseado em localização

LF Filtro colaborativo baseado em localização

LSF Filtro colaborativo baseado em localização e redes sociais

MAP Média da precisão média

MRR Erro recíproco médio

NMAE Erro médio absoluto normalizado

NRMSE Raiz do erro quadrático médio normalizado

PMF Fatoração de matrizes probabilísticas

PSF Filtro social probabilístico

RMSE Raiz do erro quadrático médio

RWR Caminhada aleatória com reinicialização

S@K Sucessos no conjunto

SC Soluções comparadas

SRS Sistema de recomendação social

SRS* Sistema de recomendação social utilizando similidade

SF Filtro social

SocialMF Fatoração de matrizes com dados sociais

TA Técnida de avaliação

Time-TSRS Sistema de recomendação social com confiança e contextualização

temporal

TSRS Sistema de recomendação social inserindo dados de confiança

UCF Filtro colaborativo baseado em usuários

WOS Web of Science

Sumário

1	Introdução	16
1.1	Objetivo	17
1.1.1	Objetivos específicos	18
1.2	Método	18
1.2.1	Procedimento de experimentação e análise de resultados	18
1.3	Organização	20
2	Conceitos fundamentais	21
2.1	Recomendações utilizando filtro colaborativo	21
2.2	Recomendações utilizando dados sociais	23
2.2.1	Recomendação baseada em confiança	24
2.3	Preocupações em sistemas de recomendação	25
2.3.1	Novos usuários do sistema e serendipidade	26
2.3.2	Segurança	26
2.4	Métricas	27
2.4.1	Sucesso no conjunto	27
2.4.2	Erro recíproco médio	28
2.4.3	Precisão e revocação	28
2.4.4	Erro absoluto médio e raiz do erro quadrático médio	29
3	Revisão da literatura correlata	31
3.1	Planejamento	31
3.2	Condução	31
3.2.1	Seleção dos dados	32
3.2.2	Extração	32
3.3	Resultados	32
3.3.1	Epinions	35
3.3.1.1	Redes de confiança	36
3.3.1.2	Fatoração de matrizes com confiança	38
3.4	Considerações	40
3.5	Outras pesquisas	41

4	Solução proposta	43
4.1	Conjunto de dados	43
4.1.1	Web of science	43
4.1.2	Seleção de dados	44
4.2	Similaridade entre autores	47
4.3	Técnicas comparativas implementadas	50
4.3.1	Filtro colaborativo baseado em usuário	50
4.3.2	Recomendação baseada em redes de confiança	52
4.3.3	Recomendação baseada em redes sociais	53
4.3.4	Recomendação baseada em redes sociais com confiança .	55
4.4	Técnicas propostas	56
5	Resultados	60
5.1	Experimentação	60
5.1.1	Experimento UCF	61
5.1.2	Experimento TRS	63
5.1.3	Experimento SRS	65
5.1.3.1	SRS	66
5.1.3.2	SRS*	68
5.1.4	Experimento TSRS	71
5.1.4.1	TSRS	71
5.1.4.2	TSRS*	73
5.1.5	Experimento Time-TSRS	76
5.1.6	Experimento regressão linear	78
5.2	Considerações	81
6	Conclusão	85
6.1	Contribuições	85
6.2	Trabalhos futuros	86
	${f Referências^1}$	87

De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.

1 Introdução

Sistemas de recomendação estão se tornando ferramentas comuns em sites e sua adoção se deve principalmente ao fato de que o volume e a complexidade da informação, assim como as diferentes atividades que as pessoas realizam diariamente, estão aumentando continuamente. A necessidade de se extrair valor dessa quantidade de dados fez com que este tipo de sistema surgisse (QUIJANO-SANCHEZ; RECIO-GARCIA; DIAZ-AGUDO, 2011).

O objetivo desse tipo de sistema é, utilizando diferentes técnicas, apresentar sugestões de itens relevantes para seus usuários. Estas sugestões estão comumente relacionadas a processos de tomada de decisão como: Qual série de televisão assistir? Qual livro comprar? Qual curso escolher para cursar? É neste contexto que atuam os sistemas de recomendação, ajudando os usuários a decidir entre os diversos itens disponíveis, por exemplo, do Netflix¹ (RICCI; ROKACH; SHAPIRA, 2011).

Prover recomendações personalizadas para usuários é uma tarefa que requer conhecimento sobre suas características. Os sistemas de recomendação, normalmente, mantêm informações sobre seus usuários, incluindo, por exemplo, suas atividades dentro de uma dada plataforma, coletando, assim, dados de maneira implícita. Além disso, há dados coletados explicitamente como, por exemplo, dados de avaliações sobre itens. Estes são importantes para esse tipo de sistema (JANNACH et al., 2010).

Em sistemas de recomendação, "item" é o termo utilizado para se referir àquilo que será sugerido para os usuários, esses sistemas normalmente são focados em calcular as recomendações para um tipo específico de item, como músicas, livros e notícias, para maximizar a efetividade dos itens recomendados.

Recomendações são usualmente apresentadas como uma lista de K itens, organizados de acordo com os critérios do algoritmo utilizado, que para ordenar os itens baseia-se nos dados coletados sobre os usuários e itens. Os dados a serem utilizados tipicamente passam por uma fase de avaliação, pois, no caso de produtos em um site de comércio eletrônico existem usuários que visitam o site e compram produtos apenas uma única vez, o que pode gerar ruídos e fazer com que o desempenho do sistema não seja ótimo (RICCI; ROKACH; SHAPIRA, 2011).

https://www.netflix.com/br/

O surgimento de redes sociais online, como o Facebook² e o Google+³ permitiu aos usuários inserir suas preferências pessoais em diferentes assuntos, além de se conectar diretamente e indiretamente a seus amigos (CHIN; XU; WANG, 2013). Esse comportamento espontâneo dos usuários bem como seu engajamento em inserir diferentes tipos de informações pessoais, forneceu novos dados que podem ser utilizados pelos sistemas de recomendação.

Uma maneira comum de utilizar estes dados é calcular a similaridade de um usuário com seus amigos da rede, utilizando seus amigos "mais similares" para sugerir os possíveis itens que o usuário possa se interessar (LEE; BRUSILOVSKY, 2010).

Neste mestrado foi construída uma rede social de pesquisadores, utilizando suas relações de coautoria e, a partir dessa rede, foi desenvolvido um sistema de recomendação de artigos. Estas recomendações visam a auxiliar pesquisadores na descoberta de artigos que sejam relevantes para suas áreas de estudo.

Este capítulo contém na seção 1.1 o detalhamento do objetivo deste projeto, a seção 1.2 descreve o método aplicado para a conclusão do objetivo e a seção 1.3 explica a organização do restante do texto.

1.1 Objetivo

A comunidade científica pode ser representada como uma rede social, na qual é possível encontrar relações de coautoria, colaboração em projetos, orientações, além de citações de trabalhos e, consequentemente, citações aos respectivos autores.

Algumas perguntas frequentemente realizadas por pesquisadores, como: Quais artigos interessantes da minha área eu ainda não tenho conhecimento?; e Quais artigos podem auxiliar em trabalhos que tenho em desenvolvimento?, não podem ser respondidas com uma consulta simples. Essas perguntas podem ser respondidas apenas dedicando-se tempo em encontrar fontes de dados relevantes (como bibliotecas digitais) e em encontrar estes trabalhos propriamente ditos (ZHOU; LUO; LIN, 2010).

Nesse contexto, um sistema de recomendação de trabalhos científicos poderá representar um facilitador para os pesquisadores. Este sistema pode ser construído a partir da rede social acadêmica descrita em Zhou, Luo e Lin (2010).

² https://www.facebook.com/

³ https://plus.google.com

Desta forma, o objetivo principal deste mestrado é propor um sistema de recomendação de artigos, que forneça dados significativos àqueles que o utilizarem, combinando informações sociais e informações bibliométricas, de forma a obter uma solução mais eficiente do que as correlatas.

Este projeto teve como premissa que a adição do contexto temporal para a recomendação de artigos, bem como a combinação de diferentes algoritmos de recomendação e, em particular, o uso de informações da rede social de coautoria dos pesquisadores possibilitaria uma melhor recomendação (segundo as métricas comumente usadas na área).

1.1.1 Objetivos específicos

Além do objetivo principal, este projeto também apresenta três objetivos específicos à ser alcançados. Estes objetivos são:

- Realizar um estudo sobre sistemas de recomendação que utilizam redes sociais, evidenciando os algoritmos utilizados e definir maneiras adequadas para avaliar os mesmos
- Comparar os algoritmos propostos por este projeto com algoritmos já consolidados na área de sistemas de recomendação e, também, com algoritmos encontradas em trabalhos correlatos
- Disponibilizar os dados filtrados e estruturados utilizados em uma plataforma online para possibilitar e facilitar trabalhos futuros

1.2 Método

Esta seção apresenta o método utilizado para a experimentação e os passos realizados no desenvolvimento deste trabalho.

1.2.1 Procedimento de experimentação e análise de resultados

Os algoritmos implementados foram executados a partir do conjunto de treinamento $(W_{treinamento})$ utilizando diferentes configurações de entrada para o tamanho da vizinhança, onde os vizinhos de um autor são os autores que atendem à uma determinada regra de

vizinhança, que pode ser definida, por exemplo, pela ordenação dos autores de acordo com o coeficiente de correlação com o autor original.

Esta variação da vizinhança, porém, não utilizou nenhum limite relacionado ao coeficiente de correlação ou de confiança gerado. Essas recomendações serão avaliadas sobre os dados do conjunto de treinamento (W_{teste}).

A avaliação dos resultados da recomendação foi feita com base na efetiva citação ou não do artigo pelo autor para o qual a recomendação foi gerada. Isto é, recomenda-se a um autor um conjunto de artigos que não constam na sua lista de artigos citados (conjunto de treinamento), após a recomendação verifica-se se um ou mais artigos recomendados foram efetivamente citados pelo autor considerando os dados do conjunto de testes.

A partir de um conjunto de vizinhos ordenado do primeiro "melhor vizinho" para o "pior vizinho", o tamanho do conjunto de entrada foi dividido selecionando-se:

- Os 5 melhores vizinhos
- Os 10 melhores vizinhos
- Os 15 melhores vizinhos
- Os 20 melhores vizinhos
- Os 25 melhores vizinhos
- Os 30 melhores vizinhos

Para cada uma das configurações de vizinhança e dos algoritmos executados, são coletados dados para avaliar a efetividade e a relevância dos primeiros elementos recomendados pela estratégia de recomendação. Estes dados são coletados variando-se a quantidade de recomendações relevantes consideradas, que é representada por um valor k. As quantidades de recomendações testadas para cada autor foram 5, 10, 15, 20 e 25.

Visto que se pretende avaliar as recomendações a cerca da qualidade dos primeiros itens ranqueados, as métricas utilizadas para proporcionar essa análise serão:

- S@K: é uma medida para verificar se foi realizado algum acerto dentro de um subconjunto K de recomendações;
- MRR: é uma medida para verificar em qual posição, em média, está ocorrendo o primeiro acerto dentro de K;
- Revocação (recall): é uma medida capaz de medir quantos itens do conjunto de testes W_{teste} foram encontrados dentro do conjunto de recomendações de todos os usuários;

• Precisão (precision): é uma medida que tem como base o Recall, porém ela adiciona em seu cálculo a quantidade de itens recomendados utilizados para obter determinada quantidade de acertos.

O detalhamento de cada uma das medidas citadas é feito na seção 2.4. Além disso essas métricas serão calculadas utilizando-se duas visões diferentes: utilizando como ponto de partida os autores presentes no conjunto de dados de teste W_{teste} e utilizando os autores presentes nos resultados das recomendações. A partir dos autores presentes no conjunto de dados de teste, pretende-se avaliar o desempenho geral do algoritmo e qualidade de suas respostas para todos os usuários considerados, enquanto que partindo dos usuários presentes na recomendação, planeja-se avaliar o desempenho do algoritmo direcionado para aqueles usuários em que foi possível gerar uma recomendação. Tipicamente, estes usuários são os que possuem mais informação disponível e, portanto, é possível comparar os resultados das recomendações de duas formas: considerando as recomendações geradas para todos os usuários do conjunto de testes ou analisando apenas os usuários para os quais uma ou mais recomendações foram geradas. No primeiro caso é possível ter uma visão geral do algoritmo, no segundo é possível analisar em mais detalhe a qualidade das recomendações efetivamente geradas.

1.3 Organização

O restante desta dissertação está organizado da seguinte forma: no capítulo 2 são apresentados os conceitos básicos necessários para o entendimento deste projeto, no capítulo 3 é apresentada a revisão da literatura correlata realizada para obter um melhor conhecimento sobre o tema, enumerando os passos realizados e as conclusões realizadas, no capítulo 4 é descrito em detalhes o conjunto de dados utilizado, os algoritmos de recomendação propostos por este trabalho e os outros algoritmos comparativos de recomendação implementadas. No capítulo 5 são apresentados os detalhes da execução de cada um dos experimentos evidenciando seus resultados. Por fim, o capítulo 6 apresenta as contribuições realizadas por este projeto e trabalhos futuros.

2 Conceitos fundamentais

Neste capítulo são apresentados os conceitos fundamentais para auxiliar na compreensão dos assuntos discutidos neste trabalho. A seção 2.1 aborda o tema de sistemas de recomendação que se utilizam de filtragem colaborativa, a seção 2.2 discute os sistemas que têm por base utilizar dados de redes sociais, destacando sistemas baseados em confiança, a seção 2.3 discute alguns dos problemas comuns em recomendações e, por fim, a seção 2.4 apresenta métricas utilizadas para avaliar a qualidade de uma seleção ordenada de itens.

2.1 Recomendações utilizando filtro colaborativo

O filtro colaborativo (*Collaborative Filtering*) é a técnica mais comumente encontrada em sistemas de recomendação. A ideia principal desta técnica é utilizar informações sobre o comportamento dos usuários, como itens comprados e avalições de itens, de maneira a calcular quais produtos os usuários têm maior probabilidade de estar interessados (RICCI; ROKACH; SHAPIRA, 2011).

As técnicas que implementam um Filtro Colaborativo puro tipicamente utilizam uma matriz de " $Usu\'{a}rios$ x Itens" contendo informações sobre as avaliações que os usu\'{a}rios inseriram sobre os itens para estimar o quanto um usu\'{a}rio pode se interessar em um item se baseando em usu\'{a}rios com comportamento similar. Após esse passo de cálculos, normalmente, é sugerida uma lista, contendo os primeiros K elementos que o usu\'{a}rio pode ter interesse (JANNACH et al., 2010).

Os métodos de Filtro Colaborativo são divididos em dois grupos: os baseados em vizinhança (Neighborhood-Based), que costumam se dividir em abordagens baseada em usuários ou itens; e os baseados em modelos (Model-Based) (RICCI; ROKACH; SHAPIRA, 2011).

Os algoritmos baseados em usuários (*User-Based*) levam em consideração as avaliações de usuários sobre itens, assim, a partir de um usuário localiza quais são seus vizinhos que têm gostos similares, ou seja, que tenham o vetor de avaliações similar. Para cada produto que o usuário ainda não tem avaliações registradas uma estimativa é calculada a partir dos usuários similares, supondo que se os usuários apresentaram gostos parecidos no passado, futuramente eles terão gostos parecidos.

Para calcular a similaridade são utilizadas técnicas para medir o coeficiente de correlação (usuários similares possuem correlação alta entre suas avaliações de itens) ou alguma medida de distância (usuários são considerados similares quando suas avaliações sobre itens possuem pequena distância) como: Pearson, Cosseno, Jaccard e Spearman. No caso de técnicas baseada em usuários é comum utilizar Pearson, que pode ser observada na equação 1, onde a e b são usuários pertencentes a um conjunto de usuários U e i é um produto pertencente a um conjunto I de produtos, $r_{u,i}$ é a avalição que o usuário u realizou sobre o item i e $\bar{r_u}$ é a avaliação média de um usuário u.

$$pearson(a,b) = \frac{\sum_{i \in I} (r_{a,i} - \bar{r_a})(r_{b,i} - \bar{r_b})}{\sqrt{\sum_{i \in I} (r_{a,i} - \bar{r_a})^2} \sqrt{\sum_{i \in I} (r_{b,i} - \bar{r_b})^2}}$$
(1)

Desta forma, o cálculo das estimativas é totalmente relacionado às similaridades dos usuários, como pode ser observado na equação 2, onde N é um conjunto com os primeiros N vizinhos selecionados para o usuário a e i é um item pertencente ao conjunto I de produtos (JANNACH et al., 2010).

$$pred_{user}(a,i) = \bar{r_a} + \frac{\sum_{b \in N} pearson(a,b) * (r_{b,i} - \bar{r_b})}{\sum_{b \in N} pearson(a,b)}$$
(2)

Devido à existência de bases de dados muito esparsa, algoritmos baseados em itens surgiram e, diferentemente dos baseados em usuários, calculam a similaridade entre os itens, ao invés de calcular a similaridade entre usuários. Assim, esta abordagem compara os vetores de avaliações dos itens, buscando encontrar aqueles que são seus similares. Esta técnica é comumente empregada utilizando uma modificação da similaridade de cosseno que pode ser observada na equação 3, onde U é o conjunto de usuários que avaliaram ambos os itens i e j.

$$cosine(i,j) = \frac{\sum_{u \in U} (r_{u,i} - \bar{r_u})(r_{u,j} - \bar{r_u})}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r_u})^2} \sqrt{\sum_{u \in U} (r_{u,j} - \bar{r_u})^2}}$$
(3)

Para predizer o possível interesse do usuário u no item i é calculado um valor utilizando as avaliações que o usuário u registrou para os primeiros K itens similares à i, pertencentes a um conjunto J. O cálculo é detalhado na equação 4 (JANNACH et al., 2010).

$$pred_{item}(u,i) = \frac{\sum_{j \in J} cosine(i,j) * r_{u,j}}{\sum_{j \in J} cosine(i,j)}$$
(4)

Os métodos baseados em modelo diferem dos baseados em vizinhança pois, ao invés de utilizar diretamente o relacionamento entre usuários e itens para gerar as predições, essa abordagem faz uma tentativa de criar um modelo preditivo usando as relações existentes para treinar um modelo a ser utilizado na predição dos itens de interesse.

O algoritmo de fatorização de matrizes ($Matrix\ Factorization$) é o mais utilizado dentro deste grupo. Seu funcionamento se dá pelo mapeamento de usuários e/ou itens em um espaço de dimensionalidade n, com as diferentes características presentes nos mesmos, modelando-os neste espaço n-dimensional.

A função deste espaço é de caracterizar itens e usuários sobre os fatores automaticamente inferidos a partir dos dados registrados pelo usuário. Por exemplo, quando os itens são músicas, o espaço pode criar dimensões tais como rock *versus* pop (RICCI; ROKACH; SHAPIRA, 2011). Este grupo de algoritmos tem mostrado resultados superiores quando comparados com os baseados em vizinhança.

2.2 Recomendações utilizando dados sociais

Estudos recentes evidenciaram que as pessoas possuem maior probabilidade em confiar mais em conteúdos que são originados a partir de pessoas que elas conhecem - como seus amigos - do que de desconhecidos - pessoas com comportamento similar em um sistema. Esse comportamento motivou a criação da técnica *Social Based Recommendation*, com o objetivo de prover recomendações com maior significância e utilidade para os usuários, utilizando o perfil dos usuários e as relações criadas entre eles, que podem ser encontradas naturalmente em sites de redes sociais como o *LinkedIn*¹ (RICCI; ROKACH; SHAPIRA, 2011).

A recomendação baseada em redes sociais é parecia com as técnicas de filtro colaborativo e a principal diferença entre elas está no uso de seus vizinhos mais próximos, uma vez que é atribuído um peso $w_{a,b}$ na similaridade entre pares levando em consideração o grau da relação (de amizade, por exemplo) existente entre eles. Este grau pode variar entre as diferentes possíveis relações entre pares, além da configuração de quantos níveis essas relações podem utilizar como, por exemplo, amigos de amigos.

A partir de um mapeamento onde a existência de uma relação direta de amizade entre dois usuários receba o peso 1 e a não existência dessa relação o peso 0, pode-se calcular a similaridade entre dois usuários como representado pela equação 5, onde a e b

¹ https://br.linkedin.com/

são usuários, I é o conjunto de itens avaliados por eles, $r_{u,i}$ é o rating de um usuário u em um item i e $\bar{r_u}$ é o rating médio de u.

$$sim(a,b) = w_{a,b} * \frac{\sum_{i \in I} (r_{a,i} - \bar{r_a})(r_{b,i} - \bar{r_b})}{\sqrt{\sum_{i \in I} (r_{a,i} - \bar{r_a})^2} \sqrt{\sum_{i \in I} (r_{b,i} - \bar{r_b})^2}}$$
 (5)

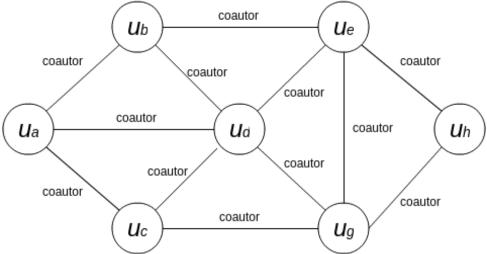
Dessa maneira é garantido que apenas usuários que possuem relações serão relevantes no cálculo das recomendações para qualquer usuário a.

Groh e Ehmig (2007) implementaram a técnica descrita na equação 5, utilizando apenas os amigos diretos dos usuários nas redes sociais *Orkut* e *Multiply* como possíveis vizinhos para o cálculo de recomendação utilizando a força da relação entre os usuários como a medida de similaridade, com a hipótese de que uma vizinhança baseada em redes sociais teria um desempenho superior a uma vizinhança baseada somente na similaridade entre pares. Esta hipótese foi confirmada ao final da experimentação.

2.2.1 Recomendação baseada em confiança

Técnicas baseadas em confiança geralmente se utilizam de uma estrutura de rede, que pode ser representada por um grafo G(V,A) no qual os vértices representam os usuários e as relações são representadas pelas arestas, como pode ser observado na figura 1. As regras de confiança aplicam um peso a essas relações criando um rank da importância das mesmas (RICCI; ROKACH; SHAPIRA, 2011).

Figura 1 – Representação de uma rede de confiança a partir de grafos.



Fonte: Arthur Patricio Grava, 2016

A confiança entre dois usuários, mesmo que essa confiança não tenha sido explicitamente demonstrada, pode seguir a fórmula geral descrita na equação 6, onde d_{max} é o grau da relação de maior valor na rede e $d_{u,v}$ o grau da relação existente entre um usuário u e um usuário v qualquer, obtendo a confiança entre estes usuário, representada por $\omega_{u,v}$ (GAO; XU; CAI, 2011a).

$$\omega_{u,v} = \frac{d_{max} - d_{u,v} + 1}{d_{max}} \tag{6}$$

A maneira mais simples de calcular recomendações a partir de uma rede de confiança é utilizando uma média ponderada, pela confiança entre usuários, das avaliações dadas a itens a partir da rede de confiança de um usuário qualquer. A equação 7 exemplifica esse cálculo, onde $r_{u,i}$ é a avaliação do usuário u no item i e A é o conjunto de usuários confiáveis na rede de um usuário u qualquer (GAO; XU; CAI, 2011a).

$$pred(u,i) = \frac{\sum_{a \in A} (\omega_{u,a} \times r_{a,i})}{\sum_{a \in A} \omega_{u,a}}$$
 (7)

Além de utilizar as regras de confiança diretamente, como exemplificado pela equação 7, o peso da confiança também pode ser utilizado como peso na similaridade entre dois usuários, fazendo uma mescla entre a técnica de filtro social com injeção de confiança.

Uma técnica chamada de $Trust\ Walker$, desenvolvida por Jamali e Ester (2009), é um exemplo no qual o uso da confiança é empregado. Nesta técnica foi usado o modelo de caminhada aleatória combinando as avaliações dos usuários aos itens e a relação de confiança entre os usuários. A probabilidade de se utilizar as avaliações de um usuário a para calcular a recomendação de um usuário u é calculada pela chance de o usuário u visitar determinado item enquanto segue dinamicamente as arestas entre usuários e itens definidas pelas avaliações e pelas relações de confiança entre os amigos (JAMALI; ESTER, 2009).

2.3 Preocupações em sistemas de recomendação

Existem alguns problemas em sistemas de recomendação que são recorrentes e devem ser levados em consideração ao desenvolver um novo sistema. Eles são: *cold-start users* (novos usuários do sistema), a serendipidade e a segurança.

2.3.1 Novos usuários do sistema e serendipidade

Cold-start users é um tipo de problema comum em métodos de Filtro Colaborativo, este evento acontece quando o sistema tem poucas informações disponíveis sobre um determinado usuário, o que acontece geralmente com novos usuários, uma vez que eles não interagiram suficientemente com o sistema, por exemplo, não registrando avaliações sobre os itens e não oferendo mais informações. Sendo assim, essa falta de informações faz com que não seja possível gerar recomendações relevantes para este tipo de usuário, pois qualquer identificação de usuários que apresentam um comportamento similar ao seu será muito superficial e provavelmente pouco relevante.

A serendipidade acontece quando o sistema não é capaz de apresentar recomendações inesperadas que possam ser do interesse do usuário, ou seja, quando o sistema apresenta itens que são muito similares àqueles que o usuário já demonstrou interesse, fazendo com que as recomendações apresentadas gerem pouco valor para o usuário, pois, mesmo que o usuário se identifique com a recomendação, ela não é algo que o surpreendeu e o fez demonstrar um interesse imediato (JANNACH et al., 2010).

Sistemas que se baseiam em uma similaridade simples entre produtos ou sistemas que somente se baseiam no comportamento passado do usuário estão passíveis de sofrer o problema da serendipidade e podem nunca surpreender seus usuários e fidelizá-los a partir deste sentimento. Podendo, então, não ajudar realmente seus usuários a encontrar novos itens já que o sistema só fará sugestões muito parecidas com os itens já avaliados (ou comprados) pelos usuários.

Para resolver esses problemas normalmente costuma-se combinar técnicas que funcionam bem nos diferentes contextos, usando abordagens que têm uma conexão maior com a similaridade entre os itens para usuários novos e técnicas mais robustas para o caso em que se têm mais informações sobre os usuários do sistema.

2.3.2 Segurança

O problema da segurança possui dois aspectos principais. O primeiro é garantir que as diferentes informações passadas pelo usuário, incluindo sua interação com o sistema, não possam ser copiadas indevidamente por terceiros. Já o segundo aspecto é não permitir que o sistema de recomendação seja indevidamente influenciado por ações de terceiros,

incluindo desde ações de *hackers* que tentem modificar o perfil dos usuários até ações de *bot* que tentem, por exemplo, superestimar um item (é comum em sistemas de busca ou recomendação por websites de interesse haver a preocupação em tentar evitar que *bots* superestimem o prestígio de um website por meio de muitas visitas artificiais a estes websites).

O problema da segurança está fora do escopo desta dissertação.

2.4 Métricas

Esta seção apresenta técnicas para avaliar a qualidade da sequência de um conjunto de itens ordenados por alguma pontuação estabelecida.

2.4.1 Sucesso no conjunto

A medida Sucesso no conjunto ($Success\ at\ rank$ - S@K) avalia a proporção da quantidade de vezes em que foi possível encontrar ao menos um item relevante nos primeiros k recomendados pelo algoritmo, sendo assim, uma medida que avalia a qualidade dos k primeiros itens previstos (HARVEY; RUTHVEN; CARMAN, 2010).

A partir disso o S@K pode ser medido utilizando-se a equação 8, onde N é o número total de usuários no conjunto de dados de teste, pos_n é a posição que o primeiro item relevante para o usuário n foi encontrado em seu conjunto de recomendações e k_{max} é a posição máxima onde esse item deveria ter sido encontrado. A função hit define se o item relevante foi encontrado ou não, de acordo com a equação 9, utilizando como entrada o primeiro item relevante encontrado (pos_n) e a posição máxima (k_{max}) onde este item pode ser encontrado.

$$S@K = \frac{1}{N} \sum_{n=1}^{N} hit(pos_n, k_{max})$$
(8)

$$hit(x,y) = \begin{cases} 1, & \text{se } x \le y \\ 0, & \text{caso contrário} \end{cases}$$
 (9)

Neste trabalho os resultados de S@K serão apresentados em porcentagens.

2.4.2 Erro recíproco médio

O Erro recíproco médio ($Mean\ reciprocal\ rank$ - MRR) é uma medida para avaliar a qualidade do conjunto de saída de um algoritmo, checando em qual posição os elementos relevantes para o usuário foram encontrados, partindo do pressuposto que o usuário está interessado no primeiro item relevante que for encontrado. A partir disso, a medida para um conjunto único é determinada pela fórmula $1/pos_i$, onde pos_i é a posição que o primeiro elemento relevante foi encontrado (CRASWELL, 2009).

Deve-se salientar que o MRR é uma medida que cria uma penalidade alta para as mudanças nas primeiras posições, por exemplo, se ao realizar uma modificação no seu algoritmo e um item que era encontrado na posição 1 passa a ser encontrado na posição 2, isso irá gerar uma mudança no MRR de 0, 5, saindo de 1,0 para 0, 5. Porém se sua mudança leva um item que era encontrado na posição 100 para a posição 1000, a diferença gerada será de apenas 0,009, variando de 0,01 para 0,001.

Assim sendo, considerando-se o contexto de recomendações, pode-se calcular o MRR para um conjunto de usuários U como descrito na equação 10, onde $pos_{u,i}$ representa a posição do primeiro elemento relevante i encontrado dentro do conjunto de recomendações de um usuário u qualquer, como pode ser observado na equação 10. O valor do MRR varia no intervalo [0,1] e, ao utilizá-lo, o valor que estiver mais próximo de 1 pode ser considerado como o mais satisfatório.

$$MRR = \frac{1}{|U|} \sum_{u \in U} \frac{1}{pos_{u,i}} \tag{10}$$

A medida MRR pode ser aplicada considerando que todos os itens possíveis (todos os itens do conjunto de dados ainda não comprados/lidos/utilizados pelo usuário) serão recomendados ou considerando que apenas um máximo de K itens será recomendado. No presente trabalho, as medidas de MRR foram realizadas para diferentes valores de K.

2.4.3 Precisão e revocação

A revocação é uma medida que avalia quantos dos itens relevantes foram encontrados em um conjunto de dados previsto, ou seja, dado um conjunto de testes com I elementos relevantes para um usuário u, e um conjunto V de itens recomendados e/ou previstos para

este usuário, a revocação para este usuário individualmente é a quantidade de acertos (#hits) - que é a quantidade de itens em I encontrados em V, do conjunto V sobre a quantidade total de itens em I, exemplificado pela equação 11 (CREMONESI; KOREN; TURRIN, 2010).

$$recall = \frac{\#hits}{|I|} \tag{11}$$

Para estender a equação 11 para uma formula generalizada deve-se utilizar todos os itens relevantes dos usuários no conjunto de testes T e confrontá-los com as recomendações de cada um dos usuários existentes em T, assim, sendo $\#hits_t$ a quantidade de acertos de T, a revocação do conjunto é calculado pela equação 12.

$$recall_T = \frac{\#hits_t}{|T|} \tag{12}$$

A precisão tem um cálculo similar à revocação, porém ela insere uma penalidade utilizando a quantidade de itens recomendados e/ou previstos que foram utilizados para determinar sua eficiência, sendo assim, ela pode ser representado pela equação 13, onde $\#hits_t$ é a quantidade de acertos de T e |N| é o tamanho máximo atribuído a um conjunto de recomendações dos usuários.

$$precision(N) = \frac{\#hits_t}{|T| * |N|} = \frac{recall_T(N)}{|N|}$$
(13)

2.4.4 Erro absoluto médio e raiz do erro quadrático médio

O erro absoluto médio (*Mean absolute error* - MAE) e a raiz do erro quadrático médio (*Root mean squared error* - RMSE) são medidas utilizadas para avaliar o desvio médio entre uma classificação prevista e a real classificação de um usuário, mostrando o quão próxima as previsões do sistema se encontram das classificações reais dos usuários, sendo medidas importantes para sistemas em que os valores previstos serão exibidos para o usuário (HERLOCKER et al., 2004).

O cálculo do MAE é comumente empregado como descrito pela equação 14, onde $r_{u,i}$ é a classificação real do usuário u no item i, $\hat{r}_{u,i}$ é a classificação prevista pelo sistema

e P é o conjunto contendo os pares (u,i) previstos. Já o cálculo do RMSE é descrito pela equação 15, utilizando as mesmas variáveis da equação 14.

$$MAE = \frac{\sum_{(u,i)\in P} |r_{u,i} - \hat{r}_{u,i}|}{|P|}$$
 (14)

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in P} (r_{u,i} - \hat{r}_{u,i})^2}{|P|}}$$
 (15)

A principal diferença entre as duas medidas é que o RMSE atribui uma ênfase maior para erros maiores, ou seja, enquanto o erro de um ponto aumenta o acumulado de erro em um ponto, um erro de dois pontos acumula quatro pontos na soma, assim, atribuindo um peso maior para erros maiores. Já o MAE utiliza os erros de maneira uniforme.

3 Revisão da literatura correlata

A fim de entender e identificar os principais conceitos e componentes da área de sistemas de recomendação que utilizam dados de redes sociais foi realizada uma revisão sistemática, cuja metodologia utilizada é apresentada na seção 3.1, em seguida os resultados são mostrados na seção 3.3 e as conclusões encontradas na seção 3.4.

Após o detalhamento da revisão e os resultados obtidos, a seção 3.5 irá explanar artigos correlatos que estão contidos na área de sistemas de recomendação de artigos científicos.

3.1 Planejamento

Para a realização da revisão, primeiramente foi realizado um estudo exploratório sobre a área de Sistemas de Recomendação para obter conhecimento sobre o assunto, a fim de elaborar uma a revisão sistemática sobre o tema (KITCHENHAM, 2004).

A partir desse estudo inicial, a seguinte pergunta foi definida: "Quais são as principais técnicas existentes para recomendação de conteúdo que se utilizam de redes sociais?" Para responder a esta pergunta a revisão foi realizada nas seguintes etapas: (i) Busca nas bibliotecas digitais, (ii) Condução e seleção de artigos e (iii) Extração dos dados. Cada uma das etapas é detalhada a seguir.

3.2 Condução

Foram utilizadas duas bibliotecas digitais para a realização da revisão: ACM Digital Library (ACM) e IEEExplore Digital Library (IEEE), por estas serem as bibliotecas que possuem um número relevante de artigos publicados nos principais eventos e periódicos científicos sobre sistemas de recomendação. Para cada uma das bases foi definida uma string de busca, uma vez que em cada base estas strings são processadas de maneiras diferentes.

As *strings* utilizadas foram:

• ACM: ((Abstract: "recommender system" or abstract: "recommendation system" or Abstract: "recommender systems" or Abstract: "recommendation systems") and (Abstract: "social network" or Abstract: "social networks"))

• IEEE: ("Abstract": "recommender system" OR "Abstract": "recommendation system" OR "Abstract": "recommender systems" OR "Abstract": "recommendation systems") AND ("Abstract": "social network" OR "Abstract": "social networks").

Com as *strings* de busca definidas, foi necessário estabelecer e aplicar os critérios de seleção nos artigos encontrados. Um artigo é selecionado somente se ele atender a todos os critério de inclusão.

Os critérios definidos são:

• Inclusão:

- Trabalhos completos publicados e disponíveis digitalmente na íntegra nas bibliotecas digitais utilizadas;
- Trabalhos que abordassem a construção de um sistema de recomendação de conteúdo utilizando-se das relações existentes entre pessoas em redes sociais;
- Trabalhos que possuíam métricas de avaliação para o sistema construído;

3.2.1 Seleção dos dados

A busca na biblioteca IEEE retornou 258 artigos, na data 31/08/2015 e destes, ao serem aplicados os critérios de seleção, 24 artigos foram selecionados. A busca na base da ACM retornou 191 artigos, na data 31/08/2015, dos quais, realizando o mesmo processo feito na IEEE, 22 artigos foram selecionados. Assim, ao todo foram selecionados 46 artigos.

3.2.2 Extração

Os 46 artigos selecionados foram lidos integralmente, extraindo-se informações a cerca da estratégia de recomendação (ER), das técnicas de avaliação (TA), das técnicas utilizadas para comparação da solução proposta (SC) e da base de dados (BD) utilizada para aplicar a solução proposta pelos autores.

3.3 Resultados

As informações coletadas nos artigos foram sumarizadas e podem ser observadas nas tabelas 1 e 2.

Tabela 1 – Sumário das Informações Extraídas na Revisão: Técnicas implementadas e dados utilizados.

Artigo	ER	BD	TA	SC
(GROH; EHMIG, 2007)	SF	Lokalisten	F-Measure, MAE	CF
(WOERNDL; GROH,	CxF	Lokalisten	F-Measure e	CF
2007)			MAE	
(GEYER et al., 2008)	SF	IBM-Social	Hit rate	Content-based
(KONSTAS; STATHOPOU-	RWR	Last.fm	MAE	CF
LOS; JOSE, 2009)				
(JAMALI; ESTER, 2010)	TSRS	Flixster, Epinions	RMSE	CF, BaseMF e STE
(MA et al., 2011)	SF	Flixster, Epinions	MAE e RMSE	UserMean, ItemMean,
		, 1		CF
(SCELLATO; NOULAS;	LF	Gowalla	AUC	Random Forests,
MASCOLO, 2011)				Naive Bayes
(YANG; STECK; LIU,	CIF	Epinions	MAE e RMSE	BaseMF e SocialMF
2012)				
(XIE et al., 2012)	SF	Facebook	NDCG-IA	BaseR e FeatureR
(MEYFFRET; MÉDINI;	SF	Epinions	RMSE e F-	GlobalCF, Ran-
LAFOREST, 2012)			Measure	domWalk e TrustWal-
·				ker
(DWIVEDI; BHARADWAJ,	SF	Survey	MAE e F-	CF
2012)			Measure	
(LIU; HE; ZHAO, 2013)	TmSF	Filmtipset	MAP, AUC,	SVD++, BPR, IFMF
			Pre@K e AP	
(GRETARSSON et al.,	SF	Facebook	MAE e Curva	-
2010)			ROC	
(CHIN; XU; WANG, 2013)	CxF	Facebook	Hit rate	Common Friends Al-
				gorithm
(LIU; ABERER, 2013)	CxF	Douban, MovieLens	MAE e RMSE	RPFM, SoReg, BMF,
				CF
(LEVANDOSKI et al.,	LF	MovieLens, Fours-	Hit rate	CF
2012)		quare		- 1 - 0
(SARWAT et al., 2013)	LF	MovieLens, Fours-	Hit rate	LARS
(220222 + 0 2010)	DILL	quare	D ' ' D ''	OD OD
(NOULAS et al., 2012)	RWR	Foursquare,	Precision, Recall e	CF e SF
(*************************************	DID	Gowalla	APR	OD D 1
(YANG; GUO; LIU, 2013)	BIB	Epinions	MAE	CF e Trust-based
(CHIU; LIN; CHEN, 2011)	SF	MovieLens	F-Measure	Random Choose
(JIN; CHEN, 2012)	SF	Last.fm	RMSE e Recall	I-CF e U-CF
(GAO; XU; CAI, 2011b)	TSRS	Epinions	MAE	CF, Trust-Based e Hybrid
(NARUCHITPARAMES;	PSF	Facebook	ARR e SD	SRS e Network-Based
GUNES; LOUIS, 2011)				RS
(SIM et al., 2012a)	CxF	-	MAE, Precision e	Rough Set-Based,
			Recall	Multidimensional
(WANG; DENG, 2011)	BF	Last.fm	Hit Ratio e Ave-	CF, Time-Weighted
			rage Age	CF
(SILVA et al., 2010)	GBA	Oro-Aro	Correct Predic-	Friend of Friends
			tion	
(FEITOSA et al., 2013)	LSF	Foursquare	Score do TF-IDF	-

Fonte: Arthur Patricio Grava, 2016

Dos 46 artigos incluídos, 85% deles foram publicados a partir de 2011, enquanto apenas 15% deles foram publicados entre 2007 e 2010. A figura 2 ilustra a distribuição dos artigos que foram selecionados de acordo com seus anos de publicação.

Tabela 2 – Sumário das Informações Extraídas na Revisão - continuação

Artigo	ER	BD	TA	SC
(WANG; LI; ZHAO, 2013)	GF	Universidade	RMSE	FMS e AHP
(SHANGGUAN et al.,	JML	Douban	AUC	Rate Matrix, Social
2012)				Matrix
(SOPCHOKE; KIJSIRI-	SF	Facebook	MAP e MPR	NMF, wALS
KUL, 2011)				
(WANG; TERROVITIS;	LBSR	Gowalla, Brightkite	Precision, Recall,	CF, LARS, Location
MAMOULIS, 2013)			Utiliy, Cov	NN, SF, RWR.
(OU, 2014)	LBSR	Foursquare	Precision, Recall	LARS
(FORSATI et al., 2014)	MF-TD	Epinions	MAE, RMSE	MF, MF-T, MF-D
(WANG et al., 2014)	Social	Epinions	MAE	CF, SF, SocialMF,
	PMF			Trust Based
(WANG et al., 2014)	SRFRM	Epinions, Douban	MAE, RMSE	PMF, SF, SoReg
(FERREIRA; SILVA; LOU-	SocialMF	Twitter, Flicker	AUC	Joint SVD, Common
REIRO, 2014)				Neighbors
(HUANG et al., 2015)	CF +	Movielens, Lastfm	MAP, F1-	CF, PureSVD, NMF,
	SF +		Measure, NDCG	SLIM
	KB			
(AKBARI; TAJFAR; NE-	Artificial	Youtube	Accuracy, Hit	KNN, MLP, SVM,
JAD, 2013)	Bee Co-		rate	FOF and GA
	lony			
(CHRISTENSEN; SCHIAF-	SocialMF		NMAE, NRMSE	MF+SC, $MF+SS$,
FINO, 2013)				MF+TR
(CHEN et al., 2013)	SF	Epinions	MAE	CF + TrustB
(OZSOY; POLAT;	LBSR	Foursquare	Precision, Cove-	CF, SF, LARS
ALHAJJ, 2014)			rage, Hit rate	
(KHALID et al., 2014)	SF+Ant	Foursquare	Precision, Recall,	SF, SVD, RWR, Po-
	colony		F-Measure	pularity
	algo-			
	rithms			
(QIAN et al., 2014)	SocialMF	1 /	MAE, RMSE	MF, ContextMF, Cir-
		Douban		cleB
(JIANG et al., 2014a)	Social	Renren, Tencent	MAE, RMSE	ContentB, CF,
	PMF	Weibo		TrustB, SF, SoReg,
				MF
(HU; ESTER, 2014)	Social	Foursquare, Yelp	Recall	PMF, Popularity,
	topic			PMF+SoReg
(LIU et al., 2015)	TSRS	Epinions, Ciaos	MAE, RMSE	MF, MF+SF, Circle

Fonte: Arthur Patricio Grava, 2016

A figura 3 ilustra a distribuição anual de todas as técnicas encontradas nos artigos analisados, mostrando que a técnica mais utilizada foi o filtro social, que foi empregada em 11 artigos. É importante notar que em 2013 as técnicas híbridas (misturando o filtro social puro com outros algoritmos, como os algoritmos baseados em localização) tiveram destaque sendo a estratégia de recomendação mais empregada. A marcação "Outros" foi utilizada para denotar técnicas que foram utilizadas em apenas um dos artigos revisados.

A figura 4 ilustra quais foram as técnicas mais utilizadas para validar e metrificar os modelos desenvolvidos, além de mostrar quantas vezes essas técnicas foram utilizadas,

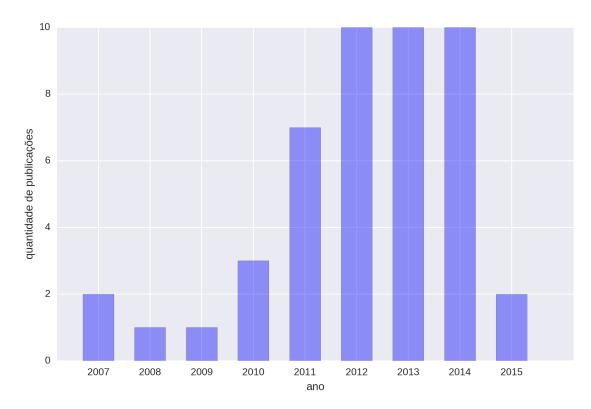


Figura 2 – Distribuição anual dos artigos publicados.

Fonte: Arthur Patricio Grava, 2016

lembrando que podem existir sobreposições, uma vez que um artigo pode utilizar de mais de uma técnica para avaliar seus resultados.

Como exposto na figura 4, as métricas mais utilizadas foram: MAE, RMSE e Precision / Recall. Além das técnicas expostas, outras técnicas também foram utilizadas, porém como elas não apareceram mais de duas vezes no conjunto elas não foram incluídas no gráfico.

3.3.1 Epinions

A base de dados com maior utilização foi a base do *Epinions*¹, presente em aproximadamente 25% dos trabalhos selecionados, na qual usuários podem submeter revisões sobre produtos, incluindo uma avaliação entre 0 e 5 estrelas, para qualquer produto.

Para aumentar o número de revisões, os revisores são pagos toda vez que suas revisões são lidas por algum outro usuário. Além disso, os usuários podem interagir

¹ http://www.epinions.com

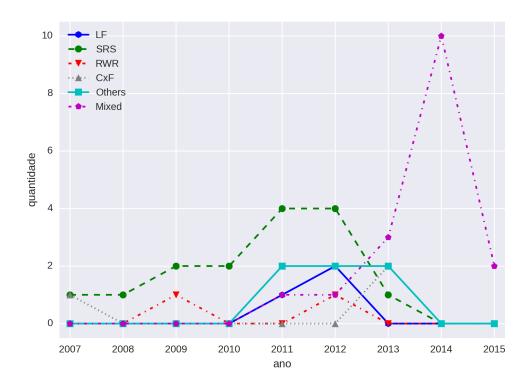


Figura 3 – Técnicas empregadas durante os anos.

entre si avaliando o quão útil aquela revisão foi para ele e listando os revisores que ele confia (RICHARDSON; DOMINGOS, 2002).

A tabela 3 mostra todos os artigos que utilizaram o *Epinions*, enfatizando o desempenho de cada um deles, destacando seus resultados pelas métricas *MAE* e *RMSE*, que foram as mais utilizados por eles. Em seguida as técnicas utilizadas por esses artigos têm uma breve explicação.

3.3.1.1 Redes de confiança

Gao, Xu e Cai (2011b) desenvolveram um sistema de recomendação híbrido que combinava informações de confiança adquiridas por meio da rede social, construindo um recomendador baseado em confiança, com um filtro colaborativo tradicional. Tendo aplicado a medida MAE, essa solução híbrida teve um resultado superior às técnicas tradicionais não combinadas;

Yang, Guo e Liu (2013) implementaram uma rede Bayesiana de inferência trocando recomendações entre amigos diretos e construindo a distribuição probabilística -

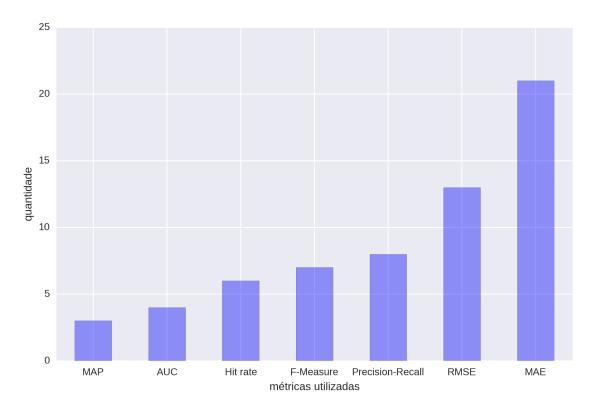


Figura 4 – Métricas utilizadas para comparar as abordagens desenvolvidas.

necessária em uma rede Bayesiana. Com essa informação o sistema estima a distribuição das recomendações dos usuários e a distribuição condicional entre usuários e seus amigos. Essa solução foi comparada com um filtro colaborativo tradicional e obteve resultados superiores;

Tabela 3 – Métricas da base de dados Epinions.

Epinions			
Artigo	Ano	MAE	RMSE
(JAMALI; ESTER, 2010)	2010	-	1,159
(MA et al., 2011)	2011	0,8256	1,0739
(GAO; XU; CAI, 2011b)	2011	0,743	-
(YANG; STECK; LIU, 2012)	2012	0,798	0,979
(MEYFFRET; MÉDINI; LAFOREST, 2012)	2012	-	1,078
(YANG; GUO; LIU, 2013)	2013	0,791	-
(CHEN et al., 2013)	2013	0,4376	-
(FORSATI et al., 2014)	2014	0,8025	1,0851
(WANG et al., 2014)	2014	0,8077	-
(WANG et al., 2014)	2014	0,2330	$0,\!3365$
(LIU et al., 2015)	2015	0,8992	1,1108

Fonte: Arthur Patricio Grava, 2016

Meyffret, Médini e Laforest (2012) construíram um sistema de recomendação social adicionando pesos entre as relações de amizade calculando a confiança entre eles. Para isso utilizaram correlação de *Pearson* e uma pontuação de confiança que leva em consideração a distância entre os usuários usando a distância entre eles no grafo de amizades. Essa solução foi comparada com um filtro colaborativo tradicional, recomendador *random walk* e um recomendador *trust walker*. A partir dos resultados coletados utilizando *RMSE*, pôde-se verificar que a solução proposta teve resultados superiores, principalmente para os novos usuários.

3.3.1.2 Fatoração de matrizes com confiança

Jamali e Ester (2010) desenvolveram um método de fatoração de matrizes utilizando características latentes de confiança - de usuários com conexão direta na rede, com caraterísticas latentes do usuário para estimar o vetor \vec{u} . Essa abordagem obteve resultados melhores quando comparada com filtro colaborativo tradicional, sistemas de recomendação social e outras técnicas que utilizam fatoração de matrizes com informação social;

Ma et al. (2011) implementaram dois métodos de fatoração de matrizes utilizando um termo social regulador. O primeiro método minimiza as preferências compartilhadas entre um usuário u e seus amigos confiáveis assumindo que essas preferências são próximas à média de confiança entre seus amigos, dando um peso maior àqueles que são os amigos mais similares. A segunda abordagem trata os usuários individualmente minimizando a distância entre os vetores de características de u e seus amigos. De acordo com os resultados compartilhados, ambos métodos tiveram uma eficiência superior ao filtro colaborativo tradicional, a técnica de fatoração de matrizes e a recomendadores baseados em confiança;

Yang, Steck e Liu (2012) desenvolveram um método de fatoração de matrizes utilizando a rede de confiança presente no conjunto de dados, mas, ao invés de usar o mesmo grupo de amigos confiáveis para todos os itens, decidiu-se encontrar os amigos confiáveis para cada uma das categorias dos itens. A ideia principal é que um usuário pode confiar em cada amigo apenas em certas categorias e não em outras, portanto o círculo de amigos pensando no estilo musical *rock* pode ser diferente do *pop*. Essa solução foi comparada com uma técnica tradicional de fatoração de matrizes e fatoração de matrizes utilizando características sociais, obtendo resultados melhores;

Chen et al. (2013) construíram uma fatoração de matrizes reduzindo o ruído em relações sociais de confiança verificando a quantidade de usuários confiáveis por um usuário local, e se estes usuários confiáveis possuem muitos usuários que explicitamente ou implicitamente lhes demostraram confiança utilizando a rede social de confiança. Com isso foi possível obter melhores resultados quando comparado com uma fatoração de matrizes probabilística, fatoração de matrizes com características sociais e o *SoRec* (MA et al., 2011);

A abordagem de Forsati et al. (2014) tem uma diferença principal dos trabalhos relacionados. Ao invés de somente utilizar a informação de confiança como característica latente, também foi utilizada a informação contrária (desconfiança), porém a desconfiança não é propagada pela rede como ocorre com a informação de confiança. Os experimentos realizados mostraram que adicionar a informação de desconfiança ajuda a encontrar recomendações mais assertivas para os usuários, quando comparada com a fatoração de matrizes, fatoração de matrizes com informações de confiança e fatoração de matrizes com informação de desconfiança;

Wang et al. (2014) implementaram uma fatoração de matrizes probabilística utilizando informação de confiança na rede social utilizando a correlação de *Pearson* para dar peso a essas relações entre amigos e uma influência pela similaridade de itens. São atribuídos pesos a essas relações para obter um modelo realista. Essa solução foi comparada com um filtro colaborativo, fatoração de matrizes probabilística e fatoração de matrizes com caraterísticas sociais obtendo resultados superiores;

Wang et al. (2014) desenvolveram uma fatoração de matrizes com caraterísticas sociais incorporando uma transformação linear da matriz *Usuário X Item*, fazendo uma combinação linear dos interesses do usuário, mapeando isso para combinar com as caraterísticas dos itens com uma transformação linear entre *Confiador X Confiado* para atender a influência do *Confiado*. Essa abordagem obteve resultados superiores à fatoração de matrizes probabilística, fatoração de matrizes com características sociais e fatoração de matrizes com características sociais utilizando regularização social;

Liu et al. (2015) construíram uma fatoração de matrizes com características sociais incorporando informações de confiança, mas, ao invés de trazer a rede de confiança de maneira genérica, foi calculada uma rede de confiança para cada domínio, já que um usuário u pode confiar em diferentes amigos para os diferentes domínios existentes. Assim, em cada domínio o usuário pode ter diferentes usuários nos quais ele confia. Com essa modificação, a técnica construída foi capaz de melhorar os resultados de uma fatoração de

matrizes tradicional, de uma fatoração de matrizes com características sociais e também a técnica *CircleCon* (YANG; STECK; LIU, 2012).

3.4 Considerações

Como mostrado na tabela 3, o melhor resultado obtido no conjunto de dados Epinions, em termos de MAE, foi 0,2330 sendo que a abordagem utilizou diferentes dimensões para realizar o cálculo de recomendação, misturando informações sociais com propagação de confiança, possibilitando um resultado melhor do que somente utilizar a informação de confiança, que foi responsável por obter o resultado menos expressivo (0,8256).

Analisando o mesmo contexto considerando o *RMSE*, pode-se observar que o segundo melhor resultado foi obtido por uma técnica que combinava círculos de confiança com a rede social em um método de fatoração de matrizes, com um resultado de 0,9790, enquanto que o pior resultado, utilizando apenas uma rede de confiança obteve 1,1590.

Na figura 3, considerando todo o intervalo de tempo, é possível observar que utilizar apenas dados da rede social entre os usuários foi a solução mais aplicada para calcular recomendações, porém, considerando os ano mais recentes, é possível identificar que o número de soluções utilizando soluções híbridas e que combinam informações estão crescendo, ano após ano.

A partir da figura 4, pode-se assumir que as métricas mais utilizadas para medir os resultados das recomendações que possuem informações de redes sociais são o MAE, RMSE e Precisão / Revocação.

Pesquisadores estão lidando com os problemas da serendipidade e de novos usuários combinando diferentes técnicas, por exemplo adicionando os dados de confiança calculados a partir de redes sociais, assim tirando vantagem dos benefícios que cada uma delas oferece. Por outro lado, alguns pesquisadores modificaram o filtro colaborativo adicionando conceitos de aprendizado de máquina e inserindo esses conceitos em diferentes técnicas.

A partir destes resultados, pode-se observar que sistemas de recomendação que combinam a informação de redes sociais com outras dimensões são aqueles que possuem melhores resultados quando comparados a sistemas que utilizam apenas uma dimensão. Portanto, ao desenvolver sistemas desse tipo, é necessário considerar e avaliar a possibilidade

de combinar dimensões e como combiná-las para tentar obter a combinação que gere o modelo mais efetivo.

Porém, também é necessário ter em mente a complexidade do modelo desenvolvido, pois adicionar novas dimensões pode causar problemas de capacidade de execução (capacidade computacional e tempo requerido) para calcular as recomendações. Dependendo do quão rápido seus resultados são necessários, as recomendações podem não mais fazer sentido. Devido à isso pode ser necessário considerar soluções menos complexas.

3.5 Outras pesquisas

Além dos trabalhos encontrados na revisão sistemática, destacam-se outros dois trabalhos que apresentam significativa relação com o presente mestrado.

Fu, Chang e Lee (2014) desenvolveram uma técnica de recomendação na qual o foco do sistema estava em realizar recomendações de possíveis eventos para os quais autores pudessem submeter seus artigos, para isso utilizaram dados sociais e informações da relação dos autores com artigos publicados.

O conjunto de dados de entrada foi extraído da *Digital Bibliography & Library Project* (DBLP) ² utilizando artigos do tipo *in-proceedings*, cujo ano de publicação variou de 1959 até 2011. Neste conjunto foram removidos autores com menos de 10 publicações entre 2004 e 2014, caracterizando a presença de 30.758 autores e 4.773 conferências.

A técnica implementada utilizou o método de caminho aleatório (Random Walk), calculando a proximidade entre dois nós na rede social de autores e itens, criada a partir dos dados sobre as publicações e contendo informações das relações de coautoria e as relações de autores com seus artigos publicados.

O sistema desenvolvido foi avaliado utilizando *RMSE*, precisão e revocação, utilizando como conjunto de testes publicações entre os anos de 2003 e 2007, entre 2005 e 2007 e o ano de 2007. Os resultados não foram comparados com outras técnicas, porém o valor de revocação mais expressivo encontrado foi de, aproximadamente, 0, 28, indicando que em 28% dos casos foi possível predizer um evento para o qual o autor efetivamente publicou ao menos um de seus artigos.

Rohani, Kasirun e Ratnavelu (2014) implementaram uma versão modificada de algoritmos baseados em conteúdo (*Content Based*) inserindo informações de redes sociais

² http://dblp.uni-trier.de/db/

acadêmicas, utilizando os amigos e companheiros de faculdade dos usuários, enquanto que a implementação clássica do algoritmo utilizava apenas as preferências dos usuários. O objetivo do sistema desenvolvido era de melhorar os resultados correlatos para a tarefa de recomendação de itens acadêmicos, no caso, notícias acadêmicas.

Para atingir esse objetivo foi desenvolvido um ambiente em que os usuários podiam cadastrar as relações indicadas anteriormente, chamado de *MyExpert*, o qual possuía 920 membros de 10 universidades da Malásia, onde, durante a experimentação, cerca de 100 itens novos eram introduzidos por semana e os diferentes algoritmos criavam uma seleção com os 10 melhores itens para cada um dos usuários. O experimento foi executado durante 14 semanas e recomendou um total de 1.398 itens.

A implementação desenvolvida foi comparada com a solução clássica baseada em conteúdo, com uma seleção aleatória dos itens acadêmicos e com filtro colaborativo durante o período de 14 semanas coletando o parecer dos usuários diretamente, por meio de sua interação com itens na plataforma *MyExpert*, e medindo os resultados utilizando as métricas precisão, revocação, *F1-Measure* e *Fallout*.

Os resultados obtidos pela experimentação mostraram que a técnica desenvolvida foi superior às demais, mostrando uma melhoria, em termos de precisão de 38% comparada a seleção aleatória, 17% ao filtro colaborativo e 14% ao algoritmo clássico baseado em conteúdo, confirmando a hipótese inicial de que os dados sociais poderiam trazer melhorias significativas às técnicas comparadas.

4 Solução proposta

Este capítulo descreve o conjunto de dados utilizados, a técnica de recomendação proposta, as atividades relacionadas aos desenvolvimentos de técnicas de recomendação e as atividades realizadas para a obtenção dos resultados deste trabalho.

Este capítulo é organizado da seguinte maneira: a descrição do conjunto de dados e as informações coletadas a partir dele é detalhada na seção 4.1, a seção 4.2 descreve a implementação da técnica de similaridade entre autores, a seção 4.3 discute as técnicas comparativas implementadas e a técnica desenvolvida e implementada é explanada na seção 4.4.

4.1 Conjunto de dados

Nesta seção é descrito o conjunto de dados utilizado no desenvolvimento deste trabalho.

4.1.1 Web of science

Para a realização deste trabalho se utilizou um conjunto de dados advindo da biblioteca científica digital Web of Science¹ (WoS), coletados por Mugnaini, Digiampietri e Mena-Chalco (2014), contendo informações sobre sobre publicações de trabalhos que tenham a participação de, pelo menos, um autor brasileiro.

Os dados básicos utilizados são registros de publicações, os quais contém, entre outras informações, o título do artigo, a lista de autores, o ano de publicação, o nome do veículo de publicação e lista dos artigos que foram citados/referenciados por cada uma das publicações. Ao longo deste capítulo os artigos citados/referenciados serão chamados de citações.

Foi realizada uma normalização inicial dos dados existentes nesse conjunto coletando as seguintes informações sobre as publicações: autores, tipo da publicação (artigo, revisão de livro e etc.), idioma utilizado, área, editor, tipo do editor (periódico, revista, etc.) e quais artigos foram citados.

¹ https://webofknowledge.com/

Neste conjunto existem 419.103 autores, 356.525 artigos publicados e 2.494.278 citações únicas. Destes 419.103 autores, 5% publicou mais de 25 artigos e realizou mais de 500 citações, sendo que o autor com o maior número de publicações atingiu 1.004 artigos publicados e realizou 26.990 citações. A figura 5 apresenta a distribuição de autores pelo número de publicações, enquanto a figura 6 mostra os autores pelo número de citações.

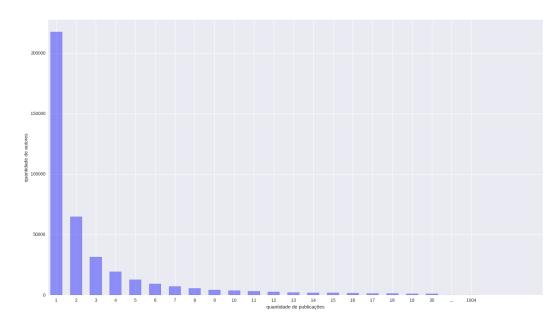


Figura 5 – Distribuição de autores por sua quantidade de publicações.

Fonte: Arthur Patricio Grava, 2016

A publicação mais recente no conjunto de dados foi realizada no ano de 2012, a distribuição geral das publicações ao longo dos anos pode ser verificada na figura 7, na qual a concentração de 50% de artigos publicados se dá a partir do ano de 2008.

4.1.2 Seleção de dados

A tarefa de comparação entre autores utilizando correlação de Pearson, tem complexidade elevada, pois realiza comparações entre todos ou autores utilizando suas citações, ou seja, é um cálculo com complexidade assintótica de $O(n^2m)$ onde n é o número de autores e m o número de artigos.

Assim, calcular a similaridade entre 419.103 autores seria uma tarefa com complexidade elevada em que, nos experimentos iniciais, estimou-se, a partir da execução de 1000

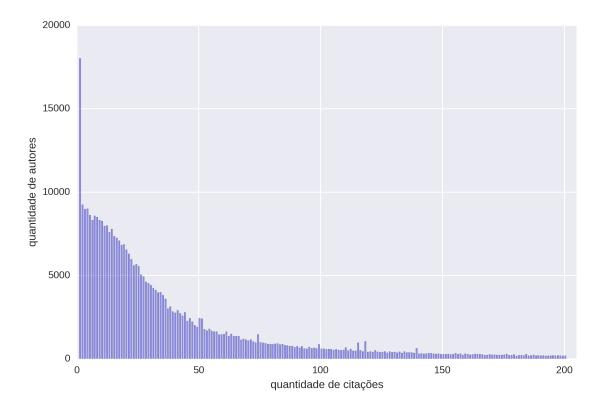


Figura 6 – Distribuição de autores por suas citações, limitados até 200 citações.

casos, a necessade de 2 meses para conclusão do cálculo da correlação *Pearson* entre todos os pares de autores.

Assim sendo, para solucionar esse problema duas estratégias foram realizadas: realizar um corte no conjunto de dados selecionando apenas autores com um número maior de citações, que será discutido nesta seção, e realizar de maneira mais eficiente o cálculo de similaridade mantendo uma lista de itens que ambos autores citaram, comparando assim uma quantidade bem menor de itens, conforme será explicado na seção 4.2.

A partir da figura 6 é possível observar que aproximadamente 50% dos autores do conjunto citaram mais de 24 artigos, portanto com uma divisão no conjunto e mantendo os autores que realizaram 24 citações, ou mais, é possível manter metade das informações sobre citações e autores. O arquivo resultante da divisão possui 221.341 autores únicos, mantendo 2.477.106 citações únicas, ou seja, embora o número de autores tenha diminuído a quantidade de citações disponíveis se manteve próxima ao conjunto original.

Por esses motivos a divisão do conjunto foi realizada podendo, também, reduzir a possibilidade de existirem autores propensos a sofrer com o problema comum em novos

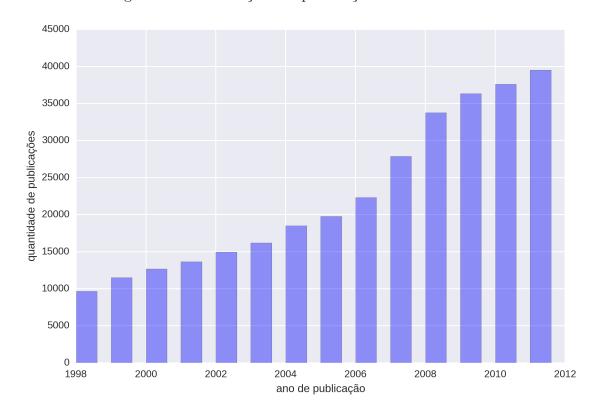


Figura 7 – Distribuição das publicações através dos anos.

usuários citado na seção 2.3, uma vez que cada autor no conjunto citou, ao menos, 24 artigos.

O conjunto resultante (W), contendo 221.341 autores únicos, foi dividido de maneira aleatória em dois subconjuntos disjuntos, ou seja, sem repetição de elementos, estes conjuntos são $W_{treinamento}$ e W_{teste} .

O conjunto $W_{treinamento}$, contendo 75% dos dados pertencentes a W, foi utilizado como base para o cálculo de recomendações, para o cálculo de similaridade entre todos os autores e para gerar dois outros conjuntos: Citações de autores e Rede de coautoria, que foram utilizados como entrada em alguns dos artigos descritos na seção 4.3.

- Citações de autores é um arquivo criado que contem os autores e suas citações. Cada linha desse arquivo é representado por um autor e uma de suas citações, sendo que citações repetidas, porém em publicações diferentes são aceitas;
- Uma *Rede de coautoria* foi criada a partir de todas as publicações existentes no conjunto original, ou seja, verificou-se os autores originais de cada uma das pu-

blicações criando relações entre autores que dividiam uma mesma publicação realizada, definindo-os como coautores. A partir disso, cada linha do arquivo final possui um autor e todos os seus coautores, juntamente com as datas em que eles possuem artigos publicados juntamente. Portanto se um autor a publicou três artigos com o autor b em diferentes anos, o autor b aparecerá três vezes na lista de coautores de a com as datas dessas publicações, existindo a reciprocidade quanto a lista de b.

Já o conjunto W_{teste} contém o restante dos dados pertencentes a W, representando 25% do total e foi utilizado nas tarefas de avaliação dos resultados dos algoritmos, portanto, este é o conjunto de dados utilizado para avaliar a qualidade do sistema de recomendação desenvolvido.

4.2 Similaridade entre autores

A similaridade entre os autores foi definida utilizando a correlação de *Pearson*, atibuindo a cada par de autores o coeficiente de correlação entre eles (RICCI; ROKACH; SHAPIRA, 2011). O cálculo de similaridade foi realizado em duas etapas: transformação do arquivo de entrada e execução da comparação.

A transformação do arquivo foi desenvolvida na linguagem de programação $Python^2$ e utilizando uma biblioteca de representações de conjuntos de dados chamada $Dato^3$. A partir dos dados da WoS foi gerado um arquivo representando o autor e cada uma de suas citações, juntamente com a quantidade de vezes que esse artigo foi citado por esse autor, realizando uma normalização desse número de artigos citados respeitando o intervalo [1, 2].

Tabela 4 – Exemplo de arquivo extraído a partir do conjunto da WoS

Autor	Citação (DOI)	Score
cornejo dr	10.1063/1.373144	1.245
cornejo dr	10.1063/1.119970	1.03
cornejo dr	10.1016/s0304-8853(01)00055-5	1.816
kuckelhaus s	$10.1002/\mathrm{jmri.}1880030131$	2.0
kuckelhaus s	10.1021/bc00006a007	1.0

Fonte: Arthur Patricio Grava, 2016

² https://www.python.org

³ http://www.dato.com



Figura 8 – Representação do arquivo de citações em memória para o cálculo da similaridade entre os autores

Sendo assim, cada linha foi organizada contendo um autor, sua citação e a quantidade de vezes que ele citou esse artigo dentro do intervalo definido. O formato definido está ilustrado na tabela 4.

A leitura do arquivo de citações gerado foi realizada utilizando a linguagem $Java^4$ representando esses dados em memória como um mapa no qual as chaves são os autores e apontam para a listas de citações realizadas pelo respectivo autor, como ilustrado pela figura 8. A partir dessa representação, para cada par de autores é calculada a intersecção de seu conjunto de citações, depois utilizando o score - que é representado pela quantidade de vezes que um autor citou determinado artigo - para calcular a correlação entre esses dois autores, assim como apresentado no algoritmo 1.

Algoritmo 1 Algoritmo para calcular a correlação de *Pearson* entre dois vetores de citações, utilizando as listas de citações pré-existente de dois autores

```
1: function PEARSON(a[vetor], b[vetor])
      mediaA = soma(a)/tamanho(a)
 2:
      mediaB = soma(b)/tamanho(b)
 3:
      intersec = a \cap b
 4:
      numerador = 0.0, denominador 1 = 0.0, denominador 2 = 0.0
 5:
       for cada item i \in intersec do
 6:
 7:
          numerador = numerador + ((a[i] - mediaA) * (b[i] - mediaB))
          denominador1 = denominador1 + (a[i] - mediaA)^2
 8:
          denominador2 = denominador2 + (b[i] - mediaB)^2
 9:
       score = numerador/(\sqrt{denominador1} * \sqrt{denominador2})
10:
       return score
11:
```

⁴ http://www.java.com

O cálculo de similaridade entre autores é uma tarefa paralelizável, uma vez que cada comparação depende de informações que não necessitam de sincronismo e que não apresentam mudanças. Assim, foi utilizado o ambiente de *threads* disponível na linguagem *Java* para realizar esta tarefa de maneira paralela.

O algoritmo 2 apresenta a estratégia utilizada, na qual cada autor foi considerado como uma possível célula de paralelismo, sendo assim *m threads*, que calculam o coeficiente de correlação entre os autores, simultâneas são permitidas.

```
Algoritmo 2 Algoritmo para paralelizar a execução da correlação Pearon
```

```
1: function CORRELAÇÃO(citacoes, threadSize)
       pool = novo pool de threads com tamanho <math>threadSize
       autores = citacoes.chaves()
 3:
 4:
       function CELULA(autor, autores, citacoes)
           candidatos = []
 5:
          for b \in autores do
 6:
              if author \neq b then
 7:
                  candidatos[b] = pearson(citacoes[autor], citacoes[b])
 8:
          return candidatos
 9:
       for a \in autores do
           while pool.cheio() == true \ do
10:
              esperar(10ms)
11:
           thread \leftarrow celula(a, autores, citacoes)
12:
           pool.executar(thread)
13:
```

O resultado de cada uma das células executadas é salvo em um arquivo separado por vírgulas (CSV) com três colunas, onde a primeira coluna representa um autor, a segunda coluna seu autor similar e a terceira coluna o coeficiente de correlação calculado. Um exemplo de como o arquivo está organizado pode ser observado no formato de grade na tabela 5.

Tabela 5 – Modelo em formato de tabela do arquivo resultante do cálculo de similaridade entre os autores

Autor	Autor Similar	Coeficiente		
cornejo dr	rezende am	0.801		
cornejo dr	azevedo a	0.754		
cornejo dr	tedesco ac	-0.234		
oliveira dm	rezende am	0.955		
oliveira dm	lacava zgm	-0.891		
lima ecd	gansau c	0.123		

Fonte: Arthur Patricio Grava, 2016

4.3 Técnicas comparativas implementadas

Esta sub-seção descreve todos os algoritmos de recomendação implementados neste trabalho, partindo do filtro colaborativo na sub-seção 4.3.1, recomendação a partir de uma rede de confiança na sub-seção 4.3.2, recomendação baseada em redes sociais na sub-seção 4.3.3 e, por fim, recomendação baseada em redes sociais com a utilização de técnicas de confiança na sub-seção 4.3.4.

4.3.1 Filtro colaborativo baseado em usuário

O filtro colaborativo baseado em usuários (UCF), como encontrado na literatura, aplicado aos dados da WoS utiliza os dados sobre citações dos autores para determinar a similaridade entre eles. O cálculo desta similaridade foi realizado separadamente (e seus resultados foram armazenados) para possibilitar a reutilização desta informação para os diferentes valores para os parâmetros de entrada do algoritmo (parâmetros relacionados ao número máximo de autores similares utilizados).

A partir da determinação dos autores similares, são escolhidos k autores para serem utilizados como autores mais similares. O valor de k pode ser baseado em algum limite (lim) estabelecido, por exemplo, apenas aqueles com coeficiente de correlação de Pearson maior que θ, θ , pois indica uma correlação positiva entre os autores, também pode-se utilizar os n primeiros autores com coeficientes mais próximos de $1, \theta$ ou realizando uma mistura das duas abordagens, selecionando os n primeiros autores que possuem coeficiente de correlação maior que lim (RICCI; ROKACH; SHAPIRA, 2011).

Selecionados os k autores, a predição de itens recomendados para um autor a é calculada como indicado pelo algoritmo 3, desenvolvido utilizando Java no qual, a partir dos k autores similares, são selecionados os artigos citados pelos autores similares e que o autor a não citou, caracterizando o possível conjunto de citações C recomendáveis para a.

Assim, para cada possível citação $c \in C$, é calculado um score ordenando cada um dos itens a partir do score calculado para cada um itens, que tem o peso indicado pela similaridade entre a e um autor $k \in K$ multiplicado pela diferença entre a quantidade de vezes que k citou o artigo c com a quantidade média de citações que ele possui.

O cálculo foi modelado para que fosse centrado nas recomendações específicas de um determinado autor a, portanto é possível paralelizar diferentes tarefas para m diferentes

Algoritmo 3 Cálculo de filtro colaborativo baseado em usuário

```
1: function PREDICT(autor, similares, citacoes)
      naoCitados = \{ \} //coleção de artigos únicos \}
      medias = []
 3:
      for similar \in similares do
 4:
          naoCitados.adicionar(\{citacoes[similar] - citacoes[autor]\})
 5:
 6:
          medias[similar] = media(citacoes[similar])
      mediaA = media(citacoes[author])
 7:
 8:
      recomendacoes = []
       for citacao \in naoCitados do
 9:
          numerador = 0.0, denominador = 0.0
10:
11:
          for similar \in similares do
12:
             parcial = (similar.score * (citacoes[similar][item] - medias[similar]))
13:
             numerador = numerador + parcial
             denominador = denominador + similar.score
14:
          recomendacoes[item] = mediaA + (numerador/denominador)
15:
      return recomendações
```

Algoritmo 4 Paralelismo obtido atravé de threads que calculam a predição dos itens para um determinado autor

```
1: function RECOMENDACOES(similares, citacoes, m)
2: pool = novo pool de threads com tamanho m
3: autores = citacoes.chaves()
4: for autor \in autores do
5: while pool.cheio() == true do
6: esperar(10ms)
7: thread \leftarrow predict(autor, similares[autor], citacoes)
8: pool.executar(thread)
```

autores. Para isso foi utilizado o ambiente de threads do *Java*, realizando um processo parecido com o cálculo de similaridade apresentado na seção 4.2.

A maneira que o paralelismo foi implementado por ser observado no algoritmo 4 no qual é utilizado como entrada o arquivo de citações gerado anteriormente e representado pela figura 8 e com o arquivo de autores similares gerado pelo cálculo de similaridade, além do parâmetro m que indica a quantidade desejada de threads em execução.

Após a execução do UCF os resultados recomendadas são coletados e salvos em um arquivo separado por vírgulas possuindo três colunas distintas. A primeira coluna contém o identificador do autor a para o qual está sendo feita a recomendação, a segunda coluna é a citação recomendada e a terceira coluna representa o score atribuído pela técnica UCF. Um exemplo de tabulação deste arquivo pode ser observado na tabela 6.

Tabela 6 – Modelo em formato de tabela do arquivo resultante do cálculo de recomendações do algoritmo UCF.

Autor	Citação Recomendada	UCF score
cornejo dr	10.1016/s0304-8853(00)01225-7	1.432
cornejo dr	10.1016/s0022-2313(02)00227-2	1.167
cornejo dr	10.1016/s0304-8853(98)00583-6	1.0002
azevedo a	10.1016/1011-1344(95)07174-z	1.988
azevedo a	10.1016/s0304-8853(00)01225-7	1.246
		•••
tedesco ac	10.1016/s0022-2313(02)00227-2	1.00234

4.3.2 Recomendação baseada em redes de confiança

Golbeck (2006) desenvolveu o algoritmo baseado em redes sociais de confiança (TRS) no qual a predição do score de um item está fortemente relacionada ao fator de confiança entre os usuários. Aplicando este algoritmo no contexto de artigos científicos, a confiança entre dois autores a e b foi modelada de acordo com a quantidade de vezes que este par de autores publicou artigos em conjunto, utilizando um processo similar ao utilizado por GAO; XU; CAI (2011a), que pode ser observado no algoritmo 5.

Algoritmo 5 Definição de confiança entre coautores.

```
1: function CONFIANCA(publicacoes)
 2:
       confianca = []
       for publicacao \in publicacoes do
 3:
           autores = publicacao.autores()
 4:
           for a \in autores do
 5:
              for b \in autores do
 6:
                  if a \neq b then
 7:
                      anterior = 0
 8:
                      if citacoes[a].possui(b) then
 9:
10:
                         anterior = citacoes[a][b]
                      citacoes[a][b] = anterior + 1
11:
       return confianca
```

Definida a confiança entre dois autores a e b, a predição do score de uma citação, sendo C o conjunto de coautores de um autor a, é definida pela somatória da confiança entre o autor a e seus coautores multiplicada pela quantidade de vezes que estes coautores citaram determinado artigo. As instruções de execução da técnica estão ilustradas no algoritmo 6.

Algoritmo 6 Cálculo do *score* para as possíveis citações para um autor a utilizando os autores em que a confia.

```
1: function PREDICT(autor, citacoes, confiados)
      naoCitados = []
 2:
      for confiado \in confiados do
 3:
          naoCitados.adicionar(\{citacoes[confiado] - citacoes[autor]\})
 4:
 5:
      recomendacoes = []
       for artigo \in naoCitados do
 6:
          numerador = 0.0
 7:
          denominador = 0.0
 8:
          for confiado \in confiados do
 9:
             numerador = numerador + (confiado.score * citacoes[confiado][artigo])
10:
11:
             denominador = denominador + confiado.score
          recomendacoes[artigo] = numerador/denominador
12:
      return recomendacoes
```

O algoritmo 6 utiliza como entrada um autor a para o qual as recomendações serão calculadas, o conjunto de citações e quais são os autores em que a confia. O retorno do algoritmo respeita o formato definido anteriormente e exemplificado pela tabela 6, também utilizando a vírgula como separador de colunas, modificando apenas o rótulo atribuído à terceira coluna de "UCF score" para "TRS score".

O cálculo do TRS foi paralelizado utilizando um processo similar ao descrito pelo algoritmo 4, apenas modificando o parâmetro de entrada trocando o conjunto de dados de similaridade pelo conjunto de confiança (resultante da execução do algoritmo 5).

4.3.3 Recomendação baseada em redes sociais

O algoritmo baseado em redes sociais (SRS), desenvolvido por Groh e Ehmig (2007), utiliza a informação da rede social criada entre os usuários como os usuários similares, partindo de grupos com 2 amigos até grupos maiores, nos quais o peso da similaridade é dado pela existência da relação. A implementação dessa técnica é descrita pelo pseudo-código descrito pelo algoritmo 7.

Neste trabalho, desenvolveu-se também uma variante desta implementação (SRS*) utilizando como peso de similaridade a existência de uma relação de coautoria entre autores (1 em caso positivo, do contrário 0), multiplicado pelo coeficiente de correlação entre os autores, como pode ser observado no algoritmo 8, que tem como referência adicional, no

Algoritmo 7 Cálculo do SRS para um autor a qualquer.

```
1: function PREDICT(autor, coautores, citacoes)
      naoCitados = \{ \}
      medias = []
 3:
      for coautor \in coautores do
 4:
          naoCitados.adicionar(\{citacoes[coautor] - citacoes[autor]\})
 5:
          medias[coautor] = media(citacoes[coautor])
 6:
      mediaA = media(citacoes[autor])
 7:
 8:
      recomendacoes = []
      for citacao \in naoCitados do
 9:
          numerador = 0.0, denominador = 0.0
10:
11:
          for coautor \in coautores do
12:
             parcial = (citacoes[coautor][item] - medias[coautor])
             numerador = numerador + parcial
13:
             denominador = denominador + 1
14:
          recomendacoes[item] = mediaA + (numerador/denominador)
15:
      return recomendações
```

parâmetro de citações para os coautores, o valor do coeficiente de correlação relacionado a um autor a.

Algoritmo 8 Cálculo do SRS * para um autor a qualquer.

```
1: function PREDICT(autor, coautores, similares, citacoes)
      naoCitados = \{ \}
 2:
 3:
      medias = []
 4:
      for coautor \in coautores do
          naoCitados.adicionar(\{citacoes[coautor] - citacoes[autor]\})
 5:
          medias[coautor] = media(citacoes[coautor])
 6:
      mediaA = media(citacoes[autor])
 7:
      recomendacoes = []
 8:
       for citacao \in naoCitados do
 9:
10:
          numerador = 0.0, denominador = 0.0
11:
          for coautor \in coautores do
                                 (coautor.coeficiente * (citacoes[coautor][item]))
             parcial
12:
   medias[coautor]))
             numerador = numerador + parcial
13:
             denominador = denominador + coautor.coeficiente
14:
          recomendacoes[item] = mediaA + (numerador/denominador)
15:
      return recomendacoes
```

Assim como o UCF, os algoritmos SRS e SRS*, descritos nesta sub-seção, também podem ser paralelizados, utilizando um processo idêntico ao descrito para o algoritmo 4, modificando apenas os parâmetros de entrada da função *predict*, atendendo aos padrões indicados pelos algoritmos 7 e 8.

A saída de ambos algoritmos respeitam o formato definido anteriormente e exemplificado pela tabela 6, também utilizando a vírgula como separador de colunas, modificando apenas o rótulo atribuído à terceira coluna de "UCF score" para "SRS score", no caso do primeiro algoritmo, e "SRS* score" para o segundo algoritmo.

4.3.4 Recomendação baseada em redes sociais com confiança

A recomendação baseada em redes sociais e com inserção de confiança (TSRS) foi implementada utilizando o conceito de SRS descrito na sub-seção 4.3.3 pelo algoritmo 7, ao qual foi inserido um peso representando a força entre a relação dos coautores, que é definida pela confiança entre eles, seguindo o cálculo de confiança descrito pelo algoritmo 5. A ilustração desta definição é encontrada no algoritmo 9.

Algoritmo 9 Cálculo do TSRS para um autor a qualquer.

```
1: function PREDICT(autor, coautores, citacoes, confianca)
      naoCitados = \{ \}
 2:
      medias = []
 3:
      for coautor \in coautores do
 4:
          naoCitados.adicionar(\{citacoes[coautor] - citacoes[autor]\})
 5:
          medias[coautor] = media(citacoes[coautor])
 6:
      mediaA = media(citacoes[autor])
 7:
 8:
      recomendacoes = []
9:
       for citacao \in naoCitados do
          numerador = 0.0, denominador = 0.0
10:
          for coautor \in coautores do
11:
                             (confianca[autor][coautor] * (citacoes[coautor][item] -
12:
             parcial
   medias[coautor]))
             numerador = numerador + parcial
13:
             denominador = denominador + confianca(autor, coautor)
14:
          recomendacoes[item] = mediaA + (numerador/denominador)
15:
      return recomendacoes
```

Assim como descrito na sub-seção 4.3.3, é possível construir uma variação desta técnica onde se pode inserir, além da força entre a relação de um par de autores a e b, o coeficiente de correlação calculado entre eles, representada pelo algoritmo 10. A variação apresentada tem como rótulo a sigla $TSRS^*$.

Assim como os algoritmos anteriores o TSRS e o TSRS* foram paralelizados utilizando a mesma estrutura definida pelo algoritmo 4, somente realizando uma modificação nos parâmetros de entrada da função *predict* para aceitar a adição do conjunto de con-

Algoritmo 10 Cálculo do TSRS* para um autor a qualquer.

```
1: function PREDICT(autor, coautores, citacoes, confianca)
      naoCitados = \{ \}
      medias = []
 3:
      for coautor \in coautores do
 4:
          naoCitados.adicionar(\{citacoes[coautor] - citacoes[autor]\})
 5:
          medias[coautor] = media(citacoes[coautor])
 6:
 7:
      mediaA = media(citacoes[autor])
      recomendacoes = []
 8:
      for citacao \in naoCitados do
 9:
          numerador = 0.0, denominador = 0.0
10:
11:
          for coautor \in coautores do
             peso = confianca[autor][coautor] * coautor.coeficienteCorrelacao
12:
             parcial = (peso * (citacoes[coautor][item] - medias[coautor]))
13:
14:
             numerador = numerador + parcial
             denominador = denominador + peso
15:
          recomendacoes[item] = mediaA + (numerador/denominador)
16:
      return recomendacoes
```

fiança calculado entre os coautores e substituir o conjunto de similaridade pelo conjunto de coautorias.

As saídas de ambos algoritmos respeitam o formato definido anteriormente e exemplificado pela tabela 6, também utilizando a vírgula como separador de colunas, modificando apenas o rótulo atribuído à terceira coluna de "UCF score" para "TSRS score", no caso do primeiro algoritmo, e "TSRS* score" para o segundo algoritmo.

4.4 Técnicas propostas

A técnica proposta foi desenvolvida utilizando a linguagem Java e tem por premissa adicionar contexto aos sistemas de recomendação baseados em redes sociais, especificamente utilizando o tempo como contexto principal entre as relações de coautoria existentes no conjunto de dados da WoS.

Jiang et al. (2014b) desenvolveram um sistema de recomendações inserindo informações de contexto social, mapeando informações de contexto como a interação existente entre usuários, além da relação existe entre eles, e também utilizando informações sobre a interação dos usuários com os itens e seu conteúdo. Os resultados obtidos utilizando medidas de ranking e métricas de predição de erro foram superiores aos correlatos.

Sim et al. (2012b) tiveram uma perspectiva diferente em relação ao contexto, realizando a contextualização através dos itens, sendo assim, as medidas de média de avaliação e predição utilizam as avaliações do determinado contexto em que o usuário está inserido. No caso de o usuário não ter nenhuma avaliação no contexto do item, a avaliação média do contexto é utilizada. A partir disso a inserção do contexto obteve melhores resultados que os correlatos em termos de *MAE*, precisão e revocação.

Liu, He e Zhao (2013) implementaram o tempo como um complemento do contexto, influenciando itens e usuários, por exemplo, a popularidade de itens é algo que está relacionado ao tempo, pois em determinado período ele pode estar em ascensão, até que pare de ser um item procurado. Já usuários, por exemplo, expressam suas preferências explicitamente ou implicitamente por meio da sua interação com o sistema, porém, essas preferências tendem a mudar com o decorrer do tempo. Modelando isso em sua técnica de recomendação LIU; HE; ZHAO (2013) tiveram resultados que superaram os correlatos em termos de precisão e MAP (Mean Average Precision).

Desta forma, a adição do tempo como uma variável de contexto importante na relação entre coautores se dá pela hipótese de que as relações de coautoria mais recentes de um autor podem ter uma relevância maior quando comparadas com relações que aconteceram muito tempo atrás. A equação 16 detalha o princípio desta relação entre dois autores a e b onde max(a) é o ano da última relação entre a e seus coautores, min(b) é o ano da primeira relação entre a e seus coautores, ano(p) é o ano de publicação de um artigo p em que os autores a e b foram autores, a é o valor mínimo da normalização dos valores (neste trabalho utilizou-se 0.25) e a0 é o valor máximo da normalização (neste trabalho utilizou-se a1.0).

$$relacao(a,b) = \sum_{p \in P} \frac{\alpha + (ano(p) - min(a)) * (\beta - \alpha)}{(max(a) - min(a))}$$
(16)

A implementação realizada para calcular as recomendações de artigos para os autores está explanada em forma de pseudo-código pelo algoritmo 11 onde é utilizado como peso a diretiva de tempo no qual a relação entre os autores aconteceu.

Além disso, ainda idealizou-se uma estratégia na qual, para cada autor, o resultado de determinado algoritmo d_1 fosse mais relevante que outro algoritmo d_n qualquer. Sendo assim, foi criado um arquivo contento os *scores* atribuídos por cada um dos algoritmos

Algoritmo 11 Cálculo do Time-TSRS para um autor a qualquer.

```
1: function PREDICT(autor, coautores, citacoes)
      naoCitados = \{ \} //coleção de artigos únicos
      medias = []
 3:
      for coautor \in coautores do
 4:
          naoCitados.adicionar(\{citacoes[coautor] - citacoes[autor]\})
 5:
          medias[coautor] = media(citacoes[coautor])
 6:
      mediaA = media(citacoes[author])
 7:
 8:
      recomendacoes = []
      for citacao \in naoCitados do
 9:
          numerador = 0.0, denominador = 0.0
10:
11:
          for coautor \in coautores do
12:
             peso = (coautor.score * relacao(autor, coautor))
             parcial = (peso * (citacoes[coautor][item] - medias[coautor]))
13:
             numerador = numerador + parcial
14:
             denominador = denominador + peso
15:
          recomendacoes[item] = mediaA + (numerador/denominador)
16:
      return recomendacoes
```

apresentados, juntamente com um rótulo dizendo se aquela recomendação foi útil ou não. o formato do arquivo segue a definição da tabela 7.

Tabela 7 – Formato do arquivo que contém todos os scores dos algoritmos utilizados.

alg_1	alg_2	alg_3	 alg_n	rótulo
1.387	1.498	1.432	 1.0023	relevante
1.654	1.852	1.002	 1.582	não relevante
1.789	1.228	1.115	 1.9956	não relevante
1.892	1.741	1.778	 1.123	não relevante
1.997	1.147	1.996	 1.458	relevante
1.336	1.493	1.994	 1.379	não relevante

Fonte: Arthur Patricio Grava, 2016

A partir desse arquivo foi utilizada a regressão linear para definir qual é o peso de cada um dos algoritmos na determinação do resultado, para, assim, criar um novo ranqueamento das recomendações de um autor a partir da utilização destes pesos. O objetivo da regressão linear foi produzir a melhor equação (os melhores pesos para cada um dos *scores* produzidos) de forma a estimar a relevância de um dado artigo. Para a realização do cálculo da regressão o rótulo "relevante" foi substituído pelo valor 1,0 e o rótulo "não relevante" foi substituído pelo valor 0,0.

Os resultados da execução destas abordagens foram salvos em um arquivo separado por vírgulas e com três colunas, onde a primeira coluna indica o autor referência, a segunda coluna o artigo recomendado e a terceira coluna o *score* atribuído pela técnica empregada.

5 Resultados

Neste capítulo serão descritos os experimentos realizados, presentes na seção 5.1, na seção 5.1 serão mostrados os resultados obtidos através dos experimentos e, por fim, na seção 5.2 serão discutidas as considerações a cerca dos resultados.

5.1 Experimentação

Nesta seção é realizado o detalhamento de todos os passos para concluir os objetivos do trabalho onde cada experimento foi dividido de acordo com o algoritmo que estava sendo executado. Tem-se assim na sub-seção 5.1.1 o experimento realizado através do algoritmo UCF, na sub-seção 5.1.2 o detalhamento do procedimento executado através do TRS, na sub-seção 5.1.3 o algoritmo SRS e SRS*, na sub-seção 5.1.4 os algoritmos TSRS e TSRS*, na sub-seção 5.1.5 a técnica proposta utilizando o tempo como contexto (Time-TSRS) e, por fim, na sub-seção 5.1.6.

Para todos os algoritmos foram consideradas diferentes quantidades de autores similares (vizinhos). Os números de vizinhos testados foram: 5, 10, 15, 20, 25 e 30 vizinhos. Para cada um dos valores de vizinhos foi computado um conjunto de recomendações R_n , em que n representa a quantidade de vizinhos utilizada, portanto, o conjunto gerado utilizando 10 vizinho é descrito por R_{10} . Para cada uma das quantidades de vizinhos, o desempenho da recomendação foi medido para diferentes quantidades de itens recomendados, isto é, testou a qualidade da recomendação para os primeiros k itens (artigos). Os valores utilizados para k foram 5, 10, 15, 20 e 25, sendo avaliados utilizando o método de pesquisa especificado.

Além disso, para cada experimento foram utilizadas duas abordagens de validação. Na primeira validação as métricas foram calculadas avaliando-se em todos os autores encontrados em W_{teste} , ou seja, para cada um dos autores serão procuradas suas recomendações nos conjuntos R_n de cada um dos algoritmos, avaliando o desempenho geral da técnica utilizada.

A segunda abordagem considera apenas os autores para os quais cada técnica conseguiu gerar recomendações, ou seja, serão avaliados apenas os resultados dos autores que possuem recomendações em R_n . Desta maneira pretende-se, em caráter específico, avaliar a qualidade das recomendações geradas, ou seja, pretende-se verificar a qualidade das

recomendações geradas considerando apenas os autores para os quais a técnica conseguiu gerar uma ou mais recomendações.

5.1.1 Experimento UCF

Para a realização do experimento utilizando o algoritmo UCF foi necessário executar o cálculo de similaridade entre todos os autores, representado pelo conjunto S, especificado pelo algoritmo 1, do conjunto de dados $W_{treinamento}$, já que uma das entradas do algoritmo é o dado sobre o coeficiente de correlação entre os autores, como especificado no algoritmo 3.

Tabela 8 – Métricas coletadas a partir da execução do algoritmo UCF verificando todos os autores e artigos no conjunto de testes W_{teste} .

			UCF - Us	ser Based C	'ollabe	prative Filte	erinq				
		5 vizinh	ios				10 vizinho	os			
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.		
5	2,383	0,0130	0,0037	0,00074	5	2,01339	0,01150	0,00330	0,00066		
10	3,599	0,0146	0,0059	0,00059	10	2,93235	0,01270	0,00496	0,00050		
15	4,399	0,0152	0,0075	0,00050	15	3,62180	0,01325	0,00615	0,00041		
20	5,062	0,0156	0,0088	0,00044	20	$4,\!15550$	0,01355	0,00713	0,00036		
25	5,668	0,0159	0,0099	0,00040	25	4,64018	0,01376	0,00799	0,00032		
	15 vizinhos					20 vizinhos					
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.		
5	2,11325	0,01202	0,00338	0,00068	5	2,21404	0,01263	0,00345	0,00069		
10	3,03268	0,01323	0,00494	0,00049	10	3,15087	0,01387	0,00504	0,00050		
15	3,66853	0,01374	0,00604	0,00040	15	3,80871	0,01439	0,00613	0,00041		
20	4,19627	0,01403	0,00695	0,00035	20	4,33416	0,01468	0,00705	0,00035		
25	4,65438	0,01423	0,00773	0,00031	25	4,80785	0,01489	0,00782	0,00031		
		25 vizinl	hos				30 vizinho	os			
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.		
5	2,27084	0,01308	0,00349	0,00070	5	2,35697	0,01367	0,00356	0,00071		
10	3,25119	0,01437	0,00513	0,00051	10	3,37626	0,01502	0,00519	0,00052		
15	3,94065	0,01491	0,00624	0,00042	15	4,09778	0,01559	0,00636	0,00042		
20	4,49267	0,01522	0,00714	0,00036	20	4,66858	0,01591	0,00727	0,00036		
25	4,98101	0,01543	0,00794	0,00032	25	5,17388	0,01613	0,00807	0,00032		

Fonte: Arthur Patricio Grava, 2016

A tabela 8 contém os resultados do algoritmo *User Based Collaborative Filtering* avaliado para o conjunto de autores presentes em W_{teste} . Os resultados em negrito ressaltam aqueles que foram os melhores resultados levando-se em consideração o k em questão.

A partir destes resultados pode-se observar que os melhores valores para S@K foram encontrados quando utilizado somente os 5 vizinhos mais próximos/similares. O

mesmo pode-se dizer para as medidas revocação e precisão, ou seja, com 5 vizinhos foi possível encontrar resultados em um maior número de oportunidades (entre os $top\ k$), além de encontrar uma maior quantidade de resultados, como indicam a revocação e a precisão. Por outro lado, os melhores resultados de MRR foram observados ao se utilizar 30 vizinhos. Pode-se dizer que os itens relevantes são encontrados em posições mais relevantes que no restante do conjunto.

Tabela 9 – Métricas coletadas a partir da execução do algoritmo UCF verificando todos os autores do conjunto de recomendações.

			UCF - Use	er Based Co	ollabo	rative Filter	ring		
		5 vizin	hos				10 vizinho	os	
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	3,63720	0,01985	0,00562	0,00112	5	3,07295	0,01756	0,00504	0,00101
10	$5,\!49356$	0,0223	0,00896	0,0009	10	$4,\!47554$	0,01939	0,00757	0,00076
15	6,71365	0,02326	0,01142	0,00076	15	$5,\!52782$	0,02022	0,00939	0,00063
20	7,72538	0,02383	0,01336	0,00067	20	6,34238	0,02067	0,01088	0,00054
25	8,65182	0,02423	0,01511	0,0006	25	7,08213	0,021	0,01219	0,00049
		15 vizin	hos		20 vizinhos				
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	3,22538	0,01834	0,00516	0,00103	5	3,3792	0,01927	0,00527	0,00105
10	4,62866	0,0202	0,00754	0,00075	10	4,80905	0,02117	0,0077	0,00077
15	5,59914	0,02096	0,00921	0,00061	15	5,81309	0,02196	0,00936	0,00062
20	6,40461	0,02142	0,0106	0,00053	20	6,61507	0,02241	0,01076	0,00054
25	7,1038	0,02172	0,01179	0,00047	25	7,33803	0,02273	0,01194	0,00048
		25 vizin	hos				30 vizinho	os	
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	3,4659	0,01996	0,00533	0,00107	5	3,59735	0,02087	0,00543	0,00109
10	4,96217	0,02193	0,00783	0,00078	10	$5,\!15305$	0,02293	0,00793	0,00079
15	6,01446	0,02276	0,00953	0,00064	15	6,25428	$0,\!0238$	0,0097	0,00065
20	6,85699	0,02323	0,01089	0,00054	20	7,12548	$0,\!02428$	0,0111	0,00056
25	7,60233	0,02356	0,01212	0,00048	25	7,89669	$0,\!02462$	0,01232	0,00049

Fonte: Arthur Patricio Grava, 2016

Os resultados para o experimento do UCF verificando todos os autores que receberam ao menos uma recomendação são apresentados na tabela 9. Os resultados em negrito ressaltam aqueles que foram os melhores resultados levando-se em consideração o k em questão, comparando-se as diferentes configurações de vizinhos utilizadas.

Observa-se que os melhores resultados para S@K foram encontrados utilizando somente 5 vizinhos, o mesmo pode-se dizer para as medidas revocação e precisão, ou seja, com 5 vizinhos foi possível encontrar resultados em um maior número de oportunidades, além de encontrar uma maior quantidade de resultados, como indicam a revocação e

a precisão. Porém, ao se utilizar 30 vizinhos, onde os melhores resultados de MRR foram observados, pode-se dizer que os itens relevantes são encontrados em posições mais relevantes que no restante do conjunto.

O resultado de ambas as perspectivas, quando considerando o número de vizinhos utilizado foi consistente, mostrando que os valores elevados de S@k, revocação e precisão foram encontrados utilizando como vizinhança os 5 melhores vizinhos, enquanto que o MRR, em ambos os casos, obteve resultados superiores utilizando os 30 primeiros vizinhos.

Porém ao comparar os resultados entre si, pode-se observar que os resultados gerados pelo UCF foram aproximadamente 2 vezes melhores, ou mais relevante, considerando-se apenas os casos para os quais os usuários receberam recomendações, neste caso, confirmando dados resultados compartilhados por RICCI; ROKACH; SHAPIRA (2011). Para técnicas mais robustas, que sejam capazes de gerar recomendações para um número maior de usuários espera-se que esta diferença não seja muito grande.

5.1.2 Experimento TRS

A realização do experimento utilizando a técnica TRS necessitou a execução do cálculo de confiança entre todos os coautores, representados pelo conjunto T e especificado pelo algoritmo 5, do conjunto de dados $W_{treinamento}$, já que a principal referência do algoritmo para a tarefa de recomendação é a confiança existente na rede de coautoria.

A partir disso, a técnica foi executada levando-se em consideração diferentes quantidades de vizinhos e avaliada para diferente quantidades de recomendação.

Os resultados obtidos avaliando-se as recomendações para os autores do conjunto W_{teste} podem ser encontrado na tabela 10. Os resultados em negrito na tabela 10 ressaltam aqueles que foram os melhores resultados levando-se em consideração o k em questão.

Observa-se que os melhores resultados para S@K foram encontrados utilizando 30 vizinhos, sendo que o mesmo pode ser dito da revocação e precisão, ou seja, com 30 vizinhos foi possível encontrar resultados em um maior número de oportunidades, além de encontrar uma maior quantidade de resultados, como indicam a revocação e a precisão. Porém, ao se utilizar 20 vizinhos foram encontrados os melhores resultados de MRR, pode-se dizer que os itens desejados são encontrados em posições mais relevantes que no restante do conjunto.

Tabela 10 – Métricas coletadas a partir da execução do algoritmo TRS verificando todos os autores no conjunto de testes W_{teste}

			TRS - Tr	rust Netwo	rk Re	ecommender	System		
		5 vizinho	OS				10 vizinh	OS	
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	27,45855	0,14288	0,05108	0,01022	5	29,10408	0,15065	0,05463	0,01093
10	39,29195	0,15883	0,07866	0,00787	10	42,15146	$0,\!16822$	0,08587	0,00859
15	45,16169	0,16349	0,09271	0,00618	15	48,40051	$0,\!17318$	0,10179	0,00679
20	48,67538	0,16547	0,10145	0,00507	20	51,98888	$0,\!17521$	0,11134	0,00557
25	51,04746	0,16651	0,10787	0,00431	25	54,32156	$0,\!17623$	0,11803	0,00472
		15 vizinh	.OS				20 vizinh	OS	
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	29,52691	0,15271	0,05558	0,01112	5	29,79765	$0,\!15326$	0,05596	0,01119
10	42,97239	0,17079	0,08783	0,00878	10	43,43279	$0,\!17164$	0,08861	0,00886
15	49,47707	0,17596	0,1043	0,00695	15	50,0465	$0,\!17688$	0,10548	0,00703
20	53,3064	0,17812	0,1144	0,00572	20	53,87995	$0,\!17905$	0,11575	0,00579
25	55,74628	0,17919	0,12147	0,00486	25	56,28685	0,1801	0,12286	0,00491
		25 vizinh	.OS				30 vizinh	OS	
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	29,93967	0,15289	0,05624	0,01125	5	30,03862	0,15269	0,05634	0,01127
10	43,7301	0,17145	0,08907	0,00891	10	43,86799	$0,\!17132$	0,08925	0,00892
15	50,42031	0,17676	0,10616	0,00708	15	50,60218	$0,\!17666$	0,10642	0,00709
20	54,23269	0,17891	0,11648	0,00582	20	54,4196	$0,\!17882$	0,11677	0,00584
25	56,69548	0,17999	0,1237	0,00495	25	56,89247	$0,\!17991$	0,12409	0,00496

Os resultados para o experimento TRS verificando todos os autores que receberam ao menos uma recomendação são apresentados na tabela 11. Os resultados em negrito na tabela 11 ressaltam aqueles que foram os melhores resultados.

Observa-se que os melhores resultados para S@K foram encontrados utilizando 30 vizinhos, o mesmo pode-se dizer para as medidas revocação e precisão, ou seja, com 30 vizinhos foi possível encontrar resultados em um maior número de oportunidades, além de encontrar uma maior quantidade de resultados, como indicam a revocação e a precisão. Porém, ao se utilizar 20 vizinhos, onde os melhores resultados de MRR foram observados, pode-se dizer que os itens de maior interesse são encontrados em posições mais relevantes que no restante do conjunto.

O resultado de ambas as perspectivas, quando considerando o número de vizinhos utilizado, teve um resultado sólido, mostrando que os valores elevados de S@k, revocação e precisão foram encontrados utilizando como vizinhança os 30 melhores vizinhos, enquanto

Tabela 11 – Métricas coletadas a partir da execução do algoritmo TRS verificando todos os autores do conjunto de recomendações.

			TRS - Tr	rust Netwo	rk Re	ecommender	System		
		5 vizinho	OS				10 vizinh	.os	
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	28,74028	0,14955	0,05346	0,01069	5	30,44305	0,15758	0,05714	0,01143
10	$41,\!12604$	0,16625	0,08233	0,00823	10	44,0907	$0,\!17596$	0,08982	0,00898
15	$47,\!26977$	0,17112	0,09703	0,00647	15	50,62725	0,18115	0,10647	0,0071
20	50,94748	0,1732	0,10618	0,00531	20	54,3807	0,18327	0,11646	0,00582
25	53,43029	0,17428	0,11291	0,00452	25	56,82071	0,18434	0,12346	0,00494
15 vizinhos							20 vizinh	os	
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	30,88475	0,15974	0,05814	0,01163	5	31,16779	0,16031	0,05853	0,01171
10	44,94854	0,17865	0,09187	0,00919	10	45,42989	$0,\!17953$	0,09268	0,00927
15	51,75234	0,18405	0,1091	0,00727	15	52,34771	$0,\!18502$	0,11033	0,00736
20	55,75777	0,18631	0,11966	0,00598	20	56,35742	$0,\!18728$	0,12107	0,00605
25	58,30985	0,18743	0,12705	0,00508	25	58,875	0,18839	0,12851	0,00514
		25 vizinh	os				30 vizinh	.OS	
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	31,31634	0,15992	0,05882	0,01176	5	31,41984	0,15972	0,05893	0,01179
10	45,74087	0,17934	0,09317	0,00932	10	45,8851	0,1792	0,09335	0,00934
15	52,73871	0,18489	0,11104	0,0074	15	52,92894	0,18479	$0,\!11132$	0,00742
20	56,72639	0,18714	0,12184	0,00609	20	56,92189	0,18704	$0,\!12214$	0,00611
25	$59,\!30242$	0,18827	0,12939	0,00518	25	59,50846	0,18818	$0,\!12979$	0,00519

que o MRR, em ambos os casos, obteve resultados superiores utilizando os 20 primeiros vizinhos.

Porém ao comparar os resultados entre si, pode-se observar que os resultados gerados pelo TRS não tiveram um incremento de resultado relevante quando comparando os autores presentes em W_{teste} e o conjunto formado apenas por autores que receberam uma ou mais recomendações. Neste caso pode-se inferir que a grande maioria dos autores recebeu ao menos uma recomendação.

5.1.3 Experimento SRS

Nesta sub-seção serão descritos os experimentos realizados utilizando os algoritmos SRS e SRS*, iniciando pelo SRS e, em seguida, finalizando com SRS*.

5.1.3.1 SRS

A execução da técnica SRS tem como requerimento de entrada as relações de coautoria (C) existentes entre os autores do conjunto $W_{treinamento}$, assim como especificado na sub-seção 4.3.3, para isso foi aproveitado o cálculo utilizado pelo algoritmo 5, porém utilizando apenas os relacionamentos entre autores.

A partir disso, o SRS foi executado levando-se em consideração as diferentes quantidades de vizinhos explanadas anteriormente na seção 1.2. Para cada uma dessas configurações de vizinhança executadas foi coletado conjuntos de recomendações, representados por R_n , onde n representa a quantidade de vizinhos utilizadas.

Tabela 12 – Métricas coletadas a partir da execução do algoritmo SRS verificando todos os autores no conjunto de testes W_{teste} .

			SRS - S	ocial Recom	meno	ler Sustem			
		5 vizinh			1100100	er bystem	10 vizinh	00	
T/	COIC OX			D	T/	COIC OX			D
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	3,60806	0,01989	0,00508	0,00102	5	3,06383	0,01704	0,00414	0,00083
10	$5,\!26641$	$0,\!0221$	0,00792	0,00079	10	4,4858	0,01893	0,00644	0,00064
15	$6,\!29853$	0,02292	0,01008	0,00067	15	$5,\!37453$	0,01964	0,00824	0,00055
20	7,0173	0,02333	0,01183	0,00059	20	5,98656	0,01998	0,00975	0,00049
25	7,51023	0,02354	0,01313	0,00053	25	6,40344	0,02016	0,01085	0,00043
		15 vizinl	20 vizinhos						
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	2,97129	0,01656	0,00401	0,0008	5	2,93922	0,01638	0,00395	0,00079
10	4,34928	0,01839	0,00623	0,00062	10	4,30805	0,0182	0,00615	0,00062
15	5,20319	0,01906	0,00796	0,00053	15	$5,\!15326$	0,01887	0,00786	0,00052
20	5,7969	0,0194	0,00941	0,00047	20	5,73323	0,01919	0,00929	0,00046
25	6,19133	0,01957	0,01047	0,00042	25	6,11941	0,01936	0,01032	0,00041
		25 vizinl	nos				30 vizinh	OS	
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	2,93327	0,01636	0,00394	0,00079	5	2,93418	0,01637	0,00394	0,00079
10	4,29156	0,01817	0,00612	0,00061	10	$4,\!28973$	0,01817	0,00612	0,00061
15	5,13402	0,01883	0,00782	0,00052	15	5,12944	0,01883	0,00782	0,00052
20	5,71215	0,01916	0,00924	0,00046	20	5,70482	0,01916	0,00923	0,00046
25	6,09559	0,01933	0,01027	0,00041	25	6,0878	0,01933	0,01026	0,00041

Fonte: Arthur Patricio Grava, 2016

A partir do R_n conjuntos gerados, as métricas descritas na seção 2.4 foram calculadas levando-se em consideração o procedimento especificado na sub-seção 1.2.1.

Os resultados obtidos calculando as métricas para os autores do conjunto W_{teste} podem ser encontrado na tabela 12, na qual variou-se o $ranking\ k$ de recomendações a ser

utilizados para calcular as métricas entre 5 e 25. Os resultados em negrito na tabela 12 ressaltam aqueles que foram os melhores resultados levando-se em consideração o k em questão.

Observa-se que os melhores resultados, para todas as métricas, foram encontrados utilizando 5 coautores, sendo que a cada vizinho a mais utilizado a métrica passa a oferecer um resultado pior, encontrando menos itens relevantes dentro do $top\ k$.

Tabela 13 – Métricas coletadas a partir da execução do algoritmo SRS verificando todos os autores do conjunto de recomendações.

			SRS - So	cial Recom	mend	er System			
		5 vizinho	OS		10 vizinhos				
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	8,12713	0,04481	0,01143	0,00229	5	6,90125	0,03839	0,00932	0,00186
10	$11,\!86255$	0,04979	$0,\!01783$	0,00178	10	10,10422	0,04265	0,01451	0,00145
15	$14,\!18739$	0,05163	$0,\!0227$	0,00151	15	12,10608	0,04423	0,01857	0,00124
20	$15,\!80642$	0,05254	$0,\!02664$	0,00133	20	13,48468	0,04501	0,02197	0,0011
25	16,91673	0,05303	$0,\!02958$	0,00118	25	14,42369	0,04542	0,02445	0,00098
		20 vizinhos							
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	6,69281	0,03729	0,00903	0,00181	5	6,62058	0,0369	0,00891	0,00178
10	9,79672	0,04142	0,01403	0,0014	10	9,70385	0,041	0,01386	0,00139
15	11,72015	0,04294	0,01792	0,00119	15	11,60768	0,0425	0,01771	0,00118
20	13,05748	0,04369	0,02119	0,00106	20	12,91404	0,04324	0,02092	0,00105
25	13,94593	0,04408	0,02358	0,00094	25	13,78392	0,04362	0,02326	0,00093
		25 vizinh	OS				30 vizinh	OS	
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	6,60716	0,03686	0,00887	0,00177	5	6,60923	0,03687	0,00888	0,00178
10	$9,\!6667$	0,04093	0,01378	0,00138	10	9,66257	0,04093	0,01378	0,00138
15	$11,\!56434$	0,04243	0,01762	0,00117	15	11,55402	0,04242	0,0176	0,00117
20	$12,\!86658$	0,04316	0,02081	0,00104	20	12,85007	0,04315	0,02079	0,00104
25	13,73027	0,04354	0,02313	0,00093	25	13,71272	0,04353	0,0231	0,00092

Fonte: Arthur Patricio Grava, 2016

Os resultados para o algoritmo SRS utilizando-se apenas os autores que receberam ao menos uma recomendação são apresentados na tabela 13. Pode-se observar que, para todas as métricas calculadas, os melhores resultados foram obtidos utilizando-se os 5 coautores mais próximos, sendo que ao aumentar o número de vizinhos de 10 até 25 estes resultados apenas pioraram, com exceção da vizinhança de tamanho 30, que apresentou resultados superiores quando comparado com o uso de 25 vizinhos.

Ao comparar os resultados entre si, observa-se que os resultados gerados pelo SRS tiveram um incremento de resultado relevante quando comparando a validação utilizando

os autores que receberam ao menos uma recomendação e todos os autores presentes em W_{teste} , obtendo resultados até duas vezes melhores.

5.1.3.2 SRS*

O SRS* além de utilizar os dados de coautoria, também faz uso do conjunto S de similaridades calculado para a execução do experimento detalhado na sub-seção 5.1.1. Sendo assim, a execução do SRS* seguiu o mesmo procedimento de execução descrito pela seção 1.2, onde, para cada uma das configurações de vizinhança executas, foi coletado conjuntos de recomendações, representados por R_n , sendo que n indica a quantidade de vizinhos utilizadas.

Tabela 14 – Métricas coletadas a partir da execução do algoritmo SRS* verificando todos os autores no conjunto de testes W_{teste} .

		SRS*	- Social Red	commender	Syste	em with Sin	nilarity		
		5 vizinho	os				10 vizinh	os	
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	5,48585	0,02993	0,00789	0,00158	5	5,21877	0,02842	0,00736	0,00147
10	8,08973	0,03339	0,01242	0,00124	10	7,6756	0,03169	0,01147	0,00115
15	9,72197	0,03468	0,01572	0,00105	15	9,23134	0,03291	0,01446	0,00096
20	10,89931	0,03534	0,01833	0,00092	20	10,32668	0,03353	0,01681	0,00084
25	11,7972	0,03573	0,02039	0,00082	25	11,23007	0,03392	0,01872	0,00075
15 vizinhos					20 vizinhos				
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	5,18212	0,02826	0,00728	0,00146	5	5,17708	0,02823	0,00727	0,00145
10	7,59681	0,03146	0,01131	0,00113	10	7,56932	0,0314	0,01124	0,00112
15	$9,\!12872$	0,03267	0,01423	0,00095	15	9,0962	0,0326	0,01415	0,00094
20	$10,\!20757$	0,03328	0,01654	0,00083	20	10,16955	0,0332	0,01643	0,00082
25	11,10042	0,03367	0,01841	0,00074	25	11,05003	0,03359	0,01826	0,00073
		25 vizinh	.OS				30 vizinh	.OS	
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	5,17021	0,02821	0,00727	0,00145	5	5,17296	0,02819	0,00727	0,00145
10	$7,\!56291$	0,03138	0,01123	0,00112	10	7,56291	0,03136	0,01123	0,00112
15	9,08154	0,03258	0,01412	0,00094	15	9,08795	0,03257	0,01413	0,00094
20	$10,\!16267$	0,03319	0,01641	0,00082	20	10,16405	0,03317	0,01641	0,00082
25	11,04316	0,03357	0,01824	0,00073	25	11,03445	0,03355	0,01823	0,00073

Fonte: Arthur Patricio Grava, 2016

Os resultados do SRS* obtidos calculando as métricas a partir de W_{teste} podem ser encontrado na tabela 14. Os resultados em negrito destacam os melhores resultados levando-se em consideração os diferentes valores de k.

Os melhores resultados para todas as métricas foram encontrados utilizando 5 coautores, sendo que ao incrementar o número de vizinhos considerados as métrica de avaliação passam atingem piores resultados, encontrando menos itens relevantes dentro do $top\ k$.

Tabela 15 – Comparação entre SRS* e SRS utilizando 5 vizinhos e as primeiras 25 recomendações a partir de W_{teste} .

Algoritmo	S@25 %	MRR	Recall	Prec.
SRS*	11,79	0,036	0,020	0,00082
SRS	7,51	0,024	0,013	0,00053

Fonte: Arthur Patricio Grava, 2016

Porém ao comparar estes resultados com os obtidos pelo SRS, pode-se observar que houve uma melhora no desempenho geral do algoritmo uma vez que todas as métricas coletadas obtiveram resultados superiores, por exemplo, para o R_5 o S@K do SRS* obteve um valor de 11,79% com k=25 enquanto o SRS um valor de 7,51%, ou seja, a chance de se recomendar um item relevante pelo SRS* foi mais de 50% maior que a produzida pelo SRS. Os valores de revocação e precisão tiveram um comportamento similar: considerando o mesmo k, o SRS* obteve 0,020 e 0,00082 respectivamente, enquanto o SRS apresentou 0,013 e 0,00053, sendo assim pode-se concluir que o SRS* consegue retornar mais recomendações relevantes a cada conjunto quando comparado ao SRS. Os valores de MRR também apresentaram o mesmo comportamento com o SRS* pontuando 0,036 e o SRS 0,024, indicando que o primeiro item relevante tem um melhor posicionamento a partir do SRS* quando comparado ao SRS. Estes resultados podem se observados na tabela 15.

Os resultados do SRS* avaliando apenas as recomendações aos autores que receberam ao menos uma recomendação podem ser encontrados na tabela 16.

Observa-se que os melhores resultados, para todas as métricas, foram encontrados utilizando 5 coautores, sendo que a cada vizinho a mais utilizado os valores destas métricas pioram, encontrando menos itens relevantes dentro do $top\ k$. Comparando os resultados entre si, pode-se observar que os resultados gerados pelo SRS tiveram um incremento de resultado relevante quando comparando os resultados utilizando apenas os autores presentes em $W_{treinamento}$ com o apenas os autores que receberam recomendações, por exemplo, o S@25 de 11,79% passou a 28,15%.

Tabela 16 – Métricas coletadas a partir da execução do algoritmo SRS* verificando todos os autores do conjunto de recomendações.

SRS* - Social Recommender System with Similarity									
5 vizinhos				10 vizinhos					
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	13,08815	0,0714	0,01882	0,00376	5	12,45095	0,06782	0,01756	0,00351
10	19,30051	0,07967	0,02964	0,00296	10	18,31248	0,07559	0,02735	0,00274
15	$23,\!19471$	0,08273	0,03751	$0,\!0025$	15	22,02415	0,07852	0,0345	0,0023
20	26,00361	0,08431	0,04373	0,00219	20	24,63741	0,07999	0,04009	0,002
25	$28,\!1458$	0,08525	0,04866	0,00195	25	26,79272	0,08093	0,04465	0,00179
15 vizinhos				20 vizinhos					
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	12,36352	0,06742	0,01738	0,00348	5	12,35149	0,06734	0,01735	0,00347
10	18,12449	0,07506	0,02699	0,0027	10	18,05891	0,07491	0,02683	0,00268
15	21,77933	0,07794	0,03396	0,00226	15	21,70173	0,07778	0,03375	0,00225
20	$24,\!35324$	0,07939	0,03947	0,00197	20	24,26253	0,07922	0,03919	0,00196
25	26,48341	0,08032	0,04391	0,00176	25	26,36319	0,08014	0,04356	0,00174
		25 vizinh	.OS		30 vizinhos				
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	12,3351	0,0673	0,01734	0,00347	5	12,34166	0,06726	0,01735	0,00347
10	18,04361	0,07488	0,0268	0,00268	10	18,04361	0,07483	0,0268	0,00268
15	21,66676	0,07773	0,03369	0,00225	15	21,68206	0,07769	0,03371	0,00225
20	24,24613	0,07918	0,03916	0,00196	20	24,24941	0,07914	0,03916	0,00196
25	26,34679	0,0801	0,04352	0,00174	25	26,32603	0,08005	0,04349	0,00174

Tabela 17 – Comparação entre SRS* e SRS utilizando 5 vizinhos e as primeiras 25 recomendações a partir de R_n .

Algoritmo	S@25 %	MRR	Recall	Prec.
SRS*	28,15	0,0853	0,0487	0,0019
SRS	16,92	0,0530	0,0295	0,0011

Fonte: Arthur Patricio Grava, 2016

Assim como observado ao comparar o comportamento das duas técnicas utilizando o conjunto W_{teste} , os resultados utilizando como ponto de partida os autores contidos em R_n também foram superiores, comparando-se especificamente o R_5 , em que o melhor resultado de ambos foi alcançado, observa-se que o SRS* obteve uma pontuação de 28, 15%, 0,0853, 0,0487 e 0,0019 em termos de S@K, MRR, revocação e precisão, respectivamente enquanto o SRS realizou 16,92, 0,0530, 0,0295 e 0,0011, superando os resultados do SRS em todas as métricas. Estes dados podem ser observados na tabela 17.

Portanto, a partir dos resultados apresentados nesta sub-seção, e também evidenciado por Golbeck (2006), ao se utilizar o peso da similaridade entre dois autores que possuem

uma relação, os resultados podem ser melhor ordenados, produzindo recomendações mais relevantes.

5.1.4 Experimento TSRS

Nesta sub-seção são descritos os experimentos realizados utilizando os algoritmos TSRS e TSRS*, iniciando pelo TSRS e, em seguida, finalizando com TSRS*.

5.1.4.1 TSRS

A execução da técnica TSRS tem como requerimento de entrada as relações de coautoria (C) existentes entre os autores do conjunto $W_{treinamento}$, assim como especificado na sub-seção 4.3.3, para isso foi aproveitado o cálculo utilizado pelo algoritmo 5, utilizando como peso na recomendação a força da relação entre eles.

Tabela 18 – Métricas coletadas a partir da execução do algoritmo TSRS verificando todos os autores no conjunto de testes W_{teste} .

TSRS - Trust Social Recommender System									
5 vizinhos				10 vizinhos					
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	4,51191	0,02416	0,00606	0,00121	5	2,73628	0,0157	0,00362	0,00072
10	7,02372	0,02748	0,00992	0,00099	10	4,05105	0,01743	0,00554	0,00055
15	$8,\!59228$	0,02872	0,01245	0,00083	15	4,90726	0,01811	0,00684	0,00046
20	$9,\!68212$	0,02933	0,01441	0,00072	20	5,54494	0,01847	0,00787	0,00039
25	$10,\!57222$	0,02972	0,01603	0,00064	25	6,09421	0,01871	0,00876	0,00035
15 vizinhos				20 vizinhos					
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	2,22732	0,01353	0,00319	0,00064	5	2,04179	0,01272	0,00304	0,00061
10	3,23058	0,01485	0,00469	0,00047	10	2,85814	0,0138	0,00439	0,00044
15	3,80871	0,01531	0,00561	0,00037	15	3,34327	0,01418	0,00523	0,00035
20	$4,\!25491$	0,01556	0,00636	0,00032	20	3,7203	0,01439	0,00591	0,0003
25	4,66034	0,01574	0,00704	0,00028	25	4,05792	0,01454	0,0065	0,00026
25 vizinhos				30 vizinhos					
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	1,96758	0,01245	0,00296	0,00059	5	1,8984	0,01205	0,00293	0,00059
10	2,67489	0,01339	0,00425	0,00042	10	2,55166	0,01291	0,0042	0,00042
15	3,09773	0,01372	0,00505	0,00034	15	2,94793	0,01322	0,00499	0,00033
20	3,45001	0,01392	0,00572	0,00029	20	3,28051	0,01341	0,00564	0,00028
25	3,74824	0,01405	0,00629	0,00025	25	3,56042	0,01353	0,00621	0,00025

Fonte: Arthur Patricio Grava, 2016

A partir do R_n conjuntos gerados, as métricas descritas na seção 2.4 foram calculadas levando-se em consideração o procedimento especificado na seção 1.2.

Os resultados obtidos calculando as métricas para as recomendações aos autores presentes em W_{teste} podem ser encontrado na tabela 18. Os valores em negrito na tabela 18 ressaltam aqueles que foram os melhores resultados levando-se em consideração o k em questão.

A partir destes resultados pode-se observar que os melhores resultados, levandose em consideração todas as métricas calculadas, foram encontrados utilizando, como configuração de vizinhança, 5 coautores, sendo que a cada vizinho a mais utilizado a métrica passa a oferecer um resultado pior, encontrando menos itens relevantes dentro do $top\ k$.

Tabela 19 – Métricas coletadas a partir da execução do algoritmo TSRS* verificando todos os autores do conjunto de recomendações.

		T	SRS - True	t Social Red	nmm	ender Suste	om			
						Chack Dysic				
	~~~~	5 vizinho				10 vizinhos				
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.	
5	4,73159	0,02534	0,00635	0,00127	5	2,86261	0,01643	0,00379	0,00076	
10	7,3657	0,02882	0,01041	0,00104	10	4,23808	0,01824	0,0058	0,00058	
15	9,01064	0,03012	0,01306	0,00087	15	5,13381	0,01894	0,00716	0,00048	
20	$10,\!15354$	0,03076	0,01511	0,00076	20	5,80094	0,01932	0,00823	0,00041	
25	11,08698	0,03117	0,01681	0,00067	25	6,37557	0,01957	0,00916	0,00037	
15 vizinhos						20 vizinhos				
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.	
5	2,32976	0,01415	0,00334	0,00067	5	2,13567	0,01331	0,00318	0,00064	
10	3,37916	0,01553	0,00491	0,00049	10	2,98956	0,01443	0,00459	0,00046	
15	3,98388	0,01601	0,00587	0,00039	15	3,497	0,01483	0,00547	0,00036	
20	$4,\!4506$	0,01627	0,00665	0,00033	20	3,89136	0,01506	0,00618	0,00031	
25	$4,\!87467$	0,01646	0,00737	0,00029	25	4,24451	0,01521	0,0068	0,00027	
		25 vizinh	.OS				30 vizinh	.OS		
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.	
5	2,05805	0,01302	0,00309	0,00062	5	1,98569	0,0126	0,00307	0,00061	
10	2,79789	0,014	0,00444	0,00044	10	2,66899	0,0135	0,00439	0,00044	
15	$3,\!24017$	0,01435	0,00528	0,00035	15	3,08348	0,01383	0,00522	0,00035	
20	$3,\!60865$	0,01456	0,00598	0,0003	20	3,43136	0,01403	0,0059	0,0003	
<b>25</b>	3,92059	0,0147	0,00658	0,00026	25	3,72413	0,01415	0,00649	0,00026	

Fonte: Arthur Patricio Grava, 2016

Os resultados para o experimento com o algoritmo TSRS e considerando apenas os autores que receberam uma ou mais recomendações são apresentados na tabela 19,

destacando-se em negrito aqueles que foram os melhores resultados para cada valor de k e de número de vizinhos.

Pode-se observar que, para todas as métricas calculadas, os melhores resultados foram obtidos através da configuração de vizinhança utilizando 5 coautores, sendo que ao aumentar o número de vizinhos, de 10 até 25, os resultados apenas pioraram, seguindo o mesmo comportamento encontrado ao partir de  $W_{teste}$ .

Ao comparar os resultados entre si, pode-se observar que os resultados gerados pelo TSRS não tiveram um incremento de resultado relevante quando comparando os autores presentes em  $W_{teste}$  e apenas os autores que receberam recomendações.

### 5.1.4.2 TSRS*

O TSRS* além de utilizar os dados de coautoria, também faz uso do conjunto S de similaridades calculado para a execução do experimento detalhado na sub-seção 5.1.1. Sendo assim, sua execução seguiu o mesmo procedimento de execução descrito pela seção 1.2.

Os resultados do TSRS* obtidos calculando as métricas a partir de  $W_{teste}$  podem ser encontrado na tabela 20. Os valores encontrados em negrito na tabela 14 ressaltam aqueles que foram os melhores resultados levando-se em consideração o k em questão.

A partir destes resultados pode-se observar que os melhores resultados, para todas as métricas, foram encontrados utilizando 5 coautores, sendo que a cada vizinho a mais utilizado a métrica passa a oferecer um resultado pior, encontrando menos itens relevantes dentro do  $top\ k$ .

Quando comparados os resultados obtidos a partir do TSRS* e TSRS, é possível observar que houve melhora no desempenho geral do algoritmo ao acrescentar a informação do coeficiente de similaridade entre os coautores, sendo que, para todas as métricas calculadas, os resultados foram superiores. Os valores de S@K, MRR, revocação e precisão para o TSRS* e TSRS são encontrados na tabela 21, evidenciando a diferença de valores resultante entre elas.

Os resultados do TSRS* originando o cálculo das métricas para os autores com recomendações em  $R_n$  estão presentes na tabela 22. Os valores destacados em negrito ressaltam aqueles que foram os melhores resultados levando-se em consideração o k em questão.

Tabela 20 – Métricas coletadas a partir da execução do algoritmo TSRS* verificando todos os autores no conjunto de testes  $W_{teste}$ .

		TSRS* - '	Trust Social	Recommen	nder S	System with	Similarit	y	
		5 vizinho	OS				10 vizinh	.os	
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	5,70574	0,03085	0,00809	0,00162	5	5,43087	0,02949	0,00757	0,00151
10	8,3733	0,03438	0,01274	0,00127	10	7,95093	0,03284	0,01175	0,00117
15	$10,\!05273$	0,0357	0,01607	0,00107	15	9,5511	0,0341	0,01478	0,00098
20	11,28504	$0,\!0364$	0,0188	0,00094	20	10,70187	0,03475	0,01725	0,00086
<b>25</b>	$12,\!18751$	0,03679	0,02084	0,00083	25	11,55899	0,03512	0,01912	0,00076
15 vizinhos						20 vizinh	.OS		
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	5,38644	0,02927	0,00751	0,0015	5	5,37957	0,0292	0,0075	0,0015
10	7,879	0,03257	0,0116	0,00116	10	7,83319	0,03245	0,01153	0,00115
15	9,44848	0,03381	0,01459	0,00097	15	9,39672	0,03369	0,01449	0,00097
20	10,59101	0,03445	0,017	0,00085	20	10,51954	0,03432	0,01685	0,00084
<b>25</b>	11,43667	0,03482	0,01881	0,00075	25	11,36658	0,03469	0,01866	0,00075
		25 vizinh	.OS				30 vizinh	.OS	
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	5,37728	0,02922	0,0075	0,0015	5	5,3704	0,02917	0,00748	0,0015
10	7,83182	0,03247	0,01153	0,00115	10	7,82678	0,03243	0,01152	0,00115
15	$9,\!38526$	0,0337	0,01446	0,00096	15	9,38068	0,03365	0,01445	0,00096
20	$10,\!51725$	0,03434	0,01685	0,00084	20	10,51084	0,03429	0,01682	0,00084
<b>25</b>	11,35879	0,0347	0,01864	0,00075	25	11,35101	0,03466	0,01861	0,00074

Tabela 21 – Comparação entre TSRS* e TSRS utilizando 5 vizinhos e as primeiras 25 recomendações a partir de  $W_{teste}$ .

Algoritmo	S@K %	MRR	Recall	Prec.
TSRS*	12,19	0,0368	0,0208	0,00083
TSRS	10,57	0,0297	0,0160	0,00064

Fonte: Arthur Patricio Grava, 2016

A partir destes resultados pode-se observar que os melhores resultados, para todas as métricas, foram encontrados utilizando 5 coautores, sendo que a cada vizinho a mais utilizado a métrica passa a oferecer um resultado pior, encontrando menos itens relevantes dentro do  $top\ k$ .

Comparando as duas abordagens de metrificação, pode-se observar que os resultados gerados pelo TSRS não tiveram um incremento de resultado relevante quando comparando os autores presentes em  $W_{teste}$  e os autores presentes em  $R_n$ .

Assim como observado ao comparar o comportamento das duas técnicas utilizando o conjunto  $W_{teste}$ , os resultados utilizando como ponto de partida os autores contidos em  $R_n$ 

Tabela 22 – Métricas coletadas a partir da execução do algoritmo TSRS* verificando todos os autores do conjunto de recomendações.

		TSRS* - '	Trust Social	Recommen	nder S	System with	Similarit	y		
		5 vizinho	OS				10 vizinh	.OS		
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.	
5	13,61277	0,0736	0,0193	0,00386	5	12,95699	0,07036	0,01807	0,00361	
10	19,97705	0,08203	0,03039	0,00304	10	18,96934	0,07834	0,02803	0,0028	
15	23,98382	0,08518	0,03834	0,00256	15	22,78704	0,08135	0,03525	0,00235	
20	26,92388	0,08684	0,04485	0,00224	20	25,53254	0,0829	0,04115	0,00206	
<b>25</b>	29,077	0,08778	0,04973	0,00199	25	27,57746	0,08379	0,04563	0,00182	
15 vizinhos						20 vizinhos				
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.	
5	12,85098	0,06983	0,01791	0,00358	5	12,83458	0,06967	0,01789	0,00358	
10	18,79775	0,07771	0,02769	0,00277	10	18,68845	0,07743	0,02752	0,00275	
15	$22,\!54222$	0,08066	0,0348	0,00232	15	22,41871	0,08037	0,03457	0,0023	
20	$25,\!26805$	0,08219	0,04057	0,00203	20	25,09755	0,08187	0,04019	0,00201	
<b>25</b>	27,28564	0,08308	0,04488	0,0018	25	27,11842	0,08276	0,04452	0,00178	
		25 vizinh	.OS				30 vizinh	OS		
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.	
5	12,82912	0,06971	0,01788	0,00358	5	12,81272	0,0696	0,01785	0,00357	
10	18,68517	0,07748	0,02752	0,00275	10	18,67315	0,07737	0,02749	0,00275	
15	22,39139	0,0804	0,03451	0,0023	15	22,38046	0,08029	0,03448	0,0023	
20	25,09208	0,08192	0,04019	0,00201	20	25,07678	0,08181	0,04014	0,00201	
25	27,09984	0,0828	0,04448	0,00178	25	27,08126	0,08268	0,04439	0,00178	

Tabela 23 – Comparação entre TSRS* e TSRS utilizando 5 vizinhos e as primeiras 25 recomendações a partir de  $R_5$ .

Algoritmo	S@K %	MRR	Recall	Prec.
TSRS*	29,08	0,0878	0,0497	0,0020
TSRS	11,09	0,0312	0,0168	0,0007

Fonte: Arthur Patricio Grava, 2016

também foram superiores, comparando-se especificamente o  $R_5$ , onde o melhor resultado de ambos foi alcançado. Os valores de S@K, MRR, revocação e precisão para o TSRS* e TSRS são encontrados na tabela 23, evidenciando a diferença de valores resultante entre elas.

Portanto, a partir dos resultados mostrados por meio deste experimento, e também evidenciado por Golbeck (2006), ao se utilizar o peso da similaridade entre dois autores que possuem uma relação, os resultados são melhor ordenados e produzem recomendações mais relevantes.

## 5.1.5 Experimento Time-TSRS

Para a realização do experimento utilizando o algoritmo Time-TRS foi necessário realizar o cálculo da confiança entre os coautores existentes em  $W_{treinamento}$  novamente, porém adicionando o peso do ano em que aquela relação de coautoria foi realizada, da mesma forma como apresentado pela equação 16.

Tabela 24 – Métricas coletadas a partir da execução do algoritmo Time-TSRS verificando todos os autores no conjunto de testes  $W_{teste}$ 

		TimeTS	RS - Time	Aware Tru	st Soc	cial Recomme	ender Syste	m		
		5 vizinho	OS				10 vizinh	ios		
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.	
5	9,38068	0,05075	0,01152	0,0023	5	9,3729	0,05073	0,01145	0,00229	
10	$9,\!38068$	0,05075	0,01152	0,00115	10	$14,\!10149$	$0,\!05699$	0,01754	0,00175	
15	$9,\!38068$	0,05075	0,01152	0,00077	15	14,10149	$0,\!05699$	0,01754	0,00117	
20	$9,\!38068$	0,05075	0,01152	0,00058	20	14,10149	$0,\!05699$	0,01754	0,00088	
<b>25</b>	9,38068	0,05075	0,01152	0,00046	25	14,10149	0,05699	0,01754	0,0007	
15 vizinhos						20 vizinhos				
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.	
5	9,36419	0,05066	0,01142	0,00228	5	9,37061	0,05066	0,01145	0,00229	
10	14,08912	0,05692	0,01748	0,00175	10	14,09782	0,05693	0,0175	0,00175	
15	$17,\!35131$	0,05948	0,02201	0,00147	15	17,33711	0,05947	0,02197	0,00146	
20	17,35131	0,05948	0,02201	0,0011	20	19,73531	0,06082	0,02554	0,00128	
<b>25</b>	17,35131	0,05948	0,02201	0,00088	25	19,73531	0,06082	0,02554	0,00102	
		25 vizinh	.OS				30 vizinh	ios		
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.	
5	9,37198	0,05071	0,01146	0,00229	5	9,36465	0,05065	0,01142	0,00228	
10	14,09828	0,05697	$0,\!01754$	0,00175	10	14,08316	0,05691	0,01747	0,00175	
15	$17,\!32474$	$0,\!0595$	0,02196	0,00146	15	$17,\!31695$	0,05945	0,02194	0,00146	
20	19,71652	0,06085	0,02551	0,00128	20	19,6982	0,06079	0,02545	0,00127	
<b>25</b>	21,63279	0,06169	0,02852	0,00114	<b>25</b>	21,6163	0,06163	0,02849	0,00114	

Fonte: Arthur Patricio Grava, 2016

Os resultados obtidos calculando as métricas a partir das recomendações aos autores presentes em  $W_{teste}$  podem ser encontrado na tabela 24, onde variou-se o  $ranking\ k$  de recomendações a ser utilizados para calcular as métricas entre 5 e 25. Os resultados em negrito na tabela 24 ressaltam aqueles que foram os melhores resultados levando-se em consideração o k em questão.

Os resultados evidenciam que houve uma variação considerável nas métricas coletadas para as diferentes configurações, por exemplo, ao se utilizar apenas as 5 primeiras melhores recomendações (k = 5), os melhores resultados foram obtidos utilizando-se os 5 melhores vizinhos, porém a diferença entre o melhor e o pior resultado não foi muito relevante, considerando o S@K a diferença de resultado foi de 0,01608.

Porém ao verificar as 10 primeiras melhores recomendações os resultados apresentam uma diferenciação maior, sendo que para os valores de precisão e revocação a melhor configuração teve um empate entre utilizar  $10 \ e \ 25 \ vizinhos$ , porém considerando os valores de S@K e MRR a melhor configuração é com  $10 \ vizinhos$ .

Ao considerar as 15 primeiras posições a configuração com 15 obteve os melhores resultados, exceto pela métrica MRR, onde o melhor resultado foi apresentado pela configuração com 25 vizinhos. As 20 primeiras posições tiveram um resultado similar com a configuração de 20 vizinhos apresentando os melhores resultados, exceto pelo valor do MRR, onde 25 vizinhos tiveram um desempenho superior.

Entretanto ao observar os resultados para as primeiras 25 recomendações os melhores resultados foram obtidos utilizando a configuração de 25 vizinhos. Assim, dado que o contexto de resultados não apresenta uma mudança significativa ao se considerar as diferentes k recomendações, pode-se dizer que a melhor configuração disponível é a de 25 vizinhos, visto que ela possui a maioria dos resultados positivos e seus resultados inferiores também se apresentam próximos aos demais.

Os resultados para este experimento são apresentados na tabela 25, destacando-se em negrito aqueles que obtiveram os melhores resultados.

Os resultados gerados a partir dos autores presentes em  $R_n$  não apresentaram a mesma diversidade encontrada ao se utilizar os autores de  $W_{teste}$  como entrada, os melhores resultados para os 5 melhores itens recomendados foram obtidos utilizando os 5 primeiro vizinhos, exceto pelo MRR, que foi obtido utilizando 25 vizinhos. Os resultados, ao se considerar os 20 melhores itens recomendados apresentou melhores S@K e MRR utilizando 25 vizinhos, enquanto que os melhores valores de revocação e precisão foram encontrados a partir do de 30. Para o restante dos k melhores itens a configuração de 25 apresentou melhores resultados que o restante.

Assim como apresentado na avaliação utilizando o conjunto  $W_{teste}$  os resultados mais relevantes foram obtidos utilizando 25 vizinhos como configuração ideal de entrada, ao passo que para valores de k pequenos o resultado para esse tamanho de vizinhança é próximo aos resultados obtidos a partir de 5 vizinhos.

Ao comparar o resultado das duas abordagens de avaliação, pode-se dizer que os resultados gerados não tiveram uma evolução relevante como no caso do SRS, por exemplo,

Tabela 25 – Métricas coletadas a partir da execução do algoritmo Time-TSRS verificando todos os autores do conjunto de recomendações.

		TimeTSR	S - Time A	Aware Trust	Soci	al Recommo	ender Syst	tem	
		5 vizinho	OS				10 vizir	nhos	
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	12,40764	0,06737	0,01752	0,0035	5	12,31065	0,06702	0,01734	0,00347
10	$12,\!40764$	0,06737	0,01752	0,00175	10	17,95827	0,07454	0,02684	0,00268
15	$12,\!40764$	0,06737	0,01752	0,00117	15	17,95827	0,07454	0,02684	0,00179
20	$12,\!40764$	0,06737	0,01752	0,00088	20	17,95827	0,07454	0,02684	0,00134
25	12,40764	0,06737	0,01752	0,0007	25	17,95827	0,07454	0,02684	0,00107
15 vizinhos							20 vizir	nhos	
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	12,35295	0,06719	0,01738	0,00348	5	12,38597	0,06723	0,01748	0,0035
10	17,982	0,07468	0,02683	0,00268	10	18,0336	0,07473	0,02685	0,00269
15	21,58129	0,07751	0,03374	0,00225	15	21,64321	0,07758	0,0337	0,00225
20	21,58129	0,07751	0,03374	0,00169	20	24,22194	0,07903	0,03922	0,00196
<b>25</b>	21,58129	0,07751	0,03374	0,00135	25	24,22194	0,07903	0,03922	0,00157
		25 vizinh	.OS				30 vizir	nhos	
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.
5	12,38804	0,06739	0,01747	0,00349	5	12,32716	0,06702	0,01742	0,00348
10	18,11202	0,07498	0,02696	$0,\!0027$	10	17,99335	0,07455	0,02682	0,00268
15	21,68655	0,0778	0,03386	0,00226	15	21,64011	0,07743	0,03385	0,00226
20	$24,\!28179$	$0,\!07926$	0,03926	0,00196	20	24,25496	0,0789	0,03931	0,00197
25	26,25067	0,08012	0,04359	0,00174	25	26,21765	0,07977	0,04356	0,00174

o que indica que não existe uma diferença relevante entre os autores em  $R_n$  e aqueles em  $W_{teste}$ . Isto indica que a técnica é capaz de realizar recomendações para grande parte dos autores.

# 5.1.6 Experimento regressão linear

Para execução do experimento utilizando regressão linear para definir o peso dos resultados de cada um dos algoritmos na relevância da recomendação, foi gerado um arquivo contendo as pontuações (scores) de todas as recomendações dos algoritmos anteriores seguindo o formato definido na seção 4.4. Foi utilizado o algoritmo de regressão linear disponível na ferramenta Weka  1 , desenvolvida em Java.

Uma pré-seleção foi realizada no arquivo de recomendações separando-o em quatro diferentes arquivos com um dos arquivos contento apenas as linhas que possuíam 3 ou

¹ http://www.cs.waikato.ac.nz/ml/weka/

Tabela 26 – Pesos encontrados para cada um dos algoritmos variando o número mínimo de *scores* necessários.

Algoritmo	3 scores	4 scores	5 scores	6 scores
UCF	-0,0766	-0,1206	-0,0932	-1,2651
TRS	0,6342	0,6082	0,45	0,502
SRS	0,0	0,0	0,0398	0,0
SRS*	-0,0485	-0,0226	0,0	0,9727
TSRS	0,0324	0,0879	0,0	1,1044
TSRS*	0,0	-0,1049	0,0315	0,0
Time-TSRS	-0,0666	0,0887	-0,0272	-3,5864
Termo regulador	0,1218	0,0	0,043	2,724

mais respostas de algoritmos (isto é, três ou mais algoritmos produziram um *score* positivo para a recomendação do respectivo artigo a um dado autor), outro contendo 4 ou mais respostas, outro contendo 5 ou mais respostas e um arquivo contendo 6 ou mais respostas.

A partir de cada um desses arquivos os pesos, que podem ser observados na tabela 26, foram calculados pela regressão linear e utilizados para determinar a nova ordenação das recomendações para, assim, calcular as métricas discutidas na seção 2.4.

Tabela 27 – Métricas coletadas a partir da hibridização utilizando regressão verificando todos os autores no conjunto de testes  $W_{teste}$ 

	HSRS - Hybrid social recommender system											
	3 -		4 va	lores pree	nchidos							
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.			
5	26,4979	0,13424	0,04889	0,00978	5	29,26212	0,14849	0,05462	0,01092			
10	$38,\!82651$	0,15081	0,0757	0,00757	10	43,02874	0,16699	0,08712	0,00871			
15	$45,\!17956$	0,15584	0,08959	0,00597	15	49,82569	0,17239	0,10397	0,00693			
20	49,03545	0,15802	0,09843	0,00492	20	53,79337	0,17463	0,11455	0,00573			
25	51,94032	0,15929	0,10617	0,00425	25	56,42703	0,17578	0,12238	0,00489			
	5	valores pree	nchidos			6 va	lores pree	nchidos				
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.			
5	29,68084	0,15059	0,0556	0,01112	5	17,69123	0,06761	0,03263	0,00653			
10	43,7246	$0,\!16947$	0,08894	0,00889	10	28,4215	0,08199	0,05603	0,0056			
15	$50,\!59394$	$0,\!17492$	0,10618	0,00708	15	34,14785	0,08651	0,06836	0,00456			
20	$54,\!58681$	$0,\!17717$	0,11691	0,00585	20	37,93274	0,08864	0,07678	0,00384			
25	57,21635	0,17833	$0,\!12477$	0,00499	25	41,71901	0,09029	0,08574	0,00343			

Fonte: Arthur Patricio Grava, 2016

Os resultados obtidos calculando as métricas avaliando as recomendações geradas para o  $W_{teste}$  podem ser encontrado na tabela 27, onde variou-se o ranking k de recomendações a ser utilizados para calcular as métricas entre 5 e 25. Os resultados

em negrito na tabela 27 ressaltam aqueles que foram os melhores resultados levando-se em consideração o k em questão, comparando-se as diferentes configurações de vizinhos utilizadas.

A partir dos resultados apresentados pode-se observar que os valores mais relevantes foram obtidos quando rodando com os pesos definidos a partir das entradas que possuíam, ao menos, 5 valores (isto é, cinco algoritmos de recomendação diferentes produziram *scores* positivos para a respectiva dupla: autor e artigo). Enquanto os valores menos expressivos foram encontrados utilizando como entrada o arquivo que possuía, ao menos 6 algoritmos com *scores* definidos para determinada recomendação.

A diferença entre o melhor e o pior resultado foi expressiva, enquanto o melhor resultado apresentou um S@5 de 29,84% o pior resultado apresentou 17,69%, ou seja o pior resultado atingiu um valor de S@5 equivalente a menos de 60% do resultado atingido pelo melhor..

Tabela 28 – Métricas coletadas a partir a partir da hibridização utilizando regressão verificando todos os autores do conjunto de recomendações.

	HSRS - Hybrid social recommender system											
	3 -	valores pree	nchidos		4 valores preenchidos							
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.			
5	26,4996	0,13425	0,04889	0,00978	5	29,264	0,1485	0,05463	0,01093			
10	38,829	0,15082	0,07571	0,00757	10	43,0315	0,167	0,08713	0,00871			
15	45,18245	0,15585	0,08959	0,00597	15	49,82889	0,1724	0,10398	0,00693			
20	49,0386	0,15803	0,09844	0,00492	20	53,79682	0,17464	0,11456	0,00573			
<b>25</b>	51,94365	0,1593	0,10617	0,00425	25	56,43065	0,17579	0,12238	0,0049			
	5	valores pree	nchidos			6 va	lores pree	nchidos	0,00573 0,0049 <b>Prec.</b> 0,00653			
K	S@K %	MRR	Recall	Prec.	K	S@K %	MRR	Recall	Prec.			
5	29,68274	0,1506	0,0556	0,01112	5	17,69236	0,06761	0,03263	0,00653			
10	43,72741	0,16948	0,08895	0,00889	10	28,42332	0,08199	0,05603	0,0056			
15	50,59718	$0,\!17493$	0,10618	0,00708	15	34,15004	0,08651	0,06836	0,00456			
20	$54,\!59031$	0,17718	$0,\!11692$	0,00585	20	37,93517	0,08864	0,07679	0,00384			
25	57,22002	0,17834	0,12478	0,00499	25	41,72168	0,0903	0,08574	0,00343			

Fonte: Arthur Patricio Grava, 2016

Os resultados para este experimento são apresentados na tabela 28, na qual variou-se o  $ranking\ k$  de recomendações a ser utilizados para calcular as métricas entre 5 e 25. Os resultados em negrito na tabela 28 ressaltam aqueles que foram os melhores resultados para cada k.

Observar que os valores mais relevantes foram obtidos quando rodando com os pesos definidos a partir das entradas que possuíam, ao menos, 5 valores a ser utilizados. Enquanto

os valores menos expressivos foram encontrados utilizando como entrada o arquivo que possuía, ao menos 6 algoritmos com *scores* definidos para determinada recomendação.

A diferença entre o melhor e o pior resultado foi expressiva, para as 5 recomendações mais relevantes da recomendação, enquanto o melhor resultado apresentou um S@5 de 29.68% o pior resultado apresentou 17.69%. Novamente, o pior resultado atingiu um valor para S@5 inferior a 60% do alcançado pelo melhor.

Porém ao comparar os resultados das duas abordagens de avaliação, pode-se observar que os resultados gerados pela regressão linear são bem próximos e não apresentam uma variação relevante.

# 5.2 Considerações

Nesta seção são sumarizados e discutidos alguns dos resultados apresentados na seção anterior.

Para todas os valores das métricas coletadas, escolheu-se sempre, para cada um dos algoritmos, a configuração de vizinhança que apresentou os resultados mais expressivos, variando também o método de avaliação entre a utilização dos autores encontrados em Wtest e os autores que possuíam ao menos uma recomendação em  $R_n$ , construindo, para cada uma das métricas, uma ilustração apresentando os resultados, nos quais, para cada uma das configurações selecionadas, mostrou-se a qualidade da recomendação para os primeiros k artigos da lista de recomendações (utilizando-se os seguintes valores para k: 5, 10, 15, 20 e 25).

A partir das recomendações a autores presentes em  $W_{teste}$  (figura 9), é possível constatar que, para valores de k pequenos, as técnicas TRS e HSRS apresentaram resultados parecidos, para os quais, utilizando-se as 25 primeiras recomendações (k = 25), o HSRS apresentou o melhor resultado em termos de S@K, com uma pontuação de 57,21%, significando que, para cada 100 autores encontrados em  $W_{teste}$ , na média, mais de 57 destes autores receberam uma recomendação correta entre os 25 primeiros elementos apresentados. Para k igual a 5, esses valores caem, com o HSRS apresentando uma pontuação de 29,68%.

Em termos de MRR o comportamento encontrado é bastante similar. Os algoritmos HSRS e TRS se destacaram novamente, porém com o TRS apresentando os valores mais positivos, obtendo uma pontuação de 0,17991 para k de valor 25, com uma diferença de apenas 0,00158 para o resultado do HSRS. A pequena diferença se mantém para o menor

Figura 9 – Ilustração dos resultados para todas as métricas e para os melhores resultados encontrados em cada uma das execuções dos algoritmos para cada K, utilizandose os autores de  $W_{teste}$ .

0.000

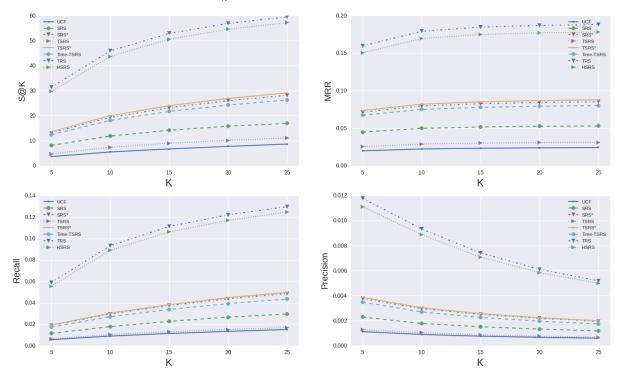
0.00

k do conjunto, onde a pontuação obtida pelo TRS é de 0,15326, com uma diferença de 0,0021 para o HSRS. Este resultado indica que, para k igual a 25 o primeiro item relevante é encontrado aproximadamente na posição de número 6, enquanto que para k de valor 5 este item é encontrado aproximadamente na posição 7 (lembrando que significa que para alguns autores o item foi encontrado em posições melhores e para outros o item não foi encontrado, então, é como se na média o primeiro item relevante estivesse na sétima posição, apesar de no teste com k=5 a lista de recomendações só possuir cinco itens).

A revocação e a precisão também apresentaram seus melhores resultados com os algoritmos TRS e HSRS, onde para um k igual a 25 elas apresentaram, respectivamente, as pontuações 0,12409 e 0,12477 de revocação e 0,00496 e 0,00499 de precisão, significando que para os artigos relevantes contidos em  $W_{teste}$  12,47% deles foram encontrados nos conjuntos de recomendação, enquanto a precisão regulariza esse número pela quantidade de recomendações que foram oferecidas ao autor. Para um k com valor 5 é possível notar que a revocação é de 0,05634 e precisão 0,01127 para o TRS, que apresentou os melhores resultados para este k, ou seja, o resultado passa a ser de 5,63% de artigos relevantes encontrados em  $W_{teste}$  por meio das recomendações.

Deve-se salientar que os resultados da revocação e precisão levam em consideração a existência de mais de 2 milhões de artigos possíveis à serem recomendados como citações para cada autor, esta diversidade apresentada, e não encontrada em outros conjuntos, justifica valores de revocação e precisão menores, sendo que, ao escolher recomendações aleatoriamente, a revocação seria de aproximadamente 0,0000033.

Figura 10 – Ilustração dos resultados para todas as métricas e para os melhores resultados encontrados em cada uma das execuções dos algoritmos para cada K, utilizandos e os autores de  $R_n$ .



Fonte: Arthur Patricio Grava, 2016

Considerando apenas os autores que possuíam uma ou mais recomendações em  $R_n$  e tendo esses resultados para as métricas ilustrados na figura 10, pode-se constatar que, para valores de k pequenos, as técnicas TRS e HSRS, assim como na utilização de  $W_{teste}$  apresentaram resultados parecidos entre si e melhores que os demais. Por exemplo, utilizando-se as 25 primeiras recomendações (k=25), o TRS apresentou o melhor resultado em termos de S@K, com uma pontuação de 59,51%, indicando que, para cada 100 autores encontrados em  $W_{teste}$ , em média, mais de 59 receberam uma recomendação relevante entre os 25 primeiros elementos apresentados. Com k de valor 5, essa pontuação diminui, tendo o TRS pontuado 31,42%.

Considerando o MRR o comportamento encontrado é similar, sendo que o TRS apresentou os valores mais positivos, obtendo uma pontuação de 0,18839 para k de valor 25, com uma diferença de apenas 0,01005 para o resultado do HSRS. Esse comportamento se mantém para o menor k do conjunto, para o qual a pontuação obtida pelo TRS é de 0,16031, com uma diferença de 0,0125 para o HSRS. Este resultado indica que, para k igual a 25 o primeiro item relevante é encontrado aproximadamente na posição de número 5, enquanto que para k de valor 5 este item é encontrado aproximadamente na posição 6.

A revocação e a precisão também apresentaram seus melhores resultados com os algoritmos TRS e HSRS. Para k igual a 25 eles apresentaram, respectivamente, as pontuações 0.12979 e 0.12477 de revocação e 0.00519 e 0.00499 de precisão, significando que 12.98% dos artigos relevantes aos autores foram encontrados nos conjuntos de recomendação. Para um k com valor 5 é possível notar que a revocação é de 0.05893 e a precisão é de 0.01179 para o TRS, que apresentou os melhores resultados para este k.

As considerações apresentadas validam a hipótese de que realizar a hibridização dos algoritmos por meio da regressão linear foi capaz de oferecer resultados melhores, no caso geral em que  $W_{teste}$  é utilizado como origem de autores a serem avaliados, quando utilizados valores de k maiores.

Porém, o maior ganho encontrado é encontrado ao observar o comportamento das recomendações que utilizam a informação de período em que as relações entre autores são criadas, para as quais, para o caso geral utilizando  $W_{teste}$ , o Time-TSRS mostrou que consegue gerar recomendações para uma quantidade maior de autores do que quando comparado com as variações TSRS*, SRS* e UCF, uma vez que seus resultados, ao ser avaliado perante os autores em  $R_n$  não sofrem alterações relevantes, enquanto que os algoritmos citados anteriormente melhoram seus resultados chegando a obter melhores resultados que o Time-TSRS, nos casos dos algoritmos TSRS* e SRS*, como pode ser observado na tabela 10.

Outro fato importante de ser mencionado é que os resultados obtidos corroboram com o levantamento realizado na revisão sistemática, onde os algoritmos baseados em redes sociais possuem melhores resultados que o filtro colaborativo baseado em usuários e a adição de contexto e confiança fazem esses resultados melhorarem ainda mais.

### 6 Conclusão

Auxiliar pesquisadores a encontrar artigos relevantes para suas pesquisas é uma atividade muito importante e desafiadora. Este trabalho apresentou uma comparação de algumas técnicas de recomendação presentes na literatura com duas técnicas propostas.

Neste capítulo são apresentadas as principais contribuições deste trabalho e também oportunidades de trabalhos futuros.

### 6.1 Contribuições

O principal objetivo deste trabalho foi desenvolver um sistema de recomendação de artigos utilizando dados sociais e informações bibliométricas. Este objetivo foi alcançado, pois foi abordada uma perspectiva diferente das soluções correlatas, caracterizando o avanço no contexto de recomendações de artigos.

Para obtenção deste objetivo foi realizada uma revisão sistemática sobre sistemas que se utilizam de informações sociais para realizar recomendações e, com base nesta revisão, foram definidas as técnicas a serem utilizadas para comparar os resultados e qual a maneira mais adequada de se avaliar um conjunto de recomendações destinadas a um autor. Além disso, a revisão auxiliou na elaboração das técnicas desenvolvidas ao embasar as hipóteses criadas.

Por meio dos resultados atingidos, acredita-se que a principal contribuição deste trabalho são as duas estratégias de recomendação desenvolvidas, que apresentaram uma abordagem ao tema de recomendação na área acadêmica de uma maneira diferente e original, atingindo resultados satisfatórios.

Adicionalmente, acredita-se que a comparação realizada entre as diferentes técnicas para o problema de recomendação de artigos também é uma contribuição que pode auxiliar outros pesquisadores na escolha de uma técnica que pretenda usar ou mesmo para auxiliar pesquisadores no desenvolvimento de novas técnicas ou extensão das existentes.

A disponibilização em uma plataforma online com os dados utilizados, após passar pelas fases de extração e normalização das informações, irão possibilitar que trabalhos futuros sejam desenvolvidos nesta área, sendo que estes possíveis trabalhos serão auxiliados pela diversidade de algoritmos implementados e testados no conjunto de dados utilizado.

#### 6.2 Trabalhos futuros

Durante o desenvolvimento e a conclusão deste trabalho algumas oportunidades de aperfeiçoamento e continuação do projeto foram identificadas.

A partir dos estudos realizados existe a oportunidade de coletar dados sobre o perfil de cada um dos autores e utilizá-los como contexto para auxiliar e melhorar o cálculo de similaridade entre eles, além de realizar o mesmo procedimento para os artigos publicados e citados, de maneira a possibilitar a implementação de uma técnica que tem por objetivo transferir conhecimento entre contextos distintos, trazendo melhorias no enfrentamento da problemática da esparsidade dos dados (CREMONESI; QUADRANA, 2014).

Outra possibilidade é utilizar métodos de fatoração de matrizes para atacar o problema mapeando as informações de contexto temporal, perfil do autor e informações dos artigos como características latentes de entrada para o algoritmo.

Disponibilizar este trabalho em uma ferramenta online para poder auxiliar definitivamente pesquisadores científicos possibilitando também uma validação da solução por meio de *feedback* real.

Além disso, também é possível utilizar de outras técnicas para realizar a hibridização dos algoritmos desenvolvidos neste trabalho e mover sua execução para sistemas de processamento distribuído, como  $Hadoop^1$  e  $Spark^2$ , desta maneira contribuindo para que o processo de calcular recomendações, que depende de um tempo de processamento elevado em conjuntos de dados completos, tendo em vista, por exemplo, o cálculo de similaridade entre todos os autores, seja otimizado e escalável de maneira simples e eficiente.

http://hadoop.apache.org/

² http://spark.apache.org/

### Referências³

- AKBARI, F.; TAJFAR, A.; NEJAD, A. Graph-based friend recommendation in social networks using artificial bee colony. In: *Dependable, Autonomic and Secure Computing (DASC), 2013 IEEE 11th International Conference on.* [S.l.: s.n.], 2013. p. 464–468. Citado na página 34.
- CHEN, C. et al. Recommender system based on social trust relationships. In: e-Business Engineering (ICEBE), 2013 IEEE 10th International Conference on. [S.l.: s.n.], 2013. p. 32–37. Citado 3 vezes nas páginas 34, 37 e 39.
- CHIN, A.; XU, B.; WANG, H. Who should i add as a "friend"?: A study of friend recommendations using proximity and homophily. In: *Proceedings of the 4th International Workshop on Modeling Social Media*. New York, NY, USA: ACM, 2013. (MSM '13), p. 7:1–7:7. ISBN 978-1-4503-2007-8. Disponível em: <a href="http://doi.acm.org/10.1145/2463656.2463663">http://doi.acm.org/10.1145/2463656.2463663</a>. Citado 2 vezes nas páginas 17 e 33.
- CHIU, Y.-S.; LIN, K.-H.; CHEN, J.-S. A social network-based serendipity recommender system. In: *Intelligent Signal Processing and Communications Systems (ISPACS)*, 2011 International Symposium on. [S.l.: s.n.], 2011. p. 1–5. Citado na página 33.
- CHRISTENSEN, I.; SCHIAFFINO, S. Matrix factorization in social group recommender systems. In: Artificial Intelligence (MICAI), 2013 12th Mexican International Conference on. [S.l.: s.n.], 2013. p. 10–16. Citado na página 34.
- CRASWELL, N. Encyclopedia of database systems. In: _____. Boston, MA: Springer US, 2009. cap. Mean Reciprocal Rank, p. 1703–1703. ISBN 978-0-387-39940-9. Disponível em:  $\frac{\text{http:}}{\text{dx.doi.org}}$  Disponível em:  $\frac{\text{citado na página 28.}}{\text{citado na página 28.}}$
- CREMONESI, P.; KOREN, Y.; TURRIN, R. Performance of recommender algorithms on top-n recommendation tasks. In: *Proceedings of the Fourth ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2010. (RecSys '10), p. 39–46. ISBN 978-1-60558-906-0. Disponível em: <a href="http://doi.acm.org/10.1145/1864708.1864721">http://doi.acm.org/10.1145/1864708.1864721</a>. Citado na página 29.
- CREMONESI, P.; QUADRANA, M. Cross-domain recommendations without overlapping data: Myth or reality? In: *Proceedings of the 8th ACM Conference on Recommender Systems.* New York, NY, USA: ACM, 2014. (RecSys '14), p. 297–300. ISBN 978-1-4503-2668-1. Disponível em: <a href="http://doi.acm.org/10.1145/2645710.2645769">http://doi.acm.org/10.1145/2645710.2645769</a>. Citado na página 86.
- DWIVEDI, P.; BHARADWAJ, K. K. e-learning recommender system for learners in online social networks through association retrieval. In: *Proceedings of the CUBE International Information Technology Conference*. New York, NY, USA: ACM, 2012. (CUBE '12), p. 676–681. ISBN 978-1-4503-1185-4. Disponível em: <a href="http://doi.acm.org/10.1145/2381716.2381846">http://doi.acm.org/10.1145/2381716.2381846</a>. Citado na página 33.
- FEITOSA, R. et al. Social recommendation in location-based social network using text mining. In: *Intelligent Systems Modelling Simulation (ISMS)*, 2013 4th International Conference on. [S.l.: s.n.], 2013. p. 67–72. ISSN 2166-0662. Citado na página 33.

³ De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.

- FERREIRA, A. P. G.; SILVA, T. H.; LOUREIRO, A. A. F. You are your check-in: Understanding the behavior of tourists and residents using data from foursquare. In: *Proceedings of the 20th Brazilian Symposium on Multimedia and the Web.* New York, NY, USA: ACM, 2014. (WebMedia '14), p. 103–110. ISBN 978-1-4503-3230-9. Disponível em: <a href="http://doi.acm.org/10.1145/2664551.2664562">http://doi.acm.org/10.1145/2664551.2664562</a>. Citado na página 34.
- FORSATI, R. et al. Matrix factorization with explicit trust and distrust side information for improved social recommendation. *ACM Trans. Inf. Syst.*, ACM, New York, NY, USA, v. 32, n. 4, p. 17:1–17:38, out. 2014. ISSN 1046-8188. Disponível em: <a href="http://doi.acm.org/10.1145/2641564">http://doi.acm.org/10.1145/2641564</a>>. Citado 3 vezes nas páginas 34, 37 e 39.
- FU, C. H.; CHANG, C. S.; LEE, D. S. A proximity measure for link prediction in social user-item networks. In: *Information Reuse and Integration (IRI)*, 2014 IEEE 15th International Conference on. [S.l.: s.n.], 2014. p. 710–717. Citado na página 41.
- GAO, Y.; XU, B.; CAI, H. Information recommendation method research based on trust network and collaborative filtering. 2013 IEEE 10th International Conference on e-Business Engineering, IEEE Computer Society, Los Alamitos, CA, USA, v. 0, p. 386–391, 2011. Citado 2 vezes nas páginas 25 e 52.
- GAO, Y.; XU, B.; CAI, H. Information recommendation method research based on trust network and collaborative filtering. In: *e-Business Engineering (ICEBE)*, 2011 IEEE 8th International Conference on. [S.l.: s.n.], 2011. p. 386–391. Citado 3 vezes nas páginas 33, 36 e 37.
- GEYER, W. et al. Recommending topics for self-descriptions in online user profiles. In: *Proceedings of the 2008 ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2008. (RecSys '08), p. 59–66. ISBN 978-1-60558-093-7. Disponível em: <a href="http://doi.acm.org/10.1145/1454008.1454019">http://doi.acm.org/10.1145/1454008.1454019</a>. Citado na página 33.
- GOLBECK, J. Generating predictive movie recommendations from trust in social networks. In: *Proceedings of the 4th International Conference on Trust Management*. Berlin, Heidelberg: Springer-Verlag, 2006. (iTrust'06), p. 93–104. ISBN 3-540-34295-8, 978-3-540-34295-3. Disponível em: <a href="http://dx.doi.org/10.1007/11755593_8">http://dx.doi.org/10.1007/11755593_8</a>. Citado 3 vezes nas páginas 52, 70 e 75.
- GRETARSSON, B. et al. Smallworlds: Visualizing social recommendations. *Computer Graphics Forum*, Blackwell Publishing Ltd, v. 29, n. 3, p. 833–842, 2010. ISSN 1467-8659. Disponível em: <a href="http://dx.doi.org/10.1111/j.1467-8659.2009.01679.x">http://dx.doi.org/10.1111/j.1467-8659.2009.01679.x</a>. Citado na página 33.
- GROH, G.; EHMIG, C. Recommendations in taste related domains: Collaborative filtering vs. social filtering. In: *Proceedings of the 2007 International ACM Conference on Supporting Group Work*. New York, NY, USA: ACM, 2007. (GROUP '07), p. 127–136. ISBN 978-1-59593-845-9. Disponível em: <a href="http://doi.acm.org/10.1145/1316624.1316643">http://doi.acm.org/10.1145/1316624.1316643</a>. Citado 3 vezes nas páginas 24, 33 e 53.
- HARVEY, M.; RUTHVEN, I.; CARMAN, M. Ranking social bookmarks using topic models. In: *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. New York, NY, USA: ACM, 2010. (CIKM '10), p. 1401–1404. ISBN 978-1-4503-0099-5. Disponível em: <a href="http://doi.acm.org/10.1145/1871437.1871632">http://doi.acm.org/10.1145/1871437.1871632</a>. Citado na página 27.

- HERLOCKER, J. L. et al. Evaluating collaborative filtering recommender systems. *ACM TRANSACTIONS ON INFORMATION SYSTEMS*, v. 22, p. 5–53, 2004. Citado na página 29.
- HU, B.; ESTER, M. Social topic modeling for point-of-interest recommendation in location-based social networks. In: *Data Mining (ICDM)*, 2014 IEEE International Conference on. [S.l.: s.n.], 2014. p. 845–850. ISSN 1550-4786. Citado na página 34.
- HUANG, S. et al. A hybrid multigroup coclustering recommendation framework based on information fusion. *ACM Trans. Intell. Syst. Technol.*, ACM, New York, NY, USA, v. 6, n. 2, p. 27:1–27:22, mar. 2015. ISSN 2157-6904. Disponível em: <a href="http://doi.acm.org/10.1145/2700465">http://doi.acm.org/10.1145/2700465</a>. Citado na página 34.
- JAMALI, M.; ESTER, M. Trustwalker: A random walk model for combining trust-based and item-based recommendation. In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2009. (KDD '09), p. 397–406. ISBN 978-1-60558-495-9. Disponível em: <a href="http://doi.acm.org/10.1145/1557019.1557067">http://doi.acm.org/10.1145/1557019.1557067</a>>. Citado na página 25.
- JAMALI, M.; ESTER, M. A matrix factorization technique with trust propagation for recommendation in social networks. In: *Proceedings of the Fourth ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2010. (RecSys '10), p. 135–142. ISBN 978-1-60558-906-0. Disponível em: <a href="http://doi.acm.org/10.1145/1864708.1864736">http://doi.acm.org/10.1145/1864708.1864736</a>. Citado 3 vezes nas páginas 33, 37 e 38.
- JANNACH, D. et al. Recommender Systems An Introduction. Cambridge University Press, 2010. 23-30 p. ISBN 978-0-521-49336-9. Disponível em: <a href="http://www.cambridge.org-/9780521493369">http://www.cambridge.org-/9780521493369</a>. Citado 4 vezes nas páginas 16, 21, 22 e 26.
- JIANG, M. et al. Scalable recommendation with social contextual information. *Knowledge and Data Engineering, IEEE Transactions on*, v. 26, n. 11, p. 2789–2802, Nov 2014. ISSN 1041-4347. Citado na página 34.
- JIANG, M. et al. Scalable recommendation with social contextual information. *IEEE Transactions on Knowledge and Data Engineering*, v. 26, n. 11, p. 2789–2802, Nov 2014. ISSN 1041-4347. Citado na página 56.
- JIN, J.; CHEN, Q. A trust-based top-k recommender system using social tagging network. In: Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on. [S.l.: s.n.], 2012. p. 1270–1274. Citado na página 33.
- KHALID, O. et al. Omnisuggest: A ubiquitous cloud-based context-aware recommendation system for mobile social networks. *Services Computing, IEEE Transactions on*, v. 7, n. 3, p. 401–414, July 2014. ISSN 1939-1374. Citado na página 34.
- KITCHENHAM, B. Systematic reviews. In: Software Metrics, 2004. Proceedings. 10th International Symposium on. [S.l.: s.n.], 2004. p. xii–xii. ISSN 1530-1435. Citado na página 31.
- KONSTAS, I.; STATHOPOULOS, V.; JOSE, J. M. On social networks and collaborative recommendation. In: *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval.* New York, NY, USA:

- ACM, 2009. (SIGIR '09), p. 195–202. ISBN 978-1-60558-483-6. Disponível em: <a href="http://doi.acm.org/10.1145/1571941.1571977">http://doi.acm.org/10.1145/1571941.1571977</a>. Citado na página 33.
- LEE, D. H.; BRUSILOVSKY, P. Social networks and interest similarity: The case of citeulike. In: *Proceedings of the 21st ACM Conference on Hypertext and Hypermedia*. New York, NY, USA: ACM, 2010. (HT '10), p. 151–156. ISBN 978-1-4503-0041-4. Disponível em: <a href="http://doi.acm.org/10.1145/1810617.1810643">http://doi.acm.org/10.1145/1810617.1810643</a>. Citado na página 17.
- LEVANDOSKI, J. et al. Lars: A location-aware recommender system. In: *Data Engineering (ICDE), 2012 IEEE 28th International Conference on.* [S.l.: s.n.], 2012. p. 450–461. ISSN 1063-6382. Citado na página 33.
- LIU, H. et al. Trucom: Exploiting domain-specific trust networks for multicategory item recommendation. *Systems Journal, IEEE*, PP, n. 99, p. 1–10, 2015. ISSN 1932-8184. Citado 3 vezes nas páginas 34, 37 e 39.
- LIU, N. N.; HE, L.; ZHAO, M. Social temporal collaborative ranking for context aware movie recommendation. *ACM Trans. Intell. Syst. Technol.*, ACM, New York, NY, USA, v. 4, n. 1, p. 15:1–15:26, fev. 2013. ISSN 2157-6904. Disponível em: <a href="http://doi.acm.org/10.1145/2414425.2414440">http://doi.acm.org/10.1145/2414425.2414440</a>. Citado 2 vezes nas páginas 33 e 57.
- LIU, X.; ABERER, K. Soco: A social network aided context-aware recommender system. In: *Proceedings of the 22Nd International Conference on World Wide Web*. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2013. (WWW '13), p. 781–802. ISBN 978-1-4503-2035-1. Disponível em: <a href="http://dl.acm.org/citation.cfm?id=2488388.2488457">http://dl.acm.org/citation.cfm?id=2488388.2488457</a>. Citado na página 33.
- MA, H. et al. Recommender systems with social regularization. In: *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining.* New York, NY, USA: ACM, 2011. (WSDM '11), p. 287–296. ISBN 978-1-4503-0493-1. Disponível em: <a href="http://doi.acm.org/10.1145/1935826.1935877">http://doi.acm.org/10.1145/1935826.1935877</a>>. Citado 4 vezes nas páginas 33, 37, 38 e 39.
- MEYFFRET, S.; MÉDINI, L.; LAFOREST, F. Trust-based local and social recommendation. In: *Proceedings of the 4th ACM RecSys Workshop on Recommender Systems and the Social Web.* New York, NY, USA: ACM, 2012. (RSWeb '12), p. 53–60. ISBN 978-1-4503-1638-5. Disponível em: <a href="http://doi.acm.org/10.1145/2365934.2365945">http://doi.acm.org/10.1145/2365934.2365945</a>. Citado 3 vezes nas páginas 33, 37 e 38.
- MUGNAINI, R.; DIGIAMPIETRI, L. A.; MENA-CHALCO, J. P. Comunicação científica no brasil (1998-2012): indexação, crescimento, fluxo e dispersão. *Transinformação*, v. 26, p. 239–252, 2014. Citado na página 43.
- NARUCHITPARAMES, J.; GUNES, M.; LOUIS, S. Friend recommendations in social networks using genetic algorithms and network topology. In: *Evolutionary Computation (CEC)*, 2011 IEEE Congress on. [S.l.: s.n.], 2011. p. 2207–2214. ISSN Pending. Citado na página 33.
- NOULAS, A. et al. A random walk around the city: New venue recommendation in location-based social networks. In: *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Conference on Social Computing (SocialCom).* [S.l.: s.n.], 2012. p. 144–153. Citado na página 33.

- OU, W. Extracting user interests from graph connections for machine learning in location-based social networks. In: *Proceedings of the MLSDA 2014 2Nd Workshop on Machine Learning for Sensory Data Analysis.* New York, NY, USA: ACM, 2014. (MLSDA'14), p. 41:41–41:47. ISBN 978-1-4503-3159-3. Disponível em: <a href="http://doi.acm.org/10.1145/2689746.2689754">http://doi.acm.org/10.1145/2689746.2689754</a>>. Citado na página 34.
- OZSOY, M.; POLAT, F.; ALHAJJ, R. Multi-objective optimization based location and social network aware recommendation. In: *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2014 International Conference on.* [S.l.: s.n.], 2014. p. 233–242. Citado na página 34.
- QIAN, X. et al. Personalized recommendation combining user interest and social circle. Knowledge and Data Engineering, IEEE Transactions on, v. 26, n. 7, p. 1763–1777, July 2014. ISSN 1041-4347. Citado na página 34.
- QUIJANO-SANCHEZ, L.; RECIO-GARCIA, J.; DIAZ-AGUDO, B. Happymovie: A facebook application for recommending movies to groups. In: *Tools with Artificial Intelligence (ICTAI)*, 2011 23rd IEEE International Conference on. [S.l.: s.n.], 2011. p. 239–244. ISSN 1082-3409. Citado na página 16.
- RICCI, F.; ROKACH, L.; SHAPIRA, B. *Introduction to Recommender Systems Handbook*. Springer US, 2011. 1-35 p. ISBN 978-0-387-85819-7. Disponível em: <a href="http://dx.doi.org/10.1007/978-0-387-85820-31">http://dx.doi.org/10.1007/978-0-387-85820-31</a>. Citado 7 vezes nas páginas 16, 21, 23, 24, 47, 50 e 63.
- RICHARDSON, M.; DOMINGOS, P. Mining knowledge-sharing sites for viral marketing. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* New York, NY, USA: ACM, 2002. (KDD '02), p. 61–70. ISBN 1-58113-567-X. Disponível em: <a href="http://doi.acm.org/10.1145/775047.775057">http://doi.acm.org/10.1145/775047.775057</a>. Citado na página 36.
- ROHANI, V. A.; KASIRUN, Z. M.; RATNAVELU, K. An enhanced content-based recommender system for academic social networks. In: *Proceedings of the 2014 IEEE Fourth International Conference on Big Data and Cloud Computing.* Washington, DC, USA: IEEE Computer Society, 2014. (BDCLOUD '14), p. 424–431. ISBN 978-1-4799-6719-3. Disponível em: <a href="http://dx.doi.org/10.1109/BDCloud.2014.131">http://dx.doi.org/10.1109/BDCloud.2014.131</a>. Citado na página 41.
- SARWAT, M. et al. Lars*: An efficient and scalable location-aware recommender system. *Knowledge and Data Engineering, IEEE Transactions on*, PP, n. 99, p. 1–1, 2013. ISSN 1041-4347. Citado na página 33.
- SCELLATO, S.; NOULAS, A.; MASCOLO, C. Exploiting place features in link prediction on location-based social networks. In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* New York, NY, USA: ACM, 2011. (KDD '11), p. 1046–1054. ISBN 978-1-4503-0813-7. Disponível em: <a href="http://doi.acm.org/10.1145/2020408.2020575">http://doi.acm.org/10.1145/2020408.2020575</a>. Citado na página 33.
- SHANGGUAN, Q. et al. Book recommendation based on joint multi-relational model. In: Cloud and Green Computing (CGC), 2012 Second International Conference on. [S.l.: s.n.], 2012. p. 523–530. Citado na página 34.

- SILVA, N. et al. A graph-based friend recommendation system using genetic algorithm. In: *Evolutionary Computation (CEC)*, 2010 IEEE Congress on. [S.l.: s.n.], 2010. p. 1–7. Citado na página 33.
- SIM, B. S. et al. Type-based context-aware service recommender system for social network. In: Computer, Information and Telecommunication Systems (CITS), 2012 International Conference on. [S.l.: s.n.], 2012. p. 1–5. Citado na página 33.
- SIM, B. S. et al. Type-based context-aware service recommender system for social network. In: Computer, Information and Telecommunication Systems (CITS), 2012 International Conference on. [S.l.: s.n.], 2012. p. 1–5. Citado na página 57.
- SOPCHOKE, S.; KIJSIRIKUL, B. A step towards high quality one-class collaborative filtering using online social relationships. In: *Advanced Computer Science and Information System (ICACSIS)*, 2011 International Conference on. [S.l.: s.n.], 2011. p. 243–248. Citado na página 34.
- WANG, D. et al. Recommendation based on weighted social trusts and item relationships. In: *Proceedings of the 29th Annual ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2014. (SAC '14), p. 254–259. ISBN 978-1-4503-2469-4. Disponível em: <a href="http://doi.acm.org/10.1145/2554850.2554884">http://doi.acm.org/10.1145/2554850.2554884</a>. Citado 3 vezes nas páginas 34, 37 e 39.
- WANG, H.; TERROVITIS, M.; MAMOULIS, N. Location recommendation in location-based social networks using user check-in data. In: *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. New York, NY, USA: ACM, 2013. (SIGSPATIAL'13), p. 374–383. ISBN 978-1-4503-2521-9. Disponível em: <a href="http://doi.acm.org/10.1145/2525314.2525357">http://doi.acm.org/10.1145/2525314.2525357</a>. Citado na página 34.
- WANG, J.; LI, H.; ZHAO, H. The contextual group recommendation. In: *Intelligent Networking and Collaborative Systems (INCoS)*, 2013 5th International Conference on. [S.l.: s.n.], 2013. p. 127–131. Citado na página 34.
- WANG, Q.; DENG, Q. Catching preference drift with initiators in social network. In: *Parallel and Distributed Systems (ICPADS)*, 2011 IEEE 17th International Conference on. [S.l.: s.n.], 2011. p. 829–834. ISSN 1521-9097. Citado na página 33.
- WANG, T. et al. User interests imbalance exploration in social recommendation: A fitness adaptation. In: *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. New York, NY, USA: ACM, 2014. (CIKM '14), p. 281–290. ISBN 978-1-4503-2598-1. Disponível em: <a href="http://doi.acm.org/10.1145/2661829.2662043">http://doi.acm.org/10.1145/2661829.2662043</a>. Citado 3 vezes nas páginas 34, 37 e 39.
- WOERNDL, W.; GROH, G. Utilizing physical and social context to improve recommender systems. In: *Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology Workshops.* Washington, DC, USA: IEEE Computer Society, 2007. (WI-IATW '07), p. 123–128. ISBN 0-7695-3028-1. Disponível em: <a href="http://dl.acm.org/citation.cfm?id=1339264.1339674">http://dl.acm.org/citation.cfm?id=1339264.1339674</a>. Citado na página 33.
- XIE, Y. et al. Crowdsourcing recommendations from social sentiment. In: *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining*. New York, NY, USA: ACM, 2012. (WISDOM '12), p. 9:1–9:8. ISBN 978-1-4503-1543-2. Disponível em: <a href="http://doi.acm.org/10.1145/2346676.2346685">http://doi.acm.org/10.1145/2346676.2346685</a>. Citado na página 33.

YANG, X.; GUO, Y.; LIU, Y. Bayesian-inference-based recommendation in online social networks. *Parallel and Distributed Systems, IEEE Transactions on*, v. 24, n. 4, p. 642–651, 2013. ISSN 1045-9219. Citado 3 vezes nas páginas 33, 36 e 37.

YANG, X.; STECK, H.; LIU, Y. Circle-based recommendation in online social networks. In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* New York, NY, USA: ACM, 2012. (KDD '12), p. 1267–1275. ISBN 978-1-4503-1462-6. Disponível em: <a href="http://doi.acm.org/10.1145/2339530.2339728">http://doi.acm.org/10.1145/2339530.2339728</a>. Citado 4 vezes nas páginas 33, 37, 38 e 40.

ZHOU, J.; LUO, T.; LIN, H. A novel recommendation system with collective intelligence. In: Web Society (SWS), 2010 IEEE 2nd Symposium on. [S.l.: s.n.], 2010. p. 151–157. Citado na página 17.