

Personalização da Experiência em Museus: Aplicação Real de um Sistema de Recomendação

Felipe Ferreira Laskoski

DISSERTAÇÃO DE MESTRADO APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
COMO REQUISITO PARA
OBTENÇÃO DO TÍTULO
DE
MESTRE EM CIÊNCIAS

Programa: Ciência da Computação
Orientador: Prof. Dr. Alfredo Goldman vel Lejbman

São Paulo, dezembro de 2019

Personalização da Experiência em Museus: Aplicação Real de um Sistema de Recomendação

Esta versão da dissertação contém as correções e alterações sugeridas pela Comissão Julgadora durante a defesa da versão original do trabalho, realizada em 17/12/2019. Uma cópia da versão original está disponível no Instituto de Matemática e Estatística da Universidade de São Paulo.

Comissão Julgadora:

- Prof. Dr. Alfredo Goldman vel Lejbman - IME-USP
- Prof^a. Dr^a. Alessandra Fernandes Bizerra - IB-USP
- Prof. Dr. Márcio Augusto de Souza - UEPG

Agradecimentos

Dedico este espaço a todas as pessoas que contribuíram direta ou indiretamente para a geração deste trabalho.

À minha família, especialmente aos meus pais Teodoro e Elizete, por todo o amor, sempre me apoiando, especialmente nos momentos de dificuldade, e à minha noiva Gisoon, por todo o seu amor, motivação e compreensão.

Ao meu orientador Prof. Dr. Alfredo Goldman, primeiramente pela confiança indescritível, e por todo o suporte, conselhos e críticas, sempre presente, me guiando ao longo de todos esses quase 3 anos de mestrado.

À prof. Dra. Alessandra Fernandes Bizerra, pela indiscutível receptividade, suporte, experiência e parceria na aplicação e execução desse trabalho.

À Cecília, pelo especial suporte, amizade, e excelente humor, no processo de coleta de dados da pesquisa.

Aos funcionários do Parque CienTec Ailton, Franco e Célia, e ao Thiago, Diego, Natali, Victoria, Junior, Isabela, Renato e demais monitores e monitoras, pela amizade e suporte na coleta de dados da pesquisa.

Aos meus amigos, pela amizade e motivação, deixando toda a trajetória mais fácil e feliz.

Resumo

FERREIRA LASKOSKI, F., **Personalização da Experiência em Museus: Aplicação Real de um Sistema de Recomendação**. 2019. 116 f. Dissertação (Mestrado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2019.

A rápida evolução tecnológica tem expandido constantemente as fronteiras das aplicações de diversas áreas na vida das pessoas. A aplicação de sistemas de recomendação é um exemplo que se beneficiou com essa evolução e que hoje é aplicado em muitos contextos novos, como em museus. Diversos estudos se propuseram a gerar um sistema de recomendação para enfrentar a típica dificuldade de sobrecarga de informação, que costuma ser experimentada pelos visitantes desses lugares. Mas esses sistemas, em sua maioria, não exploraram as capacidades tecnológicas hoje disponíveis ou não foram testados em ambientes reais de utilização. Esta pesquisa busca atender essas oportunidades com a concepção de um sistema de recomendação híbrido baseado em informações de contexto e no método de recomendação *collaborative filtering*. O sistema também é avaliado em ambiente produtivo com usuários reais, em vez de experimentos controlados. O sistema proposto gera rotas personalizadas de visita com o objetivo de maximizar a satisfação do usuário ao mesmo tempo em que minimiza a distância do percurso da visita.

Esta pesquisa realizou uma das maiores avaliações de um sistema de recomendação já divulgadas, contando com a participação de um número expressivo de visitantes reais em um museu de São Paulo. Seu desempenho foi aferido em relação à acurácia e satisfação do usuário. As avaliações foram feitas em duas etapas, sendo que a segunda foi realizada com um sistema com melhorias baseadas nos resultados da primeira etapa. Mesmo com todas as variabilidades naturais de um ambiente produtivo, os resultados indicaram que o sistema obteve altos níveis de acurácia e satisfação do usuário com as recomendações de itens e as rotas propostas. Verificou-se também uma tendência de melhora na acurácia das recomendações, tanto da primeira para a segunda etapa, quanto com o aumento da base de dados gerados pelos usuários.

Palavras-chave: Sistemas de recomendação, tecnologia em museus, *collaborative filtering*, contexto, cidades inteligentes, aplicativos.

Abstract

FERREIRA LASKOSKI, F., **Personalization of the Experience in Museums: Real Application of a Recommendation System**. 2019. 116 f. Dissertação (Mestrado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2019.

The rapid evolution of technology is constantly expanding the boundaries of applications from several fields on people's lives. The application of recommendation systems is an example that has benefited from that evolution and today is used in many new contexts, as in museums. Various studies proposed to build a recommendation system in order to deal with the typical information overload problem, which is used to be experienced by the museum visitors. However most of them did not explore the technological capacity available today or they were not been tested in real environments. This research aims to meet those opportunities by the creation of a hybrid context-based recommendation system with collaborative filtering. The system is also evaluated in the production environment with actual users, instead of a controlled experiment. The proposed system suggests personalized visiting routes with the goal to maximize the user satisfaction at the same time it minimizes the distance of the recommended route.

This research conducted one of the largest evaluations of recommendation systems ever released, counting with the participation of an expressive number of actual visitors in a museum in São Paulo. The system's performance was measured regarding accuracy and user satisfaction. The evaluations were conducted in two steps, where the second used an improved system based on the results obtained from the first step. Even with all the variability involved in a production environment, the overall results indicate the system achieved high accuracy levels and user satisfaction with the recommendations and proposed routes. The results also indicated a positive trend in terms of the recommendations accuracy, both from the first to the second step of the analysis as well as with the increase in the database built with the users interactions.

Keywords: Recommendation systems, technology in museums, collaborative filtering, context, smart cities, applications.

Sumário

Lista de Siglas	xi
Lista de Figuras	xiii
Lista de Tabelas	xv
1 Introdução	1
1.1 Motivação	1
1.2 Contexto da Pesquisa	2
1.3 Problema a Ser Estudado	3
1.4 Questão da Pesquisa	4
1.5 Organização do Texto	5
2 Sistemas de Recomendação	7
2.1 Definição e Contexto	7
2.2 Domínios para Aplicação de um SR	8
2.3 Experiência do Usuário	9
2.4 Etapas da Recomendação	9
2.5 Elementos do Sistema de Recomendação	10
2.5.1 Item	10
2.5.2 Usuário	10
2.5.3 Transação	11
2.6 Tipos de Sistemas de Recomendação	11
2.6.1 <i>Filtragem Baseada em Conteúdo</i>	11
2.6.2 Filtragem Colaborativa	12
2.6.3 Sistemas Demográficos	14
2.6.4 Baseado em Conhecimento	14
2.6.5 Sistemas Híbridos	15
2.6.6 Sistemas Baseado em Contexto	16
2.7 Avaliação de um Sistema	19
2.7.1 Tipos de Avaliação	20
2.7.2 Critérios de Avaliação	21

2.7.3	Validação	25
2.8	Desafios na Aplicação de SR em Museus	26
2.8.1	Desafios específicos para SRs aplicados em Museus	26
2.8.2	Desafios de SRs em Geral	29
3	Trabalhos Relacionados	31
3.1	SR Baseado na Ontologia CIDOC-CRM	32
3.2	ART <i>Recommender</i>	32
3.3	SR Aplicado a Pontos Turísticos	33
3.4	SR para Grupos	33
3.5	PGR	34
3.6	Aplicando PLN	35
3.7	Contribuição da Pesquisa	35
4	A Aplicação: Guia CienTec	37
4.1	Caracterização do Problema	37
4.2	Desafios Relacionados	38
4.2.1	Desafios para SRs aplicados em Museus	38
4.2.2	Desafios para SRs em Geral	40
4.3	Decisões de Projeto	41
4.4	O Sistema de Recomendação	47
4.4.1	Arquitetura do SR	47
4.4.2	Etapa 1: Modelagem Contextual com Filtragem Colaborativa	48
4.4.3	Etapa 2: Pós-filtragem Contextual	50
4.5	Detalhes do Aplicativo	55
4.5.1	Fluxo de Uso	56
4.5.2	Funções Adicionais	58
5	Coleta e Análise dos Resultados	61
5.1	Informações Gerais da Coleta dos Dados	61
5.2	Metodologia Utilizada	63
5.2.1	Metodologia da Coleta de Dados	63
5.2.2	Metodologia da Avaliação do SR	63
5.3	Fase 1: Formação da Base Inicial de Recomendações	64
5.4	Fase 2: Primeira Avaliação <i>On-line</i>	65
5.4.1	Acurácia	65
5.4.2	Satisfação do Usuário	68
5.4.3	Análise de Desempenho e Escalabilidade	69
5.5	Fase 3: Avaliação <i>Off-line</i> com Múltiplas Variações do SR	72
5.5.1	Comparação de Acurácia	73
5.5.2	Comparação do Tempo de Treinamento	74

5.6	Fase 4: Segunda Avaliação On-line	75
5.6.1	Acurácia	76
5.6.2	Satisfação do Usuário	79
5.7	Riscos à Validade dos Resultados	80
6	Conclusões	81
	Referências Bibliográficas	85
A	Definições e Algoritmos	91
A.1	Funções de Semelhança	91
A.2	Complexidade NP-Difícil	92
A.3	<i>Algoritmo de Dijkstra</i>	92
A.4	<i>Greedy Nearest Neighbors Search</i>	93

Lista de Siglas

SR	Sistema de Recomendação
CBF	Filtragem baseada em conteúdo (<i>Content-Based Filtering</i>)
CF	Filtragem colaborativa (<i>Collaborative Filtering</i>)
kNN	k vizinhos mais próximos (<i>k Nearest Neighbors</i>)
PCC	Coefficiente da Correlação de Pearson (<i>Pearson Correlation Coefficient</i>)
KB	Sistema baseado em conhecimento (<i>Knowledge-Based System</i>)
BC	Sistema baseado em contexto (<i>Context-Based System</i>)
MAE	Erro absoluto médio (<i>Mean Absolute Error</i>)
RMSE	Raiz do erro quadrático médio (<i>Root-Mean-Square Error</i>)
ROC	<i>Receiver Operation Characteristics</i>
MF	<i>Matrix Factorization</i>

Lista de Figuras

2.2	Exemplos de aplicação de recomendações baseadas na localização do usuário.	18
2.3	Exemplo de comparação de acurácia de três classificadores com uma curva ROC.	22
4.1	Principais Componentes do Sistema de Recomendação.	47
4.2	Tipos de Elementos do Sistema	51
4.3	Exemplo de uso de elementos do tipo ponto e visão das coordenadas geográficas mapeadas	52
4.4	Telas do aplicativo para entrada de informações do usuário e próximo item da rota	56
4.5	Mapa do parque construído para o aplicativo	57
4.6	Telas de detalhes dos elementos de conteúdo e avaliação de satisfação com a visita	58
4.7	Telas com lista de itens disponíveis e menu de opções	59
5.1	Gráfico com a distribuição de usuários pelo total de avaliações de elementos de conteúdo feitas.	63
5.2	Valores de MAE e RMSE do SR do aplicativo e aleatório para grupos de 10 avaliações, ordenados por data	66
5.3	Gráfico do MAE e RMSE por grupo de avaliações feitas na mesma posição cronológica pelos usuários. As colunas representam o número de avaliações consideradas naquela posição.	67
5.4	Satisfação do usuário com as diferentes características do sistema.	69
5.5	Evolução no tempo de processamento com o aumento das avaliações	70
5.6	Evolução no tempo de processamento para gerar a rota recomendada com o aumento da quantidade de itens e subitens	71
5.7	RMSE das 12 variações do SR com a base de 113 avaliações da fase 2.	74
5.8	Média de tempo de execução das 12 implementações do SR por grupos de 100 treinamentos, com acréscimo de avaliações	75
5.9	Comparação das métricas MAE e RMSE entre o SR do aplicativo e o aleatório, e em relação ao sistema aplicado na fase 2 da pesquisa	76

5.10	Comparação do MAE por grupo de avaliações feitas na mesma posição cronológica pelos usuários, entre o sistema da aplicação e o aleatório, e com o da fase 2 da pesquisa.	78
5.11	Satisfação do usuário com as diferentes características do sistema na fase 2, 4 e no geral.	79

Lista de Tabelas

2.1	Resumo das principais características dos tipos de sistemas de recomendação.	19
3.1	Tabela comparativa deste trabalho com os artigos relacionados em relação aos algoritmos utilizados, número de participantes na avaliação e tipo de avaliação, segundo a classificação em 2.7.	35
5.1	Distribuição dos usuários considerados na pesquisa por faixa etária e se já haviam visitado o parque.	62
5.2	Estatísticas de avaliações dos monitores e funcionários participantes.	64
5.3	Estatísticas de avaliação dos visitantes da primeira fase.	65
5.4	Acurácia das doze variações do SR no teste <i>10-fold cross validation</i>	72
5.5	Acurácia das doze variações do SR no teste de simulação de uso do sistema. .	75
5.6	Estatísticas de avaliações dos participantes da segunda avaliação <i>on-line</i> . . .	76

Lista de Algoritmos

- 1 Método kNN baseado em usuário 50
- 2 Função principal de obtenção da rota recomendada 54
- 3 Funções auxiliares de obtenção da rota recomendada 55

Capítulo 1

Introdução

1.1 Motivação

O uso de aparelhos celulares se tornou mundialmente acessível, tendo, por exemplo, em 2017 no Brasil ultrapassado a marca de um celular por habitante [Lima]. Lado a lado com a popularização desse dispositivo veio também a evolução dos modelos que, hoje, possuem uma numerosa variedade de sensores e funções, muitas delas gerando dados continuamente. Com o comportamento de uso e interações do usuário com os celulares, é possível aprender muito sobre as características deles. Por esse motivo, nos últimos anos, muitas aplicações tentam utilizar esses dados das mais diversas formas, produzindo informações como conteúdo personalizado, otimização de atividades do usuário, estudos de mercado, entre outras.

Outro exemplo de aplicação que pode se beneficiar desses dados são sistemas de recomendação de conteúdo. Sistemas de recomendação são ferramentas de software que provêm sugestões de opções para o usuário em um processo de tomada de decisão [Ricci et al., 2010], geralmente em forma de uma lista ordenada dos itens do mais para o menos recomendado. Esses sistemas têm sido usados em muitas áreas de atuação, como em conteúdos de multimídia [Fernández, 2018][Lin et al., 2013], turismo [Khan et al., 2017], *e-commerces* [Aggarwal, 2016], etc. Geralmente sistemas desse tipo são mais úteis em ambientes onde o usuário não tem muito conhecimento sobre o assunto ou quando há uma grande quantidade de opções para escolher [Ricci et al., 2010], e ele não possui critérios suficientes para filtrá-los. Nesse caso, torna-se uma tarefa muito difícil decidir o que fazer. Esse tipo de problema é conhecido como sobrecarga de informação [Xiang et al., 2013] [Ricci et al., 2010].

Um dos exemplos mais famosos de aplicação de sistemas de recomendação para atender esse problema é o caso da plataforma do *Netflix*. Dada a extensa quantidade de filmes, documentários e outros tipos de vídeos disponíveis em sua plataforma, acaba sendo difícil para o usuário encontrar as opções que realmente irão lhe interessar. Se os usuários tiverem dificuldades em encontrar conteúdo de interesse, podem acabar deixando de usar os serviços da empresa. Então poder contar com sugestões assertivas de escolha são vitais para o sucesso desse negócio [Fernández, 2018]. Sabendo disso, uma das medidas adotadas pela empresa

foi a promoção de um evento, chamado de *Netflix Prize*, onde foi oferecido um prêmio de 1 milhão de dólares a quem apresentasse um sistema de recomendação que melhorasse a acurácia do sistema deles em 10% [Pavlidis, 2019b].

Embora seja um contexto bem diferente, a visitação a museus e outros locais de exposição de patrimônio cultural também é uma situação que apresenta os problemas relacionados a esses sistemas. Nesses lugares frequentemente há uma numerosa quantidade de obras à disposição para o usuário visitar, dentro de uma restrição de tempo imposta por ele. Assim, o usuário precisa invariavelmente tomar decisões de qual obras (itens) são prioritárias e deverão ser visitadas. Junto a isso, pode-se considerar que é frequente o desconhecimento do usuário sobre os conteúdos dos itens lá disponíveis.

1.2 Contexto da Pesquisa

Soluções de recomendação para museus ajudam a otimizar a difusão de conhecimento e cultura, aumentando o que podemos chamar de inteligência coletiva dos habitantes de uma cidade, além de trazer mais eficiência a um tipo de atividade do dia-a-dia dessas pessoas. Com esses benefícios, a presente pesquisa pode ser considerada dentro do conceito chamado de cidades inteligentes. Esse conceito não tem um significado bem definido e já foi utilizado de diversas formas na literatura [Chourabi et al., 2012], mas todos tem como base a busca por eficiência em aspectos da vida nas cidades através da aplicação de sistemas de TI.

Há diversos aspectos relevantes conhecidos que podem ajudar os sistemas de recomendação a prever o nível de interesse dos usuários sobre os itens disponíveis para escolha e produzir resultados úteis para eles. Os dois grupos de técnicas de recomendação mais conhecidos e utilizados são os chamados Filtragem Colaborativa e Filtragem Baseada em Conteúdo [Lin et al., 2013].

Filtragem colaborativa se baseia na premissa de que um usuário toma suas decisões de escolha com forte influência das recomendações provenientes de outros usuários com gostos similares ao seu [Ricci et al., 2010]. Escolher um candidato a uma vaga na empresa de acordo com recomendações de colegas e assistir um filme porque seu amigo gostou muito e lhe indicou são dois bons exemplos desse comportamento na vida real. O exemplo do filme também é bom para explicar a importância de recomendar de acordo com usuários com preferências similares, visto que é muito comum a situação de um usuário não ter gostado de um filme que seu amigo gostou muito. Considerações desse tipo devem ser tratadas de acordo com o contexto que o sistema de recomendação está envolvido, pois, pode afetar significativamente o seu desempenho. A outra forma usual de se recomendar itens é através da filtragem baseada em conteúdo, onde a correlação entre o conteúdo e/ou características dos itens, e a relação deles com as preferências do usuário são consideradas. Cada um desses métodos possui vantagens e desvantagens, e serão abordados com mais detalhes, juntamente com outros métodos, no Capítulo 2.

Sistemas de recomendação modernos aplicados às mais variadas situações vêm sendo propostos desde o final do século 20. Porém, sistemas aplicados no domínio de patrimônios culturais, mais proeminentemente relacionados ao aperfeiçoamento da visita a museus, começaram a surgir um pouco mais tarde, no começo deste século [Pavlidis, 2019b]. Mesmo assim, há um número relevante de aplicações nesse domínio na literatura [Wang et al., 2009] [Bright et al., 2005] [Benouaret and Lenne, 2015] [Oppermann and Specht, 1999], geralmente divergindo nas técnicas utilizadas. Contudo, poucos desses trabalhos propõem sistemas que utilizem as ricas possibilidades de geração de dados de contexto dos usuários que os celulares podem oferecer. Isso ocorre, em parte, porque muitos dos trabalhos encontrados, foram produzidos na primeira década do século 21. Fato esse que pode ter inviabilizado o uso dos dados e funções dos celulares que temos hoje disponíveis, considerando que é relativamente recente a facilidade de encontrar algumas das características mais importantes dos celulares nos aparelhos das pessoas em geral, como telas sensíveis ao toque, que trazem maior interatividade do usuário com os aplicativos, sensores de GPS precisos e processadores mais potentes. A disponibilidade de processadores mais potentes facilitou funções como a revisão em tempo real de recomendações baseada em *feedbacks* do usuário.

Além dos aspectos técnicos da solução, o contexto do público alvo da aplicação também precisa ser levado em consideração. Uma pesquisa feita no estado de São Paulo pelo Instituto Datafolha em conjunto com a JLeiva Cultura & Esporte [Leiva, 2014] indica a predominância de indivíduos das classes A e B na visita a museus. 35% dos respondentes dessas classes disseram ter ido ao menos uma vez no último ano a um museu e apenas 8% dos indivíduos das classes D e E disseram o mesmo. Em relação a escolaridade, há maior concentração de indivíduos com ensino superior em relação às demais escolaridades, com 48% e 14%, respectivamente. O interesse e efetiva visitação a museus no Brasil é inversamente proporcional a idade, conforme resultados da pesquisa, tendo 33% das pessoas entre 12 e 24 anos visitado ao menos um museu no último ano, contra 17% da faixa etária de 60 anos ou mais. Importante também a indicação presente em [Chelini and Lopes, 2008] de forte inclinação por parte de visitantes espontâneos de museus pela exploração visual do ambiente, entre outras experiências sensoriais, com preferência por objetos que se movem, que podem ser sentidos e também por elementos inusitados ou recentemente adicionados.

1.3 Problema a Ser Estudado

Além de geralmente haver muitos itens disponíveis em um museu e o usuário não possuir conhecimento suficiente para escolher com eficiência quais ele deseja visitar, existem outras restrições como tempo máximo que o usuário tem para a visita e distância entre os itens disponíveis. Sistemas de recomendação buscam tratar essas restrições, mas também geram outras dificuldades como a recomendação de itens para visitantes em grupo e a coleta de *feedback* que costuma ser obtida pela ação explícita do usuário, o que nem sempre ocorre e

pode inviabilizar alguns experimentos.

Ao produzir um sistema de recomendação, é importante verificar outros tipos de informações, além do *feedback* do usuário, que influenciam na sua decisão e podem ser usados para estabelecer a nota de recomendação dos itens. Alguns exemplos que podemos citar de informações que podem ser usadas para complementar a recomendação de itens são os relacionamentos do usuário [Mo et al., 2014], comportamento do usuário em redes sociais [Both Suzy Edith and Yu, 2018], sua posição geográfica [Zhu et al., 2017] e dados mais completos do contexto do usuário gerados, por exemplo, através da Internet das Coisas [Salman et al., 2015].

1.4 Questão da Pesquisa

O presente trabalho busca aplicar e verificar o desempenho de um sistema de recomendação para museus, no que diz respeito à acurácia das recomendações e satisfação do usuário. Faz isso visando a personalização da experiência de visita do usuário, atendendo as dificuldades citadas na seção anterior, como falta de conhecimento do usuário e sobrecarga de informação e baseando-se não só em suas avaliações, mas em seu comportamento e contexto.

Desse modo, a questão da pesquisa que pretendemos responder é: **Qual desempenho que um sistema de recomendação híbrido baseado em contexto** (ver Seção 2.6.5 para mais informações) **consegue obter, em termos de acurácia e satisfação do usuário, quando aplicado no auxílio a visita a museus?**. Por acurácia, considera-se o grau de proximidade entre uma estimativa e seu valor verdadeiro [Monico et al., 2009].

Como esse contexto de aplicação implica em recomendação de itens que possivelmente estão fisicamente distantes, há também a necessidade de indicar ao usuário como chegar até eles. Por esse motivo a pesquisa também propõe uma solução para a recomendação da rota entre os itens recomendados e avalia seu desempenho pela média de satisfação do usuário.

O sistema de recomendação proposto, relacionado à questão da pesquisa, deve considerar insumos provenientes de *feedbacks* dos usuários sobre os itens disponíveis, além de outras informações úteis para a definição da recomendação, como: dados de comportamento do usuário com o celular, como a forma como ele navega pelas telas; sensores do celular como GPS; *feedbacks* que ele provê para os conteúdos visitados ao longo da visita e outras informações de interesse do usuário informadas diretamente por ele. Utilizar todas essas informações em conjunto de forma harmoniosa para gerar resultados coerentes não é trivial e, dependendo da solução definida, pode exigir um grande custo de processamento, especialmente quando as recomendações são refinadas à medida que novos *feedbacks* do usuário são obtidos. Por essa razão adequações são necessárias para que tal sistema seja viável, tanto no sentido de tempo de processamento como de experiência do usuário. Na Seção 4.3 serão apresentadas as limitações e adequações encontradas para viabilizar a aplicação do sistema.

Este trabalho teve o objetivo de avançar na pesquisa de otimização da personalização de

rotas de visitação através de experimentação da aplicação de tal sistema em um ambiente real, o Parque CienTec da USP, em São Paulo - SP. Como nesse local os itens estão fisicamente distantes, é essencial que uma rota entre os itens seja apresentada ao usuário e que ela seja otimizada para diminuição do esforço necessário para percorrê-la.

A coleta de dados da avaliação *on-line* proposta para validar o desempenho do sistema se dividiu em dois momentos. No primeiro foi testado o sistema com uma configuração de parâmetros específica com o objetivo de montar uma base de dados, testar o aplicativo como um todo e validar se o desempenho está satisfatório. Com o atingimento desse objetivo, a base histórica de dados foi utilizada para verificar o desempenho de diferentes implementações do sistema de recomendação ao variar parâmetros de sua configuração, a fim de encontrar a implementação com a melhor combinação de parâmetros. A melhor implementação foi então utilizada na segunda parte da avaliação *on-line* para verificar se haveria um ganho de desempenho significativo.

1.5 Organização do Texto

O trabalho está dividido da seguinte forma: No Capítulo 2 são introduzidos e discutidos as diversas características envolvidas em um sistema de recomendação, como avaliar seu desempenho e também quais os desafios relacionados à aplicação de um sistema desses em locais de exposição de patrimônio cultural. No Capítulo 3 alguns trabalhos que empregam sistemas de recomendação no contexto desses locais de aplicação são apresentados. Em seguida, no Capítulo 4, o sistema utilizado para a aplicação do sistema de recomendação e coleta de informações do usuários é demonstrado. Os algoritmos e ferramentas utilizadas, sua arquitetura, além de dificuldades e decisões tomadas para sua concepção, são expostas nesse capítulo. No Capítulo 5, a coleta e análise dos resultados encontrados na pesquisa serão apresentados em termos de acurácia das recomendações, custo de processamento e satisfação do usuário. Finalmente, no Capítulo 6, são trazidas as conclusões da pesquisa e possibilidades de pesquisas futuras relacionadas.

Capítulo 2

Sistemas de Recomendação

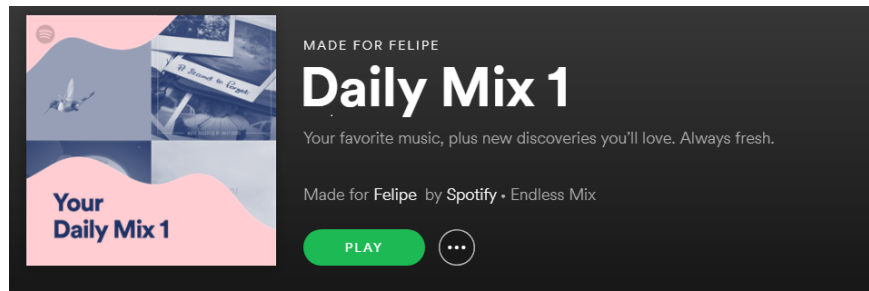
Esse capítulo abordará os principais aspectos ligados a sistemas de recomendação. Na primeira seção será apresentada a definição de um sistema de recomendação e sua motivação de uso. Na Seção 2.2 serão citados os domínios possíveis para aplicar um SR. Na Seção 2.3 são abordadas algumas características que devem ser levadas em consideração para que haja uma boa experiência do usuário da aplicação. Em 2.4 são apresentadas as etapas que constituem um SR e em 2.5 são explicados os elementos que o compõe. Em seguida, em 2.6, os principais tipos de SRs, suas características, vantagens e desvantagens são discutidas. Na Seção 2.7 são definidas as formas usuais e principais atributos utilizados para avaliar o desempenho de um sistema de recomendação. Por fim, em 2.8 são citados os principais desafios na aplicação de um SR.

2.1 Definição e Contexto

Sistemas de Recomendação (SR) são ferramentas que apoiam a decisão de um usuário sobre alguma escolha que ele pretenda fazer. Eles costumam atingir esse objetivo produzindo como saída uma lista ordenada de itens, em ordem decrescente, do mais recomendado ao menos recomendado. Eles são bem utilizados em situações onde haja sobrecarga de informação, isto é, quantidade de informação maior do que a capacidade do usuário de processá-la, e onde o usuário não tenha muito conhecimento sobre os itens a serem escolhidos.

O uso de SRs se intensificou nos últimos tempos, fortemente ligado ao incrível aumento da produção de dados mundial, puxado pelo crescimento do uso de PCs, *notebooks* e, em especial, dispositivos móveis. Além disso, diversos acessórios e eletrodomésticos, que antes não possuíam qualquer funcionalidade eletrônica, agora, influenciados inclusive pelo conceito de Internet das Coisas, geram todo tipo de informação. Indivíduos, que antes eram apenas consumidores de conteúdo, hoje se posicionam como produtores de seus próprios conteúdos [Araújo, 2011], através de serviços como Youtube, Wikipedia e redes sociais.

Auxiliando nessa tarefa, os SRs trazem diversos benefícios para os usuários, como agilidade para encontrar os itens desejados, aumentando a satisfação, e maior entendimento



(a) Recomendação de músicas do Spotify.



(b) Recomendação de filmes do Netflix.

dos itens. Também trazem benefícios para os donos dos produtos, como a capacidade de vender uma variedade maior de itens ou obter um aumento na quantidade total de vendas. Esse último provavelmente é o benefício mais importante para os sistemas de recomendação comerciais [Ricci et al., 2010][Aggarwal, 2016].

2.2 Domínios para Aplicação de um SR

O uso de Sistemas de recomendação para fins comerciais se tornou muito comum, com exemplos de grandes empresas, como o Spotify (figura 2.1a), Netflix (figura 2.1b), Youtube, Google, Amazon, etc. Pela característica de possuir com frequência as dificuldades de sobrecarga de informação e falta de conhecimento do cliente, as áreas de mídias digitais (músicas, filmes, etc) e comércio eletrônico são as que mais empregam esse tipo de sistema. Mas os SRs podem ser aplicados em diversos outros contextos, com exemplos disso disponíveis na literatura.

Abaixo são listados os domínios que segundo [Ricci et al., 2010] costumam utilizar sistemas de recomendação:

- Entretenimento: mídias digitais em geral, como serviços de *streaming* de filmes e TV.
- Conteúdo: Documentos, páginas *web*, *e-mails*, cursos recomendados, entre outros conteúdos.
- *E-Commerce*: Recomendação de produtos para comprar ou alugar.
- Serviços: Serviços de viagem, consultoria, relacionamentos, etc.

2.3 Experiência do Usuário

A correta identificação do contexto e objetivo da aplicação do SR é crucial para o seu sucesso, pois essas características determinam como deve ser a experiência de uso do sistema. Dependendo dessas características, variações no resultado gerado pelo sistema são aplicadas. Para exemplificar, em alguns casos, é mais adequado mostrar somente os 10 itens mais recomendados, em outros, pode ser mais vantajoso escolher automaticamente o item mais recomendado para o usuário, diminuindo assim a quantidade de passos que o sistema obriga o usuário a cumprir e colaborando para a fluência da experiência. Essa situação pode ser inviável em sistemas de maior risco, onde uma escolha subótima para o usuário pode acarretar consequências muito negativas.

Outra característica que deve ser estudada em relação à experiência do usuário é quais informações serão apresentadas no resultado. Às vezes é interessante divulgar dados que influenciaram a recomendação dos itens, como a nota geral de prioridade, nota média das avaliações dos usuários ou o nome dos amigos do usuário que recomendaram aquele item. Isso pode ajudar a trazer mais confiança do usuário sobre o sistema. Confiança é essencial para o sucesso da aplicação do SR, pois se os usuários não acreditarem que o sistema está realmente recomendando os melhores itens, ele poderá ignorá-lo e tentar manualmente encontrar as opções que deseja. Por isso confiança do usuário é um dos atributos que podem ser considerados para medir o desempenho de um sistema de recomendação. Vários outros atributos para a avaliação de um SR estão explicados em detalhes na Seção 2.7.

Existem também situações onde a divulgação dos dados que influenciaram a nota não é recomendada. Isso pode acontecer quando não se acredita que o usuário irá se beneficiar daqueles dados ou aumentar sua confiança sobre o sistema, o que ajudaria a diminuir a quantidade de informações apresentadas, tornando a visualização do conteúdo da aplicação mais limpa e destacando conteúdo de maior relevância para o usuário. Existem também situações que, caso sejam divulgados dados utilizados para a recomendação, podem permitir que usuários manipulem o sistema, alterando os resultados fornecidos para outros usuários.

2.4 Etapas da Recomendação

SRs podem ser entendidos como um subconjunto da área de Mineração de Dados [Ricci et al., 2010]. O processo de mineração de dados pode ser dividido em três fases, que também se aplicam a sistemas de recomendação. São elas: Pré-processamento, análise de dados e teste e validação.

O pré-processamento, como o nome indica, é realizado antes da utilização de fato dos dados para gerar o resultado do sistema. Essa etapa é necessária em quase todos os casos reais porque dificilmente os dados de entrada estão com a qualidade, quantidade, padrão e nível de detalhe pretendidos. Técnicas comumente utilizadas para tratar os dados e deixá-los com as melhores características para o processamento são amostragem, redução de dimensionalidade

e redução de ruído.

Com os dados tratados, pode-se passar para a fase de análise, onde efetivamente, no caso de SRs, é encontrada a nota de recomendação dos itens. Para isso, podem ser utilizadas técnicas de classificação, clusterização e associação. Após obter os resultados das técnicas, eles podem ser utilizados para testá-los e validá-los. A Seção 2.7 apresenta em detalhes critérios para isso.

2.5 Elementos do Sistema de Recomendação

Já temos conhecimento de diversos tipos de sistemas de recomendação, dos mais simples aos mais elaborados. Essa diferença na complexidade pode ser relacionada aos algoritmos utilizados, mas também aos dados considerados. De modo geral, os dados de entrada do SR vêm de três fontes diferentes: item, usuário e transação. O objetivo de um SR é de recomendar um ou mais itens para os usuários, possivelmente, através da observação de suas transações. A seguir mais detalhes sobre cada elemento de um SR.

2.5.1 Item

Representa o produto sendo recomendado. Exemplos: filmes, músicas, produtos à venda, imóveis, ofertas de serviços, opções de investimentos, opções de viagens, notícias e outras informações, etc.

Itens possuem atributos. Esses atributos podem ser usados como parâmetros para o sistema. Um tipo de item pode possuir muitos atributos mas nem todos serem utilizados pelo sistema. As informações dos itens juntamente com outras fontes de dados do sistema podem ser utilizados para gerar um valor de prioridade. Itens sempre tem um ou mais valores associados, que podem ser utilizados separadamente ou combinados para definir sua posição na lista de recomendação. Esse valores devem considerar a sua utilidade para o usuário e o custo. Itens que tem muita utilidade para o usuário mas também tem muito custo podem ser menos recomendados que itens com baixo custo e utilidade. Isso tudo, porém, depende do contexto de aplicação do SR.

Custo pode ser o tempo necessário estimado para que o usuário consiga utilizar o item, ou mesmo, custo monetário real, como nos casos onde itens são opções de investimento ou de seguro. Em outros casos o custo pode ser fixo, e o sistema deve maximizar o valor para o usuário dentro da restrição de custo. Um trabalho com essa característica é explicado na Seção 3.4.

2.5.2 Usuário

Da mesma forma que para os itens, o usuário possui um número variável de atributos e a escolha de quais atributos devem ser utilizados depende do contexto do sistema e dos algo-

ritmos utilizados. Pode-se utilizar informações demográficas, como idade, altura e profissão, assim como localização, preferências e necessidades. O conjunto de dados do usuário forma o modelo do usuário [Ricci et al., 2010].

2.5.3 Transação

Transações são informações geradas pela interação entre o usuário e o sistema. A análise das transações pode determinar características e padrões de comportamento do usuário, que costumam ser relevantes para influenciar uma recomendação. Por isso, é interessante guardar essas informações para também utilizá-las como entrada no SR.

A forma mais popular de transação é o *feedback* do usuário [Ricci et al., 2010]. Ele pode ser explícito ou implícito. *Feedbacks* explícitos são realizados diretamente pelo usuário, como dar uma nota para um item que ele consumiu ou marcá-lo como favorito. Esse tipo de *feedback* tende a implicar na formação de uma matriz esparsa, pois geralmente o usuário fornece sua opinião sobre um número reduzido de itens em comparação ao total disponível [Bell et al., 2009]. Por outro lado, *feedback* implícito é derivado do comportamento do usuário [Aggarwal, 2016], como o tempo do usuário gasto em determinado item, elementos clicados em uma tela, histórico de compras ou até mesmo o movimento do mouse.

2.6 Tipos de Sistemas de Recomendação

2.6.1 Filtragem Baseada em Conteúdo

Sistema do tipo filtragem baseada em conteúdo (CBF - do inglês *Content-Based Filtering*) faz a recomendação de itens com características parecidas a itens que obtiveram um *feedback* positivo do usuário [Araújo, 2011] [Ricci et al., 2010]. Ele calcula isso baseando-se principalmente em informações de duas entidades, os atributos (gênero do filme, ano de lançamento, atores participantes, etc) e/ou conteúdos dos itens (utilização de instrumento musical específico durante a música, palavras específicas em um texto, etc) e o perfil do usuário. O perfil do usuário é formado por características dos itens que ele deu *feedback*, seja de forma implícita ou explícita. Pode-se gravar no perfil que o usuário não gosta de determinado atributo porque deu *feedbacks* ruins para itens com aquelas características. O mesmo vale para atributos de itens que ele gostou, adicionando no perfil do usuário que ele se interessa por esses atributos.

Tendo o perfil do usuário e os atributos ou conteúdos dos itens disponíveis para escolha, o SR irá se basear na correlação entre essas duas informações para calcular a recomendação [Van Meteren, 2000]. Essa forma do CBF de calcular o resultado tem uma vantagem importante: mesmo que um item seja novo, e por isso não tenha ainda sido avaliado por nenhum usuário, ele pode ser recomendado [Aggarwal, 2016]. Isso acontece porque, se há outros itens com características similares que já foram avaliados, os perfis dos usuários já

vão estar formados, inclusive com os atributos daquele novo item. Isso é tudo que o sistema precisa para recomendar o item. Outra vantagem do método é que sua recomendação é transparente, isto é, pode-se com facilidade identificar para o usuário porque aquele item está sendo recomendado, ao mostrar o item similar que ele gostou.

Utilizar apenas o perfil do usuário e informações dos itens para recomendações, embora traga esses benefícios acima, também introduz uma série de desvantagens:

- Usuários novos não tem um perfil pronto, ou seja, o sistema não sabe os atributos que ele gosta e não gosta. Por isso, a assertividade das recomendações tende a ser bem baixa nesse caso.
- As recomendações tendem a ser óbvias [Aggarwal, 2016]. Por exemplo, se o usuário gostou de um filme específico que faça parte de uma série, toda a série tende a ser recomendada. O contrário também ocorre, ou seja, caso um novo item seja adicionado, e que seja diferente de todos os outros presentes na base, ele não terá chances de ser recomendado. Essa característica reduz a diversidade das recomendações [Araújo, 2011].
- Se a obtenção dos atributos dos itens não pode ser feita de forma automática, pode ser necessário muito esforço tornando-se, em alguns casos, inviável formar a base de atributos manualmente, dada a quantidade de itens na base.
- Dependendo do contexto, pode ser difícil identificar atributos dos itens que diferenciem uns dos outros [Ricci et al., 2010]. Nesse caso as recomendações podem se confundir, diminuindo a assertividade do sistema.

2.6.2 Filtragem Colaborativa

A filtragem colaborativa (CF - do inglês *Collaborative Filtering*) se baseia na ideia de que se dois usuários costumam avaliar os mesmos itens de forma similar, eles devem continuar a ter essa similaridade na avaliação dos próximos itens [Fernández, 2018], pois considera-se que eles tem preferências em comum. Essa mecânica de recomendação é interessante porque não necessita de informações dos atributos dos itens, ao contrário do CBF.

Há duas abordagens comumente usadas da filtragem colaborativa:

Memory-Based

Esta técnica, também conhecida como *Neighborhood-based algorithm*, faz a predição das avaliações de itens pelo usuário com base nos seus vizinhos [Aggarwal, 2016]. Esses vizinhos podem ser definidos como baseado em usuário ou baseado em item.

No algoritmo baseado em usuário, uma matriz relacionando todos os usuários e todos os itens é gerada. Nessa matriz são armazenadas as avaliações feitas pelos usuários sobre os itens que eles utilizaram. Usuários são então comparados e os k usuários mais parecidos entre

si (técnica *k-nearest neighbors* - kNN) são considerados. Em seguida, são desconsiderados os itens já avaliados pelo usuário, e sobre os demais itens é gerada uma nota de avaliação prevista que o usuário iria fornecer, baseando-se nas notas dos k usuários similares a ele [Fernández, 2018]. Ordenando essa lista de itens da maior avaliação prevista para a menor, obtemos uma lista de itens de recomendação, de onde pode-se extrair os N primeiros.

No caso da abordagem CF baseado em item, a previsão de nota de avaliação, ou preferência do usuário, por um item é determinada pela nota de itens similares. As formas mais utilizadas de verificar a semelhança entre dois itens é pela comparação entre as notas de todos os usuários que os avaliaram, através da medida do cosseno do ângulo que forma entre os vetores (de avaliações de usuários) dos dois itens [Linden et al., 2003] ou o coeficiente da correlação de Pearson (PCC - do inglês *Pearson Correlation Coefficient*). Ambos são apresentados na Seção A.1 do Apêndice A.

Sistemas CF costumam produzir bons resultados, são fáceis de implementar e de explicar ao usuário quais critérios que influenciaram a recomendação [Aggarwal, 2016], além de serem muito favoráveis a melhorias personalizadas [Ricci et al., 2010]. Além disso, com o método CF, o problema da superespecialização na recomendação, percebido em sistemas baseados em CBF, é superado. Isto é, não haverá o risco de ter apenas recomendações triviais de itens muito similares entre si [Araújo, 2011].

Por outro lado, algumas dificuldades aparecem. Em um contexto onde há muitos itens e usuários, a matriz de relação entre essas duas entidades costuma ser muito esparsa. Isso pode reduzir a acurácia na recomendação de itens pouco avaliados pelos usuários em geral [Fernández, 2018]. Também pode haver problemas de escalabilidade, sendo que os algoritmos envolvidos em CF são computacionalmente pesados, crescendo com o número de usuários e itens no sistema. Outra dificuldade nesse método, que também é percebida na abordagem CBF é o *cold start*, isto é, o desempenho do sistema em itens e usuários recém-criados [Ricci et al., 2010] (mais informações em 2.8.2).

Model-Based

Neste método as avaliações são utilizadas em conjunto com técnicas de mineração de dados e aprendizado de máquina para formar um modelo de predição de avaliações para recomendação do itens [Fernández, 2018]. *Principal Component Analysis* (PCA), *Singular Value Decomposition* (SVD), clusterização [Fernández, 2018], árvores de decisão e classificadores de Bayes [Aggarwal, 2016] são alguns exemplos de técnicas que podem ser aplicadas nesse contexto. Como também visto em aplicações de aprendizado de máquina, nesse método de recomendação, existe uma fase de aprendizado separada da fase de predição.

O método *model-based* tem a vantagem de trabalhar melhor com matrizes esparsas e escalabilidade de conteúdo quando comparado ao *memory-based*. Por outro lado, o modelo criado pode sofrer *overfitting* com facilidade, precisar de uma grande quantidade de dados, além de ser computacionalmente pesado [Fernández, 2018]. Recentemente verificou-se

que uma combinação dos métodos *memory-based* e *model-based* produzem resultados muito precisos [Aggarwal, 2016].

2.6.3 Sistemas Demográficos

Nesse método, são utilizados os dados demográficos do usuário, como idade, sexo, cidade em que mora, etc. Normalmente esses dados são obtidos através de consulta direta com o usuário, por meio de formulário de cadastro de perfil [Araújo, 2011]. Esse tipo de sistema de recomendação não costuma produzir bons resultados quando usado de forma isolada, mas pode melhorar consideravelmente os resultados do sistema quando unido a outros métodos [Aggarwal, 2016], formando uma estratégia de recomendação híbrida (Seção 2.6.5). Muitos *websites* utilizam sistemas demográficos simples para mostrar conteúdos personalizados para o usuário, e demonstram alta efetividade [Ricci et al., 2010], como em sites de companhias aéreas, utilizando a cidade onde o usuário mora para mostrar rotas e outros conteúdos relacionados àquela cidade.

Sistemas demográficos são simples de implementar mas sua eficácia é altamente dependente do objetivo do sistema. Em muitas situações, informações demográficas do usuário não são suficientes para prever diferença de interesse entre os itens disponíveis, além de que nem sempre é possível, ou viável, obter os dados demográficos do usuário.

2.6.4 Baseado em Conhecimento

SRs baseados em conhecimento, ou em inglês *knowledge-based* (KB), se caracterizam por considerar os requisitos do usuário, que podem ser materializados na forma de restrições, e buscar os itens que melhor atendam esses requisitos. Para cumprir esse objetivo, o sistema consulta a base de conhecimento do domínio em que o sistema está aplicado, o qual é adquirido previamente, e contém atributos dos itens e regras pertencentes ao domínio em questão. Por exemplo, em um site de reserva de hotéis, o usuário indica alguns atributos que irão restringir a busca, como cidade, preço máximo e mínimo da estadia, etc, e o sistema então utilizará esses parâmetros e possivelmente regras de associação para gerar uma lista de hotéis recomendados. Essa forma de gerar recomendações é uma grande diferença em relação aos clássicos *CBF* e *CF*, já que não utiliza histórico de avaliações dos itens para embasar as recomendações.

Por não se basear em histórico de recomendações e sim em conhecimento previamente adquirido, esse sistema pode apresentar um desempenho consistente mesmo em ambientes onde transações usuário-item não acontecem com frequência, ou quando se está iniciando a aplicação do sistema [Aggarwal, 2016] [Ricci et al., 2010], contornando o problema do *cold start*. A desvantagem disso é que a geração do conhecimento do domínio tende a ser cara. Por isso, caso a base de conhecimento não exista, o custo de gerá-la pode inviabilizar a aplicação [Araújo, 2011]. Outra característica inerente é que sua capacidade de recomendação não aumenta à medida que o sistema vai sendo utilizado, se limitando ao conhecimento previamente

disponível, ao contrário dos métodos CBF e CF, que podem se beneficiar das transações realizadas pelos usuários e à medida que se forme um histórico grande de avaliações, podem superar o desempenho dele.

Há três tipos de conhecimento envolvidos em um sistema KB [Burke, 2002]:

- **Classificação:** Envolve a relação entre os itens e outros itens ou seus atributos. Por exemplo: tanto uma cadeira como uma poltrona são tipos de assento.
- **Conhecimento Funcional:** Regras que relacionam as necessidades do usuário aos atributos do item que podem atender essas necessidades. Exemplo: um usuário que quer assistir um filme alegre pode receber recomendações de comédias.
- **Conhecimento do Usuário:** Informações sobre o usuário, como características de itens que o usuário considera obrigatórias.

Um sistema baseado em conhecimento pode ser categorizado como *constraint-based* ou *case-based* [Pavlidis, 2019b]. No primeiro tipo, atributos dos itens são especificados pelo usuário para filtrar itens não aderentes às especificações ou formar a ordem recomendada do mais aderente às regras ao menos aderente. O exemplo citado a pouco do site de reserva de hotéis seria classificado como *constraint-based*. No segundo tipo de sistema KB, o usuário especifica um ou mais itens alvo. O sistema então, compara os atributos dos itens da base com dos itens alvo, recomendando os que melhor se assemelham aos itens escolhidos pelo usuário.

2.6.5 Sistemas Híbridos

Como pôde ser visto nas seções anteriores, cada método possui uma série de vantagens e desvantagens. Filtragem colaborativa funciona bem em contextos com muitas transações, e CBF, em situações onde há bastante informações sobre os itens. Para uns, o início da aplicação é um problema e para outros não. Considerando essas restrições de cada método, aplicações que utilizam vários deles em um único sistema, chamado de sistema híbrido, surgem como a opção viável e mais eficiente em muitos casos, pois conseguem anular as desvantagens dos métodos quando aplicados isoladamente.

Um sistema híbrido pode utilizar os métodos de sete formas diferentes segundo [Burke, 2002]. No método **mesclado**, aplica-se vários recomendadores separadamente e então mostra todos os resultados para o usuário ao mesmo tempo. No **revezamento**, o sistema troca o recomendador de acordo com o contexto, e no tipo **combinação de atributo**, um único sistema é utilizado baseando-se em várias bases de recomendação.

Outro tipo de sistema híbrido traz o resultado baseando-se em uma combinação dos resultados de diversos métodos, como aplicado em [Benouaret and Lenne, 2015], onde é feita uma ponderação dos métodos CBF, CF e demográfico. Nessa método, que é do tipo **por peso**, um novo usuário receberá recomendação considerando somente o sistema demográfico.

Após o usuário fazer algumas transações, o sistema começa a considerar também o CF, e depois que o sistema já possui um perfil mais detalhado do usuário, o peso do CBF deixa de ser nulo. Com essa tática, o sistema conseguirá fazer recomendações em situações de *cold start* e também irá desenvolver suas recomendações à medida que conhece mais sobre o usuário.

Os outros três tipos de métodos híbridos, **cascata**, **acréscimo de atributo** e **meta-nível**, tem em comum a característica de executar os recomendadores em série, para utilizar informações obtidas da execução de um método como base para o seguinte. No método cascata, são passadas as recomendações para o próximo método refinar. No acréscimo de atributo, o resultado do anterior é usado para gerar um novo atributo para o próximo e no meta-nível, todo o modelo aprendido por um método é usado pelo seu sucessor.

Uma sistema híbrido muito comum é a junção de CF com CBF. Isso se dá pelo fato de ambos precisarem de um histórico de avaliações, e que já existem diversos bancos de dados desse tipo para vários domínios. Essa combinação tem a vantagem de trazer a diversidade de recomendações, que CBF sozinho não consegue obter, e maior acurácia do que se fosse somente aplicado o CF, especialmente para os casos de usuários pouco semelhantes quando comparados aos demais, o problema chamado ovelha cinza¹ [Burke, 2002]. Um ponto fraco dessa configuração é que ambos são impactados pelo problema do *cold start*. Por esse motivo a combinação de CF ou CBF com KB ou método demográfico pode ser mais vantajoso.

2.6.6 Sistemas Baseado em Contexto

Um sistema baseado em contexto (BC) personaliza as recomendações de acordo com o contexto em que o usuário se encontra [Aggarwal, 2016]. Esse tipo de sistema é essencialmente híbrido, por fazer parte de uma aplicação onde se utiliza outro dos métodos já apresentados.

Para entender como o SR usa o contexto para a recomendação, é importante que esse termo seja bem definido, Contexto é um conceito amplo e multidisciplinar e, por isso, pode ser conceituado de diferentes modos dependendo do ponto de vista da disciplina do conhecimento escolhida. Contudo, neste trabalho, do ponto de vista de contexto aplicado a sistemas de recomendação, vamos considerar a definição de contexto *representativo*, caracterizado como um conjunto pré-definido de atributos que não se alteram de forma muito significativa com o passar do tempo [Ricci et al., 2010] e que, adicionalmente, tem relação e relevância para a aplicação em questão.

Dados de contexto podem ser fornecidos diretamente pelo usuário, como, por exemplo, humor no momento, suas informações históricas de uso do sistema ou suas preferências relacionadas a determinado assunto. Dados também podem ser obtidos do ambiente, como a data/hora, estação do ano, temperatura, localização, condições climáticas, etc.

O uso do tempo pelo sistema BC pode ser feito de muitas formas diferentes. A data em que o sistema está sendo utilizado pode ser relevante para a recomendação em aplicações

¹Situação onde um usuário não possui similaridade de interesses com nenhum dos outros disponíveis.

que, por exemplo, tenham alguma característica sazonal, como lojas de decorações. Em um momento próximo ao natal, artigos natalinos podem ter um destaque maior, assim como artigos masculinos quando se está próximo ao dia dos pais e, assim por diante, para as demais festividades do ano.

O horário de uso do sistema também pode ser determinante para uma boa recomendação. Por exemplo, em um sistema de recomendação turístico, o usuário pode ter preferência em conhecer determinados pontos turísticos de uma cidade de manhã, por possuir a luz do sol, e outros seriam mais interessantes à noite. Restrições de horários de funcionamento dos locais também são um fator importante para uma recomendação satisfatória em aplicações desse tipo.

O tempo também pode ser considerado como a duração que o usuário tem para uso do sistema. Restrições de tempo mínimo e/ou máximo de alguma atividade pode ser informado pelo usuário ou estimado pelo sistema para orientar o recomendador. A duração também pode ser combinada com outras informações, como a distância percorrida/a percorrer pelo usuário ou sua localização, ajudando na previsão de possíveis necessidades.

Outro aspecto importante do tempo que o SR pode utilizar é a gravação do momento em que as transações usuário-item foram realizadas. Com isso é possível consultar a diferença entre o momento atual e o momento de realização das transações, de modo que possa tratar as transações mais antigas de um jeito diferente das mais novas. Várias soluções de SRs utilizam essa técnica para filtrar ou ponderar os efeitos das transações sobre a recomendação do sistema. Duas abordagens simples com esse propósito são os modelos *window-based* e *decay-based* [Aggarwal, 2016]. No primeiro, transações realizadas fora de uma janela de tempo específica são desconsideradas do processamento da recomendação, e, no segundo modelo, pesos calculados com base em o quão recentes são as transações, são utilizados para relevar a importância da transação para a recomendação.

A localização do usuário é uma informação de contexto que cresceu em importância a partir da popularização dos *smartphones* com função de GPS [Aggarwal, 2016]. Ela atualmente é bastante utilizada em aplicativos de pedidos de comidas, onde eles recomendam restaurantes próximos à localização atual do usuário. As figuras 2.2a e 2.2b ilustram dois exemplos que aplicam essa informação para fazer recomendações aos usuários. Outra aplicação é em situações onde se sabe o caminho que o usuário irá percorrer, para oferecer personalização nas recomendações baseando-se na combinação da distância percorrida, distância restante e posição atual.

De forma geral, as informações contextuais podem ser orientadas por perguntas, pesquisas, elicitación ou estimativa de preferências do usuário [Ricci et al., 2010]. Aplicações orientadas por perguntas costumam pedir preferências de usuários em categorias dentro do domínio da aplicação. Por exemplo, na seção de livros de um *e-commerce* o usuário pode ser requisitado a informar preferências de gêneros de literatura ou, em um SR de restaurantes, pode fornecer seu grau de interesse por diferentes estilos gastronômicos. Um sistema que utilize informações do usuário pelas redes sociais é uma aplicação de sistema BC orientado

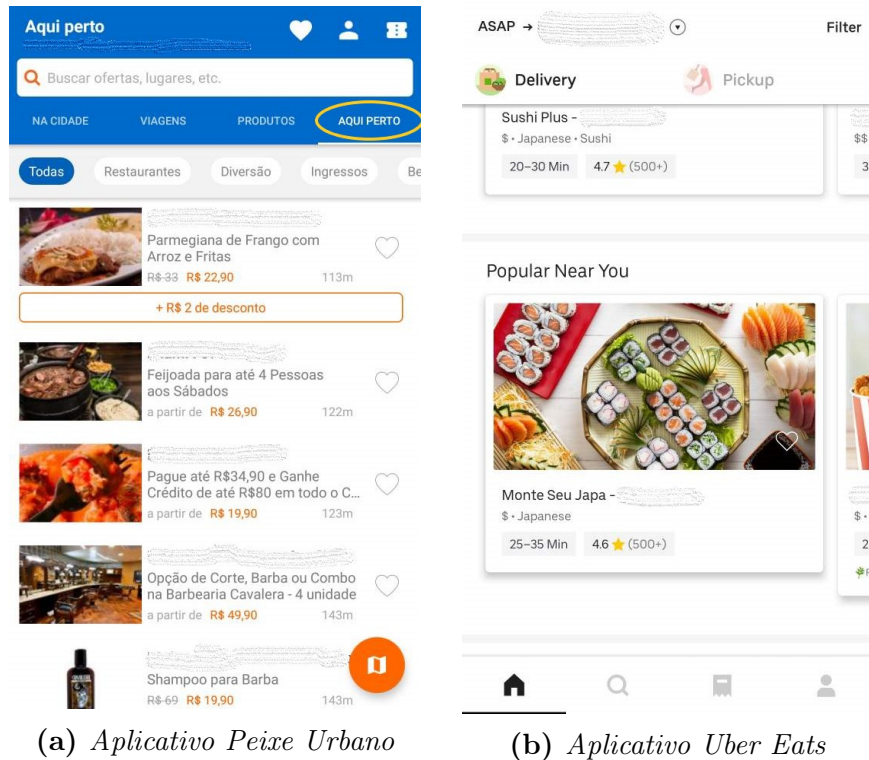


Figura 2.2: Exemplos de aplicação de recomendações baseadas na localização do usuário.

por pesquisa. De lá podem ser obtidas preferências do usuário pelos conteúdos de suas publicações, publicações de interesse, grupos o qual faz parte, lista de amigos, informações pessoais, etc.

Os sistemas BC baseados em estimativas e elicitación de preferências por contexto se referem a aplicações que observam o comportamento de uso do sistema pelo usuário, de forma isolada ou em comparação com os demais usuários. Também pode verificar o histórico de *feedbacks* informados pelo usuário em conjunto com algum dos métodos já citados nesse capítulo. As informações de contexto podem ser aplicadas de variadas formas em momentos diferentes do processamento dos dados pelo sistema de recomendação. Nesse sentido, os sistemas BC podem ser categorizados em três tipos [Ricci et al., 2015] [Aggarwal, 2016]:

Pré-Filtragem Contextual

Nesta categoria são encaixadas as aplicações de sistemas BC que utilizam as informações de contexto antes da predição de recomendação, de modo que o contexto selecione ou construa os dados que serão utilizados pelo método de recomendação de fato. A aplicação com o atributo tempo do tipo *decay-based* citado há pouco, onde transações mais antigas tem menor relevância na recomendação, é um exemplo que se enquadra nessa categoria.

Pós-Filtragem Contextual

Como o próprio nome indica, as aplicações de sistemas BC que utilizam informações contextuais após o processamento das transações, pelo método de recomendação tradicional,

fazem parte dessa categoria. Nesses casos, o SR prediz as recomendações normalmente, sem interferência do contexto. Ele é usado somente com a saída do recomendador, modificando-o de acordo. Utilizando a aplicação *decay-based* novamente, ela pode ser categorizada aqui caso o modificador de relevância pelo tempo seja feito posteriormente a recomendação. A modificação pode ser feita através da filtragem dos itens irrelevantes ou de um ajuste no valor ou *ranking* de recomendações.

Modelagem Contextual

São os sistemas BC que utilizam informações contextuais como atributos para o método de recomendação padrão da mesma forma que utiliza as transações do usuário. Por exemplo, o tempo de vida da transação pode ser utilizada como uma das dimensões de um recomendador CF *model-based* em diversas técnicas de aprendizado de máquina. Outro exemplo é utilizar o tempo na equação de semelhança usuário-usuário ou item-item, conforme proposto em [Adomavicius and Tuzhilin, 2018], formando uma distância multidimensional entre duas transações e quanto mais distantes elas estão, menor é o efeito dela sobre a recomendação.

Tipo	Forma de Recomendação
CBF	Utiliza atributos dos itens como parâmetro para as recomendações. Um perfil do usuário com os atributos que ele mais gosta é gerado e itens com atributos que mais se relacionam ao perfil são priorizados.
CF	Realiza recomendações se baseando nos usuários (ou itens) da base de dados com avaliações mais semelhantes com o usuário (ou item) em questão.
Demográfico	Utiliza informações demográficas dos usuários como base para estabelecer grupos com tratamentos diferenciados.
KB	Pode se basear em classificações e/ou relacionamento entre os itens, além de restrições definidas pelo usuário como filtro para as recomendações.
BC	Se refere a todas as formas de incorporar informações de contexto do usuário em um SR, como tempo, clima, localização, humor, etc.
Híbrido	Combina dois ou mais dos métodos anteriores como forma de reduzir as desvantagens que haveria se fossem implantados individualmente.

Tabela 2.1: Resumo das principais características dos tipos de sistemas de recomendação.

2.7 Avaliação de um Sistema

Como visto na Seção 2.6, existem diversos tipos básicos de sistemas de recomendação, além da possibilidade de combiná-los das mais variadas formas (a Tabela 2.1 faz um resumo dos tipos apresentados). Cada uma dessas possibilidades traz suas características especiais que irão definir se o sistema resultante será ou não efetivo para sua aplicação. Dadas tantas possibilidades, definir quais serão as características do sistema final não é simples. Devido a essa dificuldade, a avaliação do desempenho de sistemas de recomendação se constitui um importante processo no *design* do sistema final.

Realizar a avaliação de um SR tem diversas semelhanças com a de classificação e modelos de regressão [Aggarwal, 2016]. Isso porque vários métodos de recomendação utilizam, por vezes, as mesmas técnicas de mineração de dados.

Diversos critérios devem ser considerados para que se faça uma avaliação completa de um sistema de recomendação. Por vezes o único critério utilizado para comparar diferentes métodos é a acurácia dos valores de recomendação obtidos em relação aos valores reais. Porém, muitas vezes, apenas ter uma acurácia alta não garante que o sistema será eficaz. Outros critérios, como cobertura e confiança, podem ser usados para avaliar o desempenho em relação aos objetivos do sistema. Na Seção 2.7.2 serão detalhados os critérios que podem ser considerados ao avaliar um SR.

2.7.1 Tipos de Avaliação

Há três tipos básicos de avaliação de um SR: Avaliações *on-line*, *off-line* e estudos do usuário.

Estudos do Usuário

Nessa abordagem usuários são explicitamente convidados a participar da avaliação. Informações podem ser obtidas em três etapas: coleta inicial de *feedback* dos voluntários, informações coletadas pelo sistema durante o uso pelo usuário e *feedback* coletado após o uso.

Essa técnica tem o risco de enviesar o resultado da avaliação, pelo fato de somente usuários que aceitam participar da avaliação são considerados e por eles saberem que estão avaliando o sistema. Levanta-se a questão que possivelmente um perfil específico de usuários, como os mais engajados com o contexto da aplicação, terá maior representatividade na amostra de participantes do que na população de usuários, tornando o experimento estatisticamente inconsistente.

Avaliação *On-line*

Pode ser considerada a mais precisa e também a mais difícil de implementar dos três tipos. Mais precisa porque costuma ter menos *bias* que os demais [Aggarwal, 2016], pois é normalmente conduzido no ambiente produtivo, com clientes reais, inclusive com a possibilidade do cliente não saber que está avaliando o sistema, nos casos de avaliação indireta.

Nesse contexto é possível implementar diversas técnicas eficazes de avaliação, como o teste A/B². Nesse teste os usuários são separados explícita ou implicitamente em dois sistemas distintos. Em seguida métricas podem ser coletadas, como a taxa de conversão, que mede a frequência com que usuários escolhem os itens recomendados pelo sistema. Sendo ela

²Método de teste de design onde duas versões diferentes de um sistema são submetidos a dois grupos separados de usuários (idealmente de mesmo tamanho) e seus resultados são comparados em relação a uma métrica específica [Kohavi and Longbotham, 2017].

mensurada para ambos os grupos, ao final de um período específico as taxas de conversão obtidas são comparadas para verificar qual obteve melhor desempenho. Esse teste pode ser feito com qualquer número de métodos ao mesmo tempo, desde que seja possível dividir o público entre esses métodos. A desvantagem dessa técnica é que, para obter um resultado confiável, é necessário uma quantidade grande de usuários utilizando o sistema, o que nem sempre é possível. Também para implementar a rotina de avaliação geralmente é necessário ter permissão para modificar a aplicação original.

Avaliação *Off-line*

Nesse tipo de teste, histórico de avaliações dos usuários e outros dados são aplicados nos sistemas sendo avaliados. Tendo esse histórico, a execução desse teste é mais razoável, considerando que costuma-se gerar resultados bem apurados. Essa é uma técnica muito atrativa pois não requer interação com usuários reais [Ricci et al., 2015]. O ponto fraco dela se dá nos casos onde não existe histórico de avaliações no mesmo contexto onde será aplicado o sistema a ser avaliado, nem em contextos similares. Também, caso exista, pode haver diferenças temporais que influenciam significativamente os resultados do sistema, sendo que os dados podem ter sido obtidos em um período muito anterior ao momento de avaliação e, por isso, podem não mais representar fielmente as reações do usuário no futuro [Aggarwal, 2016]. Outras variações nas características da aplicação, além do tempo, que possam ter ocorrido entre o momento da coleta dos dados e o momento atual da avaliação, também podem impactar nos resultados.

2.7.2 Critérios de Avaliação

Os critérios de avaliação, são os conceitos considerados ao avaliar um SR. Nesta seção serão apresentados os principais deles. Mesmo assim, a escolha de quais critérios irão compor o procedimento de avaliação do sistema deve ser feita com base nos objetivos que a aplicação deseja alcançar.

Acurácia

Acurácia é sem dúvida o mais explorado dos critérios [Ricci et al., 2010]. Um dos motivos se dá por serem facilmente encontradas métricas quantitativas que o representem, permitindo assim uma classificação absoluta entre os SRs. A importância dessa métrica para os objetivos de um sistema de recomendação está relacionada à premissa de que um sistema que traz resultados precisos para o usuário será mais útil e aceito por ele.

Há basicamente dois tipos de métricas de acurácia para SRs: acurácia da previsão da nota do usuário e do *ranking* das recomendações. Para medir a acurácia do primeiro, as métricas mais comumente utilizadas são o erro absoluto médio (MAE) ou a raiz do erro quadrático médio (RMSE) [Pavlidis, 2019b][Ricci et al., 2015] e para obter a acurácia do *ranking*, que

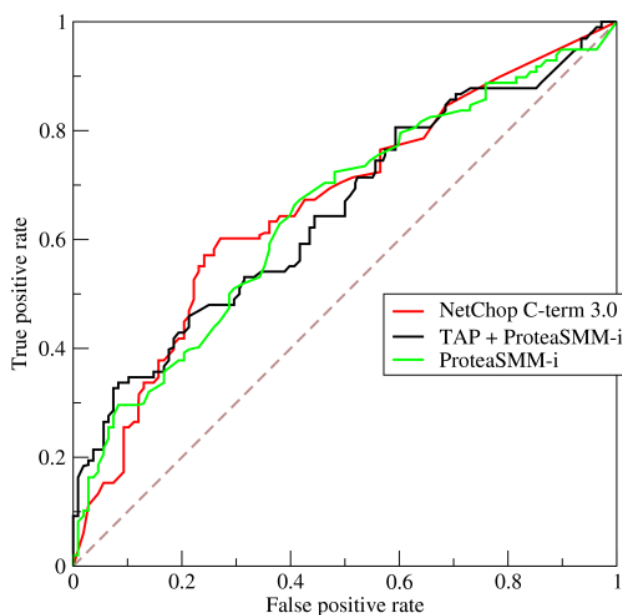


Figura 2.3: Exemplo de comparação de acurácia de três classificadores com uma curva ROC.

Fonte: https://en.wikipedia.org/wiki/Receiver_operating_characteristic

é gerado por sistemas que mostram seu resultado sob o formato de uma lista ordenada de recomendações, pode-se comparar com um *ranking* de referência ou com a utilidade do *ranking* para o usuário. A diferença entre o MAE e RMSE é que o segundo penaliza erros grandes com maior intensidade que o primeiro [Ricci et al., 2015]. Para exemplificar, consideremos uma situação onde há dois sistemas, A e B, que podem ser aplicados em determinado contexto. O sistema A gera muitos erros pequenos e B costuma ser bem assertivo mas com alguns erros grandes. Caso essa característica do sistema A seja melhor para o contexto em questão, o RMSE seria a métrica ideal para comparar os sistemas.

Outra métrica muito utilizada para avaliar a acurácia de classificadores em geral, assim como SRs, é a curva ROC (*Receiver Operation Characteristics*) [Fawcett, 2006]. Uma curva ROC é um gráfico que informa a relação entre falsos e verdadeiros positivos [Ricci et al., 2015]. Isto é, suponha que existam duas classes, positivo e negativo. O gráfico apresenta o percentual de acertos na classe positivo (eixo y do gráfico), de todas as vezes que classificou-se como positivo, em relação ao percentual de erros ao classificar como positivo (eixo x). A figura 2.3 ilustra esse exemplo com três classificadores sendo comparados. Quanto mais próximo do valor 1 no eixo y, melhor é o desempenho do classificador. A linha pontilhada indicada um classificador aleatório.

Essa métrica é muito útil para identificar um ponto de corte, a partir do qual o classificador não possui um desempenho satisfatório [Araújo, 2011]. Contudo ela se limita a avaliar classificadores binários. Verificações de classificadores de n classes, desde que seja um n pequeno, é possível mas se torna muito mais complexo [Fawcett, 2006].

Cobertura

Mede o percentual de usuários ou itens que o sistema é capaz de recomendar. Mesmo tendo uma acurácia muito alta para alguns itens, um SR pode não conseguir recomendar certos itens. As matrizes esparsas que formam a base dos métodos de recomendação são a causa de uma cobertura menor que 100%.

Confiança

Dois tipos de confiança são considerados aqui. Confiança do sistema sobre o resultado e confiança do usuário sobre o sistema. O primeiro tem como métrica mais comum um valor percentual da confiança de que o resultado está correto [Ricci et al., 2010], dentro de uma margem de erro da distribuição de probabilidade. Por exemplo, o sistema pode mostrar a avaliação de determinado item com nota 3 de 5, com 95% de confiança, dada uma margem de erro de $\pm 0,5$ pontos.

Muitas vezes é interessante divulgar o valor de confiança para o usuário, pois isso pode influenciar na sua tomada de decisão. Uma recomendação de um item com nota 9/10 com 99% de confiança tem um efeito bem diferente de recomendar um item com a mesma nota, mas 25% de confiança. No segundo caso o usuário estará ciente que a recomendação não é totalmente confiável e poderá verificar outros detalhes do item antes de tomar uma ação.

O segundo tipo de confiança, do usuário pelo sistema, também chamado de **credibilidade**, pode ser medido de forma mais precisa através de estudos com usuários, perguntando-os diretamente se confiam nos resultados que o sistema gera [Aggarwal, 2016] [Ricci et al., 2010].

Uma forma de aumentar a confiança do usuário é deixá-lo ciente do racional por trás da recomendação, como por exemplo, informando o intervalo de confiança do sistema e, no caso de CBF, quais atributos daquele item que mais influenciaram no resultado.

A confiança em testes *on-line* pode ser verificada, é claro, perguntando diretamente para o usuário, mas também pode ser estimada. A estimativa pode se basear na quantidade de usuários que voltam a utilizar o sistema e/ou pela evolução do uso deles. Se essas métricas estão aumentando, podem indicar confiança do usuário. Entretanto, essas métricas também são influenciadas por outros critérios, então pode não ser possível afirmar que a confiança aumentou. É incerto como verificar a confiança em testes *off-line*.

Novidade

Se refere a chance do sistema recomendar itens que o usuário não conhecia [Aggarwal, 2016]. Enquanto verificar esse critério pode ser realizado pela pergunta direta ao usuário em estudos com usuários e teste *on-line*, essa verificação é mais complexa em testes *off-line*. [Ricci et al., 2010] sugere, nesse caso, escolher um momento no tempo entre as recomendações da base, ocultar as avaliações feitas após esse período, e algumas avaliações feitas anteriormente, simulando os itens que o usuário conhecia mas não avaliou. Depois, verifica-se se o sistema

irá recomendar itens dentro desse grupo ou não, assim consegue estimar a proporção de novidades dentre todas as recomendações que o sistema sugere. A forma como se escolherá quais itens terão suas avaliações ocultas é crítico para o sucesso do teste. Ele deve ser semelhante a como isso se comporta na realidade.

Serendipidade

Serendipidade vem da palavra da língua inglesa *serendipity*, que significa "Ato ou capacidade de descobrir coisas boas por mero acaso, sem previsão". Esse critério mede o grau de surpresa, ou criatividade, das recomendações feitas ao usuário. Serendipidade é um critério mais restritivo que novidade, pois exige que seja uma novidade para o usuário (item que ele não conhecia) mas também que essa novidade não seja óbvia. Por exemplo, se o usuário leu o livro de um autor que ele gosta, e o sistema recomenda outro livro desse autor, mas que o usuário não conhecia, isso se caracteriza como novidade, contudo não é considerado serendipidade, pois a relação entre os dois itens é óbvia.

Uma forma de mensurar a serendipidade do sistema pode ser feita comparando-o com um sistema primitivo que tem forte tendência a fornecer recomendações mais óbvias, como o CBF. Então são extraídos os N resultados mais recomendados pelo sistema e comparados com a lista de recomendação do sistema primitivo. A razão entre a quantidade de itens que não foram encontrados no sistema primitivo e são considerados úteis para o usuário sobre o total de itens considerados nos dá a taxa de serendipidade estimada.

Diversidade

Mensura a diferenciação entre os itens recomendados pelo sistema. Se os primeiros itens recomendados de um site de filmes são todos parte de uma sequência, o sistema possui baixa diversidade.

Robustez/Estabilidade

Mede a estabilidade do sistema ao ser alvo de informações falsas. Como exemplo, aplicações comerciais precisam ter alta robustez, pois pode ser de interesse de alguns usuários que certo item seja mais recomendado, pois influenciará nas vendas do mesmo. Se, de forma maliciosa, um usuário puder interferir nas recomendações, é considerado que o sistema possui uma baixa robustez.

Escalabilidade

Verifica a capacidade do sistema em suportar um número grande de usuários, itens e transações. Tempo de treinamento do sistema, predição das recomendações e capacidade de memória são métricas que podem ser utilizadas nesse critério.

2.7.3 Validação

Foram apresentados diversos critérios que podem ser usados para medir o desempenho de SRs e compará-los entre si. Contudo, os testes de desempenho do sistema não podem ser feitos sobre os mesmos dados utilizados para treiná-lo. Na maioria dos casos, qualquer que seja o conjunto de dados utilizado para treino, ele não irá determinar fielmente as previsões futuras para o resto da população. Isso porque esses dados apresentam ruído ou um viés, que geram um erro na predição do resultado em relação ao resultado real [Abu-Mostafa et al., 2012].

Por exemplo, um sistema CF foi treinado com base em N transações realizadas pelos usuários. Ao se medir o MAE do sistema utilizando essas N transações, obteremos um valor de erro que chamaremos de E_{in} . Esse é o chamado erro da amostra (N). Se em seguida pudéssemos usar esse mesmo sistema treinado com N para prever os resultados com toda a população em potencial, o MAE calculado, que chamaremos de E_{out} , provavelmente seria maior que E_{in} . Isso porque o sistema forma sua solução com base nas características de N , que pode não representar fielmente as características da população.

Como o que nos interessa é saber se o sistema terá um bom desempenho com a população e não com a amostra já obtida, divide-se os dados em dois conjuntos, N e M , onde N se usa para treinar o sistema e M para testá-lo. Assim, o teste é feito para estimar o E_{out} . Com um conjunto M muito pequeno, a estimativa de E_{out} pode ser muito enviesada/com ruído, invalidando-a. Por outro lado, se designar muitos dos dados para o conjunto M , o tamanho de N pode ser muito pequeno e comprometer o treino do sistema, especialmente em casos onde se dispõe de um conjunto pequeno de dados.

Situações com poucos dados para treino/testes é relativamente comum. Por isso a técnica de *validação cruzada* é muito utilizada nesses casos [Araújo, 2011]. A seguir serão detalhados os passos da técnica *k-fold cross validation*, que é uma das formas de validação cruzada, para estimar o E_{out} .

1. Separam-se os dados em k porções.
2. $k - 1$ porções são destinadas ao treino do sistema.
3. A porção restante é usada para testar o sistema, obtendo uma estimativa de erro.
4. Substitui-se a porção de teste por uma que foi usada para treino, com a de teste passando a integrar as $k - 1$ porções de treino.
5. Repete-se os passos 2 a 4 até que todas as k porções tenham sido usadas exatamente uma vez como a porção de testes.
6. O E_{out} estimado é encontrado através da média aritmética de todas as estimativas de erro encontradas no passo 3.

Embora não haja um melhor valor inteiro para atribuir a k , o *10-fold cross validation* é uma escolha comum na prática. Essa técnica é utilizada na verificação do SR de melhor desempenho na Seção 5.5.

2.8 Desafios na Aplicação de SR em Museus

Ao visitar um museu é usual o desconhecimento do usuário sobre a experiência e a numerosa oferta de itens disponíveis no local. Essa característica se acentua pelo fato da visita a museus ser uma atividade relativamente pouco popular no Brasil, e quem vai geralmente está em sua primeira visita. Considerando que o tempo de visita geralmente não é suficiente para o visitante conhecer todo o acervo do local, se torna clara a necessidade de algum tipo de auxílio na decisão do que visitar e o que deixar de fora dentro do limite de tempo.

Além do problema principal, de dificuldade na escolha e restrição de tempo, relacionado à aplicação de sistemas de recomendação, há diversos outros desafios e limitações que devem ser levados em consideração ao desenvolver um sistema de recomendação. A seguir serão detalhados alguns dos desafios mais relevantes e na Seção 4.1 será descrito qual o impacto de cada um deles na aplicação que foi realizada deste trabalho.

2.8.1 Desafios específicos para SRs aplicados em Museus

1. **Distância a percorrer:** A distância a ser percorrida pode inviabilizar algumas recomendações propostas em aplicações onde os itens possam estar muito longes entre si. O usuário pode não estar disposto a se deslocar tanto quanto a rota proposta exige. Também a ordem recomendada de visita dos itens, caso o sistema tenha essa proposta, não pode apresentar ineficiências muito grandes de trajetória, pois irá impactar na confiança do usuário sobre o sistema e sua retenção. Em alguns casos pode ser interessante deixar a cargo do usuário definir de que forma visitar os itens recomendados, se esta decisão não for muito complexa.
2. **Tempo disponível:** Em sistemas que mostram todos os itens disponíveis com sua nota de recomendação, ou apenas ordenados do mais para o menos recomendado, o tempo que o usuário pretende passar no local não precisa ser considerado pelo sistema, pois ele é controlado pelo próprio usuário. Mas há situações onde é importante para a experiência do usuário que seu tempo disponível seja considerado. Um exemplo é quando há a expectativa de visitar muitos itens e eles estão distantes entre si. Nesse cenário, sem considerar o tempo, o usuário poderia acabar não visitando os itens mais recomendados para ele pela dificuldade em organizar o seu tempo. Também caso o usuário estivesse seguindo uma lista priorizada por recomendação, ele correria o risco de fazer uma rota muito ineficiente, no caso, por exemplo, do primeiro item da lista estar muito mais distante do segundo item do que dos demais e assim por diante.

Há também a incerteza do tempo estimado que o visitante planeja passar no local e do tempo que ele precisará para visitar cada item. Esse tempo pode ser facilmente aumentado ou diminuído devido a influência de diversos fatores [Pavlidis, 2019a], como a experiência inicial da visita, o nível de interesse demonstrado pelos itens, a velocidade de deslocamento entre os itens recomendados, etc.

3. **Disponibilidade dos itens:** Um item pode não estar disponível em todos os momentos. Pode estar aberto em apenas um período do dia ou mesmo apenas em algum dia da semana. Pode também ser um item com horário específico de visitação, como uma apresentação, que depende de uma agenda dos organizadores. Para qualquer um desses casos, pode ser que cada situação se aplique para apenas alguns itens do local. Quanto maior a diversidade da forma como os itens são disponibilizados, mais complexa deverá ser a solução que atenda os requisitos.
4. **Ordem fixa de visitação:** Esta é uma característica comum em museus, que buscam, ao oferecer uma ordem padrão recomendada para todos os visitantes, otimizar o tempo/distância necessária para realizar a visita, ou formar coleções³ organizadas por algum tipo de ordem lógica, como cronológica, por nível de importância, região, categoria, etc. Locais que obrigatoriamente possuem ordem de visitação bem definida, de certa forma, reduzem a necessidade de um sistema de recomendação, pois restringem as possibilidades do usuário de realizar a visita. Contudo, o problema de sobrecarga de informação e desconhecimento do usuário, em menor nível, se mantém.
5. **Itens temporários:** Itens ou coleções do museu podem ter caráter rotativo, tendo sua exposição realizada por um período, para depois ser levado para outro local. Esse tipo de característica adiciona bastante complexidade à viabilização do sistema de recomendação no local. Quanto maior a rotatividade dos itens disponíveis no local, maior é a dificuldade de manter a cobertura e acurácia do recomendador. Pois, se for uma implementação baseada em CF, deverá esperar novas avaliações serem realizadas pelos primeiros usuários a conhecer os itens. Já para os SRs KB e CBF, haverá o esforço de continuamente gerar atributos sobre os novos itens.
6. **Sinergia entre itens:** Os itens da exposição muitas vezes têm uma relação entre si, seja por trazer uma continuidade, como sequências de filmes e livros, por exemplo, seja simplesmente por pertencer a uma mesma categoria, autor, momento histórico, etc. Então a ordem pela qual o usuário conhece os itens do museu pode ter um forte impacto na sua satisfação. Se, por exemplo, em um sistema *CBF*, um usuário avalia positivamente um item o qual o museu possui vários outros itens muito semelhantes, é provável que o sistema irá recomendar esses outros itens para o usuário, trazendo uma sensação de repetitividade. Se, por outro lado, itens são partes de um trabalho maior

³Conjunto de objetos reunidos, de forma coerente e significativa, em um contexto seguro, por um estabelecimento ou indivíduo, e que é comunicado a um público. [Desvallées and Mairesse, 2013]

e a visualização deles em sequência permite adquirir um conhecimento mais completo daquele assunto, seria importante que o sistema os recomendasse em sequência.

Contudo, a definição de um modificador de satisfação entre itens não é algo simples e pode impactar tanto positivamente como negativamente na experiência total do usuário. Além disso, encontrar esses valores se torna mais uma atividade de coleta de informações, onerando ainda mais a aplicação do sistema em um local. De qualquer forma, existem propostas de SRs aplicados a museus que consideram esse fator em suas recomendações [Rossi et al., 2016] [Pavlidis, 2019a].

7. **Visita a itens fora da rota recomendada:** No caso de sistemas onde a recomendação é no formato de uma rota formada pela visita a um conjunto de itens, não se pode assumir que o sistema conseguirá recomendar exatamente tudo que o usuário irá querer ver. Durante a visita, o usuário pode encontrar itens que despertem seu interesse. Nesse caso, seria útil se o SR permitisse adicionar o novo item à rota, ou que o usuário pudesse ver suas informações pelo sistema, mesmo não fazendo parte da recomendação. Essa ação também é uma oportunidade para o sistema aprender mais sobre o usuário, considerando-a como um *feedback* implícito.
8. **Visitas em grupo:** Visitas a museus em grupos são muito comuns. Essa característica do público alvo do SR pode trazer um grande impacto em sua eficácia, especialmente para grupos em que seus membros possuam preferências muito heterogêneas.
9. **Pontos de interesse:** Os usuários têm outras necessidades além da visita em si aos itens. Pontos de interesse como banheiros, locais de alimentação, bebedouros, pontos de encontro, áreas de *wi-fi* e outros pontos de interesse do local podem ser indicados ou considerados em uma rota recomendada para o usuário.
10. **Dificuldade em obter *feedbacks* explícitos do usuário:** É muito comum que uma tela de *feedback* seja apresentada ao usuário, na conclusão de uma atividade, requisitando sua satisfação. Porém, essa funcionalidade deve ser usada com parcimônia, pois pode atrapalhar a experiência de uso e também pode levar a conclusões equivocadas caso a pergunta que o usuário precise responder não seja de fácil entendimento. Também pode não ser possível obter esse *feedback* pela característica da aplicação, em situações como quando o cliente não precisa voltar ao aplicativo ou quando lhe falta tempo.
11. **Sinal de *Internet* / *GPS*:** A maioria dos novos aplicativos móveis desenvolvidos se comunica com servidores para buscar as informações necessárias para o seu funcionamento, por isso a comunicação com a *Internet* é um requisito imprescindível nesses casos. Contudo, a falta de cobertura de rede móvel e redes *wi-fi* não é rara no país, com maior probabilidade de se verificar isso em áreas mais afastadas dos grandes centros

urbanos. Como também não são raros os casos de museus e parques que se situam nessas áreas, essa dificuldade pode influenciar bastante na concepção da aplicação.

2.8.2 Desafios de SRs em Geral

1. **Cold Start:** Define a dificuldade que um sistema de recomendação pode ter para dar cobertura ou acurácia nas recomendações para novos usuários e/ou de novos itens [Ricci et al., 2015][Pavlidis, 2019b]. Obviamente o momento inicial de uso do sistema com uma base de dados vazia é o principal momento de *cold start*. Porém, mesmo com uma base grande de dados, novos itens adicionados não terão, por exemplo, avaliações de usuários, no caso do CF, e, por isso, podem acabar sendo desconsiderados na recomendação. Da mesma forma, novos usuários, dos quais o sistema não possui nenhuma informação, podem não receber recomendações ou recebê-las com baixíssima acurácia.
2. **Matrizes de Avaliações Esparsas:** É uma dificuldade muito característica de sistemas de recomendação. Mesmo em aplicações relativamente pequenas, grandes matrizes esparsas podem ser geradas. Isso ocorre devido a grande quantidade de itens e usuários e todas as relações entre eles, que, em sua maioria, possuem valores nulos [Ricci et al., 2015]. Por consequência disso, um alto consumo de memória e processamento acaba sendo necessário. Por isso são recomendadas técnicas de redução da dimensionalidade nessas aplicações como um pré-filtro da base de dados, resultando em uma base com menos dimensões e, idealmente, mantendo uma representatividade muito próxima da base original.
3. **Fraudes e Ataques:** Qualquer sistema está sujeito a ataques e sistemas com aprendizado de máquina têm uma especial sensibilidade a isso, pois eles muitas vezes utilizam as informações dadas pelos usuários para aperfeiçoar seu processamento. No caso de SRs, ataques podem ser feitos de forma a manipular o resultado de recomendações, gerando resultados enviesados [Pavlidis, 2019a]. Isso pode ser facilmente feito em um SR tradicional através da entrada de informações idênticas ou parecidas em grande quantidade. Geralmente, esse tipo de ataque pode ter o objetivo de aumentar a avaliação ou popularidade de um determinado item ou diminuir a avaliação dos itens competidores, em uma situação de oferta de produtos, por exemplo [Aggarwal, 2016].
4. **Falta de conteúdo:** Nem sempre as informações sobre os itens ou sobre o local podem ser facilmente adquiridas. Em muitos casos, uma quantidade significativa de informações para cada item deve estar disponível no aplicativo para que este se torne útil ao usuário. Além de atributos e textos explicativos, outros tipos de mídia, como fotos, áudios ou animações, podem ser necessários, dificultando a completude da aplicação.
5. **Desempenho:** Sistemas de recomendação em muitos casos podem exigir grande quantidade de memória e processamento. Embora os aparelhos móveis tenham se desen-

volvido rapidamente para ambos os atributos, uma aplicação de SR ineficiente pode facilmente inviabilizar o processamento do sistema em um celular. Adicionalmente, há uma enorme variedade de celulares disponíveis atualmente e muitos dos que estão em uso possuem uma capacidade múltiplas vezes inferior aos de maior desempenho do mercado.

Uma característica da aplicação que traz grande impacto no desempenho é o retreino do sistema de recomendação. Muitas das soluções de recomendação existentes passam por um treino, computacionalmente pesado, com os dados disponíveis para então, rapidamente, gerar as avaliações de todos os itens requisitados. Entretanto, geralmente não temos muitas informações de antemão e, à medida que o usuário vai gerando dados de uso, o sistema pode fornecer uma recomendação mais assertiva. Por essa razão é importante que seja viável o retreino do recomendador durante o uso do sistema.

Por mais eficiente que seja a implementação, sistemas de recomendação, assim como outras aplicações de aprendizado de máquina, costumam processar muitos dados, impossibilitando, em muitos casos, seu processamento em um dispositivo móvel comum. Nesse contexto, o processamento do sistema pode ser feito remotamente, por exemplo, por um servidor na nuvem, que contém toda a base histórica de dados e vai recebendo em tempo real os novos dados recém gerados pelos usuários ativos. A troca de informações entre o aplicativo e o servidor se restringe a esses dados e aos valores de recomendações encontrados pelo servidor, liberando a responsabilidade do celular pela parte mais pesada do processamento.

No Capítulo 5 é apresentado o tempo de processamento do sistema de recomendação desenvolvido em relação à quantidade de dados utilizada.

Capítulo 3

Trabalhos Relacionados

Neste capítulo serão apresentados alguns trabalhos que utilizaram sistemas de recomendação aplicados a locais físicos com exposições com um número considerável de itens. São seis trabalhos apresentados da Seção 3.1 a 3.6, finalizando na Seção 3.7 com a explicação da contribuição dessa pesquisa em relação às demais.

Os trabalhos foram selecionados de resultados da busca utilizando as palavras *recommendation E system E museum* em diversas bases de trabalhos científicos:

- *Web of Science*
- *Springer Link*
- *IEEE Xplore Digital Library*
- *ACM Digital Library*
- *Science Direct*

Dos resultados encontrados foram desconsiderados os trabalhos onde a pesquisa focou em outro aspecto que não seja na recomendação de itens, onde não se identificou o uso de dispositivos móveis ou se o recomendador era aplicado em situações não relacionadas a locais de exposição de patrimônio cultural. A maioria dos trabalhos citados abaixo tem em comum a utilização de sistemas de recomendação híbridos, o qual, pelas vantagens descritas na Seção 2.6.5, também foi o tipo produzido neste trabalho. Este capítulo apresenta alguns dos resultados obtidos com essa pesquisa, de modo a abordar diferentes técnicas que foram utilizadas na solução proposta deste trabalho.

3.1 SR Baseado na Ontologia CIDOC-CRM

Nesta aplicação [Benouaret and Lenne, 2015] o autor busca diminuir o problema de falta de relação semântica entre os itens, através do uso da ontologia CIDOC-CRM¹ e da construção de um modelo semântico dos itens para gerar uma base de conhecimento. O sistema híbrido consome essa base com o método CBF e também considera outros dois métodos de recomendação, CF e sistema demográfico, para produzir o resultado.

A composição da nota final de recomendação é feita através da soma ponderada dos valores devolvidos pelos três métodos que compõe o recomendador. Os pesos de cada método na nota final é variável e depende da quantidade de itens avaliados pelo usuário ativo. Se o usuário está utilizando o sistema pela primeira vez e, por isso, ainda não tem nenhuma avaliação realizada, o sistema considera somente o resultado do sistema demográfico. Após algumas avaliações, o método CF passa a ser considerado. Depois do usuário ter formado melhor seu perfil através das avaliações, o SR KB é considerado na recomendação.

Após as recomendações serem calculadas, o sistema faz um *post-filtering* para combinar o resultado com o ambiente físico do museu, o local onde o usuário se encontra e quanto tempo ele deseja utilizar na visita.

3.2 ART Recommender

O *ART Recommender* [Wang et al., 2009] [van Hage et al., 2010] é aplicado no museu *Rijksmuseum*. O SR é baseado no método CBF e utiliza uma base de conhecimento do museu em conjunto com as preferências do usuário, sua localização, um grafo representando os espaços do museu e a localização das obras para fornecer suas recomendações.

Em [Wang et al., 2009], o trabalho busca verificar quais relações semânticas entre os itens são mais úteis para o CBF. Para isso, um experimento com um total de 48 participantes foi conduzido, levando em torno de 20-35 minutos por pessoa, onde se verificou a precisão das recomendações feitas para os usuários. As avaliações poderiam ter valores de 1 a 5 estrelas. Foi definido que todas as avaliações de 4 e 5 são consideradas relevantes e as demais não relevantes. A métrica de precisão utilizada era obtida pela razão entre a quantidade de itens considerados relevantes para o usuário pelo total de itens recomendados que o usuário avaliou. A relação entre itens diferentes que possuem o mesmo criador foi a que trouxe o melhor resultado em termos de precisão da recomendação.

Em [van Hage et al., 2010], o trabalho se preocupa em explorar a característica dos visitantes de museus em querer conhecer obras além das presentes nas recomendações do sistema. Para atender essas mudanças no rumo da visita, o usuário clica em um botão "visualizado" sobre a obra que ele viu e que não estava prevista no *tour* programado pelo

¹sigla para *Center for Intercultural Documentation-Conceptual Reference Model*, uma ontologia (conjunto de conceitos e categorias com suas propriedades e relações) referência para a descrição semântica de patrimônio cultural [Benouaret and Lenne, 2015]

SR. Essa ação faz o sistema considerar essa nova obra nas preferências do usuário e recalculas as recomendações.

A avaliação do resultado foi feita considerando a utilidade para o usuário e se a lista de recomendações é encontrada em um tempo razoável para permitir interações em tempo real e se segue uma rota eficiente pelo museu. A utilidade foi verificada separadamente em um grupo de iniciantes, usuários com pouco conhecimento sobre o conteúdo do museu, e em outro grupo que possui bastante conhecimento. Verificou-se que no grupo de iniciantes o SR trouxe maior benefício.

3.3 SR Aplicado a Pontos Turísticos

O sistema [Cao et al., 2018] busca recomendar pontos turísticos da *Xiamen University*, considerando as preferências do usuário, e otimizar o percurso necessário para os visitantes se deslocarem, visto que os pontos são relativamente distantes entre si.

As preferências do usuários são obtidas pela interação direta dele com o sistema no momento inicial de utilização do mesmo. A aplicação oferece seis "aspectos" para terem sua preferência informada pelo usuário (foto, shopping, comida, esportes, leitura e história) com a opção do usuário fornecer uma nota de 1 a 6 para cada aspecto que ele prefere. O usuário também indica quantos pontos ele deseja visitar e onde ele está no momento. Com essas informações o sistema verifica quais dos itens disponíveis mais combinam com suas preferências e gera uma lista de locais.

Um algoritmo guloso² modificado de busca de melhor rota entre pontos utiliza a lista de pontos de visitação recomendada para ordená-los de forma eficiente para o usuário visitá-los.

3.4 SR para Grupos

SR para museus que visa atender problemas de recomendação comuns nesse tipo de domínio como visitas de grupos de pessoas e distância a ser percorrida considerando a lista recomendada de obras [Rossi et al., 2016]. Em relação ao primeiro problema, a aplicação permite o cadastro de cada indivíduo para então formarem um grupo através da ferramenta. À medida que o grupo vai passando pelas obras, os membros do grupo as avaliam e o recomendador, baseado em CF, começa a prever as avaliações para as outras obras que cada indivíduo do grupo daria. Os resultados dos indivíduos são combinados para fornecer a recomendação para o grupo.

Para endereçar o problema da distância percorrida, a aplicação trata o museu como um grafo, de acordo com a disposição física do museu e de possíveis limitações promovidas pelos organizadores, para encontrar o menor caminho entre as obras. O algoritmo busca montar

²Tipo de algoritmo que busca fazer as escolhas que irão convergir para uma solução ótima local, com a expectativa de que essa solução é ou está próxima da solução ótima global.

um caminho que maximize a satisfação do grupo sem passar de uma distância máxima a ser percorrida. A satisfação é uma função calculada considerando a avaliação prevista do usuário.

A aplicação é avaliada de forma *off-line* e com estudos de usuário em um piloto. A avaliação *off-line* é feita com o uso da base MovieLens, que contém milhares de avaliações de centenas de usuários sobre filmes em uma escala de 1 a 5 estrelas. O estudo do piloto foi feito com 29 usuários, formando 9 grupos. Em ambos os casos foram utilizadas as métricas Ganho Cumulativo (do inglês *Cumulative Gain* - CG) e Ganho Cumulativo Descontado (do inglês *Discounted Cumulative Gain* - DCG) para avaliar a utilidade das recomendações para determinado grupo. As duas métricas são utilizadas para identificar um valor de relevância esperado para o usuário. Para o cálculo das métricas, eles consideraram como ganho esperado a média das avaliações previstas de cada usuário do grupo sobre a lista recomendada.

As avaliações comparam o desempenho da sequência de obras recomendada em relação às recomendações feitas considerando apenas as maiores notas previstas sem se preocupar com a distância entre elas. Em ambos os casos há o limitador de distância máxima. Segundo o estudo, os resultados apontam que as recomendações considerando a distância trazem maior satisfação e também aumentam o número de obras visitadas.

3.5 PGR

O sistema PGR (*Personalized Guide Recommendation*) proposto em [Huang et al., 2012] tem como principal objetivo diminuir a sobrecarga de informação através do uso do sistema de recomendação pelos usuários. Os usuário podem clicar nas obras apresentadas pelo aplicativo para ver mais detalhes. Ao fazer isso, o sistema guarda essa ação do usuário e recomenda itens relacionados ao pesquisado.

O sistema de recomendação é híbrido e formado pelos métodos CF e CBF. Para implementar o CF, o estudo usou a técnica ARM³, e para CBF, foi criada uma fórmula para relacionar a categoria e localização dos itens com interesses e localização do usuário.

Para avaliar o desempenho do sistema, foi utilizado o critério de satisfação do usuário. Para isso, foi aplicado um modelo de avaliação proposto por [Ong et al., 2009], com 4 categorias de perguntas: Facilidade de uso percebida, utilidade percebida, qualidade de serviço percebida e qualidade da informação percebida. Junto a essas o autor adicionou uma quinta categoria, chamada "sobrecarga de informação percebida". Cada categoria tinha um conjunto de perguntas com 5 opções de resposta em escala *likert*⁴. Um total de 72 usuários participaram da avaliação e o resultado mostrou que os usuários em média se sentiram satisfeitos em relação às 5 perspectivas sobre o sistema, obtendo uma nota próxima a 4 (de 1

³*Association Rule Mining* - Ferramenta de mineração de dados usada para encontrar relações em um banco de dados com muitas informações, através da busca de itens que aparecem, com frequência, juntos em algum contexto. [Agrawal et al., 1993].

⁴Concordo fortemente, concordo, não concordo nem discordo, discordo, discordo fortemente.

Sistema	Método de Recomendação	Função Semelhança	Tipo de Avaliação	Número Usuários
Guia CienTec	BC/CF kNN	PCC	On-line	76
CIDOC-CRM	CF kNN/CBF	PCC	Não Avaliado	N/A
ART <i>Recommender</i>	BC / CBF	Nenhuma	Estudo	48
SR para Grupos	CF MF	PCC	Estudo	29
PGR	CF/CBF	Nenhuma	Estudo	72
PLN	CBF	Cosseno	Off-line	Simulação

Tabela 3.1: Tabela comparativa deste trabalho com os artigos relacionados em relação aos algoritmos utilizados, número de participantes na avaliação e tipo de avaliação, segundo a classificação em 2.7.

a 5) em todas elas.

3.6 Aplicando PLN

[Mathias et al., 2014] utiliza técnicas de processamento de linguagem natural (PLN) para extrair características a partir do título, autor, descrição, entre outras informações do item, e diferenciar entre si os itens presentes em um museu. Essas características são então comparadas com as preferências do usuário usando a fórmula de semelhança do cosseno, formando assim o método CBF para recomendação de itens. As rotas são formuladas com base na aplicação de programação linear sobre um grafo, que representa o museu, onde os vértices são as salas e os arcos são os caminhos que ligam diferentes salas.

A avaliação do sistema foi feita a partir de dados gerados aleatoriamente. A relevância dos resultados obtidos entre usuário e item recomendado era calculada para definir o desempenho geral do sistema. O comportamento do usuário durante a visita não é levado em consideração, mas essa análise é citada como um possível trabalho futuro para o estudo.

3.7 Contribuição da Pesquisa

Esta pesquisa concebe e avalia um sistema de recomendação híbrido, baseado em contexto e no método CF baseado em usuário, utilizado através de um dispositivo móvel. Ele combina informações como a localização do usuário, o tempo que ele tem disponível para visita, preferências e avaliações prévias de itens. Essas informações são combinadas de forma harmoniosa a fim de apresentar uma rota recomendada que busca oferecer ao usuário uma visita que passe pelos itens que ele tem mais probabilidade de se interessar, minimizando o deslocamento necessário e dentro do limite de tempo proposto. A proposta do sistema é feita com o objetivo de trazer simplicidade para facilitar a adaptação em diversos locais de patrimônio cultural, evitando a necessidade de fornecer dados adicionais de itens apenas para um melhor funcionamento do SR.

Com esse mesmo propósito, a pesquisa emprega uma técnica para reduzir a dificuldade de aplicação de um SR do tipo CF devido ao *cold start*, através da aplicação de modelagem contextual no início da aplicação utilizando as informações de preferências do usuário, similar às técnicas propostas em [Wang et al., 2008] e [Cao et al., 2018], e idade, tempo e localização. Os resultados obtidos com essa técnica foram positivos em relação ao tratamento do *cold start* e também são apresentados neste trabalho.

Nesta pesquisa, a validação do SR proposto é feita em um ambiente real de utilização com uma quantidade considerável de usuários. A maioria dos trabalhos relacionados a aplicação de SRs em exposições de patrimônio cultural tiveram testes realizados de forma experimental em ambiente controlado, ambiente real com poucos usuários, simulações com dados off-line ou mesmo sem qualquer tipo de validação. O trabalho avalia o sistema em termos de acurácia das recomendações, satisfação do usuário, tanto com as recomendações como rota proposta, e desempenho, com relação ao tempo de processamento, obtido ao escalar o número de avaliações e itens na base de dados. Os resultados encontrados são apresentados no Capítulo 5.

A Tabela 3.1 compara os dados do sistema de recomendação e avaliação do trabalho atual em relação aos já apresentados nesse capítulo.

Capítulo 4

A Aplicação: Guia CienTec

Este capítulo trata de detalhar o problema que a pesquisa aborda assim como a solução proposta, com os algoritmos e ferramentas utilizados. O capítulo está dividido da seguinte forma: A Seção 4.1 tem foco em sintetizar o problema no qual a pesquisa se baseia. A Seção 4.2 relaciona como os desafios citados na Seção 2.8 se relacionam à aplicação desta pesquisa. Na Seção 4.3 são justificadas as decisões que definiram a estratégia macro da solução que atende o problema estabelecido, incluindo as bibliotecas e ferramentas escolhidas. Em 4.4 será apresentado em detalhes o sistema de recomendação aplicado, incluindo os métodos de recomendação e outros algoritmos utilizados na sua concepção. Por último, na Seção 4.5, é explicado como o usuário interage com o aplicativo desenvolvido, sua plataforma, interface e funcionalidades.

4.1 Caracterização do Problema

Sobrecarga de informação, como já dito anteriormente, é um problema encontrado facilmente em museus que, minimamente, possuem uma quantidade de itens de exposição maior do que a quantidade que o visitante está disposto a visitar. Isso não é diferente no Parque CienTec, local escolhido para a aplicação do sistema proposto. O Parque CienTec possui um conjunto de exposições de variadas áreas do conhecimento, como meteorologia, geologia, física e ótica. Essas exposições são compostas por experimentos interativos e objetos expositivos como, por exemplo, ferramentas antigas, rochas e fotografias. Além disso o parque é uma reserva ecológica, com 112 hectares de mata, declarada de caráter de preservação permanente¹, que conta também com um lago artificial e uma das nascentes do Riacho do Ipiranga.

Entre as exposições e outros atrativos do parque, há caminhos com distâncias consideráveis. Não há um mapa das localizações e poucas sinalizações estão disponíveis para auxiliar o visitante a ir até um lugar específico. A princípio, os visitantes são guiados por monitores do parque até as atrações, os quais também protegem os itens de possíveis avarias que os

¹O Parque <<https://parquecientec.usp.br/o-parque>>. Acessado em 1/out/2019.

visitantes possam cometer. Cada monitor tem uma ou mais especialidades relacionadas às exposições. Eles as apresentam, de acordo com suas especialidades, aos visitantes. A ordem de visitação é definida de acordo com a disponibilidade de cada monitor e quantidade de visitantes presentes. Além disso, há o planetário, onde são realizadas algumas sessões de apresentações sobre o espaço em horários específicos do dia.

Algumas dificuldades emergem em momentos de grande fluxo de visitantes, para que os monitores coordenem a ordem de visita de todos eles, que podem ter interesses diferentes ou já terem visitado alguns itens que os outros não visitaram. Outra limitação é que a operação do parque depende da disponibilidade de vários monitores. Isso impacta, por exemplo, na possibilidade de abrir o parque aos domingos. Em qualquer uma das duas situações, o visitante acabará tendo que decidir por si só a rota de visitação, se encaixando no problema clássico da aplicação de sistemas de recomendação.

4.2 Desafios Relacionados

Nesta seção serão apresentados *quais* os desafios citados na Seção 2.8 que se aplicam no contexto dessa pesquisa e *como* eles se aplicam. Para isso, será utilizada a mesma ordem numérica da Seção 2.8, onde podem ser consultados os detalhes do que se trata cada desafio.

4.2.1 Desafios para SRs aplicados em Museus

- 1. Distância:** O parque é um amplo complexo de exposições e espaços naturais, portanto a distância pode ser considerado um fator decisivo na escolha das recomendações. Na Seção 4.4 é explicado como o sistema busca minimizar a distância necessária para visitar todos os itens da rota recomendada.
- 2. Tempo disponível:** Como já foi mencionado, o parque tem um espaço grande e suas atrações estão espalhadas. Por isso não seria adequado deixar o controle do tempo totalmente a cargo do usuário, então o fator tempo disponível é levado em consideração na definição da rota recomendada. Mais informações sobre o funcionamento da montagem da rota está disponível na Seção 4.4.

Assim como a distância entre os itens impacta no tempo, há também uma variabilidade no tempo necessário para conhecer cada atração do parque. Assim, se o tempo considerado para visitar cada atração recomendada for um valor fixo, muito provavelmente ou o usuário vai terminar sua rota recomendada mais rápido do que o previsto, ou não conseguirá terminá-la, dependendo dos itens que tenha visitado. Para reduzir o impacto desse desafio, é possível definir tempos médios de visita específicos por atração. Assim, o gerador de *Van der Graaf*, por exemplo, que é um experimento físico que exige seguir um procedimento para testá-lo, pode ter um tempo de visita, considerado

pelo sistema, maior do que um objeto da exposição do museu de meteorologia, que não possui qualquer interação disponível para o usuário.

3. Disponibilidade dos itens: Vários dos espaços do parque não ficam disponíveis todos os dias. Isso depende da quantidade de monitores que estão disponíveis para atender os locais. Essa questão foi atendida com a criação de um atributo para os itens que define se eles estão disponíveis ou não naquele momento. Este atributo define se o item será ou não considerado para recomendação (ver algoritmo 3 - linha 11). Para facilitar o gerenciamento dos itens disponíveis, um aplicativo foi criado para os funcionários do parque com a opção de ativar/desativar cada atração.

Outra questão relacionada a esse desafio são as atrações que tem horários específicos de visita, como o planetário, que possui normalmente duas apresentações por dia, de 50 minutos cada. A solução do sistema para esse desafio foi de separar as atrações com essa característica dos itens considerados na rota recomendada. Os horários disponíveis de visita desses itens ficam expostos em uma tela separada e o próprio usuário controla quando deseja se dirigir ao item. Contudo, essa solução não é adequada em locais que existam muitos itens dessa natureza, pois demandaria uma parcela grande de responsabilidade do próprio usuário sobre o controle do tempo disponível, se caracterizando no desafio do tempo descrito nesta seção (2).

4. Ordem fixa de visitação: O parque CienTec não possui uma ordem fixa. Porém, como geralmente há mais exposições do que monitores disponíveis e é exigido, por motivos de segurança, que sempre haja algum monitor em uma exposição, quando um visitante se encontra nela, o formato padrão de visitação adotado pelo parque é de um direcionamento feito pelos monitores para que os visitantes se agrupem em uma rota em comum, diminuindo a quantidade de locais com visitantes presentes. Como a ordem fixa de visitação reduz os benefícios de um SR, mudanças na organização dos monitores foram propostas para que o sistema fosse implantado no parque: Nos dias com baixo fluxo de visitantes, cada monitor observava à distância um grupo de visitantes, e quando haviam muitos visitantes podia-se tomar a decisão de manter os monitores em pontos estratégicos, fechando outros locais que não pudessem ficar sem monitoramento, em caso de falta de monitores para cobrir todos eles.

5. Itens temporários: A grande maioria das atrações do Parque CienTec é de caráter permanente, por isso esse desafio não foi considerado para o desenvolvimento do sistema.

6. Sinergia entre itens: Embora exista uma sinergia entre itens em várias exposições do parque, essa característica não foi considerada a fim de simplificar a solução e permitir que a mesma solução seja adaptada para diversos outros locais de patrimônio cultural, sem muitas dificuldades. Além disso, dependendo da solução proposta, a identificação de sinergias entre itens poderia exigir um esforço considerável de curadoria para cadastrar essas relações na base de dados.

7. **Itens fora da rota:** Situação identificada como comum na utilização do aplicativo no parque, pois várias das atrações são visivelmente atraentes e ao seguir a rota de itens recomendada pode-se passar por diversos desses itens. A fim de tratar essa necessidade, o aplicativo desenvolvido permite adicionar outros itens para o grupo de itens recomendados (ver Seção 4.5).
8. **Visitas em grupos:** Padrão que também se aplica ao Parque CienTec. Apesar de haver propostas de SRs para lidar com grupos de visitantes [Rossi et al., 2016], a fim de manter uma solução mais simples de uso do sistema, optou-se por considerar as avaliações apenas de um usuário. Assim sendo, nos eventos de visitantes em grupo, é recomendado a eles que as avaliações sejam feitas considerando as preferências de todo o grupo ou, caso prefiram, os integrantes visitem o local de forma independente.
9. **Pontos de interesse:** Como as visitas não costumam ser tão longas e há muita variabilidade nas necessidades dos visitantes, somente atrações do parque são consideradas na rota recomendada.
10. **Dificuldade na obtenção de *feedbacks*:** Como a proposta principal do sistema é que o usuário siga a rota recomendada para ele, decidiu-se que, a partir do término da visita a um item da rota, seja requisitada a ele a avaliação, de 1 a 5, sobre o item visitado. Com isso, a experiência de uso é comprometida. Visando diminuir esse impacto, uma solução foi estabelecida para que o usuário realize a avaliação com apenas um clique.
11. **Sinal de Internet/GPS:** Este é um problema importante para a aplicação no Parque CienTec. O local não possui um sinal de rede móvel muito forte e a rede *wi-fi* está disponível em apenas uma parte pequena de toda a extensão do parque. O sinal de GPS é mais presente devido à organização dos itens do parque, sendo que muitos deles estão em um ambiente externo e os demais estão dispostos em pequenas salas separadas. Como as outras redes colaboram para a indicação da localização do usuário e elas tem baixa intensidade, a precisão na localização acaba sendo comprometida.

Para diminuir esse problema, todas as informações que precisam de *Internet* são obtidas no início da utilização do aplicativo. A única atividade que precisa de rede é para gravar as avaliações do usuário no banco de dados. Mais detalhes dessa solução estão presentes no tópico *Desempenho* da Seção 4.3.

4.2.2 Desafios para SRs em Geral

1. ***Cold Start*:** Embora seja um problema muito relevante em SRs baseados em CF, como o proposto nesta pesquisa, isto é tratado pela coleta de informações de contexto realizada no começo da visita. Entretanto, essa abordagem é útil apenas para novos visitantes. A dificuldade com novos itens adicionados ao sistema se mantém. Para contornar esse

problema, a solução escolhida foi a de recomendar o item com uma nota média até que se obtenham algumas avaliações daquele item. Também, como não há necessidade de muitas avaliações para que um novo item seja recomendado, pode-se pedir para que os funcionários do parque também avaliem aquele novo item. Assim, o novo item já estará sendo coberto pelo SR com maior assertividade.

- 2. Matrizes esparsas:** Este desafio é gerado tanto por um número alto de itens, como de visitantes. O Parque CienTec, assim como muitas outras exposições de patrimônio cultural, não possui uma quantidade de itens tão grande que possa inviabilizar a solução de SR com base em CF. Com relação ao número de visitantes, este pode aumentar indefinidamente. Por esse motivo um limite máximo de 250 usuário foi adicionado na função que carrega as avaliações da base de dados. A Seção 5.4 mostra o impacto nos tempos de execução à medida que novas avaliações são consideradas pelo aplicativo.
- 3. Fraudes:** Não foi criada uma funcionalidade para tratar deste desafio na versão do sistema utilizada nesta pesquisa. Isso pois, embora haja um risco, ele não é considerado alto devido a ausência de benefícios financeiros decorrente de uma possível manipulação nos resultados de recomendações para os visitantes do local de aplicação proposto.
- 4. Falta de Conteúdo:** Pouco se aplica, pois a grande maioria das atrações do local possuem informações de conhecimento geral amplamente disponíveis.
- 5. Desempenho:** Dada a quantidade de itens disponíveis e limite definido de usuários considerados pelo SR conforme item 2, o desempenho do sistema se mantém sob controle. O Capítulo 5 apresenta a progressão no tempo necessário de treino do SR e definição de rota recomendada em relação ao número de avaliações consideradas e itens disponíveis.

4.3 Decisões de Projeto

Durante o transcorrer desta pesquisa, especialmente nos momentos de estudo inicial e desenvolvimento do aplicativo, e nos momentos iniciais de testes do aplicativo no Parque CienTec, muitas lições foram aprendidas e decisões foram tomadas. As decisões mais importantes estão descritas nesta seção com o objetivo de deixar mais claro a evolução do trabalho e sanar possíveis dúvidas relacionadas à concepção da solução.

Local de aplicação

O Parque CienTec foi escolhido para a aplicação por trazer uma série de exposições geograficamente separadas entre si, fato que exige o fornecimento de um direcionamento para o usuário. Como a pesquisa se propunha a utilizar dados de localização de usuário para trazer uma melhor experiência, o Parque CienTec, com essa característica, se encaixava bem para a avaliação do sistema. Outro ponto positivo do parque para a pesquisa é a diversidade

de assuntos e atividades disponíveis. Esse fator deveria aumentar a importância de uma recomendação de rota que atenda aos interesses dos visitantes. Por exemplo, o usuário pode ter muito interesse em astronomia mas não gostar de geologia. Se o museu fosse totalmente relacionado apenas a astronomia, possivelmente não haveria tanta diferença de interesse entre os itens para esse usuário como nesse local.

Plataforma

O aplicativo foi desenvolvido para a plataforma *Android* através do ambiente de desenvolvimento *Android Studio*² e linguagem de programação *Kotlin*³. A escolha dessa plataforma se deve a predominância dela no mercado brasileiro de *smartphones* (cerca de 86% do mercado de celulares no país⁴) e também pela facilidade no desenvolvimento de solução *mobile* com aplicação de funcionalidades como GPS e integração com o *Google Maps* e serviços de *backend* do *Firebase*, além de serviços de *login*, câmera e escaneamento de código de barras, que embora não implementados até a escrita deste trabalho, têm possibilidade de virem a ser adicionados como novas funcionalidades do sistema.

A escolha da linguagem *Kotlin* foi feita pela sua capacidade de interoperabilidade com a linguagem *Java*, que por sua vez, possui uma numerosa base de bibliotecas *open source* disponíveis, incluindo a biblioteca de métodos de recomendação utilizada no sistema dessa pesquisa. Além disso, possui uma vantagem em relação ao *Java* de ser uma linguagem mais adaptada ao desenvolvimento para *Android*, por características sintáticas como no uso de interfaces e funções *lambda* e funcionalidades que apoiam desenvolvimento de sistemas com execuções assíncronas.

Método de Recomendação

Uma das preocupações na ideação do aplicativo foi a possibilidade de adaptá-lo com relativa facilidade para diversos outros locais onde haja múltiplos itens para escolher. Considerando essa premissa, os métodos de recomendação CBF e KB acabam não sendo os mais adequados para esta aplicação, pois exigem a construção de uma base de conhecimento sobre o conjunto de itens, seja de regras de inferência e conjuntos, no caso do KB, como atributos dos itens, no caso do CBF. Essa base geralmente é obtida/organizada pela coleta manual de informações e também se aplica apenas ao conjunto de itens para os quais foi levantada. Ou seja, para cada novo local de utilização do aplicativo, uma nova base teria que ser gerada.

Assim, para simplificar a adaptação do aplicativo, as abordagens baseada em contexto e filtragem colaborativa foram escolhidas, pois não necessitam de geração de muito conteúdo referente aos itens para sua aplicação. Contudo, este também tem seu próprio problema de

²<https://developer.android.com/studio/>

³<https://kotlinlang.org/>

⁴Segundo o site statcounter <<https://gs.statcounter.com/os-market-share/mobile/brazil>> no mês de setembro de 2019.

cold start, como explorado na Seção 2.6.2, que traz seus próprios desafios para utilização em novas exposições.

De modo a suavizar o *cold start* de novos usuários do sistema, um recurso foi adicionado ao sistema, que consiste em pedir algumas informações simples ao usuário assim que ele começa a utilizá-lo: idade, tempo disponível para visita e nível de interesse sobre algumas frases que referenciam categorias ou características encontradas em itens do local. Essas informações são transformadas em atributos para o método *CF* desenvolvido da seguinte forma: o tempo disponível de visita é usado como uma restrição de quantidade de itens recomendados para o visitante (mais informações em 4.4); a idade do usuário é convertida em um valor de 0 a 5, igual a escala de recomendações feitas pelo SR. Assim, a idade também é considerada pelo CF para gerar a recomendação. A equação 4.1 define a função de conversão da idade i para um valor real v do intervalo $[0,5]$.

$$v(i) = \begin{cases} i/12 & , \text{ se } i \leq 12 \\ 1 + (i - 12)/5 & , \text{ se } 13 \leq i \leq 17 \\ 2 + (i - 17)/7 & , \text{ se } 18 \leq i \leq 24 \\ 3 + (i - 24)/35 & , \text{ se } 25 \leq i \leq 59 \\ 4 + (i - 59)/41 & , \text{ se } 60 \leq i \leq 100 \\ 5 & , \text{ se } i > 100 \end{cases} \quad (4.1)$$

Essa equação foi criada para que casos de pessoas em fases da vida distintas, mesmo que com idades parecidas, tenham uma diferença maior no valor que pessoas na mesma fase da vida, mas com uma diferença de idade absoluta maior. Por exemplo, utilizando essa equação, para um adolescente de 13 anos, o valor resultante será de 1,2 e para um jovem de 24 anos o valor resultante será 3, diferença de 1,6 no resultado, com diferença de idade de 11 anos. Em outro exemplo com dois adultos, um de 25 e outro de 50 anos, os resultados serão de 3,03 e 3,74, respectivamente. A diferença no resultado nesse exemplo é de 0,71, menor que o primeiro exemplo, embora a diferença de idade seja muito maior, de 25 anos. O motivo para que as idades absolutas não sejam utilizadas diretamente se baseia na premissa de que pessoas em fases de vida distintas têm preferências muito mais distintas que pessoas que, embora tenham uma diferença de idade maior, estão na mesma fase de vida.

Em relação as frases para definição das preferências do usuário, no caso da aplicação no Parque CienTec, foram definidas sete:

1. O que é eletricidade e como ela funciona;
2. A água é composta de dois átomos de hidrogênio e um de oxigênio;
3. O diâmetro do Sol é mais de 100 vezes maior que o da terra;
4. O granito é um tipo de rocha que se origina de magma como o expelido por vulcões;

5. A bactéria é um tipo de micro-organismo que está presente em quase todos os lugares da terra;
6. Segundo a mitologia grega, Atena é a deusa da sabedoria;
7. Gosto de andar próximo da natureza.

Com esses dados já informados pelo usuário, ao ser requisitada a rota recomendada pela primeira vez, o SR já possui oito entradas do usuário para o SR considerar (o nível de interesse das sete frases mais a idade do usuário).

Prioridades da Recomendação

Dentre os atributos utilizados pelo SR, nota de recomendação, quantidade de itens, tempo e distância necessária, foi preciso estabelecer a ordem de prioridade entre eles para que haja um determinismo na escolha do subgrupo de itens que irá compor a rota recomendada. Essa prioridade define se o recomendador irá preferir, por exemplo, indicar três itens próximos com nota de recomendação média ou um item que se encontra longe do usuário, porém com nota de recomendação alta.

A ordem escolhida para o sistema foi de priorizar os itens com maior nota de recomendação, independentemente da quantidade que será possível visitar, respeitando o tempo disponível definido pelo usuário. Essa decisão foi feita tomando como premissa que os usuários iriam preferir ver poucos itens que, porém, são os mais interessantes para eles, do que ver muitos itens e deixar uma das melhores atrações de fora.

Desempenho

Como já mencionado no item 5 do Capítulo 2, sistemas de recomendação baseados no método filtragem colaborativa podem gerar alto consumo de memória e processamento. Soluções que suportem tanto custo de processamento acarretariam despesas extras de contratação de recursos para isso.

A fim de manter a solução simples e sem custos adicionais, a solução do sistema pretendida foi de manter o processamento do treino do SR, e geração da rota, no próprio aparelho móvel. Considerando a limitação da capacidade de processamento e memória desses dispositivos, a viabilidade de tal arquitetura se confirmou ao constatar-se o número de atrações disponíveis no local de aplicação que faziam sentido serem considerados no aplicativo (≈ 100) e a possibilidade de limitar o número de usuários com avaliações a serem consideradas pelo SR. Testes de desempenho para validar a viabilidade da aplicação em locais com mais conteúdo ou mais avaliações foram feitos e são apresentados no Capítulo 5.

Outra característica avaliada para o desempenho do aplicativo foi o armazenamento de dados da nuvem, tanto pelo uso de dados móveis como pelo tempo de recuperação da informação. Ainda costuma ser baixa a capacidade de *download/upload* dos planos móveis

mais populares no país. Assim, um consumo extenso desse recurso pode inviabilizar o uso do aplicativo para uma série de usuários. O tempo necessário para recuperar os dados geralmente não é um problema, mas se considerar aplicações que utilizem centenas de milhares de avaliações, o tempo pode ser maior que o aceitável. Nesses casos o treinamento/predição da recomendação dos itens poderia ser totalmente feita em um servidor externo.

Mesmo que o tempo e quantidade de dados não sejam um problema, a recuperação de dados da nuvem exige conexão com a *Internet*. No caso do Parque CienTec, o sinal de redes móveis e *wi-fi*, como explicado em 4.2, apresenta baixo sinal na região do parque como um todo, com indisponibilidade em diversos locais. Considerando essas dificuldades, decidiu-se manter as imagens dos itens dentro do próprio pacote do aplicativo e todas as informações atualizadas dos itens e avaliações, que são necessárias para o funcionamento do sistema, são recuperados, do banco de dados na nuvem, no início de sua utilização. A única interação com a *Internet* feita pelo aplicativo durante o andamento da visita é ao avaliar algum item ou responder a avaliação de satisfação. Mesmo assim, caso a *Internet* não esteja disponível no momento do *feedback*, as informações são guardadas em *cache* até que a conexão com a *Internet* seja reestabelecida.

Backend

O aplicativo é suportado por um *backend* formado por alguns dos serviços da plataforma do *Google*, *Firebase: Cloud Firestore*⁵, *Cloud Storage*⁶ e *Crashlytics*⁷. Essas ferramentas foram escolhidas pela facilidade de implantação em aplicativos Android e na comunicação entre elas.

O *Cloud Firestore* é um banco de dados *NoSQL* disponível na nuvem que é utilizado para manter as informações de usuários, itens e avaliações. Uma funcionalidade importante desse banco de dados, e sua biblioteca para *Kotlin - Android*, é que ele já possui um sistema de cache padrão, que permite que requisições de envio de dados para o banco de dados fiquem guardadas até que uma conexão com a *Internet* esteja disponível. Essa característica é importante para tratar do problema de rede descrito em 4.3.

Usuários, itens, avaliações e questionário são as quatro coleções presentes no banco de dados. Questionário é a coleção que contém as respostas dos usuários sobre o questionário de satisfação. Nas coleções de usuários e itens, cada documento representa um usuário ou item, respectivamente. No caso das avaliações, inicialmente foi implementado da mesma forma que as demais coleções. Mas devido à numerosa quantidade de avaliações que precisavam ser lidas para o treinamento do SR e o custo, do serviço, associado a essas leituras, mudou-se a organização da coleção, agrupando as avaliações por usuário. Assim otimizou-se o número de leituras necessárias. Por exemplo, em uma situação onde houvesse 100 pessoas e 1200 avaliações no banco de dados, a quantidade de leituras necessárias passou de 1200 para 100

⁵<https://firebase.google.com/docs/firestore/>

⁶<https://firebase.google.com/docs/storage>

⁷<https://firebase.google.com/docs/crashlytics>

com essa otimização.

O *Cloud Storage* é utilizado para manter os documentos com os termos de uso e políticas de privacidade do aplicativo e o *Crashlytics* é usado para verificar *logs* e estatísticas de erros que os usuários venham a ter com o uso do aplicativo.

Persistência de Dados do Usuário

Para eliminar a complexidade no tratamento, armazenamento e condições de uso de informações pessoais de usuários, a possibilidade de criação de perfil do usuário e *login* foi desconsiderada nas versões do aplicativo utilizadas nessa pesquisa. Deste modo, apenas informações relevantes para fornecer recomendação para o usuário são salvas, como a idade do usuário, o tempo que ele tem disponível de visita, suas preferências iniciais e seus *feedbacks* de avaliação dos itens visitados. A explicação sobre como essas informações são utilizadas para obter a recomendação podem ser encontradas na Seção 4.4. As informações de uso do aplicativo são armazenadas no banco de dados *Firebase Firestore* e também no próprio celular.

Como não há possibilidade de criar uma conta de usuário existe o risco de que duas pessoas distintas visitem o aplicativo usando um mesmo celular, o que acarretaria em incoerências nas recomendações feitas ao segundo usuário, pois o aplicativo estaria considerando as preferências do primeiro. Esse problema afetaria a acurácia das recomendações pro usuário, conseqüentemente, impactando nas conclusões desse trabalho. A fim de evitar essa situação, após o usuário declarar que concluiu a visita, os dados de preferências e avaliações que ele informou são removidos do celular e o aplicativo volta a tela inicial para um novo usuário poder começar a visita.

Geração da Rota

Definidos os itens a serem visitados, basta montar a rota para visitá-los. Porém, encontrar a melhor rota que passe por todas as atrações recomendadas pode ser caracterizado como o clássico problema do caixeiro viajante⁸. Como a solução deste problema tem complexidade NP-Difícil, o problema foi simplificado para o algoritmo guloso *Nearest Neighbor Search* (definições no Apêndice A) assim como proposto em [van Hage et al., 2010], que encontra uma solução sub-ótima do problema original.

Outro problema para gerar a rota é a apresentação dela no aplicativo. O Parque CienTec é extenso, não possuía um mapa próprio disponível e o mapa da API do *Google Maps* continha poucos detalhes, dificultando a navegação. Além disso, os caminhos entre os itens em muitos casos não são linhas retas. Para contornar esse problema, coordenadas geográficas intermediárias entre as atrações foram definidas e adicionadas ao conjunto de vértices de um grafo representando o local. Entre cada par de coordenadas que continham um caminho,

⁸Problema que propõe encontrar a rota de menor custo através de todos os nós de um grafo ponderado e voltar ao nó inicial.

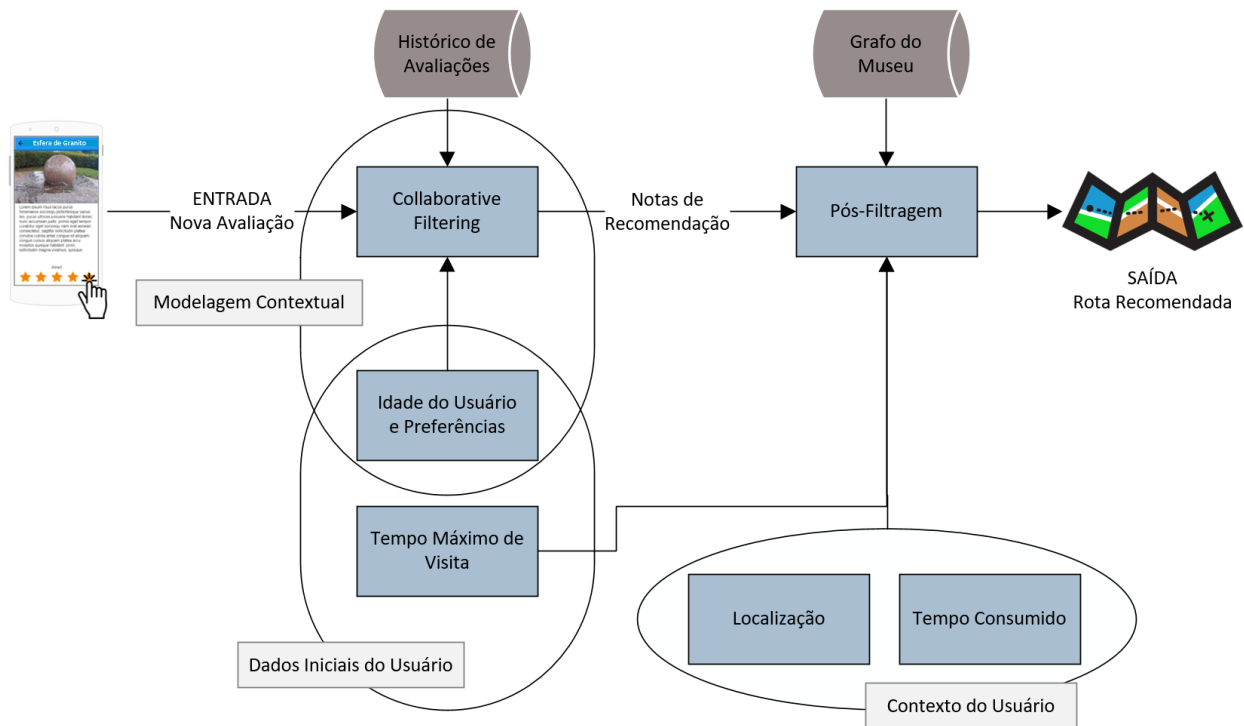


Figura 4.1: Principais Componentes do Sistema de Recomendação.

uma aresta foi criada. Em relação ao mapa do parque, uma ilustração foi feita, detalhando os caminhos disponíveis, e foi posicionada sobre o mapa do *Google Maps* para facilitar a navegação do usuário.

O uso da API do *Google Maps* foi definido pela facilidade de sua implementação para a plataforma *Android* e suas funcionalidades, como movimentar, ampliar e rotacionar o mapa, sobrepor imagens ao mapa e adicionar ícone interativos, textos, linhas e polígonos formatáveis.

4.4 O Sistema de Recomendação

O objetivo do SR proposto é de fornecer as atrações mais recomendadas para o usuário em uma sequência que lhe permita visitá-las minimizando o tempo gasto no caminho entre elas. Nesta seção será explicado como o sistema trabalha para atingir esse objetivo, primeiro trazendo uma visão macro do sistema, com todos os recursos envolvidos, para então descrever detalhadamente cada recurso separadamente e como eles se relacionam.

4.4.1 Arquitetura do SR

O sistema deste trabalho foi construído com a combinação de um sistema CF *memory-based* e técnicas provenientes de SR baseado em contexto. Das entradas fornecidas ao sistema até a conclusão da geração da rota, com os itens recomendados para o usuário, os dados passam essencialmente por duas etapas, como mostra a figura 4.1.

O fluxo principal do sistema começa com uma nova avaliação sendo realizada pelo usuário no aplicativo. Essa ação provoca o retreino do sistema central de recomendação, que é a primeira etapa do processo. Esse sistema consiste no método CF já mencionado em conjunto com a modelagem contextual. O contexto aqui considerado são as informações de idade e preferências do usuário, que são requisitadas ao usuário no momento inicial de uso do sistema. O sistema utiliza o histórico de avaliações disponível na base de dados no *Firestore*, que inclui as preferências iniciais do usuário, juntamente com a nova entrada, para produzir notas previstas de satisfação do usuário em relação a cada um dos itens disponíveis.

As notas de recomendação indicam a ordem de recomendação dos itens para o usuário ativo. Isto é, quanto maior a nota prevista de satisfação do usuário, mais recomendado é aquele item.

Esses valores então passam para a segunda etapa do sistema, a pós-filtragem contextual. Essa etapa tem o objetivo de limitar uma quantidade de itens, de modo que seja viável visitá-los dentro da restrição de tempo informada pelo usuário e dispô-los em uma ordem eficiente de visita. Para realizar essa tarefa, o sistema consome as notas previstas na etapa anterior, a atual localização do usuário, os caminhos existentes no parque e o tempo restante disponível de visita.

Tanto o histórico de avaliações como o grafo do parque, que contém seus itens e caminhos, são carregados do banco de dados no início de uso do aplicativo. Isso é feito devido ao problema de indisponibilidade de rede mencionado na Seção (2.8).

4.4.2 Etapa 1: Modelagem Contextual com Filtragem Colaborativa

O objetivo desse passo do sistema é o de obter as notas de recomendação de cada item para o usuário ativo. O sistema começa a agir quando uma nova avaliação é feita pelo usuário. Se é a primeira vez que ele acessa o sistema, a primeira recomendação será feita considerando somente os atributos de contexto, que são: os níveis de interesse do usuário sobre temas relacionados ao local de aplicação e a idade convertida no intervalo $[0,5]$, ambos apresentados na Seção Método de Recomendação, em 4.3.

Na aplicação no parque CienTec foram proposto sete frases para representar conceitos relacionados às atrações do parque. Chegou-se a esse número específico de frases para que se atingisse um equilíbrio. Não muito poucas, de modo que se obtenha um entendimento mínimo dos interesses do usuário, mas também não muito numerosas, para que o usuário não se canse de responder o questionário e acabe desistindo do uso do aplicativo. Independente se é a primeira recomendação, somente com dados de contexto, ou a adição de uma nova avaliação feita pelo usuário a um item, o método CF kNN implementado gerará as notas de recomendação da mesma forma.

O método foi implementado no sistema através da adaptação da biblioteca java Librec⁹ (licença GPL). Ela contém uma série de classes que implementam diversos dos sistemas

⁹<https://www.librec.net/>

de recomendação baseados na relação usuário-item, além de uma série de métricas para avaliá-los [Guo et al., 2015].

Foram três adaptações feitas na biblioteca para que ela pudesse ser integrada ao sistema: a alteração da entrada da base de dados de avaliações, a remoção de funções que utilizavam a biblioteca *java.nio.file* e a implementação de uma configuração que não separe as avaliações em dados de treino e dados de teste. A entrada das informações na implementação original era através de arquivos texto mantidos localmente. As classes alteradas são responsáveis por organizar as entradas no formato necessário para o correto funcionamento dos métodos de recomendação. A leitura de arquivo estático foi removida e a função passou a receber a base vinda diretamente de um novo parâmetro criado nela. Esse parâmetro é a estrutura de dados contendo o histórico de avaliações dos usuários recuperada da base de dados no *Firestore*. A segunda alteração, remoção do uso da biblioteca *java.nio.file*, foi necessária para manter a compatibilidade com aparelhos *Android* com versões mais antigas do SO do que a 8.0 (API *build 26*)¹⁰, pois a biblioteca *java.nio.file* foi integrada nessa versão¹¹ e ainda há muitos usuários de versões mais antigas do *Android*.

O processamento feito por todos os métodos de recomendação dessa biblioteca, utilizados nessa pesquisa, para gerar as notas de recomendação dos itens, seguem os mesmos três passos: Construção do modelo de dados, construção da matriz de semelhança e treinamento do recomendador. O primeiro passo é responsável por gerar as matrizes usuário-item com a lista de avaliações de entrada. Nessa etapa também que comumente as avaliações são divididas nos dados de treinamento e de teste. A terceira adaptação feita na biblioteca foi para que esses dados não sejam divididos, de modo que todos eles sejam usados para treinar o recomendador.

O segundo passo é a formação de uma matriz de semelhanças usuário-usuário ou item-item, dependendo do método adotado, a partir da matriz de treino. Nessa etapa foi utilizada a equação de Semelhança do Cosseno ou Correlação de Pearson (PCC), dependendo do sistema proposto, para encontrar as semelhanças entre os itens, no caso do CF baseado em item, ou entre os usuários, no CF baseado em usuário.

O último passo realiza efetivamente a predição da satisfação do usuário para cada um dos itens disponíveis, com o uso de um vetor com os usuários (ou itens, no baseado em item) mais similares com o usuário (ou item) em questão. O algoritmo 1 apresenta a função de predição da satisfação do usuário ativo com todos os itens, considerando o método CF baseado em usuário. O código utiliza a matriz de semelhanças para encontrar os **k** usuários mais semelhantes com o usuário ativo para cada item e que tenham o avaliado. Nesta pesquisa várias variações do método foram implementadas para comparar o desempenho entre elas. A diferença entre as variações está no valor definido para **k**, função de semelhança e base (usuário ou item). Na Seção 5.5 essa comparação é apresentada.

¹⁰Lista de APIs e versões do SO: <https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>. Acessado em 03/nov/2019.

¹¹<https://developer.android.com/reference/java/nio/file/package-summary>

Algoritmo 1 Método kNN baseado em usuário

```

1: função OBTENOTASDERECOMENDAÇÃO
2:   para cada item na itemList faça
3:     userSet = todos os usuários que avaliaram item
4:     kUserList = lista em branco
5:     para cada usuarioSimilar na lista de usuários ordenada por nível de semelhança
6:       faça
7:         se usuarioSimilar existe em userSet então
8:           adiciona usuarioSimilar em kUserList
9:           se kUserList.tamanho = k então break for
10:        fim se
11:     fim para
12:     soma = 0, valorPonderacao = 0
13:     para cada usuário em kUserList faça
14:       sim = nível de semelhança entre usuário e usuário ativo
15:       soma = soma + sim * (avaliação do usuário - média das avaliações do usuário)
16:       valorPonderacao = valorPonderacao + sim
17:     fim para
18:     avaliaçãoPrevista(item) = média de avaliações do usuário ativo + soma / valor-
19:     Ponderacao
fim para

```

O valor de predição retornado pelo algoritmo se trata da média de avaliações que o usuário ativo fez, adicionado à média, ponderada pela similaridade entre os usuários, da diferença entre a avaliação de cada usuário pela sua própria média (linhas 12 a 18 do algoritmo 1). As médias são utilizadas no algoritmo pelo fato de usuários realizarem avaliações em escalas diferentes [Aggarwal, 2016]. Por exemplo, há usuários que, se não gostaram muito de algum item, ao avaliá-lo, numa escala de 1 a 5, dão nota 4. Outros, nesse mesmo cenário, podem ter a característica de avaliar de forma mais contundente, com notas 1 ou 2.

O algoritmo do método CF baseado em item tem a mesma estrutura do código 4.1, bastando trocar a lista de usuários que avaliaram o item pela lista de itens que o usuário avaliou e a lista de usuários mais similares pela lista de itens mais similares.

4.4.3 Etapa 2: Pós-filtragem Contextual

Com as avaliações previstas de satisfação do usuário ativo, passa-se para a segunda etapa da recomendação, que é a definição do tamanho e ordem de visita da lista de itens a visitar. Para ficar mais claro como isso é feito, primeiramente serão detalhados os tipos de itens definidos e como o sistema os considera. Em seguida o grafo que representa o parque será definido, para, por fim, expor o algoritmo desenvolvido para combinar todas essas informações com o objetivo de gerar a rota de recomendação ao usuário.

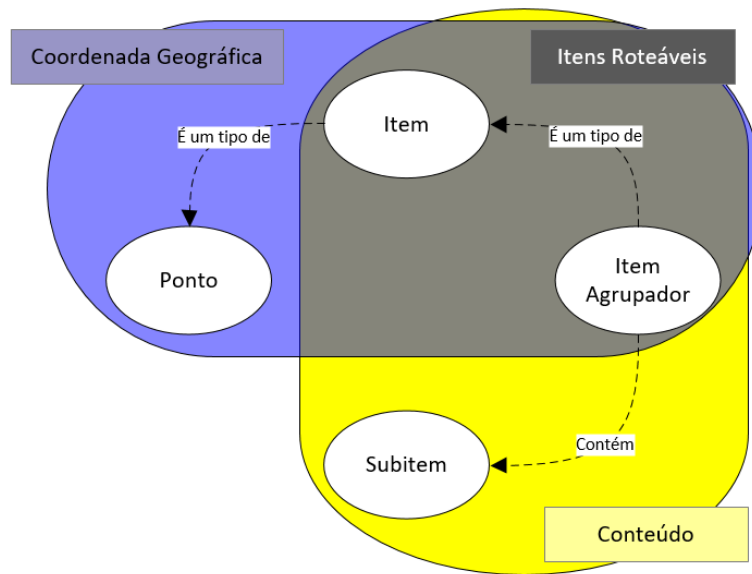


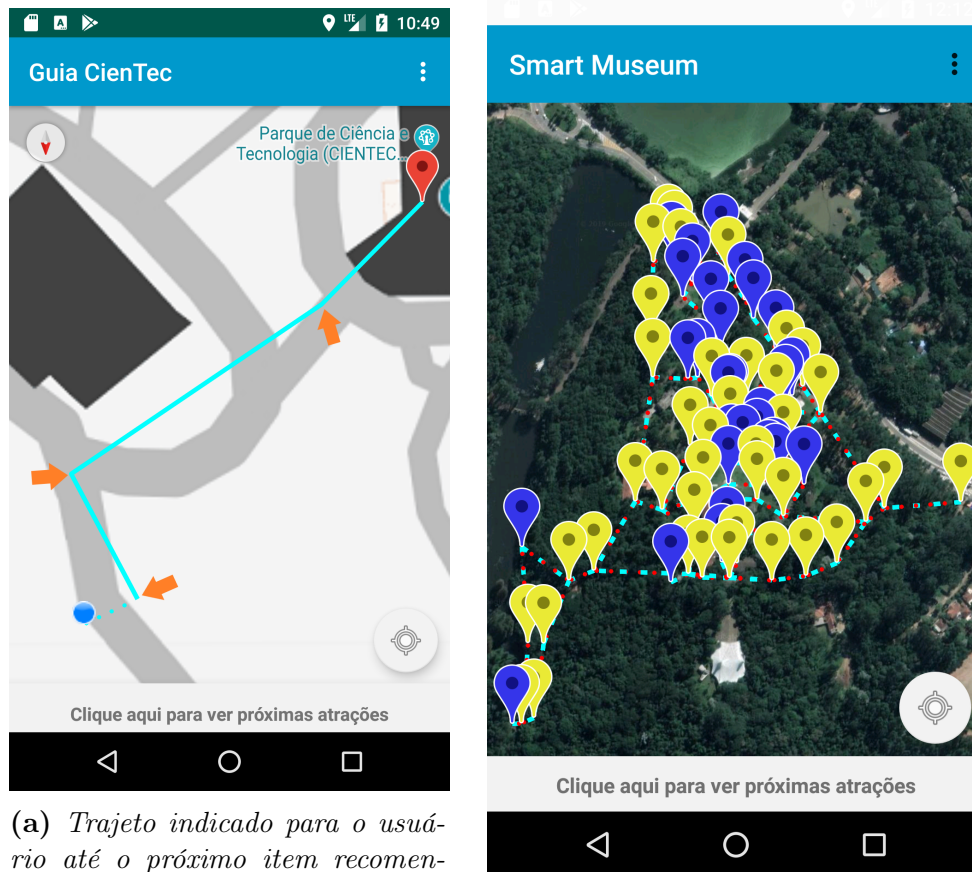
Figura 4.2: *Tipos de Elementos do Sistema*

Tipos de Itens

Os espaços de exposição de patrimônio cultural são geralmente formados por exposições, que agrupam uma série de itens por algum critério específico. Adicionalmente, há algumas atrações que, por seu tamanho, importância, necessidade de espaço ou algum outro fator, ficam dispostos fisicamente bem separados de outros itens. Como o sistema tem o objetivo de gerar a rota recomendada, efetivamente mostrando como chegar até determinado item, era uma necessidade considerar essas diferenças na sua concepção e por isso estabeleceu-se alguns tipos de itens. A figura 4.2 ilustra os tipos definidos. A partir daqui, um item que pode ser de qualquer uma das quatro categorias será chamado de **elemento**.

Um elemento que está geograficamente separado, seja por estar sozinho em uma sala ou visivelmente bem separado dos demais elementos de um local, é considerado do tipo **item**. Elementos de conteúdo do parque que estão próximos uns dos outros e que fazem parte de uma mesma exposição são considerados *subitens*. A exposição que é formada pelo conjunto de subitens é um **item agrupador**. A única diferença entre um item e um item agrupador é que o segundo possui subitens. Esses três tipos de elementos formam a categoria **conteúdo**, pois todos eles possuem título, descrição, imagem associada, podem ser recomendados aos visitantes e receber avaliações. Há ainda uma subcategoria do conteúdo, formada pelos itens e itens agrupadores, que são os **itens roteáveis**. Eles se diferem dos subitens por possuírem coordenadas geográficas e serem considerados na rota recomendada ao usuário.

Um último tipo de elemento definido é o tipo **ponto**. Ele também possui uma coordenada geográfica associada e é utilizado como intermediário entre os itens para moldar um caminho mais preciso da rota do visitante até o próximo item recomendado. A figura 4.3a ilustra esse benefício com um exemplo. Sem os pontos, a rota traçada no mapa mostraria apenas uma linha reta entre a posição atual do visitante (círculo azul) e o próximo item (ícone indicador



(a) Trajeto indicado para o usuário até o próximo item recomendado. Composto por 3 pontos (indicados pelas setas) mais o item

(b) Coordenadas geográficas e caminhos mapeados do Parque CienTec

Figura 4.3: Exemplo de uso de elementos do tipo ponto e visão das coordenadas geográficas mapeadas

vermelho). Juntos, pontos, itens e itens agrupadores formam a categoria coordenada geográfica, pois todos possuem latitudes e longitudes. A figura 4.3b mostra todas as coordenadas geográficas mapeadas, onde os ícones em amarelo representam os pontos e os em azul, os itens e itens agrupadores.

Grafo de Representação do Local

O local de aplicação, com seus elementos e caminhos, foi representado por um grafo que pode ser definido pela 3-upla $G = (V, A, E)$, onde:

- V é o conjunto de vértices do grafo, que são todos os elementos do tipo coordenada geográfica.
- A é o conjunto de arestas, onde cada $a = (i, j, d)$, com $a \in A$ e $i, j \in V$, representa um caminho existente no local que liga o vértice i ao vértice j com uma distância d .
- E é o conjunto de entradas do local, onde $E \subseteq V$. Esse conjunto foi necessário para os casos onde o usuário não está dentro do local, assim a rota recomendada começa pela entrada mais próxima da posição do usuário.

Gerador da Rota de Recomendação

Dada a lista de elementos de conteúdo mais recomendados para o usuário, resta verificar quantos deles podem ser visitados dentro do tempo disponível e assim definir qual a rota proposta para visitá-los. Entretanto, a ordem de visita definida também impacta no tempo necessário de visita, que, por conseguinte, pode alterar o número de elementos que podem ser visitados dentro da restrição de tempo. Portanto a melhor rota recomendada é aquela que otimize tanto o número de itens a visitar quanto a distância percorrida para chegar até eles.

Esta etapa do SR se propõe a isso. Ela é constituída pelo algoritmo 2, uma função que utiliza a lista de elementos de conteúdo (E), ordenada de forma decrescente pelas notas de recomendação encontradas na etapa anterior, e gera uma lista de *elementos* (L) dispostos por ordem recomendada de visita. Além da lista de elementos, a função recebe a localização atual do usuário (*posição do usuário*), e o tempo restante disponível de visita (T_d), que é a diferença entre o tempo disponível definido pelo usuário no início da visita e o tempo que já foi consumido.

O algoritmo inicia chamando a função *ConfiguraçõesIniciais*, apresentada no algoritmo 3, para recuperar os itens roteáveis que o usuário adicionou manualmente à rota de visita (detalhes da funcionalidade na Seção 4.5). Ele também obtém o elemento, do tipo *coordenada geográfica*, mais próximo do usuário (*início da rota*). Por fim a função adiciona ao tempo total (T_n) o tempo necessário para visitar os elementos adicionados, sem considerar o tempo de traslado entre eles. Se esse total de tempo já ultrapassar o tempo disponível T_d , a função principal conclui retornando L .

Caso ainda haja tempo disponível para mais elementos de conteúdo, a função *AdicionarItensPrioritários* é chamada. Ela verifica para cada elemento de E , ordenada por recomendação, se é possível adicioná-lo a L sem que ultrapasse o tempo disponível. Os elementos considerados não podem ter sido já visitados, estarem com status fechado ou removido, ou ter horário específico de funcionamento. Para cada elemento adicionado, seu tempo de visita é somado em T_n .

Em seguida o algoritmo chama a função *AdicionarItensAgrupadores*. Essa função garante que os subitens presentes em L tenham seus respectivos itens agrupadores também dentro de L . Também é confirmado que cada item agrupador esteja ao menos uma posição antes do seu primeiro subitem na ordem dos itens em L . Isso é importante porque os subitens não são roteáveis pelo mapa, então direciona-se o usuário para seu item agrupador.

Nesse momento, a quantidade máxima de elementos de conteúdo recomendados para o usuário dentro do seu tempo limite já está definida. O restante do algoritmo é para verificar se, utilizando uma rota eficiente entre todos esses elementos, é possível visitá-los somando o tempo necessário para visitá-los T_{itens} (que nesse momento é igual a T_n) ao tempo de deslocamento entre eles.

Para cumprir essa tarefa, considera-se primeiramente todos os elementos em L . O tempo

Algoritmo 2 Função principal de obtenção da rota recomendada

E = lista de elementos de conteúdo, T_n = tempo total necessário, T_d = tempo disponível

```

1: função OBTERROTARECOMENDADA( $E$ , posição do usuário,  $T_d$ )
2:   ( $L$ ,  $T_n$ , início da rota) = ConfiguraçõesIniciais( $E$ , posição do usuário)
3:   se  $T_n \geq T_d$  então retorna  $L$  ▷  $L$  = Lista recomendada
4:   fim se
5:   AdicionarItensPrioritários( $L$ ,  $T_n$ ,  $E$ )
6:   AdicionarItensAgrupadores( $L$ ,  $E$ )
7:    $T_{itens} = T_n$ 
8:   para  $i \leftarrow L.tamanho$  até 1 faça
9:     tempo foi suficiente = true, pontoInicial = início da rota
10:    para  $j \leftarrow 1$  até  $i$  faça ▷ Verificar Tempo da Rota do primeiro ao  $i$ -ésimo item
11:      proximoItem = ObterItemMaisPertoDaLista(pontoInicial,  $L$ )
12:      se proximoItem existe então pontoInicial = proximoItem
13:        se  $T_n + T_{pi} \leq T_d$  então
14:           $T_n = T_n + T_{pi}$  ▷  $T_{pi}$  é o tempo necessário para ver proximoItem
15:          proximoItem.foiVisitado = true
16:          proximoItem.ordemRecomendada =  $j$ 
17:        senão tempo suficiente = false
18:        break for
19:      fim se
20:    fim se
21:  fim para
22:  para cada  $L_i \in L$  faça
23:     $L_i.foiVisitado = \text{False}$ 
24:    se (tempo foi suficiente = false) então  $L_i.ordemRecomendada = 0$ 
25:  fim se
26:  fim para
27:  Remove o último item de  $L$  e subtrai de  $T_{itens}$  o tempo de visita dele
28:   $T_n = T_{itens}$ 
29:  fim para
30:  retorna  $L$ 
31: fim função=0

```

de deslocamento T_{pi} , do início da rota até o item roteável em L mais próximo, é calculado através da função *ObterItemMaisPertoDaLista*. Se $T_n + T_{pi} \leq T_d$, guarda-se a informação de que esse item poderá ser o primeiro da rota recomendada. A seguir verifica-se o tempo de deslocamento entre este item e o item roteável em L que esteja mais perto dele e que ainda não tenha sido "visitado" pela função. Repete-se esse processo até que $T_n + T_{pi}$ ultrapasse o tempo disponível ou até que todos os itens roteáveis em L tenham sido visitados.

Se o tempo disponível não tiver sido ultrapassado depois de visitar todos os itens roteáveis em L , está definida a rota recomendada e sua ordem. Caso tenha ultrapassado, remove-se o último elemento de L (o menos recomendado), iguala novamente T_n a T_{itens} subtraído do tempo de visita do elemento que acabou de ser removido e repete-se a tentativa de montar uma rota dentro do tempo disponível.

Algoritmo 3 Funções auxiliares de obtenção da rota recomendada

```

1: função CONFIGURAÇÕESINICIAIS( $E$ , posição do usuário)
2:   se posição do usuário está dentro do local de aplicação então
3:     início da rota = ponto mais próxima da posição do usuário
4:   senão início da rota = entrada mais próxima da posição do usuário
5:   fim se
6:   adicione todos os itens manualmente adicionados pelo usuário a  $L$ 
7:    $T_n = \sum_{i=1}^{I_a} (t_i + 0,4)$   $\triangleright I_a$  é o conjunto de itens roteáveis manualmente adicionados
   pelo usuário e  $t_i$  é o tempo necessário para visitar o item  $i$ .
8:   retorna ( $L$ ,  $T_n$ , início da rota)
9: fim função
10: função ADICIONARITENSPRIORITÁRIOS( $L$ ,  $T_n$ ,  $E$ )
11:   para  $i \leftarrow (i \in E, i$  não é [visitado, fechado, removido, tem horário específico]) até
    $T_n + T_i > T_d$  faça
12:      $T_n = T_n + T_i$ 
13:     Adiciona  $i$  na lista  $L$ 
14:   fim para
15: fim função
16: função ADICIONARITENSAGRUPADORES( $L$ ,  $T_n$ ,  $E$ )
17:   para cada  $i$ , onde  $i \in E$ ,  $i$  é um subitem faça
18:      $A = i.itemAgrupador$ 
19:     se  $a \notin L$  então Adiciona  $A$  em  $L$  na posição anterior a  $i$ 
20:     senão
21:       se  $L.posição(A) > L.posição(i)$  então traz  $A$  para posição anterior a  $i$ 
22:     fim se
23:   fim se
24: fim para
25: fim função

```

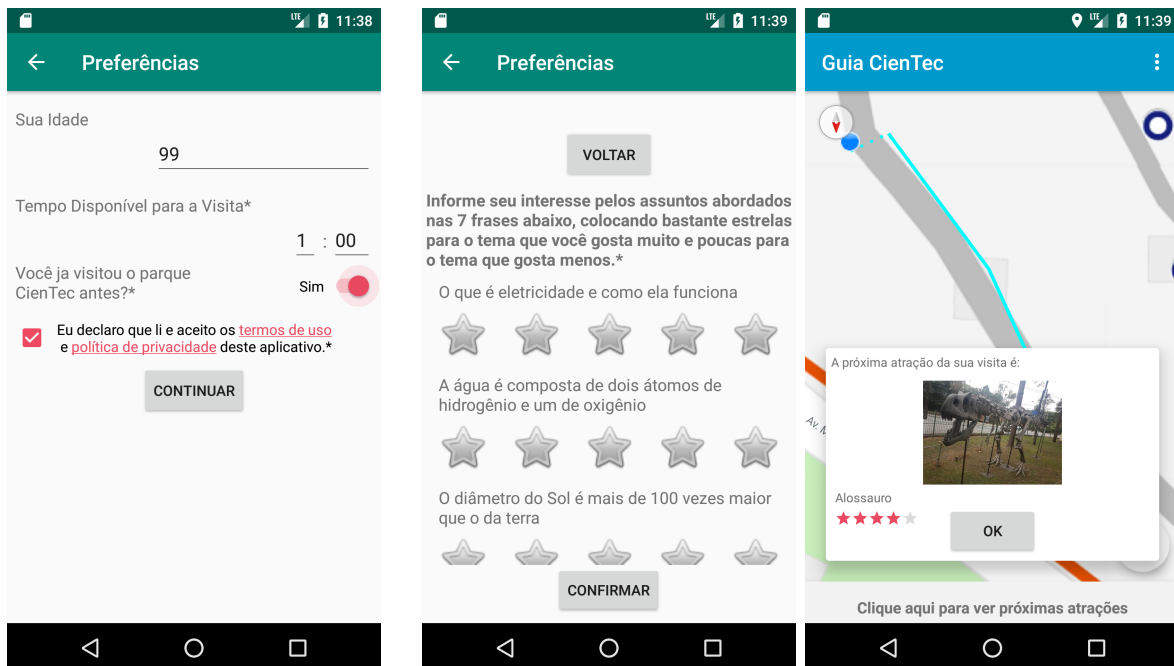
Esse algoritmo garante que uma rota com os elementos de conteúdo mais recomendados para o usuário seja criada, considerando tanto o tempo de visita de cada elemento como o deslocamento entre eles. Ela busca maximizar a quantidade de elementos que possam ser visitados considerando um deslocamento eficiente entre eles.

A função *ObterItemMaisPertoDaLista*, que busca qual item roteável de uma lista está mais próximo de outro dado item, é implementada com um algoritmo guloso do *Nearest Neighbor Search*. Ele utiliza o grafo do parque e uma implementação do clássico algoritmo de *Dijkstra* (definição no Apêndice A.3). Sempre que um item é visitado pelo algoritmo, verifica-se se ele está dentro da lista de itens recomendados e ainda não foi visitado pelo usuário. O primeiro item visitado que satisfaz essas características é retornado.

4.5 Detalhes do Aplicativo

Esta seção apresenta o aplicativo desenvolvido para suportar o sistema de recomendação e as interações com o usuário. As funcionalidades serão apresentadas seguindo o fluxo natural

de uso. O aplicativo está disponível para *download* no site do *Google Play*¹² e o código do aplicativo está disponível em repositório público¹³ licenciado pela *MIT License*¹⁴.



(a) *Informações de contexto do usuário*

(b) *Questionário de preferências*

(c) *Apresentação do próximo item da rota*

Figura 4.4: *Telas do aplicativo para entrada de informações do usuário e próximo item da rota*

4.5.1 Fluxo de Uso

A primeira tela apresentada ao usuário é a de inclusão das informações de contexto (figura 4.4a): idade e tempo de visita, além da informação se o usuário já conhece o parque e os links para os termos de uso e política de privacidade e opção para aceitá-los. Ao clicar em continuar, é mostrado o questionário de preferências pessoais (figura 4.4b). Com exceção da idade, todas as outras informações são de preenchimento obrigatório para continuar.

Após o usuário preencher as informações obrigatórias e clicar em confirmar, o sistema de recomendação utiliza as informações para seu primeiro treinamento e geração da rota recomendada. A próxima tela apresentada ao usuário apresenta informações do primeiro item da rota (figura 4.4c). Ao clicar em "OK", o aplicativo enquadra a posição do usuário e o item de destino na tela, onde é também apresentado o trajeto recomendado do usuário até o item com uma linha azul sobre o mapa do parque. No mapa a posição do usuário é representada por um círculo azul e o destino com um marcador vermelho. Caso algum item já tenha sido visitado, ele aparecerá no mapa como um marcador verde.

O aplicativo apresentava inicialmente o mapa com visão de satélite disponível na API do

¹²<https://play.google.com/store/apps/details?id=flaskoski.rs.smartmuseum>

¹³<https://github.com/flaskoski/SmartMuseum>

¹⁴Tipo de licença para softwares *open source*. Mais detalhes em: <https://opensource.org/licenses/MIT>

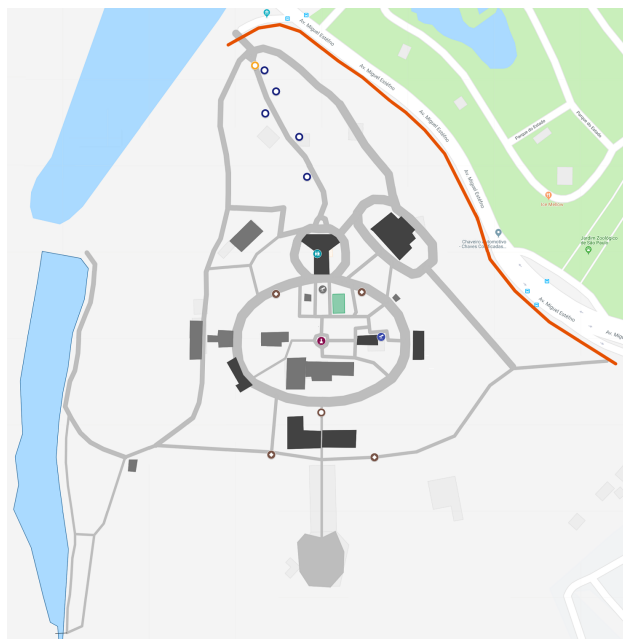


Figura 4.5: Mapa do parque construído para o aplicativo

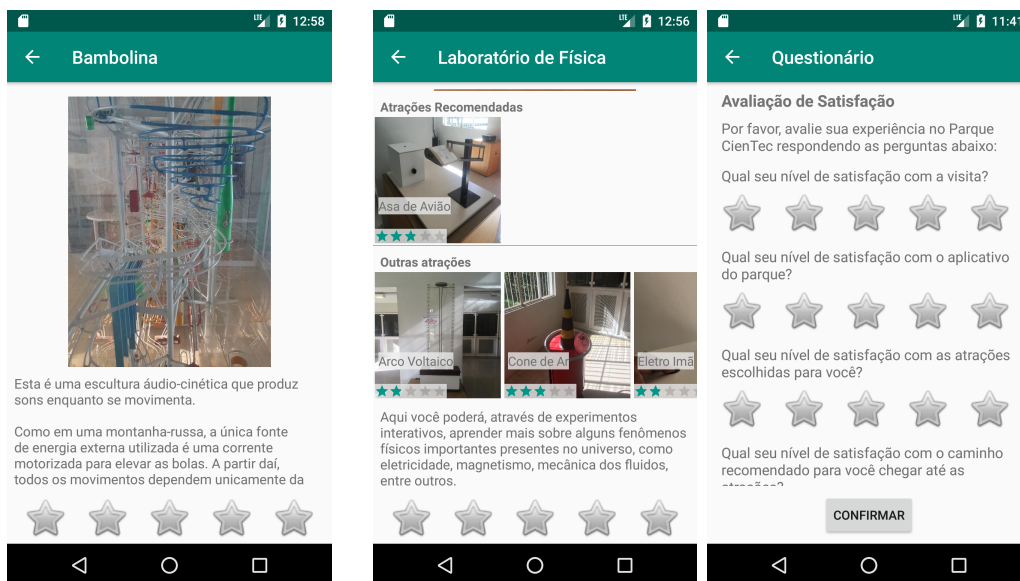
Google Maps e foi utilizado dessa forma na condução da maior parte da primeira avaliação *on-line* desta pesquisa (5.4). A partir da versão 1.1 do sistema, um novo mapa (figura 4.5) desenvolvido para o aplicativo foi implantado e utilizado no parque a partir de 19 de outubro de 2019.

Quando o usuário chegar a menos de 15 metros do destino em vermelho, uma janela com o aviso de chegada é apresentada. Ao clicar nessa janela, o aplicativo é direcionado para a tela de detalhes do item (figura 4.6a). Nela o usuário pode consultar a foto e descrição do item, além de avaliá-lo, de 1 a 5 estrelas. Se o usuário entrou nessa tela após ter se aproximado do item (15 metros), um botão de continuar rota será mostrado nessa tela. Ele só ficará habilitado após o usuário avaliar o item atual. Caso ele queira voltar para a tela do mapa, ele pode apertar no botão de voltar. Nesse caso uma mensagem aparecerá ao usuário perguntando se ele já visitou o item. Se selecionar que sim, um aviso para avaliar o item aparecerá, caso contrário ele voltará para a tela do mapa.

No caso do item ser do tipo *item agrupador*, a tela de detalhes mostrará também os subitens que ele contém separados em duas listas: os recomendados para o usuário em sua rota e tempo disponível, e os demais itens disponíveis no local, como mostra a figura 4.6b.

Após avaliar um item na tela de detalhes, ao voltar para a tela principal o sistema de recomendação irá realizar um novo treinamento, já utilizando a nova transação. Nesse caso é possível que a rota recomendada seja modificada, considerando o tempo já gasto pelo usuário até aquele momento e sua atual localização geográfica.

Ao terminar de visitar o último item recomendado, ou ao acabar o tempo de visita informado no início, o aplicativo irá avisar o usuário que o tempo pretendido já passou e o conduzirá para a tela de avaliação de satisfação com a visita (4.6c). Após responder o questionário, o usuário será questionado se quer continuar a usar o aplicativo. Se ele clicar



(a) Detalhes do item/subitem

(b) Detalhes do item agrupador

(c) Avaliação de satisfação

Figura 4.6: Telas de detalhes dos elementos de conteúdo e avaliação de satisfação com a visita

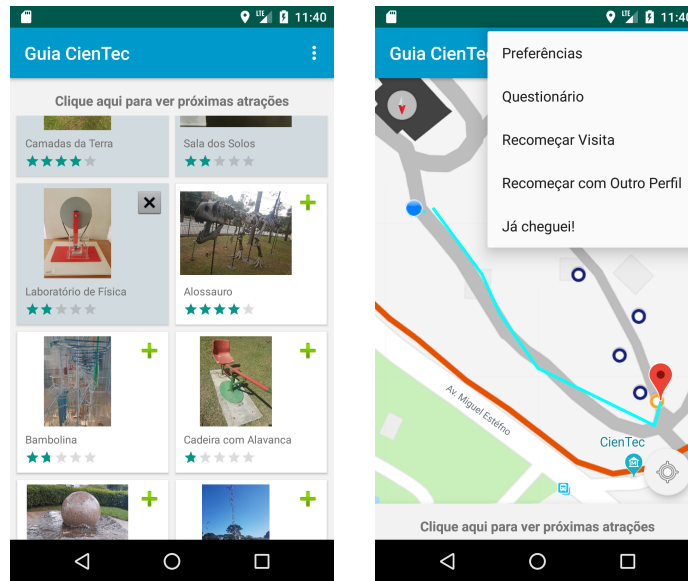
em não, suas informações de uso no aplicativo são removidas e o aplicativo retorna para a tela de informações de contexto para um novo usuário/visita.

4.5.2 Funções Adicionais

A qualquer momento durante a visita e utilização do aplicativo o usuário pode consultar sua rota recomendada e todos os itens disponíveis no local pelo menu (figura 4.7a) que aparece ao clicar, ou puxar, a barra inferior na tela do mapa. No menu, os primeiros itens estão em azul. Eles são os que constam na rota de visitação formada na ordem lá apresentada. Por essa tela o usuário pode também escolher adicionar um novo item (roteável), que não havia sido recomendado para ele, à rota. Ele faz isso clicando no símbolo de + no canto dos itens em branco (fora da rota). Ao fazer isso, o sistema de recomendação entrará em ação para definir a melhor posição na rota recomendada para adicionar esse novo item. A partir disso, ao retreinar o SR, esse item adicionado será adicionado à rota independentemente da sua nota de recomendação (linhas 6 e 7 do algoritmo 3).

O usuário também pode remover um item clicando no X no canto dos itens em azul na lista. Ao fazer isso, o SR irá verificar se, com o tempo disponível pela exclusão do item, é possível adicionar um novo item aos recomendados.

Por fim, há um menu superior, conforme figura 4.7b, que permite ao usuário voltar para a tela de informações iniciais, ir para o questionário de satisfação, recomeçar visita, recomeçar com outro perfil e dizer que já chegou no próximo item da rota. Ao clicar em recomeçar visita, o sistema irá zerar o tempo gasto até então e irá considerar novamente o tempo disponível informado na tela de informações de contexto (4.4a) para remontar a rota recomendada. No caso dele clicar em começar com um novo perfil, todas as informações do usuário no

(a) *Lista de itens*(b) *Menu superior***Figura 4.7:** *Telas com lista de itens disponíveis e menu de opções*

aplicativo serão removidas e o aplicativo voltará à primeira tela de informações. Se ele clicar em "Já cheguei!", resultará no mesmo efeito que acontece quando ele fica a menos de 15 metros do item, de aparecer a janela com os detalhes do item.

Capítulo 5

Coleta e Análise dos Resultados

Neste capítulo será apresentado como foi feita a coleta de dados da aplicação do sistema no Parque CienTec e uma análise desses dados para entender o desempenho do sistema em termos de acurácia da recomendação e satisfação do usuário. O capítulo é composto por sete seções: apresentação de dados coletados com estatísticas gerais na primeira seção, a metodologia utilizada para a coleta de dados e avaliação dos sistemas de recomendação, na Seção 5.2, as quatro fases que compuseram a avaliação nas seções seguintes, terminando com uma discussão sobre os riscos à validade das avaliações na Seção 5.7. As quatro fases da avaliação são:

1. Formação da base inicial de recomendações;
2. Avaliação *on-line* com sistema de recomendação inicial;
3. Avaliação *off-line* comparativa de acurácia entre doze implementações de SR com os dados obtidos na fase anterior;
4. Avaliação *on-line* com o SR com melhor desempenho na avaliação *off-line* e posterior comparação com desempenho obtido pelo SR na fase 1.

5.1 Informações Gerais da Coleta dos Dados

Os dados foram coletados entre 10 de agosto e 10 de novembro de 2019 com visitantes e monitores do Parque CienTec. Ao todo 76 usuários participaram da pesquisa, sendo 14 **monitores** e 62 **visitantes**. Somando as avaliações feitas por todos os participantes, chegou-se a um total de 1472 avaliações, sendo 607 referentes aos atributos da modelagem contextual (preferências iniciais e idade) e 865 dos elementos de conteúdo disponíveis no local. A Tabela 5.1 mostra a proporção de participantes por faixa etária e se haviam visitado o parque anteriormente.

Como o aplicativo com o SR estava publicamente disponível na plataforma do *Google Play*, houve alguns casos de *downloads* de pessoas que não efetivamente fizeram a visita ao

Faixa Etária	Já conhecem o Parque	
	Sim	Não
Até 18 anos	5	3
19 a 25 anos	10	8
26 a 40 anos	10	23
40 anos ou mais	5	11
Não informado	0	1
Total	30	46

Tabela 5.1: Distribuição dos usuários considerados na pesquisa por faixa etária e se já haviam visitado o parque.

Parque CienTec. Esses casos não estão considerados entre os 76 participantes. Para filtrá-los, utilizou-se a localização de GPS salva para cada avaliação. Foram considerados os usuários que fizeram avaliações dentro do perímetro do parque.

Os monitores que participaram da pesquisa não efetivamente fizeram a visita ao parque. Eles apenas contribuíram com suas avaliações em relação à maioria dos elementos do parque para formar uma base de dados inicial de modo que as fases 2 e 4 de coleta de dados, com visitantes reais, se tornasse viável. Por essa razão os monitores não serão considerados nos resultados das análises feitas nas fases 2 a 4.

Considerando somente os usuários do sistema que são visitantes, o tempo médio de visita pretendido inicialmente (informado pelo usuário no aplicativo ao iniciar a visita) foi de 123 minutos. Para avaliar se o tempo de visita pretendido é próximo ao tempo efetivamente utilizado, calculou-se a diferença entre a primeira e a última avaliação feita pelo usuário, considerando as preferências iniciais, avaliações dos elementos e questionário de satisfação. Foram desconsiderados os casos de usuários que adicionaram uma nova transação ao sistema mais de quatro horas após a última avaliação, pois caracterizou-se que ele já concluiu a visita, mas acessou o sistema posteriormente. Também foram desconsiderados os usuários que levaram menos de 15 minutos entre a primeira e a última avaliação no aplicativo. Com esse critério, os dados de tempo de 32 usuários foram considerados e a média e mediana de tempo pretendidos de visita encontrados foram de 98 e 90 minutos, respectivamente. A média e mediana reais encontradas foram de 83 e 76 minutos, respectivamente, valores 15,9% e 15,5% inferiores ao pretendido.

Em relação às avaliações de elementos realizadas pelos visitantes, a média geral foi de 4,27, em uma escala de 1 a 5 estrelas. Em média 6,5 avaliações foram feitas pelos usuários durante suas visitas. O gráfico 5.1 mostra a distribuição de usuários pela quantidade de avaliações feitas, sem considerar as preferências do questionário inicial e avaliação de satisfação.

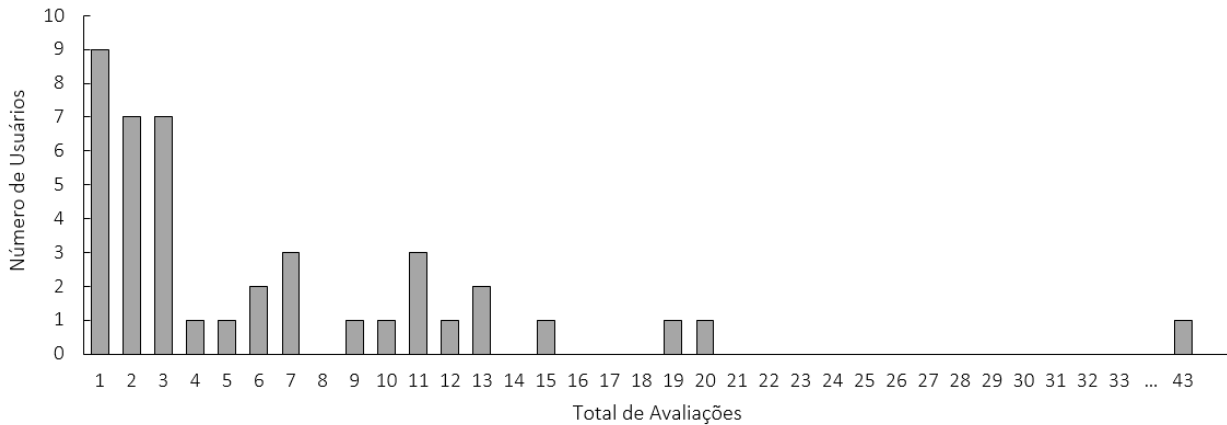


Figura 5.1: Gráfico com a distribuição de usuários pelo total de avaliações de elementos de conteúdo feitas.

5.2 Metodologia Utilizada

5.2.1 Metodologia da Coleta de Dados

A coleta foi feita pelo aplicativo Guia CienTec instalado nos aparelhos pessoais dos visitantes. Eles eram convidados a experimentar o aplicativo logo ao entrar no parque. Porém nem todos acabaram o utilizando, por razões como incompatibilidade do sistema operacional (ou de sua versão) com o aplicativo, ausência de aparelho móvel, tempo disponível de visita muito curto ou mesmo porque simplesmente não tinham planos de visitar as atrações do parque, já que haviam grupos que iam até lá por outros motivos. Havia também os casos que o visitante já sabia de antemão a atração que queria visitar e negou o uso do aplicativo.

Ao convidar os visitantes a testar o aplicativo era-lhes informado que ele serviria para ajudá-los a conhecer o parque, contendo mais informações de suas atrações, além de recomendar itens de forma personalizada e mostrar o caminho até eles. Dependendo da dificuldade que o visitante demonstrava em usar o aplicativo, era também dada uma breve explicação de como utilizá-lo.

Durante a visita, os monitores se mantinham à vista dos visitantes, apenas os observando. Caso o usuário apresentasse grandes dificuldades em seguir o caminho orientado pelo aplicativo, os monitores intervinham pontualmente ou os acompanhavam até o próximo destino recomendado.

5.2.2 Metodologia da Avaliação do SR

Para avaliar a acurácia do sistema de recomendação utilizamos as métricas RMSE e MAE mensuradas do SR e comparamos às de um SR hipotético que fornece valores inteiros aleatórios de recomendação entre 1 e 5, chamado de sistema aleatório. Como explicado na Seção 2.7.2, tanto MAE como RMSE são amplamente utilizadas para verificar a acurácia de um SR e também podem ser aplicadas neste trabalho, pois a nota de recomendação produzida pelo sistema é um valor numérico contínuo e há disponibilidade do *feedback* do usuário com

sua própria avaliação. As análises foram feitas comparando com um sistema aleatório em vez de resultados de outras pesquisas, pois resultados em outros contextos podem não ser comparáveis com os desta pesquisa, dadas as diferenças no contexto de aplicação, no objetivo do SR ou nas métricas obtidas.

Como 5 é a maior nota possível para ser fornecida pelo usuário, nos casos que o sistema calculou uma nota superior a 5, a nota era convertida para 5. O mesmo se aplicou ao extremo inferior, onde notas menores que 1 eram convertidas para 1 para fins de cálculo do MAE e RMSE.

Além da nota de recomendação, uma rota recomendada é produzida, pelo SR do aplicativo, com os itens roteáveis mais recomendados, considerando o tempo disponível de visita. Isso gera uma classificação binária do recomendador: Está ou não está na rota. Entretanto, mesmo com essa classificação, não há como validar se um item que foi considerado dentro da rota teve uma classificação correta ou não. Isto porque não há uma resposta clara de qual a classificação correta, visto que a adição ou não de um item a rota está condicionado a outros fatores além da recomendação, como o tempo que o usuário tem disponível e a disposição dos itens no local. Portanto o cálculo de falso/verdadeiro positivo é impossibilitada, assim como a geração de uma curva ROC do SR.

5.3 Fase 1: Formação da Base Inicial de Recomendações

Esta etapa foi planejada para superar o *cold start* e prover recomendações relevantes aos primeiros usuários do aplicativo. Ao todo 14 monitores e funcionários do Parque CienTec forneceram suas avaliações e preferências, totalizando 704 entradas. Dessas, 592 são avaliações de elementos de conteúdo e 112 provêm das sete avaliações de preferências mais a idade convertida em nota de 1 a 5. A média de idade desses participantes foi de 28 anos.

Para realizar a coleta das informações dos monitores foi utilizado o próprio aplicativo Guia CienTec. Aqueles que não possuíam aparelhos compatíveis com o aplicativo utilizaram um celular à parte. A Tabela 5.2 mostra algumas informações consolidadas das avaliações fornecidas pelos monitores, sem considerar as 14 transações relacionadas a idade do usuário.

Métrica	Itens	Preferências	Total
Número de avaliações	592	98	690
Nota média das avaliações	3,93	4,14	3,96
Quantidade média de avaliações	42,2	7	49,2

Tabela 5.2: Estatísticas de avaliações dos monitores e funcionários participantes.

Desde o começo foi deixado claro para os monitores que as notas de avaliação e preferências iniciais deveriam ser respondidas, respectivamente, baseadas no nível de satisfação e interesse deles com os itens. Os monitores inicialmente preencheram sua idade e preferências na tela inicial, deixando a pergunta relacionada ao tempo de visita com um valor aleatório, dado que eles não fariam a visita proposta, mas apenas avaliariam os itens. Em seguida

Métrica	Itens	Preferências	Total
Número de avaliações	113	217	330
Nota média das avaliações (1 a 5)	4,3	4,2	4,2
Quantidade média de avaliações	3,6	7	10,6

Tabela 5.3: *Estatísticas de avaliação dos visitantes da primeira fase.*

foram requisitados a entrar na página de detalhes de cada um dos elementos de conteúdo que já conheciam para fornecer suas avaliações. Alguns monitores avaliaram também itens que não conheciam, após terem a experiência de visitaç o com o uso do aplicativo.

Durante esta etapa, *feedbacks* relacionados ao mapa do parque, descriç o, localizaç o e fotos dos itens, experi ncia de uso e forma de recomendar rota ao visitante foram coletados e utilizados na evoluç o do aplicativo.

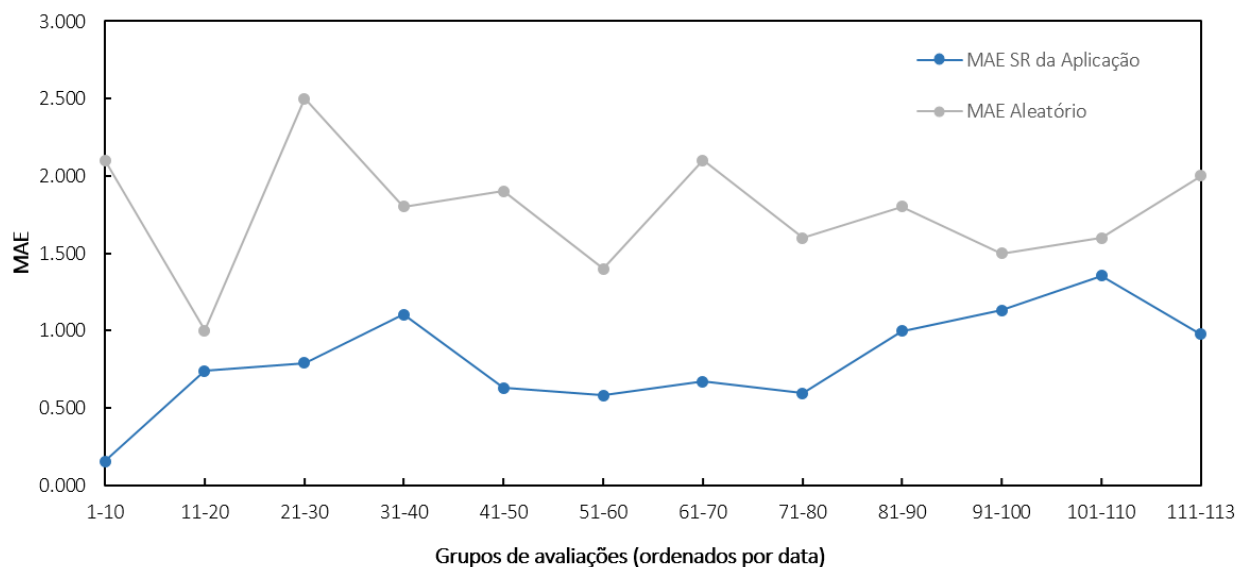
5.4 Fase 2: Primeira Avaliaç o *On-line*

Esta fase teve o objetivo de aplicar o sistema de recomendaç o em um ambiente real para coleta e an lise de desempenho do SR e comparaç o com outras implementaç es do sistema, feitas na terceira fase. Al m disso, esta fase tamb m foi importante para obter *feedbacks* dos visitantes reais, colaborando para que as funcionalidades essenciais para um bom aproveitamento do uso do aplicativo fossem desenvolvidas.

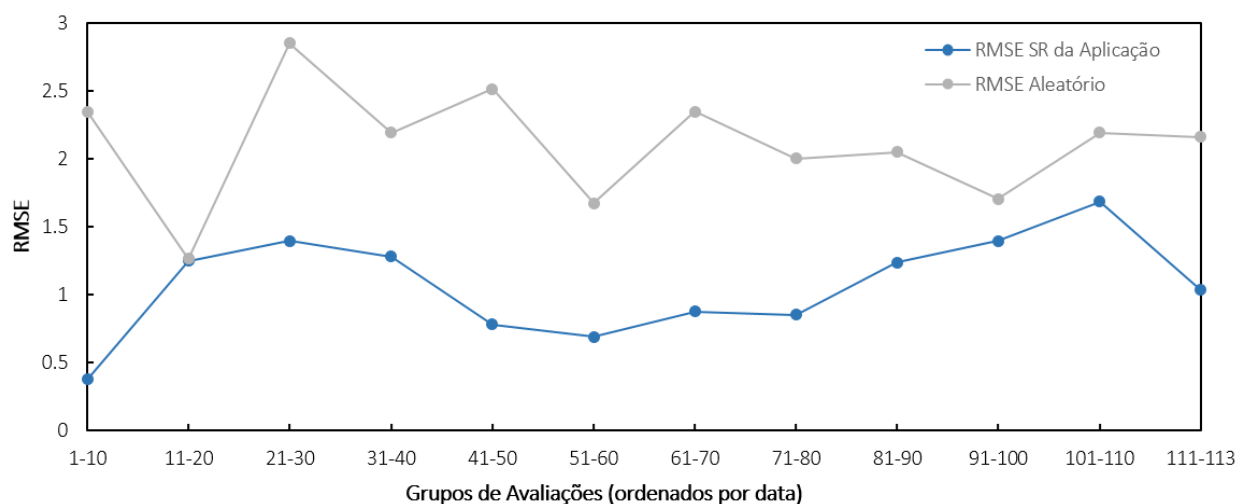
Seguindo a metodologia descrita na Seç o 5.2.1, nesta etapa todos os 31 usu rios que utilizaram o aplicativo, a partir do momento de conclus o da etapa anterior at  o dia 19 de outubro de 2019, foram considerados. O aplicativo ficou dispon vel para uso no Parque CienTec somente aos s bados desse per odo. A Tabela 5.3 resume as informaç es gerais de avaliaç es feitas pelos usu rios, sem considerar as 31 transaç es relacionadas a idade do usu rio.

5.4.1 Acur cia

Em relaç o a acur cia dos resultados, o sistema de recomendaç o implementado no aplicativo, utilizando o m todo h brido apresentado no Cap tulo 4 baseado no CF baseado em usu rio com m todo de comparaç o PCC e kNN com $k = 4$, demonstrou claramente possuir um melhor desempenho que o sistema aleat rio em ambas as m tricas MAE e RMSE. O MAE obtido para o SR do aplicativo foi de 0,799 contra um MAE de 1,761 do sistema aleat rio. Em relaç o ao RMSE, o valor obtido pelo SR proposto foi de 1,132 e 2,143 para o sistema aleat rio. As figuras 5.2a e 5.2b comparam, respectivamente, o MAE e RMSE, dos dois sistemas, por grupo de 10 avaliaç es, seguindo a ordem cronol gica de obtenç o dessas avaliaç es. O melhor desempenho do SR implementado no aplicativo se mant m na maioria dos casos. Era tamb m esperada uma melhora na acur cia   medida que a base hist rica de avaliaç es aumentasse, mas isso n o foi percebido pelas m tricas considerando essa vis o.



(a) MAE

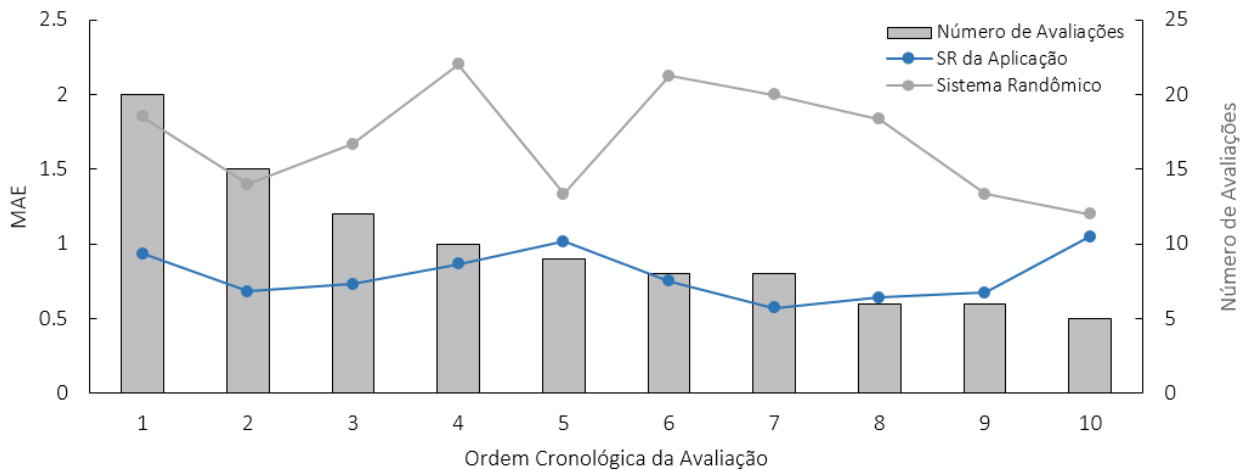


(b) RMSE

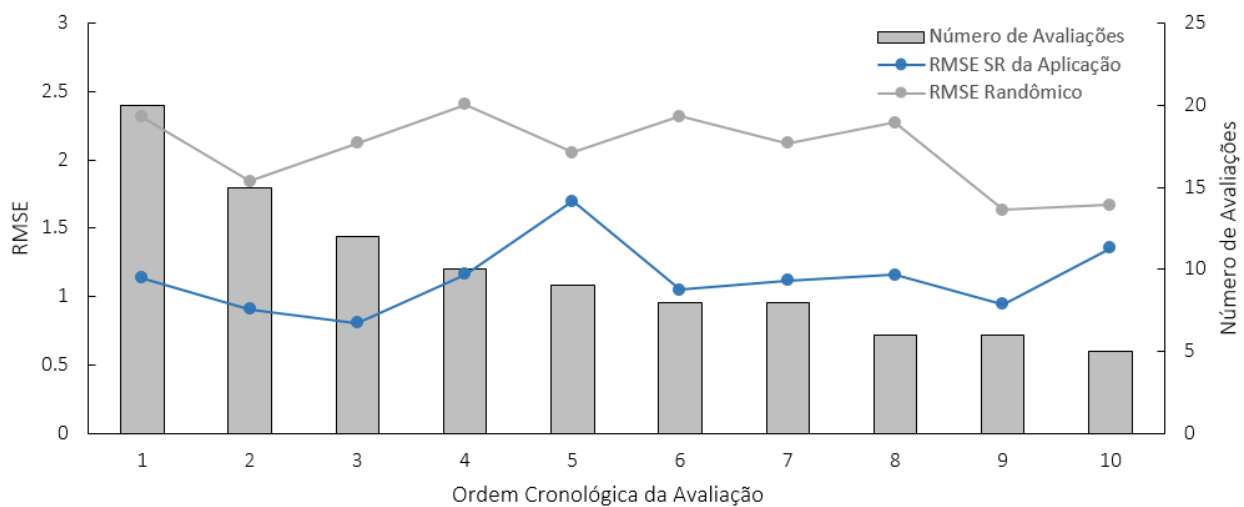
Figura 5.2: Valores de MAE e RMSE do SR do aplicativo e aleatório para grupos de 10 avaliações, ordenados por data

Como a visão das figuras 5.2 mostra as avaliações ordenadas unicamente por data, avaliações mais novas, mesmo tendo uma base histórica maior, poderiam não obter um desempenho melhor que as anteriores. Isso é plausível nos casos de avaliações feitas por novos usuários e/ou para itens novos ou que tenham poucas avaliações na base. Por isso também foi analisado o desempenho do SR em relação a quantidade de avaliações já feitas por cada usuário.

Os gráficos das figuras 5.3 buscam validar isso: Se para os usuários que o sistema possui mais informações, ou seja, detém um maior número de avaliações em sua base, ele produz resultados com maior acurácia. Para isso, agrupou-se as avaliações pela ordem que elas foram feitas pelos usuários para comparar se o MAE e RMSE (linha azul do gráfico) das primeiras avaliações foi menor que o das avaliações posteriores. Os gráficos também apresentam a



(a) MAE



(b) RMSE

Figura 5.3: Gráfico do MAE e RMSE por grupo de avaliações feitas na mesma posição cronológica pelos usuários. As colunas representam o número de avaliações consideradas naquela posição.

quantidade de avaliações consideradas em cada posição através das barras em cinza. Valores de MAE e RMSE para posições com menos de 5 avaliações foram omitidos, por terem, estatisticamente, baixa representatividade.

Na figura 5.3a o MAE do SR da aplicação se situa entre 0,5 e 1 na maior parcela das avaliações enquanto que o MAE do sistema aleatório, como era de se esperar, oscila ao redor de sua média de 1,76. Percebe-se também uma leve tendência de melhora na métrica do SR da aplicação até a nona ou mesmo até a décima terceira avaliação. Contudo, essa tendência pode ser considerada muito pequena para caracterizar uma real evolução na recomendação. Assim, para confirmar essa tendência, a análise foi revista após a coleta de dados da segunda avaliação *on-line*, conforme mostram os resultados da quarta fase, na Seção 5.6. O gráfico do RMSE (5.3b) se comporta de forma semelhante, embora a tendência de queda não seja percebida na evolução das primeiras posições.

Em relação a quantidade de avaliações, da décima posição em diante, menos de cinco

avaliações foram consideradas, o que diminui a precisão da medida. Por esse motivo também que o MAE e RMSE da aplicação se tornam altamente instáveis da décima quarta posição em diante, já que apenas um visitante considerado forneceu mais de 13 avaliações. Já a média de tempo disponível para visita, informada pelos visitantes, foi de 1 hora e 37 minutos.

5.4.2 Satisfação do Usuário

Ao passar o tempo definido pelo usuário no início da visita, era requisitada sua avaliação de satisfação com relação a quatro características: a visita como um todo, as recomendações feitas pelo aplicativo, a rota estabelecida e o aplicativo em si. O gráfico da figura 5.4 apresenta a média de satisfação com cada um desses itens.

No geral as avaliações tiveram notas altas, inclusive quando comparadas a outros trabalhos relacionados. O item de menor nota entre eles foi sobre a experiência geral com o aplicativo, com média igual a 4,0. De qualquer forma, são citadas abaixo algumas dificuldades que foram identificadas no transcorrer da pesquisa e que trazem oportunidades para uma avaliação ainda mais alta pelos usuários:

- **Fluxo de uso do aplicativo:** Foram identificadas dificuldades dos usuário em dar continuidade à rota recomendada após ter já visitado algum item. Uma situação típica é quando o usuário desvia da rota recomendada para conhecer alguma atração que lhe chamou a atenção e quer ver mais informações dela. Outro caso é quando o visitante conhece algumas atrações pelo monitor que estava guiando outro grupo e depois quer retomar a visitação pelo aplicativo.
- **Imprecisão na localização do usuário:** O posicionamento não possui uma precisão muito boa pois o local tem um sinal baixo de redes móveis e a *Wi-Fi* do parque cobre apenas uma pequena área do todo. Como o ideal para navegar no parque é manter o *zoom* do mapa no nível máximo, pequenas imprecisões impactam a experiência de uso. Essa dificuldade era maior antes da implantação do novo mapa. Além disso, a métrica da posição do usuário não apresenta o sentido para o qual o usuário está apontando, dificultando a navegação.
- **Dificuldade em entender a orientação do mapa do parque:** Por exemplo, para identificar para que lado o usuário estava voltado, alguns usuários andavam alguns passos para algum lado para ver como o ícone da posição do usuário se movia no mapa.
- **Locais recomendados para o usuário:** Em alguns momentos estavam equivocadamente fechados, isto é, um local poderia estar fechado, embora estivesse em um momento do dia que deveria estar disponível.
- **Personalização da ordem dos itens na rota:** Embora o aplicativo possibilite adicionar novos itens a rota, ele não permite reorganizá-los na ordem que o usuário preferir.

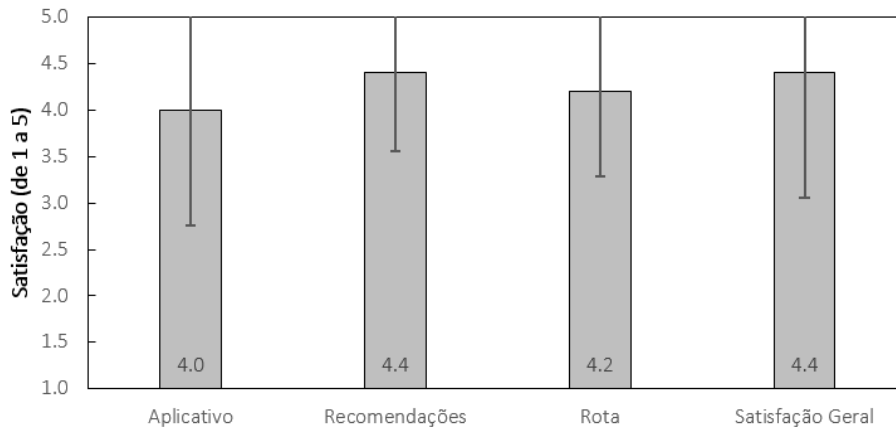


Figura 5.4: Satisfação do usuário com as diferentes características do sistema.

- **Tempo necessário para iniciar:** Como o aplicativo pede um conjunto de informações iniciais, ele pode desagradar alguns usuários, como aqueles que não tem muita familiaridade com o manuseio de aplicativos de celular ou estão com pressa para iniciar a visita.
- **Obrigatoriedade da avaliação das atrações:** Para dar continuidade a rota recomendada o aplicativo pede a avaliação do último item visitado, o que pode desagradar alguns usuários. Houve também situações de usuários que confundiram o botão de continuar rota com o de voltar para o mapa. Nesse caso é pedido sua avaliação, embora ele não tenha visitado o item ainda, gerando dúvida e insatisfação.

Embora esses itens possam explicar o impacto na satisfação com o aplicativo, eles podem também ser facilmente considerados pelos usuários nas demais características avaliadas. A Seção 5.6 mostra como essa nota aparentemente inferior do aplicativo não se concretiza na segunda avaliação *on-line* do sistema. Na realidade tanto a satisfação com o aplicativo quanto a no geral apresentam uma imprecisão alta nos resultados, com desvios padrão de 1,25 e 1,35, respectivamente.

5.4.3 Análise de Desempenho e Escalabilidade

Como abordado na Seção 2.8, o desempenho de um SR, à medida que aumentam o número de usuários, itens ou transações, pode ser um fator determinante de sua viabilidade. Por isso, o sistema utilizado nessa fase de análise foi testado e teve seu tempo de processamento verificado para três situações, como será mostrado nesta seção.

Todos as medições aqui apresentadas foram feitas no mesmo aplicativo utilizado no parque, adaptado para guardar os tempos de processamento requeridos para as análises já mencionadas. Os tempos de processamento foram obtidos com testes feitos com um aparelho *Samsung S7 SM-G930F*.

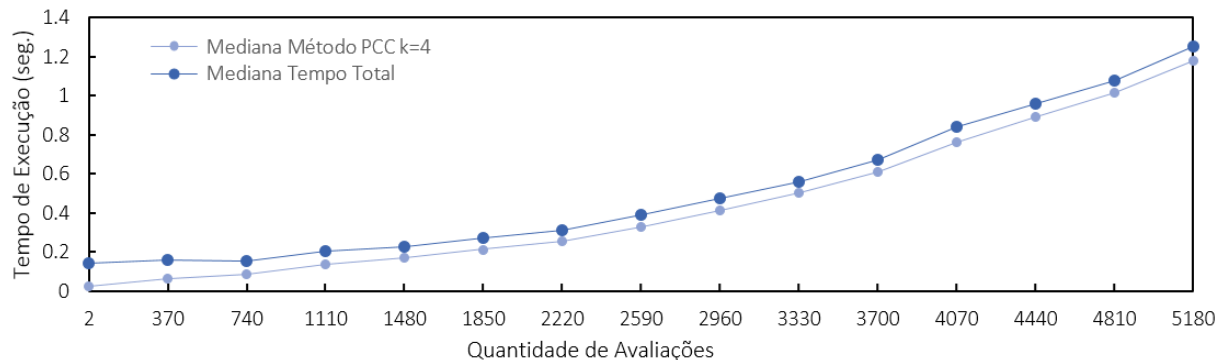


Figura 5.5: *Evolução no tempo de processamento com o aumento das avaliações*

Aumento no número de avaliações

Quanto mais usuários utilizarem o aplicativo, mais avaliações serão mantidas no banco de dados. Conforme solução do SR apresentado no Capítulo 4, o número de avaliações deve afetar o tempo de treino do método CF implementado e isso é confirmado nos testes de desempenho. O gráfico da figura 5.5 mostra como o tempo de execução aumenta à medida em que novos usuários e avaliações são adicionados à base de treino. Cada ponto do gráfico representa a mediana do tempo de processamento de cinco treinamentos realizados com a quantidade de avaliações informadas no gráfico e número fixo de itens (92) igual à quantidade disponível na aplicação do Parque CienTec. O algoritmo para o teste primeiro criou a base com as novas avaliações. Em seguida, para cada conjunto de cinco testes, o algoritmo utilizou uma subdivisão da base completa contendo o número de avaliações apresentadas no eixo x do gráfico.

A linha em azul claro representa o tempo da primeira etapa da geração da rota recomendada (como apresentado em 4.4.2, o treino do método CF baseado em usuário com $k = 4$ e uso do PCC, e a linha escura representa o tempo total da geração da rota, com as duas etapas do SR, que inclui o cálculo do máximo de itens a visitar dado o tempo disponível e menor caminho entre os itens. Como o gráfico mostra, com o aumento da quantidade de usuários/avaliações, apenas a primeira etapa do SR é afetada, pois a diferença entre as duas linhas se mantém constante, indicando estabilidade no tempo da segunda etapa.

Se considerar que o tempo máximo aceitável é de 1,5 segundos, mesmo com mais de 5000 avaliações sendo consideradas o tempo de treino se mantém abaixo desse limite. Como o SR do aplicativo limita a utilizar as avaliações de no máximo 250 usuários e considerando 20 avaliações em média por usuário (a média encontrada no cenário real foi de $6,5 + 8 = 14,5$ transações), o tempo de processamento tende a sempre ficar dentro do aceitável para esta aplicação.

Aumento no número de itens roteáveis

O gráfico da figura 5.6a mostra como, nesse caso, o efeito no tempo de processamento é o oposto ao visto no caso anterior. Isto é, o tempo de treino do método CF se mantém

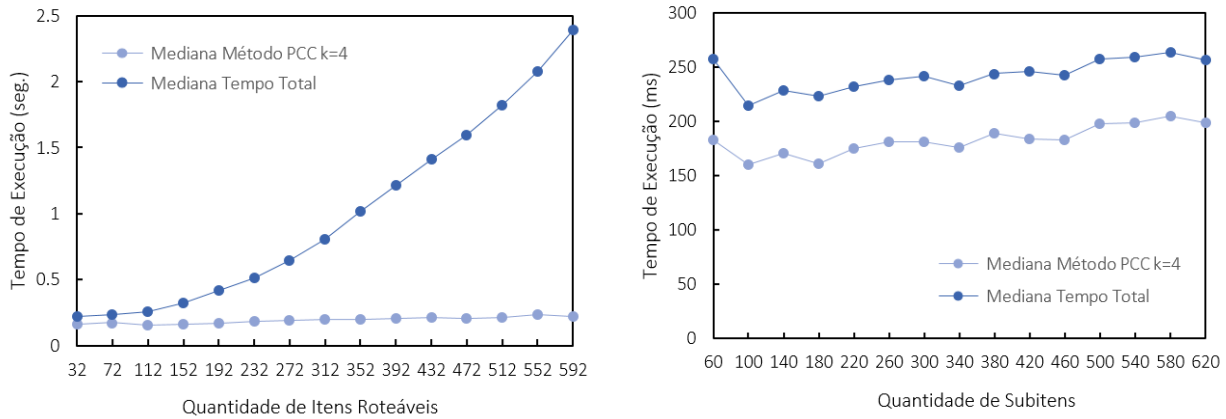
(a) *Variação na quantidade de itens roteáveis*(b) *Variação na quantidade de subitens*

Figura 5.6: *Evolução no tempo de processamento para gerar a rota recomendada com o aumento da quantidade de itens e subitens*

constante ao passo que o tempo total de geração das recomendações cresce rapidamente.

Cada ponto do gráfico foi obtido também com o uso da mediana de cinco treinamentos de uma simulação de visita de duas horas. O primeiro ponto do gráfico utilizou a base original, com 92 itens ao todo, dos quais 32 são roteáveis e 60 são subitens, além dos 37 pontos, e as 1065 transações provenientes da modelagem contextual e avaliações de itens pelos usuários. A cada conjunto de cinco treinamentos, 40 novos itens eram adicionados, todos com uma avaliação de cada.

A primeira etapa do SR, que faz uma matriz de semelhança entre usuários e calcula notas de recomendação baseando-se em avaliações feitas, demonstra não ter sido afetada pela inclusão dos novos itens, apenas com um leve aumento provavelmente devido as 560 novas avaliações que também foram adicionadas à base, seguindo o comportamento identificado no teste de aumento de avaliações.

A segunda etapa da recomendação tem forte impacto no custo de processamento com a inclusão de novos itens roteáveis porque eles influenciam, tanto no cálculo do menor caminho do algoritmo *greedy nearest neighbors search*, quanto na otimização da relação quantidade de itens pelo tempo de visita. Essa otimização é feita de forma iterativa sobre a lista de itens recomendados como explicado no algoritmo 2.

Considerando novamente 1,5 segundos como o limite de tempo aceitável, a aplicação não deveria conter um número de itens roteáveis maior que 400. Esse número é bastante adequado para as aplicações em geral, lembrando que os subitens dentro de um item agrupador não são considerados itens roteáveis. No caso do Parque CienTec, são 32 itens roteáveis ao todo.

Aumento no número de subitens

Também foi verificado o impacto no tempo ao adicionar itens não roteáveis, que são os subitens de um agrupador, ou seja, cada objeto pertencente a uma determinada exposição. Os resultados desse teste estão representado no gráfico da figura 5.6b.

Método CF	k	Fun. Semelhança	MAE	RMSE
Baseado em usuário	5	PCC	0.753	0.989
Baseado em usuário	4	Cosseno	0.763	0.991
Baseado em usuário	5	Cosseno	0.77	1.001
Baseado em usuário	4	PCC	0.773	1.003
Baseado em usuário	3	PCC	0.785	1.027
Baseado em item	5	Cosseno	0.789	1.046
Baseado em item	4	Cosseno	0.794	1.051
Baseado em usuário	3	Cosseno	0.796	1.038
Baseado em item	4	PCC	0.806	1.061
Baseado em item	3	PCC	0.806	1.07
Baseado em item	3	Cosseno	0.806	1.078
Baseado em item	5	PCC	0.815	1.073

Tabela 5.4: Acurácia das doze variações do SR no teste 10-fold cross validation.

Os testes foram feitos considerando inicialmente a mesma base inicial de elementos e avaliações do teste de aumento itens roteáveis. Foram feitos quinze grupos de cinco treinamentos, cada um adicionando 40 novos subitens à base original, e a mediana de cada grupo foi selecionada para compor o gráfico.

Verifica-se uma leve tendência de aumento no tempo de processamento da primeira fase (em azul escuro) que é acompanhada pelo tempo total. Ou seja, a primeira etapa do SR tem um pequeno impacto no processamento devido ao aumento no número de itens e avaliações, que é compatível com o aumento visto nos outros dois testes à medida que aumentam o número de avaliações, neste caso, até 560 novas avaliações, no último ponto. O impacto na segunda etapa apresenta-se como nulo visto que a diferença entre o tempo total e o da primeira etapa segue constante.

5.5 Fase 3: Avaliação *Off-line* com Múltiplas Variações do SR

Esta fase teve como objetivo utilizar a base de dados obtida nas fases anteriores para comparar o desempenho de 12 variações de implementação do sistema de recomendação para definir a implementação a ser empregada na última fase da pesquisa. As variações testadas se diferem entre si pelos seguintes atributos:

- **Abordagem do método CF *memory-based*:** baseado em item ou baseado em usuário
- **Parâmetro k :** 3, 4 ou 5 "vizinhos"
- **Função de semelhança:** coeficiente de Pearson ou semelhança do cosseno

Cada uma das doze variações, que é uma combinação diferente desses três atributos, foram testadas e comparadas em relação à acurácia, com as métricas MAE e RMSE, e tempo de treinamento.

5.5.1 Comparação de Acurácia

Foram feitos dois testes comparativos desses sistemas. O primeiro consistiu em obter a média de MAE e RMSE de cada um dos doze sistemas através da técnica *10-fold cross validation*. Foram realizados os seguintes passos:

1. Selecionar uma variação do sistema de recomendação.
2. Seguir a técnica de validação cruzada, dividindo toda a base de transações em dez partes iguais.
3. Treinar o sistema com 9 das partes e obter as métricas pela validação do sistema com a décima parte.
4. Realizar os passos 2 e 3 dez vezes, até que todas as partes tenham sido usadas como a base de teste, e obter as médias das métricas pelos dez conjuntos encontrados.
5. Repetir o processo com outra variação do sistema

Os valores encontrados de MAE e RMSE com esse procedimento, apresentados na Tabela 5.4, apontam que a variação de melhor desempenho é a baseado em usuário com $k = 5$ que utiliza, como função de semelhança, o coeficiente da correlação de Pearson. Os resultados também demonstram que as variações que aplicam CF baseado em usuário conseguem, em geral, um melhor desempenho nessa aplicação. Contudo, não está conclusivo qual o melhor tipo em relação a k e a função de semelhança.

O segundo teste de acurácia foi feito simulando a evolução do sistema seguindo a dinâmica de funcionamento no ambiente real. Os seguintes passos foram realizados nesse teste:

1. Para cada variação do SR, executar as etapas seguintes.
2. Considerar a base de dados de treino, T, apenas com a transação mais antiga da base original e ordena-se a base original da mais antiga à mais nova, pela data/hora de avaliação.
3. Selecionar a avaliação A mais antiga da base original que ainda não consta em T.
4. Se A é um atributo da modelagem contextual, repetir a etapa 3. Caso contrário, treinar o SR com a base T.
5. Calcular a diferença entre a nota de recomendação que o sistema calculou para A e a avaliação real do usuário. Guardar o resultado.

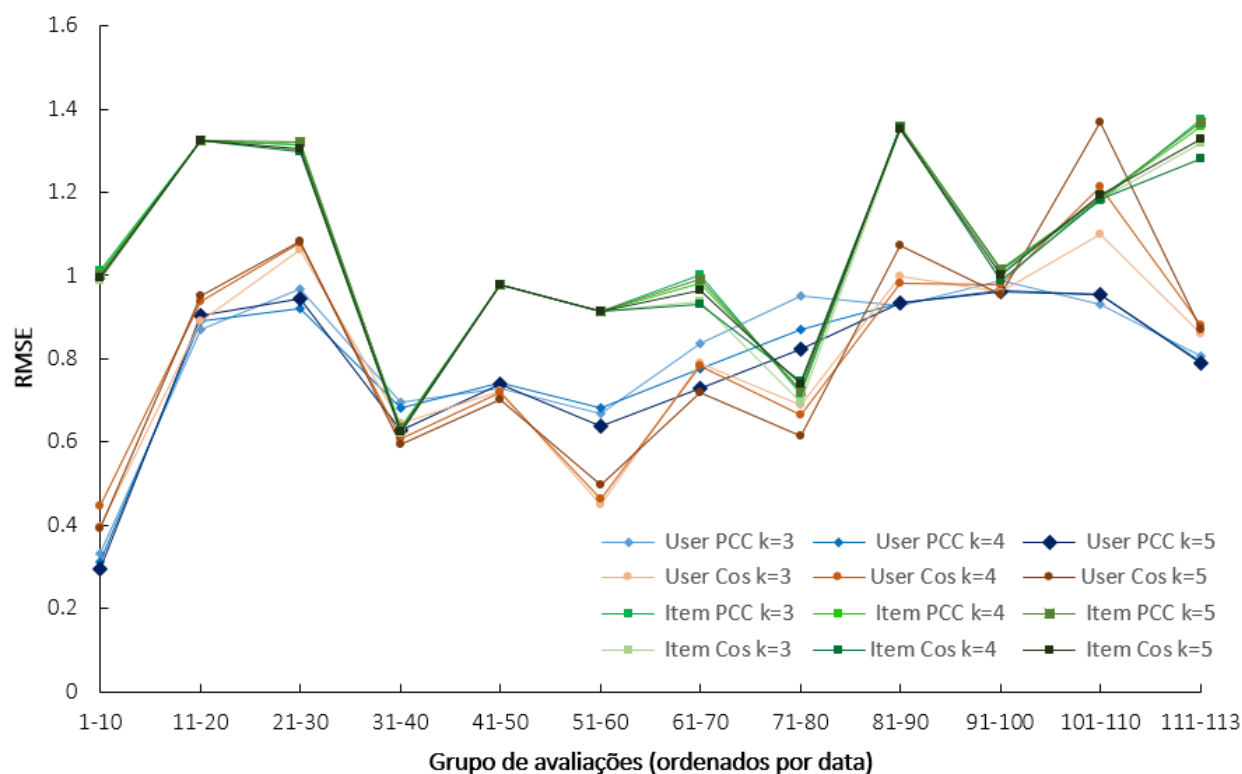


Figura 5.7: RMSE das 12 variações do SR com a base de 113 avaliações da fase 2.

6. Repetir as etapas 3 a 5 até que todas as avaliações da base original constem em T.
7. Calcular o MAE e RMSE com base nos valores encontrados.

O gráfico da figura 5.7 mostra como cada implementação do SR se comporta com a base de avaliações da fase 2. Percebe-se o comportamento similar e com menor acurácia de todas as variações que utilizam o método baseado em item (quadrados, em tons de verde, no gráfico). Em relação aos métodos baseados em usuário, os valores de RMSE demonstram uma similaridade na acurácia entre as implementações com similaridade do cosseno (círculos, em tons de laranja), que apresentam certa volatilidade, e entre as que utilizam o PCC (losangos, em tons de azul), com comportamento mais estável.

A Tabela 5.5 contém os valores médios gerais de MAE e RMSE encontrados com esse segundo teste. Ratificando o resultado encontrado no primeiro teste e na análise do gráfico 5.7, a implementação do SR utilizando o método CF baseado em usuário, com $k = 5$ e função de semelhança PCC se mostrou superior em acurácia e foi implementado no aplicativo para os testes da fase 4.

5.5.2 Comparação do Tempo de Treinamento

A alteração no tipo do método CF influencia apenas na execução da primeira etapa do SR, por esse motivo o teste de desempenho das variações do método foram realizados considerando somente essa etapa. Esse teste foi realizado ao mesmo tempo que o segundo

Método CF	k	Fun. Semelhança	MAE	RMSE
Baseado em usuário	5	PCC	0.611	0.802
Baseado em usuário	4	PCC	0.628	0.815
Baseado em usuário	3	PCC	0.634	0.824
Baseado em usuário	3	PCC	0.641	0.829
Baseado em usuário	4	Cosseno	0.655	0.843
Baseado em usuário	5	Cosseno	0.658	0.861
Baseado em item	3	Cosseno	0.909	1.06
Baseado em item	4	Cosseno	0.914	1.062
Baseado em item	5	Cosseno	0.918	1.068
Baseado em item	5	PCC	0.926	1.074
Baseado em item	4	PCC	0.927	1.073
Baseado em item	3	PCC	0.928	1.074

Tabela 5.5: Acurácia das doze variações do SR no teste de simulação de uso do sistema.

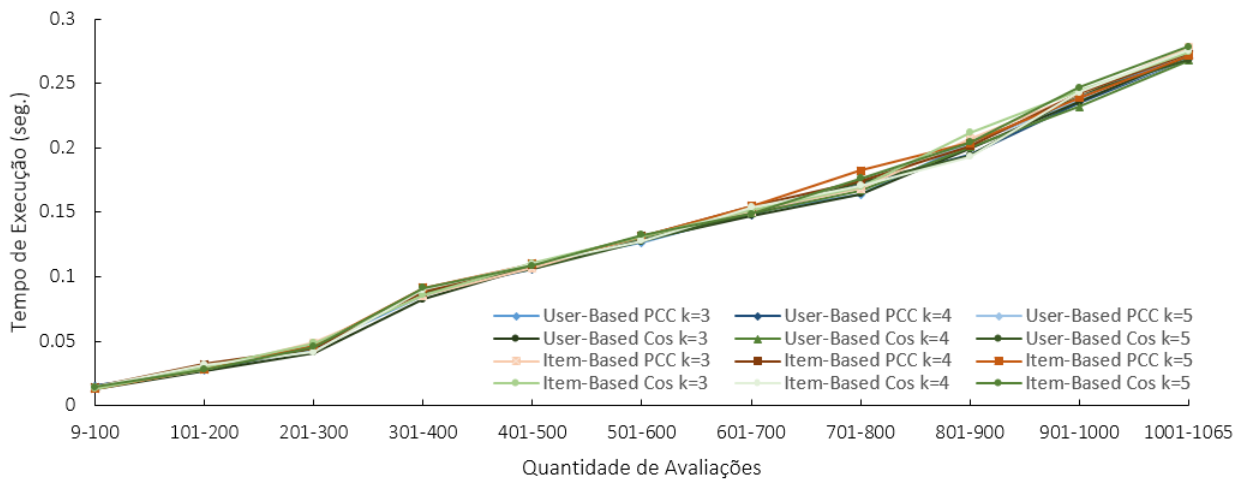


Figura 5.8: Média de tempo de execução das 12 implementações do SR por grupos de 100 treinamentos, com acréscimo de avaliações

teste de acurácia, apresentado na Seção 5.5.1, através da captura dos tempos de início e término da execução do método de recomendação para cada treinamento realizado. O gráfico da figura 5.8 apresenta os resultados encontrados.

Pelo gráfico fica clara a equivalência na complexidade de execução do SR entre todas as 12 variações. Isso implica que a escolha da implementação do SR, dentre as 12 variações, não é relevante do ponto de vista de tempo de execução. Assim, manteve-se a escolha da variação que apresentou o melhor desempenho, em relação à acurácia, para a aplicação na fase 4 da pesquisa.

5.6 Fase 4: Segunda Avaliação On-line

Esta última etapa da pesquisa teve o objetivo de confirmar o melhor desempenho da implementação do SR, definida na fase 3, e verificar, assim como feito na primeira avaliação

on-line na fase 2, se o SR demonstra uma evolução, em termos de acurácia nas recomendações, com o aumento da base de transações. Essa avaliação *on-line* foi realizada nos sábados em que o Parque CienTec esteve em operação normal do período de 26 de outubro a 9 de novembro de 2019.

Um total de 31 usuários utilizaram o sistema, gerando 407 avaliações. A média de idade dos usuários que forneceram essa informação foi de 31 anos. A Tabela 5.6 apresenta informações consolidadas das avaliações, sem considerar as 30 transações relacionadas a idade do usuário.

Métrica	Itens	Preferências	Total
Número de avaliações	160	217	377
Nota média das avaliações	4,3	4,1	4,2
Quantidade média de avaliações	5,2	7	12,2

Tabela 5.6: Estatísticas de avaliações dos participantes da segunda avaliação *on-line*.

5.6.1 Acurácia

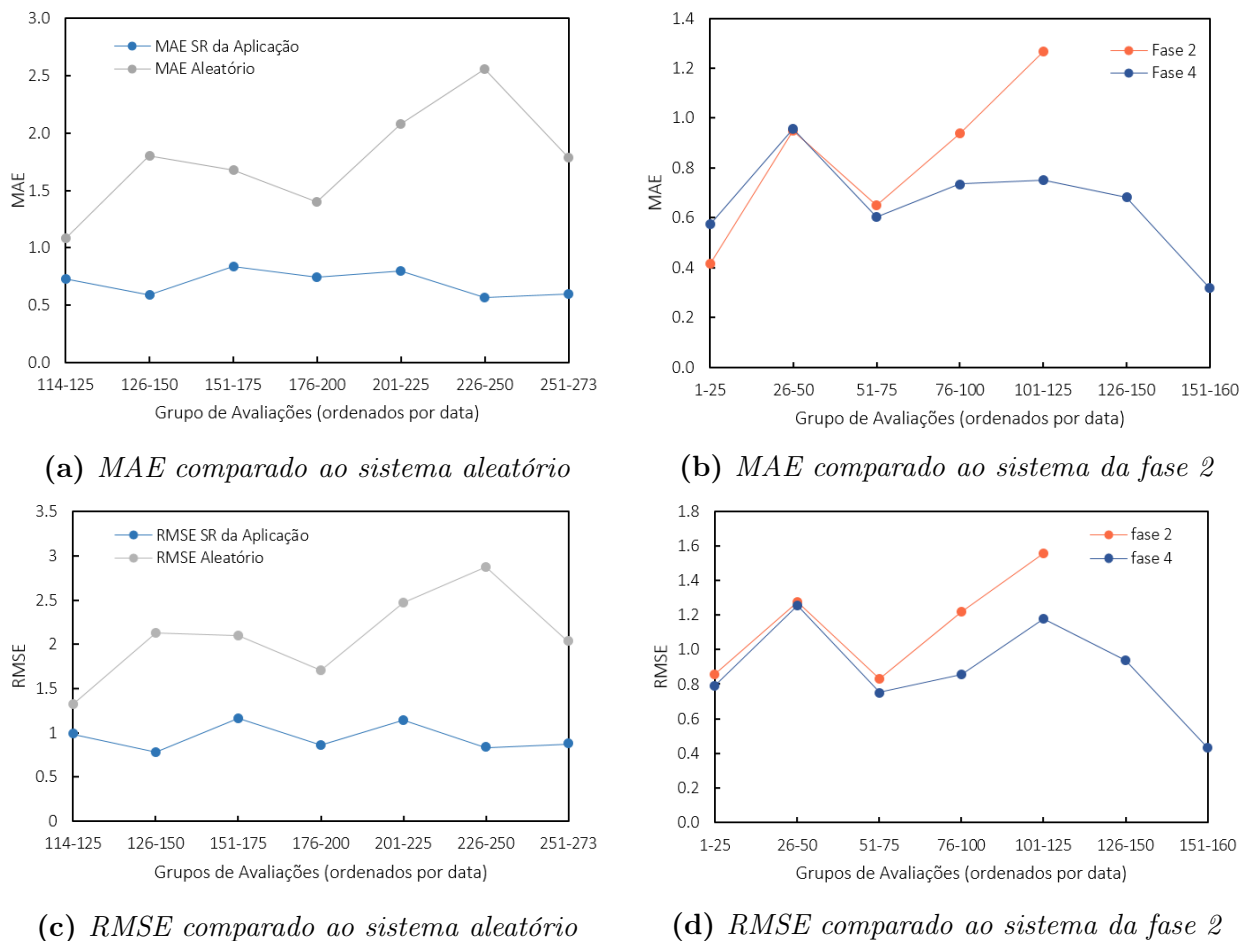


Figura 5.9: Comparação das métricas MAE e RMSE entre o SR do aplicativo e o aleatório, e em relação ao sistema aplicado na fase 2 da pesquisa

O novo método CF implementado, baseado em usuário com $k = 5$ e função de semelhança PCC, também foi avaliado em relação às métricas MAE e RMSE. Os gráficos das figuras 5.9a e 5.9c apresentam os resultados de acurácia do sistema em comparação com o sistema aleatório, onde cada ponto é a média da métrica para um grupo de 25 avaliações de itens feitos pelos usuários. Esses grupos foram formados pela divisão da lista de avaliações ordenada cronologicamente. Por essa característica, pode-se também verificar por este gráfico se houve uma melhora ao longo do tempo à medida que a base de avaliações aumentava. Essa melhora não pode ser confirmada por essa análise, que aparenta configurar uma estabilidade no erro de recomendação entre 0,5 e 1 para o MAE, e 0,7 e 1,3 para o RMSE.

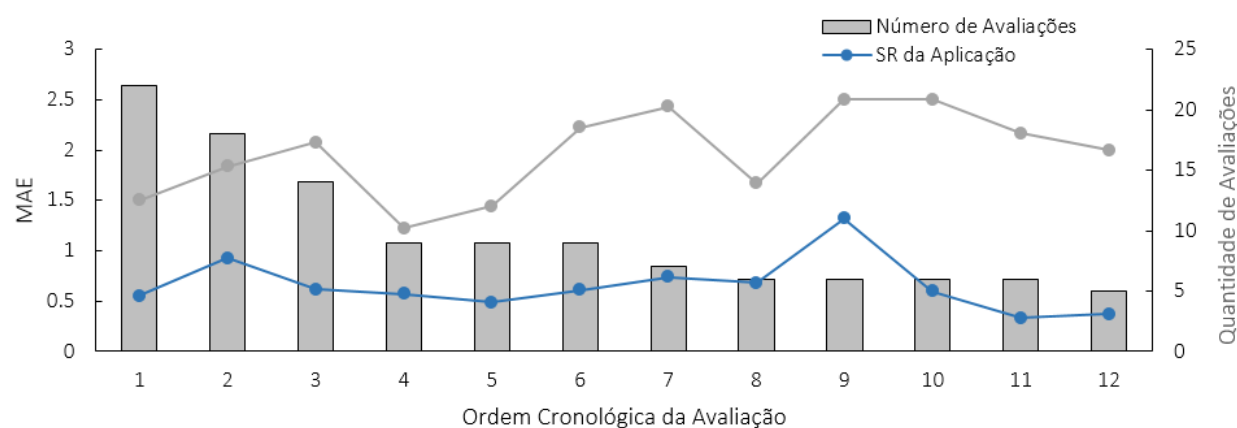
Novamente o SR proposto possui uma acurácia muito superior ao aleatório, porém, o objetivo nesse caso é verificar se ele supera o desempenho obtido pelo sistema que usava o método original, na primeira avaliação *on-line* (fase 2). Para confirmar isso, os valores obtidos na segunda avaliação *on-line* (fase 4) foram comparados aos obtidos na fase 2. Os gráficos das figuras 5.9b e 5.9d apresentam essa comparação, para MAE e RMSE respectivamente. Os resultados dos dois sistemas foram dispostos novamente em grupos de 25 avaliações, em ordem cronológica. Como na fase 2, apenas 113 avaliações de itens haviam sido feitas, contra 160 da fase 4, o novo sistema possui mais pontos no gráfico. Os resultados da nova implementação mostrados nos gráficos indicam, em geral, uma leve melhora em relação à anterior, em ambas as métricas, e a média geral de ambas as métricas apresentou uma melhora considerável, como será explicado a seguir. Nota-se também que os gráficos de MAE e RMSE tem comportamentos muito semelhantes entre si, inclusive para as duas versões do sistema de recomendação. Isso indica que não há um problema maior com notas de recomendação muito distantes das reais avaliações do usuário, o que geraria um RMSE mais próximo do sistema aleatório do que o MAE.

Seguindo os indicadores anteriores, as médias gerais de MAE e RMSE na fase 4, que foram 0,692 e 0,955, também foram melhores que as médias da fase 2, de 0,799 e 1,132, respectivamente. A diferença entre as médias das duas fases foi de 0,107 para o MAE, melhora bem mais significativa que a verificada nos testes comparativos da fase 3 entre os dois métodos, onde a diferença foi de apenas 0,010 no primeiro teste e 0,017 no segundo. A diferença do RMSE entre os métodos das fases 2 e 4 foi de 0,177, também bem superior as diferenças encontradas nos testes da fase 3, que foram de 0,002, no primeiro teste, e 0,013 no segundo.

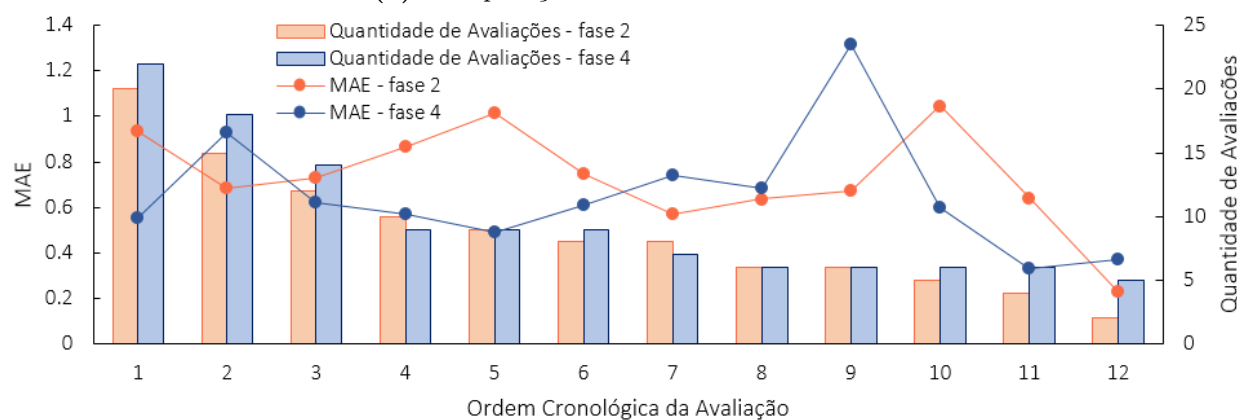
Quando comparados a avaliações feitas de sistemas propostos em outras pesquisas, os resultados deste sistema também se mostram positivos. Em [Su et al., 2019] o SR proposto, do tipo híbrido baseado no sistema KB e informações de contexto e redes sociais, obteve notas de MAE entre 0,6 e 0,9, e RMSE entre 0,6 e 1,6. Nessa pesquisa os autores também desenvolveram um aplicativo *Android*, o qual eles pediram para que um grupo de participantes adicionasse, entre outras informações, suas notas de preferência para os itens recomendados. Embora essa pesquisa tenha utilizado as mesmas métricas de acurácia, os resultados de desempenho não devem ser diretamente comparados, devido a possíveis diferenças na

metodologia de avaliação e no contexto de aplicação.

O sistema nesta fase também foi analisado do ponto de vista da posição cronológica das avaliações por usuário, para comparar a acurácia das recomendações feitas em momento de poucas informações do usuário e quando já se tem disponível uma série de avaliações do mesmo. A figura 5.10a mostra essa análise comparando os valores de MAE com o sistema aleatório, e, na figura 5.10b, entre as fases 2 e 4. Como na Seção 5.4, os valores de MAE para posições com menos de 5 avaliações foram omitidos, por terem, estatisticamente, baixa representatividade.



(a) Comparação com o sistema aleatório



(b) Comparação com o sistema da fase 2

Figura 5.10: Comparação do MAE por grupo de avaliações feitas na mesma posição cronológica pelos usuários, entre o sistema da aplicação e o aleatório, e com o da fase 2 da pesquisa.

Os resultados seguem um padrão muito parecido aos obtidos na fase 2, com valores bem superiores ao aleatório nas posições onde há uma base mínima de avaliações, como das posições 1 ao 11, onde há pelo menos 5 avaliações. Os valores apresentados nas últimas posições são menos precisos e devem ter menor peso na análise.

Os gráfico apresentam novamente uma leve tendência de melhora na métrica, com várias flutuações. Essas flutuações podem ter sido causadas por algumas influências fora do formato padrão de utilização do aplicativo, explicadas na Seção 5.7.

A semelhança da acurácia na posição 1 (primeira avaliação de item feita por todos os

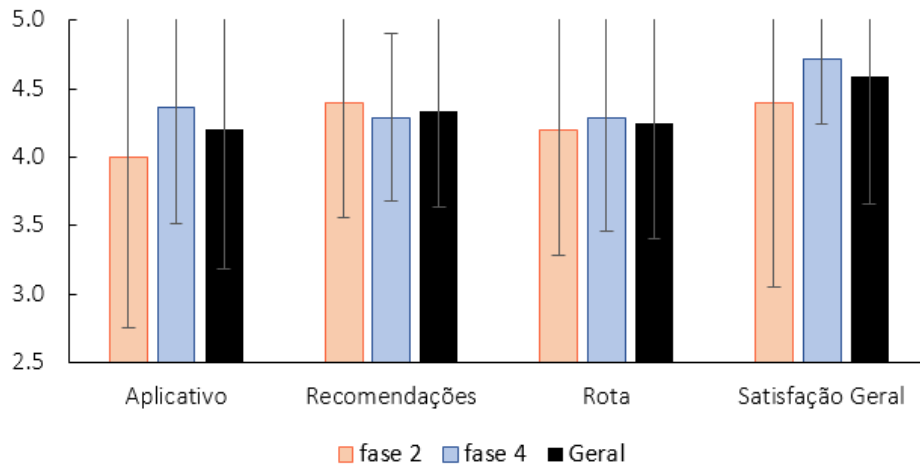


Figura 5.11: Satisfação do usuário com as diferentes características do sistema na fase 2, 4 e no geral.

usuários) em relação a todas as posições com pelo menos 5 avaliações, juntamente com a superioridade em relação ao sistema aleatório indicam que as informações de modelagem contextual utilizadas pelo SR efetivamente reduzem, ou eliminam, o problema de *cold start* para novos usuários. Caso o problema persistisse, as primeiras posições deveriam se aproximar dos valores encontrados para o sistema aleatório, pois nada se saberia do usuário para embasar sua nota de recomendação.

5.6.2 Satisfação do Usuário

Novamente foram coletadas informações de satisfação do usuário com relação ao aplicativo, recomendações, rota e satisfação geral. Nesta fase 14 usuários avaliaram sua satisfação. Somando aos usuários da fase 2, foram 24 no total. A figura 5.11 mostra as médias obtidas das avaliações desses usuários, com a comparação entre as avaliações da segunda e quarta fase.

Os resultados mostram pouca variação na satisfação com relação às recomendações e à rota. Em relação às recomendações, como a melhora apresentada no SR na fase 4 é relativamente pequena, o impacto para o usuário pode não ter sido percebido. As variações em relação ao aplicativo e à satisfação geral foram maiores, variaram positivamente, alcançando notas ainda melhores que na fase 2, com média 4,4 para o aplicativo e 4,7 na satisfação geral. Sem considerar a mudança na implementação do SR entre as duas avaliações *on-line* realizadas, houve também a alteração do mapa do parque disponível no aplicativo, que antes era uma visão de satélite do Google Maps e foi substituído pelo mapa do parque (figura 4.5) criado para o aplicativo. Essa mudança pode ter gerado impacto positivo na satisfação com o aplicativo.

De qualquer forma, considerando que os desvios padrão obtidos são bem mais altos que todas as variações de satisfação verificadas entre as duas fases, não há confiança para se afirmar que houve uma real mudança na satisfação do usuário entre as duas avaliações

on-line realizadas.

5.7 Riscos à Validade dos Resultados

No decorrer do Capítulo 4 e 5 foram citados diversos desafios encontrados na aplicação e avaliação do SR proposto. Vários deles poderiam impactar na obtenção dos dados e sua confiabilidade. Nesta seção discorreremos sobre eles.

A frequente visita em grupos aos locais de aplicação do SR deste trabalho poderia causar ruídos na geração dos dados, pois os critérios de avaliação poderiam mudar ao longo da visita por influências de membros diferentes do grupo. Contudo, notou-se que, em muitos desses casos, uma pessoa se mantinha em posse do celular, guiando o grupo e realizando as avaliações, fato que pode ter colaborado para uma maior assertividade nas avaliações.

O risco de fraudes, como já descrito anteriormente, embora não seja nulo, não foi tratado na solução do SR por ser considerado de baixa probabilidade, dado o contexto de aplicação.

A possibilidade de usuários que não visitaram o parque, ou atrações específicas, e as avaliou, também é um risco à análise da pesquisa. Uma medida adotada sobre isso foi a de filtrar os usuários que não fizeram avaliações dentro do parque, como descrito na Seção 5.1. Contudo, o segundo risco, do usuário ter ido ao local mas não ter visitado alguns itens e tê-los avaliado mesmo assim, permanece.

Os usuários também poderiam avaliar o item de forma aleatória para conseguir continuar a rota recomendada, dada a restrição implementada no aplicativo de não permitir, pelo fluxo normal, avançar na rota sem a avaliação do item corrente.

Outra influência no ruído dos resultados é a possibilidade de interação entre os usuários e os funcionários/monitores do parque. Uma possível explicação dos itens realizada por eles a um usuário poderia acarretar uma mudança na avaliação que o usuário faria sobre o item. Durante vários dos dias de uso do aplicativo no parque, os usuários estavam visitando o local ao mesmo tempo em que grupos escolares, ou de outros propósitos, faziam visitas monitoradas. Nessas condições, por exemplo, usuários poderiam acabar ouvindo parte da explicação dos guias que outros que visitaram os mesmos itens não puderam experimentar. As avaliações dadas nesses casos podem ter sofrido interferências baseadas na qualidade das explicações.

Apesar dos riscos apresentados, os resultados das duas avaliações *on-line* demonstraram consistência nos valores de acurácia e satisfação do usuário. Eles consideraram centenas de avaliações de usuários diferentes e os resultados obtidos mantiveram um padrão comportamental similar, se mantendo em níveis parecidos de forma geral e na visão de evolução das avaliações. Para exemplificar, no caso do MAE, as médias dos grupos de 25 avaliações sem mantiveram entre 0,3 e 1,3 para todas as avaliações, ao passo que o sistema aleatório oscila em um intervalo de erros muito maiores, de aproximadamente 1,0 na mínima, atingindo 2,5 na maior média (ver figura 5.9).

Capítulo 6

Conclusões

Pesquisas relacionadas a sistemas de recomendação estão disponíveis para uma variedade de aplicações. Em se tratando das que visam a aplicação no contexto de visita a museus, há pesquisas relativamente mais antigas, que não utilizam aparelhos celulares modernos e, por isso, não teriam condições de obter dados de contexto mais precisos para basear a recomendação. Um outro subgrupo dessas pesquisas, nessa área de aplicação, não apresenta avaliação *on-line* do sistema proposto, ou seja, no ambiente e condições reais de aplicação.

Esta pesquisa propôs um sistema de recomendação, com o objetivo de personalizar a visita em um museu, e o submeteu a avaliações *on-line* com um número significativo de usuários. A avaliação foi feita para verificar o desempenho da aplicação através da mensuração da acurácia e satisfação do usuário com o sistema.

O sistema de recomendação proposto é do tipo híbrido baseado no popular método de filtragem colaborativa em conjunto com informações de contexto. As características do sistema foram definidas com o objetivo de obter maior acurácia / satisfação nas recomendações do que um sistema CF simples, ser fácil de adaptar para um conjunto diferente de itens e também suavizar o problema do *cold start* para novos usuários do sistema. Buscando a adaptabilidade do sistema evitou-se métodos que necessitam de adição de informações de novos itens, como o CBF e KB.

A proposta foi avaliada através de uma avaliação *on-line* em duas fases, onde a primeira gerou dados que foram utilizados para o aperfeiçoamento do sistema aplicado na segunda fase. Isso foi feito para que se obtivesse *feedbacks* e verificasse a melhor combinação para o sistema em relação aos atributos função de semelhança, número k de vizinhos (usuários/itens similares) considerados e abordagem do método CF. As análises realizadas conduziram à escolha da implementação com função de semelhança PCC, $k = 5$ e abordagem CF baseado em usuário.

Respondendo a questão da pesquisa, relacionado ao desempenho do sistema aplicado a museus, o sistema se mostrou eficaz, com altos níveis tanto de acurácia como de satisfação. Os resultados mostram também que a segunda fase gerou melhores resultados, indicando que a análise, e posterior aperfeiçoamento do sistema, com base nos dados da primeira etapa se

mostrou efetiva. Os resultados de acurácia obtidos na etapa final foram de 0,69 para o MAE e 0,95 para o RMSE. A média geral de satisfação do usuário com relação às recomendações foi de 4,4, em uma escala de 1 a 5. Esses resultados apresentam uma importante confirmação da viabilidade do uso de sistemas de recomendação em museus e da aplicabilidade e eficácia do tipo de sistema proposto.

Os resultados também apresentaram certa variabilidade nos valores de acurácia ao longo do tempo, como esperado para uma avaliação *on-line* nessas condições. Verificou-se uma tendência de diminuição do erro médio de suas recomendações na maioria das visões analisadas, o que indica que o sistema conseguiu se adaptar ao contexto de aplicação à medida que novos dados de usuário iam sendo gerados. A modelagem contextual, como técnica para reduzir os efeitos do *cold start*, também se mostra muito eficaz, trazendo ótimos valores de acurácia mesmo nas primeiras transações de novos usuários.

As propostas e resultados desta pesquisa abrem espaço para uma série de pesquisa futuras que incluem:

- **Melhorias na modelagem contextual:** Em relação à solução proposta, pode-se verificar como as informações iniciais de contexto impactam nos resultados do sistema. Por exemplo, informações demográficas do usuário, quais e quantas perguntas relacionadas a preferência do usuário podem melhorar o desempenho do sistema.

Também pode-se propor novas informações de contexto a serem consideradas pelo recomendador, como as relacionadas ao comportamento do usuário. Exemplos de informações desse tipo, algumas das quais já possuem pesquisas relacionadas, incluem a velocidade média de deslocamento do usuário, suas condições físicas e tempo que ele utiliza para visitar os itens, itens que ele consulta mais detalhes no aplicativo ou que passa mais tempo visualizando. Outras possibilidades incluem a aplicação de tecnologias como *Internet of Things*, *QR Code*, realidade virtual/aumentada, etc.

- **Melhorias na pós-filtragem contextual:** A segunda etapa do sistema, onde se encontra a rota recomendada, pode ter oportunidades de melhorias de eficiência no tempo de processamento à medida em que aumentam o número de itens roteáveis. Além disso, outros critérios podem ser estudados para definir a melhor rota.
- **Interface do aplicativo:** Inúmeras melhorias na interface podem ser implementadas para trazer maior fluidez e intuitividade ao usuário. Por exemplo, a possibilidade de ver a localização dos demais itens da lista recomendada de antemão, a criação da conta do usuário onde se salvam as informações de histórico de visitas dele, mais conteúdos multimídia disponíveis no aplicativo, como áudio explicativo dos itens do parque. Cada proposta de melhoria da interface também tem que considerar o impacto sobre o sistema de recomendação.
- **Reprodutibilidade dos resultados:** Pode ser verificada aplicando o sistema em

outras exposições de patrimônio cultural e comparando os resultados com os obtidos no Parque CienTec.

- **Prevenção a fraudes e atividades maliciosas:** Para a utilização do aplicativo em outros locais, é importante estudar a necessidade de incluir métodos de prevenção à manipulação dos resultados do SR pelos usuários, em especial a criação de regras de detecção de usuários falsos.

Referências Bibliográficas

- Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin. *Learning From Data*. AMLBook, 2012. ISBN 1600490069, 9781600490064. 25
- G. Adomavicius and A. Tuzhilin. Incorporating context into recommender systems using multidimensional rating estimation methods. 2018. 19
- C. C. Aggarwal. *Recommender Systems: The Textbook*. Springer Publishing Company, Incorporated, 1st edition, 2016. ISBN 3319296574, 9783319296579. 1, 8, 11, 12, 13, 14, 16, 17, 18, 20, 21, 23, 29, 50
- R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2):207–216, June 1993. ISSN 0163-5808. doi: 10.1145/170036.170072. URL <http://doi.acm.org/10.1145/170036.170072>. 34
- T. C. Araújo. Apprecommender: um recomendador de aplicativos gnu/linux. Master’s thesis, Instituto de Matemática e Estatística - USP, 2011. 7, 11, 12, 13, 14, 22, 25
- S. Arumugam, A. Brandstädt, T. Nishizeki, et al. *Handbook of graph theory, combinatorial optimization, and algorithms*. Chapman and Hall/CRC, 2016. 92
- R. Bell, Y. Koren, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(08):30–37, aug 2009. ISSN 1558-0814. doi: 10.1109/MC.2009.263. 11
- I. Benouaret and D. Lenne. Personalizing the museum experience through context-aware recommendations. In *2015 IEEE International Conference on Systems, Man, and Cybernetics*, pages 743–748, Oct 2015. doi: 10.1109/SMC.2015.139. 3, 15, 32
- M. Both Suzy Edith and W. Yu. Friend recommendation system based on mobile data. pages 326–329, 08 2018. doi: 10.1109/ESAIC.2018.00082. 4
- A. Bright, J. Kay, D. Ler, K. Ngo, W. Niu, and A. Nuguid. Adaptively recommending museum tours. In *In Proceedings: Workshop on Smart Environments and Their Applications to Cultural Heritage at UbiComp 2005*, pages 29–32, 2005. 3

- R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, Nov 2002. ISSN 1573-1391. doi: 10.1023/A:1021240730564. URL <https://doi.org/10.1023/A:1021240730564>. 15, 16
- L. Cao, J. Tao, and C. Bilian. Implementation of personalized scenic spots route recommendation system. pages 1–6, 08 2018. doi: 10.1109/ICCSE.2018.8468845. 33, 36
- M.-J. A. E. A. Chelini and S. A. G. B. d. C. Lopes. ExposiÃ§Ãem museus de ciÃ: reflexÃe critÃpara anÃlise. *Anais do Museu Paulista: HistÃe Cultura Material*, 16:205 – 238, 12 2008. ISSN 0101-4714. URL http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0101-47142008000200007&nrm=iso. 3
- H. Chourabi, T. Nam, S. Walker, J. R. Gil-Garcia, S. Mellouli, K. Nahon, T. A. Pardo, and H. J. Scholl. Understanding smart cities: An integrative framework. In *2012 45th Hawaii International Conference on System Sciences*, pages 2289–2297, Jan. 2012. doi: 10.1109/HICSS.2012.615. 2
- A. Desvallées and F. Mairesse. *Conceitos-chave de Museologia*. Comitê Brasileiro do Conselho Internacional de Museus : Pinacoteca do Estado de São Paulo, 2013. ISBN 978-85-8256-025-9. 27
- T. Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861 – 874, 2006. ISSN 0167-8655. doi: <https://doi.org/10.1016/j.patrec.2005.10.010>. URL <http://www.sciencedirect.com/science/article/pii/S016786550500303X>. ROC Analysis in Pattern Recognition. 22
- L. E. M. Fernández. Recommendation system for netflix. Technical report, Faculty of Science Business Analytics at VRIJE UNIVERSITEIT AMSTERDAM, 2018. 1, 12, 13
- G. Guo, J. Zhang, Z. Sun, and N. Yorke-Smith. Librec: A java library for recommender systems. In *UMAP Workshops*, 2015. 49
- Y.-M. Huang, C.-H. Liu, and C.-Y. Lee. Designing a personalized guide recommendation system to mitigate information overload in museum learning. *Educational Technology and Society*, 15:150–166, 01 2012. 34
- H. Khan, N. Mannan, S. C. Eshan, M. M. Rahman, K. M. M. H. Sonet, W. U. Hasan, and R. M. Rahman. Tourist spot recommendation system using fuzzy inference system. In *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pages 1532–1539, July 2017. doi: 10.1109/FSKD.2017.8392993. 1
- R. Kohavi and R. Longbotham. *Online Controlled Experiments and A/B Testing*, pages 922–929. 01 2017. doi: 10.1007/978-1-4899-7687-1_891. 20

- J. Leiva. *Cultura SP: Hábitos culturais dos paulistas*. Tuva Editora, São Paulo, SP, 2014. 3
- A. Levitin. *Introduction To Design And Analysis Of Algorithms, 3/E*. Pearson Education, 2012. 92
- M. Lima. Brasil já tem mais de um smartphone ativo por habitante. *Link - Estadão*. Disponível em: <<https://link.estadao.com.br/noticias/geral,brasil-ja-tem-mais-de-um-smartphone-ativo-por-habitante-diz-estudo-da-fgv,70002275238>>. Acessado em: 16 de fevereiro de 2019. 1
- K. Lin, K. Lin, and K. Chung. Automatic recommendation system of iptv contents for baseball video. In *2013 International Symposium on Intelligent Signal Processing and Communication Systems*, pages 714–717, Nov 2013. doi: 10.1109/ISPACS.2013.6704642. 1, 2
- G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7:76–80, 01 2003. ISSN 1089-7801. doi: 10.1109/MIC.2003.1167344. URL doi.ieeecomputersociety.org/10.1109/MIC.2003.1167344. 13
- M. Mathias, A. Moussa, F. Zhou, J. Torres-Moreno, M. Poli, D. Josselin, M. El-Bèze, A. C. Linhares, and F. Rigat. Optimisation using natural language processing: Personalized tour recommendation for museums. In *2014 Federated Conference on Computer Science and Information Systems*, pages 439–446, Sep. 2014. doi: 10.15439/2014F336. 35
- Y. Mo, J. Chen, X. Xie, C. Luo, and L. T. Yang. Cloud-based mobile multimedia recommendation system with user behavior information. *IEEE Systems Journal*, 8(1):184–193, March 2014. ISSN 1932-8184. doi: 10.1109/JSYST.2013.2279732. 4
- G. Monico, A. Dal Poz, M. Galo, M. Santos, and L. Oliveira. Acurácia e precisão: revendo os conceitos de forma acurada / accuracy and precision: Reviewing the concepts by means of an accurate procedure. *Boletim de Ciências Geodesicas*, 15:469–483, 07 2009. doi: 10.13140/2.1.1815.8086. 4
- C.-S. Ong, M.-Y. Day, and W.-L. Hsu. The measurement of user satisfaction with question answering systems. *Inf. Manage.*, 46(7):397–403, Oct. 2009. ISSN 0378-7206. doi: 10.1016/j.im.2009.07.004. URL <http://dx.doi.org/10.1016/j.im.2009.07.004>. 34
- R. Oppermann and M. Specht. A nomadic information system for adaptive exhibition guidance. *Archives and Museum Informatics*, 13(2):127–138, Jun 1999. ISSN 1573-7500. doi: 10.1023/A:1016619506241. URL <https://doi.org/10.1023/A:1016619506241>. 3
- G. Pavlidis. Towards a novel user satisfaction modelling for museum visit recommender systems. In M. Duguleană, M. Carrozzino, M. Gams, and I. Tanea, editors, *VR Technologies in Cultural Heritage*, pages 60–75, Cham, 2019a. Springer International Publishing. ISBN 978-3-030-05819-7. 27, 28, 29

- G. Pavlidis. Recommender systems, cultural heritage applications, and the way forward. *Journal of Cultural Heritage*, 35:183 – 196, 2019b. ISSN 1296-2074. doi: <https://doi.org/10.1016/j.culher.2018.06.003>. URL <http://www.sciencedirect.com/science/article/pii/S1296207418302577>. *Modern and Contemporary Art*, 2, 3, 15, 21, 29
- F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor. *Recommender Systems Handbook*. Springer-Verlag, Berlin, Heidelberg, 1st edition, 2010. ISBN 0387858199, 9780387858197. 1, 2, 8, 9, 11, 12, 13, 14, 16, 17, 21, 23
- F. Ricci, L. Rokach, and B. Shapira. *Recommender Systems Handbook*. Springer Publishing Company, Incorporated, 2nd edition, 2015. ISBN 1489976361, 9781489976369. 18, 21, 22, 29, 91
- S. Rossi, F. Barile, C. Galdi, and L. Russo. Artworks sequences recommendations for groups in museums. In *2016 12th International Conference on Signal-Image Technology Internet-Based Systems (SITIS)*, pages 455–462, Nov 2016. doi: 10.1109/SITIS.2016.77. 28, 33, 40
- Y. Salman, A. Abu-Issa, I. Tumar, and Y. Hassouneh. A proactive multi-type context-aware recommender system in the environment of internet of things. In *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pages 351–355, Oct 2015. doi: 10.1109/CIT/IUCC/DASC/PICOM.2015.50. 4
- X. Su, G. Sperli, V. Moscato, A. Picariello, C. Esposito, and C. Choi. An edge intelligence empowered recommender system enabling cultural heritage applications. *IEEE Transactions on Industrial Informatics*, 15(7):4266–4275, July 2019. ISSN 1941-0050. doi: 10.1109/TII.2019.2908056. 77
- W. R. van Hage, N. Stash, Y. Wang, and L. Aroyo. Finding your way through the rijksmuseum with an adaptive mobile museum guide. In *Proceedings of the 7th International Conference on The Semantic Web: Research and Applications - Volume Part I, ESWC'10*, pages 46–59, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-13485-8, 978-3-642-13485-2. doi: 10.1007/978-3-642-13486-9_4. URL http://dx.doi.org/10.1007/978-3-642-13486-9_4. 32, 46, 93
- R. Van Meteren. Using content-based filtering for recommendation. 06 2000. 11
- Y. Wang, N. Stash, L. Aroyo, P. Gorgels, L. Rutledge, and G. Schreiber. Recommendations based on semantically enriched museum collections. *Journal of Web Semantics*, 6(4):283 – 290, 2008. ISSN 1570-8268. doi: <https://doi.org/10.1016/j.websem.2008.09.002>. URL <http://www.sciencedirect.com/science/article/pii/S1570826808000681>. *Semantic Web Challenge 2006/2007*. 36

- Y. Wang, N. Stash, L. Aroyo, L. Hollink, and G. Schreiber. Using semantic relations for content-based recommender systems in cultural heritage. In *Proceedings of the 2009 International Conference on Ontology Patterns - Volume 516*, WOP'09, pages 16–28, Aachen, Germany, Germany, 2009. CEUR-WS.org. URL <http://dl.acm.org/citation.cfm?id=2889761.2889763>. 3, 32
- B. Xiang, Z. Zhang, H. Dong, Q. Wu, and L. Hu. Research of mobile recommendation system based on hybrid recommendation technology. In *2013 3rd International Conference on Consumer Electronics, Communications and Networks*, pages 508–512, Nov 2013. doi: 10.1109/CECNet.2013.6703381. 1
- K. Zhu, L. Zhang, and A. Pattavina. Learning geographical and mobility factors for mobile application recommendation. *IEEE Intelligent Systems*, 32(3):36–44, May 2017. ISSN 1541-1672. doi: 10.1109/MIS.2017.52. 4

Apêndice A

Definições e Algoritmos

A.1 Funções de Semelhança

Funções de semelhança são usadas para comparação entre objetos. Eles são bastante aplicados em sistemas de recomendação *neighborhood based* pois permitem diferenciar os vizinhos mais e os menos relevantes para a recomendação [Ricci et al., 2015]. Das funções de semelhança disponíveis, duas delas se destacam por serem mais empregadas em SRs:

Semelhança do Cosseno

Consideremos dois vetores x_a e x_b , que representam as avaliações feitas pelos usuários a e b , respectivamente, a dois conjunto de itens I_a e I_b , onde cada elemento r_{ui} em x_a ou x_b , representa a avaliação feita pelo usuário u ao item i , temos que a função de semelhança do cosseno $SC(x_a, x_b)$ é definida pela equação:

$$SC(x_a, x_b) = \frac{\sum_{i \in I_a \cap I_b} r_{ai} r_{bi}}{\sqrt{\sum_{i \in I_a} r_{ai}^2 \sum_{j \in I_b} r_{bj}^2}} \quad (\text{A.1})$$

Uma característica que pode ser considerada negativa [Ricci et al., 2015] para a aplicação da semelhança do cosseno em sistema de recomendação é que ela não considera diferenças na média e variância dos dois usuários. Assim usuários que tenham gostos similares, mas sejam mais rigorosos que outros ao avaliar os itens, podem não ser considerados tanto quanto usuários com gostos diferentes, mas mesmo estilo de avaliação.

PCC

O PCC, do inglês *Pearson Correlation Coefficient*, é empregado em SRs para atender ao problema descrito na função anterior, pois ele considera as médias e desvio padrão das avaliações dos usuários em seu cálculo. Utilizando as definições anteriores, a função PCC é dada por:

$$PCC(x_a, x_b) = \frac{\sum_{i \in I_a \cap I_b} (r_{ai} - \bar{r}_{ai})(r_{bi} - \bar{r}_{bi})}{\sigma_a \sigma_b} \quad (\text{A.2})$$

A.2 Complexidade NP-Difícil

Para definir este conceito, primeiramente é necessário apresentar as classes de problemas P, NP e NP-completo. Problemas da classe P são aqueles que podem ser resolvidos em tempo polinomial, incluindo o pior cenário de eficiência no tempo de processamento, por algoritmos determinísticos [Levitin, 2012]. Denota-se que problemas desse tipo possuem complexidade $O(p(n))$ onde n é o tamanho da entrada de dados e $p(n)$ é um polinômio.

A classe de problemas NP, que são aqueles que podem ser resolvidos por algoritmos não-determinísticos polinomiais¹. Nota-se que $P \subseteq NP$, pois, para todo algoritmo determinístico, existe um algoritmo não determinístico polinomial equivalente.

Por fim, um problema x é considerado do tipo NP-completo se x pertence a classe NP e todo problema em NP pode ser reduzido, em tempo polinomial, ao problema x .

Com essas definições, pode-se agora definir o conceito de problema NP-difícil como "Aqueles ao menos tão difíceis para encontrar um algoritmo que os resolva quanto os problemas NP-completo"[Levitin, 2012]. Com essa definição, indica-se que a classe de problemas NP-completo é tanto um subgrupo de NP como de NP-Difícil, e os problemas NP-Completo são a única intersecção entre as classes NP-difícil e NP.

A.3 Algoritmo de Dijkstra

O algoritmo de *Dijkstra* é proposto para encontrar o menor caminho em um grafo a partir de um dado vértice x até todos os demais vértices do conjunto de vértices V do grafo. Ele é utilizado nos casos onde o problema não apresenta distâncias (custo de ir de um vértice a outro) negativas [Arumugam et al., 2016]. O algoritmo utiliza um vetor $c(i)$ que armazena as distâncias mínimas possíveis entre x e os demais i vértices e um vetor $\text{pred}(i)$ que armazena os vértices predecessores até chegar a i . O algoritmo também define uma propriedade para cada vértice, guardando a informação se o menor caminho até ele já foi encontrado ou não. Chamaremos essa propriedade de $\text{encontrado}(i)$, onde $i \in V$.

O algoritmo começa inicializando o vetor c com infinito para todo i , onde $i \neq x$, e $c(x) = 0$, e $\text{encontrado}(j) = \text{Falso}$ para todo $j \in V$.

A iteração principal do algoritmo consiste em selecionar o vértice v que possui o menor valor em c e atualizar $\text{encontrado}(v) = \text{Verdadeiro}$. Então para todas as arestas disponíveis (v, i) , de v até o vértice i , onde $\text{encontrado}(i) = \text{Falso}$, verifica-se se o custo de v até i mais $c(v)$ é menor que $c(i)$. Se verdadeiro, $c(i)$ é atualizado com esse valor e atualiza-se $\text{pred}(i) = v$. Repete-se essa iteração tantas vezes quanto for a quantidade de vértices presentes no grafo.

¹Algoritmos que em ao menos uma de suas possíveis soluções, acerta o resultado, consegue validá-lo como correto e a validação pode ser feita em tempo polinomial.

A.4 Greedy Nearest Neighbors Search

Trata-se de um algoritmo para encontrar um caminho entre uma lista de vértices L , com tamanho t , de modo que minimize a distância necessária para visitar todos os vértices, ou seja, tenha o menor custo para chegar até todos eles. Ele pode ser aplicado ao problema do caixeiro viajante, encontrando uma solução sub-ótima [van Hage et al., 2010].

Para realizar essa tarefa, executa-se t vezes o algoritmo de Dijkstra com a modificação de que se o vértice v que acabou de ser visitado - seguindo nomenclatura da seção anterior: encontrado(v) = Verdadeiro - pertencer a L , retorna-se v como a resposta da iteração. A iteração segue o procedimento abaixo, onde v_O é o vértice de origem:

1. Executar algoritmo de Dijkstra modificado a partir de v_O . Retorna-se o vértice v_D , onde $v_D \in L$ e é o mais próximo de v_O de todos os vértices em L
2. Define v_D como visitado e $v_O = v_D$
3. Repete-se passos anteriores até que todos os vértices em L tenham sido visitados.

Este é considerado um algoritmo guloso por utilizar os caminhos mínimos entre pares de vértices como a solução global para o problema original. Embora ele não encontre sempre a melhor solução, ele é muito eficiente se comparado à solução ideal, que é do tipo NP-difícil.

