

Ricardo Erikson Veras de Sena Rosa

Identificação de Uma Vizinhança Relevante Em Sistemas de Recomendação Para TV Interativa Utilizando Tela Secundária

Belo Horizonte - MG - Brazil

fevereiro/2021

Ricardo Erikson Veras de Sena Rosa

Identificação de Uma Vizinhança Relevante Em Sistemas de Recomendação Para TV Interativa Utilizando Tela Secundária

Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito parcial à obtenção do título de Doutor em Engenharia Elétrica.

Universidade Federal de Minas Gerais – UFMG

Escola de Engenharia

Programa de Pós-Graduação em Engenharia Elétrica

Orientador: Dr. -Ing. Vicente Ferreira de Lucena Júnior

Belo Horizonte - MG - Brazil

fevereiro/2021

R788i

Rosa, Ricardo Erikson Veras de Sena.

Identificação de uma vizinhança relevante em sistemas de recomendação para TV interativa utilizando tela secundária [recurso eletrônico] / Ricardo Erikson Veras de Sena Rosa- 2021.
1 recurso online (148 f : il., color.) : pdf.

Orientador: Vicente Ferreira de Lucena Junior.

Tese (doutorado) - Universidade Federal de Minas Gerais, Escola de Engenharia.

Apêndices: f. 128-148.

Bibliografia: f. 117-127.

Exigências do sistema: Adobe Acrobat Reader.

1. Engenharia elétrica - Teses. 2. Sistemas de recomendação - Teses.
3. Filtragem colaborativa - Teses. 4. Multimídia interativa - Teses.
I. Lucena Junior, Vicente Ferreira de. II. Universidade Federal de Minas Gerais. Escola de Engenharia. III. Título.

CDU: 621.3(043)

**"Identificação de Uma Vizinhança Relevante Em Sistemas de
Recomendação Para TV Interativa Utilizando Tela Secundária"**

Ricardo Erikson Veras de Sena Rosa

Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Doutor em Engenharia Elétrica.

Aprovada em 25 de fevereiro de 2021.

Por:



**Prof. Dr. Vicente Ferreira de Lucena Junior
PPGEE (UFAM)**



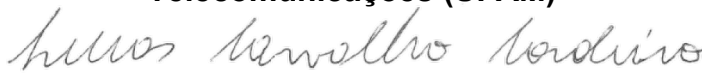
**Prof. Dr. Walmir Matos Caminhas
DELT (UFMG)**



**Prof. Dr. Guilherme Augusto Silva Pereira
DEE (UFMG)**



**Prof. Dr. Cicero Ferreira Fernandes Costa Filho
Faculdade de Tecnologia, Departamento de Eletrônica e
Telecomunicações (UFAM)**



**Prof. Dr. Lucas Carvalho Cordeiro
School of Computer Science (University of Manchester)**

*Este trabalho é dedicado a todos aqueles que são
resilientes e persistentes na busca pelos seus objetivos.*

Agradecimentos

Os agradecimentos principais são direcionados a Deus e ao Universo, que me proporcionaram as oportunidades e a inspiração necessária para conclusão desse trabalho. A minha relação com Deus e com o Universo talvez não seja das mais convencionais, mas eu sei que de alguma forma eles estão trabalhando para ajudar aqueles que se ajudam.

Agradecimentos especiais são direcionados à Fabiana que esteve ao meu lado durante toda a realização desse trabalho. Nos momentos em que eu perdia o foco, era ela que me trazia de volta para continuar trabalhando na pesquisa. O título é meu e dela.

Agradeço à minha família, que sempre me apoiou na busca pelo título de doutor. Mesmo com a distância física por vários anos, eles sempre estiveram presentes com ligações ou da maneira que fosse possível. Meu pai e minha mãe nunca deixaram me faltar nada e eu sei que a prioridade deles era ver os filhos formados e educados. Eu sei também que, além de mim, meu pai e minha mãe sonharam com esse momento. Por isso, esse título também é deles.

Agradeço também ao professor Vicente Lucena que não desistiu de mim durante todos esses anos e me ajudou a chegar até aqui. Obrigado também a todos os amigos e professores que participaram desse processo na Universidade Federal do Amazonas (UFAM), no Doutorado Interinstitucional (DINTER) do Instituto Federal do Amazonas (IFAM) e na Universidade Federal de Minas Gerais (UFMG). Obrigado a todos que de alguma maneira participaram do processo, seja no trabalho ou na universidade.

Agradeço ao Centro de Pesquisa e Desenvolvimento em Tecnologia Eletrônica e da Informação (CETELI) pelo espaço que foi cedido para a realização das minhas pesquisas desde o mestrado até o doutorado. Foi um ambiente de muito aprendizado e muito crescimento pessoal e profissional. Os agradecimentos também são direcionados à Fundação de Amparo à Pesquisa do Estado do Amazonas (FAPEAM) e à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pela disponibilização de bolsas de pesquisa que foram fundamentais para a realização do trabalho.

“Life is poker, not chess.”
(Annie Duke)

Resumo

Os avanços nas tecnologias de comunicação permitiram a expansão dos serviços de TV, o que possibilitou a disponibilização de uma quantidade crescente de conteúdo audiovisual. Essa grande quantidade de conteúdo introduz um problema conhecido como sobrecarga de informação, onde os usuários não conseguem encontrar conteúdo de seu interesse de maneira eficiente e em tempo hábil. Nesse sentido, os sistemas de recomendação (SRs) surgem como uma ferramenta promissora para ajudar os usuários a superarem o problema da sobrecarga de informação. Esse trabalho explora dois desafios dos SRs para TV Interativa: obtenção dos dados de interação e a melhoria da precisão das recomendações. O primeiro desafio ocorre devido à natureza coletiva dos ambientes de TV, onde a interação tipicamente acontece por meio de um controle remoto que muitas vezes é compartilhado por um grupo de usuários. Por conta disso, é difícil identificar o autor de cada interação de maneira que seja possível utilizar os dados obtidos para customizar conteúdo para um usuário específico com base em suas interações individuais. O segundo desafio é inerente dos SRs. Uma precisão maior se traduz em menos erros de recomendação, o que eleva a confiança dos usuários na utilização do sistema. Para atacar esses desafios, esse trabalho se propõe, primeiramente, a utilizar dispositivos de tela secundária para facilitar a captura de dados individuais dos usuários. Por serem de uso pessoal, as interações realizadas por meio dos dispositivos de tela secundária podem ser identificadas a partir da sua origem. Para melhorar a precisão das recomendações, esse trabalho busca encontrar uma vizinhança mais adequada com base na similaridade entre usuários que avaliaram grupos de objetos relacionados, chamada de similaridade local. Nesse sentido, o método proposto utiliza técnicas baseadas em clusterização, alocação de recursos e normalização. A avaliação dos métodos propostos é realizada por meio da identificação e implementação de cenários para a captura individualizada de dados de interação, e por meio da utilização de métricas de precisão utilizando três bases de dados amplamente adotadas na literatura: MovieLens 100k, MovieLens 1M e Netflix. Os resultados alcançados sugerem a viabilidade técnica do método de captura de dados utilizando telas secundárias e uma melhora na precisão das predições do método proposto para a recomendação de conteúdo.

Palavras-chave: sistemas de recomendação. filtragem colaborativa. multimídia interativa. similaridade. TV interativa.

Abstract

Advances in communication technologies have enabled the expansion of TV services, which has made it possible to make available an increasing amount of audiovisual content. This large amount of content introduces a problem known as information overload, where users cannot find the content of their interest in an efficient and timely manner. Recommender systems (RS) emerge as a promising tool to help users overcome information overload. This work explores two challenges for RS for Interactive TV: obtaining interaction data and improving the prediction accuracy for recommendations. The first challenge arises from the collective nature of TV environments, where interaction often occurs by using a remote control, which is shared by a group of users. As a result, it is difficult to identify the author of each interaction in a way that would be possible to customize the content for a specific user based on their individual interactions. The second challenge is inherent to RS. A greater accuracy translates into fewer recommendation errors, which increases users' confidence in using the system. To tackle these challenges, this work proposes, first, to use second screen devices to facilitate the capture of users' individual data. Since they are intended for personal use, interactions carried out using second screen devices can be individualized. To improve the prediction accuracy, this work seeks to find a more suitable neighborhood that relies on the similarity between users who evaluated groups of related objects, called local similarity. For this, the proposed method uses techniques that are based on clustering, resource allocation, and normalization. The evaluation of the proposed methods is performed by identifying and implementing scenarios for the individualized capture of interaction data and using prediction accuracy metrics on three databases widely used in the literature: MovieLens 100k, MovieLens 1M, and Netflix. The achieved results suggest the technical feasibility of the method for data capture using secondary screens and an improvement in the accuracy of the prediction approach that was proposed in this work.

Keywords: recommender systems. collaborative filtering. interactive media. similarity. interactive TV.

Lista de ilustrações

Figura 1 – Grafo bipartido representando uma rede usuário-objeto.	35
Figura 2 – Os vértices redondos são os usuários (\mathcal{U}) e os vértices quadrados são os objetos (\mathcal{O}) da rede usuário-objeto. Os recursos unitários u_1 , u_2 e u_3 são inicialmente atribuídos aos vértices de usuários. Então, cada recurso unitário é igualmente distribuído fluindo dos usuários para os objetos ($a \rightarrow b$). No passo seguinte, os recursos são igualmente distribuídos de volta dos objetos para os usuários ($b \rightarrow c$).	46
Figura 3 – Visão geral da abordagem para a captura dos dados individuais de interação dos usuários.	73
Figura 4 – Arquitetura representando a integração dos componentes.	76
Figura 5 – Estrutura básica para a representação dos dados de interação.	78
Figura 6 – Representação JSON dos dados específicos de interação.	78
Figura 7 – Representação JSON dos dados contextuais que são utilizados para complementar os dados do evento de interação.	79
Figura 8 – Visão geral do método proposto. O método consiste em dois processos: modelagem e predição. O processo de modelagem visa a construção dos modelos de similaridade local para cada <i>cluster</i> , e o processo de predição é focado em estimar as preferências de usuários em objetos cujo $r_{u_i o_k} = \emptyset$	83
Figura 9 – Interface com os métodos que são chamados para cada tipo de interação. A imagem também apresenta as anotações com o mapeamento da interação para a URI e para o método da requisição HTTP que deve ser feita para enviar os dados da interação ao RDU.	96
Figura 10 – Implementação básica da classe <i>InteractionsHandler</i> que intercepta as chamadas aos métodos do <i>proxy</i> de interação.	97
Figura 11 – Passo a passo de como os dados da interação são capturados desde o momento do início da interação até o momento em que o serviço web é consumido com o envio dos dados ao RDU.	98
Figura 12 – Mapeamento de uma requisição HTTP de um serviço web para um método que faz o tratamento dos dados no RDU.	99

Figura 13 – Exemplo de dados de interação capturados para um evento de troca de canais.	101
Figura 14 – Exemplo dos dados de interação capturados para um evento de avaliação de programa de TV.	102
Figura 15 – Curva do MAE do conjunto de dados \mathcal{R}_{test} para as bases ML 100k, ML 1M e Netflix.	104
Figura 16 – Curva do RMSE do conjunto de dados \mathcal{R}_{test} para as bases ML 100k, ML 1M e Netflix.	105
Figura 17 – Mapa de calor dos valores de MAE e RMSE para a partição 1 considerando um número variável de vizinhos e um número variável de <i>clusters</i> para o método SL_KM.	108
Figura 18 – Mapa de calor dos valores de MAE e RMSE para a partição 1 considerando um número variável de vizinhos e um número variável de <i>clusters</i> para o método SL_KM.	134
Figura 19 – Mapa de calor dos valores de MAE e RMSE para a partição 2 considerando um número variável de vizinhos e um número variável de <i>clusters</i> para o método SL_KM.	135
Figura 20 – Mapa de calor dos valores de MAE e RMSE para a partição 3 considerando um número variável de vizinhos e um número variável de <i>clusters</i> para o método SL_KM.	135
Figura 21 – Mapa de calor dos valores de MAE e RMSE para a partição 4 considerando um número variável de vizinhos e um número variável de <i>clusters</i> para o método SL_KM.	136

Lista de algoritmos

Algoritmo 1 – Estágio de modelagem da similaridade	85
Algoritmo 2 – Seleção das preferências.	86
Algoritmo 3 – Cálculo da média e do desvio padrão considerando objetos perten- centes ao <i>cluster</i> de interesse \mathcal{C}_c	87
Algoritmo 4 – Normalização das preferências considerando o <i>cluster</i> \mathcal{C}_c	87
Algoritmo 5 – Cálculo das similaridades para os usuários.	88
Algoritmo 6 – Seleção da vizinhança considerando um <i>cluster</i> de interesse \mathcal{C}_c	90
Algoritmo 7 – Agregação das preferências.	91

Lista de tabelas

Tabela 1	– Comparação entre as abordagens baseadas em colaboração e abordagens baseadas em conteúdo.	41
Tabela 2	– Classificação dos objetos em função de sua relevância para as preferências observadas e para as preferências preditas.	49
Tabela 3	– Resumo dos métodos e técnicas de recomendação utilizadas nos trabalhos relacionados. Os campos abreviados na tabela possuem os seguintes significados: FC (filtragem colaborativa), CONT (conteúdo), MEM (memória), MOD (modelo), IND (individual), GRUP (grupo), SIM (similaridade) e CL (clusterização).	63
Tabela 4	– Exemplo de uma matriz de avaliações R com 3 usuários e 8 objetos. As preferências são indicadas em uma escala de 1 a 5. O usuário u_1 é considerado o usuário ativo e os objetos são categorizados em dois grupos: grupo A e grupo B.	82
Tabela 5	– Métricas de similaridade utilizadas na comparação com o método proposto nos resultados experimentais.	103
Tabela 6	– Melhorias de performance da abordagem proposta nesse trabalho em relação os métodos de referência considerando as métricas MAE e o RMSE para a partição 1 da base de dados Netflix. Os valores em negrito em cada linha dos métodos de similaridade global indicam a pior performance, enquanto os valores em negrito em cada linha dos métodos baseados em similaridade local indicam a melhor performance.	107
Tabela 7	– Tempo médio de execução dos métodos usados nas comparações. Os valores de cada linha são apresentados em segundos. TC e TT são o Tempo de Clusterização e o Tempo de Treinamento, respectivamente. .	111

- Tabela 8 – Melhorias de performance da abordagem proposta nesse trabalho em relação os métodos de referência considerando as métricas MAE e o RMSE para a partição 1 da base de dados ML 100k. Os valores em negrito em cada linha dos métodos de similaridade global indicam a pior performance, enquanto os valores em negrito em cada linha dos métodos baseados em similaridade local indicam a melhor performance. 137
- Tabela 9 – Melhorias de performance da abordagem proposta nesse trabalho em relação os métodos de referência considerando as métricas MAE e o RMSE para a partição 2 da base de dados ML 100k. Os valores em negrito em cada linha dos métodos de similaridade global indicam a pior performance, enquanto os valores em negrito em cada linha dos métodos baseados em similaridade local indicam a melhor performance. 138
- Tabela 10 – Melhorias de performance da abordagem proposta nesse trabalho em relação os métodos de referência considerando as métricas MAE e o RMSE para a partição 3 da base de dados ML 100k. Os valores em negrito em cada linha dos métodos de similaridade global indicam a pior performance, enquanto os valores em negrito em cada linha dos métodos baseados em similaridade local indicam a melhor performance. 139
- Tabela 11 – Melhorias de performance da abordagem proposta nesse trabalho em relação os métodos de referência considerando as métricas MAE e o RMSE para a partição 4 da base de dados ML 100k. Os valores em negrito em cada linha dos métodos de similaridade global indicam a pior performance, enquanto os valores em negrito em cada linha dos métodos baseados em similaridade local indicam a melhor performance. 140
- Tabela 12 – Melhorias de performance da abordagem proposta nesse trabalho em relação os métodos de referência considerando as métricas MAE e o RMSE para a partição 1 da base de dados ML 1M. Os valores em negrito em cada linha dos métodos de similaridade global indicam a pior performance, enquanto os valores em negrito em cada linha dos métodos baseados em similaridade local indicam a melhor performance. 141
- Tabela 13 – Melhorias de performance da abordagem proposta nesse trabalho em relação os métodos de referência considerando as métricas MAE e o RMSE para a partição 2 da base de dados ML 1M. Os valores em negrito em cada linha dos métodos de similaridade global indicam a pior performance, enquanto os valores em negrito em cada linha dos métodos baseados em similaridade local indicam a melhor performance. 142

- Tabela 14 – Melhorias de performance da abordagem proposta nesse trabalho em relação os métodos de referência considerando as métricas MAE e o RMSE para a partição 3 da base de dados ML 1M. Os valores em negrito em cada linha dos métodos de similaridade global indicam a pior performance, enquanto os valores em negrito em cada linha dos métodos baseados em similaridade local indicam a melhor performance. 143
- Tabela 15 – Melhorias de performance da abordagem proposta nesse trabalho em relação os métodos de referência considerando as métricas MAE e o RMSE para a partição 4 da base de dados ML 1M. Os valores em negrito em cada linha dos métodos de similaridade global indicam a pior performance, enquanto os valores em negrito em cada linha dos métodos baseados em similaridade local indicam a melhor performance. 144
- Tabela 16 – Melhorias de performance da abordagem proposta nesse trabalho em relação os métodos de referência considerando as métricas MAE e o RMSE para a partição 1 da base de dados Netflix. Os valores em negrito em cada linha dos métodos de similaridade global indicam a pior performance, enquanto os valores em negrito em cada linha dos métodos baseados em similaridade local indicam a melhor performance. 145
- Tabela 17 – Melhorias de performance da abordagem proposta nesse trabalho em relação os métodos de referência considerando as métricas MAE e o RMSE para a partição 2 da base de dados Netflix. Os valores em negrito em cada linha dos métodos de similaridade global indicam a pior performance, enquanto os valores em negrito em cada linha dos métodos baseados em similaridade local indicam a melhor performance. 146
- Tabela 18 – Melhorias de performance da abordagem proposta nesse trabalho em relação os métodos de referência considerando as métricas MAE e o RMSE para a partição 3 da base de dados Netflix. Os valores em negrito em cada linha dos métodos de similaridade global indicam a pior performance, enquanto os valores em negrito em cada linha dos métodos baseados em similaridade local indicam a melhor performance. 147
- Tabela 19 – Melhorias de performance da abordagem proposta nesse trabalho em relação os métodos de referência considerando as métricas MAE e o RMSE para a partição 4 da base de dados Netflix. Os valores em negrito em cada linha dos métodos de similaridade global indicam a pior performance, enquanto os valores em negrito em cada linha dos métodos baseados em similaridade local indicam a melhor performance. 148

Lista de abreviaturas e siglas

ATV	Ambiente de TV
CBLS	<i>Cluster-based Local Similarity</i>
DVR	<i>Digital Video Recorder</i>
EIT	<i>Event Information Table</i>
EPG	<i>Electronic Program Guide</i>
FC	Filtragem Colaborativa
HCA	<i>Hierarchical Cluster Analysis</i>
IPTV	<i>Internet Protocol Television</i>
JSON	<i>JavaScript Object Notation</i>
KNN	<i>K-Nearest Neighbors</i> (<i>k</i> vizinhos mais próximos)
MAE	<i>Mean Absolute Error</i> (Erro Absoluto Médio)
NCL	<i>Nested Context Language</i>
PCC	<i>Pearson's Correlation Coefficient</i> (Coeficiente de Correlação de Pearson)
PSTV	Provedor de Serviço de TV
PVR	<i>Personal Video Recorder</i>
RDU	Repositório de Dados dos Usuários
RMSE	<i>Root Mean Squared Error</i> (Raiz do Erro Quadrático Médio)
SAU	Serviço de Autenticação do Usuário
SPCC	<i>Sigmoid Pearson's Correlation Coefficient</i>
SR	Sistema de Recomendação

STB	<i>Set-top box</i>
SVD	<i>Singular Value Decomposition</i>
TVDI	TV Digital Interativa
VoD	<i>Video on Demand</i> (vídeo sob demanda)
WPCC	<i>Weighted Pearson's Correlation Coefficient</i>
XML	<i>eXtensible Markup Language</i>

Sumário

1	Introdução	20
1.1	Definição do Problema	23
1.1.1	Problema 1: Captura de Dados de Maneira Organizada e Individualizada	23
1.1.2	Problema 2: Recomendação de Conteúdo utilizando uma Vizinhaça Relevante	24
1.2	Solução Proposta	25
1.3	Objetivos	26
1.3.1	Objetivo Geral	27
1.3.2	Objetivos Específicos	27
1.4	Contribuições	27
1.5	Organização da Tese	28
2	Sistemas de Recomendação	29
2.1	Aplicações Reais	29
2.2	Desafios dos Métodos Existentes	32
2.3	Notações Básicas de Sistemas de Recomendação	33
2.4	Métodos de Recomendação	35
2.4.1	Abordagens Baseadas em Colaboração	36
2.4.1.1	Algoritmos Baseados em Memória	37
2.4.1.2	Algoritmos Baseados em Modelo	39
2.4.2	Abordagens Baseadas em Conteúdo	40
2.4.3	Abordagens Híbridas	41
2.5	Métricas de Similaridade	42
2.5.1	Métricas Tradicionais	43
2.5.2	Outras Métricas de Similaridade	44
2.5.3	Similaridade Baseada em Objetos	45
2.6	Alocação de Recursos	45
2.7	Métricas de Avaliação para Recomendações	47
2.7.1	Precisão	48
2.7.2	Classificação	48
2.7.3	Diversidade	50
2.7.4	Cobertura	51
2.8	Conclusões	52
3	Personalização de Conteúdo para Usuários de TV	53

3.1	Motivação para a Personalização de Conteúdo para TV	53
3.2	Dados de Entrada para Personalização	55
3.2.1	Dados de Interação dos Usuários	55
3.2.2	Metadados do Conteúdo	57
3.3	Recomendação para TV	59
3.4	Vizinhança na Personalização de Conteúdo	64
3.5	Telas Secundárias na Interatividade	66
3.6	Conclusões	68
4	Materiais e Métodos	70
4.1	Captura de Dados de Maneira Organizada e Individualizada	70
4.1.1	Descrição do Problema	71
4.1.2	Visão Geral da Proposta	72
4.1.3	Captura dos Eventos de Interação	75
4.1.4	Representação dos Dados de Interação	77
4.1.4.1	Representação dos Dados Específicos de Interação	78
4.1.4.2	Representação dos Dados Contextuais	79
4.2	Recomendação de Conteúdo utilizando uma Vizinhança Relevante	80
4.2.1	Descrição do Problema	80
4.2.2	Visão Geral da Proposta	81
4.2.3	Método Proposto	83
4.2.3.1	Processo de Modelagem	84
4.2.3.2	Processo de Predição	89
4.3	Bases de Dados	90
4.4	Métricas	92
4.5	Conclusões	93
5	Resultados Experimentais	95
5.1	Captura de Dados de Maneira Organizada e Individualizada	95
5.1.1	Implementação do Método	95
5.1.1.1	Mapeamento das Interações para Requisições HTTP	96
5.1.1.2	Design da API de Serviços Web e Implementação no RDU	99
5.1.1.3	Consumo dos Serviços Web	100
5.1.2	Protótipo Experimental	100
5.1.2.1	Cenário 1: Mudança de Canal	101
5.1.2.2	Cenário 2: Avaliação de um Programa	102
5.2	Recomendação de Conteúdo Utilizando uma Vizinhança Relevante	102
5.2.1	Resultados e Discussão	103
5.2.1.1	Métodos Usados para Comparação	103
5.2.1.2	Efeitos do Tamanho da Vizinhança	104

5.2.1.3	Efeitos do Número de Vizinhos e <i>Clusters</i>	108
5.2.1.4	Análise da Complexidade de Tempo	109
5.3	Conclusões	110
6	Considerações Finais	112
6.1	Limitações	114
6.2	Contribuições Alcançadas	114
6.3	Publicações Científicas Alcançadas	115
6.4	Trabalhos Futuros	115
	Referências	117
	 Apêndices	 128
	APÊNDICE A Notações Matemáticas e Símbolos	129
	APÊNDICE B Algoritmos de Clusterização	131
B.1	<i>Affinity Propagation</i>	131
B.2	K-Means	133
	APÊNDICE C Performance dos Métodos de Recomendação	134

Capítulo 1

Introdução

A expansão dos serviços baseados em internet tem permitido que muitos produtos como vídeos, músicas, livros e filmes cheguem ao alcance dos usuários por meio de poucos cliques na tela de um computador ou de dispositivos móveis. As empresas usualmente disponibilizam um enorme catálogo de produtos por meio de sites de *e-commerce* e aplicativos móveis para atender a demanda dos mais variados clientes. Diante de todo o universo de produtos disponíveis, cada cliente deve ativamente se engajar na atividade de procurar os produtos de seu interesse e determinar se o grau de relevância desses produtos está em conformidade com as suas preferências. Sem os mecanismos que facilitem a atividade de buscar por conteúdo de interesse, esse cenário com inúmeros produtos disponíveis caracteriza um problema conhecido como sobrecarga de informação. Devido à grande quantidade de alternativas, os clientes não conseguem encontrar os produtos mais relevantes de maneira eficiente e em tempo hábil.

De acordo com Hinz e Eckert (2010), uma grande variedade de produtos só é benéfica aos consumidores se as tecnologias existentes os ajudarem a encontrar produtos de seu interesse de maneira eficiente e eficaz. Nesse sentido, os Sistemas de Recomendação (SRs) surgem como uma solução promissora para ajudar os consumidores a superarem o problema da sobrecarga de informação. Com o uso de SRs, ocorre uma inversão de responsabilidades. Em vez de ativamente procurar pelos produtos, os usuários assumem uma posição passiva e os SRs atuam encontrando e sugerindo produtos potencialmente relevantes a esses usuários.

SRs apresentam uma grande significância econômica e social. Muitas empresas os usam como meio de aumentar a satisfação dos usuários e conseqüentemente aumentar as

taxas de conversão¹ e os lucros (JANNACH; JUGOVAC, 2019) ao aproximar produtos de consumo a potenciais clientes. Empresas como a *Google*, *Amazon*, *Spotify* e *Netflix*, por exemplo, fazem forte uso de SRs para apresentar produtos diversificados e potencialmente atrativos para os usuários, com isso aumentando as vendas em seus serviços online. Redes sociais como o *Twitter* e *Instagram* utilizam SRs para recomendar perfis semelhantes a outros que já são seguidos pelo usuário. Muitas vezes, os próprios SRs usam as redes sociais para aprimorar a precisão e a relevância das recomendações ao investigar os relacionamentos entre os usuários e sugerir produtos que foram selecionados por usuários semelhantes (MA et al., 2011; SUN et al., 2015).

A ideia principal dos SRs é modelar as preferências dos usuários e antecipar potenciais interesses por produtos. A modelagem das preferências é tipicamente realizada por meio da análise dos dados obtidos a partir do histórico de atividades dos usuários de um sistema. Cada atividade pode ser representada por meio de um relacionamento que expressa a preferência de um usuário em relação a determinado produto em uma rede usuário-objeto (SHANG et al., 2010). Esse relacionamento pode expressar uma preferência unária (onde os usuários simplesmente mostram interesse por um produto relevante), binária (onde é possível distinguir entre bom ou ruim), ou numérica que comumente utiliza uma escala de classificação de 1 a 5 (onde 1 representa baixa apreciação e 5 representa alta apreciação) (CACHEDA et al., 2011). Em outras palavras, essas preferências são avaliações representadas de maneira organizada e sistematizada. Exemplos dessas preferências aparecem atualmente nas seguintes situações em sistemas do mundo real:

- **Unária:** Na plataforma de compartilhamento de fotos *Instagram*, os usuários podem clicar no ícone *curtir* para expressar sua apreciação por uma foto. Atualmente, essa é a única opção disponível para que os usuários explicitamente expressem suas preferências pelos produtos da plataforma (fotos).
- **Binária:** Na plataforma de compartilhamento de vídeos *YouTube*, os usuários têm as opções *Marcar como “gostei”* e *Marcar como “Não gostei”* disponíveis para expressar sua preferência por um determinado vídeo.
- **Numérica:** No site de vendas da *Amazon*, os usuários podem atribuir de 1 até 5 estrelas para classificar cada produto da plataforma. Essa classificação quantifica a apreciação do usuário por determinado produto.

Quando obtidas de um grande número de usuários, as preferências dos usuários pelos produtos têm um papel fundamental em SRs baseados em Filtragem Colabora-

¹ A taxa de conversão é uma métrica de marketing digital que mensura a quantidade de clientes que realizaram alguma ação que possa ser convertida em valor para o negócio (por exemplo, a assinatura de um serviço). No contexto de SRs, a taxa de conversão pode se dar pela quantidade de usuários que assinam o serviço após um período experimental utilizando um novo SR.

tiva (FC). FC é um dos métodos de recomendação mais bem sucedidos e estudados na literatura de SRs (LIU et al., 2014; CACHEDA et al., 2011; CHEN et al., 2018; ADOMAVICIUS; TUZHILIN, 2005; JALILI et al., 2018). De acordo com Cacheda et al. (2011), os métodos de FC podem ser classificados em duas categorias: algoritmos baseados em modelo e algoritmos baseados em memória. Os algoritmos baseados em modelo utilizam as preferências dos usuários na construção de modelos de fatores latentes com alta capacidade de generalização. Essas técnicas incluem fatoração de matrizes (KOREN; BELL; VOLINSKY, 2009; ZHOU et al., 2015; BOBADILLA et al., 2018; MA et al., 2011), modelos probabilísticos (VALDIVIEZO-DIAZ et al., 2019) e métodos baseados em clusterização (NAJAFABADI et al., 2017; MIRBAKHS; LING, 2018).

Os algoritmos baseados em memória se fundamentam na ideia de que os usuários que atribuíram preferências similares a produtos já observados², provavelmente possuem preferências similares para produtos ainda não observados (CHEN et al., 2018). Seguindo esse raciocínio, é necessária a construção de modelos de similaridade que permitam a identificação de usuários com preferências similares. De maneira geral, essa identificação é feita por meio de métricas de similaridade que mensuram a existência de uma relação linear entre as preferências atribuídas a produtos observados em comum por dois usuários. Com isso, busca-se aumentar a precisão na predição, que é um dos objetivos a serem alcançados na modelagem de SRs, uma vez que os usuários constroem uma relação de confiança com sistemas que geram recomendações mais precisas (AYUB et al., 2019).

Uma área desafiadora para a construção de SRs é o domínio da TV Digital Interativa (TVDI). Uma grande quantidade de conteúdo é disponibilizada diariamente para os usuários através de meios de distribuição de conteúdo como: difusão terrestre, satélite, cabo, IPTV (*Internet Protocol Television*) e TV pela Internet³. Essa grande quantidade de conteúdo reduz drasticamente a visibilidade de cada programa, potencialmente inibindo os usuários de encontrar conteúdo de seu interesse na TV (BAMBINI; CREMONESI; TURRIN, 2012). Nesse cenário de sobrecarga de conteúdo, a TVDI tornou-se uma área particularmente interessante para aplicações de SRs e vem sendo explorada em muitas pesquisas (VIEIRA; LINO, 2015; ALAM; KHUSRO; KHAN, 2019; LI et al., 2020).

Os avanços nas tecnologias de TVDI (que incluem, por exemplo, a adição de uma unidade de processamento, canal de retorno e capacidade de comunicação sem fio) permi-

² Neste trabalho, o termo “observado” (quando relacionado a um produto de um SR) se refere à qualidade de um produto que já sofreu a ação de ter uma preferência atribuída a ele por parte de um usuário. Por exemplo, se um usuário u avalia um produto o com uma nota qualquer, é dito que esse produto o já foi observado pelo usuário u .

³ IPTV e TV pela Internet são conceitos frequentemente confundidos, mas é importante distingui-los. Ambos disponibilizam conteúdo de TV sobre a rede IP, porém com diferentes níveis de controle e qualidade (ZAPATER; BRESSAN, 2007; MAISONNEUVE et al., 2009). Enquanto IPTV é um serviço de vídeo oferecido por um provedor de serviço através de uma rede proprietária com qualidade de serviço garantida, a TV pela Internet disponibiliza conteúdo através da Internet e sem garantia de qualidade de serviço para dispositivos que suportem *streaming* de vídeo.

tiram a concepção de um ambiente multi-plataforma onde o conteúdo pode ser entregue por meio de uma variedade de dispositivos computacionais como SmartTVs, dispositivos móveis e *Set-Top Boxes* (STBs) (VIEIRA; LINO, 2015). Esses avanços também incluem o desenvolvimento de uma miríade de serviços para TV, o que naturalmente enriquecem a experiência dos usuários com novas possibilidades de interação. Essas novas possibilidades de interação fazem da TVDI um ambiente conveniente para a obtenção de dados em abundância, o que é essencial para SRs baseados em FC (TEIXEIRA et al., 2010). Porém, as vantagens dessas novas possibilidades de interação esbarram no modelo de interação herdado da TV tradicional, onde um controle remoto convencional é compartilhado em um ambiente multiusuário.

1.1 Definição do Problema

Neste trabalho, dividimos o problema em duas partes. A primeira parte está relacionada à captura de dados de maneira organizada e individualizada em um ambiente de TVDI (*Problema 1*). A segunda parte está relacionada à otimização da recomendação de conteúdo por meio da utilização de uma vizinhança que seja mais relevante ao usuário de interesse e ao item recomendado (*Problema 2*).

1.1.1 Problema 1: Captura de Dados de Maneira Organizada e Individualizada

Os dados disponibilizado pelos usuários são uma das principais fontes de entrada de dados para alimentar os SRs voltados para a TVDI. Porém, muitas vezes, as novas tecnologias de TVDI esbarram no formato herdado da TV analógica tradicional. De acordo com Alam, Khusro e Khan (2019), dois fatores contribuem para limitar a interação em ambientes de TVDI:

- O amplo uso de modelos legados de controle remoto que acabam limitando a frequência e os tipos de interação em novos ambientes de TV. Em muitas situações, o controle remoto é utilizado somente para mudar o canal e ajustar o volume.
- O fato de tradicionalmente a TV ser um dispositivo especializado para a exibição de conteúdo audiovisual, que normalmente não estimula a interação por conta da própria natureza passiva do ambiente.

O resultado disso é que o formato de interação herdado da TV analógica tradicional acaba limitando o potencial de interatividade dos novos ambientes de TVDI. Modelos de interação mais avançados são forçados a serem simplificados para que se adaptem ao modelo tradicional de interação.

Além das dificuldades de interação, pesquisas revelam que aproximadamente 58% das pessoas assistem TV em um ambiente compartilhado com a família ou amigos, onde todos utilizam um único controle remoto (ABREU et al., 2013). Como resultado, o ambiente de TV se configura como um cenário desafiador para coletar os dados dos usuários de maneira organizada e individualizada (BAMBINI; CREMONESI; TURRIN, 2012). Utilizando somente um controle remoto, um grupo com usuários possivelmente seria tratado como um único usuário durante a captura de dados. Ao modelar o comportamento desse grupo a partir de suas interações, os usuários que interagem mais com a TV acabariam impondo suas preferências sobre as dos usuários que interagem menos. Por exemplo, considerando uma residência onde as crianças assistem programas infantis durante a maior parte do dia, uma aplicação que recomenda programas de TV recomendaria programas infantis para os adultos que assistem outros programas de TV por poucas horas durante o dia.

1.1.2 Problema 2: Recomendação de Conteúdo utilizando uma Vizinhança Relevante

Uma vez que seja possível obter os dados de interação dos usuários de maneira organizada e individualizada, a identificação de mecanismos para mensurar a similaridade entre usuários se torna uma tarefa crucial em algoritmos baseados em memória (LIU et al., 2014). Caso as métricas de similaridade não reflitam as correlações existentes entre os usuários, o resultado pode ser uma queda na precisão das recomendações. Normalmente, esses algoritmos utilizam as avaliações de produtos observados em comum por dois usuários para calcular os valores de similaridade. As similaridades calculadas a partir de todos os pares de usuários corresponde a uma matriz de similaridade.

Métricas tradicionais de similaridade incluem o Coeficiente de Correlação de Pearson (RESNICK et al., 1994; SARWAR et al., 2001; SHARDANAND; MAES, 1995) e a similaridade do Coseno (SARWAR et al., 2001). Complementando essas métricas, muitas pesquisas propõem novas métricas de similaridade com o intuito de melhorar a qualidade das recomendações (LIU et al., 2014; WANG et al., 2017; POLATIDIS; GEORGIADIS, 2016; ZHANG et al., 2016; AYUB et al., 2019). Essas métricas são essenciais para se identificar uma vizinhança⁴ composta pelos usuários mais similares que serão utilizados pelo modelo de recomendação.

No cenário mais convencional de cálculo da similaridade, todas as técnicas citadas anteriormente têm algo em comum: elas exploram a correlação existente entre dois usuários e tentam representá-la por meio de um valor global de similaridade. Usualmente,

⁴ Na literatura, a vizinhança de um usuário é conhecida como *K-Nearest Neighbors* (KNN) ou *k* vizinhos mais próximos. As técnicas baseadas em vizinhança selecionam até *k* usuários mais similares a um usuário de interesse para utilizar na função de agregação do modelo de predição.

todos os produtos considerados no cálculo da similaridade são tratados da mesma maneira e têm a mesma importância no valor final da similaridade. Porém, no mundo real, embora os usuários sejam livres para expressar suas preferências por uma variedade de produtos, eles tendem a ter mais simpatia por produtos em áreas específicas de seu interesse. Não por acaso, essas áreas de interesse concentram a maior parte dos produtos observados. Nesse cenário de similaridade global, os produtos contidos em áreas de interesse tendem a influenciar fortemente a correlação final entre dois usuários, se impondo sobre qualquer correlação diferenciada que possa existir em outras áreas de menor interesse.

Em um exemplo do mundo real que demonstra esse raciocínio, pode-se considerar um sistema de avaliação de filmes online onde os usuários atribuem notas para cada filme observado. Dois usuários expressam um interesse primário por filmes do gênero de ação, que concentra a maior parte dos filmes observados por esses usuários. Os filmes de outros gêneros representam uma porção menor de todo o conjunto de filmes observados. Ao se considerar que as preferências desses dois usuários convergem em filmes de ação, mas divergem em filmes do gênero comédia, tem-se uma situação que merece atenção em SRs. Uma vez que os filmes categorizados no gênero ação têm uma maior contribuição no valor final de similaridade entre esses dois usuários, pode-se dizer que eles são similares de uma maneira geral. Porém, ao considerar somente o gênero de comédia, esses dois usuários são dissimilares e, portanto, são inadequados para compor a vizinhança na predição de filmes do gênero comédia.

Esse cenário de similaridade global acaba expondo uma situação em que os usuários, embora semelhantes de uma maneira geral, podem não ser relevantes para formar a vizinhança em todas os casos. Ao se observar somente as categorias de produtos relacionados, as preferências dos usuários podem divergir fortemente, o que os torna inadequados para compor a vizinhança e pode conseqüentemente levar à queda na precisão da predição para alguns produtos.

1.2 Solução Proposta

Como solução para o *Problema 1*, busca-se utilizar um mecanismo que já faz parte da interação diária das pessoas e que tem emergido naturalmente como uma maneira de consumir conteúdo enquanto se assiste TV (tela primária). Esse mecanismo é conhecido como tela secundária e consiste na utilização de dispositivos móveis de computação pessoal (como *smartphones* e *tablets*) para executar diferentes atividades (relacionadas ou não ao conteúdo da tela principal) enquanto se assiste TV (VANATTENHOVEN; GEERTS, 2017; BROWN et al., 2019; LOHMÜLLER; WOLFF, 2019).

Atualmente, o uso de dispositivos móveis enquanto se assiste TV é comum e pro-

dutores de conteúdo vem explorando cada vez mais esses dispositivos como um meio de controlar e prover informações adicionais ao conteúdo da tela principal (BROWN et al., 2019). Diferentemente dos controles remotos convencionais, que são compartilhados e computacionalmente limitados, os dispositivos móveis são pessoais e apresentam capacidades mais avançadas de processamento computacional. Essas características pessoais e computacionais dos dispositivos móveis facilitam a distinção das pessoas que estão interagindo com a TV, o que conseqüentemente permite a captura de dados de maneira individualizada.

Como solução para o *Problema 2*, este trabalho propõe um método de FC baseada em memória para a construção de modelos de similaridade local para usuários. O método explora o uso de múltiplas correlações baseadas em grupos de produtos relacionados. Em vez de utilizar uma similaridade global que é obtida a partir da correlação entre todos os produtos, a abordagem proposta neste identifica grupos de produtos relacionados e cria modelos de similaridade baseados em memória para cada grupo. Dessa maneira, dado um usuário ativo u e um produto de interesse o , a vizinhança utilizada para prever a preferência do usuário u pelo produto o é obtida a partir do modelo de similaridade local correspondente ao grupo de produtos no qual o produto o está contido. Em outras palavras, a vizinhança utilizada para prever a preferência do usuário u pelo produto o considera somente o grupo de produtos que são relacionados ao produto de interesse.

O método proposto para resolver o *Problema 2* utiliza técnicas de clusterização para identificar os grupos de produtos relacionados. A abordagem também utiliza técnicas de normalização e um modelo de alocação de recursos baseado em informações estruturais de redes usuário-objeto para melhorar o modelo de similaridade local (ZHOU et al., 2007; JAVARI; GHARIBSHAH; JALILI, 2014; ZHOU et al., 2010; JALILI et al., 2018). Para simular as preferências dos usuários, são utilizadas 3 bases de dados na avaliação da performance do modelo. Os resultados são comparados com abordagens tradicionais e com o estado da arte de métricas de similaridade para FC. O modelo proposto é avaliado em termos de precisão utilizando métricas como o *Mean Absolute Error*⁵ (MAE) e *Root-mean-square Error*⁶ (RMSE).

1.3 Objetivos

A seguir, serão apresentados o objetivo geral e os objetivos específicos alcançados com o desenvolvimento deste trabalho.

⁵ *Erro Absoluto Médio em português.*

⁶ Raiz do Erro Quadrático Médio em português.

1.3.1 Objetivo Geral

O objetivo geral a ser atingido neste trabalho é propor, construir e avaliar um modelo de recomendação de conteúdo para TVDI baseado em FC, que utilize dados individuais de interação que possam ser obtidos por meio de dispositivos de tela secundária para encontrar uma vizinhança relevante de usuários que possa contribuir para a melhora na precisão da predição de preferências.

1.3.2 Objetivos Específicos

Para alcançar o objetivo geral proposto, os seguintes objetivos específicos foram definidos:

1. Investigar o estado da arte em sistemas de recomendação baseados em FC e sua aplicação no domínio da TVDI.
2. Construir um protótipo de um modelo de interação baseado em telas secundárias que permita a coleta de dados de interação dos usuários de maneira organizada e individualizada em ambientes de TVDI.
3. Criar cenários de utilização do protótipo construído e demonstrar sua aplicabilidade na coleta de dados para um ambiente de TV.
4. Investigar e definir maneiras de agregar os dados de interação dos usuários em um modelo de recomendação que considere a relevância da vizinhança para a predição das preferências.
5. Avaliar objetivamente o desempenho do modelo de recomendação desenvolvido em termos de precisão utilizando bases de dados reconhecidamente relevantes na literatura de SRs.
6. Comparar os resultados obtidos com o estado da arte em recomendação de conteúdo, de acordo com as métricas estudadas.
7. Publicar os resultados da pesquisa em revistas científicas relevantes para a comunidade.

1.4 Contribuições

As principais contribuições desse trabalho são:

- Uma abordagem baseada em telas secundárias para facilitar a captura de dados de interação dos usuários de maneira individualizada enquanto eles interagem com a TV.
- Um modelo de similaridade que considera grupos de produtos relacionados e explora múltiplas estruturas de correlação entre pares de usuários para aprimorar a seleção da vizinhança.
- Algoritmos que implementam um modelo de similaridade local inspirado no Coeficiente de Correlação de Pearson, no emprego de técnicas de normalização e na utilização de informações estruturais de redes usuário-objeto para melhorar a precisão da predição.
- Um modelo de recomendação baseado em técnicas de clusterização que é desacoplado da implementação do algoritmo de clusterização, permitindo a utilização de algoritmos que melhor se adaptem ao domínio dos dados e do problema.

1.5 Organização da Tese

Os outros capítulos desse trabalho destacam a importância dos temas relacionados, a proposta e os resultados alcançados. O capítulo 2 apresenta uma visão geral sobre SRs, assim como algumas técnicas empregadas em recomendações e métricas de avaliação. O capítulo 3 descreve algumas aplicações relacionadas à personalização de conteúdo para TVDI e tecnologias relacionadas. Além disso, o capítulo também apresenta algumas aplicações de dispositivos de tela secundária no domínio da TV. O capítulo 4 descreve as soluções propostas para o problema de captura de dados de maneira individualizada em ambientes de TVDI e para o problema da recomendação de conteúdo utilizando uma vizinhança relevante. Os resultados alcançados por meio dos experimentos realizados são apresentados no capítulo 5. Por fim, o capítulo 6 apresenta as considerações finais sobre este trabalho, assim como as contribuições científicas alcançadas.

Capítulo 2

Sistemas de Recomendação

ESTE capítulo apresenta uma visão geral dos principais conceitos relacionados ao tema de SRs. São apresentados exemplos de aplicações reais e os desafios enfrentados na pesquisas relacionadas ao tema de SRs. O capítulo também apresenta as definições e notações utilizadas ao longo do trabalho. Essas definições e notações são utilizadas como base para o entendimento de um conjunto de técnicas de recomendação assim como das métricas para a avaliação das predições realizadas pelos SRs.

2.1 Aplicações Reais

Existe uma abundância de aplicações práticas que reforçam a importância das tecnologias baseadas em SRs. Muitas dessas aplicações estão disponíveis na Internet e atingem uma variedade de nichos de mercado como vendas online, músicas, vídeo sob demanda, aplicativos móveis, redes sociais e páginas web. Empresas como *Google*, *Netflix* e *Amazon.com* têm apostado na utilização de SRs visando o aumento na margem de lucros e a melhora na satisfação dos clientes ao aproximá-los de produtos que possam ser de seu interesse (LINDEN; SMITH; YORK, 2003; HINZ; ECKERT, 2010; JANNACH; JUGOVAC, 2019). Exemplos claros disso são a apresentação de itens relacionados em áreas de destaque nas páginas web durante a utilização dos serviços das empresas.

Segundo nota publicada em seu blog oficial¹, a empresa *Google* utiliza um mecanismo de busca com resultados personalizados chamado *Search plus Your World*. Esse mecanismo analisa o histórico de buscas da conta pessoal dos usuários com o objetivo de

¹ <http://googleblog.blogspot.com.br/2012/01/search-plus-your-world.html>.

inferir as preferências de cada um. Dessa maneira, os usuários podem obter resultados mais úteis e relevantes na realização de buscas no futuro. Além disso, os resultados das buscas são agregados de conteúdo compartilhado por pessoas conectadas ao usuário na rede social da empresa. Assim, dois usuários que executam a mesma busca podem obter resultados diferentes, personalizados de acordo com o perfil de navegação e relacionamentos em redes sociais.

Outro caso de sucesso dos algoritmos de recomendação do *Google* é o algoritmo do *YouTube*. Esse algoritmo faz uso de um sistema de *deep learning* para processar a grande quantidade de dados que alimenta diariamente o serviço. Entre outras coisas, o algoritmo leva em consideração os dados dos vídeos já consumidos pelo usuário como “likes” e “dislikes”. Outros fatores que influenciam na recomendação dos vídeos da plataforma são os canais assinados pelo usuário, as pesquisas realizadas e as tendências de vídeos para um período. Para se ter ideia do impacto dos algoritmos de recomendação da plataforma, uma pesquisa realizada pelo *Pew Research Center*² revela que 81% dos usuários da plataforma dizem assistir os vídeos recomendados. De acordo com Neal Mohan (diretor de produtos do *YouTube*) em um painel de discussão na CES³ de 2018, mais de 70% do tempo de visualização da plataforma são provenientes dos vídeos sugeridos pelo algoritmo de recomendação⁴.

Também utilizando recomendações personalizadas, a empresa *Amazon.com* explora uma estratégia de varejo conhecida como cauda longa (ANDERSON, 2006; PARK; TUZHILIN, 2008; HAMEDANI; KAEDI, 2019). Essa estratégia consiste em explorar as vendas de uma grande variedade de itens onde cada um vende poucas unidades, ao invés de concentrar os esforços apenas nos itens que vendem mais. Em outras palavras, o volume acumulado de vendas dos itens menos populares tende a ser muito significativo quando comparado com o volume de vendas dos itens mais populares. De acordo com Brynjolfsson, Hu e Smith (2003), para alcançar os itens menos populares, a *Amazon.com* faz uso de pelo menos 7 SRs que auxiliam os usuários em suas compras. Em uma das técnicas, após a seleção de um item pelo usuário, o sistema analisa as compras realizadas por outros usuários e apresenta uma lista com itens similares em uma seção chamada: “Consumidores que compraram esse item também compraram” (LINDEN; SMITH; YORK, 2003; SMITH; LINDEN, 2017). Segundo Brynjolfsson, Hu e Smith (2003), utilizando essa estratégia no nicho de livros, por exemplo, de um total estimado em 2.300.000 títulos, aproximadamente 48% das unidades vendidas correspondem a títulos que estão abaixo do ranking dos 40.000 mais vendidos.

Um caso controverso do uso de SRs foi o da empresa de consultoria política e

² <https://www.pewresearch.org/internet/2018/11/07/many-turn-to-youtube-for-childrens-content-news-how-to-lessons/>

³ *Consumer Electronics Show*.

⁴ <https://www.cnet.com/news/huawei-employees-hit-with-us-visa-restrictions/>

análise de dados *Cambridge Analytica* que trabalhou para a campanha presidencial de Donald Trump nos Estados Unidos nas eleições de 2016. Esse caso ficou conhecido como um escândalo⁵ quando foi revelado que a empresa utilizou dados de mais de 50 milhões de usuários do *Facebook* para promover engajamento político. Os dados foram obtidos a partir de um teste de personalidade realizado na rede social. Cada usuário que participou do teste autorizou a coleta de informações não somente do próprio usuário, mas de todos os amigos na rede social. A *Cambridge Analytica* criou um sistema para segmentar os usuários e direcionar notícias personalizadas de acordo com cada segmento, tendo como principal objetivo a divulgação de conteúdo pró-Trump e conteúdo contrário à Hillary Clinton (adversária de Trump nas eleições) para influenciar eleitores indecisos. Uma das principais acusações contra a empresa é a de que esse sistema tenha permitido prever e influenciar as escolhas dos eleitores nas eleições presidenciais americanas de 2016.

No mercado brasileiro, algumas empresas têm oferecido serviços de recomendação no *e-commerce* de varejo. Entre elas é possível destacar a *Omnilogic*⁶, a *Neemu* e a *Chaordic*. Essas empresas basicamente analisam os dados obtidos a partir do comportamento dos consumidores nas lojas virtuais e inferem o interesse deles por produtos da loja. A *Omnilogic* é uma empresa focada no uso de tecnologias de inteligência de dados para otimizar os resultados das buscas em sistemas de *e-commerce*. A *Neemu* analisa (para cada consumidor) o fluxo de navegação, cliques, filtros usados e produtos adicionados ao carrinho de compras. Após essa análise, é criado um ranking personalizado de produtos para cada consumidor. Já a *Chaordic* utiliza técnicas de aprendizado de máquina para encontrar relações entre pessoas, produtos e serviços. Em 2015, *Neemu* e *Chaordic* foram compradas pela *Linx*, maior empresa de tecnologias de varejo na América Latina.

Na área de TVDI e tecnologias relacionadas, algumas empresas utilizam SRs na personalização de conteúdo audiovisual para os usuários. Exemplos disso incluem o *Chromecast* e *YouTube TV*, do *Google*, e os serviços oferecidos pelas empresas *ContentWise*⁷, *Ooyala*⁸ e *Netflix*. Ainda no domínio da TVDI, a empresa *Apple* possui um dispositivo set-top box chamado *AppleTV*. Esse dispositivo possui um serviço chamado *Genius* que sugere novos títulos de filmes e programas de TV baseando-se em compras realizadas anteriormente pelos consumidores. Além disso, os algoritmos do serviço também analisam os hábitos de visualização dos consumidores.

As empresas *ContentWise* e o *Ooyala* exploram o compartilhamento de conteúdo audiovisual através de múltiplos dispositivos e oferecem recomendações baseadas nas preferências do usuário e suas interações com redes sociais. As duas empresas têm em comum

⁵ <https://www.theguardian.com/technology/2018/mar/17/facebook-cambridge-analytica-kogan-data-algorithm>

⁶ <http://www.omnilogic.com.br>

⁷ <http://www.contentwise.tv>

⁸ <http://www.ooyala.com/>

a personalização de conteúdo audiovisual, porém a *ContentWise* vai além e inclui em seu catálogo recomendações de vídeo sob demanda (VoD⁹), programação de TV, jogos, música e aplicativos.

Netflix é uma empresa cujo principal negócio são as assinaturas baseadas em serviços de *streaming* online que disponibilizam uma variedade de filmes e séries de TV. Em 2006, a empresa lançou uma competição, chamada *Netflix Prize* (BENNETT; LANING, 2007), e liberou um conjunto de dados com 100 milhões de avaliações anônimas sobre filmes. Nessa competição, desenvolvedores e pesquisadores foram desafiados a desenvolver um algoritmo que superasse a precisão do sistema utilizado pela companhia, o *Cinematch*. Em 2009, um prêmio de \$ 1.000.000 foi concedido ao time que superou a precisão do *Cinematch* em 10%. Durante a apresentação de um tutorial na *ACM Conference on Recommender Systems* de 2009, Jon Sanders, engenheiro de SRs da *Netflix*, declarou que cerca de 60% dos DVDs alugados¹⁰ eram selecionados com base em recomendações personalizadas, demonstrando assim a importância dos SRs para a empresa. Desde 2013, a *Netflix* utiliza perfis de usuários que podem ser associados a usuários individuais ou temas (por exemplo, “filmes para a família”). Os perfis segregam os hábitos de visualização e cada perfil recebe sugestões individualizadas. De acordo com Todd Yellin, então vice presidente de inovação de produtos da *Netflix*, é importante entender os perfis dos usuários porque aproximadamente 75% a 80% do que as pessoas assistem na *Netflix* vêm de recomendações e não das buscas realizadas no serviço.

2.2 Desafios dos Métodos Existentes

Embora os SRs reconhecidamente adicionem valor aos negócios, vários são os desafios enfrentados para se alcançar recomendações de qualidade (JANNACH; JUGOVAC, 2019; CHEN et al., 2018). Várias pesquisas relacionadas à resolução desses problemas vem sendo desenvolvidas e focam basicamente na qualidade e desempenho dos SRs (LÜ et al., 2012; ADOMAVICIUS; TUZHILIN, 2005; BOBADILLA et al., 2013; AYUB et al., 2019). Alguns desses problemas são:

- **Esparsidade:** no mundo real, o número de usuários e produtos em um sistema pode ser extremamente grande. Os serviços de *streaming* da *Netflix* e o *e-commerce* da *Amazon* contam com milhões de clientes em todo o globo e milhares de produtos acessíveis a esses clientes. Porém, mesmo os usuários mais ativos observam somente um subconjunto (provavelmente pequeno) de todos os produtos disponíveis. Da

⁹ Do inglês *Video on Demand*.

¹⁰ O modelo de negócio inicial da empresa era venda e aluguel de DVDs online. Posteriormente, a empresa abandonou as vendas para focar no aluguel de DVDs. Em 2007, a empresa expandiu seus negócios e introduziu o serviço de *streaming* de mídia online.

mesma maneira, mesmo os produtos mais populares são observados por somente uma pequena parcela de todos os usuários disponíveis. Assim, os relacionamentos usuário-produto existentes representam um percentual extremamente pequeno de todas as possibilidades de relacionamento do sistema. Como resultado, a quantidade de dados disponíveis é normalmente muito pequena para se fazer uma predição de qualidade em algumas situações.

- **Escalabilidade:** embora as bases de dados para SRs sejam comumente esparsas, muitos serviços contam com milhões de usuários e produtos que acabam gerando um grande volume de dados. Um dos requisitos de bons SRs é ter a capacidade de lidar com grandes volumes de dados e prover recomendações em tempo real (AYUB et al., 2019). Porém, conforme o número de usuários e produtos aumenta, o custo computacional para calcular as predições também aumenta. Como resultado, muitos algoritmos podem se tornar inapropriados para cenários que exigem baixos tempo de resposta.
- **Cold-start:** esse problema ocorre quando novos usuários ou novos produtos são adicionados ao sistema. Inicialmente, novos usuários e novos produtos não possuem dados históricos de interação no sistema e é difícil prover recomendações para esses usuários ou fazer com que esses novos produtos sejam recomendados. Certamente é necessário algum tempo até que o novo usuário interaja com o sistema e o novo produto receba algum tipo de interação.
- **Superespecialização:** ocorre quando o sistema é capaz de recomendar somente produtos muito similares ao perfil do usuário. Em muitos casos, os produtos recomendados são populares e fáceis de serem encontrados mesmo sem o auxílio de um SR. Isso acaba não adicionando o valor esperado de um SR e os usuários ficam limitados a receber recomendações de produtos óbvios. Um bom sistema de recomendação deve levar em conta a diversidade e recomendar produtos menos óbvios ou difíceis de serem encontrados ao mesmo tempo que atende às preferências dos usuários.

2.3 Notações Básicas de Sistemas de Recomendação

A representação de SRs através de grafos pode ser uma ferramenta versátil para a análise e compreensão de tais sistemas. De acordo com Rosen (2006), grafos são estruturas formadas por um conjunto de elementos, conhecidos como vértices, ligados entre si através de conexões, conhecidas como arestas. Formalmente, grafos seguem a seguinte definição:

Definição 1. (Grafo) Um grafo $G = (\mathcal{V}, \mathcal{E})$ consiste em \mathcal{V} , um conjunto não vazio de vértices, e \mathcal{E} , um subconjunto de $\mathcal{V} \times \mathcal{V}$ onde os pares são elementos distintos pertencentes a \mathcal{V} , conhecidos como arestas.

Em algumas situações os grafos podem ser utilizados para modelar relações entre elementos de duas classes diferentes. Por exemplo, uma rede de publicações científicas, onde autores e artigos são representados por duas classes diferentes de vértices e a autoria é representada por uma aresta entre um autor e um artigo científico. Outro exemplo inclui uma rede de compras de *e-commerce*, onde usuários e produtos são os vértices e o relacionamento entre um usuário e um produto representa uma compra. Esse tipo de grafo é frequentemente utilizado na representação de SRs e é conhecido como *grafo bipartido*. Formalmente, grafos bipartidos seguem a seguinte definição:

Definição 2. (Grafo bipartido) Um grafo $G(\mathcal{V}, \mathcal{E})$ é bipartido se existirem dois conjuntos não vazios distintos \mathcal{V}_1 e \mathcal{V}_2 , tal que, $\mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$, $\mathcal{V}_1 \cup \mathcal{V}_2 = \mathcal{V}$ e cada aresta em \mathcal{E} conecta um vértice em \mathcal{V}_1 a um vértice em \mathcal{V}_2 .

Uma classe especial de grafo bipartido é conhecida como *rede usuário-objeto*¹¹ baseada em web e desempenha um papel importante em muitos serviços online que requerem a interação dos usuários, principalmente em SRs (SHANG et al., 2010). A Figura 1 exibe uma representação gráfica de uma rede usuário-objeto em um grafo bipartido. De acordo com Shang et al. (2010), as redes usuário-objeto baseadas em web possuem duas características que as diferem de outras redes:

- Os usuários são ativos e os objetos são passivos. Ou seja, os usuários executam a ação de selecionar¹² os objetos, enquanto os objetos são selecionados pelos usuários.
- As conexões entre os usuários e os objetos podem ser geradas a qualquer momento à medida que os usuários selecionam os objetos. Essa característica não ocorre, por exemplo, em uma rede que conecta autores a artigos científicos, onde um autor não pode ser adicionado após a publicação do artigo.

No cenário mais usual, usuários e objetos de um SR são conjuntos finitos e distintos. $\mathcal{U} = \{u_1, u_2, u_3, \dots, u_m\}$ denota o conjunto de todos os usuários, e $\mathcal{O} = \{o_1, o_2, o_3, \dots, o_n\}$ representa o conjunto de todos os objetos. \mathcal{U}_{o_k} representa o conjunto de todos os usuários que selecionaram o objeto o_k . De maneira recíproca, \mathcal{O}_{u_i} representa o conjunto de todos os

¹¹ “Objeto” é o termo utilizado para representar qualquer item que possa ser recomendado para um usuário. Esses itens incluem livros, músicas, vídeos, filmes, etc. No restante deste trabalho, o termo “objeto” será utilizado para se referir a esses itens de recomendação.

¹² O termo “selecionar” pode ser utilizado para descrever a ação de um usuário em relação a um objeto. Essa ação usualmente expressa uma preferência implícita ou explícita que pode ser uma compra, avaliação, *like/dislike*, ou mesmo uma visualização.

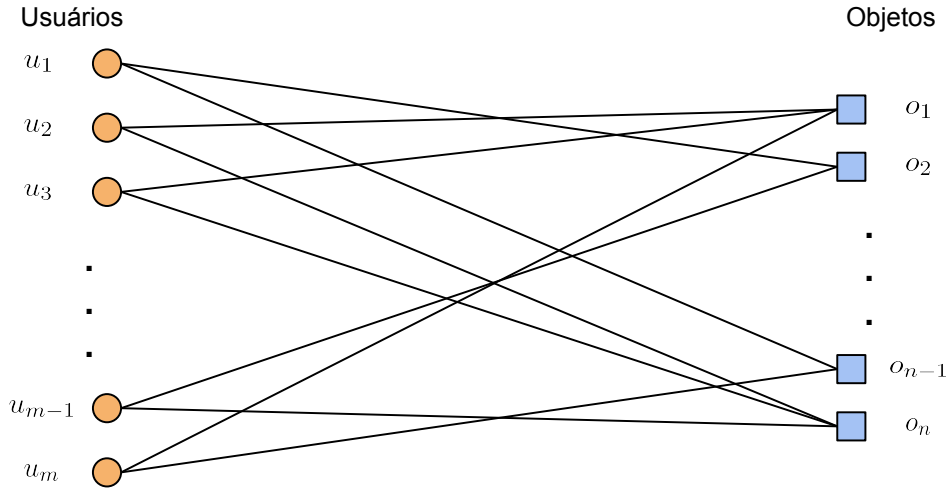


Figura 1: Grafo bipartido representando uma rede usuário-objeto.

objetos selecionados pelo usuário u_i . O número total de usuários no conjunto \mathcal{U} é denotado por $\#\mathcal{U} = m$ e o número total de objetos no conjunto \mathcal{O} é dado por $\#\mathcal{O} = n$. O número total de elementos em qualquer conjunto é representado pela mesma notação.

O relacionamento entre usuários e objetos é tipicamente feito por meio de avaliações. Cada avaliação corresponde a um valor numérico, que indica o grau de preferência de um usuário por um dado objeto. O conjunto de todos os valores possíveis de avaliações é normalmente dado por $\mathcal{X} = \{1, 2, 3, 4, 5\}$, onde 1 é a pior avaliação e 5 é a melhor. O conjunto de todas as preferências possíveis do sistema é representado por uma matriz de avaliações $\mathcal{U} \times \mathcal{O} \rightarrow R$, onde as linhas da matriz denotam os usuários e as colunas denotam os objetos.

O valor de cada célula da matriz de avaliações $R = \{r_{u_i o_k}\}^{m \times n}$ consiste em avaliações $r_{u_i o_k} \in \mathcal{X} \cup \{\emptyset\}$ para todo usuário $u_i \in \mathcal{U}$ e objeto $o_k \in \mathcal{O}$. Enquanto $r_{u_i o_k} \in \mathcal{X}$ indica que o usuário u_i selecionou o objeto o_k e atribuiu uma avaliação, a notação $r_{u_i o_k} \in \{\emptyset\}$ indica que o usuário u_i ainda não avaliou o objeto o_k . A predição da preferência de um usuário ativo¹³ u_i por um objeto de interesse¹⁴ o_k é dada por $\hat{r}_{u_i o_k}$.

2.4 Métodos de Recomendação

Os SRs tentam prever as preferências dos usuários por objetos baseando-se em conjuntos de dados distintos. Esses dados podem ser informações dos usuários (perfil do usuário com informações demográficas e preferências explícitas), histórico de interação (que o usuário avaliou, comprou, visualizou, etc.) ou informações dos objetos. Para fazer as predições, são utilizados diferentes métodos de recomendação que normalmente são

¹³ Usuário para o qual se pretende fazer a predição de uma preferência para recomendação.

¹⁴ Objeto para o qual se pretende fazer uma predição de preferência para recomendação.

classificados nas seguintes categorias (ADOMAVICIUS; TUZHILIN, 2005): *abordagens baseadas em colaboração* (também conhecidas como FC), *abordagens baseadas em conteúdo* e *abordagens híbridas*. As seções a seguir apresentam algumas métricas de similaridade e uma visão geral sobre cada categoria de métodos de recomendação.

2.4.1 Abordagens Baseadas em Colaboração

As abordagens baseadas em colaboração (ou FC) são conhecidas como algumas das mais bem sucedidas em SRs (LIU et al., 2014; CHEN et al., 2018; ADOMAVICIUS; TUZHILIN, 2005). De maneira geral, sistemas baseados em colaboração tentam sugerir novos objetos para um usuário tomando como base as avaliações atribuídas por outros usuários no passado (SCHAFER et al., 2007; HERLOCKER et al., 2004). Mais formalmente, uma preferência $\hat{r}_{u_i o_k}$ de um usuário u_i em um objeto o_k é estimada com base nas preferências $r_{u_j o_k}$ de outros usuários $u_j \in \mathcal{U} | u_j \neq u_i$ pelo objeto o_k .

Muitos sistemas baseados em filtragem colaborativa foram desenvolvidos na academia e na indústria. De acordo com Resnick e Varian (1997), o termo *filtragem colaborativa* foi introduzido pelos pesquisadores que desenvolveram o primeiro sistema comercial de recomendação, o *Tapestry* em 1992 (GOLDBERG et al., 1992).

O *Tapestry* foi um sistema experimental desenvolvido na *Xerox PARC*¹⁵ em que a principal motivação para sua construção foi evitar que os usuários fossem sobrecarregados com o grande volume de documentos que chegavam até eles através de grupos de notícias. O princípio básico da utilização do *Tapestry* consistia em possibilitar que os usuários anotassem suas opiniões a respeito das mensagens lidas, e permitir que essas opiniões fossem acessadas por outras pessoas. Além disso, o conteúdo das mensagens dos grupos era armazenado no sistema juntamente com informações como assunto, emissor e destinatário. Então, os usuários do sistema poderiam manualmente definir suas preferências e os seus usuários de interesse por meio da criação de filtros de mensagens. Por exemplo, alguém poderia criar uma expressão para filtrar mensagens que contenham as palavras “verificação” e “formal”, que tenham sido aprovadas pela moderadora “Alice” e que também tenham sido marcadas como “interessante” pelo usuário “Bob”.

Diferentemente da abordagem manual do *Tapestry*, outros sistemas que surgiram posteriormente automatizaram a identificação dos usuários de interesse e a predição das preferências com o objetivo de gerar recomendações também de maneira automatizada. Entre os primeiros SRs automatizados, podem-se destacar os seguintes: *EachMovie* e *MovieLens* para filmes (HARPER; KONSTAN, 2015), *Ringo* para músicas e artistas (SHARDANAND; MAES, 1995), e *Bellcore Video Recommender* para vídeos (HILL et al., 1995).

¹⁵ Palo Alto Research Center.

Desde o surgimento dos primeiros SRs baseados em FC, várias técnicas de recomendação têm sido desenvolvidas. Essas técnicas são classificadas de acordo com a estratégia adotada pelos algoritmos (BREESE; HECKERMAN; KADIE, 1998; ADOMAVICIUS; TUZHILIN, 2005). Usualmente, esses algoritmos são divididos em duas categorias:

- *Algoritmos baseados em memória:* calculam as predições a partir das preferências de uma vizinhança formada pelos usuários mais similares a um usuário ativo.
- *Algoritmos baseados em modelo:* aprendem um modelo a partir do histórico de atividades dos usuários e usa esse modelo para prever as preferências dos usuários por novos objetos.

2.4.1.1 Algoritmos Baseados em Memória

Algoritmos baseados em memória essencialmente fazem uso de heurísticas para prever a avaliação de um usuário ativo com relação a um objeto de interesse. Essas heurísticas se baseiam nas preferências de outros usuários pelo objeto de interesse, dado que esses usuários são selecionados de acordo com algum critério de similaridade. Em outras palavras, dado que $\hat{r}_{u_i o_k}$ seja o valor estimado da preferência do usuário u_i pelo objeto o_k , esse valor é calculado como a agregação das preferências dos usuários $u_j \in \mathcal{N}_{u_i o_k}$, onde $\mathcal{N}_{u_i o_k}$ representa o conjunto formado pela vizinhança do usuário u_i para o objeto o_k . Ou seja, os usuários mais similares ao usuário u_i que também avaliaram o objeto o_k . Usualmente, os algoritmos de FC selecionam no máximo k usuários mais similares para formar a vizinhança.

Uma das formas mais simples de função de agregação é apresentada na Equação (2.1). Nesse caso, o valor resultante é simplesmente a média dos valores de preferência dos usuários mais similares ao usuário u_i .

$$\hat{r}_{u_i o_k} = \frac{1}{\#\mathcal{N}_{u_i o_k}} \cdot \sum_{u_j \in \mathcal{N}_{u_i o_k}} r_{u_j o_k} \quad (2.1)$$

Segundo Adomavicius e Tuzhilin (2005), a função de agregação mais utilizada em SRs utiliza a soma ponderada dos valores de preferência da vizinhança. Essa função é apresentada na Equação (2.2).

$$\hat{r}_{u_i o_k} = \frac{\sum_{u_j \in \mathcal{N}_{u_i o_k}} s_{u_i u_j} \cdot r_{u_j o_k}}{\sum_{u_j \in \mathcal{N}_{u_i o_k}} |s_{u_i u_j}|} \quad (2.2)$$

onde $s_{u_i u_j}$ é o valor de similaridade entre os usuários u_i e u_j . A Seção 2.5 apresenta algumas métricas de similaridade comumente utilizadas em métodos de FC.

Resnick et al. (1994) propôs uma função que é uma variação da Equação (2.2). Essa nova função utiliza a soma ponderada das preferências desviadas da média. A fórmula é descrita conforme a Equação (2.3).

$$\hat{r}_{u_i o_k} = \bar{r}_{u_i} + \frac{\sum_{u_j \in \mathcal{N}_{u_i o_k}} (r_{u_j o_k} - \bar{r}_{u_j}) \cdot s_{u_i u_j}}{\sum_{u_j \in \mathcal{N}_{u_i o_k}} |s_{u_i u_j}|} \quad (2.3)$$

onde \bar{r}_{u_i} é a média das preferências do usuário u_i e a avaliação desviada da média ($r_{u_j o_k} - \bar{r}_{u_j}$) é ponderada pela similaridade entre um usuário ativo u_i e um vizinho u_j .

Funções de agregação alternativas podem considerar a variação na escala de preferências para diferentes usuários, uma vez que cada usuário pode adotar um sistema particular de avaliações. Enquanto alguns usuários atribuem avaliações ruins somente para objetos extremamente desagradáveis, outros usuários podem escolher atribuir boas avaliações somente para objetos notavelmente apreciados. Ao considerar essa diversidade na distribuição das avaliações entre os usuários, pode ser interessante utilizar mecanismos de normalização das preferências dos usuários. Uma função de normalização muito popular em técnicas de SRs é conhecida como *z-score* (CACHEDA et al., 2011; HERLOCKER; KONSTAN; RIEDL, 2002). Essa função é descrita conforme a Equação (2.4).

$$z_{u_i o_k} = \frac{r_{u_i o_k} - \bar{r}_{u_i}}{\sigma_{u_i}} \quad (2.4)$$

onde σ_{u_i} é o desvio padrão das preferências do usuário u_i .

Estudos mostram que a aplicação de procedimentos de normalização utilizando a função *z-score* melhora os resultados de precisão na predição (HERLOCKER; KONSTAN; RIEDL, 2002). A função de agregação utilizando *z-score* é computada conforme a Equação (2.5).

$$\hat{r}_{u_i o_k} = \bar{r}_{u_i} + \sigma_{u_i} \cdot \frac{\sum_{u_j \in \mathcal{N}_{u_i o_k}} z_{u_j o_k} \cdot s_{u_i u_j}}{\sum_{u_j \in \mathcal{N}_{u_i o_k}} |s_{u_i u_j}|} \quad (2.5)$$

De maneira geral, algoritmos baseados em memória são mais simples que outras abordagens utilizadas em SRs (CACHEDA et al., 2011). Apesar da simplicidade, eles apresentam resultados interessantes quanto à precisão das predições realizadas. Um dos problemas dessa categoria de algoritmos está relacionado à escalabilidade. Embora os recursos de computação e armazenamento estejam cada vez mais baratos atualmente, os algoritmos não evoluíram para processar de maneira online um volume de dados cada vez maior. Estratégias para superar esse problema envolvem o particionamento dos dados e

a amostragem de usuários e objetos, o que acaba impactando diretamente na qualidade das recomendações.

2.4.1.2 Algoritmos Baseados em Modelo

Diferentemente dos algoritmos baseados em memória, os algoritmos baseados em modelo baseiam-se na construção de modelos de fatores latentes com alta capacidade de generalização para prever as preferências para objetos ainda não avaliados. Em outras palavras, os algoritmos baseados em modelo consistem em aprender um modelo de predição a partir da análise estatística dos dados e/ou utilização de técnicas de aprendizagem. Algumas das técnicas baseadas em modelo mais bem sucedidas incluem fatoração de matrizes (KOREN; BELL; VOLINSKY, 2009; ZHOU et al., 2015; BOBADILLA et al., 2018), modelos probabilísticos (VALDIVIEZO-DIAZ et al., 2019) e métodos baseados em clusterização (NAJAFABADI et al., 2017; MIRBAKHSI; LING, 2018).

Uma das técnicas baseadas em modelo mais famosas em SRs foi popularizada por Simon Funk¹⁶ durante a competição *Netflix Prize*. Essa técnica é similar à técnica conhecida como *Singular Values Decomposition* (SVD) e trata-se de um problema de otimização que utiliza o algoritmo gradiente descendente. O objetivo é encontrar matrizes P e Q (fatores latentes dos usuários e objetos) de tal maneira que $R \approx P \cdot Q$, onde R é a matriz de avaliações.

Além das matrizes de fatores latentes, a técnica adiciona vieses de modo que a predição $\hat{r}_{u_i o_k}$ é definida como:

$$\hat{r}_{u_i o_k} = \mu + b_{u_i} + b_{o_k} + q_{o_k}^T \cdot p_{u_i} \quad (2.6)$$

onde μ é a média de todas as avaliações do sistema. b_{u_i} e b_{o_k} são os vieses do usuário u_i e do objeto o_k , respectivamente. p_{u_i} e q_{o_k} são os vetores com os fatores latentes do usuário u_i e do objeto o_k , respectivamente.

O algoritmo de aprendizagem consiste em minimizar o erro quadrático regularizado utilizando o conjunto de dados de treinamento, conforme definido na Equação (2.7).

$$\sum_{r_{u_i o_k} \in R_{train}} (r_{u_i o_k} - \hat{r}_{u_i o_k})^2 + \lambda (b_{u_i}^2 + b_{o_k}^2 + \|q_{o_k}\|^2 + \|p_{u_i}\|^2) \quad (2.7)$$

onde λ é o fator de regularização.

O algoritmo gradiente descendente estocástico é utilizado para minimizar a Equação (2.7) em relação às diferentes variáveis. A atualização das variáveis para a minimização

¹⁶ <https://sifter.org/~simon/journal/20061211.html>

é executada conforme as equações a seguir:

$$\begin{aligned}
 b_{u_i} &\leftarrow b_{u_i} + \gamma \cdot (e_{u_i o_k} - \lambda \cdot b_{u_i}) \\
 b_{o_k} &\leftarrow b_{o_k} + \gamma \cdot (e_{u_i o_k} - \lambda \cdot b_{o_k}) \\
 p_{u_i} &\leftarrow p_{u_i} + \gamma \cdot (e_{u_i o_k} \cdot q_{o_k} - \lambda \cdot p_{u_i}) \\
 q_{o_k} &\leftarrow q_{o_k} + \gamma \cdot (e_{u_i o_k} \cdot p_{u_i} - \lambda \cdot q_{o_k})
 \end{aligned} \tag{2.8}$$

onde $e_{u_i o_k} = r_{u_i o_k} - \hat{r}_{u_i o_k}$ e γ é a taxa de aprendizado. As equações de atualização são executadas em todo o conjunto de dados de treinamento por uma quantidade de épocas. Os vieses são inicializados com 0 e as matrizes de fatores latentes são inicializadas com valores aleatórios seguindo uma distribuição normal.

2.4.2 Abordagens Baseadas em Conteúdo

Algoritmos de recomendação baseados em conteúdo têm sua base conceitual estabelecida nas técnicas de recuperação de informação, que proveem os métodos necessários para busca e mineração de dados em informação textual (PAZZANI; BILLSUS, 2007). Diferentemente de SRs baseados em filtragem colaborativa, que utilizam dados obtidos a partir das interações dos usuários, SRs baseados em conteúdo utilizam dados sobre objetos. Para realizar as recomendações, os sistemas verificam o conteúdo dos objetos observados anteriormente e buscam recomendar outros objetos que possuam conteúdo semelhante.

Os objetos são caracterizados por um conjunto pré-definido de características relacionadas ao domínio do sistema. Em um SR para filmes, por exemplo, as características de um objeto poderiam ser definidas como título, gênero, elenco, diretor e etc. Uma maneira bastante convencional de representar as características de um objeto o_k em recomendações baseadas em conteúdo é por meio de um vetor n -dimensional \vec{w}_{o_k} , conforme apresentado abaixo:

$$\vec{w}_{o_k} = (w_{1o_k}, w_{2o_k}, \dots, w_{no_k}) \tag{2.9}$$

onde cada elemento w_{jo_k} representa o peso de uma característica referente ao conteúdo do objeto o_k .

Um perfil dos usuários mantém informações sobre as suas preferências. Esse perfil pode ser construído explicitamente ou pode ser aprendido por meio das interações dos usuários com os objetos. Cada perfil de usuário também pode ser representado por um vetor de características n -dimensional $\vec{w}_{u_i} = (w_{1u_i}, w_{2u_i}, \dots, w_{nu_i})$. As abordagens baseadas em conteúdo utilizam esse perfil para determinar o grau de interesse de um usuário

por um objeto e recomendar os objetos mais interessantes. O grau de interesse de um usuário u_i por um objeto o_k pode ser determinado por meio de métricas de similaridade que medem o grau de correlação entre os vetores \vec{w}_{u_i} e \vec{w}_{o_k} . Ou seja, a similaridade entre o vetor que representa o usuário e o vetor que representa o conteúdo.

2.4.3 Abordagens Híbridas

As abordagens de recomendação híbridas consistem na combinação de métodos baseados em colaboração e métodos baseados em conteúdo. Com isso, busca-se aproveitar as melhores características e também superar as limitações individuais de cada abordagem. A Tabela 1 apresenta as vantagens e desvantagens individuais de cada abordagem (BARRAGÁNS-MARTÍNEZ et al., 2010).

Tabela 1: Comparação entre as abordagens baseadas em colaboração e abordagens baseadas em conteúdo.

	Colaboração	Conteúdo
Pontos positivos	<ul style="list-style-type: none"> • Pode trabalhar com qualquer tipo de objeto • Favorece a diversidade de recomendações • Sem necessidade de metadados de conteúdo 	<ul style="list-style-type: none"> • <i>Cold-start</i> não afeta a qualidade das recomendações • Baixo custo computacional para predição
Pontos negativos	<ul style="list-style-type: none"> • A esparsidade pode reduzir a qualidade das predições • Problemas de escalabilidade para prever as preferências dos usuários • <i>Cold-start</i> afeta a qualidade das recomendações 	<ul style="list-style-type: none"> • Necessidade de conteúdo disponível para análise • Superespecialização das recomendações • Alto custo computacional para a criação do modelo

Existem diferentes estratégias para combinar abordagens baseadas em colaboração e abordagens baseadas em conteúdo na construção de um sistema híbrido. Adomavicius e Tuzhilin (2005) descrevem quatro dessas estratégias:

- Obter as predições de métodos colaborativos e baseados em conteúdo de maneira separada e então combiná-las durante a recomendação.

- Incorporar características de métodos baseados em conteúdo em sistemas colaborativos.
- Incorporar características colaborativas em sistemas baseados em conteúdo.
- Construir um modelo unificado que incorpore as características de métodos colaborativos e baseados em conteúdo.

2.5 Métricas de Similaridade

O princípio básico das abordagens colaborativas baseadas em memória é a identificação e a utilização de usuários que compartilham opiniões semelhantes para predizer as preferências de outros usuários (LATHIA; HAILES; CAPRA, 2008). Assim, as opiniões de um grupo de usuários considerados semelhantes são agregadas de maneira que possam gerar opiniões aproximadas das preferências reais de um dado usuário ativo.

A similaridade é representada por meio de uma medida de correlação entre pares de usuários. Desse modo, os relacionamentos entre os usuários de um SR podem ser vistos como uma rede onde cada usuário é um vértice e as conexões entre os pares de usuários são arestas. Cada aresta possui um valor que representa a distância (ou semelhança) entre os usuários que formam o par. De maneira formal, o conceito de similaridade entre usuários segue a seguinte definição:

Definição 3. (*Similaridade*) Em um SR, $sim(u_i, u_j)$ é uma função que estima a similaridade entre um usuário u_i e um usuário u_j . A matriz de similaridade é definida como $sim : \mathcal{U} \times \mathcal{U} \rightarrow S$, onde os valores de similaridade para todos os usuários são denotados por $S = \{s_{u_i u_j}\}^{m \times m}$ para todo usuário $u_i \in \mathcal{U}$ e $u_j \in \mathcal{U}$.

De acordo com Amatriain et al. (2011), uma das abordagens baseada em similaridade mais utilizadas em sistemas de filtragem colaborativa é a utilização do classificador dos k vizinhos mais próximos (KNN¹⁷). Essa abordagem assume intuitivamente que os usuários ou objetos mais próximos em distância são potencialmente mais similares. O cálculo da similaridade depende das preferências que os usuários atribuíram aos objetos e da definição de uma métrica apropriada de similaridade que faça uso dessas preferências. Após o cálculo da similaridade, a vizinhança de um usuário é formada por um subconjunto dos k usuários que possuem os maiores valores de similaridade (ou seja, os usuários mais próximos). Os usuários que compõem a vizinhança de um dado usuário podem variar dependendo da métrica de similaridade adotada.

¹⁷ Do inglês *K-Nearest Neighbors*.

2.5.1 Métricas Tradicionais

Uma das técnicas utilizadas e mais simples para o cálculo de similaridade de uma maneira geral é a *Distância Euclidiana* (COSTA et al., 2007). A equação para o cálculo da distância euclidiana é apresentada a seguir:

$$s_{u_i u_j}^{\text{EUC}} = \sqrt{\sum_{o_k \in \mathcal{O}_{u_i u_j}} (r_{u_i o_k} - r_{u_j o_k})^2} \quad (2.10)$$

onde $\mathcal{O}_{u_i u_j} = \mathcal{O}_{u_i} \cap \mathcal{O}_{u_j}$ é o conjunto dos objetos que foram observados em comum pelos usuários u_i e u_j .

Em SRs, embora existam várias técnicas de similaridade, o *Coefficiente de Correlação de Pearson* (PCC^{18}). (RESNICK et al., 1994; SHARDANAND; MAES, 1995; COSTA et al., 2007) e a *similaridade do cosseno* (SARWAR et al., 2001) são duas das técnicas mais populares. Ambas as técnicas também são muito exploradas na literatura de SRs baseados em colaboração.

O PCC é calculado conforme a Equação (2.11). Essa métrica mede a força e a direção da correlação linear entre dois usuários. A interpretação da correlação leva em consideração o valor do coeficiente, que pode variar de -1 até $+1$. O valor $+1$ significa forte correlação positiva entre os usuários envolvidos. Ou seja, se um usuário avalia um objeto de uma maneira, o outro usuário tende a avaliar da mesma maneira. Por exemplo, dados dois usuários u_i e u_j positivamente correlacionados, se um usuário u_i atribui uma preferência alta para um objeto o_k , o usuário u_j tende a atribuir uma preferência também alta para o objeto o_k . Já o valor -1 significa forte correlação negativa entre os usuários. Ou seja, se um usuário avalia um objeto de uma maneira, o outro usuário tende a avaliar o mesmo objeto de maneira contrária. Valores próximos a 0 indicam fraca correlação entre os usuários. Na prática, formar uma vizinhança com usuários positivamente correlacionados tende a melhorar a precisão das predições, enquanto que formar uma vizinhança com usuários negativamente correlacionados não resulta em melhoras significativas em termos de precisão (HERLOCKER et al., 1999).

$$s_{u_i u_j}^{\text{PCC}} = \frac{\sum_{o_k \in \mathcal{O}} (r_{u_i o_k} - \bar{r}_{u_i})(r_{u_j o_k} - \bar{r}_{u_j})}{\sqrt{\sum_{o_k \in \mathcal{O}} (r_{u_i o_k} - \bar{r}_{u_i})^2} \sqrt{\sum_{o_k \in \mathcal{O}} (r_{u_j o_k} - \bar{r}_{u_j})^2}} \quad (2.11)$$

A similaridade do cosseno é outro exemplo de métrica tradicional bastante utilizada na literatura. Essa técnica consiste na representação dos usuários como vetores de preferências em um espaço n -dimensional onde $n = \#\mathcal{O}$. Assim, o usuário u_i é representado pelo vetor $\vec{u}_i = (r_{u_i o_1}, r_{u_i o_2}, r_{u_i o_3}, \dots, r_{u_i o_n})$. A similaridade do cosseno estima o

¹⁸ Do inglês *Pearson's Correlation Coefficient*.

ângulo do cosseno entre dois vetores \vec{u}_i e \vec{u}_j . O valor resultante pode variar de 0 a 1, onde valores mais baixos significam menor similaridade entre os usuários e valores mais altos significam maior similaridade. A Equação (2.12) apresenta a similaridade do cosseno.

$$\begin{aligned}
 s_{u_i u_j}^{\text{COS}} &= \cos(\vec{u}_i, \vec{u}_j) \\
 &= \frac{\langle \vec{u}_i, \vec{u}_j \rangle}{\|\vec{u}_i\| \cdot \|\vec{u}_j\|} \\
 &= \frac{\sum_{k \in \mathcal{O}} r_{u_i o_k} \cdot r_{u_j o_k}}{\sqrt{\sum_{k \in \mathcal{O}} (r_{u_i o_k})^2} \sqrt{\sum_{k \in \mathcal{O}} (r_{u_j o_k})^2}}
 \end{aligned} \tag{2.12}$$

onde $\langle \vec{u}_i, \vec{u}_j \rangle$ denota o produto interno entre os vetores \vec{u}_i e \vec{u}_j . O termo $\|\vec{u}_i\|$ denota o a norma do vetor \vec{u}_i .

2.5.2 Outras Métricas de Similaridade

Outras métricas, muitas vezes referenciadas como o estado da arte em métricas de similaridade, foram propostas como adaptações das técnicas tradicionais (AYUB et al., 2019). Duas dessas técnicas são o *Weighted PCC* (WPCC) (HERLOCKER et al., 1999) e *Sigmoid PCC* (SPCC) (JAMALI; ESTER, 2009).

A métrica WPCC melhora o PCC tradicional ao dar mais peso para usuários cujo número de objetos observados em comum é maior que um certo limiar. Essa métrica considera que quanto maior o número de objetos selecionados em comum, mais verossímil é o valor da similaridade. O WPCC é calculado a partir do PCC tradicional, de acordo com a Equação (2.13).

$$s_{u_i u_j}^{\text{WPCC}} = \begin{cases} \frac{\#\mathcal{O}_{u_i u_j}}{H} \cdot s_{u_i u_j}^{\text{PCC}} & , \#\mathcal{O}_{u_i u_j} < H \\ s_{u_i u_j}^{\text{PCC}} & , \text{caso contrário} \end{cases} \tag{2.13}$$

onde H é o valor do limiar, que é normalmente definido como 50, de acordo com Herlocker et al. (1999).

A métrica SPCC segue a mesma ideia do WPCC. Porém, ela inclui uma função sigmoideal que enfatiza a importância de objetos observados em comum. O SPCC também é calculado a partir do PCC tradicional, de acordo com a Equação (2.14).

$$s_{u_i u_j}^{\text{SPCC}} = \frac{1}{1 + \exp\left(-\frac{\#\mathcal{O}_{u_i u_j}}{2}\right)} \cdot s_{u_i u_j}^{\text{PCC}} \tag{2.14}$$

2.5.3 Similaridade Baseada em Objetos

As técnicas de similaridade apresentadas até aqui são baseadas somente na obtenção da correlação entre dois usuários distintos. Porém, também é possível calcular a similaridade entre dois objetos. Na similaridade baseada em objetos, supõe-se que objetos que compartilham o mesmo padrão de preferências por diferentes usuários são provavelmente similares.

O cálculo da similaridade entre dois objetos funciona da mesma maneira que para usuários. Porém, os algoritmos computam a similaridade entre dois objetos comparando as preferências de todos os usuários que avaliaram esses objetos. Assim, $s_{o_k o_l}$ denota a similaridade entre os objetos o_k e o_l .

2.6 Alocação de Recursos

Alocação de recursos é um método baseado em grafos que consiste em uma dinâmica de distribuição de recursos em uma rede usuário-objeto. Esse método vem sendo explorado com sucesso em SRs (JAVARI; GHARIBSHAH; JALILI, 2014; ZHOU et al., 2007). A dinâmica da alocação de recurso é um processo no qual os vértices de usuários distribuem seus recursos iniciais para todos os seus vértices de objetos conectados. Então, cada vértice de objeto envia os seus recursos acumulados de volta aos vértices de usuários conectados. O resultado desse processo é uma métrica de similaridade baseada em grau na qual a similaridade entre dois usuários depende do grau¹⁹ dos objetos observados em comum por ambos os usuários. A alocação de recursos é baseada na ideia de que objetos populares deveriam ter menor impacto na similaridade dos usuários, uma vez que esses objetos provavelmente são apreciados pela maioria dos usuários (JALILI et al., 2018).

A Figura 2 ilustra como os recursos são distribuídos após cada passo do processo. Os recursos alocados finais podem ser representados da seguinte maneira:

$$\begin{pmatrix} u'_1 \\ u'_2 \\ u'_3 \end{pmatrix} = \begin{pmatrix} 1/2 & 1/4 & 1/3 \\ 1/6 & 1/2 & 1/6 \\ 1/3 & 1/4 & 1/2 \end{pmatrix} \cdot \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \quad (2.15)$$

A generalização do processo de alocação de recursos definido por Zhou et al. (2007) é definido a seguir. No primeiro passo da técnica, o k -ésimo objeto recebe os recursos de

¹⁹ O grau de um objeto é igual ao número de usuários que avaliou esse objeto. Ou seja, dado um objeto o_k , o grau desse objeto é igual a $\#\mathcal{U}_{o_k}$. Da mesma maneira, o grau do usuário u_i é igual a $\#\mathcal{O}_{u_i}$.

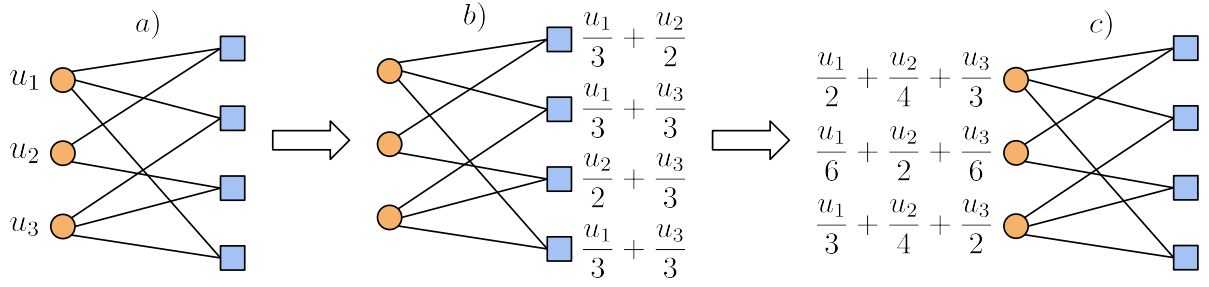


Figura 2: Os vértices redondos são os usuários (\mathcal{U}) e os vértices quadrados são os objetos (\mathcal{O}) da rede usuário-objeto. Os recursos unitários u_1 , u_2 e u_3 são inicialmente atribuídos aos vértices de usuários. Então, cada recurso unitário é igualmente distribuído fluindo dos usuários para os objetos ($a \rightarrow b$). No passo seguinte, os recursos são igualmente distribuídos de volta dos objetos para os usuários ($b \rightarrow c$).

cada usuário $u_i \in \mathcal{U}$, conforme definido na Equação (2.16).

$$f(o_k) = \sum_{i=1}^n \frac{a_{u_i o_k} \cdot f_i(u_i)}{\#\mathcal{O}_{u_i}} \quad (2.16)$$

onde $f_i(u_i)$ é o recurso unitário atribuído ao usuário u_i e $\#\mathcal{O}_{u_i}$ corresponde ao grau do usuário u_i , ou seja, o número total de objetos avaliados pelo usuário u_i . A notação $a_{u_i o_k}$ corresponde ao valor da matriz de adjacência para o usuário $u_i \in \mathcal{U}$ e para o objeto $o_k \in \mathcal{O}$, onde $a_{u_i o_k} = 1$ indica que o usuário u_i observou o objeto o_k , e $a_{u_i o_k} = 0$ indica que o usuário u_i ainda não avaliou o objeto o_k , conforme a Equação (2.17).

$$a_{u_i o_k} = \begin{cases} 1, & r_{u_i o_k} \in \mathcal{V} \\ 0, & r_{u_i o_k} = \emptyset \end{cases} \quad (2.17)$$

No próximo passo, o i -ésimo usuário recebe de cada objeto $o_k \in \mathcal{O}_{u_i}$ os recursos obtidos no passo anterior, conforme apresentado na Equação (2.18).

$$\begin{aligned} f'(u_i) &= \sum_{k=1}^n \frac{a_{u_i o_k} \cdot f(o_k)}{\#\mathcal{U}_{o_k}} \\ &= \sum_{k=1}^n \frac{a_{u_i o_k}}{\#\mathcal{U}_{o_k}} \sum_{j=1}^m \frac{a_{u_j o_k} \cdot f_i(u_j)}{\#\mathcal{O}_{u_j}} \end{aligned} \quad (2.18)$$

onde $\#\mathcal{U}_{o_k}$ corresponde ao grau do objeto o_k . A Equação (2.18) pode ser reescrita como:

$$f'(u_i) = \sum_{j=1}^m w_{u_i u_j} \cdot f_i(u_j) \quad (2.19)$$

onde

$$w_{u_i u_j} = \frac{1}{\#\mathcal{O}_{u_j}} \sum_{k=1}^n \frac{a_{u_i o_k} \cdot a_{u_j o_k}}{\#\mathcal{U}_{o_k}} \quad (2.20)$$

O resultado do processo inteiro de alocação de recursos pode ser representado na forma de matriz $\vec{f}' = W \cdot \vec{f}_i$, onde \vec{f}_i denota o vetor inicial de recursos, $W = \{w_{u_i u_j}\}^{m \times m}$ é a matriz de similaridade baseada em grau e \vec{f}' representa o vetor final com os recursos alocados.

Javari, Gharibshah e Jalili (2014) destacam que o processo de alocação de recursos possui influência significativa na performance de abordagens colaborativas baseadas em memória. Considerando a matriz de similaridade baseada em grau obtida a partir do processo de alocação de recursos, os autores introduziram o conceito de confiabilidade na similaridade entre dois usuários. Dessa maneira, os valores de similaridade obtidos inicialmente por meio de métricas de similaridade são atualizados como:

$$s_{u_i u_j} \leftarrow s_{u_i u_j} \cdot w_{u_i u_j} \quad (2.21)$$

2.7 Métricas de Avaliação para Recomendações

Para avaliar o desempenho de um SR de maneira quantitativa, o conjunto de todas as preferências \mathcal{R} normalmente é dividido em três partições não sobrepostas: treinamento (\mathcal{R}_{train}), validação (\mathcal{R}_{val}) e testes (\mathcal{R}_{test}). A partição de treinamento é utilizada para treinar o modelo. O modelo obtido é validado utilizando-se a partição de validação. Essa partição auxilia no ajuste dos hiper parâmetros do modelo e na prevenção de *overfitting*²⁰. Por fim, a partição de testes, que não participa da construção do modelo, é utilizada para prover uma avaliação independente do modelo final. Em abordagens colaborativas baseadas em memória, muitas vezes utiliza-se somente as partições de treinamento e testes. Nessas abordagens não ocorre o ajuste de parâmetros e os dados de treinamento são utilizados para a obtenção dos modelos de similaridade. O desempenho do sistema é avaliado diretamente sobre a partição de testes. Nesta seção, utiliza-se a partição \mathcal{R}_{test} como referência para a avaliação do modelo final dos SRs.

Na literatura, existe uma variedade de métricas aplicadas na avaliação de SRs (HERLOCKER et al., 2004). Porém, segundo Schafer et al. (2007), não existe uma métrica adequada para avaliar todos os critérios importantes relacionados ao desempenho de um SR. A escolha da métrica mais apropriada depende dos objetivos que o SR se propõe a cumprir. De maneira geral, esses objetivos são avaliados com métricas que são classificadas em quatro categorias: precisão, classificação, diversidade, e cobertura (HERLOCKER et al., 2004; ADOMAVICIUS; TUZHILIN, 2005; LÜ et al., 2012).

²⁰ Termo utilizado para descrever uma falha na capacidade de generalização de um modelo. Em outras palavras, um modelo aprende excessivamente os dados da partição de treinamento de maneira que perde a capacidade de previsão para dados que não estejam nessa partição.

2.7.1 Precisão

Duas métricas frequentemente utilizadas para avaliar a precisão de modelos preditivos em SRs são (ADOMAVICIUS; TUZHILIN, 2005; JALILI et al., 2018): *Erro Médio Absoluto* (MAE ²¹) e *Raiz do Erro Quadrático Médio* (RMSE ²²). A métrica MAE mede o desvio médio entre os valores preditos e os valores reais das preferências dos usuários. Ou seja, essa métrica calcula a soma da diferença absoluta entre as preferências preditas e as preferências reais, e divide pelo número total de preferências na partição de testes, conforme descrito na Equação (2.22):

$$\text{MAE} = \frac{\sum_{r_{u_i o_k} \in \mathcal{R}_{test}} |\hat{r}_{u_i o_k} - r_{u_i o_k}|}{\#\mathcal{R}_{test}} \quad (2.22)$$

A métrica RMSE é uma variação da métrica MAE. Porém, enquanto no MAE os valores dos erros são ponderados igualmente, o RMSE penaliza erros mais significativos ao elevá-los ao quadrado antes da soma. Ou seja, o RMSE dá mais relevância para valores de erro maiores e menos relevância para valores de erro menores. Com isso, o RMSE é mais apropriado em situações onde erros maiores são indesejados. A métrica RMSE é definida na (2.23):

$$\text{RMSE} = \sqrt{\frac{\sum_{r_{u_i o_k} \in \mathcal{R}_{test}} (\hat{r}_{u_i o_k} - r_{u_i o_k})^2}{\#\mathcal{R}_{test}}} \quad (2.23)$$

Valores menores para MAE e RMSE correspondem à uma maior precisão na predição. Ambas as métricas dão a mesma importância para todas as preferências na partição de testes. Ou seja, MAE e RMSE não consideram o ranking das preferências na lista de recomendações.

2.7.2 Classificação

De acordo com Herlocker et al. (2004), as métricas de classificação são apropriadas em situações em que o sistema tenta recomendar objetos considerando um critério binário de classificação. Então, partindo do princípio de que as preferências dos usuários são binárias, ou seja, “relevante” ou “não relevante”, as métricas de classificação medem a frequência com que um SR classifica corretamente ou incorretamente um objeto.

No mundo real, ao receber uma lista com recomendações, os usuários normalmente estão mais interessados nos n primeiros objetos recomendados nessa lista. Considerando

²¹ Do inglês *Mean Absolute Error*.

²² Do inglês *Root-mean-square Error*.

Tabela 2: Classificação dos objetos em função de sua relevância para as preferências observadas e para as preferências preditas.

	Selecionados	Não selecionados	Total
Relevantes	\mathcal{L}^{RS}	\mathcal{L}^{RN}	\mathcal{L}^R
Irrelevantes	\mathcal{L}^{IS}	\mathcal{L}^{IN}	\mathcal{L}^I
Total	\mathcal{L}^S	\mathcal{L}^N	

essa premissa, uma abordagem interessante para avaliar um sistema consiste em verificar se os objetos relevantes para um usuário estão entre os n primeiros lugares do ranking de todos os objetos que fazem parte da lista. *Precision* e *recall* são duas métricas bastante utilizadas por técnicas de recuperação de informação que têm sido frequentemente aplicadas na área de SRs. Para o cálculo dessas métricas, primeiramente é necessário que um conjunto de objetos \mathcal{L} seja categorizado levando em consideração as preferências observadas (reais) e as preferências preditas. Os objetos são classificadas em “relevantes” e “irrelevantes” de acordo com as preferências observadas. Os mesmos objetos também são classificados como “selecionados” (seriam recomendados) e “não selecionados” (não seriam recomendados) de acordo com as preferências preditas. A interseção entre essas classificações resulta na Tabela 2.

A métrica *precision* representa a probabilidade de que um objeto selecionado a partir da lista de recomendações seja relevante. Em outras palavras, *precision* é definida como a razão entre o número de objetos relevantes selecionados e o total de objetos selecionados, conforme a Equação (2.24).

$$precision = \frac{\#\mathcal{L}^{RS}}{\#\mathcal{L}^S} \quad (2.24)$$

A métrica *recall* é definida como a razão entre o número de objetos relevantes selecionados e o total de objetos relevantes. Ou seja, representa a probabilidade de que um objeto relevante seja selecionado. A métrica é calculada conforme a Equação (2.25).

$$recall = \frac{\#\mathcal{L}^{RS}}{\#\mathcal{L}^R} \quad (2.25)$$

Para avaliar o desempenho geral do sistema, utiliza-se então a média harmônica entre *precision* e *recall*, chamada de *F1*. A métrica *F1* é definida na Equação (2.26).

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \quad (2.26)$$

As métricas *precision*, *recall* e *F1* normalmente são calculadas em função do tamanho da lista de recomendação \mathcal{L} . O conjunto \mathcal{L} é formado pelos N objetos melhor

ranqueados na lista de recomendação. Então, a avaliação dos SRs por essas métricas de classificação é feita com base na curva obtida a partir da variação no tamanho de N .

2.7.3 Diversidade

Tratar problemas de recomendação considerando somente critérios de precisão e classificação pode não ser a maneira mais adequada para avaliar alguns SRs de maneira geral. Conforme destacado por Herlocker et al. (2004), recomendar um objeto que é popular e que praticamente todos os usuários têm interesse se traduz em alta precisão do sistema. Porém, recomendações desse tipo podem ser óbvias e também podem não ter nenhum valor para o usuário, uma vez que já se espera que esse objeto seja recomendado por conta da sua popularidade. Assim, objetos considerados populares, não adicionam informação nova aos usuários e não precisam de SRs para que sejam percebidos (HERLOCKER et al., 2004; SWEARINGEN; SINHA, 2001). Então, um critério que também deve ser considerado em SRs é a diversidade das recomendações. Métricas relacionadas à diversidade têm sido objeto de estudos em algumas publicações (ZHOU et al., 2010; CASTELLS; VARGAS; WANG, 2011).

Segundo Lü et al. (2012), a diversidade se refere ao quão diferente os objetos recomendados são entre eles e pode ser interpretada de duas maneiras: diversidade inter-usuários (diversidade entre as listas recomendadas) e diversidade intra-usuário (diversidade dos objetos dentro de uma lista com recomendações). Dados dois usuários u_i e u_j , a diversidade inter-usuários é dada pela distância de *Hamming* $H_{u_i u_j}$, conforme a equação a seguir:

$$H_{u_i u_j} = 1 - \frac{\#\mathcal{L}_{u_i} \cap \mathcal{L}_{u_j}}{N} \quad (2.27)$$

onde $\mathcal{L}_{u_i} \cap \mathcal{L}_{u_j}$ representa o conjunto dos objetos em comum entre as listas de recomendação dos usuários u_i e u_j , e N representa o tamanho da lista de recomendações. A diversidade geral do sistema H é obtida através da média entre todos os pares de usuários. Quanto maior for o valor obtido de H , maior é a diversidade (personalização) entre as listas recomendadas.

A diversidade intra-usuário tem como base a similaridade entre os objetos distintos contidos na lista de recomendações \mathcal{L}_{u_i} para um usuário u_i . Essa similaridade pode ser obtida através da utilização de uma métrica de similaridade, conforme apresentado na Seção 2.5. A diversidade intra-usuário I_{u_i} pode ser calculada através da seguinte equação:

$$I_{u_i} = \frac{1}{N \cdot (N - 1)} \cdot \sum_{o_k \in \mathcal{L}_{u_i}, o_l \in \mathcal{L}_{u_i} | o_k \neq o_l} s_{o_k o_l} \quad (2.28)$$

onde $s_{o_k o_l}$ é a similaridade entre os objetos o_k e o_l .

A avaliação geral do sistema I com relação à diversidade intra-usuários é obtida através da média de todos os usuários. Quanto menor o valor de I , mais diversificados são os objetos das listas recomendadas aos usuários.

A métrica de novidade é mais uma métrica de diversidade em SRs. De acordo com Lü et al. (2012), a maneira mais simples de medir o quanto um algoritmo é capaz de recomendar novos objetos é através do cálculo da popularidade média dos objetos recomendados, conforme a Equação (2.29).

$$Q = \frac{1}{N \cdot \#\mathcal{U}} \sum_{u_i \in \mathcal{U}} \sum_{o_k \in \mathcal{L}_{u_i}} \#\mathcal{U}_{o_k} \quad (2.29)$$

onde $\#\mathcal{U}$ representa a quantidade total de usuários no sistema, \mathcal{L}_{u_i} denota a lista dos objetos recomendados ao usuário u_i e $\#\mathcal{U}_{o_k}$ representa o grau do objeto o_k , ou seja, o número de usuários que selecionaram o objeto o_k . Quanto menor a popularidade média Q , maior é a novidade dos objetos recomendados.

2.7.4 Cobertura

Métricas de cobertura usualmente medem a porcentagem dos objetos presentes no sistema que são utilizados nas recomendações feitas aos usuários (HERLOCKER et al., 2004; LÜ et al., 2012). Uma maneira intuitiva de se determinar a cobertura de um sistema é pelo cálculo da razão entre o número total de objetos distintos recomendados e o número total de objetos no sistema. Uma baixa taxa de cobertura indica que o sistema tende a recomendar somente um pequeno número de objetos distintos, resultando em recomendações pouco diversificadas. Por outro lado, uma alta taxa de cobertura indica maior tendência em prover recomendações diversificadas.

A métrica de cobertura é calculada pela seguinte equação:

$$COV = \frac{\#\mathcal{L}^*}{\#\mathcal{O}} \quad (2.30)$$

onde $\#\mathcal{L}^*$ é o número total de objetos distintos nas top N posições das listas de recomendações de todos os usuários. \mathcal{L}^* é definido conforme Equação (2.31).

$$\mathcal{L}^* = \bigcup_{u_i \in \mathcal{U}} \mathcal{L}_{u_i} \quad (2.31)$$

2.8 Conclusões

Este capítulo apresentou uma visão geral dos principais conceitos relacionados aos SRs. Algumas das principais técnicas apresentadas foram classificadas em: abordagens baseadas em colaboração, abordagens baseadas em conteúdo e abordagens híbridas. Enquanto as abordagens baseadas em colaboração e as abordagens baseadas em conteúdo apresentam vantagens e desvantagens quando empregadas individualmente, as abordagens híbridas buscam tirar vantagem do melhor de cada uma das duas primeiras abordagens.

Muitas técnicas de recomendação utilizam a similaridade entre usuários ou objetos para realizar predições individuais da preferência de um usuário por um objeto não observado. A partir da similaridade é possível formar vizinhanças de usuários ou objetos semelhantes que são utilizadas nas predições. Dependendo da função de predição utilizada, a similaridade de cada vizinho pode atuar como fator de ponderação aumentando ou reduzindo sua importância no modelo de predição. Ou seja, a opinião de um vizinho mais similar tem um peso maior que a opinião de um vizinho menos similar. Uma vez obtidas as predições, é possível utilizar uma variedade de métricas para avaliar o desempenho dos SRs sob diferentes aspectos.

Nesse trabalho, o escopo da pesquisa relacionada ao tema de SRs se limita às abordagens colaborativas baseadas em memória, uma vez que elas levam em consideração o conceito de similaridade e agregação baseada em vizinhança. A avaliação dos resultados numéricos se dá por meio das métricas de precisão conhecidas como MAE e RMSE. Além disso, outros conceitos apresentados (como alocação de recursos e técnicas de normalização) também são utilizados como base para a abordagem de recomendação proposta nesse trabalho de pesquisa. O capítulo seguinte apresenta alguns exemplos da aplicação dessas técnicas para recomendações e personalização de conteúdo para o domínio da TV interativa.

Capítulo 3

Personalização de Conteúdo para Usuários de TV

PERSONALIZAÇÃO de conteúdo para TV é uma das áreas de aplicação mais desafiadoras e promissoras para SRs. A evolução tecnológica do ambiente de TV, com o advento da TV Digital, tem possibilitado o rápido aumento da quantidade de conteúdo disponibilizado aos usuários. Com isso, torna-se crítica a introdução de mecanismos que tornem a busca por conteúdo de interesse uma atividade rápida e confortável.

Este capítulo apresenta uma visão geral das principais pesquisas na área de personalização de conteúdo para usuários de TV. Além disso, o capítulo sintetiza os trabalhos relacionados da área e destaca as motivações para a personalização de conteúdo na TV, as fontes de dados utilizadas por algoritmos de personalização e a utilização de telas secundárias para a interação no ambiente de TV.

3.1 Motivação para a Personalização de Conteúdo para TV

A busca canal a canal, conhecida como *zapping*, é uma maneira bastante tradicional de se encontrar um programa ou conteúdo de interesse na TV convencional. Essa busca envolve duas ações:

- *Troca de canal*: o usuário navega sequencialmente por múltiplos canais buscando encontrar um canal que esteja exibindo um conteúdo potencialmente interessante.

- *Observação*: a cada troca de canal, o usuário para e observa brevemente o conteúdo exibido. Após uma rápida avaliação do conteúdo, ele toma uma decisão entre permanecer no canal ou seguir para o próximo canal continuando a busca.

Na TV aberta, o *zapping* pode ser considerado simples e até mesmo confortável do ponto de vista prático, uma vez que a quantidade de emissoras de TV é relativamente pequena. Dessa maneira, os usuários podem rapidamente encontrar o conteúdo de seu interesse navegando entre os canais por meio do controle remoto.

Com a melhoria dos sistemas de comunicação e da infraestrutura de TV, muitos provedores de TV fechada oferecem uma grande quantidade de canais. Por exemplo, Sky e Net, duas operadoras de TV por assinatura que atuam no Brasil, oferecem planos com pacotes que atualmente variam entre 100 e 250 canais. Diferentemente da TV aberta, fazer o *zapping* entre todas essas opções de canais em busca de conteúdo relevante para assistir não é uma tarefa simples e nem confortável. O usuário acaba sobrecarregado com o excesso de informação. Para auxiliar o usuário na busca por conteúdo, a indústria oferece acesso ao Guia Eletrônico de Programação (EPG¹) com informações textuais e horários de exibição dos programas para cada canal. Dessa maneira, os usuários podem navegar pelo EPG até encontrar os programas de seu interesse.

Embora exista o EPG para facilitar a busca de conteúdo, a grande quantidade de programas reduz a visibilidade de cada programa individual e inibe a busca por conteúdo. Assim, diante de uma experiência frustrante na busca de conteúdo, os usuários acabam consumindo somente uma pequena fração de todos os programas disponíveis (BAMBINI; CREMONESI; TURRIN, 2012; SMYTH; COTTER, 2004; ZIMMERMAN et al., 2004). Algumas empresas oferecem também um serviço para a gravação de conteúdo conhecido como *Personal Video Recorder* (PVR) ou *Digital Video Recorder* (DVR). Com esse serviço, os usuários navegam pelo EPG e podem agendar a gravação do conteúdo com horas ou até mesmo dias de antecedência. O problema dessa abordagem é que quanto maior for a janela temporal considerada, maior é a quantidade de conteúdo que o usuário precisa avaliar.

Muitos fabricantes de TV agregam serviços de *streaming* nas chamadas *Smart TVs*. Dessa maneira, utilizando uma conexão com a Internet, os usuários podem consumir conteúdo de serviços como *Netflix* e *YouTube*, por exemplo. Essa abordagem aumenta a quantidade de conteúdo que pode chegar até os usuários por meio da TV, o que agrava ainda mais o problema da sobrecarga de informação.

Além de trazer frustração para os usuários de TV, a sobrecarga de informação também traz problemas para as emissoras que disponibilizam o conteúdo. Um problema

¹ O Guia Eletrônico de Programação é comumente conhecido pelo acrônimo EPG, do inglês *Electronic Program Guide*.

significativo é como assegurar que os telespectadores notarão a programação de um canal específico dentro do universo de todos os canais. Smyth e Cotter (2004) destaca que isso é particularmente problemático para canais pequenos, uma vez que a baixa audiência pode resultar em impacto negativo nos lucros de publicidade do canal.

Diante dos problemas apresentados até aqui, a personalização de conteúdo de TV pode ser vista como uma necessidade real para muitos usuários e sistemas de TV. A personalização ameniza o problema da sobrecarga de conteúdo e pode contribuir com o aumento da satisfação dos usuários em relação às recomendações recebidas, ajudando-os a encontrar conteúdo relevante, e conseqüentemente contribuindo para o aprimoramento da publicidade direcionada (VELUSAMY et al., 2008; LI et al., 2020; VIEIRA; LINO, 2015). Através da personalização, as operadoras de TV buscam assegurar que um anúncio efetivamente alcance o público pretendido.

3.2 Dados de Entrada para Personalização

Em ambientes de TV, os dados de entrada podem ser classificados em duas categorias: dados de interação dos usuários e metadados de conteúdo. No caso dos dados de interação dos usuários, os dados de entrada são obtidos por meio da interação natural dos usuários com os serviços de TV utilizando o controle remoto ou outro dispositivo de interação. Já no caso dos metadados de conteúdo, as operadoras de serviços de TV fornecem os dados que servem para identificar e descrever o conteúdo. As seções a seguir descrevem as principais características dessas duas categorias de dados de entrada, assim como os trabalhos relacionados.

3.2.1 Dados de Interação dos Usuários

Os hábitos de visualização de conteúdo dos usuários em ambientes de TV configuram-se como uma fonte de dados particularmente interessantes para SRs baseados em FC. Porém, embora esse hábitos de visualização já venham sendo estudados há algum tempo, a captura dos dados de interação gerados em ambientes de TV ainda é uma área de pesquisa desafiadora (TOPALLI; KILINC, 2016; ALAM; KHUSRO; KHAN, 2019). Na era da TV analógica, a interação dos usuários com o aparelho de TV é bem limitada. O único modelo de interação possível para os usuários é o modelo de interação tradicional, que consiste somente na troca de canal e ajuste do volume utilizando um controle remoto convencional.

Mesmo a captura dos dados de audiência de TV era e ainda é realizada por meio de entrevistas ou por meio de um aparelho chamado *peoplemeter*. O *peoplemeter* é um dispositivo medidor de audiência que identifica qual canal de TV está sendo assistido em

uma residência e transmite os dados para uma companhia que faz a medição em tempo real (ALVAREZ et al., 2009). A medição de audiência é feita por amostragem e o *people-meter* é instalado nas residências de pessoas criteriosamente selecionadas para refletir os perfis socioeconômicos que compõem a sociedade daquele local. Dessa maneira, ao juntar os dados de outros aparelhos, a companhia consegue prover informações estatisticamente significativas sobre o número de pessoas assistindo um dado canal de TV em um dado instante.

Com os avanços nas tecnologias de TV, o modelo de interação tradicional começou a mudar. Entre outras coisas, a TVDI incluiu uma unidade de processamento e um canal de retorno que tornaram possível o desenvolvimento de uma miríade de serviços digitais para a TV. Isso abriu espaço para que o modelo de interação tradicional evoluísse e se tornasse mais elaborado, permitido aos usuários interagirem com aplicações e serviços de áudio/vídeo sob demanda, *e-commerce*, navegação na Internet e jogos por meio da TV (MORRIS; SMITH-CHAIGNEAU, 2005).

Essas novas possibilidades de interação tornaram a TV uma plataforma conveniente para capturar uma maior riqueza de dados dos usuários (TEIXEIRA et al., 2010). Os dados obtidos a partir das interações dos usuários (por exemplo, histórico de canais, preferências e hábitos de visualização) podem ser úteis para aplicações que incluem medição de audiência (AINA et al., 2014; ALVAREZ et al., 2009; LIM et al., 2014), publicidade direcionada (KIM; KANG, 2011; WANG; WANG; LU, 2013) e personalização de conteúdo por meio de SR (ABREU; ALMEIDA; TELES, 2014; BAMBINI; CREMONESI; TURRIN, 2012; SANCHEZ et al., 2013). Dessa maneira, os dados capturados de uma porção significativa da população podem se transformar em informações de grande valor para muitos profissionais no domínio da TV, que incluem produtores, provedores de serviço de TV e publicitários (TEIXEIRA et al., 2010).

A interação entre os usuários e a TV pode ser classificada em duas categorias: implícita e explícita. A interação implícita acontece como parte natural da interação do usuário com a TV. Por exemplo, informações como mudança de volume, programas assistidos, tempo de permanência em um determinado programa e horários em que a TV foi ligada e desligada são obtidas como resultado das ações dos usuários. Na interação explícita, os usuários executam ações que usualmente não acontecem na TV convencional. Ou seja, os usuários contribuem ativamente fornecendo informações sobre suas preferências. Exemplos desse tipo de interação incluem expressar suas opiniões sobre o conteúdo assistido (por exemplo, avaliar o conteúdo atribuindo uma avaliação em uma escala de 1 a 5) ou fornecer informações pessoais como idade e orientação sexual, ou mesmo os gêneros cinematográficos preferidos. Assim, considerando a interação de milhões de telespectadores, os dados capturados, seja de maneira implícita ou explícita, fornecem informações valiosas para os serviços que realizam personalização de conteúdo.

Alguns estudos descreveram estratégias para a captura de dados de interação dos usuários de TV (TEIXEIRA et al., 2010; BASILIO; MORENO; BARRÉRE, 2013). Teixeira et al. (2010) descreveram uma infraestrutura para a captura de informações implícitas e explícitas através de um controle remoto convencional. Os eventos do controle remoto são registrados localmente no STB em formato XML (*eXtensible Markup Language*). Nesse XML são definidos os elementos para descrever informações contextuais, e interações implícitas e explícitas. Basilio, Moreno e Barrére (2013) descreveram dois métodos para capturar os dados de interação dos usuários: 1) utilizando uma extensão de *middleware* para TV, que adiciona novas funcionalidades a um padrão existente; e 2) utilizando aplicações que são adicionadas ao fluxo de dados do provedor de TV, o que não requer mudanças no *middleware* de TV.

As interações sociais que ocorrem no domínio da TV também podem ser capturadas como dados dos usuários (LOCHRIE; COULTON, 2011; MITCHELL et al., 2011; CESAR; CHORIANOPOULOS; JENSEN, 2008). Exemplos desse tipo de interação incluem o compartilhamento de fotos, compartilhamento de vídeos e o envio de sugestões de programas para outros usuários. Essas informações podem ser utilizadas para enriquecer os dados de interação referente a cada usuário na personalização de conteúdo.

É importante ressaltar que o canal de retorno tem papel fundamental na captura dos dados dos usuários para a construção de uma base de dados robusta para a personalização. Várias soluções comerciais de STBs, como o *AppleTV*, *AndroidTV* e o PVR da empresa *TiVo* oferecem conectividade com a Internet para o envio de dados através do canal de retorno. Na TV Digital brasileira, a própria especificação propõe uma variedade de técnicas (por exemplo, redes de dados de telefonia celular e redes WiFi) que podem ser utilizadas para o canal de interatividade (ABNT, 2018b). Em alguns trabalhos, os autores consideram a ausência do canal de retorno e propõem soluções em que os dados dos usuários são coletados e processados localmente no próprio STB (BJELICA, 2010; LÓPEZ-NORES et al., 2008). O problema com esse tipo de abordagem é que a base de dados coletada possivelmente terá um tamanho reduzido, o que pode dificultar a identificação de padrões de interação que são fundamentais para a qualidade da personalização.

3.2.2 Metadados do Conteúdo

Uma das características da TV é que os programas são exibidos em horários e canais predeterminados, repedindo-se diariamente ou semanalmente. Normalmente, os programas são listados em um guia de programação fornecido pela operadora de serviços de TV informando o horário e o canal de cada programa. Esse guia de programação pode ser acessado por meio de uma página na Internet ou por meio do EPG. As informações contidas no guia são compostas de um conjunto de dados, que são conhecidos como

metadados. Esses metadados são utilizados para descrever os programas (HJELM, 2008). Em outras palavras, os metadados identificam e descrevem o conteúdo de TV utilizando dados como canal, título, horário de início, duração do programa e sinopse.

A finalidade dos metadados não é somente prover informações e identificar o conteúdo digital. Eles também podem ser utilizados para alimentar serviços que automatizam o gerenciamento do conteúdo nos provedores de TV ou para a realização de buscas por conteúdo, como acontece com o mecanismo de busca por vídeos no YouTube. Além disso, as operadoras de serviços de TV podem utilizar os metadados como entrada para técnicas de recomendação baseadas em conteúdo com o objetivo de modelar as preferências dos usuários.

Existem vários mecanismos de padronização que podem ser utilizados para metadados de conteúdo de TV. O *Dublin Core* é um dos padrões mais antigos de vocabulário de metadados para o gerenciamento de conteúdo digital genérico (DCMI, 2021). O padrão foi inicialmente formado por um consórcio de empresas, chamado *Dublin Core Metadata Initiative*, sediado na cidade de Dublin, Ohio nos Estados Unidos. O padrão *Dublin Core* consiste em dois níveis: simples e qualificado. O nível simples inclui 15 elementos de metadados (*Title, Subject, Description, Creator, Publisher, Contributor, Date, Type, Format, Identifier, Source, Language, Relation, Coverage e Rights*) e o nível qualificado adiciona 3 elementos ao nível anterior (*Audience, Provenance e RightsHolder*).

Especificamente para o domínio da TV, *TV-Anytime* é um conjunto de especificações proposta pelo *TV-Anytime Forum*, uma associação de organizações que busca desenvolver especificações para serviços multimídia baseados em armazenamento digital em massa para plataformas de consumo (TV-ANYTIME, 2021). A especificação do padrão *TV-Anytime* tem a finalidade de prover suporte para a busca, seleção e aquisição de conteúdo dos provedores de serviço através dos metadados. Uma das vantagens do *TV-Anytime* é que os metadados podem ser obtidos sob demanda através do canal de interatividade ou via IPTV. Os metadados do *TV-Anytime* são representados por meio de estruturas XML, o que o torna um padrão extensível. Os elementos XML são agrupados em três categorias: metadados de descrição do conteúdo, metadados de descrição de instância e metadados do consumidor. Os metadados definidos através do *TV-Anytime* têm sido objeto de estudo em muitos trabalhos envolvendo personalização de conteúdo (BLANCO-FERNÁNDEZ et al., 2011; LEE et al., 2005; BARRAGÁNS-MARTÍNEZ et al., 2010). Ainda no formato XML existe o XMLTV² que fornece um formato para representação de programação de TV através de arquivos XML. No Brasil, o XMLTV é mantido pela Revista Eletrônica³, que fornece a programação semanal da TV fechada. Esses dados podem ser utilizados para visualização em diversos dispositivos e programas leitores do formato

² <http://wiki.xmltv.org/index.php/XMLTVProject>

³ <http://revistaeletronica.com.br>

XMLTV.

No Sistema Brasileiro de TV Digital Terrestre (SBTVD-T), a especificação define um conjunto de metadados para representar os dados do conteúdo (ABNT, 2020; ABNT, 2007). A tabela *Event Information Table* (EIT) de serviços de informação contém as informações sobre o conteúdo de um determinado evento. Assim, no sistema brasileiro, espera-se que os metadados utilizados para personalização de conteúdo sejam obtidos a partir dos dados contidos na tabela EIT.

3.3 Recomendação para TV

Muitas das técnicas de recomendação empregadas no domínio da TVDI são baseadas nos métodos de filtragem colaborativa e filtragem de conteúdo descritos na Seção 2.4. Assim, a TVDI compartilha muitos dos desafios de SRs presentes em outros domínios e ainda por cima introduz novos. Entre os novos desafios, pode-se destacar requisitos de tempo real e escalabilidade, dificuldades em coletar as interações dos usuários e dificuldades em coletar metadados do conteúdo da TV (BAMBINI; CREMONESI; TURRIN, 2012; LI et al., 2020).

Conforme enfatizado na Seção 3.2, o histórico de interação dos usuários e os metadados da programação são fontes ricas de obtenção de dados de entrada para realizar a personalização de conteúdo para a TV. Esses dados se tornam ainda mais relevantes diante da ubiquidade da TV. No Brasil, por exemplo, estima-se que 97,1% dos domicílios possuem aparelho de TV (IBGE, 2016). Então, com o suporte tecnológico adequado e uma audiência diária de milhões de usuários, é possível obter uma quantidade significativa de dados de entrada para gerar recomendações.

Produzir recomendações baseadas na análise de grandes quantidades de dados de entrada, em um tempo relativamente curto, pode significar um grande desafio em sistemas de TV interativa. De maneira geral, usuários de TV estão acostumados com tempos de resposta bem curtos. Então, gerar recomendações, principalmente em sistemas baseados em heurísticas, pode requerer uma grande quantidade de recursos computacionais e tempo. Buscando reduzir o tempo de processamento, alguns trabalhos propõem o uso de computação nas nuvens para processar os dados de milhares de usuários (LEE; LEE; LEE, 2010; LAI et al., 2011), uma vez que uma das características de computação nas nuvens é a disponibilidade de recursos computacionais conforme a demanda por processamento (ARMBRUST et al., 2010).

Lai et al. (2011) propõem um método baseado em computação nas nuvens para gerar recomendações de programas de TV. O método utiliza implementações dos algoritmos

K-Means e KNN adaptadas para o modelo de programação MapReduce⁴ (DEAN; GHEMAWAT, 2008). Nesse método, dados como programa selecionado, data do programa, canal, título, classificação e perfil dos usuários são armazenados em servidores da nuvem de computadores. A cada programa é associado um peso, que consiste na soma dos tempos de duração da audiência, indicando a popularidade do programa. O algoritmo *K-Means* é utilizado para agrupar usuários similares em *clusters* e o método KNN é utilizado para adicionar novos usuários dentro dos *clusters*. Dessa maneira, programas populares entre os usuários de um *cluster* ou entre os usuários de *clusters* similares são recomendados aos demais usuários.

Outro trabalho para recomendação de programas de TV baseado em computação nas nuvens foi desenvolvido por Lee, Lee e Lee (2010). Os autores propõem uma infraestrutura que analisa os padrões de navegação dos usuários para recomendar programas de acordo com suas preferências. Os métodos utilizados consistem basicamente em analisar o histórico de interação dos usuários e criar um perfil de preferências utilizando uma representação de vetores. Por meio da similaridade do cosseno, os usuários mais similares são encontrados e os programas com maior grau de preferência são recomendados. O grau de preferência por um programa é determinado por meio de um conjunto de heurísticas baseadas na vizinhança encontrada.

As abordagens propostas por Lai et al. (2011) e Lee, Lee e Lee (2010) consideram a presença do canal de interatividade. Uma das principais vantagens desses trabalhos é a possibilidade de utilizar dados de milhões de usuários. Esse cenário favorece recomendações mais diversificadas obtidas a partir de outros usuários. Além disso, o uso de computação nas nuvens para processar os dados de entrada contribui para melhorar a escalabilidade do sistema, reduzindo o tempo de resposta das recomendações.

A abordagem proposta por Bjelica (2010) considera a ausência de um canal de interatividade e utiliza somente os dados coletados localmente para as recomendações. Esses dados são a representação vetorial dos metadados referentes ao conteúdo dos programas previamente assistidos. Ou seja, um programa o_k é representado por um vetor n -dimensional $\vec{w}_{o_i} = (w_{1o_i}, w_{2o_i}, \dots, w_{no_i})$ onde cada elemento do vetor representa um atributo do programa de TV. Um modelo do usuário u_i é determinado como sendo o centroide do *cluster* de vetores que correspondem aos programas previamente assistidos pelo usuário u_i . Ou seja, o modelo do usuário u_i é representado por um vetor n -dimensional $\vec{w}_{u_i} = (w_{1u_i}, w_{2u_i}, \dots, w_{nu_i})$ obtido a partir de todos os programas assistidos anteriormente. Os programas que devem ser recomendados no futuro são definidos por meio da similaridade entre os programas e o modelo que representa usuário. Os programas mais próximos (que apresentam maior similaridade) são então recomendados. Os resultados

⁴ MapReduce é um modelo de programação utilizado no processamento distribuído de grandes quantidades de dados em *clusters* de computadores.

experimentais da abordagem proposta por Bjelica (2010) revelam que o sistema possui melhor performance quando são utilizados entre 20 e 30 programas para determinar o modelo para o usuário. Uma desvantagem da abordagem é a baixa capacidade de prover recomendações diversificadas, uma vez que a abordagem utiliza somente os dados do próprio usuário para prover as recomendações.

Yu et al. (2010) propõem uma abordagem de recomendação de conteúdo para TV que combina as similaridades de duas características: descrição e gênero. Na similaridade de descrição, os programas são representados por vetores e a similaridade é calculada utilizando o cosseno. A similaridade de gênero é calculada com base na hierarquia de gêneros da especificação do *TV-Anytime*. A similaridade final entre os programas é a combinação das duas similaridades. Os resultados experimentais revelam que a combinação das similaridades das duas características produziram resultados melhores que utilizando a similaridade de uma característica isolada.

Barragáns-Martínez et al. (2010) propõem um sistema online para recomendar conteúdo de TV como uma abordagem híbrida de recomendação. Na parte baseada em conteúdo, o sistema emprega a similaridade do cosseno para determinar a semelhança entre os programas e o modelo de usuário, gerando uma lista de recomendações. Na parte colaborativa, os autores utilizaram uma técnica baseada em *Singular Value Decomposition* (SVD). As recomendações provenientes das duas partes são sugeridas para o usuário final, sendo que os programas recomendados pelas duas partes aparecem visualmente destacados na página de exibição.

Em um ambiente doméstico de TV, normalmente existem vários usuários que fazem parte do mesmo grupo familiar ou social. Com base nisso, alguns trabalhos buscam realizar recomendações não somente para um usuário individualmente, mas sim para um grupo de usuários (YU et al., 2006; MASTHOFF, 2004).

Masthoff (2004) apresenta uma série de estratégias baseadas na Teoria da Escolha Social para agregar modelos individuais e prover recomendações para grupos de usuários. Basicamente, as técnicas recebem como entrada uma matriz com as avaliações dos usuários para um conjunto de objetos (programas ou filmes) e geram uma única nota para cada objeto. Essas notas devem expressar as preferências de todos os usuários.

Com o mesmo objetivo de prover recomendações para um grupo de usuários, Yu et al. (2006) descrevem uma técnica que combina os perfis individuais dos usuários para se obter um único perfil que melhor representa as preferências do grupo. Cada usuário é representado por um vetor de características (que são as mesmas utilizadas para representar os programas) e o perfil que descreve o grupo de usuários é obtido por meio de heurísticas aplicadas a cada característica. Os programas recomendados ao grupo são os que possuem maior similaridade com o vetor que representa o grupo. Para a obtenção dos dados de interação dos usuários, o trabalho requer que todos os usuários se identifiquem

por meio de uma interface gráfica. A partir de então, os eventos capturados na interação são associados aos usuários que estão identificados no momento.

Li et al. (2020) apontam um problema interessante relacionado aos SRs no domínio da TV. Embora o consumo de conteúdo audiovisual na TV possa acontecer de maneira linear ou sob demanda, muitos trabalhos focam somente em soluções que resolvem o problema de recomendação para consumo de conteúdo sob demanda. Ou seja, essas abordagens consideram que qualquer usuário pode consumir o conteúdo quando quiser e assim os SRs podem também prover recomendações conforme a demanda. Porém, uma das maneiras mais comuns de consumo de conteúdo de TV é o consumo linear de programação. Ou seja, os programas são transmitidos de maneira sequencial, seguindo um fluxo temporal em que todos os usuários consomem o mesmo conteúdo ao mesmo tempo. O problema acontece porque nem sempre os metadados referentes aos programas estão disponíveis. Nesse sentido, Li et al. (2020) propõem um sistema baseado em redes neurais profundas que não depende dos metadados e nem da análise do conteúdo da programação. Esse sistema se baseia na análise das interações implícitas durante a troca de canais para aprender padrões de comportamento dos usuários em relação aos programas. Como resultado, o sistema sugere uma lista de programas em tempo real sempre que o usuário inicia a troca de canais buscando um conteúdo de seu interesse.

Topalli e Kilinc (2016) descrevem uma abordagem de recomendação para padrões que suportam distribuição de conteúdo híbrida. Ou seja, para distribuição linear de conteúdo e sob demanda. Nessa abordagem, os autores construíram uma aplicação que registra os dados do programa assistido caso as interações implícitas dos usuários atendam a um conjunto de regras. Esses dados são posteriormente utilizados para treinar uma rede neural para prover recomendações para programação linear e sob demanda.

A Tabela 3 apresenta um resumo comparativo dos métodos e das técnicas empregadas nos trabalhos relacionados. Embora os trabalhos desenvolvidos por Lai et al. (2011), Lee, Lee e Lee (2010), Bjelica (2010) e Yu et al. (2010) apresentem abordagens para a personalização de conteúdo voltada para usuários individuais, esses trabalhos não apresentam evidências de como os dados foram ou podem ser adquiridos dentro de um ambiente de TV. Somente os trabalhos desenvolvidos por Barragáns-Martínez et al. (2010) e Yu et al. (2006) apontam como os dados de interação dos usuários são capturados. Porém, esses autores propõem a utilização de interações e interfaces incompatíveis com o ambiente de TV. Barragáns-Martínez et al. (2010) propõem a utilização de uma interface web e um computador tanto para a obtenção dos dados quanto para a recomendação de conteúdo. Já, Yu et al. (2006) propõem uma interface gráfica no computador para que todos os usuários presentes se identifiquem de maneira que os eventos capturados sejam associados a esses usuários.

Uma das abordagens que mais se aproxima de um ambiente de TV é a abordagem

Tabela 3: Resumo dos métodos e técnicas de recomendação utilizadas nos trabalhos relacionados. Os campos abreviados na tabela possuem os seguintes significados: FC (filtragem colaborativa), CONT (conteúdo), MEM (memória), MOD (modelo), IND (individual), GRUP (grupo), SIM (similaridade) e CL (clusterização).

Trabalhos	Abordagens de Recomendação		Classificação do Método		Tipo de Personalização		Técnicas Utilizadas			
	FC	CONT	MEM	MOD	IND	GRUP	SIM	CL	Outras	
Masthoff (2004)	X			X		X				
Yu et al. (2006)	X			X		X	X			
Barragáns-Martínez et al. (2010)	X	X	X		X		X		SVD	
Bjelica (2010)		X		X	X		X			
Lee, Lee e Lee (2010)	X		X		X		X		KNN	
Yu et al. (2010)		X	X		X		X	X	HCA	
Lai et al. (2011)	X		X		X		X	X	KNN, K-Means	
Topalli e Kilinc (2016)		X		X	X				Redes neurais	
Li et al. (2020)		X		X	X		X		Redes neurais profundas	

proposta por Topalli e Kilinc (2016), onde os autores construíram uma aplicação de TV para capturar dados de interação implícita dos usuários. Nesse trabalho, há pouca modificação do ambiente habitual da TV. Porém, da mesma maneira que alguns trabalhos descritos anteriormente, esse trabalho não dá detalhes de como consegue obter informações individuais dos usuários.

O trabalho proposto por Li et al. (2020) também descreve um ambiente bem próximo de um ambiente real de TV. Os dados de interação do usuário são capturados durante uma atividade bem típica de ambientes de TV, que é a troca de canais. Além disso, a troca de canais também é o gatilho para a exibição da lista de recomendações, uma vez que os eventos gerados por essa atividade sinalizam a busca por conteúdo de interesse do usuário.

3.4 Vizinhaça na Personalização de Conteúdo

A obtenção de uma vizinhaça de qualidade é uma atividade crítica para melhorar a qualidade das recomendações em SRs baseados em FC (LIU et al., 2014). Essa vizinhaça se refere a usuários que são de alguma maneira similares. A vizinhaça é importante não somente para a personalização de conteúdo para TV, mas para SRs em qualquer área de aplicação. Não por acaso, muitos esforços têm sido empregados na criação de técnicas de similaridade e clusterização que possibilitem encontrar vizinhos cada vez mais relevantes.

Muitas abordagens de recomendação utilizam informações estruturais de redes usuário-objeto para melhorar os modelos de similaridade (AN et al., 2016; ZHOU et al., 2007; JAVARI; GHARIBSHAH; JALILI, 2014; ZHOU et al., 2010). Algumas dessas abordagens se baseiam em um método conhecido como *alocação de recursos* (descrito na Seção 2.6). Zhou et al. (2007) propõem um método de ponderação baseado em alocação de recursos conhecido como *propagação probabilística* (ProbS). Esse método usa uma projeção de modo único de uma rede usuário-objeto para obter pesos assimétricos como um modelo de similaridade. Zhou et al. (2010) contribuem com o trabalho anterior propondo outro método baseado em alocação de recursos conhecido como *propagação de calor* (HeatS). Uma vez que os resultados experimentais sugeriram que o ProbS favorece a precisão enquanto que o HeatS favorece a diversidade, os autores sugeriram uma abordagem híbrida. Essa abordagem combina ProbS e HeatS para favorecer precisão e diversidade de acordo com um parâmetro de hibridização.

A abordagem proposta por Javari, Gharibshah e Jalili (2014) utiliza a alocação de recursos como modelo de confiabilidade para melhorar a precisão das recomendações em sistemas baseados em FC. Essa abordagem consiste na modificação da função de agregação

para incorporar confiabilidade como meio de reforçar ou enfraquecer a similaridade entre dois usuários. Os resultados numéricos mostram que empregar alocação de recursos como modelo de confiabilidade em abordagens clássicas de FC melhora significativamente a precisão de SRs. Ayub et al. (2019) propõem um trabalho similar para obter uma métrica de similaridade melhorada ao incorporar a variação da escala de preferências dos usuários ao PCC.

Todos os trabalhos descritos anteriormente demonstram efetividade para alcançar melhores modelos de recomendação. Entretanto, eles descrevem métodos que consideram todos os objetos para se computar uma similaridade global entre dois usuários distintos do sistema. Uma vez que os usuários podem expressar diferentes opiniões sobre diferentes grupos de objetos relacionados, explorar a similaridade entre os usuários dentro desses grupos pode ser um tópico de estudo interessante. Amatriain et al. (2011) apresentam uma visão geral de algumas técnicas de clusterização que são amplamente empregadas em muitos esforços para melhorar a qualidade de sistemas de recomendação (CHANG; HARPER; TERVEEN, 2015; KANT; MAHARA, 2018; ZAHRA et al., 2015; JEONG; LEE; CHO, 2010; KOOHI; KIANI, 2016; TRAN et al., 2019; XUE et al., 2005).

Chang, Harper e Terveen (2015) propõem uma técnica que tem como objetivo melhorar a qualidade inicial da personalização para novos usuários. A ideia é que esses usuários podem inicialmente expressar as suas preferências por grupos de objetos em vez de objetos individuais. Os autores usam clusterização espectral para agrupar objetos em *clusters*, cuja qualidade é avaliada por meio de inspeção manual. Embora os resultados obtidos na precisão das recomendações sejam ligeiramente piores quando comparados com outros métodos, o trabalho mostra a importância das técnicas baseadas em *cluster*. Os resultados experimentais revelam que os *clusters* de objetos reduzem o esforço dos usuários ao pensar em grupos de objetos relacionados.

Os estudos conduzidos por Jeong, Lee e Cho (2010) propõem um método de similaridade que é baseado no algoritmo de clusterização conhecido como *affinity propagation*. Após encontrar os *clusters*, a similaridade é determinada pela soma entre *responsabilities* e *availabilities*. Xue et al. (2005) empregam o algoritmo *K-Means* como um mecanismo de pré-processamento para ajudar na seleção da vizinhança. Em vez de procurar pelos vizinhos em toda a base de dados, a abordagem limita a busca aos *clusters* que são mais próximos do usuário ativo. Os resultados experimentais revelaram que esse método melhora significativamente a precisão da predição. Kim e Kang (2011) usam dados demográficos e *K-Means* para agrupar usuários similares. Assim, esses usuários podem prover conteúdo de TV personalizado para um usuário ativo.

3.5 Telas Secundárias na Interatividade

A proliferação de dispositivos móveis como *tablets* e *smartphones* tem mudado a maneira como as pessoas se comportam diante da TV. Esses dispositivos vêm conquistando cada vez mais espaço e têm sido frequentemente incorporados como telas secundárias na experiência dos usuários ao assistir TV (COURTOIS; D'HEER, 2012). De acordo com um relatório da *The Nielsen Company*, uma empresa de medição de audiência, 88% dos usuários de TV americanos utilizaram um dispositivo secundário enquanto assistiam TV durante o ano de 2018 (NIELSEN, 2019). Ainda de acordo com o relatório, 45% dos telespectadores utilizam dispositivos secundários com muita frequência ou sempre. Com relação ao motivo da utilização dos dispositivos secundários, 71% informaram que estavam buscando conteúdo adicional relacionado ao que estava sendo exibido na TV.

Vários trabalhos recentes apontam a importância do uso de dispositivos móveis como telas secundárias para enriquecer a interação dos usuários com a TV (CRUICKSHANK et al., 2007; CESAR; BULTERMAN; JANSEN, 2009; TEIXEIRA et al., 2010; CESAR; KNOCHE; BULTERMAN, 2010; TSEKLEVES et al., 2011; MEIXNER et al., 2017; VANATTENHOVEN; GEERTS, 2018; GUO, 2020). Porém, a ideia de utilização de telas secundárias para interação com a TV não é recente. Em 1996, ROBERTSON et al. desenvolveram um protótipo em que um dispositivo móvel de computação pessoal foi utilizado como dispositivo auxiliar para interação com a TV. Os usuários poderiam utilizar o PDA tanto para controlar a TV quanto para exibir conteúdo adicional (por exemplo, descrição textual) sobre imagens e vídeos exibidos. Na arquitetura do sistema, as ações realizadas no PDA eram enviadas para um STB, que por sua vez solicitava vídeos ou imagens de um servidor de conteúdo. O conteúdo solicitado poderia ser exibido diretamente na tela da TV ou poderia ser enviado para o dispositivo móvel.

Com os avanços das tecnologias de TV e tecnologias para dispositivos móveis, vários outros estudos foram desenvolvidos. Em 2007, Cruickshank et al. (2007) selecionaram 62 pessoas (de 20 residências) pertencentes a diferentes grupos socioeconômicos e avaliaram a utilidade de um dispositivo móvel como tela secundária na interação com a TV. Os usuários avaliaram positivamente a utilização de telas secundárias na interação, principalmente na exibição de elementos visuais que antes eram exibidos somente na tela da TV. Tseklevs et al. (2011) dão continuidade ao trabalho anterior conduzindo um novo estudo com 27 famílias sobre a utilização de diferentes mídias no ambiente de TV. Um protótipo de tela secundária foi desenvolvido e utilizado durante os experimentos. A pesquisa revelou que os usuários avaliaram positivamente a utilização desse dispositivo em atividades sociais como o compartilhamento de fotos e vídeos entre membros da família na tela do próprio dispositivo ou na tela da TV. Esses resultados demonstram que a introdução de um dispositivo pessoal pode de fato enriquecer as atividades sociais inerentes do ambi-

ente de TV, conforme destacado em outros trabalhos (CESAR; CHORIANOPOULOS; JENSEN, 2008; CESAR; GEERTS, 2011).

Cesar, Bulterman e Jansen (2009) contribuem significativamente ao identificar e modelar os requisitos para a implementação de uma arquitetura para TV interativa com interação baseada em telas secundárias. Os requisitos foram divididos em três categorias: controle, enriquecimento e compartilhamento de conteúdo. Resultados experimentais realizados na arquitetura destacam que os usuários avaliaram positivamente a integração de telas secundárias no ambiente de TV. Cesar, Knoche e Bulterman (2010) observam a utilização de telas secundárias como uma potencial mudança de hábito no modo como as pessoas consomem conteúdo audiovisual.

Teixeira et al. (2010) descrevem uma infraestrutura para capturar eventos de interação entre o usuário e a TV para uso em serviços de personalização e recomendação. A infraestrutura proposta é baseada na interação com o controle remoto, porém os autores apontam que é importante suportar diferentes dispositivos (como *smartphones* e *tablets*), uma vez que eles podem promover uma interação mais elaborada para a captura de dados.

Comercialmente, algumas empresas já exploram o desenvolvimento de aplicações de tela secundária para controlar ou adicionar mais funcionalidades à TV. Exemplos dessas empresas incluem a *TiVo*, com o seu PVR, e o *Google*, com o *AndroidTV*. Além disso, essas empresas oferecem *kits* de desenvolvimento que permitem a criação de aplicações móveis customizadas que ajudam a enriquecer a experiência dos usuários com os seus dispositivos.

Vanattenhoven e Geerts (2018) e Meixner et al. (2017) exploram o impacto de aplicações para tela secundária tanto na experiência do telespectador quanto na produção de conteúdo. Na experiência do telespectador, Vanattenhoven e Geerts (2018) falam sobre dispositivos de tela secundária como uma maneira de trazer a atenção do usuário de volta para tela principal. Realizar essa tarefa sem impactar negativamente a experiência do telespectador envolve vários desafios como melhorar a usabilidade nas aplicações de tela secundária e a sincronização de conteúdo entre a tela principal e a tela secundária. Meixner et al. (2017) conduzem entrevistas com profissionais do setor de produção de conteúdo interativo para TV com o objetivo de melhorar a experiência dos telespectadores na exibição de conteúdo para múltiplas telas. Como resultado, os autores propõem novos fluxos de trabalho, requisitos de ferramentas de produção de conteúdo e a criação de novos cargos especializados para a produção de conteúdo interativo para múltiplas telas.

Embora a atividade de assistir TV seja tradicionalmente coletiva, a introdução de dispositivos secundários de computação pessoal traz a personalização como um requisito na oferta dos serviços. Dessa maneira, as recomendações, que antes poderiam ser direcionadas para um grupo de usuários, possuem agora também um requisito de individualização para direcionar as recomendações para usuários específicos conforme a demanda.

No modelo de interação tradicional, essa individualização é dificultada pelo uso de um único controle remoto compartilhado entre todos os usuários. Assim, é difícil identificar o autor da interação e produzir conteúdo personalizado para ele. Como uma potencial solução, o uso de dispositivos de tela secundária surgem como um mecanismo interessante de aplicação na interação devido as suas características de individualização inerentes da computação pessoal.

A partir da análise dos trabalhos relacionados, pode-se concluir que o emprego de telas secundárias na interatividade com a TV possui várias vantagens. Algumas dessas vantagens são resumidas a seguir:

- Os elementos visuais de navegação não aparecem na tela da TV (tela principal), mas sim na tela secundária. Dessa maneira, a experiência visual compartilhada por todos os membros do ambiente TV não é afetada quando um usuário particular visualiza o EPG ou um serviço interativo na tela secundária.
- O dispositivo convencional de controle remoto pode ser substituído por um dispositivo com capacidades de interação mais sofisticadas. Isso abre possibilidades para a adição de modelos de interação mais elaborados para enriquecer e facilitar a interação dos usuários. Esses modelos de interação podem incluir controles com tela de toque, comandos por voz e gestos.
- A utilização de dispositivos pessoais como telas secundárias facilita a captura de dados de interação individuais que podem ser empregados em serviços de personalização e recomendação.

3.6 Conclusões

Como pode ser observado nos trabalhos apresentados nesse capítulo, as técnicas de recomendação descritas de maneira geral no capítulo 2 são as mesmas utilizadas no domínio da TV. A diferença é que o domínio da TV possui requisitos específicos para a implementação de aplicações e para a personalização de conteúdo. Os usuários de TV possuem necessidades, mecanismos de interação e dinâmica comportamental diferentes quando comparados a usuários que utilizam o computador para acessar serviços de *e-commerce*, por exemplo. Um exemplo disso é a natureza coletiva de um ambiente de TV e a interação que muitas vezes só acontece por meio de um controle remoto.

Embora exista a possibilidade de a atividade de assistir TV ser individual, os telespectadores tipicamente assistem TV em grupo e tomam decisões em grupo. Dessa maneira, os dados capturados sobre os usuários de TV não são individuais, mas sim do grupo que está assistindo. Como resultado, um usuário que possui maior controle sobre

o ambiente pode acabar impondo as suas preferências particulares sobre as preferências dos outros usuários. Nesse cenário, a criação de mecanismos para individualizar a captura dos dados de interação acaba se fazendo um requisito necessário para promover a personalização de conteúdo a nível também individual.

Como também pode ser observado nos trabalhos apresentados, uma tendência que tem se revelado no ambiente da TV é que os usuários realizem diversas atividades como navegar na web e procurar informações adicionais sobre a programação enquanto assistem. A introdução de um dispositivo de tela secundária tem sido avaliada positivamente em vários estudos realizados e esse dispositivo tem funcionado como um agregador de conteúdo e não como um substituto ou intruso no ambiente da TV. Isso se confirma com pesquisas que revelam a naturalidade e ampla utilização de dispositivos de tela secundário durante o consumo de conteúdo de TV.

A análise das informações contidas nesse capítulo servem de base para a elaboração dessa tese de doutorado. O capítulo a seguir descreve a proposta de um mecanismo para a captura de informações individuais de usuários de TV por meio da utilização de dispositivos de tela secundária. O próximo capítulo também descreve uma proposta de personalização de conteúdo baseada em FC, que tem como objetivo alcançar maior precisão das recomendação a partir uma vizinhança de qualidade obtida por meio da observação de *clusters* de objetos.

Capítulo 4

Materiais e Métodos

ESTE capítulo descreve os materiais e métodos utilizados para resolver os problemas 1 e 2 apresentados no Capítulo 1. O capítulo apresenta um método para a captura de dados de maneira organizada e individualizada (*Problema 1*) e um método para a recomendação de conteúdo utilizando uma vizinhança relevante baseada em similaridade local obtida a partir de *clusters* de objetos (*Problema 2*). Cada problema é descrito de maneira detalhada e em seguida é apresentado o método proposto para resolver esse problema. Ao final do capítulo, também são apresentadas as bases de dados e as métricas utilizadas nos resultados experimentais, assim como as conclusões relacionadas às soluções apresentadas.

4.1 Captura de Dados de Maneira Organizada e Individualizada

As interações dos usuários com o conteúdo de TV se configuram como uma valiosa fonte de dados para muitos profissionais no domínio da TVDI, o que inclui produtores de conteúdo, provedores de TV e publicitários (TEIXEIRA et al., 2010). Esses dados tipicamente são o histórico de visualização de canais, preferências implícitas, preferências explícitas e hábitos de visualização. No campo das aplicações, esses dados são úteis para áreas que incluem medição de audiência (ALVAREZ et al., 2009; AINA et al., 2014; LIM et al., 2014), publicidade direcionada (WANG; WANG; LU, 2013; KIM; KANG, 2011) e sistemas de recomendação (SANCHEZ et al., 2013; ABREU; ALMEIDA; TELES, 2014; BAMBINI; CREMONESI; TURRIN, 2012). Porém, coletar esses dados ainda é uma tarefa desafiadora (BAMBINI; CREMONESI; TURRIN, 2012).

4.1.1 Descrição do Problema

No ambiente web, as interações dos usuários são tipicamente utilizadas como dados de entrada para muitas aplicações. Muitas vezes, essas interações são capturadas de maneira transparente por meio da utilização de aplicações no lado cliente. Os usuários são identificados por meio de suas contas pessoais e dados de interação são registrados para uso futuro. Algumas empresas utilizam esses dados para melhorar a qualidade dos serviços oferecidos aos clientes (KAZIENKO; ADAMSKI, 2007; ROMERO et al., 2013). Essa melhora se dá pela análise do histórico de interações para descobrir padrões comportamentais dos usuários (SRIVASTAVA et al., 2000).

No ambiente da TV, a realidade da captura de dados é diferente. É de conhecimento comum que a atividade de assistir TV é coletiva em muitas residências. Pesquisas revelam que aproximadamente 58% dos telespectadores assistem TV em grupo com a família ou amigos (ABREU et al., 2013). Tipicamente, esses grupos são diversificados e seus membros são pessoas de diferentes idades, diferentes gêneros e com preferências variadas. Os SRs de conteúdo muitas vezes dependem da identificação precisa das pessoas que fazem parte do ambiente da TV para capturar os dados e posteriormente prover as recomendações baseadas na análise do histórico de interações. Infelizmente, fazer a identificação precisa dessas pessoas é um desafio por conta da própria natureza dos ambientes de TV.

O ambiente de TV é compartilhado e os usuários podem aleatoriamente entrar e sair desse ambiente. Esse tipo de comportamento dificulta a manutenção e atualização dos dados dos múltiplos usuários. Na tentativa de solucionar esse problema, pesquisas propõem a utilização de aplicações na tela da TV para que todos usuários presentes se identifiquem com usuário e senha (YU et al., 2006). Porém, esse tipo de interação não parece natural no contexto da utilização da TV e acabaria atrapalhando a experiência coletiva. A atividade de assistir TV precisaria ser interrompida para adicionar ou remover usuários que entram ou saem do ambiente aleatoriamente.

Além da identificação de quem está no ambiente de TV, outro problema é identificar quem foi o autor da interação. As interações em ambientes de TV tipicamente ocorrem por meio do controle remoto, que é utilizado como dispositivo principal de interação por aproximadamente 86% dos telespectadores (ALAM; KHUSRO; KHAN, 2019). Porém, por conta da natureza compartilhada desse dispositivo de interação, é difícil identificar quem foi o autor de uma dada interação com a TV. Em uma situação comum, um grupo inteiro (por exemplo, uma família) interagindo com uma TV é tratado como um único usuário. Ou seja, todas as interações são atribuídas ao grupo e não aos devidos autores. Dessa maneira, após a análise do histórico de interações, as preferências dos usuários que mais interagem com a TV acabam se sobrepondo às preferências dos usuários que interagem menos.

Os avanços nas tecnologias de TVDI possibilitaram a utilização de interações mais elaboradas que vão além da troca de canais e alteração de volume da TV tradicional. Porém, muitas vezes esses modelos mais elaborados de interação precisam ser adaptados aos modelos legados de controle remoto que ainda são amplamente utilizados. Algumas pesquisas propõem a utilização de uma aplicação baseada na web 2.0 para incrementar a interatividade e possibilitar a captura individualizada das interações (BARRAGÁNS-MARTÍNEZ et al., 2010). Dessa maneira, os usuários poderiam utilizar essa aplicação para se cadastrar, definir explicitamente as suas preferências, avaliar a programação, enviar comentários, etc. Porém, executar essas atividades relacionadas à web 2.0 é uma tarefa complexa e incômoda para ser realizada quando o dispositivo primário de interação com a TV é o controle remoto (ALAM; KHUSRO; KHAN, 2019). Talvez a interface mais adequada para a interação com a web 2.0 seja um computador com teclado e mouse, mas isso representa a introdução de um dispositivo estranho ao ambiente de TV.

Embora existam iniciativas para tentar resolver os problemas descritos aqui, os questionamentos a seguir ainda ficam em aberto sistematicamente:

- Quem realizou a interação?
- Qual interação foi realizada?

Nos trabalhos analisados, as duas questões não conseguem ser respondidas ao mesmo tempo no contexto de um ambiente de TV. Alguns trabalhos propõem a utilização de interações e dispositivos estranhos ao ambiente da TV. Outros trabalhos propõem a utilização do histórico de interações para a personalização de conteúdo, mas não dão detalhes de como os dados individualizados são capturados. Outra linha de pesquisa usualmente adotada é a personalização para um grupo de usuários, onde nenhum tipo de individualização é realizada.

4.1.2 Visão Geral da Proposta

Essa parte do trabalho propõe a utilização de dispositivos de tela secundária para capturar as interações individualmente, uma vez que esses dispositivos são pessoais e estão se estabelecendo naturalmente dentro do ambiente de TV (NIELSEN, 2019). A abordagem proposta permite a identificação de cada usuário por meio de um mecanismo de autenticação no dispositivo de tela secundária (onde a experiência é individual e não atrapalha a experiência coletiva). As interações executadas por cada usuário no dispositivo de tela secundária (por exemplo, navegação no EPG e avaliação de conteúdo) são capturadas e armazenadas em um repositório remoto para posterior processamento e utilização em sistemas de personalização de conteúdo. A Figura 3 apresenta uma visão geral

da proposta desenvolvida neste trabalho. Essa abordagem consiste em 4 módulos principais: Ambiente de TV (ATV), Serviço de Autenticação do Usuário (SAU), Provedor de Serviço de TV (PSTV) e Repositório de Dados dos Usuários (RDU).

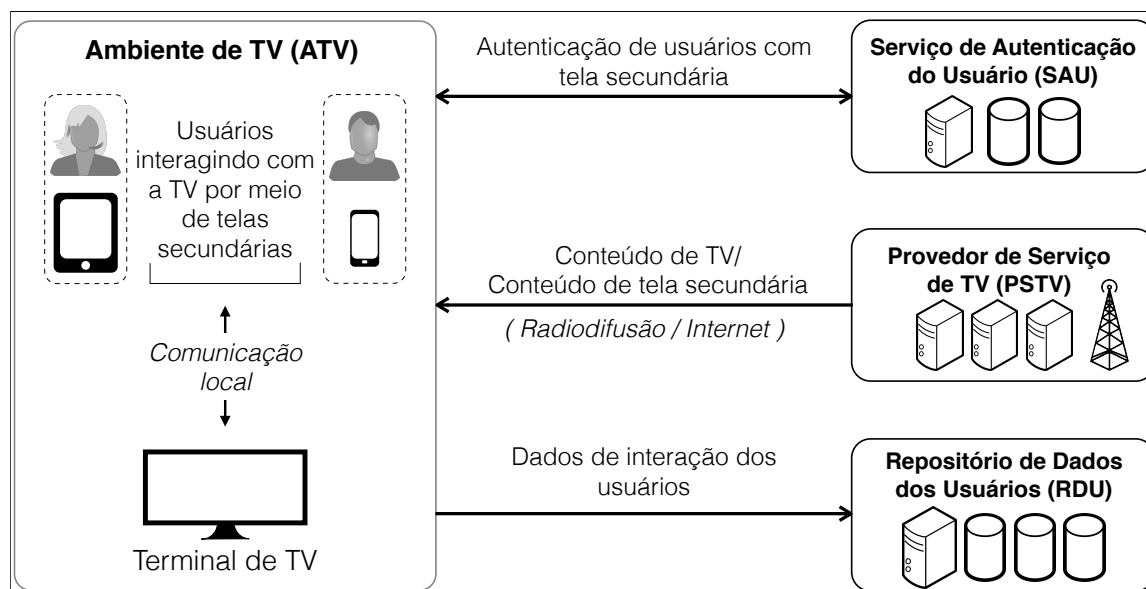


Figura 3: Visão geral da abordagem para a captura dos dados individuais de interação dos usuários.

O ATV consiste no espaço físico ou cômodo compartilhado pelos usuários ao assistir e interagir com o terminal de TV por meio dos seus dispositivos secundários. Os usuários que estão no ATV precisam se autenticar em suas contas pessoais criadas previamente utilizando o SAU. Essa autenticação é realizada individualmente sem atrapalhar a experiência coletiva é utilizada com o objetivo de manter um histórico individual de interações para cada usuário.

O PSTV é uma operadora de TV que disponibiliza a infraestrutura para transmitir o conteúdo até os receptores de TV no ATV. Nos padrões modernos de TVDI, os provedores disponibilizam conteúdo multimídia (vídeo, áudio e dados) por meio de diferentes sistemas de distribuição como satélites, redes de difusão terrestres e cabos. Isso coloca os sistemas de TV como poderosas plataformas de entrega de serviços de dados. Além disso, embora exista um custo de programação e produção do conteúdo interativo, o custo para embarcá-lo no sinal de difusão e distribuí-lo até milhares de receptores domésticos é baixo (RAO; BOJKOVIC; MILOVANOVIC, 2006; SRIVASTAVA, 2001).

A aquisição dos dados a partir do sinal de difusão permite a utilização de aplicações interativas que são tipicamente exibidas na tela da TV. Padrões modernos de TVDI também incluem a possibilidade de utilizar dispositivos de tela secundária, o que é compatível com as tendências de transformação do ambiente de TV (NIELSEN, 2013; NIELSEN, 2019). O padrão brasileiro, por exemplo, suporta a exibição de conteúdo em múltiplos dispositivos por meio da linguagem NCL (*Nested Context Language*), que pos-

sibilita dois modelos de interação: passivo e ativo (SOARES et al., 2009; ABNT, 2018a). No modelo passivo, o mesmo conteúdo é exibido em múltiplos dispositivos sob o controle de um dispositivo principal. No modelo ativo, o conteúdo é decodificado e renderizado por cada dispositivo individualmente, permitindo interação e navegação independentes.

Uma vez que nem todos os dispositivos de tela secundária contam com um receptor de TV embarcado para receber dados de interatividade por meio do sinal de difusão, a Internet pode ser utilizada como um meio alternativo para a aquisição de dados de interatividade. A utilização da Internet para a aquisição de dados não é um tópico de estudo novo no domínio da TV. Essa técnica tem sido estudada e implementada em pesquisas anteriores que utilizam serviços web baseados em nuvem para adquirir dados de conteúdo de TV (KULESZA et al., 2012; PIJETLOVIC et al., 2014). Além disso, essa técnica tira vantagem da maturidade técnica e da viabilidade de implementação de serviços de requisição e disponibilização de dados em várias linguagens de programação.

Neste trabalho, supõe-se que os dados sejam recuperados do PSTV pela Internet e apresentados na aplicação de tela secundária¹, permitindo a interação dos usuários com a aplicação de TVDI. As iterações que são executadas por cada usuário na tela secundária são capturadas e enviadas para o RDU por meio de requisições HTTP (*Hypertext Transfer Protocol*) à uma API (*Application Programming Interface*) de serviços web. Cada interação é identificada por um URI (*Uniform Resource Identifier*) da API de serviços, que pode ser customizada conforme a necessidade da aplicação seguindo o estilo arquitetural REST (*Representational State Transfer*). Dessa maneira, quando a camada de aplicação da TV requisita acesso a um serviço específico por meio de uma URI durante uma interação, é possível identificar *qual interação foi executada*. Os dados específicos da interação são enviados na carga de dados da requisição. É importante destacar que o propósito principal do RDU é armazenar os dados de interação dos usuários e não realizar processamento ou personalização.

A identificação de qual usuário está realizando a interação está diretamente relacionada com o mecanismo de autenticação do SAU. Cada interação capturada é rotulada com a identificação do usuário que a realizou. Uma vez que dispositivos móveis são usualmente para uso pessoal, pode-se assumir que as interações executadas em um dispositivo com uma conta autenticada estão relacionadas a um único usuário. Dessa maneira, consegue-se estabelecer *quem executou a interação*.

No ATV, a comunicação entre os dispositivos de tela secundária e o terminal de TV é realizada por meio de um processo de pareamento. Esse processo permite a comunicação entre os dispositivos durante uma sessão de pareamento, onde é possível capturar informações contextuais relevantes como o canal que está sendo assistido, programa, volume

¹ Essas aplicações também podem utilizar redes de comunicação local e redes de celular como canal de retorno para possibilitar a comunicação do terminal de TV com o provedor de TV.

e data/horário atuais.

4.1.3 Captura dos Eventos de Interação

A Figura 4 apresenta a arquitetura e os principais componentes da proposta descrita neste trabalho para a captura de dados dos usuários. No ATV, dois tipos de dispositivos físicos se comunicam entre si: os dispositivos de tela secundária e o terminal de TV. A condição inicial para o correto funcionamento dessa proposta é a existência das aplicações de software instaladas e sendo executadas no dispositivo de tela secundária e no terminal de TV. Essas aplicações são necessárias para habilitar os recursos de interatividade e comunicação entre os dispositivos por meio de serviços.

Muitos sistemas modernos permitem a criação de serviços que podem ser empacotados e distribuídos juntamente com as aplicações principais. Tipicamente, esses serviços são executados em segundo plano (não disponibilizam interface para o usuário) e continuam sendo executados mesmo quando outras aplicações estão sendo executadas na tela principal. Diversas ações podem ser executadas de maneira transparente pelos serviços em segundo plano. Dessa maneira, a experiência do usuário não é interrompida e ele não tem a percepção de que algo está sendo executado em segundo plano.

Após a instalação das aplicações/serviços, os usuários da aplicação de tela secundária precisam requisitar a autenticação utilizando nome de usuário e senha no SAU. Uma vez que a autenticação esteja efetuada, os usuários podem ser identificados e ter suas interações capturadas.

Os usuários que estão presentes no ATV utilizam seus dispositivos de tela secundária para se comunicar com o terminal de TV ao enviar comandos por meio de um protocolo de comunicação². A captura dos dados da interação é permitida ao explicitamente adicionar anotações no código fonte do trecho de código que é executado no acontecimento de um evento da interação³, como será apresentado mais adiante na Seção 5.1.1.1. Quando um evento de interesse acontece, os dados específicos da interação (incluindo o autor da interação) gerados por esse evento são automaticamente enviados ao RDU. Porém, para que isso aconteça, cada aplicação de tela secundária precisa ser customizada para que os eventos de interesse sejam identificados no código fonte.

Além dos dados específicos da interação, todas as vezes que um evento de interesse é executado, os dados contextuais daquele momento específico são associados ao evento. Esses dados contextuais consistem em localização, programa atual, data, hora, tipo de

² Os protocolos de comunicação são disponibilizados pela empresa desenvolvedora do sistema utilizado no terminal de TV. A utilização desses protocolos normalmente é efetuada por meio do kit de desenvolvimento da plataforma.

³ Um evento de interação é tipicamente o clique de um botão na interface do usuário. No acontecimento do evento, um trecho de código associado ao evento é automaticamente executado.

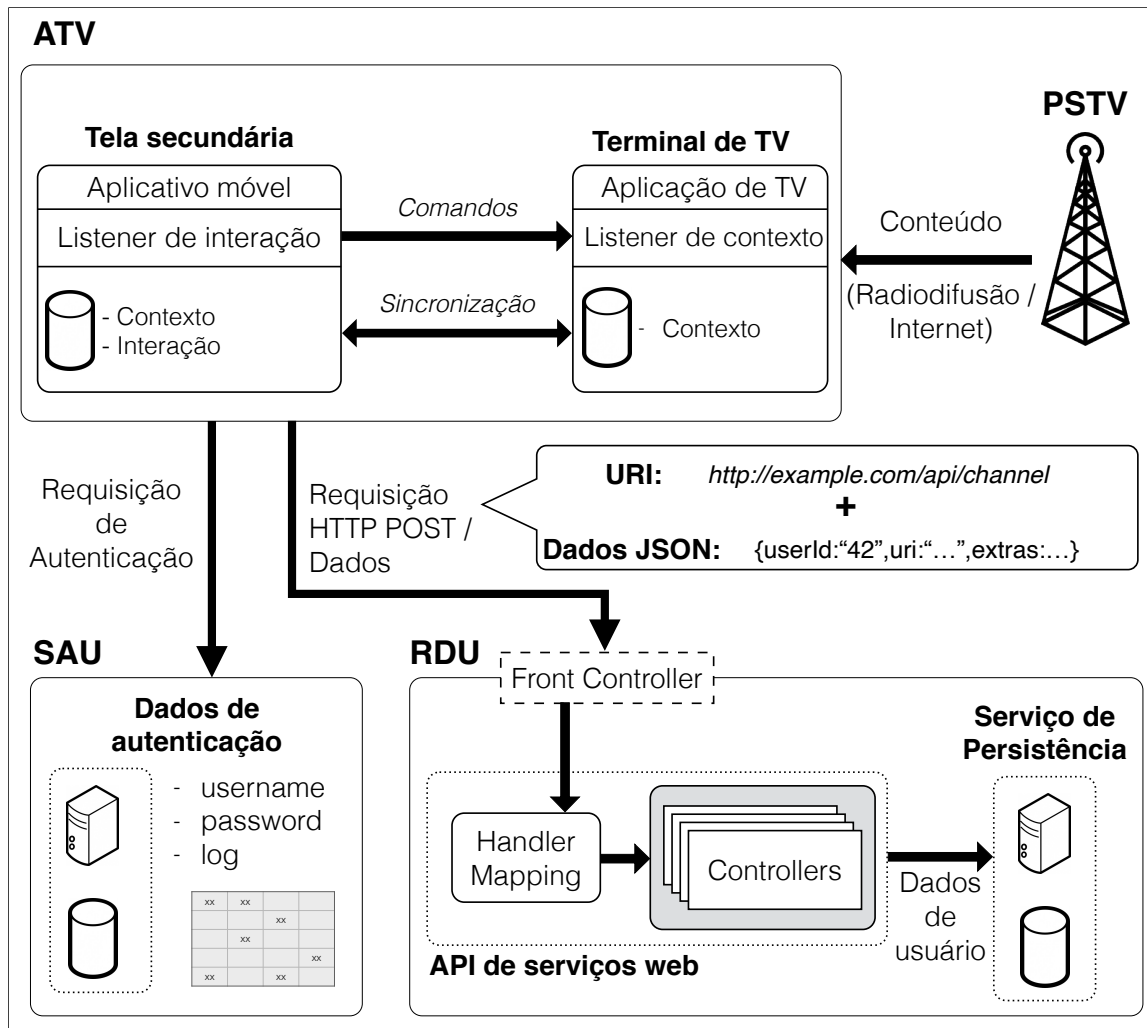


Figura 4: Arquitetura representando a integração dos componentes.

dispositivo sendo utilizado. A representação desses dados será apresentada posteriormente na Seção 4.1.4.

Para capturar os dados da interação, foram definidos dois tipos de *listeners* no ATV, identificados na Figura 4 como *listener de contexto* e *listener de interação*. O *listener de contexto* está presente no terminal de TV e é responsável por atualizar dados como o programa atual, canal atual, o momento em que a interação ocorre e a localização do terminal de TV (latitude e longitude). Esses são os dados contextuais e podem ser utilizados para complementar os dados gerados pelos eventos de interação.

O *listener de interação* está presente nos dispositivos de tela secundária e monitora eventos de interesse. Quando esses eventos de interesse ocorrem, os dados gerados pelo evento são capturados. Por exemplo, quando um usuário requisita informações sobre um programa em uma aplicação de EPG no dispositivo de tela secundária, essa interação é identificada por uma URI (por exemplo “`/epg/programs/info_request`”). No evento da interação, o *listener de interação* captura o ID do programa requisitado e o associa ao autor da interação. Além disso, os dados da interação podem ser complementados com os

dados contextuais adquiridos a partir do terminal de TV.

A comunicação entre o ATV e o RDU é realizada por meio de requisições HTTP dos dispositivos de tela secundária no ATV para o RDU. A base para essas requisições consiste na API de serviços web que identifica as interações e promove a interoperabilidade entre o ATV e o RDU. Uma vez que as requisições HTTP são executadas somente para persistir os dados capturados, somente o método POST⁴ é utilizado nas requisições.

No RDU, as requisições HTTP são tratadas por um padrão arquitetural chamado *front controller* (BUSCHMANN; HENNEY; SCHMIDT, 2007). Esse padrão é empregado para centralizar o processamento das requisições em um único componente, o que facilita o projeto e a implementação da API de serviços web. Cada requisição HTTP que é feita para o RDU é tratada pelo *front controller*, conforme pode ser visto na Figura 4. O *front controller* encaminha cada requisição para um componente de mapeamento que identifica um *controller* apropriado para tratar os dados de cada requisição. Uma vez que o *controller* apropriado é encontrado, os dados que foram submetidos são encaminhados para esse *controller*, onde é feito o armazenamento por meio de um serviço de persistência. Uma vez que os dados estejam armazenados, o PSTV pode acessá-los e analisá-los para obter qualquer informação que seja útil para o negócio.

No protótipo construído, a comunicação entre os dispositivos de tela secundária e o terminal de TV é realizada por meio de um mecanismo de comunicação WIFI P2P disponibilizado pela API do Android. Dessa maneira, sempre que uma interação ocorre, o dispositivo de tela secundária se conecta a um serviço de comunicação do terminal de TV para receber as informações contextuais mais recentes.

4.1.4 Representação dos Dados de Interação

A Figura 5 apresenta a estrutura básica para a representação dos dados de interação dos usuários. Essa estrutura contém os dados específicos de interação e os dados contextuais. Os dados são apresentados no formato *JavaScript Object Notation* (JSON), que é um formato de mídia amplamente utilizado para representação de dados na web. Embora outros formatos de representação de dados possam ser utilizados (como XML, Atom e YAML), o formato JSON foi escolhido por conta da popularidade e simplicidade. A estrutura apresentada na Figura 5 se propõe a resolver as questões apresentadas na Seção 4.1.1. Ou seja, *quem executou a interação e qual interação foi executada*.

As informações referentes aos dados de interação são obtidas a partir do ATV, seja do dispositivo de tela secundária ou do terminal de TV. Esses dados são estruturados e

⁴ Método HTTP projetado para requisitar o armazenamento de dados. Requisições feitas utilizando esse método requisitam que o servidor web aceite os dados enviados no corpo da requisição para que esses dados sejam possivelmente tratados e armazenados.

```
{
  userId: "...",
  uri: "...",
  extras: {
    //...
  },
  context: {
    //...
  }
}
```

Figura 5: Estrutura básica para a representação dos dados de interação.

```
...
userId: "28",
uri: "/epg/view",
extras: {
  channelId: "13",
  programId: "512"
},
...
```

Figura 6: Representação JSON dos dados específicos de interação.

enviados em uma requisição para que sejam armazenados no RDU. Os dados de interação são representados basicamente de duas maneiras: dados específicos de interação e dados contextuais. Uma explicação detalhada desses dados é apresentada nas próximas seções.

4.1.4.1 Representação dos Dados Específicos de Interação

Cada evento de interesse (interação) gera um conjunto de dados com propriedades e valores. Essas propriedades e valores representam os dados que são gerados pelo evento e indicam qual interação foi executada, assim como o autor da interação. A Figura 6 demonstra como os dados específicos de um evento chamado “*visualização de informações no EPG*” são apresentados no formato JSON.

A propriedade `userId` representa um identificador único para cada usuário do sistema. Essa propriedade indica *quem é o autor da interação*. A propriedade `uri` indica *qual interação foi executada*. No exemplo da Figura 6, os dados gerados pelo evento de visualização de informações do EPG são enviados ao serviço web por meio do valor da propriedade `uri`. Cada tipo de interação deve ser tratado por um serviço web específico para persistir os eventos separadamente. Ou seja, cada evento de interação possui uma propriedade `uri` específica. A propriedade `extras` descreve os dados relacionados à interação especificamente. No caso da Figura 6, as propriedades `channelId` e `programId` representam o canal e o programa visualizados durante a navegação no EPG. A proprie-

```
...
context: {
  currentProgram: { channelId: "4", programId: "512"},
  when: { date: "2014-12-12", time: "10:30:25" },
  location: {
    latitude: -3.090404,
    longitude: -59.963927
  },
  device: {
    type: "smartphone",
    os: "Android",
    osVersion: "4.4.2"
  }
}
...
```

Figura 7: Representação JSON dos dados contextuais que são utilizados para complementar os dados do evento de interação.

dade **extras** é flexível e pode receber diferentes propriedades e valores que dependem do tipo de interação executada.

Uma descrição textual para os dados apresentados na Figura 6 é dada a seguir: *um telespectador identificado pelo ID "28" visualizou informações referentes ao programa de TV identificado pelo ID "512" no canal identificado pelo ID "10"*. Essa descrição informa somente os dados relativos à interação como autor, tipo de interação e propriedades geradas pelo evento. A representação dos dados que podem dar informações mais detalhadas sobre o contexto em que a interação ocorreu é apresentada na próxima seção.

4.1.4.2 Representação dos Dados Contextuais

Além dos dados que são gerados pelo evento de uma interação, existem ainda os dados contextuais que podem ser utilizados para complementar os dados específicos da interação. Esses dados são definidos na propriedade **context** e não estão diretamente relacionados com *quem executou a interação* ou *qual interação foi executada*. Os dados contextuais são utilizados para complementar os eventos de interação do usuário quando necessário. A Figura 7 apresenta a representação dos dados contextuais no formato JSON.

A propriedade **currentProgram** contém um objeto com as propriedades **channelId** e **programId** que representam o conteúdo audiovisual que está sendo exibido na tela da TV no momento da interação⁵. O momento em que a interação ocorre é representado pela propriedade **when**, que é um objeto contendo as propriedades **date** e **time**. A propriedade

⁵ A nomenclatura utilizada nas propriedades *channelId* e *programId* são compatíveis com a utilizada no padrão brasileiro de TV Digital (ABNT, 2007).

`location` indica a latitude e longitude atual do telespectador. Por fim, a propriedade `device` descreve os dados sobre o dispositivo de tela secundária utilizado para interagir com o terminal de TV.

A descrição literal dos dados apresentados na Figura 7 pode ser apresentada da seguinte forma: *Uma interação foi executada no dia 1 de janeiro de 2021 às 10:30:25. No momento desta interação, o programa de TV foi identificado pelo ID “512” em um canal identificado pelo ID “10”. O usuário está localizado na latitude -3.090404 e longitude -59.963927. A interação foi realizada por meio de um dispositivo de tela secundária Android versão 4.4.2.* Essa descrição revela informações adicionais que podem ser úteis para os serviços que futuramente utilizarem os dados.

4.2 Recomendação de Conteúdo utilizando uma Vizinhança Relevante

Muitas técnicas de FC baseadas em memória são fundamentadas no conceito de vizinhança. Nesse conceito, pressupõe-se que usuários que atribuem preferências semelhantes aos mesmos objetos, provavelmente possuem gostos similares (CHEN et al., 2018). Essas semelhanças entre os usuários são muitas vezes capturadas por meio de métricas de similaridade. Dessa maneira, calcular uma matriz de similaridade se torna uma tarefa crítica em técnicas baseadas em memória (LIU et al., 2014). De maneira usual, os algoritmos utilizam os k usuários mais similares para prover recomendações, onde k representa o tamanho da vizinhança.

4.2.1 Descrição do Problema

Métricas tradicionais de similaridade incluem o PCC (RESNICK et al., 1994; SHARDANAND; MAES, 1995; COSTA et al., 2007) e a similaridade do cosseno (SARWAR et al., 2001). Além dessas técnicas, muitos pesquisadores propuseram novos modelos de similaridade para melhorar a qualidade das recomendações (LIU et al., 2014; WANG et al., 2017; POLATIDIS; GEORGIADIS, 2016; ZHANG et al., 2016; AYUB et al., 2019). De maneira geral, as métricas de similaridade assumem que existe uma relação linear entre dois usuários e ela depende de todos os objetos avaliados em comum por esses usuários. O problema dessa abordagem é que ela resulta em um valor de similaridade global para cada par de usuários, o que pode enfraquecer relacionamentos significativos entre esses usuários.

No mundo real, embora um usuário possa expressar suas preferências por uma ampla variedade de objetos, ele normalmente expressa mais interesse por objetos que estejam

relacionados a um tópico de interesse específico. Não por acaso, esse tópico de interesse tende a concentrar grande parte dos objetos observados e podem ativamente contribuir para a similaridade geral. Por exemplo, um profissional psicólogo tende a consumir um número maior de livros da área de psicologia do que de outras áreas do conhecimento. Assim, mesmo que esse profissional consuma livros de áreas secundárias de interesse (provavelmente em menor número), os livros relacionados à área de psicologia tendem a impactar fortemente no valor global da similaridade pelo fato de estarem em maior número.

Uma das implicações de se considerar um valor global de similaridade é que, para um dado usuário ativo u_i , os usuários vizinhos sempre contribuem na mesma quantidade para prever a preferência para qualquer objeto de interesse. Em outras palavras, durante a agregação para prever a preferência de u_i por qualquer objeto de interesse, um vizinho sempre vai contribuir com o mesmo valor de similaridade. Outra implicação é que um dado usuário, que seja similar ao usuário ativo u_i , tem alto potencial para compor a vizinhança de u_i mesmo que ambos apresentem preferências divergentes dentro do grupo de objetos relacionados ao objeto de interesse. Esse último caso pode acabar levando à baixa qualidade das predições para grupos de objetos onde há divergência de preferências.

4.2.2 Visão Geral da Proposta

Uma vez que os usuários podem expressar preferências particulares por diferentes grupos de objetos relacionados, uma maneira mais natural de calcular a similaridade levaria em consideração esses diferentes grupos de objetos. Como resultado, em vez de um único valor de similaridade, dois usuários quaisquer teriam $\#\mathcal{C}$ valores de similaridade, onde \mathcal{C} é o conjunto dos *clusters* de objetos. Ou seja, dois usuários possuem um valor de similaridade para cada *cluster* e esse valor é calculado com base nos objetos observados em comum dentro do *cluster*.

Em um exemplo do mundo real para demonstrar a aplicabilidade desse raciocínio, pode-se considerar um sistema de avaliação de filmes onde os usuários atribuem suas preferências para cada filme assistido. Ao se considerar as preferências de dois usuários quaisquer para filmes da categoria de ação, a convergência dessas preferências se traduz em alta similaridade dentro desse grupo de filmes. Com isso, esses usuários são potenciais candidatos para compor a vizinhança um do outro na predição das preferências de filmes do gênero ação. Agora, considerando que as preferências desses mesmos usuários sejam divergentes para filmes da categoria comédia, a similaridade entre eles tende a ser baixa nessa categoria. Como resultado, esses usuários não seriam selecionados para compor a vizinhança um do outro para filmes do gênero comédia.

A Tabela 4 apresenta uma situação hipotética e simplificada para ilustrar diferentes cenários de similaridade. Nesse exemplo, o objetivo é verificar se os usuários u_2 e u_3 podem

Tabela 4: Exemplo de uma matriz de avaliações R com 3 usuários e 8 objetos. As preferências são indicadas em uma escala de 1 a 5. O usuário u_1 é considerado o usuário ativo e os objetos são categorizados em dois grupos: grupo A e grupo B.

	Grupo A				Grupo B			
	o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8
u_1	\emptyset	3	5	3	5	5	4	\emptyset
u_2	4	5	1	\emptyset	4	4	3	1
u_3	4	3	4	3	2	\emptyset	5	5

ser selecionados como vizinhos de um usuário ativo u_1 . As preferências são representadas por uma matriz usuário-objeto onde cada valor indica a avaliação que um usuário atribuiu a um objeto. O símbolo \emptyset é utilizado para denotar que a avaliação de um usuário para um objeto é desconhecida. O valor da similaridade $s_{u_i u_j}$ é calculado usando uma função $sim(u_i, u_j) \rightarrow [-1, 1]$ que leva em conta os objetos avaliados em comum pelos dois usuários u_i e u_j . Para esse exemplo em particular, o PCC é utilizado como métrica para calcular a similaridade considerando os objetos em três cenários: todos os objetos, objetos do grupo A e objetos do grupo B.

No primeiro cenário, os valores de similaridade obtidos entre o usuário u_1 e os demais usuários são os seguintes: $s_{u_1 u_2} = -0,5$ e $s_{u_1 u_3} = 0$. De acordo com esses valores, os usuários u_2 e u_3 não seriam selecionados para formar a vizinhança porque os seus valores de similaridade com o usuário ativo u_1 não são maiores do que 0. Esse cenário é particularmente interessante porque ele considera o conjunto de objetos inteiro. Assim, não existe distinção entre categorias de objetos e o resultado é um único valor de similaridade global.

No segundo cenário, que considera somente os objetos do grupo A, os valores de similaridade são: $s_{u_1 u_2} = -1$ e $s_{u_1 u_3} = 1$. Uma vez que $s_{u_1 u_3} > 0$, o usuário u_3 seria selecionado para prever as preferências do usuário ativo u_1 para objetos que pertençam ao grupo A.

No terceiro cenário, que considera somente os objetos pertencentes ao grupo B, os valores de similaridade são: $s_{u_1 u_2} = 0,99$ e $s_{u_1 u_3} = -1$. Nesse último cenário, uma vez que somente $s_{u_1 u_2} > 0$, o usuário u_2 seria o único selecionado para formar a vizinhança do usuário ativo u_1 para prever as preferências por objetos do grupo B. Diferentemente do primeiro cenário, os dois últimos cenários consideram somente um subconjunto particular do conjunto de todos os objetos para calcular a similaridade. Como resultado, é possível verificar que a similaridade global pode descartar vizinhos que podem ser altamente similares e relevantes dentro de contextos específicos. Dessa maneira, essa parte do trabalho propõe uma metodologia para a identificação desses usuários relevantes dentro de contextos particulares relacionados a categorias de objetos semelhantes.

4.2.3 Método Proposto

A Figura 8 apresenta uma visão geral do método de recomendação proposto. Esse método é formado por dois processos: modelagem e predição. Cada processo é composto por um ou mais estágios e cada estágio possui uma ou mais etapas.

O processo de modelagem tem o conjunto de dados de treinamento como entrada para o processo, enquanto que a saída é o modelo de similaridade local baseado em *cluster* (CBLIS⁶), que é utilizado durante o processo de predição. No processo de predição, as entradas são o conjunto de dados de teste e o modelo CBLIS. A saída desse processo é um conjunto de predições de preferências para as entradas do conjunto de testes.

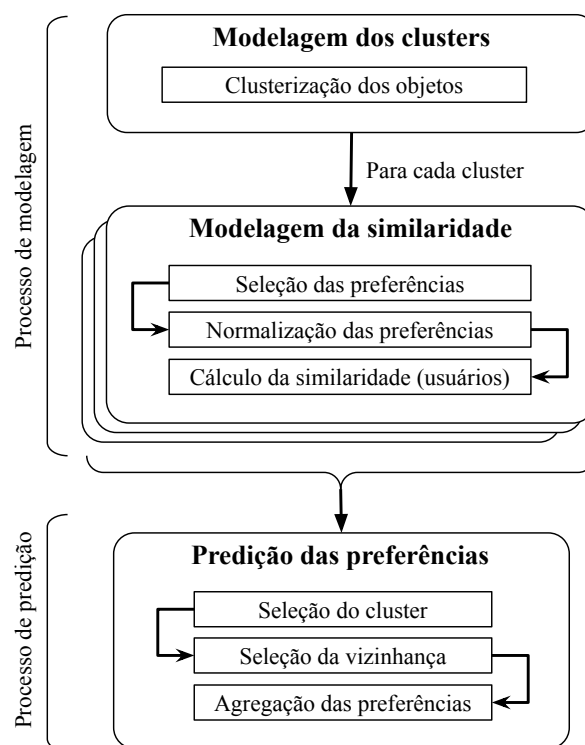


Figura 8: Visão geral do método proposto. O método consiste em dois processos: modelagem e predição. O processo de modelagem visa a construção dos modelos de similaridade local para cada *cluster*, e o processo de predição é focado em estimar as preferências de usuários em objetos cujo $r_{u_i o_k} = \emptyset$.

Para melhor entendimento do método, os estágios e etapas de cada processo são descritos detalhadamente a seguir. Também, são apresentadas as entradas e saídas esperadas de cada uma das etapas.

⁶ *Cluster-based Local Similarity*

4.2.3.1 Processo de Modelagem

No processo de modelagem, primeiramente os objetos similares são agrupados em *clusters*. Então, para cada *cluster*, é construído um modelo CBLIS baseado em usuários levando em conta somente as preferências atribuídas aos objetos que pertencem ao *cluster*. De maneira mais detalhada, o processo de modelagem consiste em dois estágios: modelagem do *cluster* e modelagem da similaridade.

O estágio de modelagem do *cluster* é independente dos detalhes da implementação do algoritmo de clusterização. Dessa maneira, o método proposto nesse trabalho não fica vinculado a um algoritmo de clusterização em particular. Uma vez que os resultados experimentais desse trabalho são obtidos utilizando os algoritmos de clusterização *affinity propagation* e *K-Means*, o estágio de modelagem dos *clusters* é descrito para cada um deles⁷. Entretanto, é importante destacar que qualquer um desses algoritmos pode ser utilizado, inclusive outros algoritmos que não são descritos nesse trabalho.

Estágio de Modelagem dos *Clusters* (*Affinity Propagation*)

O algoritmo *affinity propagation* é uma técnica de clusterização baseada em similaridade. Essa técnica depende de uma matriz de similaridade criada a partir da matriz de avaliações. A etapa de clusterização dos objetos é definida a seguir:

- **Clusterização dos objetos:**

Entradas: (1) matriz de avaliações $R^{m \times n}$.

Descrição: A matriz de avaliações é utilizada para criar uma matriz de similaridade $S^{n \times n}$ baseada em objetos. Embora Frey e Dueck (2007) utilize a distância Euclidiana quadrática negativa para computar as similaridades, o algoritmo *affinity propagation* desacopla o algoritmo de clusterização dos detalhes da implementação da métrica de similaridade, o que traz flexibilidade e possibilita a utilização de outras métricas. Após uma extensiva avaliação experimental utilizando PCC, cosseno e RA como métricas de similaridade, concluiu-se que o método proposto nesse trabalho tem melhor performance quando o PCC é utilizado como métrica de similaridade. A matriz de similaridade baseada em objetos é utilizada como entrada para o algoritmo de clusterização. Nem a quantidade de *clusters* e nem os próprios *clusters* são conhecidos antecipadamente. Eles emergem gradualmente durante o procedimento de passagem de mensagens. Ao fim da etapa de clusterização de objetos, um conjunto de *clusters* é obtido e cada *cluster* é um subconjunto dos objetos em \mathcal{O} . O número de *clusters* depende de dois parâmetros do algoritmo *affinity propagation*: preferência (*preference*) e amortecimento (*damping*). Esses parâmetros podem

⁷ Esses dois algoritmos são apresentados em detalhes no APÊNDICE B

ser ajustados para estabilizar a quantidade de *clusters*. Entretanto, é importante destacar que quanto maior for o número de *clusters*, menor será a quantidade de avaliações a ser considerada durante o estágio de modelagem da similaridade. Esse pouca quantidade de dados pode levar à baixa qualidade da similaridade entre os usuários.

Saída: rótulos dos *clusters* para cada objeto do conjunto de dados.

Estágio de Modelagem dos *Clusters* (*K-Means*)

A etapa de clusterização dos objetos utilizando o algoritmo de clusterização *K-Means* é descrito a seguir:

- **Clusterização dos objetos:**

Entrada: (1) matriz de avaliações $R^{m \times n}$; (2) número de *clusters*.

Descrição: diferentemente do *affinity propagation*, no algoritmo *K-Means*, o número desejado de *clusters* é especificado antecipadamente como parâmetro de entrada. O objetivo do algoritmo é encontrar partições que minimizem a distância entre os pontos e os centroides dos *clusters*. Os objetos na matriz de avaliações (colunas) são os pontos de dados utilizados pelo algoritmo durante o cálculo.

Saída: rótulos dos *clusters* para cada objeto do conjunto de dados.

Estágio de Modelagem da Similaridade

O estágio de modelagem de similaridade é executado para cada *cluster*. Dado um *cluster* de interesse, esse estágio consiste em 3 etapas, conforme apresentado no Algoritmo 1:

Algoritmo 1 Estágio de modelagem da similaridade

Require: \mathcal{C}

▷ Todos os *clusters*.

- 1: **for all** $\mathcal{C}_c \in \mathcal{C}$ **do**
 - 2: Seleção das preferências
 - 3: Normalização das preferências
 - 4: Cálculo das similaridades
 - 5: **end for**
-

- **Seleção das preferências:**

Entrada: (1) base de dados de treinamento; (2) *cluster* de interesse.

Descrição: todas as preferências atribuídas aos objetos que compõem o *cluster* de interesse devem ser selecionadas. Uma vez que a intenção é produzir um modelo de similaridade local, os dados utilizados no modelo são limitados ao *cluster* de

interesse em vez de utilizar todo o conjunto de dados. O Algoritmo 2 descreve os passos necessários para a seleção das preferências a partir dos dados de treinamento. Cada entrada do conjunto de treinamento \mathcal{R}_{train} é formada pela trinca $(u_i, o_k, r_{u_i o_k})$, que corresponde a usuário, objeto e avaliação, respectivamente. O conjunto das preferências selecionadas $\mathcal{S}_{\mathcal{C}_c}$ é inicializado como vazio (\emptyset) e vai acumulando as trincas cujos objetos pertencem ao cluster \mathcal{C}_c .

Saída: preferências selecionadas.

Algoritmo 2 Seleção das preferências.

Require: \mathcal{C}_c ▷ Cluster de interesse formado por objetos.
Require: \mathcal{R}_{train} ▷ Conjunto de dados de treinamento.
 1: $\mathcal{S}_{\mathcal{C}_c} \leftarrow \emptyset$ ▷ Conjunto vazio das preferências selecionadas.
 2: **for all** $(u_i, o_k, r_{u_i o_k}) \in \mathcal{R}_{train}$ **do**
 3: **if** $o \in \mathcal{C}_c$ **then**
 4: $\mathcal{S}_{\mathcal{C}_c} \leftarrow \mathcal{S}_{\mathcal{C}_c} + (u_i, o_k, r_{u_i o_k})$ ▷ Seleciona entrada
 5: **end if**
 6: **end for**

- **Normalização das preferências:**

Entrada: (1) preferências selecionadas; (2) usuários que avaliaram objetos no cluster.

Descrição: as preferências selecionadas são normalizadas utilizando uma função z-score sensível aos dados do *cluster*:

$$z_{u_i o_k}^{\mathcal{C}_c} = \frac{r_{u_i o_k} - \bar{r}_{u_i}^{\mathcal{C}_c}}{\sigma_{u_i}^{\mathcal{C}_c}} \quad (4.1)$$

onde $\bar{r}_{u_i}^{\mathcal{C}_c}$ e $\sigma_{u_i}^{\mathcal{C}_c}$, respectivamente, são a média e o desvio padrão das preferências do usuário u_i considerando somente os objetos pertencentes ao *cluster*. A Equação (4.1) é a mesma função *z-score* (CACHEDA et al., 2011; HERLOCKER; KONSTAN; RIEDL, 2002) apresentada na Seção 2.4.1.1. Porém, no contexto desse trabalho, somente os dados referentes ao *cluster* de interesse são utilizados para a normalização. Os algoritmos 3 e 4 descrevem os passos para o cálculo da média e do desvio padrão das avaliações dos usuários e para o cálculo da normalização. $\mathcal{U}_{\mathcal{C}_c}$ representa o conjunto de todos os usuários que avaliaram algum objeto do *cluster* \mathcal{C}_c .

Saída: preferências normalizadas.

- **Cálculo das similaridades (usuários):**

Entrada: (1) preferências normalizadas; (2) preferências selecionadas; (3) recursos alocados para os usuários que avaliaram objetos no *cluster* de interesse; (4).

Descrição: essa etapa descreve como o modelo CBLS é criado. Em vez de usar as preferências brutas (reais), as preferências normalizadas são utilizadas para calcular

Algoritmo 3 Cálculo da média e do desvio padrão considerando objetos pertencentes ao *cluster* de interesse \mathcal{C}_c .

Require: $\mathcal{S}_{\mathcal{C}_c}$ ▷ Conjunto das preferências selecionadas.
Require: $\mathcal{U}_{\mathcal{C}_c} = \{u_i \in \mathcal{U} : r_{u_i o_k} \neq \emptyset \wedge o_k \in \mathcal{C}_c\}$
1: **for all** $u_j \in \mathcal{U}_{\mathcal{C}_c}$ **do** ▷ Inicializa a soma.
2: $soma_{u_j} \leftarrow 0$
3: $count_{u_j} \leftarrow 0$ ▷ Inicializa o contador.
4: **for all** $(u_i, o_k, r_{u_i o_k}) \in \mathcal{S}_{\mathcal{C}_c}$ **do**
5: **if** $u_j = u_i$ **then**
6: $soma_{u_j} \leftarrow soma_{u_j} + r_{u_i o_k}$
7: $count_{u_j} \leftarrow count_{u_j} + 1$
8: **end if**
9: **end for**
10: $\bar{r}_{u_j}^{\mathcal{C}_c} \leftarrow soma_{u_j} / count_{u_j}$ ▷ Cálculo da média.
11: $dev_{u_j} \leftarrow 0$
12: **for all** $(u_i, o_k, r_{u_i o_k}) \in \mathcal{S}_{\mathcal{C}_c}$ **do**
13: **if** $u_j = u_i$ **then**
14: $dev_{u_j} \leftarrow dev_{u_j} + (r_{u_i o_k} - \bar{r}_{u_i o_k}^{\mathcal{C}_c})^2$
15: **end if**
16: **end for**
17: $\sigma_{u_j}^{\mathcal{C}_c} \leftarrow \sqrt{dev_{u_j} / count_{u_j}}$ ▷ Cálculo do desvio padrão.
18: **end for**

Algoritmo 4 Normalização das preferências considerando o *cluster* \mathcal{C}_c .

Require: $\mathcal{S}_{\mathcal{C}_c}$ ▷ Conjunto das preferências selecionadas.
1: $\mathcal{S}_{\mathcal{C}_i}^z \leftarrow \emptyset$ ▷ Conjunto vazio das preferências normalizadas.
2: **for all** $(u_i, o_k, r_{u_i o_k}) \in \mathcal{S}_{\mathcal{C}_c}$ **do**
3: $z_{u_i o_k}^{\mathcal{C}_c} = (r_{u_i o_k} - \bar{r}_{u_i}^{\mathcal{C}_c}) / \sigma_{u_i}^{\mathcal{C}_c}$
4: $\mathcal{S}_{\mathcal{C}_i}^z \leftarrow \mathcal{S}_{\mathcal{C}_i}^z + (u_i, o_k, z_{u_i o_k}^{\mathcal{C}_c})$
5: **end for**

a similaridade local entre os usuários. A similaridade local $\lambda_{u_i u_j}$ entre os usuários u_i e u_j é denotada por:

$$\lambda_{u_i u_j} = \frac{\sum_{o_k \in \mathcal{O}_{u_i u_j}} (z_{u_i o_k}^{\mathcal{C}_c} - \bar{z}_{u_i}^{\mathcal{C}_c})(z_{u_j o_k}^{\mathcal{C}_c} - \bar{z}_{u_j}^{\mathcal{C}_c})}{\sqrt{\sum_{o_k \in \mathcal{O}_{u_i u_j}} (z_{u_i o_k}^{\mathcal{C}_c} - \bar{z}_{u_i}^{\mathcal{C}_c})^2} \sqrt{\sum_{o_k \in \mathcal{O}_{u_i u_j}} (z_{u_j o_k}^{\mathcal{C}_c} - \bar{z}_{u_j}^{\mathcal{C}_c})^2}} \quad (4.2)$$

onde $\bar{z}_{u_i}^{\mathcal{C}_c}$ é a média das preferências normalizadas para o usuário u_i considerando somente os objetos pertencentes ao cluster. A Equação (4.2) é a mesma do PCC (RESNICK et al., 1994; SHARDANAND; MAES, 1995; COSTA et al., 2007). Porém, em vez de considerar todos os dados do conjunto de treinamento para o cálculo da similaridade, no contexto desse trabalho, a equação utiliza somente os dados referentes ao *cluster* de interesse. Além disso, foi incorporado um procedimento de alocação de recursos como um método de confiabilidade para enriquecer o modelo de similaridade, conforme foi proposto por Javari, Gharibshah e Jalili (2014) (Equação (4.3)).

As preferências que foram selecionadas na etapa de *seleção das preferências* são utilizadas para produzir uma matriz de similaridade baseada em grau (que foi apresentada na Seção 2.6). Essa matriz é utilizada para atualizar a similaridade local entre os usuários u_i e u_j conforme a seguir:

$$s_{u_i u_j}^{\text{CBLS}} = \lambda_{u_i u_j} \cdot w_{u_i u_j}^{\mathcal{C}_c} \quad (4.3)$$

onde $s_{u_i u_j}^{\text{CBLS}}$ e $w_{u_i u_j}^{\mathcal{C}_c}$, representam respectivamente, o valor do modelo de similaridade local CBLS e o recurso alocado entre os usuários u_i e u_j considerando somente os objetos pertencentes ao *cluster* de interesse. A cálculo da similaridade é realizado utilizando o Algoritmo 5.

Saída: matriz de similaridade local baseada em usuário para o *cluster* de interesse.

Algoritmo 5 Cálculo das similaridades para os usuários.

Require: \mathcal{S}_c ▷ Conjunto das preferências selecionadas.
Require: $\mathcal{U}_c = \{u_i \in \mathcal{U} : r_{u_i o_k} \neq \emptyset \wedge o_k \in \mathcal{C}_c\}$
Require: $W = \{w_{u_i u_j}^{\mathcal{C}_c}\}$ ▷ Recursos alocados para o *cluster*.
Require: $\mathcal{O}_{u_i u_j}$ ▷ Objetos avaliados em comum.

for all $u_i \in \mathcal{U}_c$ **do**
 for all $u_j \in \mathcal{U}_c$ **do**
 if $u_j \neq u_i$ **then**
 $sumDevz \leftarrow 0$
 $sumSDevui \leftarrow 0$
 $sumSDevuj \leftarrow 0$
 for all $o_k \in \mathcal{O}_{u_i u_j}$ **do**
 if $o_k \in \mathcal{C}_c$ **then**
 $sumDevz \leftarrow sumDevz + (z_{u_i o_k}^{\mathcal{C}_c} - \bar{z}_{u_i}^{\mathcal{C}_c}) \times (z_{u_j o_k}^{\mathcal{C}_c} - \bar{z}_{u_j}^{\mathcal{C}_c})$
 $sumSDevui \leftarrow sumSDevui + (z_{u_i o_k}^{\mathcal{C}_c} - \bar{z}_{u_i}^{\mathcal{C}_c})^2$
 $sumSDevuj \leftarrow sumSDevuj + (z_{u_j o_k}^{\mathcal{C}_c} - \bar{z}_{u_j}^{\mathcal{C}_c})^2$
 end if
 end for
 $\lambda_{u_i u_j} \leftarrow sumDevz / (\sqrt{sumSDevui} \times \sqrt{sumSDevuj})$
 $s_{u_i u_j}^{\text{CBLS}} \leftarrow \lambda_{u_i u_j} \times w_{u_i u_j}^{\mathcal{C}_c}$
 end if
 end for
end for

Como resultado, cada *cluster* possui o seu respectivo modelo de similaridade local que leva em conta somente as preferências associadas aos objetos que compõem o *cluster* de interesse. Cada modelo representa a matriz de similaridade baseada em usuário que é obtida a partir de objetos que são provavelmente similares ou relacionados por fazerem parte do mesmo *cluster*.

4.2.3.2 Processo de Predição

Durante o processo de predição, os modelos de similaridade local obtidos como saída do processo de modelagem são utilizados para prever as preferências de cada usuário ativo por um objeto de interesse. O processo de predição possui somente um estágio: predição das preferências. Este estágio consiste em três etapas:

- **Seleção do *cluster*:**

Entrada: (1) Rótulo dos *clusters*; (2) objeto de interesse.

Descrição: o *cluster* de interesse é selecionado de acordo com o rótulo do objeto de interesse. Uma vez que cada *cluster* possui um modelo de similaridade local baseado em usuário, saber qual é o *cluster* do objeto de interesse é fundamental para selecionar o modelo que melhor representa os usuários que usualmente atribuem suas preferências aos objetos contidos no *cluster*.

Saída: modelo de similaridade local para o *cluster* selecionado.

- **Seleção da vizinhança:**

Entrada: (1) modelos de similaridade local; (2) usuário ativo; (3) objeto de interesse; (4) quantidade de vizinhos (k).

Descrição: essa etapa é responsável por selecionar a vizinhança que será utilizada na etapa de agregação. Dado um usuário ativo u_i , e um objeto de interesse o_k , as abordagens tradicionais para seleção de vizinhança usualmente selecionam no máximo k usuários filtrados de acordo com as seguintes condições Cacheda et al. (2011):

- Usuários que selecionaram o objeto o_k .
- Usuários cujo valor da similaridade com o usuário u_i , ultrapasse um determinado limite (usualmente 0, que também é o valor utilizado como referência nesse trabalho).
- Usuários que são mais similares ao usuário u_i , de acordo com a matriz de similaridade.

A metodologia descrita acima é a mesma utilizada nesse trabalho. Porém, enquanto as abordagens tradicionais focam na similaridade global para encontrar os usuários mais similares, esse trabalho foca na similaridade local, que se baseia no *cluster* do objeto de interesse. O Algoritmo 6 descreve os passos necessários para a realização dessa etapa.

Saída: k vizinhos mais similares.

Algoritmo 6 Seleção da vizinhança considerando um *cluster* de interesse \mathcal{C}_c .

Require: $S = \{s_{u_i u_j}^{\text{CBLS}}\}$ ▷ Modelo de similaridade local.
Require: u_i ▷ Usuário ativo.
Require: o_k ▷ Objeto de interesse.
Require: k ▷ Quantidade de vizinhos.
1: $\mathcal{N}_{u_i o_k}^{\mathcal{C}_c} \leftarrow \emptyset$ ▷ Inicializa o conjunto de vizinhos.
2: **for all** $u_j \in \mathcal{U}_{o_k}$ **do**
3: **if** $u_i \neq u_j \wedge s_{u_i u_j}^{\text{CBLS}} > 0$ **then**
4: $\mathcal{N}_{u_i o_k}^{\mathcal{C}_c} \leftarrow \mathcal{N}_{u_i o_k}^{\mathcal{C}_c} + u_j$ ▷ Adiciona u_j à vizinhança.
5: **end if**
6: **end for**
7: $\mathcal{N}_{u_i o_k}^{\mathcal{C}_c} \leftarrow \text{top}k(\mathcal{N}_{u_i o_k}^{\mathcal{C}_c}, k)$ ▷ Seleciona somente os k usuários mais similares.

- **Agregação das preferências:**

Entrada: (1) k vizinhos mais similares; (2) usuário ativo; (3) objeto de interesse; (4) preferências normalizadas; (5) Modelo de similaridade local

Descrição: dado um usuário u_i e um objeto de interesse o_k , as preferências normalizadas dos k usuários mais semelhantes são agregadas conforme um método proposto por Resnick et al. (1994), que é amplamente utilizado na literatura:

$$\hat{r}_{u_i o_k} = \bar{r}_{u_i}^{\mathcal{C}_c} + \frac{\sum_{u_j \in \mathcal{N}_{u_i o_k}^{\mathcal{C}_c}} z_{u_j o_k}^{\mathcal{C}_c} \cdot s_{u_i u_j}^{\text{CBLS}}}{\sum_{u_j \in \mathcal{N}_{u_i o_k}^{\mathcal{C}_c}} |s_{u_i u_j}^{\text{CBLS}}|} \quad (4.4)$$

onde $\mathcal{N}_{u_i o_k}^{\mathcal{C}_c}$ é o conjunto dos k vizinhos mais similares considerando o *cluster* do objeto de interesse. Ou seja, usuários que são mais semelhantes ao usuário u_i que também avaliaram o objeto o_k , de acordo com o modelo de similaridade local. A notação $z_{u_j o_k}^{\mathcal{C}_c}$ representa a preferência normalizada de acordo com o *cluster* do objeto de interesse.

Saída: preferência predita.

4.3 Bases de Dados

A performance do método proposto foi avaliada por meio de experimentos utilizando três bases de dados amplamente adotadas como referência na literatura de FC: MovieLens 100k (ML 100k) (HARPER; KONSTAN, 2015), MovieLens 1M (ML 1M) (HARPER; KONSTAN, 2015) e Netflix (BENNETT; LANNING, 2007). Os detalhes dessas bases de dados são descritos a seguir:

ML 100k: o grupo de pesquisa *GroupLens Research Group* coletou as bases de dados MovieLens na Universidade de Minnesota como produto do SR de filmes chamado

Algoritmo 7 Agregação das preferências.

Require: $\mathcal{N}_{u_i o_k}^{\mathcal{C}_c}$ ▷ Vizinhança.
Require: u_i ▷ Usuário ativo.
Require: o_k ▷ Objeto de interesse.
Require: $S = \{s_{u_i u_j}^{\text{CBLS}}\}$ ▷ Modelo de similaridade local.

- 1: $sumNum \leftarrow 0$
- 2: $sumDen \leftarrow 0$
- 3: **for all** $u_j \in \mathcal{N}_{u_i o_k}^{\mathcal{C}_c}$ **do**
- 4: $sumNum \leftarrow sumNum + z_{u_j o_k}^{\mathcal{C}_c} \times s_{u_i u_j}^{\text{CBLS}}$
- 5: $sumDen \leftarrow sumDen + |s_{u_i u_j}^{\text{CBLS}}|$
- 6: **end for**
- 7: $\hat{r}_{u_i o_k} \leftarrow \bar{r}_{u_i}^{\mathcal{C}_c} + sumNum / sumDen$

MovieLens. Essas bases de dados foram amplamente utilizadas na literatura de pesquisas envolvendo SRs. A base ML 100k contém 100.000 avaliações de 943 usuários em 1.682 filmes. As avaliações são expressas em uma escala de 1 (pior avaliação) a 5 (melhor avaliação). Essa base de dados tem uma média de 106,04 avaliações por usuário e cada usuário avaliou pelo menos 20 filmes. O usuário mais ativo (ou seja, o usuário que mais avaliou filmes), avaliou 737 filmes. O nível de esparsidade da base de dados é de 93,7% (ou seja, somente 6,3% de todas as células da matriz de avaliações foram observadas).

O nível de esparsidade é calculado da seguinte maneira:

$$esparsidade = 100 \cdot \left(1 - \frac{\text{todas as avaliações observadas}}{m \cdot n} \right) \quad (4.5)$$

onde m é o número total de usuários e n é o número total de objetos.

ML 1M: essa base de dados contém 1.000.209 avaliações de 6.040 usuários em 3.900 filmes, o que representa um nível de esparsidade de 95,5%. Cada usuário avaliou pelo menos 20 filmes. As avaliações contidas nessa base de dados são expressas em uma escala de 1 (pior avaliação) a 5 (melhor avaliação). Essa base de dados tem uma média de 165,60 avaliações por usuário e o usuário mais ativo avaliou 2.314 filmes.

Netflix: a base de dados Netflix contém 100 milhões de avaliações e ela foi disponibilizada durante a competição *Netflix Prize*. Nos experimentos realizados nesse trabalho, foi utilizado um subconjunto da base de dados do Netflix. Esse subconjunto consiste em 1.378.159 avaliações de 6.000 usuários em 6.344 filmes, onde usuários e filmes foram aleatoriamente selecionados e filtrados de tal modo que cada usuário avaliou pelo menos 17 filmes. A base de dados tem uma média de 229,6 filmes por usuário. O usuário mais ativo avaliou 3.894 filmes. O subconjunto da base de dados utilizado nos experimentos tem um nível de esparsidade de 96,4% e as avaliações são expressas em uma escala de 1 (pior avaliação) a 5 (melhor avaliação).

Nos experimentos, foram geradas 4 partições para cada base de dados. Essas partições foram formadas por meio de permutações aleatórias de todo o conjunto de dados. Em outras palavras, para cada base de dados, foram realizadas 4 iterações de particionamento para a obtenção de conjuntos aleatórios de treinamento e teste. Em cada iteração, o conjunto total das avaliações \mathcal{R} foi dividido em 2 conjuntos mutuamente exclusivos \mathcal{R}_{train} e \mathcal{R}_{test} , contendo 90% e 10% das avaliações, respectivamente. O subconjunto \mathcal{R}_{train} é utilizado durante o processo de modelagem para a obtenção dos modelos de similaridade local e o subconjunto \mathcal{R}_{test} é utilizado durante o processo de predição para avaliar a performance do modelo.

4.4 Métricas

Dois métricas foram utilizadas para avaliar a performance do método proposto: MAE e RMSE. Ambas as métricas são frequentemente usadas para avaliar a precisão de modelos de recomendação em SRs (ADOMAVICIUS; TUZHILIN, 2005; JALILI et al., 2018). O MAE mede o desvio médio entre as preferências preditas e as preferências reais. Ou seja, o MAE consiste na soma da diferença absoluta entre as avaliações preditas e as avaliações reais dividida pelo número total de avaliações no conjunto de teste ($\#\mathcal{R}_{test}$), conforme descrito na Equação (4.6).

$$\text{MAE} = \frac{\sum_{r_{u_i o_k} \in \mathcal{R}_{test}} |\hat{r}_{u_i o_k} - r_{u_i o_k}|}{\#\mathcal{R}_{test}} \quad (4.6)$$

A métrica RMSE é uma variação do MAE. Contudo, diferentemente do MAE onde todos os erros possuem o mesmo peso, o RMSE penaliza erros mais significativos ao elevá-los ao quadrado antes de somá-los. Ou seja, o RMSE dá mais peso para erros maiores e menos peso para erros menores. O RMSE é definido na Equação (4.7).

$$\text{RMSE} = \sqrt{\frac{\sum_{r_{u_i o_k} \in \mathcal{R}_{test}} (\hat{r}_{u_i o_k} - r_{u_i o_k})^2}{\#\mathcal{R}_{test}}} \quad (4.7)$$

Valores menores de MAE e RMSE correspondem a maior precisão na predição. Dada uma lista de recomendação, o MAE e o RMSE desconsideram o ranking dos objetos na lista de recomendação. Ou seja, diferentemente das métricas de classificação, a ordem em que os objetos aparecem na lista de recomendação não impacta o resultado dessas métricas de precisão.

4.5 Conclusões

Esse capítulo apresentou uma visão geral e a descrição da metodologia utilizada para abordar os dois problemas descritos nesse trabalho:

- Captura dos dados de interação dos usuários de maneira organizada e individualizada.
- Recomendação de conteúdo utilizando uma vizinhança relevante.

O primeiro problema normalmente não é abordado por trabalhos relacionados a sistemas de recomendação. Diante da dificuldade de se identificar os usuários em ambientes de TV, normalmente os trabalhos focam no desenvolvimento de técnicas de recomendação para grupos de usuários. Alguns trabalhos apresentam recomendações individualizadas, porém recorrem ao uso de dispositivos estranhos ao ambiente de TV, o que acaba impactando fortemente na experiência do usuário. Esse trabalho propõe a utilização de dispositivos de tela secundária para capturar as interações e identificar os usuários de maneira que se possa realizar a personalização de conteúdo de maneira individualizada.

O segundo problema destaca a utilização de uma única medida de similaridade para todas as situações. Nesse trabalho é argumentado que uma maneira mais natural de se calcular a similaridade seria considerando grupos de objetos similares. Ao dar a mesma importância para todos os objetos, os objetos do grupo que representa o interesse primário do usuário tende a impactar fortemente no valor global da similaridade. Ao considerar os grupos de objetos relacionados, é possível obter um valor de similaridade para cada grupo de objetos. Dessa maneira, é possível obter vizinhanças de usuários que sejam relevantes para o grupo de objetos que melhor representa o objeto de interesse. As equações e algoritmos utilizados na formulação dos métodos são os mesmos empregados nos métodos existentes, como a função de normalização *z-score* e o Coeficiente de Correlação de Pearson. Porém, diferentemente das abordagens existentes que consideram todo o universo do conjunto de dados de treinamento na aplicação dos algoritmos, esse trabalho aplica os algoritmos considerando somente o conjunto de dados referente ao *cluster* de interesse, obtido a partir de um processo de clusterização da base de dados.

Esse capítulo descreve detalhadamente a metodologia utilizada para resolver esses dois problemas dentro do contexto dessa pesquisa utilizando telas secundárias e clusterização dos objetos utilizando modelos baseados em FC. Os conceitos apresentados aqui são a base para a implementação e para os resultados obtidos nesse trabalho. A proposta para a captura de dados foi implementada e os resultados alcançados com os cenários de uso são apresentados no próximo capítulo. A proposta para a recomendação de conteúdo

utilizando uma vizinhança relevante também foi implementada e os resultados numéricos obtidos a partir dessa implementação são apresentados no capítulo a seguir.

Capítulo 5

Resultados Experimentais

ESSE capítulo descreve os resultados experimentais obtidos a partir da implementação das propostas apresentadas no capítulo anterior. A Seção 5.1 apresenta os resultados obtidos em nível de implementação, detalhando pontos chaves que contribuem para a concretização da proposta de captura de dados individualizada (apresentada na Seção 4.1). A Seção 5.2 descreve os resultados numéricos em termos de precisão para a proposta de recomendação de conteúdo utilizando uma vizinhança relevante (Seção 4.2). Por fim, a Seção 5.3 descreve as conclusões obtidas a partir desses resultados experimentais.

5.1 Captura de Dados de Maneira Organizada e Individualizada

Essa seção descreve os detalhes principais relacionados à implementação do método de captura de dados utilizando dispositivos de tela secundária. Além disso, a seção também dá exemplos de cenários de interação que podem ser alcançados com o uso do protótipo experimental do método de captura proposto nesse trabalho.

5.1.1 Implementação do Método

O protótipo para a captura de dados de maneira organizada e individualizada foi implementado utilizando a linguagem de programação Java na plataforma Android. Embora o protótipo tenha sido desenvolvido em Java na plataforma Android, ele pode ser implementado em qualquer outra linguagem ou plataforma que seja compatível com os conceitos adotados nesse trabalho em nível de implementação. Dispositivos móveis

baseados em Android foram utilizados no ATV como dispositivos de tela secundária e no terminal de TV foi utilizada a plataforma AndroidTV, também baseada no sistema operacional Android.

A implementação do método de captura pode ser alcançada por meio da implementação de uma API de serviços web que são mapeados para cada tipo de interação. As seções a seguir descrevem como é realizado o mapeamento das interações para requisições HTTP, como a API de serviços é projetada e como é realizado o consumo dos serviços web.

5.1.1.1 Mapeamento das Interações para Requisições HTTP

Os eventos gerados pelos usuários em aplicações podem ser interceptados por métodos de *callback*. Assim, todas as vezes em que um evento particular ocorrer, um método (ou função em nível de implementação) é executado para fazer o tratamento desse evento. Tipicamente, um método nomeado como *onClick* é o *callback* executado quando um usuário toca um botão na tela de um dispositivo móvel. Nesse trabalho, foi criado um mecanismo que permite a execução de métodos de tratamento das interações sempre que um métodos de *callback* for executado. Para que o funcionamento desse mecanismo seja possível, primeiramente é necessária a definição de uma interface que contenha os métodos que devem ser executados juntamente com os *callbacks*.

```
public interface InteractionsProxy {  
  
    @InteractionMapping(uri="/channels/change", method=RequestMethod.POST)  
    public void changeChannel(Map<String, Object> data);  
  
    @InteractionMapping(uri="/programs/evaluate", method=RequestMethod.POST)  
    public void evaluateProgram(Map<String, Object> data);  
  
}
```

Figura 9: Interface com os métodos que são chamados para cada tipo de interação. A imagem também apresenta as anotações com o mapeamento da interação para a URI e para o método da requisição HTTP que deve ser feita para enviar os dados da interação ao RDU.

A Figura 9 apresenta um exemplo da declaração de uma interface customizada (*InteractionsProxy*) que contém a assinatura dos métodos para fazer a captura dos dados relacionados aos eventos de mudança de canal (*changeChannel*) e avaliação de programas (*evaluateProgram*). De maneira geral, a interface *InteractionsProxy* representa um *proxy* para todas as interações que devem ser capturadas. Esses métodos são responsáveis somente por capturar os dados de interação e automaticamente enviá-los ao RDU por meio do consumo da API de serviços. Em outras palavras, os métodos do

proxy de interações não possuem comportamento customizado para cada interação. O comportamento propriamente dito é implementado nos métodos de *callback*, que ativamente chamam os métodos do *proxy* passando os argumentos com os dados produzidos pela interação.

Os métodos que fazem a captura dos dados são mapeados para requisições HTTP com o objetivo de associar um evento de interação específico ao serviço que fará o armazenamento dos dados desse evento no RDU. Esse mapeamento foi realizado por meio de uma anotação¹ customizada chamada `@InteractionMapping`. Essa anotação possui dois campos: `uri`, para definir a URI da requisição (que também identifica o tipo de interação); e `method`, para definir o método HTTP a ser usado na requisição. A anotação precede o método utilizado para capturar os dados da interação. Cada método da interface `InteractionsProxy` também possui um argumento (chamado *data*) que consiste em pares chave/valor contendo os dados produzidos durante a interação.

Um mecanismo de *listener* foi criado para que todas as chamadas aos métodos do *proxy* de interação sejam interceptadas e tratadas em um objeto *handler* do tipo `InteractionsHandler`. Esse objeto invoca o método `invoke` e faz a leitura dos metadados contidos na anotação do método que foi chamado no *proxy*. Com base nos metadados contidos na anotação, o *handler* automaticamente encaminha os dados gerados pela interação para o serviço web adequado. A Figura 10 apresenta a estrutura básica da classe `InteractionsHandler`.

```
public class InteractionsHandler implements InvocationHandler {

    @Override
    public Object invoke(Object proxy, Method method, Object[] args) throws Throwable {
        InteractionMapping m = method.getAnnotation(InteractionMapping.class);
        if (m == null)
            return null;
        // requisição HTTP utilizando as informações contidas em
        // m.uri() e m.method()
    }
}
```

Figura 10: Implementação básica da classe `InteractionsHandler` que intercepta as chamadas aos métodos do *proxy* de interação.

O método `invoke`² recebe três argumentos: `proxy`, que é o objeto *proxy* (do tipo `InteractionsProxy`) que teve o método chamado; `method`, que é o método que representa

¹ Uma anotação (*annotation*) é um mecanismo sintático da linguagem Java para disponibilizar metadados no código fonte e associá-los a classes, métodos, atributos, parâmetros, etc. A anotação por si só não possui comportamento e precede a estrutura que deve ser anotada utilizando o símbolo `@` seguido do tipo da anotação e seus parâmetros.

² Esse método faz parte da API de introspecção da linguagem Java e foi sobrescrito para fazer o tratamento das chamadas aos métodos do *proxy* de interação `InteractionsProxy`.

a interação na interface `InteractionsProxy`; e `args`, que é um *array* com os argumentos passados durante a chamada do método do *proxy*. É importante notar que o método `invoke` é invocado para todas as chamadas de métodos do *proxy*, porém a chamada só é completamente executada caso o método chamado tenha sido anotado com a anotação `@InteractionMapping`.

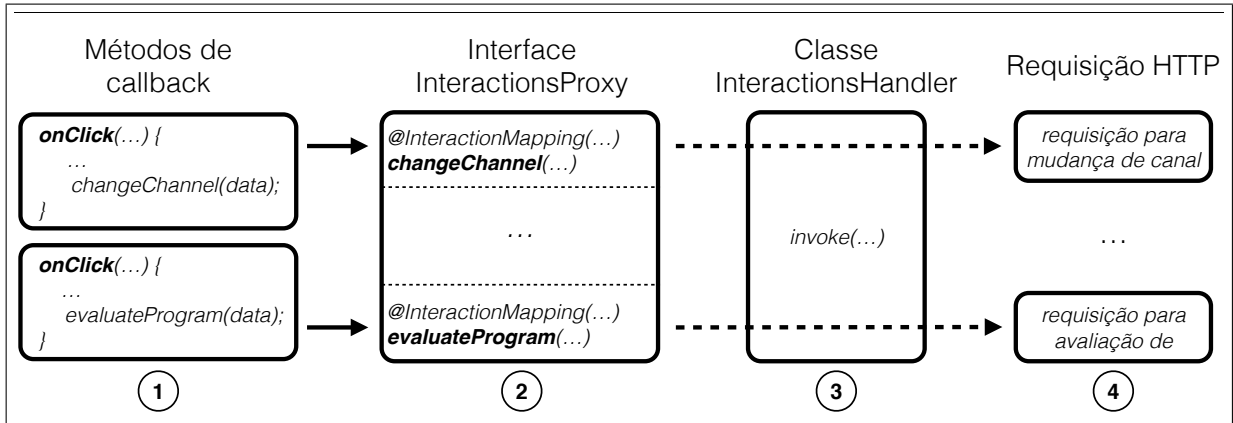


Figura 11: Passo a passo de como os dados da interação são capturados desde o momento do início da interação até o momento em que o serviço web é consumido com o envio dos dados ao RDU.

A Figura 11 apresenta as etapas desde o momento inicial em que uma interação é executada pelo usuário até o momento em que os dados gerados pela interação são enviados para o RDU. A sequência de etapas da esquerda para a direita é explicada a seguir:

1. O usuário interage com uma aplicação de tela secundária. Quando o evento de interesse acontece, um método de *callback* é executado processando a interação e coletando os dados gerados pelo evento. O método associado a essa interação na interface `InteractionsProxy` é chamado passando os dados da interação como argumento.
2. O método associado ao evento possui uma anotação `@InteractionMapping` na declaração da interface `InteractionsProxy`. Essa anotação contém os metadados necessários (URI e método HTTP) para automaticamente enviar os dados da interação para um serviço web.
3. Cada chamada aos métodos da interface `InteractionsProxy` é tratada por um objeto do tipo `InteractionsHandler`. O método `invoke` recebe os dados da interação e os metadados da anotação do método chamado (URI e método HTTP).
4. Sabendo o que precisa ser enviado (dados), para onde enviar (URI) e como enviar (método HTTP), o método `invoke` constrói uma requisição HTTP e consome o

serviço web associado à interação. Dessa maneira, os dados da interação são enviados ao RDU.

5.1.1.2 Design da API de Serviços Web e Implementação no RDU

Os dispositivos de tela secundária precisam se comunicar com o RDU para enviar os dados de interação dos usuários para o serviço de persistência. Para que isso seja possível o RDU precisa expor uma API de serviços web como uma interface no lado servidor para que as aplicações clientes (aplicações de tela secundária) possam consumir esses serviços por meio de requisições HTTP.

O *framework* Spring (SPRING, 2021) foi utilizado para desenvolver a API de serviços web nesse trabalho. Esse *framework* facilita o desenvolvimento do padrão de projeto chamado *front controller* em aplicações web. Todas as requisições são encaminhadas para um único ponto de entrada que faz o roteamento e despacha as requisições para os controladores apropriados. Esse mecanismo expõe uma estrutura escalável e flexível para uma API que precisa ser consumida por clientes de diferentes plataformas.

```
@RequestMapping(value="/programs/evaluate", method=RequestMethod.POST)
public String actionEvaluateProgram(EvaluationData data) {
    // implementação
}
```

Figura 12: Mapeamento de uma requisição HTTP de um serviço web para um método que faz o tratamento dos dados no RDU.

Os serviços web são configurados usando o anotação `@RequestMapping`. A Figura 12 demonstra como um serviço web que recebe os dados de interação de um evento de avaliação de programa pode ser configurado no lado servidor. Nessa figura, a anotação `@RequestMapping` mapeia a URI `/programs/evaluate` para um método POST. Essa anotação é adicionada logo acima do método `actionEvaluateProgram`. Dessa maneira, todas as requisições HTTP que forem enviadas para a URI `/programs/evaluate` utilizando o método POST serão tratadas pelo método `actionEvaluateProgram`. É importante mencionar que essa URI e esse método HTTP são os mesmos que foram definidos na anotação `@InteractionMapping` do método `evaluateProgram` da interface `InteractionsProxy` na aplicação de tela secundária no lado cliente. Esse casamento dos metadados é que faz a ligação entre as requisições que são enviadas do lado cliente ao lado servidor.

O parâmetro `data` do tipo `EvaluationData`, é um objeto que contém os dados que devem ser persistidos pelo serviço de persistência no RDU. Esses dados são os mesmos que foram enviados juntamente com a requisição automática realizada pela aplicação de tela secundária do lado cliente.

5.1.1.3 Consumo dos Serviços Web

Uma vez que exista uma API de serviços web bem definida como a interface de comunicação entre o ATV e o RDU, é possível consumir essa API, seja para receber ou para enviar dados. A URI base, que contém o domínio do serviço web, é utilizada como parte da API. Assim, a implementação das requisições HTTP na aplicação de tela secundária deve usar a URI inteira dos serviços web.

O efeito da requisição HTTP tipicamente depende do tipo de método que é usado na requisição: GET, POST, PUT ou DELETE. Por convenção, esses métodos se comportam como descrito a seguir:

- GET: as requisições usando o método GET devem somente recuperar os dados do servidor.
- POST: demandam que o servidor aceite um novo recurso identificado pela URI.
- PUT: modificam o dado identificado pela URI.
- DELETE: removem um recurso identificado pela URI.

No que diz respeito às requisições HTTP nesse trabalho, interessa somente o envio dos dados capturados para que sejam armazenados no RDU. Assim, somente o método POST é utilizado para enviar e persistir novos dados em um serviço remoto. De maneira geral, os dados de interação de um usuário específico não requerem alteração e nem remoção³ uma vez que compõem um histórico e podem ser utilizados com a intenção de identificar padrões de comportamento.

5.1.2 Protótipo Experimental

Essa seção apresenta a prova de conceito que foi desenvolvida, que consiste de dois cenários envolvendo as interações dos usuários: mudança de canal e avaliação de programas de TV. O protótipo da aplicação tela secundária (cliente) foi desenvolvido para a plataforma Android. O RDU, incluindo o serviço de persistência, foi desenvolvido no ambiente de computação na nuvem do *Google Cloud Platform* chamado AppEngine. Além disso, o serviço de *background* foi desenvolvido no terminal de TV baseado em Android para permitir a comunicação com os dispositivos de tela secundária.

Em ambos os cenários é requerido que o usuário esteja autenticado. Para isso, são utilizados serviços web que disponibilizam a autenticação do usuário por meio de seus

³ Com a exceção de casos previstos nas leis de proteção de dados que dispõem sobre o tratamento de dados, direito à privacidade, etc.

dispositivos de tela secundária. Esses serviços são particularmente importantes porque eles permitem a identificação dos usuários. Como resultado, todas as interações são associadas aos seus autores. Os serviços de autenticação possuem funcionalidades especiais e são descritos a seguir:

- `/login`: permite que o usuário faça o login em sua conta particular. Cada acesso à API de serviços web é interceptada e só é completada caso o usuário esteja logado em sua conta. Os campos que permitem o login são: *usuário* e *senha*. Esses campos são enviados em uma requisição do tipo POST. Se os dados de usuário e senha estiverem corretos, uma sessão é criada para o usuário.
- `/logout`: uma requisição do tipo POST para esse serviço permite que uma sessão HTTP criada no login seja invalidada. Uma vez que a sessão seja invalidada, um novo acesso ao sistema só será permitido com uma nova sessão criada por meio de login.

As seções a seguir descrevem os diferentes cenários implementados no protótipo experimental e as situações onde produtores de conteúdo e anunciantes podem tirar vantagem dos dados que são gerados pela interação do usuário.

5.1.2.1 Cenário 1: Mudança de Canal

Esse cenário consiste da captura de informações relacionadas à troca de canais no ATV. O método descrito neste trabalho pode ser usado para identificar padrões de migração entre canais de TV. Por exemplo, se um grande grupo de usuários diariamente abandona uma programação trocando para um canal diferente logo no início de um novo programa de TV, esse comportamento pode revelar um padrão de migração entre canais de TV por conta de um conteúdo desinteressante. Com o auxílio de técnicas de análise de dados, os produtores de conteúdo podem segmentar o perfil dos usuários que praticaram essa interação e trabalhar na produção de conteúdo para reter a audiência.

```
...
userId: "28",
uri: "/channels/change",
extras: {
  from: { channelId: "4", programId: "512" },
  to: { channelId: "13", programId: "256" }
},
...
```

Figura 13: Exemplo de dados de interação capturados para um evento de troca de canais.

A Figura 13 apresenta os dados obtidos após um evento de troca de canal. Nesse exemplo, um usuário identificado pelo ID “28” executou um evento de troca de canal mudando do canal “10” (onde o programa de TV no momento do evento foi identificado pelo ID “512”) para o canal “14” (onde o programa de TV no momento do evento foi identificado pelo ID “513”). Conforme explicado ao longo do capítulo, esses dados são enviados ao RDU por meio de uma requisição HTTP para a URI `/channels/change` utilizando o método POST. Anunciantes podem utilizar esses dados associados a dados demográficos para direcionar o conteúdo dos anúncios e aumentar a aderência aos comerciais de TV.

5.1.2.2 Cenário 2: Avaliação de um Programa

```
...
userId: "28",
uri: "/programs/evaluate",
extras: {
  programId: "512",
  rating: "5"
},
...
```

Figura 14: Exemplo dos dados de interação capturados para um evento de avaliação de programa de TV.

Nesse cenário, o usuário pode utilizar o aplicativo de tela secundária para avaliar programas de TV que estão sendo exibidos na tela principal. As avaliações variam de 1 a 5, onde 1 representa baixo interesse e 5 representa alto interesse pelo conteúdo. A Figura 14 apresenta os dados de interação para um evento de avaliação de programa de TV. Nessa interação, somente os campos `programId` e `rating` são capturados. O usuário associado a essa interação já está identificado pela propriedade `userId`. A URI que é usada para enviar os dados de interação para o RDU nesse cenário é `/programs/evaluate`.

Após um usuário em particular avaliar alguns programas, um SR pode analisar os dados e esse usuário pode começar a receber recomendações com base no seu histórico de avaliações. Outras estratégias de recomendação podem ser adotadas baseadas em segmentação do perfil dos usuários, horários, localização, etc.

5.2 Recomendação de Conteúdo Utilizando uma Vizinhança Relevante

Nessa seção, são apresentados o projeto experimental e os resultados obtidos a partir dos experimentos utilizando 3 bases de dados de referência para a literatura de

SRs: MovieLens 100k, MovieLens 1M e Netflix. Todo o código que foi desenvolvido durante esses experimentos foi escrito em Python 3.7 e executado no ambiente Anaconda⁴. Todos os experimentos foram executados na mesma máquina (processador: Intel®Core™ i5-5200U CPU @ 2.20 GHz x 2, RAM: 8 GB de RAM, OS: Microsoft Windows 10 Home Edition 64-bit).

5.2.1 Resultados e Discussão

Nesse trabalho o modelo de recomendação proposto (modelo CBLS) foi avaliados em dois casos diferentes. No primeiro caso, a performance da precisão do modelo CBLS foi comparada com diferentes métricas de similaridade tradicionais e do estado da arte em termos de tamanho de vizinhança. No segundo caso, o MAE e RMSE do modelo proposto foram avaliados em termos do número de *clusters* e do tamanho da vizinhança. Além disso, essa seção também apresenta uma breve discussão sobre a complexidade de tempo do método proposto.

5.2.1.1 Métodos Usados para Comparação

Os métodos de predição utilizados para comparação com CBLS incluem métricas de similaridade como o PCC, WPCC, SPCC e similaridade do cosseno. A Tabela 5 apresenta um resumo das técnicas utilizadas na comparação dos resultados experimentais.

Tabela 5: Métricas de similaridade utilizadas na comparação com o método proposto nos resultados experimentais.

	Escopo da Similaridade	Métricas de Similaridade	Algoritmo de Clusterização
SG_COS	Similaridade Global	Cosseno	-
SG_PCC	Similaridade Global	PCC	-
SG_WPCC	Similaridade Global	WPCC	-
SG_SPCC	Similaridade Global	SPCC	-
SL_KM	Similaridade Local	CBLS	<i>K-Means</i>
SL_AP	Similaridade Local	CBLS	<i>Affinity Propagation</i>

As métricas de similaridade tradicionais e do estado da arte são utilizadas para construir os modelos de similaridade dos métodos de recomendação que se baseiam em similaridade global (SG_COS, SG_PCC, SG_WPCC e SG_SPCC). A Equação (2.5) apresentada na Seção 2.4.1.1 (Capítulo 2) descreve a função de agregação que é utilizada para prever as preferências com esses métodos. Uma vez que a literatura de FC amplamente utiliza os métodos baseados em similaridade global, esses métodos são utilizados como métodos de referência para comparação com o método proposto em termos de precisão (MAE e RMSE).

⁴ Anaconda é uma plataforma de código aberto para o desenvolvimento de computação científica utilizando as linguagens Python e R.

Os métodos de predição que dependem de métricas de similaridade local (SL_AP e SL_KM) são baseados no método proposto nesse trabalho. A Equação (4.4) apresentada na Seção 4.2.3.2 (Capítulo 4) descreve a função de agregação que considera a similaridade local com a função de normalização z-score (sensível ao *cluster*) e o processo de alocação de recursos para prever as preferências de um usuário por um objeto dentro de um *cluster*. Nos experimentos com o modelo de similaridade CBLs foram utilizados dois algoritmos de clusterização (*affinity propagation* e *K-Means*) para demonstrar a flexibilidade do método proposto. Por conta da esparsidade, o número de *clusters* nos experimentos é definido como 2. Experimentos adicionais apresentados nesse capítulo mostram que um aumento no número de *clusters* pode reduzir a precisão das predições.

5.2.1.2 Efeitos do Tamanho da Vizinhaça

Nessa seção são investigados os efeitos do tamanho da vizinhaça nos métodos de recomendação. Para uma comparação mais justa, todos os modelos utilizam a função de normalização z-score para normalizar as preferências existentes. Essa função de normalização é utilizada na função de agregação para prever as preferências em todos os métodos de recomendação. Porém, diferentemente dos métodos baseados em similaridade global onde a normalização se baseia em todos os dados da base de treinamento, nos métodos baseados em similaridade local a função z-score se baseia somente nas preferências do *cluster* de interesse.

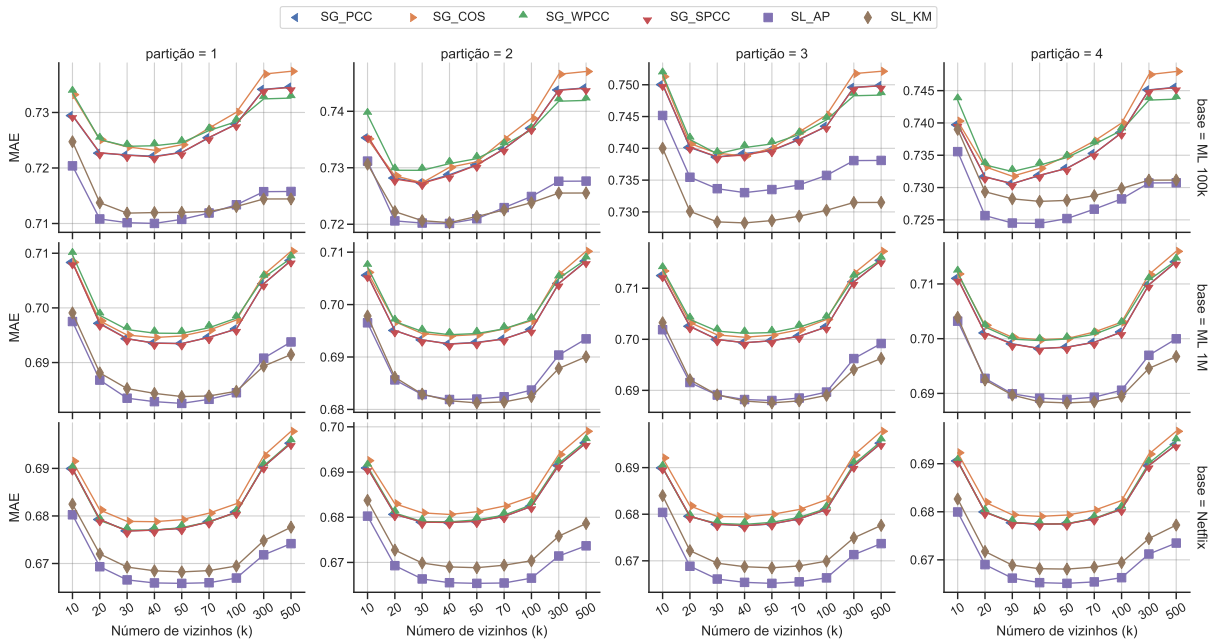


Figura 15: Curva do MAE do conjunto de dados \mathcal{R}_{test} para as bases ML 100k, ML 1M e Netflix.

As figuras 15 e 16, respectivamente, apresentam as curvas MAE e RMSE para

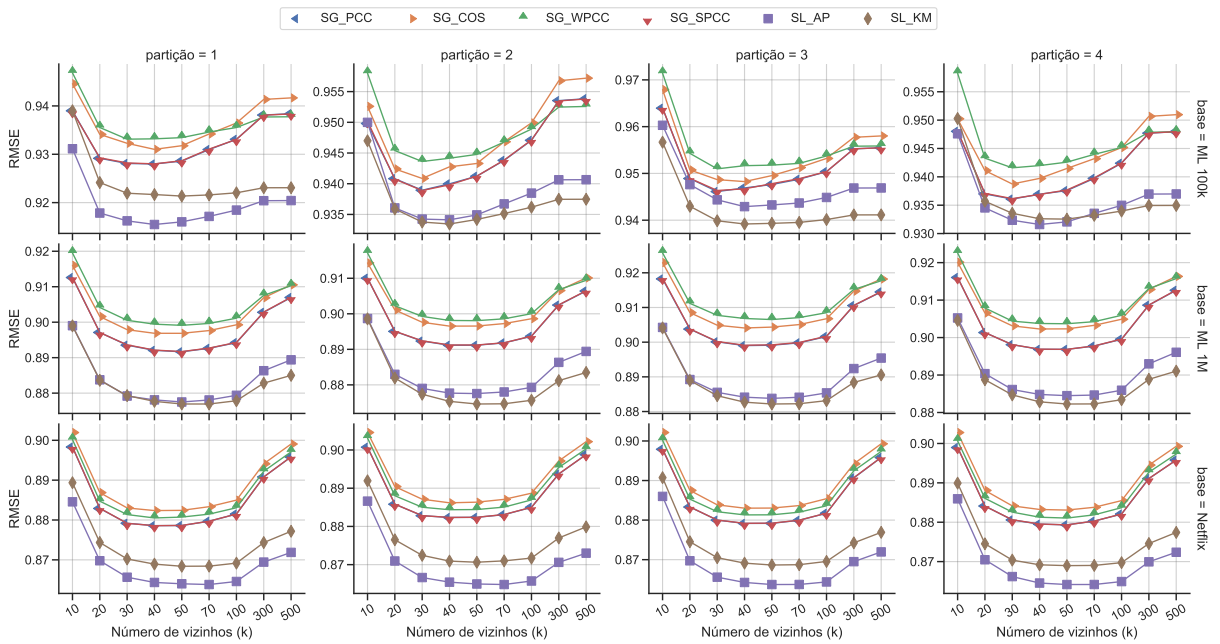


Figura 16: Curva do RMSE do conjunto de dados \mathcal{R}_{test} para as bases ML 100k, ML 1M e Netflix.

cada partição do conjunto \mathcal{R}_{test} . A variação no valor de k , que representa o tamanho utilizado como referência na função de agregação, impacta nos valores dos erros de predição. As curvas de erro dos métodos baseados em similaridade global apresentam um comportamento semelhante entre si conforme o valor de k varia. É importante ressaltar que as curvas de erro SG_PCC e SG_SPCC estão muito próximas e a diferença é quase imperceptível nas figuras. Por exemplo, na partição 1 da base de dados ML 1M, para k igual a 70, o valor do MAE é de 0,694623 para o método SG_PCC, enquanto que para o método SG_SPCC, o valor do MAE é de 0.694608.

Quanto menor forem os valores do MAE e do RMSE, melhor é a precisão da predição. Diante disso, é possível observar que os métodos de predição que são baseados em similaridade local (CBLs) superam os métodos utilizados como referência (baseados em similaridade global) em termos de precisão de predição para todos os valores de k .

De maneira geral, na base de dados ML 100k, os valores de MAE diminuem com o aumento do número de vizinhos de 10 para 40. Entretanto, para valores de $k > 40$, é possível observar um aumento nos valores de MAE. Os experimentos na base ML 1M mostram um comportamento similar das curvas de erro: o MAE diminui com o aumento da quantidade de vizinhos de 10 para 50, aumentando somente para $k > 50$. As curvas de erro para a base de dados Netflix mostram também um comportamento similar ao apresentado na base de dados ML 1M.

Um número maior de vizinhos impacta negativamente as predições realizadas para objetos populares. Uma possível razão para esse comportamento é que objetos mais popu-

lares têm potencial para incluir vizinhos de baixa qualidade no processo de predição, uma vez que esses objetos evidentemente também alcançam mais usuários. Para os métodos de predição baseados em similaridade global, a curva do MAE cresce rapidamente com o aumento da vizinhança k , enquanto que para os métodos baseados em similaridade local (CBLS), o crescimento da curva de erro é suavizado. Com isso, pode-se concluir que a abordagem de similaridade local descrita nesse trabalho suaviza o impacto da introdução de mais vizinhos no processo de predição para um dado objeto.

As técnicas baseadas em similaridade local (CBLS) levam em conta as correlações que podem existir entre usuários que observaram um grupo específico de objetos relacionados. Dessa maneira, a seleção da vizinhança só leva em consideração a similaridade obtida a partir dos objetos que formam o *cluster* do objeto de interesse. Diferentemente, as abordagens baseadas em similaridade global só levam em consideração a correlação obtida a partir de toda a base de treinamento. Ou seja, os métodos de similaridade global formam uma vizinhança com usuários similares, mas não dão atenção especial às correlações obtidas a partir de objetos relacionados que também são similares ao objeto de interesse.

As curvas do RMSE mostram um comportamento similar ao observado nas curvas do MAE. Entretanto, como o RMSE penaliza de forma mais severa os erros maiores, nós podemos observar que os erros mais significativos tipicamente ocorrem para tamanhos de vizinhança próximos de 10. Na base de dados ML 100k, todos os métodos comparados possuem performance semelhante para $k = 10$, o que significa que a magnitude dos erros para esse tamanho de vizinhança é quase a mesma. Porém, para $k > 20$, os métodos baseados em CBLS claramente se sobressaem com os valores mais baixos de RMSE. Na base de dados ML 1M e Netflix, os métodos de similaridade local (SL_AP e SL_KM) superam os métodos baseados em similaridade global apresentando valores mais baixos de RMSE para todos os valores de k .

A Tabela 6 apresenta algumas das melhorias de performance alcançadas pela abordagem descrita nesse trabalho em relação aos métodos de referência. Essa tabela apresenta os valores de MAE e RMSE para a partição 1 da base de dados Netflix⁵. Os valores mais altos de MAE e RMSE dos métodos baseados em similaridade global foram comparados aos valores mais baixos de MAE e RMSE dos métodos de similaridade local. A última coluna apresenta a melhora percentual da performance que foi alcançada para cada tamanho de vizinhança. Esses resultados confirmam o que foi mostrado nas figuras 15 e 16.

⁵ As tabelas completas com os dados das demais partições e bases de dados serão apresentadas no Apêndice C.

Tabela 6: Melhorias de performance da abordagem proposta nesse trabalho em relação os métodos de referência considerando as métricas MAE e o RMSE para a partição 1 da base de dados Netflix. Os valores em negrito em cada linha dos métodos de similaridade global indicam a pior performance, enquanto os valores em negrito em cada linha dos métodos baseados em similaridade local indicam a melhor performance.

k	Similaridade global												Similaridade local					
	SG_COS		SG_PCC		SG_SPPC		SG_WPCC		SL_AP		SL_KM		Melhoria(%)					
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE				
10	0,692	0,902	0,690	0,898	0,690	0,898	0,690	0,900	0,680	0,885	0,682	0,889	1,629	1,928				
20	0,681	0,887	0,679	0,883	0,679	0,883	0,679	0,885	0,669	0,870	0,672	0,874	1,752	1,928				
30	0,679	0,883	0,677	0,879	0,677	0,879	0,677	0,881	0,667	0,866	0,669	0,870	1,812	1,968				
40	0,679	0,882	0,677	0,879	0,677	0,879	0,677	0,881	0,666	0,864	0,668	0,869	1,897	2,044				
50	0,679	0,882	0,677	0,879	0,677	0,879	0,678	0,881	0,666	0,864	0,668	0,868	1,971	2,088				
70	0,681	0,883	0,679	0,880	0,679	0,880	0,679	0,881	0,666	0,864	0,669	0,868	2,150	2,211				
100	0,683	0,885	0,681	0,881	0,681	0,882	0,681	0,883	0,667	0,865	0,669	0,869	2,290	2,307				
300	0,693	0,894	0,690	0,891	0,690	0,891	0,691	0,892	0,672	0,869	0,675	0,874	3,010	2,759				
500	0,698	0,899	0,695	0,896	0,695	0,896	0,696	0,897	0,674	0,872	0,678	0,877	3,384	3,025				

5.2.1.3 Efeitos do Número de Vizinhos e *Clusters*

Nesse experimento, o método de predição SL_KM foi utilizado para investigar a performance dos modelos CBLS para um número variável de *clusters*. Esse método foi escolhido porque no algoritmo de clusterização *K-Means*, o número de *clusters* pode ser definido diretamente como parâmetro do algoritmo⁶. Os experimentos consistem na variação do número de *clusters* de 2 a 7, e do número de vizinhos de 10 a 500.

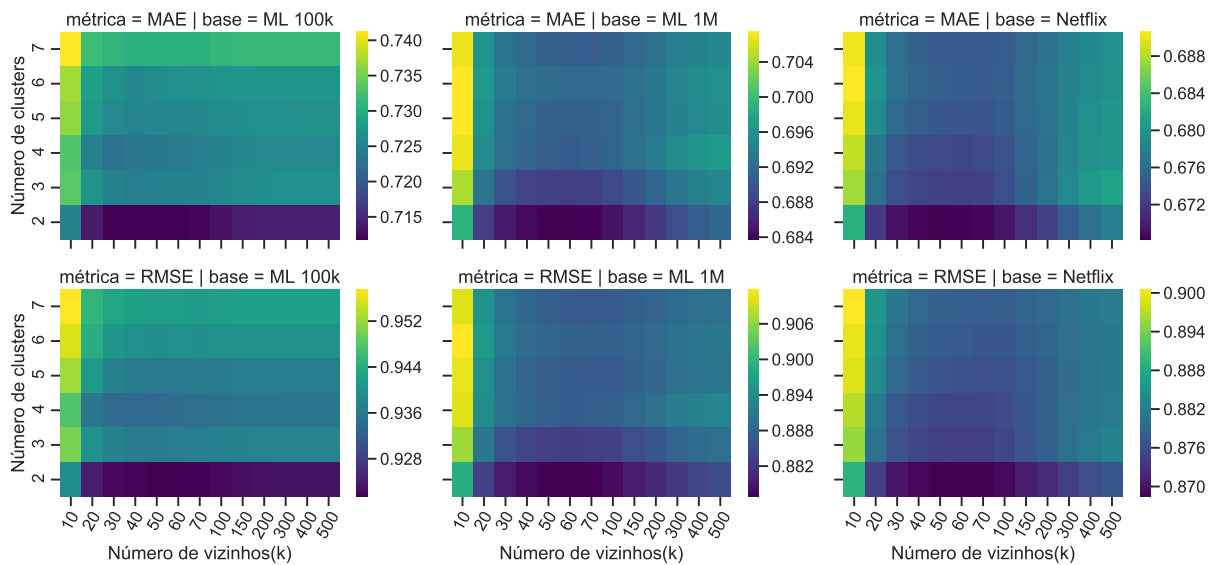


Figura 17: Mapa de calor dos valores de MAE e RMSE para a partição 1 considerando um número variável de vizinhos e um número variável de *clusters* para o método SL_KM.

A Figura 17 apresenta um mapa de calor dos valores de MAE e RMSE para a partição 1 das bases de dados ML 100k, ML 1M e Netflix utilizando o método de predição SL_KM⁷. Uma consideração a ser feita é que quando o número de *clusters* aumenta, o número de avaliações por *cluster* diminui. Em outras palavras, um número maior de *clusters* diminui o número de objetos por *cluster*, o que conseqüentemente reduz a quantidade de avaliações que são necessárias para computar o modelo de similaridade local. Uma consequência direta disso é que a baixa quantidade de avaliações por *cluster* acaba levando à degradação na performance do modelo, uma vez que alguns usuários e objetos passam a ser afetados pelo problema de *cold start*. Ou seja, dentro do *cluster* não existem avaliações o suficiente para calcular uma similaridade de qualidade com outros usuários.

Quando os efeitos do número de vizinhos para um dado número de *clusters* é avaliado, observa-se o mesmo comportamento descrito na Seção 5.2.1.2. O modelo sofre

⁶ No caso do método de predição SL_AP, que utiliza o algoritmo *affinity propagation*, a quantidade de *clusters* é desconhecida e depende do ajuste dos parâmetros *preferência* e *amortecimento*.

⁷ O Apêndice C apresenta as imagens do mapa de calor para todas as partições considerando cada base de dados.

degradação na performance quando o número de vizinhos varia abaixo de 30 ou acima de 70.

Com relação ao efeito da quantidade de *clusters* para qualquer tamanho de vizinhança, o modelo de similaridade local apresenta a melhor performance para uma quantidade menor de *clusters*. Entretanto, nós podemos observar que os valores de MAE e o RMSE tendem à estabilidade conforme a quantidade de *clusters* aumenta em bases de dados maiores como a ML 1M e Netflix. Uma vez que a base de dados ML 100k é relativamente pequena, um aumento no número de *clusters* acentua o efeito da esparsidade. Ou seja, o número de avaliações por *cluster* diminui resultando em um rápido aumento dos valores de MAE e RMSE.

5.2.1.4 Análise da Complexidade de Tempo

O alto custo do cálculo da similaridade entre usuários é uma típica desvantagem de abordagens de FC baseadas em memória devido ao tempo de execução quadrático. Apesar disso, o cálculo da similaridade é uma das etapas mais críticas na construção de modelos que visam a melhora da precisão das predições. Sejam m , n e k o número de usuários, o número de objetos e o número de vizinhos, respectivamente, a complexidade das abordagens tradicionais de FC baseadas em memória é $O(m^2n + mnk)$, onde o primeiro termo representa o cálculo da similaridade entre os usuários e o segundo termo representa o cálculo das predições.

A análise da complexidade do método proposto depende da complexidade do algoritmo de clusterização utilizado durante o processo de treinamento. Uma vez que o método de similaridade local é desacoplado dos detalhes da implementação do algoritmo de clusterização, pode-se utilizar qualquer algoritmo que seja adequado ao domínio do problema. Ignorando a complexidade do processo de clusterização, o método proposto não aumenta a complexidade computacional quando comparado com abordagens tradicionais de FC.

Uma vez que cálculos adicionais são incluídos para construir o modelo de similaridade local e para prever as avaliações não observadas, é importante notar que as técnicas baseadas em CBLS sofrem um aumento nos tempos de execução para treinamento e teste quando comparadas com abordagens tradicionais. A Tabela 7 apresenta evidências que confirmam esse aumento no tempo de execução. Os tempos de execução foram medidos para cada partição da base de dados e o valor de cada célula da tabela é a média desses tempos de execução em segundos. As colunas TC e TT representam o tempo de clusterização e o tempo de treinamento, respectivamente. O tempo de clusterização é apresentado somente para os métodos baseados em similaridade local uma vez que os outros métodos não são baseados em clusterização. O tempo de treinamento representa o tempo total necessário para a construção do modelo (que inclui o tempo de clusteri-

zação quando aplicável). Ao observar os tempos de execução nos modelos baseados em similaridade local, é possível verificar que o tempo de clusterização representa uma parte considerável do tempo total de treinamento. Com relação ao tempo de teste, é possível verificar um aumento que acompanha o aumento no tamanho da vizinhança.

5.3 Conclusões

Essa seção apresentou os principais resultados experimentais e resultados numéricos alcançados ao longo do desenvolvimento desse trabalho. A Seção 5.1 descreveu como foi realizada a implementação dos principais componentes do método de captura das interações dos usuários de maneira organizada e individualizada. O método descrito permite a identificação dos usuários por meio de serviços de autenticação.

Os mecanismos de implementação descritos permitem o encadeamento dos eventos de interação desde o gatilho que aciona o evento até o envio dos dados gerados para o RDU. A implementação desses mecanismos permite o mapeamento das interações do usuário para requisições HTTP utilizando a API de reflexão da linguagem Java (*Reflection API*). Essa API permitiu o desenvolvimento da estrutura responsável pelo envio automático dos dados da interação para o RDU, de acordo com os metadados declarados no código fonte.

Ainda na Seção 5.1, foram descritos dois cenários compatíveis com o protótipo desenvolvido. Esses cenários envolvem a captura de dados para eventos de mudança de canal e para eventos de avaliação de conteúdo (programas de TV). Além disso, na descrição desses cenários foram apresentadas situações em que os dados obtidos poderiam ser de grande valia para profissionais da área de produção de conteúdo e publicidade no domínio da TVDI.

Na Seção 5.2, foram apresentados os resultados numéricos dos experimentos realizados com três bases de dados comparando abordagens tradicionais de SRs e o método proposto nesse trabalho (similaridade local com o método CBLS). Os experimentos avaliaram a performance do método proposto em termos de acurácia das recomendações obtidas. Nesses experimentos, a acurácia das recomendações foi avaliada em termos do tamanho da vizinhança e da quantidade de *clusters*. Os resultados obtidos a partir dos experimentos mostram que o método proposto possui uma precisão superior aos métodos utilizados como referência para comparação.

Tabela 7: Tempo médio de execução dos métodos usados nas comparações. Os valores de cada linha são apresentados em segundos. TC e TT são o Tempo de Clusterização e o Tempo de Treinamento, respectivamente.

Base de Dados	Método	TC (s)	TT (s)	Tempo de Teste (s)								
				10	20	30	40	50	70	100	300	500
ML 100k	SG_COS	-	1,30	2,16	2,19	2,20	2,42	2,45	2,38	2,46	2,71	2,76
	SG_PCC	-	1,28	2,14	2,22	2,20	2,29	2,31	2,37	2,45	2,70	2,82
	SG_SPCC	-	1,61	1,90	1,91	1,94	2,05	2,05	2,14	2,26	2,51	2,56
	SG_WPCC	-	1,73	2,73	2,11	2,14	2,23	2,39	2,37	2,43	2,92	2,82
	SL_AP	10,85	3,78	3,94	4,07	4,18	4,33	4,73	4,94	6,60	6,17	5,96
	SL_KM	1,18	3,36	3,78	4,00	4,31	4,41	5,28	4,92	5,80	48,32	48,00
ML 1M	SG_COS	-	10,56	31,82	33,56	32,18	33,18	33,60	35,34	35,42	42,79	48,32
	SG_PCC	-	10,63	30,00	33,90	33,10	32,30	33,78	34,28	35,62	42,84	48,00
	SG_SPCC	-	24,26	34,65	35,30	35,83	36,05	36,54	37,58	39,36	47,36	53,20
	SG_WPCC	-	23,71	34,54	34,13	35,32	36,08	35,71	36,71	39,63	46,71	53,40
	SL_AP	20,31	47,73	36,21	37,40	39,15	41,43	43,63	48,27	71,48	88,98	
	SL_KM	15,78	38,19	37,95	39,32	41,29	43,55	46,38	50,71	75,45	92,78	
Netflix	SG_COS	-	16,85	50,41	50,92	52,38	52,68	74,35	63,03	60,68	72,67	93,94
	SG_PCC	-	17,18	50,34	50,83	52,13	52,74	53,53	54,87	57,17	67,50	76,63
	SG_SPCC	-	38,49	49,20	49,39	50,18	51,49	51,83	53,59	55,12	65,54	73,40
	SG_WPCC	-	38,81	50,91	49,67	50,43	51,41	52,47	53,19	55,07	66,23	73,35
	SL_AP	77,90	114,88	64,27	66,32	69,65	71,00	76,22	81,39	116,31	135,67	
	SL_KM	22,93	57,42	63,96	66,31	68,16	70,12	73,99	79,80	107,35	128,44	

Capítulo 6

Considerações Finais

Os algoritmos de SRs surgem como um serviço cada vez mais necessário aos grandes provedores de mídia e conteúdo. Diante da realidade da sobrecarga de informação, os SRs ajudam os usuários a rapidamente encontrar conteúdo que seja apropriado ao seu interesse particular. Para que os algoritmos de SRs funcionem de maneira precisa, o acesso ao histórico individual das atividades de cada usuário do sistema é imprescindível. Porém, embora o acesso a esses dados individuais seja uma realidade em muitos serviços pessoais baseados em web como *Amazon*, *YouTube*, *Spotify* e muitos mecanismos de busca, isso ainda é um desafio em ambientes de TV por conta da sua natureza coletiva e dos limitados mecanismos de interação.

Para criar um ambiente mais viável para a captura do histórico de interações de maneira individualizada em ambientes de TV, esse trabalho propôs a utilização de dispositivos móveis como um meio facilitador para a captura de dados. A ubiquidade dos dispositivos móveis já é uma realidade em muitos países e a inserção desses dispositivos em ambientes de TV surge como uma tendência cada vez mais forte e natural. No ambiente de TV, os dispositivos móveis são conhecidos como telas secundárias, uma vez que são utilizados como dispositivos auxiliares para interação e visualização de conteúdo. Algumas das vantagens da utilização de telas secundárias incluem a aceitação desses dispositivos no ambiente de TV, seu caráter pessoal e a sua capacidade computacional.

Por serem considerados de uso pessoal, os dispositivos móveis (telas secundárias) foram utilizados como um mecanismo para identificar os usuários. As interações são capturadas quando os usuários interagem com as aplicações de tela secundária. Para possibilitar a captura das interações individuais foi desenvolvida uma arquitetura que possibilita a autenticação dos usuários, a captura dos dados de interação e o envio automático desses

dados para um repositório remoto. A viabilidade técnica dessa arquitetura foi demonstrada com o desenvolvimento de um protótipo que inclui dois cenários de aplicação: troca de canal e avaliação de programas.

A principal vantagem dessa abordagem é que ela não é limitada à uma linguagem de programação, plataforma de TV ou sistema operacional específico. A base tecnológica para aquisição de dados no ambiente de TV e para o envio desses dados para um repositório remoto depende de protocolos de comunicação (que normalmente são disponibilizados por meio de bibliotecas) e tecnologias que são bem conhecidas e estabelecidas na indústria (JSON, HTTP e serviços web RESTful). Assim, é possível utilizar diferentes linguagens de programação e *frameworks* de desenvolvimento para executar requisições HTTP, o que pode expandir a aplicabilidade deste trabalho para diferentes plataformas de dispositivos móveis e padrões de TVDI, como o brasileiro ISDB-TB e o japonês ISDB-T que suportam o uso de dispositivos móveis como parte da interação.

Uma vez que seja possível individualizar os dados, os SRs podem fazer uso desses dados e se destacam como uma área promissora de aplicação. Assim, esse trabalho também descreve uma pesquisa na área de SRs que trata da similaridade local obtida a partir da análise de grupos de objetos relacionados, chamada de CBLs. Essa pesquisa abordou a utilização de técnicas de clusterização, normalização das preferências e a utilização de informações estruturais para melhorar a precisão das recomendações em sistemas de FC baseados em vizinhança. A pesquisa considerou que usuários que avaliam grupos de objetos relacionados podem compartilhar correlações que não são propriamente capturadas por modelos de similaridade global.

A abordagem de recomendação proposta identifica a correlação existente entre usuários que compartilham o interesse pelo mesmo grupo de objetos. Uma vez que podem existir múltiplos grupos de objetos relacionados, é possível identificar também múltiplas correlações entre os usuários ao analisar esses grupos de objetos. Como resultado, para um objeto de interesse e para um usuário ativo, é possível identificar uma vizinhança que seja mais relevante e adequada para o processo de predição ao considerar o grupo de objetos relacionados a que esse objeto de interesse pertence.

Os resultados experimentais foram avaliados em três bases de dados amplamente adotadas na literatura (ML 100k, ML 1M, e Netflix). Os resultados mostram que a abordagem de recomendação proposta é competitiva e supera os algoritmos tradicionais de FC em termos de precisão das predições. A abordagem proposta também é comparada a outras métricas de similaridade do estado da arte. Os resultados das predições foram avaliados com as métricas MAE e RMSE de acordo com o tamanho da vizinhança e a quantidade de *clusters*. Os experimentos mostram que o método proposto: a) melhora a precisão das recomendações, e b) alivia o impacto de se introduzir vizinhos de baixa qualidade que normalmente são incluídos na predição de objetos populares.

6.1 Limitações

O método proposto para a captura de dados tem o objetivo de individualizar as interações dos usuários, porém uma possível limitação do método acontece por conta da natureza intrinsecamente coletiva do ambiente de TV. Embora as interações capturadas sejam estritamente individuais, algumas interações podem ser resultado de decisões coletivas. Por exemplo, quando um grupo de usuários toma a decisão de assistir a um determinado programa de TV, a ação de fazer a troca de canal partirá de um único usuário. Os dados do evento capturado serão individualizados, mas por serem resultado de uma decisão coletiva, podem não expressar as preferências particulares do usuário que executou o evento de interação. Como resultado, esse tipo de comportamento pode comprometer a qualidade dos algoritmos que precisam inferir comportamentos individuais dos usuários.

No caso do método de recomendação proposto, algumas limitações em potencial incluem o consumo de memória e o tempo de execução. Uma vez que essa proposta de recomendação requer um modelo de similaridade para cada *cluster*, ela requer mais memória para armazenar os modelos. Um elevado número de *clusters* representa um desafio em cenários do mundo real onde o número de usuários e objetos é grande. Com respeito ao tempo de execução, embora o método proposto seja relativamente rápido para treinar um modelo de predição em comparação às abordagens tradicionais de FC, ele requer mais passos computacionais para predizer as avaliações. Isso acaba por aumentar o tempo de execução durante os experimentos com os conjuntos de teste.

6.2 Contribuições Alcançadas

As seguintes contribuições foram alcançadas ao longo do desenvolvimento das pesquisas descritas nesse trabalho:

- Construção de um protótipo de um modelo de interação baseado em telas secundárias que permite a coleta de dados de interação dos usuários de maneira organizada e individualizada em ambientes de TVDI.
- Criação de dois cenários de utilização do protótipo construído para demonstrar a sua aplicabilidade na coleta de dados para um ambiente de TV.
- Criação de um modelo de recomendação de conteúdo baseado em similaridade local que permite a escolha de uma vizinhança mais apropriada de acordo com os objetos avaliados em comum dentro de um *cluster* de objetos semelhantes.

- Avaliação objetiva do desempenho do modelo de recomendação em termos de precisão utilizando bases de dados reconhecidamente relevantes na literatura de SRs (MovieLens 100k, MovieLens 1M e Netflix).
- Comparação dos resultados obtidos com o estado da arte em recomendação de conteúdo, de acordo com as métricas de precisão estudadas (MAE e RMSE) em termos de tamanho da vizinhança e quantidade de *clusters*.
- Publicação de 4 artigos científicos em conferências e revistas.

6.3 Publicações Científicas Alcançadas

Os trabalhos a seguir foram publicados ao longo do desenvolvimento desse trabalho:

- R. E. V. de S. Rosa; V. F. de Lucena Jr. “Using a Voting Classification Algorithm with Neural Networks and Collaborative Filtering in Movie Recommendations for Interactive Digital TV,” in Adjunct Proceedings of the 10th European Interactive TV Conference, 2012, pp. 224–228.
- R. E. V. de S. Rosa; L. C. Cordeiro; V. F. de Lucena Junior. “A secondary screen architecture to accurately capture viewers’ interactions in an iTV environment,” in 2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE), 2014, pp. 264–265.
- R. Erikson V. de S. Rosa; V. Ferreira de Lucena Jr. “Contextualizing and capturing individual user interactions in shared iTV environments,” *Multimed. Tools Appl.*, vol. 76, no. 6, pp. 8573–8595, Mar. 2017.
- R. E. Veras De Sena Rosa, F. A. S. Guimaraes, R. da S. Mendonca, and V. F. de Lucena, “Improving Prediction Accuracy in Neighborhood-Based Collaborative Filtering by Using Local Similarity,” *IEEE Access*, vol. 8, pp. 142795–142809, 2020.

6.4 Trabalhos Futuros

A implementação da pesquisa de captura de dados foi desenvolvida como um protótipo experimental para a avaliar a viabilidade técnica da proposta. Uma vez que essa pesquisa implica na identificação de pessoas e na possível disponibilização e tratamento de seus dados, um trabalho futuro importante seria adequar a implementação da arquitetura para aderir às leis de proteção de dados vigentes em caso de uso comercial da pesquisa.

Como trabalhos futuros para a técnica de similaridade local proposta neste trabalho, pode-se considerar um experimento detalhado utilizando outros métodos de clusterização. Uma vez que o método proposto desacopla o método de clusterização dos detalhes da implementação, ele abre espaço para a utilização de algoritmos de clusterização alternativos. Outra possibilidade de trabalhos futuros é otimizar o consumo de memória e o tempo de execução. Também seria interessante avaliar os efeitos do aumento do número de *clusters* utilizando métricas de classificação (*precision*, *recall* e *F1*), uma vez que um aumento no número de *clusters* pode levar à degradação na precisão da predição.

Referências

ABNT. *ABNT NBR 15603-3. Televisão digital terrestre - Multiplexação e Serviços de Informação (SI). Parte 3 - Sintaxes e Definições da Informação Estendida do SI (Versão corrigida 4:2020)*. Rio de Janeiro, 2007.

ABNT. *ABNT NBR 15606-2. Parte 2: Ginga-NCL para receptores fixos e móveis – Linguagem de aplicação XML para codificação de aplicações. Parte 2: Ginga-NCL para receptores fixos e móveis – Linguagem de aplicação XML para codificação de aplicações*. Rio de Janeiro, 2018.

ABNT. *ABNT NBR 15606-3. Televisão digital terrestre - Codificação de dados e especificações de transmissão para radiofusão digital. Parte 3: Especificação de transmissão de dados (Versão corrigida: 2019)*. Rio de Janeiro, 2018.

ABNT. *ABNT NBR 15603-2. Televisão digital terrestre - Multiplexação e Serviços de Informação (SI). Parte 2: Estrutura de dados e definições da informação básica de SI*. Rio de Janeiro, 2020.

ABREU, J.; ALMEIDA, P.; TELES, B. TV discovery & enjoy: a new approach to help users finding the right TV program to watch. In: *Proceedings of the 2014 ACM international conference on Interactive experiences for TV and online video - TVX '14*. New York, New York, USA: ACM Press, 2014. p. 63–70. ISBN 9781450328388.

ABREU, J.; ALMEIDA, P.; TELES, B.; REIS, M. Viewer behaviors and practices in the (new) television environment. In: *Proceedings of the 11th european conference on Interactive TV and video - EuroITV '13*. New York, New York, USA: ACM Press, 2013. p. 5–12. ISBN 9781450319515.

ADOMAVICIUS, G.; TUZHILIN, A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, v. 17, n. 6, p. 734–749, jun 2005. ISSN 1041-4347.

AINA, T.; YE, Z.; DAI, Z.; JIANGHUI, C. Field tests of two-way television audience measurement system. In: *2014 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*. Beijing, China: [s.n.], 2014. p. 1–6. ISBN 978-1-4799-1654-2.

ALAM, I.; KHUSRO, S.; KHAN, M. Factors Affecting the Performance of Recommender Systems in a Smart TV Environment. *Technologies*, v. 7, n. 2, p. 41, may 2019. ISSN 2227-7080.

- ALVAREZ, F.; MARTIN, C. A.; ALLIEZ, D.; ROC, P. T.; STECKEL, P.; MENENDEZ, J. M.; CISNEROS, G.; JONES, S. T. Audience Measurement Modeling for Convergent Broadcasting and IPTV Networks. *IEEE Transactions on Broadcasting*, v. 55, n. 2, p. 502–515, 2009. ISSN 00189316.
- AMATRIAIN, X.; JAIMES, A.; OLIVER, N.; PUJOL, J. M. Data mining methods for recommender systems. In: RICCI, F.; ROKACH, L.; SHAPIRA, B.; KANTOR, P. B. (Ed.). *Recommender Systems Handbook*. [S.l.]: Springer, 2011. cap. 2, p. 39–71.
- AN, Y.-H.; DONG, Q.; SUN, C.-J.; NIE, D.-C.; FU, Y. Diffusion-like recommendation with enhanced similarity of objects. *Physica A: Statistical Mechanics and its Applications*, Elsevier B.V., v. 461, p. 708–715, 2016. ISSN 03784371.
- ANDERSON, C. *The Long Tail: Why the Future Is Selling Less of More?* 1. ed. New York, NY, USA: Hyperion, 2006. ISBN 9781401384630.
- ARMBRUST, M.; STOICA, I.; ZAHARIA, M.; FOX, A.; GRIFFITH, R.; JOSEPH, A. D.; KATZ, R.; KONWINSKI, A.; LEE, G.; PATTERSON, D.; RABKIN, A. A view of cloud computing. *Communications of the ACM*, ACM, v. 53, n. 4, p. 50–58, abr. 2010. ISSN 00010782.
- AYUB, M.; GHAZANFAR, M. A.; MEHMOOD, Z.; SABA, T.; ALHARBEY, R.; MUNSHI, A. M.; ALRIGE, M. A. Modeling user rating preference behavior to improve the performance of the collaborative filtering based recommender systems. *PLoS ONE*, v. 14, n. 8, 2019. ISSN 19326203.
- BAMBINI, R.; CREMONESI, P.; TURRIN, R. Recommender Systems for Interactive TV. In: KOMPATSIARIS, Y.; MÉRIALDO, B.; LIAN, S. (Ed.). *TV Content Analysis: Techniques and Applications*. Boca Raton, FL: CRC Press, 2012, (Multimedia Computing, Communication and Intelligence). cap. 10, p. 277–307. ISBN 1439855609.
- BARRAGÁNS-MARTÍNEZ, A. B.; COSTA-MONTENEGRO, E.; BURGUILLO, J. C.; REY-LÓPEZ, M.; MIKIC-FONTE, F. a.; PELETEIRO, A. A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition. *Information Sciences*, v. 180, n. 22, p. 4290–4311, nov. 2010. ISSN 00200255.
- BASILIO, S. d. C. A.; MORENO, M. F.; BARRÉRE, E. Supporting interaction and audience analysis in interactive TV systems. In: *Proceedings of the 11th european conference on Interactive TV and video - EuroITV '13*. New York, New York, USA: ACM Press, 2013. p. 23–30. ISBN 9781450319515.
- BENNETT, J.; LANNING, S. The Netflix Prize. In: *Proceedings of KDD Cup and Workshop*. San Jose, California, USA: [s.n.], 2007. p. 3–6. ISBN 9781595938343.
- BJELICA, M. Towards TV recommender system: experiments with user modeling. *IEEE Transactions on Consumer Electronics*, v. 56, n. 3, p. 1763–1769, ago. 2010. ISSN 0098-3063.
- BLANCO-FERNÁNDEZ, Y.; LÓPEZ-NORES, M.; GIL-SOLLA, A.; RAMOS-CABRER, M.; PAZOS-ARIAS, J. J. User-generated contents and reasoning-based personalization: Ingredients for a novel model of mobile TV. *Expert Systems with Applications*, v. 38, n. 5, p. 5289–5298, maio 2011. ISSN 09574174.

- BOBADILLA, J.; BOJORQUE, R.; ESTEBAN, A. H.; HURTADO, R. Recommender Systems Clustering Using Bayesian Non Negative Matrix Factorization. *IEEE Access*, IEEE, v. 6, p. 3549–3564, 2018. ISSN 2169-3536.
- BOBADILLA, J.; ORTEGA, F.; HERNANDO, A.; GUTIÉRREZ, A. Recommender systems survey. *Knowledge-Based Systems*, Elsevier B.V., v. 46, p. 109–132, 2013. ISSN 09507051.
- BREESE, J. S.; HECKERMAN, D.; KADIE, C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In: *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998. p. 43–52.
- BROWN, A.; AIZPURUA, A.; JAY, C.; EVANS, M.; GLANCY, M.; HARPER, S. Contrasting delivery modes for second screen TV content—Push or pull? *International Journal of Human-Computer Studies*, Elsevier Ltd, v. 129, n. May 2017, p. 15–26, sep 2019. ISSN 10715819.
- BRYNJOLFSSON, E.; HU, Y. J.; SMITH, M. D. Consumer Surplus in the Digital Economy: Estimating the Value of Increased Product Variety at Online Booksellers. *Management Science*, v. 49, n. 11, p. 1580–1596, nov. 2003. ISSN 0025-1909.
- BUSCHMANN, F.; HENNEY, K.; SCHMIDT, D. *Pattern-Oriented Software Architecture: A Pattern Language for Distributed Computing*. [S.l.]: John Wiley & Sons, 2007. 339–340 p. ISBN 0470059028, 9780470059029.
- CACHEDA, F.; CARNEIRO, V.; FERNÁNDEZ, D.; FORMOSO, V. Comparison of collaborative filtering algorithms. *ACM Transactions on the Web*, ACM, v. 5, n. 1, p. 1–33, fev. 2011. ISSN 15591131.
- CASTELLS, P.; VARGAS, S.; WANG, J. Novelty and Diversity Metrics for Recommender Systems : Choice , Discovery and Relevance. In: *Proceedings of International Workshop on Diversity in Document Retrieval (DDR)*. [S.l.: s.n.], 2011. p. 1–8.
- CESAR, P.; BULTERMAN, D. C. a.; JANSEN, J. Leveraging user impact: an architecture for secondary screens usage in interactive television. *Multimedia Systems*, v. 15, n. 3, p. 127–142, jun. 2009. ISSN 0942-4962.
- CESAR, P.; CHORIANOPOULOS, K.; JENSEN, J. F. Social television and user interaction. *Computers in Entertainment*, ACM, v. 6, n. 1, p. 1–10, maio 2008. ISSN 15443574.
- CESAR, P.; GEERTS, D. Past, present, and future of social TV: A categorization. In: *2011 IEEE Consumer Communications and Networking Conference (CCNC)*. Las Vegas, NV, USA: IEEE, 2011. p. 347–351. ISBN 978-1-4244-8789-9.
- CESAR, P.; KNOCHE, H.; BULTERMAN, D. C. A. From One to Many Boxes: Mobile Devices as Primary and Secondary Screens. In: MARCUS, A.; ROIBÁS, A. C.; SALA, R. (Ed.). *Mobile TV: Customizing Content and Experience*. London: Springer London, 2010, (Human-Computer Interaction Series). p. 327–348. ISBN 978-1-84882-702-8.

- CHANG, S.; HARPER, F. M.; TERVEEN, L. Using Groups of Items to Bootstrap New Users in Recommender Systems. In: *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing - CSCW '15*. New York, New York, USA: ACM, 2015. p. 1258–1269. ISBN 9781450329224.
- CHEN, R.; HUA, Q.; CHANG, Y.-S.; WANG, B.; ZHANG, L.; KONG, X. A Survey of Collaborative Filtering-Based Recommender Systems: From Traditional Methods to Hybrid Methods Based on Social Networks. *IEEE Access*, IEEE, v. 6, p. 64301–64320, 2018. ISSN 2169-3536.
- COSTA, L. d. F.; RODRIGUES, F. A.; TRAVIESO, G.; BOAS, P. R. V. Characterization of complex networks: A survey of measurements. *Advances in Physics*, v. 56, n. 1, p. 167–242, jan. 2007. ISSN 0001-8732.
- COURTOIS, C.; D'HEER, E. Second screen applications and tablet users. In: *Proceedings of the 10th European conference on Interactive tv and video - EuroITV '12*. New York, New York, USA: ACM Press, 2012. v. 11, p. 153–156. ISBN 9781450311076.
- CRUICKSHANK, L.; TSEKLEVES, E.; WHITHAM, R.; HILL, A.; KONDO, K. Making Interactive TV Easier to Use: Interface Design for a Second Screen Approach. *The Design Journal*, Berg Publishers, v. 10, n. 3, p. 41–53, nov. 2007.
- DCMI. *Dublin Core Metada Initiative*. 2021. Disponível em: <<http://dublincore.org>>. Acesso em: fev. 2021.
- DEAN, J.; GHEMAWAT, S. MapReduce: simplified data processing on large clusters. *Communications of the ACM*, ACM, v. 51, n. 1, p. 107–113, jan. 2008. ISSN 00010782.
- FREY, B. J.; DUECK, D. Clustering by Passing Messages Between Data Points. *Science*, v. 315, n. 5814, p. 972–976, feb 2007. ISSN 0036-8075.
- GOLDBERG, D.; NICHOLS, D.; OKI, B. M.; TERRY, D. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, v. 35, n. 12, p. 61–70, dez. 1992. ISSN 00010782.
- GUO, M. Second Screening: Measuring Second Screen User Behavior in a Social Television Viewing Environment. *International Journal on Media Management*, Routledge, p. 1–20, aug 2020. ISSN 1424-1277.
- HAMEDANI, E. M.; KAEDI, M. Recommending the long tail items through personalized diversification. *Knowledge-Based Systems*, Elsevier B.V., v. 164, p. 348–357, jan 2019. ISSN 09507051.
- HARPER, F. M.; KONSTAN, J. A. The MovieLens Datasets. *ACM Transactions on Interactive Intelligent Systems*, v. 5, n. 4, p. 1–19, dec 2015. ISSN 21606455.
- HERLOCKER, J.; KONSTAN, J. A.; RIEDL, J. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval*, v. 5, n. 4, p. 287–310, 2002. ISSN 13864564.
- HERLOCKER, J. L.; KONSTAN, J. A.; BORCHERS, A.; RIEDL, J. An algorithmic framework for performing collaborative filtering. In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information*

- retrieval - *SIGIR '99*. New York, New York, USA: ACM Press, 1999. p. 230–237. ISBN 1581130961.
- HERLOCKER, J. L.; KONSTAN, J. A.; TERVEEN, L. G.; RIEDL, J. T. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, v. 22, n. 1, p. 5–53, jan. 2004. ISSN 10468188.
- HILL, W.; STEAD, L.; ROSENSTEIN, M.; FURNAS, G. Recommending and evaluating choices in a virtual community of use. In: *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '95*. New York, New York, USA: ACM Press, 1995. p. 194–201. ISBN 0201847051.
- HINZ, O.; ECKERT, J. The Impact of Search and Recommendation Systems on Sales in Electronic Commerce. *Business & Information Systems Engineering*, v. 2, n. 2, p. 67–77, mar. 2010. ISSN 1867-0202.
- HJELM, J. *Why IPTV?: Interactivity, Technologies and Services*. 1st. ed. Chichester, UK: John Wiley & Sons, Ltd, 2008. ISBN 9780470751695.
- IBGE. *Pesquisa Nacional por Amostra de Domicílios (PNAD) - Síntese dos Indicadores 2015*. Rio de Janeiro, RJ, BR: IBGE, 2016.
- JAIN, A. K. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, Elsevier B.V., v. 31, n. 8, p. 651–666, jun 2010. ISSN 01678655.
- JALILI, M.; AHMADIAN, S.; IZADI, M.; MORADI, P.; SALEHI, M. Evaluating Collaborative Filtering Recommender Algorithms: A Survey. *IEEE Access*, v. 6, p. 74003–74024, 2018. ISSN 2169-3536.
- 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 - KDD '09*. New York, New York, USA: ACM Press, 2009. p. 397. ISBN 9781605584959.
- JANNACH, D.; JUGOVAC, M. Measuring the Business Value of Recommender Systems. *ACM Transactions on Management Information Systems*, v. 10, n. 4, p. 1–23, dec 2019. ISSN 2158-656X.
- JAVARI, A.; GHARIBSHAH, J.; JALILI, M. Recommender systems based on collaborative filtering and resource allocation. *Social Network Analysis and Mining*, v. 4, n. 1, p. 234, dec 2014. ISSN 1869-5450.
- JEONG, B.; LEE, J.; CHO, H. Improving memory-based collaborative filtering via similarity updating and prediction modulation. *Information Sciences*, Elsevier Inc., v. 180, n. 5, p. 602–612, mar 2010. ISSN 00200255.
- KANT, S.; MAHARA, T. Nearest biclusters collaborative filtering framework with fusion. *Journal of Computational Science*, Elsevier B.V., v. 25, p. 204–212, mar 2018. ISSN 18777503.
- KAZIENKO, P.; ADAMSKI, M. AdROSA—Adaptive personalization of web advertising. *Information Sciences*, v. 177, n. 11, p. 2269–2295, jun 2007. ISSN 00200255.

- KIM, J.; KANG, S. An ontology-based personalized target advertisement system on interactive TV. *Multimedia Tools and Applications*, v. 64, n. 3, p. 517–534, dez. 2011. ISSN 1380-7501.
- KOOHI, H.; KIANI, K. User based Collaborative Filtering using fuzzy C-means. *Measurement*, Elsevier Ltd, v. 91, p. 134–139, sep 2016. ISSN 02632241.
- KOREN, Y.; BELL, R.; VOLINSKY, C. Matrix Factorization Techniques for Recommender Systems. *Computer*, John Wiley & Sons, Singapore Pte. Ltd, Singapore, v. 42, n. 8, p. 30–37, aug 2009. ISSN 0018-9162.
- KULESZA, R.; MEIRA, S. R.; FERREIRA, T. P.; ALEXANDRE, E. S.; FILHO, G. L.; NETO, M. C. M.; SANTOS, C. a.S. A Model-driven Approach for Integration of Interactive Applications and Web Services: A Case Study in Interactive Digital TV Platform. In: *2012 IEEE International Conference on Multimedia and Expo Workshops*. [S.l.]: IEEE, 2012. p. 266–271. ISBN 978-1-4673-2027-6.
- LAI, C.-F.; CHANG, J.-H.; HU, C.-C.; HUANG, Y.-M.; CHAO, H.-C. CPRS: A cloud-based program recommendation system for digital TV platforms. *Future Generation Computer Systems*, Elsevier B.V., v. 27, n. 6, p. 823–835, jun. 2011. ISSN 0167739X.
- LATHIA, N.; HAILES, S.; CAPRA, L. The effect of correlation coefficients on communities of recommenders. In: *Proceedings of the 2008 ACM symposium on Applied computing - SAC '08*. New York, New York, USA: ACM Press, 2008. p. 2000–2005. ISBN 9781595937537.
- LEE, H.; KIM, J.-g.; YANG, S.-j.; HONG, J. Personalized TV services based on TV-anytime for personal digital recorder. *IEEE Transactions on Consumer Electronics*, v. 51, n. 3, p. 885–892, ago. 2005. ISSN 0098-3063.
- LEE, S.; LEE, D.; LEE, S. Personalized DTV program recommendation system under a cloud computing environment. *IEEE Transactions on Consumer Electronics*, v. 56, n. 2, p. 1034–1042, maio 2010. ISSN 0098-3063.
- LI, G.; QIU, L.; YU, C.; CAO, H.; LIU, Y.; YANG, C. IPTV Channel Zapping Recommendation with Attention Mechanism. *IEEE Transactions on Multimedia*, IEEE, v. 9210, n. c, p. 1–1, 2020. ISSN 1520-9210. Aceito para publicação.
- LIM, C.; CHOI, J.-H.; NAM, S. W.; CHANG, J.-H. A new television audience measurement framework using smart devices. *Multimedia Tools and Applications*, v. 73, p. 1757–1776, 2014. ISSN 1380-7501.
- LINDEN, G.; SMITH, B.; YORK, J. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, v. 7, n. 1, p. 76–80, 2003. ISSN 1089-7801.
- LIU, H.; HU, Z.; MIAN, A.; TIAN, H.; ZHU, X. A new user similarity model to improve the accuracy of collaborative filtering. *Knowledge-Based Systems*, Elsevier B.V., v. 56, p. 156–166, jan 2014. ISSN 09507051.

- LOCHRIE, M.; COULTON, P. Mobile phones as second screen for TV, enabling inter-audience interaction. In: *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology - ACE '11*. New York, New York, USA: ACM Press, 2011. p. 1–2. ISBN 9781450308274.
- LOHMÜLLER, V.; WOLFF, C. Towards a Comprehensive Definition of Second Screen. In: *Proceedings of Mensch und Computer 2019 on - MuC'19*. New York, New York, USA: ACM Press, 2019. p. 167–177. ISBN 9781450371988.
- LÓPEZ-NORES, M.; BLANCO-FERNÁNDEZ, Y.; PAZOS-ARIAS, J. J.; GARCÍA-DUQUE, J.; RAMOS-CABRER, M.; GIL-SOLLA, A.; DÍAZ-REDONDO, R. P.; FERNÁNDEZ-VILAS, A. Receiver-side semantic reasoning for digital TV personalization in the absence of return channels. *Multimedia Tools and Applications*, v. 41, n. 3, p. 407–436, out. 2008. ISSN 1380-7501.
- LÜ, L.; MEDO, M.; YEUNG, C. H.; ZHANG, Y.-C.; ZHANG, Z.-K.; ZHOU, T. Recommender systems. *Physics Reports*, Elsevier B.V., v. 519, n. 1, p. 1–49, oct 2012. ISSN 03701573.
- MA, H.; ZHOU, D.; LIU, C.; LYU, M. R.; KING, I. Recommender systems with social regularization. In: *Proceedings of the fourth ACM international conference on Web search and data mining - WSDM '11*. New York, New York, USA: ACM Press, 2011. p. 287–296. ISBN 9781450304931.
- MAISONNEUVE, J.; DESCHANEL, M.; HEILES, J.; SHARPE, R. An Overview of IPTV Standards Development. *IEEE Transactions on Broadcasting*, v. 55, n. 2, p. 315–328, jun. 2009. ISSN 0018-9316.
- MASTHOFF, J. Group Modeling: Selecting a Sequence of Television Items to Suit a Group of Viewers. In: ARDISSONO, L.; KOBSA, A.; MAYBURY, M. (Ed.). *Personalized Digital Television - Targeting Programs to Individual Viewers*. Dordrecht: Kluwer Academic Publishers, 2004, (Human-Computer Interaction Series, v. 6). cap. 5, p. 93–141.
- MEIXNER, B.; GLANCY, M.; ROGERS, M.; WARD, C.; RÖGGLA, T.; CESAR, P. Multi-Screen Director. In: *Adjunct Publication of the 2017 ACM International Conference on Interactive Experiences for TV and Online Video*. New York, NY, USA: ACM, 2017. p. 57–62. ISBN 9781450350235.
- MIRBAKHS, N.; LING, C. X. Leveraging clustering to improve collaborative filtering. *Information Systems Frontiers*, v. 20, n. 1, p. 111–124, feb 2018. ISSN 1387-3326.
- MITCHELL, K.; JONES, A.; ISHMAEL, J.; RACE, N. J. P. Social TV: The impact of social awareness on content navigation within IPTV systems. *Computers in Entertainment*, v. 9, n. 3, p. 1–29, nov. 2011. ISSN 15443574.
- MORRIS, S.; SMITH-CHAIGNEAU, A. *Interactive TV Standards: A Guide to MHP, OCAP, and JavaTV*. Burlington: Focal Press, 2005. ISBN 0240806662.
- NAJAFABADI, M. K.; MAHRIN, M. N.; CHUPRAT, S.; SARKAN, H. M. Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data. *Computers in Human Behavior*, Elsevier Ltd, v. 67, p. 113–128, feb 2017. ISSN 07475632.

- NIELSEN. *Action Figures: How Second Screens Are Transforming TV Viewing*. 2013. Disponível em: <<http://www.nielsen.com/us/en/insights/news/2013/action-figures-how-second-screens-are-transforming-tv-viewing.html>>. Acesso em: out. 2021.
- NIELSEN. *The Nielsen Total Audience Report: Q2 2018*. The Nielsen Company, 2019. Disponível em: <<https://www.nielsen.com/wp-content/uploads/sites/3/2019/04/q2-2018-total-audience-report.pdf>>. Acesso em: set. 2020.
- PARK, Y.-J.; TUZHILIN, A. The long tail of recommender systems and how to leverage it. In: *Proceedings of the 2008 ACM conference on Recommender systems - RecSys '08*. New York, New York, USA: ACM Press, 2008. p. 11–18. ISBN 9781605580937.
- PAZZANI, M. J.; BILLSUS, D. Content-Based Recommendation Systems. In: BRUSILOVSKY, E.; KOBASA, A.; NEJDL, W. (Ed.). *The Adaptive Web: Methods and Strategies of Web Personalization*. 1. ed. Berlin, Heidelberg: Springer-Verlag, 2007, (Lecture Notes In Computer Science, v. 4321). cap. 10, p. 325–341. ISBN 978-3-540-72078-2.
- PIJETLOVIC, S.; JOVANOV, N.; VUKOBRAT, V.; BASICEVIC, I. One solution of a RESTful API for a cloud based DTV content provider. In: *2014 IEEE Fourth International Conference on Consumer Electronics Berlin (ICCE-Berlin)*. Berlin, Germany: IEEE, 2014. p. 384–387.
- POLATIDIS, N.; GEORGIADIS, C. K. A multi-level collaborative filtering method that improves recommendations. *Expert Systems with Applications*, Elsevier Ltd, v. 48, p. 100–110, apr 2016. ISSN 09574174.
- RAO, K. R.; BOJKOVIC, Z. S.; MILOVANOVIC, D. A. *Introduction to Multimedia Communications*. [S.l.]: Wiley, 2006. ISBN 0-471-46742-1.
- RESNICK, P.; IACOVOU, N.; SUCHAK, M.; BERGSTROM, P.; RIEDL, J. GroupLens: an open architecture for collaborative filtering of netnews. In: *Proceedings of the 1994 ACM conference on Computer supported cooperative work - CSCW '94*. New York, New York, USA: ACM Press, 1994. p. 175–186. ISBN 0897916891.
- RESNICK, P.; VARIAN, H. R. Recommender systems. *Communications of the ACM*, ACM, v. 40, n. 3, p. 56–58, mar. 1997. ISSN 00010782.
- ROBERTSON, S.; WHARTON, C.; ASHWORTH, C.; FRANZKE, M. Dual device user interface design: PDAs and interactive television. In: *Proceedings of the SIGCHI conference on Human factors in computing systems common ground - CHI '96*. New York, New York, USA: ACM Press, 1996. p. 79–86. ISBN 0897917774.
- ROMERO, C.; ESPEJO, P. G.; ZAFRA, A.; ROMERO, J. R.; VENTURA, S. Web usage mining for predicting final marks of students that use Moodle courses. *Computer Applications in Engineering Education*, v. 21, n. 1, p. 135–146, mar 2013. ISSN 10613773.
- ROSEN, K. H. *Discrete Mathematics and Its Applications*. 6. ed. New York, NY, USA: McGraw-Hill Higher Education, 2006. ISBN 0072880082.
- SANCHEZ, F.; BARRILERO, M.; ALVAREZ, F.; CISNEROS, G. User interest modeling for social TV-recommender systems based on audiovisual consumption. *Multimedia Systems*, v. 19, p. 493–507, 2013. ISSN 09424962.

- SARWAR, B.; KARYPIS, G.; KONSTAN, J.; REIDL, J. Item-based collaborative filtering recommendation algorithms. In: *Proceedings of the tenth international conference on World Wide Web - WWW '01*. New York, New York, USA: ACM Press, 2001. p. 285–295. ISBN 1581133480.
- SCHAFER, J. B.; FRANKOWSKI, D.; HERLOCKER, J.; SEN, S. Collaborative filtering recommender systems. In: BRUSILOVSKY, P.; KOBASA, A.; NEJDL, W. (Ed.). *The Adaptive Web: Methods and Strategies of Web Personalization*. Berlin, Heidelberg: Springer-Verlag, 2007, (Lecture Notes in Computer Science, v. 4321). cap. 9, p. 291–324. ISBN 978-3-540-72078-2.
- SHANG, M.-S.; LÜ, L.; ZHANG, Y.-C.; ZHOU, T. Empirical analysis of web-based user-object bipartite networks. *EPL (Europhysics Letters)*, v. 90, n. 4, p. 48006, maio 2010. ISSN 0295-5075.
- SHARDANAND, U.; MAES, P. Social information filtering: algorithms for automating “word of mouth”. In: *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '95*. New York, New York, USA: ACM Press, 1995. p. 210–217. ISBN 0201847051.
- SMITH, B.; LINDEN, G. Two Decades of Recommender Systems at Amazon.com. *IEEE Internet Computing*, IEEE, v. 21, n. 3, p. 12–18, may 2017. ISSN 1089-7801.
- SMYTH, B.; COTTER, P. Case-Studies on the Evaluation of the Personalized Electronic Program Guide. In: ARDISSONO, L.; KOBASA, A.; MAYBURY, M. (Ed.). *Personalized Digital Television - Targeting Programs to Individual Viewers*. Dordrecht: Kluwer Academic Publishers, 2004, (Human-Computer Interaction Series, v. 6). cap. 3, p. 53–71.
- SOARES, L. F. G.; COSTA, R. M.; MORENO, M. F.; MORENO, M. F. Multiple exhibition devices in DTV systems. In: *Proceedings of the seventeen ACM international conference on Multimedia*. Beijing, China: ACM Press, 2009. p. 281–290. ISBN 978-1-60558-608-3.
- SPRING. *Spring Framework*. 2021. Disponível em: <<https://spring.io/projects/spring-framework>>. Acesso em: fev. 2021.
- SRIVASTAVA, H. *Interactive TV Technology and Markets*. [S.l.]: Artech House, 2001. ISBN 9781580535489.
- SRIVASTAVA, J.; COOLEY, R.; DESHPANDE, M.; TAN, P.-n. Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *ACM SIGKDD Explorations Newsletter*, ACM, v. 1, n. 2, p. 12–23, jan 2000. ISSN 19310145.
- SUN, Z.; HAN, L.; HUANG, W.; WANG, X.; ZENG, X.; WANG, M.; YAN, H. Recommender systems based on social networks. *Journal of Systems and Software*, Elsevier Inc., v. 99, p. 109–119, 2015. ISSN 01641212.
- SWEARINGEN, K.; SINHA, R. Beyond Algorithms: An HCI Perspective on Recommender Systems. In: *Proceedings of the ACM SIGIR 2001 Workshop on Recommender Systems*. [S.l.: s.n.], 2001. p. 1–11.
- TEIXEIRA, C. a. C.; MELO, E. L.; CATTELAN, R. G.; PIMENTEL, M. D. G. C. Taking advantage of contextualized interactions while users watch TV. *Multimedia Tools and Applications*, v. 50, n. 3, p. 587–607, fev. 2010. ISSN 1380-7501.

- TOPALLI, I.; KILINC, S. Modelling user habits and providing recommendations based on the hybrid broadcast broadband television using neural networks. *IEEE Transactions on Consumer Electronics*, IEEE, v. 62, n. 2, p. 182–190, may 2016. ISSN 0098-3063.
- TRAN, C.; KIM, J.-Y.; SHIN, W.-Y.; KIM, S.-W. Clustering-Based Collaborative Filtering Using an Incentivized/Penalized User Model. *IEEE Access*, v. 7, p. 62115–62125, 2019. ISSN 2169-3536.
- TSEKLEVES, E.; WHITHAM, R.; KONDO, K.; HILL, A. Investigating media use and the television user experience in the home. *Entertainment Computing*, International Federation for Information Processing, v. 2, n. 3, p. 151–161, jan. 2011. ISSN 18759521.
- TV-ANYTIME. *TV-Anytime*. 2021. Disponível em: <<http://www.tv-anytime.org/>>. Acesso em: fev. 2021.
- VALDIVIEZO-DIAZ, P.; ORTEGA, F.; COBOS, E.; LARA-CABRERA, R. A Collaborative Filtering Approach Based on Naïve Bayes Classifier. *IEEE Access*, v. 7, p. 108581–108592, 2019. ISSN 2169-3536.
- VANATTENHOVEN, J.; GEERTS, D. Social experiences within the home using second screen TV applications. *Multimedia Tools and Applications*, Multimedia Tools and Applications, v. 76, n. 4, p. 5661–5689, feb 2017. ISSN 1380-7501.
- VANATTENHOVEN, J.; GEERTS, D. Synchronization for Secondary Screens and Social TV: User Experience Aspects. In: MONTAGUD, M.; CESAR, P.; BORONAT, F.; JANSEN, J. (Ed.). *MediaSync*. Cham: Springer International Publishing, 2018. p. 271–294. ISBN 978-3-319-65839-1.
- VELUSAMY, S.; GOPAL, L.; BHATNAGAR, S.; VARADARAJAN, S. An efficient ad recommendation system for TV programs. *Multimedia Systems*, v. 14, n. 2, p. 73–87, abr. 2008. ISSN 0942-4962.
- VIEIRA, P. K. M.; LINO, N. C. Q. Recommendation of Multimedia Content in Digital Convergence Environments: TV and Web. *IEEE Latin America Transactions*, v. 13, n. 10, p. 3467–3475, oct 2015. ISSN 1548-0992.
- WANG, B.; WANG, J.; LU, H. Exploiting content relevance and social relevance for personalized ad recommendation on internet tv. *ACM Trans. Multimedia Comput. Commun. Appl.*, ACM, New York, NY, USA, v. 9, n. 4, p. 26:1–26:23, ago. 2013. ISSN 1551-6857.
- WANG, Y.; DENG, J.; GAO, J.; ZHANG, P. A hybrid user similarity model for collaborative filtering. *Information Sciences*, Elsevier Inc., v. 418-419, p. 102–118, dec 2017. ISSN 00200255.
- XUE, G. R.; LIN, C.; YANG, Q.; XI, W.; ZENG, H. J.; YU, Y.; CHEN, Z. Scalable collaborative filtering using cluster-based smoothing. *SIGIR 2005 - Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 114–121, 2005.
- YU, Z.; ZHOU, X.; HAO, Y.; GU, J. TV Program Recommendation for Multiple Viewers Based on user Profile Merging. *User Modeling and User-Adapted Interaction*, v. 16, n. 1, p. 63–82, jun. 2006. ISSN 0924-1868.

- YU, Z.; ZHOU, X.; ZHOU, L.; DU, K. A hybrid similarity measure of contents for TV personalization. *Multimedia Systems*, v. 16, n. 4-5, p. 231–241, maio 2010. ISSN 0942-4962.
- ZAHRA, S.; GHAZANFAR, M. A.; KHALID, A.; AZAM, M. A.; NAEEM, U.; PRUGEL-BENNETT, A. Novel centroid selection approaches for KMeans-clustering based recommender systems. *Information Sciences*, Elsevier Inc., v. 320, p. 156–189, nov 2015. ISSN 00200255.
- ZAPATER, M.; BRESSAN, G. A Proposed Approach for Quality of Experience Assurance of IPTV. In: *First International Conference on the Digital Society (ICDS'07)*. [S.l.]: IEEE, 2007. p. 1–6. ISBN 0769527604.
- ZHANG, J.; LIN, Y.; LIN, M.; LIU, J. An effective collaborative filtering algorithm based on user preference clustering. *Applied Intelligence*, Applied Intelligence, v. 45, n. 2, p. 230–240, sep 2016. ISSN 0924-669X.
- ZHOU, T.; KUSCSIK, Z.; LIU, J.-G.; MEDO, M.; WAKELING, J. R.; ZHANG, Y.-C. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences of the United States of America*, v. 107, n. 10, p. 4511–4515, mar. 2010. ISSN 1091-6490.
- ZHOU, T.; REN, J.; MEDO, M.; ZHANG, Y.-C. Bipartite network projection and personal recommendation. *Physical Review E*, v. 76, n. 4, p. 1–7, out. 2007. ISSN 1539-3755.
- ZHOU, X.; HE, J.; HUANG, G.; ZHANG, Y. SVD-based incremental approaches for recommender systems. *Journal of Computer and System Sciences*, Elsevier Inc., v. 81, n. 4, p. 717–733, jun 2015. ISSN 00220000.
- ZIMMERMAN, J.; KURAPATI, K.; BUCZAK, A. L.; SCHAFFER, D.; GUTTA, S.; MARTINO, J. TV Personalization System: Design of a TV Show Recommender Engine and Interface. In: ARDISSONO, L.; KOBSA, A.; MAYBURY, M. (Ed.). *Personalized Digital Television - Targeting Programs to Individual Viewers*. Dordrecht: Kluwer Academic Publishers, 2004, (Human-Computer Interaction Series, v. 6). cap. 2, p. 27–51.

Apêndices

APÊNDICE A

Notações Matemáticas e Símbolos

ESSE apêndice dá uma visão geral das notações matemáticas e dos símbolos utilizados no trabalho:

- \mathcal{U} = conjunto dos usuários $\{u_0, u_1, \dots, u_m\}$.
- \mathcal{O} = conjunto dos objetos $\{o_0, o_1, \dots, o_n\}$.
- \mathcal{X} = conjunto dos valores possíveis de avaliação. Tipicamente $\mathcal{X} = \{1, 2, 3, 4, 5\}$.
- $r_{u_i o_k}$ = avaliação observada do usuário u_i pelo objeto o_k .
- $\hat{r}_{u_i o_k}$ = avaliação predita do usuário u_i pelo objeto o_k .
- \bar{r}_{u_i} = média das avaliações observadas do usuário u_i .
- $z_{u_i o_k}$ = avaliação do usuário u_i para o objeto o_k normalizada pela função *z-score*.
- $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_c\}$ = conjunto dos clusters, onde cada elemento \mathcal{C}_c é um subconjunto distinto e não vazio dos objetos de \mathcal{O} .
- $\bar{r}_{u_i}^{\mathcal{C}_c}$ = média das avaliações observadas do usuário u_i considerando somente o cluster de interesse \mathcal{C}_c .
- $z_{u_i o_k}$ = avaliação do usuário u_i para o objeto o_k normalizada pela função *z-score*.
- $z_{u_i o_k}^{\mathcal{C}_c}$ = avaliação do usuário u_i para o objeto o_k normalizada pela função *z-score* considerando somente objetos do cluster de interesse \mathcal{C}_c .
- $\mathcal{U}_{o_k} = \{u_i \in \mathcal{U} : r_{u_i o_k} \neq \emptyset\}$ (conjunto dos usuários que avaliaram o objeto o_k).

- $\mathcal{U}_{\mathcal{C}_c} = \{u_i \in \mathcal{U} : r_{u_i o_k} \neq \emptyset \wedge o_k \in \mathcal{C}_c\}$ (conjunto dos usuários que avaliaram pelo menos um objeto $o_k \in \mathcal{C}_c$).
- $\mathcal{O}_{u_i} = \{o_k \in \mathcal{O} : r_{u_i o_k} \neq \emptyset\}$ (conjunto de todo os objetos avaliados pelo usuário u_i).
- $\mathcal{O}_{u_i u_j} = \{\mathcal{O}_{u_i} \cap \mathcal{O}_{u_j}\}$ (conjunto dos objetos avaliados em comum pelos usuários u_i e u_j).
- $\mathcal{N}_{u_i o_k} =$ conjunto de usuários formado pela vizinhança do usuário u_i para o objeto o_k .
- $\mathcal{N}_{u_i o_k}^{\mathcal{C}_c} =$ conjunto de usuários formado pela vizinhança do usuário u_i para o objeto o_k considerando a matriz de similaridade do cluster de interesse \mathcal{C}_c .
- $R = \{r_{u_i o_k}\}^{m \times n} =$ matriz de avaliações $\mathcal{U} \times \mathcal{O}$ consistindo em avaliações $r_{u_i o_k} \in \mathcal{X} \cup \{\emptyset\}$ para todo usuário $u_i \in \mathcal{U}$ e objeto $o_k \in \mathcal{O}$.
- $\mathcal{R}_{train} =$ conjunto de dados de treinamento.
- $\mathcal{R}_{test} =$ conjunto de dados de teste.
- $\mathcal{S}_{\mathcal{C}_c} = \{(u_i, o_k, r_{u_i o_k}) \in \mathcal{R}_{train} : o_k \in \mathcal{C}_c\}$ (conjunto das preferências selecionadas cujos objetos pertencem ao cluster de interesse \mathcal{C}_c).
- $\#\mathcal{A} =$ número de itens no conjunto \mathcal{A} .
- $s_{u_i u_j} =$ similaridade entre os usuários u_i e u_j .
- $s_{o_k o_l} =$ similaridade entre os objetos o_k e o_l .
- $w_{u_i u_j} =$ alocação de recursos entre os usuários u_i e u_j .
- $S = \{s_{u_i u_j}\}^{m \times m} =$ matriz de similaridade para todo usuário $u_i \in \mathcal{U}$ e $u_j \in \mathcal{U}$ composta pelos valores de similaridade $s_{u_i u_j}$ entre os usuários u_i e u_j .

APÊNDICE B

Algoritmos de Clusterização

ESSE apêndice descreve duas técnicas de clusterização utilizadas nesse trabalho: *affinity propagation* (FREY; DUECK, 2007) e K-Means (JAIN, 2010). Essas técnicas foram utilizadas para encontrar grupos de objetos relacionados, o que é requerido para o método de recomendação baseado em similaridade local proposto neste trabalho.

B.1 *Affinity Propagation*

O algoritmo *affinity propagation* é baseado no conceito de troca de mensagens entre pontos de dados (usuários/objetos). No procedimento de troca de mensagens, cada ponto de dado enviam mensagens para todos os seus pares. Esse procedimento é um processo de atualização iterativo que visa encontrar “exemplares”. Um exemplar é um ponto de dado que melhor representa todos os outros pontos de dados em um *cluster*, ou seja, o centro do cluster. Inicialmente, todos os pontos de dados são candidatos a exemplar. O conjunto dos exemplares é refinado a cada iteração até se alcançar a convergência dos pontos de dados. Uma particularidade desse algoritmo é que o número de *clusters* não é definido com antecedência. Em vez disso, o algoritmo encontra um número apropriado de *clusters* (número de exemplares) durante o processo de troca de mensagens.

Existem dois tipos de mensagens trocadas entre os pares de pontos: *responsabilidade* e *disponibilidade*¹. A mensagem de *responsabilidade* $r(i, j)$ é enviada do ponto i ao ponto j e quantifica o quão adequado seria para o ponto j servir como um exemplar para

¹ Do inglês *responsibility* e *availability*, que significam *responsabilidade* e *disponibilidade*, respectivamente.

o ponto i . A mensagem de *disponibilidade* $a(i, j)$ é enviada do ponto j ao ponto i e estima o quão apropriado seria para o ponto i escolher o ponto j como o seu exemplar.

O algoritmo é executado primeiramente inicializado todas as *disponibilidades* com 0 (zero). Então, a matriz de similaridade $S = \{s_{ij}\}$ é utilizada como entrada para o processo, onde inicialmente s_{ij} quantifica o quão adequado seria para o ponto j servir como um exemplar para o ponto i . Um outro parâmetro de entrada, conhecido como *preferência*, é definido com a diagonal da matriz S . O valor da *preferência* indica o quão provável é para um ponto de dado ser escolhido como um exemplar. Valores menores de *preferência* significam menos clusters, enquanto que valores mais altos significam mais clusters.

Após da inicialização e definição dos parâmetros de entrada, o processo se inicia e cada iteração consiste de três passos: 1) calcular todas as *responsabilidades*, 2) calcular todas as *disponibilidades* e 3) combinar *responsabilidades* e *disponibilidades* para identificar os exemplares.

No primeiro passo de cada iteração, cada *responsabilidade* $r(i, j)$ é calculada de acordo com a Equação (B.1).

$$r(i, j) = s_{ij} - \max_{j' | j' \neq j} \{a(i, j') + s_{ij'}\} \quad (\text{B.1})$$

No segundo passo, todos os pontos i e j onde $i \neq j$, a *disponibilidade* $a(i, j)$ é calculada usando a Equação (B.2). A *disponibilidade* $a(j, j)$, também conhecida como *autodisponibilidade*, é calculada usando a Equação (B.3).

$$a(i, j) = \min \left\{ 0, r(j, j) + \sum_{i' | i' \notin \{i, k\}} \max\{0, r(i', j)\} \right\} \quad (\text{B.2})$$

$$a(j, j) = \sum_{i' \neq j} \max\{0, r(i', j)\} \quad (\text{B.3})$$

As mensagens são atualizadas utilizando um fator de amortecimento λ , que pode ser entendido como um fator de convergência para evitar oscilações numéricas elevadas. Cada mensagem é definida como λ vezes o valor da iteração anterior mais $1 - \lambda$ vezes o valor calculado na iteração atual, onde λ tipicamente varia de 0.5 a 1.

No terceiro e último passo de cada iteração, o algoritmo combina as *responsabilidades* e *disponibilidades* para identificar os exemplares candidatos. O ponto j que maximiza $r(i, j) + a(i, j)$ corresponde ao exemplar candidato para o ponto i . Dessa maneira, cada ponto de dado i atualiza seu exemplar candidato e então o algoritmo itera novamente.

O algoritmo continuar iterando pelos passos descritos anteriormente até que um dos seguintes critérios seja atendido: 1) ao atingir um certo número de iterações, ou 2)

os exemplares candidatos permanecerem inalterados por um certo número de iterações. Quando o algoritmo alcança um critério de parada, os centros dos *clusters* são os exemplares obtidos a partir de cada combinação das *responsabilidades* e *disponibilidades*.

No contexto da FC, os pontos de dados do algoritmo *affinity propagation* podem ser ou usuários ou objetos. O resultado do processo de clusterização depende de qual tipo de dado é utilizado como entrada. Assim, se a intenção é encontrar *clusters* de usuários, uma matriz de similaridade baseada em usuários deve ser provida como entrada. Se a intenção é encontrar *clusters* de objetos, uma matriz de similaridade baseada em objetos deve ser passada como entrada do algoritmo.

B.2 K-Means

O K-Means é um algoritmo de clusterização baseado em centroide muito popular. O objetivo do algoritmo é encontrar partições de pontos de dados que minimizem a distância dos pontos do *cluster* em relação ao seu centroide (JAIN, 2010). Formalmente, Dada uma quantidade esperada de *clusters* K e um conjunto de pontos de dados $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$ onde cada ponto tem dimensão d , o objetivo do algoritmo é encontrar K subconjuntos $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k\}$ de pontos de dados não sobrepostos (clusters). Para isso, dado que μ_k seja o centroide do *cluster* \mathcal{C}_k , o objetivo do K-Means é encontrar é encontrar uma configuração de partições para \mathcal{C} que minimize a soma do erro quadrático para todos os K clusters:

$$\varepsilon(\mathcal{C}) = \arg \min_{\mathcal{C}} \sum_{k=1}^K \sum_{x_i \in \mathcal{C}_k} \|x_i - \mu_k\|^2 \quad (\text{B.4})$$

O algoritmo K-Means começa selecionando K pontos aleatoriamente, que representam o centroide de cada *cluster*. Então, o algoritmo executa cálculos iterativos para otimizar a posição de cada centroide e, conseqüentemente, encontrar uma configuração de partições que minimiza o erro $\varepsilon(\mathcal{C})$. O algoritmo itera sobre os passos a seguir até que a pertinência de cada ponto de dado ao seu respectivo *cluster* se estabilize:

1. Atualizar as partições ao associar cada ponto de dado ao *cluster* mais próximo.
2. Atualizar o centroide dos *clusters* para cada partição.

O centroide do *cluster* μ_k é atualizado da seguinte maneira:

$$\mu_k = \frac{\sum_{x_i \in \mathcal{C}_k} x_i}{\#\mathcal{C}_k} \quad (\text{B.5})$$

APÊNDICE C

Performance dos Métodos de Recomendação

ESSE apêndice apresenta gráficos e tabelas para complementar os resultados apresentados no Capítulo 5. As figuras 18, 19, 20 e 21 apresentam o mapa de calor do erro a partir da relação entre o número de *clusters* e o tamanho da vizinhança.

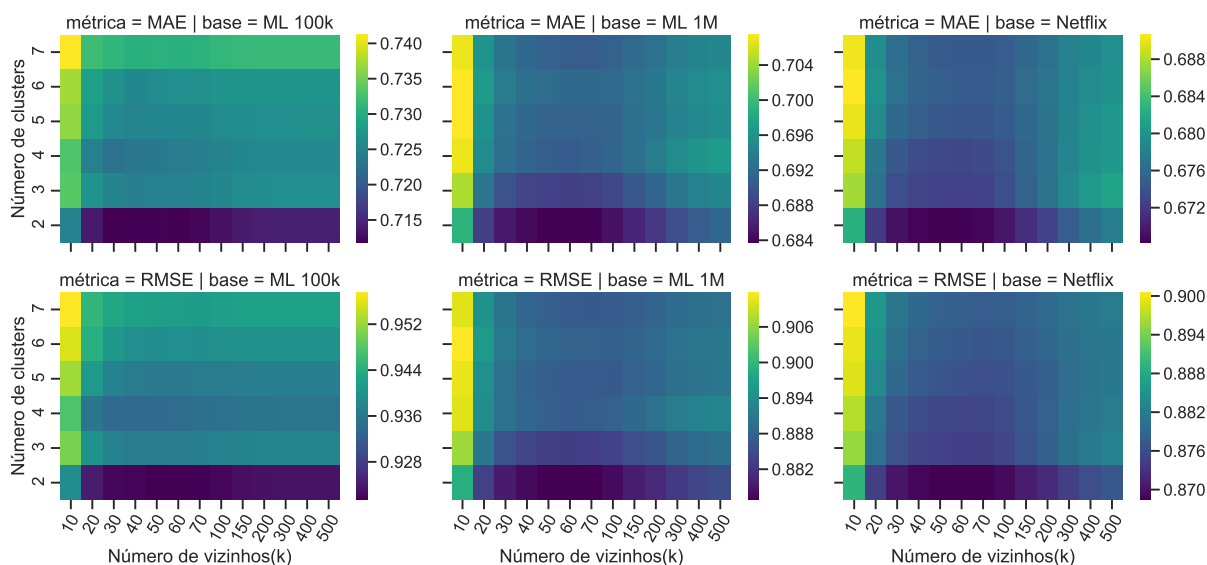


Figura 18: Mapa de calor dos valores de MAE e RMSE para a partição 1 considerando um número variável de vizinhos e um número variável de *clusters* para o método SL_KM.

Mais adiante, as tabelas 8 a 19 apresentam os dados das melhorias alcançadas pelos algoritmos baseados em similaridade local sobre os algoritmos baseados em similaridade global. As tabelas mostram os dados de precisão da predição (MAE e RMSE) para as

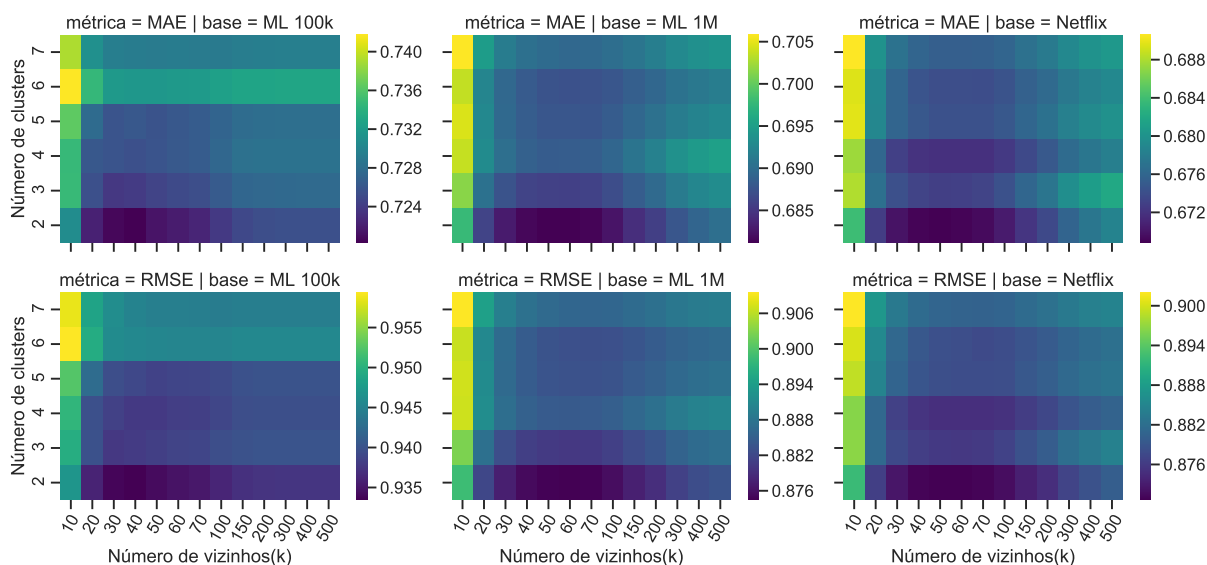


Figura 19: Mapa de calor dos valores de MAE e RMSE para a partição 2 considerando um número variável de vizinhos e um número variável de *clusters* para o método SL_KM.

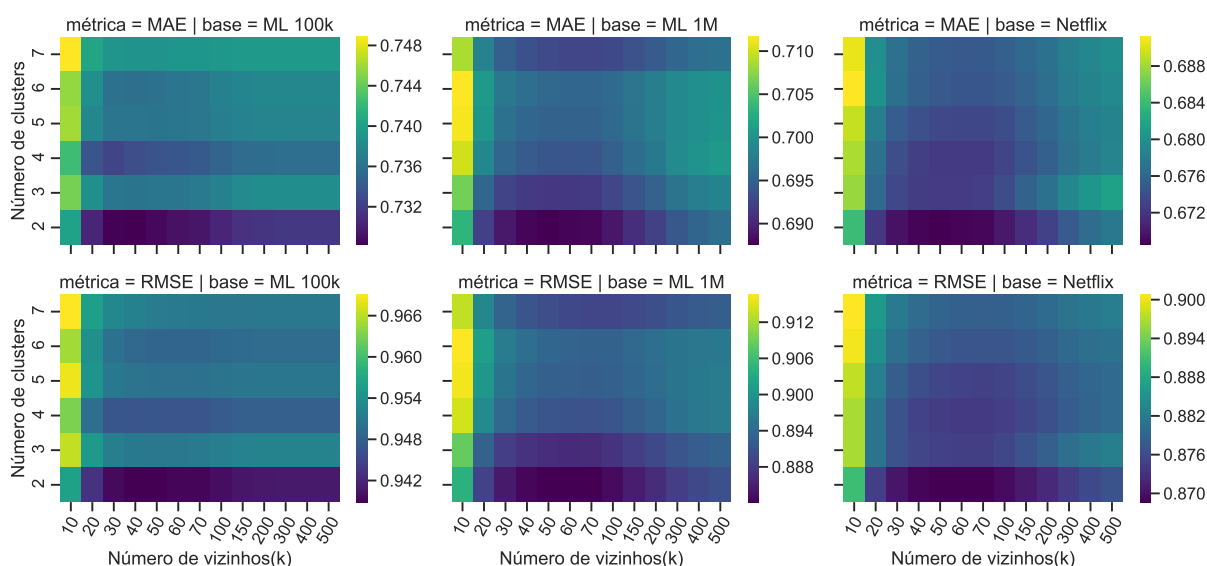


Figura 20: Mapa de calor dos valores de MAE e RMSE para a partição 3 considerando um número variável de vizinhos e um número variável de *clusters* para o método SL_KM.

bases de dados ML 100k, ML 1M e Netflix, considerando as partições de 1 a 4 em cada base de dados.

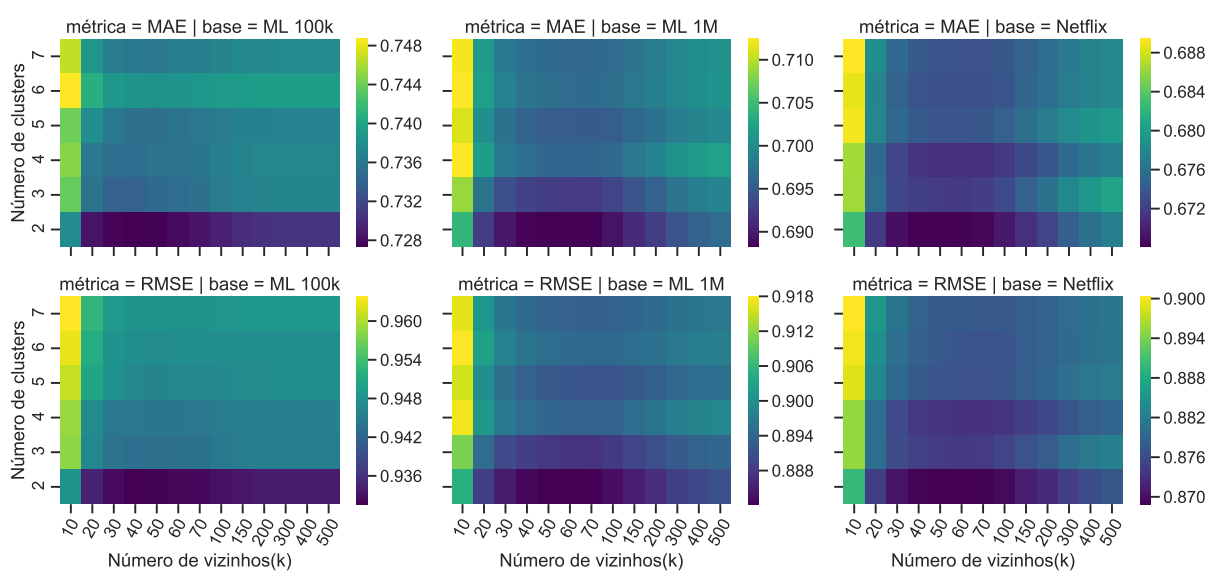


Figura 21: Mapa de calor dos valores de MAE e RMSE para a partição 4 considerando um número variável de vizinhos e um número variável de *clusters* para o método SL_KM.

Tabela 8: Melhorias de performance da abordagem proposta nesse trabalho em relação os métodos de referência considerando as métricas MAE e o RMSE para a partição 1 da base de dados ML 100k. Os valores em negrito em cada linha dos métodos de similaridade global indicam a pior performance, enquanto os valores em negrito em cada linha dos métodos baseados em similaridade local indicam a melhor performance.

k	Similaridade global												Similaridade local					
	SG_COS		SG_PCC		SG_SPPC		SG_WPCC		SL_AP		SL_KM		Melhoria(%)					
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE				
10	0,733	0,945	0,729	0,939	0,73	0,939	0,734	0,947	0,72	0,931	0,725	0,939	1,797	1,658				
20	0,725	0,934	0,723	0,929	0,723	0,929	0,725	0,936	0,711	0,918	0,714	0,924	1,972	1,891				
30	0,724	0,932	0,722	0,928	0,722	0,928	0,724	0,933	0,71	0,916	0,712	0,922	1,9	1,811				
40	0,723	0,931	0,722	0,928	0,722	0,928	0,724	0,933	0,71	0,915	0,712	0,922	1,933	1,899				
50	0,724	0,932	0,723	0,929	0,723	0,929	0,725	0,933	0,711	0,916	0,712	0,921	1,904	1,866				
70	0,727	0,934	0,725	0,931	0,726	0,931	0,727	0,935	0,712	0,917	0,712	0,922	2,104	1,866				
100	0,73	0,936	0,728	0,933	0,728	0,933	0,728	0,936	0,713	0,918	0,713	0,922	2,325	1,923				
300	0,737	0,941	0,734	0,938	0,734	0,938	0,732	0,938	0,716	0,92	0,714	0,923	3,053	2,226				
500	0,737	0,942	0,735	0,938	0,735	0,938	0,733	0,938	0,716	0,92	0,714	0,923	3,118	2,256				

Tabela 9: Melhorias de performance da abordagem proposta nesse trabalho em relação os métodos de referência considerando as métricas MAE e o RMSE para a partição 2 da base de dados ML 100k. Os valores em negrito em cada linha dos métodos de similaridade global indicam a pior performance, enquanto os valores em negrito em cada linha dos métodos baseados em similaridade local indicam a melhor performance.

k	Similaridade global												Similaridade local					
	SG_COS		SG_PCC		SG_SPPC		SG_WPCC		SL_AP		SL_KM		Melhoria(%)					
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE				
10	0,735	0,953	0,735	0,95	0,736	0,95	0,739	0,958	0,731	0,95	0,731	0,947	1,183	1,149				
20	0,729	0,942	0,728	0,941	0,728	0,941	0,73	0,945	0,721	0,936	0,722	0,936	1,229	1,006				
30	0,727	0,941	0,727	0,939	0,727	0,939	0,73	0,944	0,72	0,934	0,721	0,934	1,28	1,045				
40	0,73	0,943	0,729	0,94	0,729	0,94	0,731	0,944	0,72	0,934	0,72	0,933	1,449	1,134				
50	0,731	0,943	0,731	0,941	0,73	0,941	0,732	0,945	0,721	0,935	0,721	0,934	1,455	1,121				
70	0,735	0,947	0,733	0,944	0,733	0,944	0,734	0,947	0,723	0,937	0,723	0,935	1,703	1,234				
100	0,739	0,95	0,737	0,947	0,737	0,947	0,737	0,949	0,725	0,938	0,724	0,936	2,022	1,446				
300	0,747	0,957	0,744	0,954	0,744	0,953	0,742	0,952	0,728	0,941	0,726	0,937	2,818	2,017				
500	0,747	0,957	0,744	0,954	0,744	0,954	0,742	0,953	0,728	0,941	0,726	0,937	2,88	2,06				

Tabela 10: Melhorias de performance da abordagem proposta nesse trabalho em relação os métodos de referência considerando as métricas MAE e o RMSE para a partição 3 da base de dados ML 100k. Os valores em negrito em cada linha dos métodos de similaridade global indicam a pior performance, enquanto os valores em negrito em cada linha dos métodos baseados em similaridade local indicam a melhor performance.

k	Similaridade global												Similaridade local					
	SG_COS		SG_PCC		SG_SPPC		SG_WPCC		SL_AP		SL_KM		Melhoria(%)					
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE				
10	0,751	0,968	0,75	0,964	0,75	0,964	0,752	0,971	0,745	0,96	0,74	0,957	1,547	1,519				
20	0,741	0,951	0,74	0,949	0,74	0,949	0,741	0,954	0,735	0,948	0,73	0,943	1,524	1,185				
30	0,739	0,949	0,739	0,946	0,739	0,946	0,739	0,951	0,734	0,944	0,728	0,94	1,462	1,167				
40	0,739	0,948	0,739	0,947	0,739	0,947	0,74	0,952	0,733	0,943	0,728	0,939	1,602	1,31				
50	0,74	0,95	0,74	0,948	0,74	0,948	0,741	0,952	0,734	0,943	0,729	0,939	1,628	1,308				
70	0,743	0,951	0,741	0,949	0,741	0,949	0,742	0,952	0,734	0,944	0,729	0,94	1,787	1,326				
100	0,745	0,953	0,743	0,95	0,743	0,95	0,744	0,954	0,736	0,945	0,73	0,94	2,011	1,425				
300	0,752	0,958	0,75	0,955	0,75	0,955	0,748	0,956	0,738	0,947	0,731	0,941	2,697	1,732				
500	0,752	0,958	0,75	0,955	0,75	0,955	0,748	0,956	0,738	0,947	0,731	0,941	2,743	1,762				

Tabela 11: Melhorias de performance da abordagem proposta nesse trabalho em relação os métodos de referência considerando as métricas MAE e o RMSE para a partição 4 da base de dados ML 100k. Os valores em negrito em cada linha dos métodos de similaridade global indicam a pior performance, enquanto os valores em negrito em cada linha dos métodos baseados em similaridade local indicam a melhor performance.

k	Similaridade global												Similaridade local					
	SG_COS		SG_PCC		SG_SPPC		SG_WPCC		SL_AP		SL_KM		Melhoria(%)					
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE				
10	0,74	0,95	0,74	0,948	0,74	0,948	0,744	0,958	0,736	0,948	0,739	0,95	1,07	1,113				
20	0,733	0,941	0,732	0,937	0,732	0,937	0,734	0,943	0,726	0,935	0,729	0,936	1,069	0,929				
30	0,732	0,939	0,731	0,936	0,731	0,936	0,732	0,942	0,725	0,932	0,728	0,934	1,088	0,981				
40	0,733	0,94	0,732	0,937	0,732	0,937	0,733	0,942	0,724	0,932	0,728	0,933	1,23	1,103				
50	0,735	0,941	0,733	0,938	0,733	0,938	0,735	0,943	0,725	0,932	0,728	0,933	1,321	1,117				
70	0,737	0,943	0,735	0,94	0,735	0,94	0,737	0,944	0,727	0,934	0,729	0,933	1,432	1,139				
100	0,74	0,945	0,738	0,942	0,738	0,942	0,739	0,945	0,728	0,935	0,73	0,934	1,587	1,186				
300	0,747	0,951	0,745	0,948	0,745	0,948	0,744	0,948	0,731	0,937	0,731	0,935	2,244	1,653				
500	0,748	0,951	0,746	0,948	0,745	0,948	0,744	0,948	0,731	0,937	0,731	0,935	2,302	1,683				

Tabela 12: Melhorias de performance da abordagem proposta nesse trabalho em relação os métodos de referência considerando as métricas MAE e o RMSE para a partição 1 da base de dados ML 1M. Os valores em negrito em cada linha dos métodos de similaridade global indicam a pior performance, enquanto os valores em negrito em cada linha dos métodos baseados em similaridade local indicam a melhor performance.

k	Similaridade global												Similaridade local					
	SG_COS		SG_PCC		SG_SPPC		SG_WPCC		SL_AP		SL_KM		Melhoria(%)					
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE				
10	0,708	0,916	0,708	0,913	0,708	0,913	0,71	0,92	0,698	0,899	0,699	0,899	1,719	2,234				
20	0,698	0,902	0,697	0,897	0,697	0,897	0,699	0,904	0,687	0,884	0,688	0,884	1,696	2,271				
30	0,695	0,898	0,694	0,894	0,694	0,894	0,696	0,901	0,683	0,879	0,685	0,879	1,805	2,372				
40	0,695	0,897	0,694	0,892	0,694	0,892	0,695	0,899	0,683	0,878	0,684	0,878	1,802	2,419				
50	0,695	0,897	0,693	0,892	0,693	0,892	0,695	0,899	0,683	0,878	0,684	0,877	1,842	2,466				
70	0,696	0,898	0,695	0,893	0,695	0,893	0,696	0,9	0,683	0,878	0,684	0,877	1,88	2,525				
100	0,698	0,899	0,696	0,894	0,696	0,894	0,698	0,901	0,684	0,879	0,685	0,878	1,955	2,583				
300	0,706	0,907	0,704	0,903	0,704	0,903	0,706	0,908	0,691	0,886	0,689	0,883	2,368	2,727				
500	0,71	0,911	0,709	0,907	0,709	0,907	0,709	0,91	0,694	0,889	0,691	0,885	2,659	2,802				

Tabela 13: Melhorias de performance da abordagem proposta nesse trabalho em relação os métodos de referência considerando as métricas MAE e o RMSE para a partição 2 da base de dados ML 1M. Os valores em negrito em cada linha dos métodos de similaridade global indicam a pior performance, enquanto os valores em negrito em cada linha dos métodos baseados em similaridade local indicam a melhor performance.

k	Similaridade global												Similaridade local					
	SG_COS		SG_PCC		SG_SPPC		SG_WPCC		SL_AP		SL_KM		Melhoria(%)					
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE				
10	0,706	0,914	0,706	0,91	0,706	0,91	0,707	0,917	0,697	0,899	0,698	0,899	1,512	2,025				
20	0,697	0,901	0,695	0,895	0,695	0,895	0,697	0,902	0,686	0,883	0,686	0,882	1,597	2,253				
30	0,694	0,898	0,693	0,892	0,693	0,892	0,695	0,899	0,683	0,879	0,683	0,877	1,734	2,435				
40	0,694	0,897	0,693	0,891	0,692	0,891	0,694	0,898	0,682	0,878	0,682	0,875	1,815	2,535				
50	0,694	0,897	0,693	0,891	0,693	0,891	0,694	0,898	0,682	0,878	0,681	0,875	1,905	2,612				
70	0,695	0,897	0,693	0,892	0,693	0,892	0,695	0,899	0,682	0,878	0,681	0,875	2,006	2,661				
100	0,697	0,899	0,695	0,894	0,695	0,894	0,697	0,9	0,684	0,879	0,682	0,876	2,103	2,708				
300	0,706	0,907	0,704	0,902	0,704	0,903	0,705	0,907	0,69	0,886	0,688	0,881	2,535	2,823				
500	0,71	0,91	0,708	0,906	0,708	0,906	0,709	0,909	0,693	0,889	0,69	0,883	2,833	2,923				

Tabela 14: Melhorias de performance da abordagem proposta nesse trabalho em relação os métodos de referência considerando as métricas MAE e o RMSE para a partição 3 da base de dados ML 1M. Os valores em negrito em cada linha dos métodos de similaridade global indicam a pior performance, enquanto os valores em negrito em cada linha dos métodos baseados em similaridade local indicam a melhor performance.

k	Similaridade global												Similaridade local					
	SG_COS		SG_PCC		SG_SPPC		SG_WPCC		SL_AP		SL_KM		Melhoria(%)					
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE				
10	0,713	0,923	0,712	0,918	0,713	0,918	0,714	0,926	0,702	0,904	0,703	0,904	1,665	2,336				
20	0,703	0,909	0,703	0,904	0,702	0,904	0,704	0,911	0,692	0,889	0,692	0,889	1,755	2,441				
30	0,701	0,905	0,7	0,9	0,7	0,9	0,702	0,908	0,689	0,886	0,689	0,885	1,784	2,552				
40	0,7	0,904	0,699	0,899	0,699	0,899	0,701	0,907	0,688	0,884	0,688	0,883	1,892	2,661				
50	0,701	0,904	0,7	0,899	0,7	0,899	0,701	0,907	0,688	0,884	0,688	0,882	1,963	2,686				
70	0,702	0,905	0,701	0,9	0,701	0,9	0,702	0,907	0,688	0,884	0,688	0,882	2,057	2,738				
100	0,704	0,907	0,702	0,902	0,702	0,902	0,704	0,908	0,69	0,885	0,689	0,883	2,141	2,792				
300	0,713	0,915	0,711	0,911	0,711	0,911	0,712	0,915	0,696	0,892	0,694	0,888	2,64	2,931				
500	0,717	0,918	0,715	0,914	0,715	0,915	0,716	0,918	0,699	0,895	0,696	0,891	2,926	3,01				

Tabela 15: Melhorias de performance da abordagem proposta nesse trabalho em relação os métodos de referência considerando as métricas MAE e o RMSE para a partição 4 da base de dados ML 1M. Os valores em negrito em cada linha dos métodos de similaridade global indicam a pior performance, enquanto os valores em negrito em cada linha dos métodos baseados em similaridade local indicam a melhor performance.

k	Similaridade global												Similaridade local					
	SG_COS		SG_PCC		SG_SPPC		SG_WPCC		SL_AP		SL_KM		Melhoria(%)					
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE				
10	0,712	0,92	0,711	0,916	0,711	0,916	0,712	0,923	0,703	0,905	0,704	0,905	1,248	1,952				
20	0,702	0,907	0,701	0,901	0,701	0,901	0,702	0,908	0,693	0,89	0,692	0,889	1,421	2,109				
30	0,7	0,903	0,699	0,898	0,699	0,898	0,7	0,904	0,69	0,886	0,69	0,885	1,506	2,177				
40	0,7	0,902	0,698	0,897	0,698	0,897	0,7	0,904	0,689	0,885	0,688	0,883	1,623	2,313				
50	0,7	0,902	0,698	0,897	0,698	0,897	0,7	0,904	0,689	0,884	0,688	0,882	1,679	2,371				
70	0,701	0,903	0,699	0,898	0,699	0,898	0,701	0,904	0,689	0,885	0,688	0,882	1,819	2,428				
100	0,703	0,905	0,701	0,9	0,701	0,9	0,703	0,906	0,691	0,886	0,689	0,883	1,928	2,487				
300	0,712	0,913	0,71	0,909	0,71	0,909	0,711	0,913	0,697	0,893	0,695	0,889	2,409	2,652				
500	0,716	0,916	0,714	0,913	0,714	0,913	0,714	0,916	0,7	0,896	0,697	0,891	2,684	2,759				

Tabela 16: Melhorias de performance da abordagem proposta nesse trabalho em relação os métodos de referência considerando as métricas MAE e o RMSE para a partição 1 da base de dados Netflix. Os valores em negrito em cada linha dos métodos de similaridade global indicam a pior performance, enquanto os valores em negrito em cada linha dos métodos baseados em similaridade local indicam a melhor performance.

k	Similaridade global												Similaridade local					
	SG_COS		SG_PCC		SG_SPPC		SG_WPCC		SL_AP		SL_KM		Melhoria(%)					
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE				
10	0,692	0,902	0,69	0,898	0,69	0,898	0,69	0,9	0,68	0,885	0,682	0,889	1,629	1,928				
20	0,681	0,887	0,679	0,883	0,679	0,883	0,679	0,885	0,669	0,87	0,672	0,874	1,752	1,928				
30	0,679	0,883	0,677	0,879	0,677	0,879	0,677	0,881	0,667	0,866	0,669	0,87	1,812	1,968				
40	0,679	0,882	0,677	0,879	0,677	0,879	0,677	0,881	0,666	0,864	0,668	0,869	1,897	2,044				
50	0,679	0,882	0,677	0,879	0,677	0,879	0,678	0,881	0,666	0,864	0,668	0,868	1,971	2,088				
70	0,681	0,883	0,679	0,88	0,679	0,88	0,679	0,881	0,666	0,864	0,669	0,868	2,15	2,211				
100	0,683	0,885	0,681	0,881	0,681	0,882	0,681	0,883	0,667	0,865	0,669	0,869	2,29	2,307				
300	0,693	0,894	0,69	0,891	0,69	0,891	0,691	0,892	0,672	0,869	0,675	0,874	3,01	2,759				
500	0,698	0,899	0,695	0,896	0,695	0,896	0,696	0,897	0,674	0,872	0,678	0,877	3,384	3,025				

Tabela 17: Melhorias de performance da abordagem proposta nesse trabalho em relação os métodos de referência considerando as métricas MAE e o RMSE para a partição 2 da base de dados Netflix. Os valores em negrito em cada linha dos métodos de similaridade global indicam a pior performance, enquanto os valores em negrito em cada linha dos métodos baseados em similaridade local indicam a melhor performance.

k	Similaridade global												Similaridade local					
	SG_COS		SG_PCC		SG_SPPC		SG_WPCC		SL_AP		SL_KM		Melhoria(%)					
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE				
10	0,693	0,905	0,691	0,901	0,691	0,901	0,691	0,903	0,68	0,887	0,684	0,892	1,773	1,986				
20	0,683	0,89	0,681	0,886	0,681	0,886	0,681	0,888	0,669	0,871	0,673	0,877	2,008	2,187				
30	0,681	0,887	0,679	0,883	0,679	0,883	0,679	0,885	0,666	0,867	0,67	0,872	2,148	2,31				
40	0,681	0,886	0,679	0,882	0,679	0,882	0,679	0,884	0,666	0,865	0,669	0,871	2,219	2,342				
50	0,681	0,886	0,679	0,882	0,679	0,882	0,679	0,884	0,665	0,865	0,669	0,871	2,326	2,415				
70	0,682	0,887	0,68	0,883	0,68	0,883	0,68	0,885	0,665	0,865	0,669	0,871	2,492	2,511				
100	0,685	0,889	0,682	0,885	0,682	0,885	0,683	0,887	0,667	0,866	0,67	0,872	2,629	2,584				
300	0,694	0,897	0,691	0,894	0,692	0,894	0,692	0,895	0,671	0,871	0,676	0,877	3,237	2,957				
500	0,699	0,902	0,696	0,899	0,696	0,899	0,697	0,9	0,674	0,873	0,679	0,88	3,625	3,233				

Tabela 18: Melhorias de performance da abordagem proposta nesse trabalho em relação os métodos de referência considerando as métricas MAE e o RMSE para a partição 3 da base de dados Netflix. Os valores em negrito em cada linha dos métodos de similaridade global indicam a pior performance, enquanto os valores em negrito em cada linha dos métodos baseados em similaridade local indicam a melhor performance.

k	Similaridade global												Similaridade local					
	SG_COS		SG_PCC		SG_SPPC		SG_WPCC		SL_AP		SL_KM		Melhoria(%)					
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE				
10	0,692	0,902	0,69	0,898	0,69	0,898	0,69	0,9	0,68	0,886	0,684	0,891	1,689	1,789				
20	0,682	0,888	0,68	0,883	0,68	0,883	0,679	0,885	0,669	0,87	0,672	0,875	1,9	2,006				
30	0,68	0,884	0,678	0,88	0,678	0,88	0,678	0,882	0,666	0,866	0,669	0,87	1,975	2,068				
40	0,679	0,883	0,678	0,879	0,678	0,879	0,678	0,881	0,665	0,864	0,669	0,869	2,069	2,126				
50	0,68	0,883	0,678	0,879	0,678	0,879	0,678	0,881	0,665	0,864	0,668	0,869	2,177	2,189				
70	0,681	0,884	0,679	0,88	0,679	0,88	0,679	0,882	0,665	0,864	0,669	0,869	2,28	2,256				
100	0,683	0,885	0,681	0,882	0,681	0,882	0,681	0,884	0,666	0,864	0,67	0,87	2,453	2,376				
300	0,693	0,894	0,69	0,891	0,69	0,891	0,691	0,892	0,671	0,869	0,675	0,874	3,084	2,765				
500	0,698	0,899	0,695	0,896	0,695	0,896	0,696	0,897	0,674	0,872	0,678	0,877	3,454	3,038				

Tabela 19: Melhorias de performance da abordagem proposta nesse trabalho em relação os métodos de referência considerando as métricas MAE e o RMSE para a partição 4 da base de dados Netflix. Os valores em negrito em cada linha dos métodos de similaridade global indicam a pior performance, enquanto os valores em negrito em cada linha dos métodos baseados em similaridade local indicam a melhor performance.

k	Similaridade global												Similaridade local					
	SG_COS		SG_PCC		SG_SPPC		SG_WPCC		SL_AP		SL_KM		Melhoria(%)					
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE				
10	0,692	0,903	0,691	0,899	0,691	0,899	0,69	0,901	0,68	0,886	0,683	0,89	1,78	1,865				
20	0,682	0,888	0,68	0,884	0,68	0,884	0,68	0,886	0,669	0,87	0,672	0,875	1,907	1,983				
30	0,679	0,884	0,678	0,881	0,678	0,881	0,678	0,883	0,666	0,866	0,669	0,87	1,939	2,032				
40	0,679	0,883	0,677	0,88	0,677	0,88	0,677	0,881	0,665	0,865	0,668	0,869	2,031	2,116				
50	0,679	0,883	0,678	0,879	0,677	0,879	0,678	0,881	0,665	0,864	0,668	0,869	2,092	2,142				
70	0,68	0,884	0,679	0,88	0,679	0,88	0,679	0,882	0,665	0,864	0,668	0,869	2,189	2,219				
100	0,682	0,885	0,681	0,882	0,681	0,882	0,681	0,884	0,666	0,865	0,669	0,87	2,352	2,322				
300	0,692	0,895	0,69	0,891	0,69	0,891	0,69	0,893	0,671	0,87	0,674	0,875	3,004	2,746				
500	0,697	0,899	0,694	0,896	0,694	0,896	0,695	0,897	0,673	0,872	0,677	0,877	3,337	2,991				