

Universidade Federal de Campina Grande  
Centro de Engenharia Elétrica e Informática  
Coordenação de Pós-Graduação em Ciência da Computação

Diversificação em Sistemas de Recomendação  
Utilizando uma Abordagem Baseada em Aspectos

Ricardo Santos de Oliveira

Tese submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande - Campus I como parte dos requisitos necessários para obtenção do grau de Doutor em Ciência da Computação.

Área de Concentração: Ciência da Computação  
Linha de Pesquisa: Metodologia e Técnicas da Computação

Leandro Balby Marinho  
(Orientador)

Campina Grande, Paraíba, Brasil  
©Ricardo Santos de Oliveira, abril/2020

## Resumo

Muitas abordagens de sistemas de recomendação bem sucedidas são baseadas na otimização de uma função de utilidade explícita definida em termos da diferença entre o item previsto e os itens presentes no histórico do usuário. Dessa forma, busca-se aumentar a probabilidade de itens relevantes serem recomendados ao usuário. Apesar de efetiva, essa abordagem pode levar a recomendações que, apesar de relevantes, sejam óbvias e desinteressantes, uma vez que o usuário poderia, em muitas ocasiões, descobrir estes itens de interesse sem o auxílio de um sistema de recomendação. Muitas abordagens investigam esse problema tentando evitar listas de recomendação cujos itens sejam muito similares entre si, caracterizando uma recomendação diversa, com respeito a alguns aspectos do item, sem que isso acarrete numa consequente degradação da relevância dos itens recomendados. Entretanto, os usuários podem ter preferências diferentes no que diz respeito a quais aspectos devem ser diversificados e quais aspectos devem ser semelhantes ao seu gosto passado e/ou atual.

Neste trabalho, essas considerações são levadas em conta para a proposição de modelos baseados em otimização multiobjetivo para a geração de listas de recomendação que buscam o balanço ótimo entre os aspectos que devem ser mantidos fixos, maximizando a similaridade com itens previamente consumidos pelo usuário, e aspectos que devem ser diversificados, minimizando-se a similaridade com outros itens na lista de recomendação. Para avaliar os modelos propostos foram conduzidos experimentos utilizando-se dados reais de históricos de usuários do Last.fm, em conjunto com metadados acerca dos itens recomendados, obtidos em múltiplas fontes. Foram realizadas avaliações tanto offline, isto é, utilizando todo o histórico do usuário para produzir a avaliação, como online, onde recomendações são geradas ao longo de uma linha do tempo, levando em consideração alterações nos interesses do usuário que devem ser refletidas na recomendação, o que reflete a aplicação real de um sistema de recomendação. Os resultados obtidos nos experimentos demonstram a efetividade dos modelos propostos tanto offline como online, em comparação a abordagens estado-da-arte da literatura com propostas semelhantes.

**Palavras-chave:** Recomendação. Diversificação. Multiobjetivo. Multi-Armed Bandits.

## Abstract

Many successful recommendation approaches are based on the optimization of some explicit utility function defined in terms of the misfit between the predicted and the actual items of the user. Although effective, this approach may lead to recommendations that are relevant but obvious and uninteresting. Many approaches investigate this problem by trying to avoid recommendation lists in which items are very similar to each other (aka diversification) with respect to some aspect of the item. However, users may have very different preferences concerning what aspects should be diversified and what should match their past/current preferences. In this work, we take this into consideration by proposing solutions based on multiobjective optimization for generating recommendation lists featuring the optimal balance between the aspects that should be held fixed (maximize similarity with users actual items) and the ones that should be diversified (minimize similarity with other items in the recommendation list).

In order to evaluate the proposed models, simulations were carried out using real Last.fm datasets, together with metadata about the recommended items. Evaluations were carried out both offline, that is, using the entire user history to produce the recommendation, and online, where recommendations are generated along the timeline, taking into account changes in the user's interests that should be reflected in the recommendation, which resembles the actual application of a recommendation system. The results obtained in the experiments demonstrate the effectiveness of the models proposed both offline and online, compared to state-of-the-art approaches in the literature with similar proposals.

**Keywords:** Recommender Systems. Diversification. Multiobjective. Multi-Armed Bandits.

## **Agradecimentos**

A minha esposa Umara, que aceitou o desafio de conduzir a nossa família praticamente sozinha enquanto precisei me afastar para a realização deste trabalho. A meus filhos, que entenderam e aceitaram a minha ausência. Obrigado pelo apoio incondicional e por apostar junto comigo neste projeto de vida.

A meu pai e minha irmã que, mesmo à distância, foram incentivadores indispensáveis. A minha mãe, que não pôde ver a conclusão deste trabalho, mas que permaneceu presente.

Aos amigos do laboratório, com quem convivi tão intensamente nesta jornada. Compartilhadores de conhecimento, de bons momentos, de cansaço, de incentivos. Passamos por isso juntos, e valeu a pena.

A Leandro. Excelente profissional, apoiador, conselheiro, amigo. Tenho o privilégio de encerrar meu doutorado sendo fã do meu orientador, em um processo que tão comumente gera desgaste e desentendimento. Somente um ser humano da sua qualidade torna isso possível.

Muito obrigado a todos.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Motivação . . . . .	2
1.2	Objetivos . . . . .	4
1.3	Contribuições . . . . .	6
1.4	Estrutura . . . . .	7
<b>2</b>	<b>Contextualização e Definição do Problema</b>	<b>9</b>
2.1	Sistemas de Recomendação . . . . .	9
2.1.1	Tipos Básicos de Sistemas de Recomendação . . . . .	10
2.1.2	Recomendações Baseadas em Relevância . . . . .	11
2.1.3	Avaliação Além da Acurácia . . . . .	12
2.2	Recomendação Multiobjetivo . . . . .	13
2.2.1	Otimidade de Pareto . . . . .	13
2.3	Formalização do Problema . . . . .	15
2.4	Considerações Finais . . . . .	17
<b>3</b>	<b>Modelos Base</b>	<b>18</b>
3.1	Algoritmos Evolucionários para Otimização Multiobjetivo . . . . .	18
3.1.1	NSGA-II . . . . .	19
3.2	Multi-Armed Bandits . . . . .	23
3.2.1	Exploration x Exploitation Dilemma . . . . .	23
3.2.2	Ranked Bandits . . . . .	27
3.2.3	MAB para Recomendação Multiobjetivo . . . . .	28
3.2.4	Considerações Finais . . . . .	29

<b>4</b>	<b>Trabalhos Relacionados</b>	<b>31</b>
4.1	Avaliação Além da Acurácia . . . . .	31
4.2	Recomendações Multiobjetivo . . . . .	34
4.3	Recomendação usando Multi-Armed Bandits . . . . .	35
4.4	Outras Abordagens . . . . .	37
4.5	Considerações Finais . . . . .	38
<b>5</b>	<b>Sistemas de Recomendação Multiobjetivo Baseados em Aspectos</b>	<b>39</b>
5.1	Diversificação Multiobjetivo Baseada em Aspectos . . . . .	39
5.2	Representação de Itens e Métricas de Similaridade . . . . .	40
5.2.1	Definição dos Aspectos . . . . .	40
5.2.2	Métricas de Similaridade . . . . .	41
5.3	Modelo MOAD . . . . .	45
5.4	Modelo <i>Ranked Bandits</i> . . . . .	47
5.5	Aplicação do <i>Ranked Bandits</i> em Recomendações Online . . . . .	48
5.5.1	Propensão à Diversidade Diária . . . . .	48
5.5.2	Execução do Modelo . . . . .	49
5.6	Considerações Finais . . . . .	51
<b>6</b>	<b>Tratamento e Análise dos Dados</b>	<b>52</b>
6.1	Coleta de Dados e Preparação . . . . .	52
6.2	Verificação da Propensão a Diversidade . . . . .	54
6.3	Cenários de Propensão . . . . .	55
6.4	Distribuição de Usuários . . . . .	56
6.4.1	Histórico Completo . . . . .	56
6.4.2	Histórico Particionado em Dias . . . . .	56
6.5	Considerações Finais . . . . .	58
<b>7</b>	<b>Avaliação Offline</b>	<b>60</b>
7.1	Determinação de Propensão à Diversidade . . . . .	61
7.2	Algoritmos <i>Baselines</i> . . . . .	62
7.2.1	<i>Random</i> . . . . .	62

7.2.2	Mais Popular . . . . .	63
7.2.3	<i>User-Based Collaborative Filtering</i> (UBCF) . . . . .	63
7.2.4	<i>Maximal Marginal Relevance</i> (MMR) . . . . .	63
7.2.5	<i>Relevance-based eXplicit Query Aspect Diversification</i> (RxQuAD) .	63
7.3	Avaliação do MOAD . . . . .	64
7.3.1	Protocolo de Avaliação . . . . .	64
7.3.2	Ajuste de Parâmetros do <i>Non-dominated Sorting Genetic Algorithm II</i> (NSGA II) . . . . .	65
7.4	Avaliação <i>Ranked Bandits</i> . . . . .	67
7.4.1	Ajuste de Parâmetros . . . . .	67
7.5	Resultados . . . . .	69
7.5.1	Resultados MOAD . . . . .	71
7.5.2	Resultados <i>Ranked Bandits</i> . . . . .	72
7.5.3	Exemplo de Recomendação . . . . .	74
7.5.4	Benefícios Colaterais . . . . .	74
7.6	Considerações Finais . . . . .	77
<b>8</b>	<b>Avaliação Online</b>	<b>79</b>
8.1	Protocolo de Avaliação Online de Recomendações . . . . .	80
8.1.1	Divisão do Histórico . . . . .	80
8.1.2	Execução . . . . .	80
8.1.3	<i>Baselines</i> . . . . .	81
8.2	Resultados . . . . .	82
8.3	Considerações Finais . . . . .	86
<b>9</b>	<b>Conclusões</b>	<b>87</b>
9.1	Trabalhos Futuros . . . . .	88

# Lista de Siglas

**CBRS** *Content-based Recommender Systems*. 10

**CDB** *Counterfactual Dueling Bandits*. 36

**CF** *Collaborative Filtering*. 10

**DCG** *Discounted Cumulative Gain*. 11

**ILD** *Intra-List Diversity*. 12, 16, 40, 49, 54–56, 61, 62, 66

**ILS** *Intra-List Similarity*. 12, 16

**kNN** *k-Nearest Neighbors*. 11, 63

**MAB** *Multi-Armed Bandits*. xii, 6–8, 18, 23–25, 27, 29, 31, 35, 36, 47–50, 67, 88, 89

**MAE** *Mean Absolute Error*. 11

**MLHD** *Music Listening Histories Dataset*. 56

**MMR** *Maximal Marginal Relevance*. vi, 32–34, 62, 63, 74, 81, 84

**MOAD** *Multiobjective Aspect Diversification*. 6, 7, 45, 47, 60, 62, 64, 69, 71–75, 77, 81, 87

**MOEA** *Multiobjective Evolutionary Algorithm*. 7, 19, 20, 29, 35, 45, 60

**MSI** *Mean Self-information*. 75

**NSGA II** *Non-dominated Sorting Genetic Algorithm II*. vi, xi, xii, 19–23, 34, 35, 45, 46, 64–66



**OF** *Occurrence Frequency*. 43

**PD-ILD** *Previous Day Intra-List Diversity*. 49, 50, 81

**RMSE** *Root Mean Square Error*. 11

**RxQuAD** *Relevance-based eXplicit Query Aspect Diversification*. vi, 33, 34, 62, 63, 81

**TD** *Topic Diversification*. 32, 33

**TF-IDF** *Term Frequency - Inverse Document Frequency*. 37

**UBCF** *User-Based Collaborative Filtering*. vi, 45, 47, 48, 63, 64, 74, 76

**UCB** *Upper Confidence Bound*. 26, 35, 36, 48, 88

**xQuAD** *eXplicit Query Aspect Diversification*. 33

# Lista de Figuras

1.1	Efeito portfólio no site da Amazon. . . . .	3
2.1	Fronteira de Pareto. . . . .	14
3.1	Calculando o crowding distance para a i-ésima solução em sua faixa de dominância. . . . .	21
3.2	Passo principal do NSGA II. . . . .	23
3.3	Multi-Armed Bandit. . . . .	24
3.4	Política de Seleção de Ações Upper Confidence Bound. . . . .	27
5.1	Melhor e pior caso para a métrica de contemporaneidade. . . . .	42
5.2	Fluxo do MOAD. . . . .	46
5.3	Fluxo dos <i>Ranked Bandits</i> . . . . .	47
5.4	Modelo <i>Ranked Bandits</i> para execução online. . . . .	50
6.1	Distribuição de usuários nos perfis de propensão a diversidade considerando histórico completo. . . . .	57
6.2	Distribuição de usuários avaliados nos perfis de propensão a diversidade considerando histórico completo. . . . .	57
6.3	Distribuição de usuários nos perfis de propensão a diversidade considerando histórico completo. . . . .	58
6.4	Distribuição acumulada de usuários de acordo com o número de mudanças de cenários de propensão. . . . .	59
7.1	Comparativo da evolução da média dos objetivos escalarizados a cada 100 iterações. . . . .	68

---

7.2	Resultados para métricas Diversidade e Afinidade, todos os algoritmos, cenários 3, 4, 6 e 7. . . . .	70
7.3	Resultados de MSI de todos os usuários da amostra testada. . . . .	76
7.4	Resultados de acurácia de todos os usuários da amostra testada. . . . .	76
8.1	Resultados para recomendação online em cenários com diversificação de um aspecto (resultados do <i>Ranked Bandits</i> em verde). . . . .	83
8.2	Resultados para recomendação online em cenários com diversificação de mais de um aspecto. . . . .	85

# Lista de Tabelas

6.1	Estatísticas acerca da base de dados utilizada . . . . .	54
6.2	Cenários avaliados . . . . .	55
7.1	ILD e desvio padrão da base analisada, por aspecto. . . . .	61
7.2	Valores padrão para os parâmetros do NSGA II. . . . .	65
7.3	<i>p</i> -valor resultante dos testes de Wilcoxon para a determinação dos parâmetros do NSGA II. . . . .	66
7.4	<i>p</i> -valor do teste de Wilcoxon para diferentes valores de $\epsilon$ no algoritmo <i>Ranked Bandits <math>\epsilon</math>-greedy</i> . . . . .	67
7.5	<i>p</i> -valor do teste de Wilcoxon com opção <i>greater</i> na métrica <i>Diversidade</i> , favorável ao MOAD, para diferentes cenários de recomendação. . . . .	71
7.6	<i>p</i> -valor do teste de Wilcoxon com opção <i>greater</i> na métrica <i>Afinidade</i> , favorável ao MOAD, para diferentes cenários de recomendação. . . . .	71
7.7	<i>p</i> -valor do teste de Wilcoxon com opção <i>greater</i> na métrica <i>Diversidade</i> , favorável aos Ranked Bandits, para diferentes cenários de recomendação. . . . .	74
7.8	<i>p</i> -valor do teste de Wilcoxon com opção <i>greater</i> na métrica <i>Afinidade</i> , favorável aos Ranked Bandits, para diferentes cenários de recomendação. . . . .	74
7.9	Top-10 recomendações para um usuário real do Last.fm. . . . .	78
8.1	Valores dos parâmetros usado na avaliação online do <i>Ranked Bandits</i> . . . . .	81
8.2	Comparativo de custos computacionais entre os modelos MOAD e Ranked Bandits para pior caso. . . . .	81
8.3	<i>p</i> -valor do teste de Wilcoxon com opção <i>greater</i> na métrica favoráveis ao <i>Ranked Bandits</i> , para diferentes cenários de recomendação da avaliação online. . . . .	86

# Lista de Quadros

3.1	Pseudocódigo do algoritmo NSGA II. . . . .	22
3.2	Pseudocódigo do <i>Multi-Armed Bandits</i> (MAB). . . . .	25
3.3	Pseudocódigo de Ranked Bandits. . . . .	28

# Capítulo 1

## Introdução

Alguns serviços prestados na Web, tais como sites de vendas, *streaming* de músicas e filmes, redes sociais, etc., possuem conteúdo vasto e dinâmico. Com isso, deixar exclusivamente para o usuário a tarefa de encontrar conteúdo de interesse pode ser ineficiente e desestimulante. Grande parte dos provedores de serviços usa sistemas de recomendação para automatizar a tarefa de descoberta de conteúdo, oferecendo itens ao usuário sem que este necessite realizar uma consulta para isso. O uso de sistemas de recomendação pode ser um agente de acréscimo em vendas, aumento da satisfação do usuário, com um consequente aumento da fidelidade do usuário no uso de determinado serviço, podendo ainda resultar numa maior compreensão do usuário do que ele de fato deseja (RICCI; ROKACH; SHAPIRA, 2015).

Neste contexto, o desenvolvimento de sistemas de recomendação no sentido de aumentar sua eficiência desperta interesse tanto no espectro dos negócios como na academia. O aumento da eficiência no uso deste tipo de sistema está diretamente ligado à relevância dos itens recomendados para o usuário, o que fez com que muitas abordagens avaliassem seus modelos usando apenas métricas de relevância como precisão. Porém, diversas linhas de pesquisa têm sido realizadas no sentido de considerar também diversidade, novidade, serendipidade, entre outros fatores, como impactantes na eficiência do sistema de recomendação. Esta tese propõe novas soluções no sentido de aumentar a diversidade dos itens recomendados, sem que isso resulte em prejuízo da relevância destes itens para os usuários.

## 1.1 Motivação

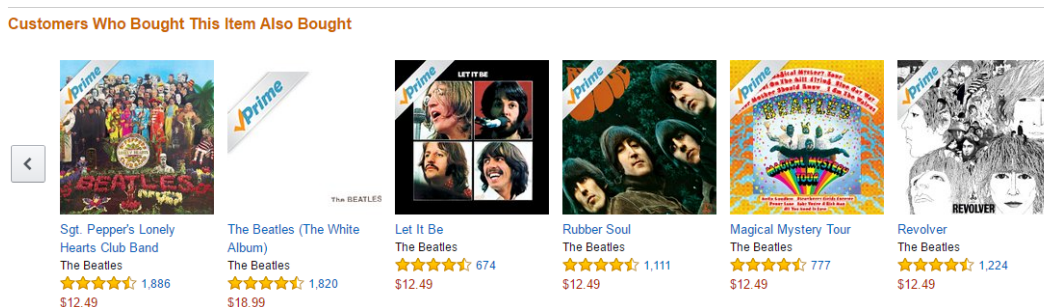
Em cenários em que a disponibilidade de conteúdo é vasta e dinâmica, tal como ocorre em serviços de *streaming* de músicas, vídeos, etc., usuários são rapidamente sobrecarregados com um grande e sempre crescente número de possibilidades de escolha. Sistemas de recomendação são ferramentas reconhecidamente bem sucedidas para resolver, ou ao menos amenizar este problema, no sentido de oferecer ao usuário, de forma automática, uma seleção de itens várias ordens de grandeza menor que a totalidade dos itens. O uso de sistemas de recomendação é fundamentado na premissa de que as pessoas recorrem a recomendações para as atividades que pretendem realizar, seja na forma de indicações de conhecidos, seja através da leitura de críticos especializados, ou mesmo a solicitação expressa de recomendações, como no caso de um empregador que pretende contratar um funcionário (RICCI; ROKACH; SHAPIRA, 2015).

Historicamente, o desenvolvimento de sistemas de recomendação é focado na relevância dos itens recomendados para o usuário. Por relevância, entende-se a compatibilidade entre os itens recomendados e os interesses do usuário. Considerando aplicações de diferentes domínios, um item relevante seria uma música ou artista compatível com a preferência musical do usuário, no caso de recomendações musicais, ou itens que o usuário pretende adquirir, no caso de um site de vendas, ou ainda pessoas conhecidas, no caso de recomendações de amigos em redes sociais. O aferimento da relevância de recomendações é geralmente feito através de métricas como a acurácia, através de experimentos offline, onde a razão entre os itens recomendados e os itens presentes no histórico do usuário (selecionados para teste) é calculada.

Uma das maneiras de aumentar a efetividade de um sistema de recomendação é personalizar as recomendações, com base no histórico de consumo de itens por parte do usuário. Diferentes usuários possuem diferentes interesses, o que faz com que recomendações não personalizadas, como aquelas baseadas na popularidade dos itens, embora possam agradar a um bom número de usuários, não sejam efetivas para tantos outros. Dessa forma, as abordagens de sistemas de recomendação mais bem sucedidas são geralmente aquelas que promovem a personalização, por exemplo, otimizando alguma função de utilidade definida pela similaridade entre os itens recomendados e o histórico de consumo do usuário.

Apesar deste tipo de abordagem ser efetivo em muitos cenários, algoritmos de recomendação que otimizam este tipo de função são propensos a oferecer recomendações que são relevantes, porém, possivelmente desinteressantes para o usuário (ZHANG; HURLEY, 2008). Nestes casos pode ocorrer o chamado efeito portfólio (BURKE, 2002), em que os itens recomendados são muito similares uns aos outros. Uma situação de efeito portfólio é mostrada na Figura 1.1. Trata-se da recomendação feita no site da Amazon<sup>1</sup> ao se realizar uma pesquisa pelo álbum *Abbey Road*, dos *Beatles*. Álbuns dos *Beatles* que não foram comprados previamente por um certo usuário que apenas comprou *Beatles* é certamente uma recomendação extremamente acurada. Entretanto, um fã dos *Beatles* poderia ser positivamente surpreendido ao receber recomendações de outras bandas, eventualmente diferentes dos Beatles em muitos aspectos como gênero musical, mas que tiveram a participação do membro dos Beatles Paul McCartney. Neste caso, tem-se uma recomendação que mantém algum grau de relação com as preferências do usuário, nesse caso Paul McCartney, entretanto, menos óbvia.

Figura 1.1: Efeito portfólio no site da Amazon.



Fonte: Autoria própria.

Apesar da personalização, uma recomendação unicamente baseada em relevância não consegue lidar com o problema do efeito portfólio (VARGAS; CASTELLS, 2011). Muitas abordagens surgiram tendo como objetivo lidar com esse problema por meio da promoção de diversidade entre os itens recomendados (ZIEGLER et al., 2005; ZHANG; HURLEY, 2008; VARGAS; CASTELLS; VALLET, 2011; NOIA et al., 2015). Este objetivo é geralmente alcançado com o uso de mecanismos que evitam a geração de listas de recomendação cujos itens sejam muito similares entre si com respeito a alguns aspectos (por exemplo, gênero

<sup>1</sup><https://www.amazon.com/>



da música ou do filme). Tais abordagens podem aumentar, potencialmente, a satisfação do usuário ao fornecer recomendações menos óbvias. Entretanto, os usuários têm diferentes preferências no que diz respeito a que aspectos devem ser diversificados, enquanto outros aspectos devem manter semelhança com seu histórico de consumo. Por exemplo, um usuário pode ser bastante conservador em se tratando do gênero musical que costuma ouvir, selecionando sempre Bossa Nova, mas ser bastante aberto a diversidade em outro aspecto como a nacionalidade, tendo interesse em conhecer como esse gênero é tocado em diferentes países, como Japão ou Índia.

Um possível efeito colateral decorrente da diversificação é o aumento de novidade e serendipidade. Por novidade, entende-se a presença de itens não familiares ao usuário, mas que casam com as suas preferências, enquanto serendipidade é a sensação de encontrar um item valioso, surpreendentemente positivo, que dificilmente seria encontrado numa busca exaustiva (HERLOCKER et al., 2004). Entretanto, promover diversificação de forma inadvertida pode prejudicar a relevância das recomendações, sendo necessário portanto firmar um compromisso entre o grau ótimo de diversificação e relevância. Em outras palavras, nesse cenário, tem-se então dois objetivos possivelmente conflitantes a serem otimizados para cada usuário:

1. Encontrar os itens que maximizem a similaridade com as suas preferências;
2. Encontrar os itens que minimizem a similaridade entre os elementos selecionados para a recomendação.

O fato de haver dois objetivos nos quais a tentativa de se otimizar um deles possivelmente resulta em prejuízo para o outro dá um caráter de otimização multiobjetivo ao problema. Aliado ao fato de que cada usuário tem diferentes preferências quanto a aspectos que aceita serem diversificados, tem-se o problema de pesquisa objeto de estudo deste trabalho.

## 1.2 Objetivos

Embora os problemas levantados, assim como as soluções propostas, sejam um quadro genérico em sistemas de informação, este trabalho é voltado para a realização de recomendações em domínio musical, especificamente para a recomendação de artistas desconhecidos pelo

usuário, ou seja, artistas que não constem previamente do seu histórico de consumo. Para essas recomendações serão utilizados metadados acerca dos artistas disponíveis na base de dados, tendo sido estes metadados divididos em aspectos musicais, onde um ou mais atributos de artista podem compor um aspecto musical. Ao considerar os diferentes aspectos musicais busca-se considerar para as recomendações características que influenciem na preferência musical do usuário, como gênero musical ou localidade de origem do artista. Isso resultaria em uma recomendação mais precisamente ajustada aos seus interesses. Por exemplo, seríamos capazes de modelar um usuário fã de bandas de *Heavy Metal* que tem interesse, dentro deste gênero, de conhecer artistas de *Heavy Metal* de diferentes países, que tiveram sua carreira iniciada em diferentes décadas.

O objetivo principal deste trabalho é produzir um sistema de recomendação que gere itens diversificados de acordo com um determinado conjunto de aspectos, ao mesmo tempo em que estes itens mantêm semelhança com as preferências do usuário, quando considerado outro conjunto de aspectos. Este objetivo pode ser dividido em três objetivos específicos:

- Determinar, a partir do histórico do usuário, a sua propensão em consumir música de maneira diversificada ou não, levando em consideração diferentes aspectos musicais. Isso permitirá usar essa informação para configurar o sistema de recomendação de maneira personalizada.
- Conduzir experimentos offline, usando o histórico dos usuários para avaliar os resultados, comparando os modelos propostos com algoritmos estado-da-arte no tópico de diversificação em sistemas de recomendação.
- Conduzir experimentos online, gerando recomendações com base no consumo diário do usuário, numa avaliação mais próxima do cenário real de uso de um sistema de recomendação. Os experimentos devem mostrar a capacidade de adaptação dos modelos a mudanças nos interesses dos usuários ao longo da linha do tempo, mantendo a recomendação satisfatória frente a essa mudança, em tempo escalável para uma grande quantidade de usuários.

## 1.3 Contribuições

Em conformidade com os objetivos estabelecidos, foi proposto o modelo *Multiobjective Aspect Diversification* (MOAD), uma abordagem de recomendação que usa o histórico de consumo de itens do usuário para determinar a acurácia da lista gerada, ao mesmo tempo em que compara os itens da lista par a par para verificação de diversidade. Utilizando metadados acerca dos itens a serem recomendados, foram identificados aspectos musicais que possibilitam determinar, de acordo com a preferência do usuário, algumas características aceitáveis para serem diversificadas e outras que devem ser conservadas de acordo com seu histórico. A divisão em aspectos é utilizada para determinar que metadados são utilizados para se comparar os itens da lista entre si e com o histórico do usuário, determinando índices de diversidade e acurácia, respectivamente. Dessa forma, buscou-se garantir que os índices de diversidade e de acurácia fossem otimizados em relação ao conjunto de metadados que forma os aspectos selecionados para cada um dos objetivos. Foi utilizada a base de dados do Last.fm<sup>2</sup>, um serviço online que armazena todo o histórico de consumo de músicas de seus usuários. Essa base foi completada com metadados acerca dos artistas musicais, provenientes de Music Brainz<sup>3</sup> e DBPedia<sup>4</sup>.

Mantendo o objetivo de obter uma recomendação simultaneamente diversa e adequada à preferência do usuário, foi proposto também o modelo de *Ranked Bandits*, em que a função multiobjetivo usada no modelo MOAD é executada menos vezes, permitindo ao modelo atingir resultados satisfatórios com menos tempo de execução. Além desta característica, modelos de MAB são eficientes em adaptar-se rapidamente a mudanças de interesse do usuário, permitindo a realização de uma avaliação *online*, ou seja, fornecendo recomendações que levam em consideração o interesse momentâneo do usuário.

Foram produzidas recomendações para cada um dos usuários da base de dados do experimento, utilizando cada um dos algoritmos *baseline*, o MOAD e os *Ranked Bandits*. Essas recomendações foram avaliadas em relação a ambos os objetivos e a significância estatística de suas distribuições foi testada, para que se pudesse ter um gráfico de comparação.

Destacam-se, então, as seguintes contribuições:

---

<sup>2</sup><http://www.last.fm/>

<sup>3</sup><https://musicbrainz.org/>

<sup>4</sup><http://wiki.dbpedia.org/>

1. O modelo MOAD, que usa um *Multiobjective Evolutionary Algorithm* (MOEA) para obter uma otimização aproximada dos objetivos propostos. O MOAD garante uma aproximação do ponto de *trade-off* ótimo entre aspectos a serem diversificados e aspectos a serem conservados.
2. O modelo *Ranked Bandits*, que usa múltiplas execuções de MABs para construir uma lista de recomendação, permitindo sua avaliação quanto à diversidade, além do ganho em adaptação a mudanças de interesse do usuário.
3. Experimentos foram conduzidos comparando os resultados obtidos pelo MOAD e pelos *Ranked Bandits* com algoritmos estado-da-arte em diversificação de recomendações. De acordo com os experimentos realizados, MOAD e *Ranked Bandits* obtiveram, na grande maioria dos casos testados, resultados superiores em diversidade, mantendo a competitividade em acurácia, quando comparados aos algoritmos propostos como *baseline*.
4. A obtenção da propensão do usuário a diversificar ou não os itens consumidos, propensão esta que é utilizada para personalização das recomendações. Como os diferentes aspectos possuem atributos de diferentes tipos, variando entre categóricos, contínuos, binários, etc., foram estabelecidas métricas de comparação individuais por aspecto, com o intuito de capturar as características de cada aspecto e ter uma noção mais específica de quanto dois itens são semelhantes ou diversos de acordo com cada aspecto.
5. O protocolo de geração de recomendações online usando o modelo *Ranked Bantits*, que se adapta a mudanças de perfil de propensão do usuário com base no seu último dia de consumo. Este protocolo se aproxima da situação real, onde o histórico de uso do usuário vai sendo incrementado ao longo do tempo e o sistema de recomendação deve considerar essas novas informações.

## 1.4 Estrutura

O restante deste documento está organizado da seguinte forma: o capítulo 2 traz uma fundamentação teórica, referente aos conceitos necessários para a compreensão deste trabalho,

notadamente no que se refere a otimização multiobjetivo, além de uma definição formal do problema abordado por este trabalho; o capítulo 3 mostra os modelos de recomendação com base em algoritmos evolucionários e MAB, usados como referencial teórico para as abordagens propostas nesta tese; o capítulo 4 discorre sobre alguns trabalhos relacionados, organizados pela relação que guardam para com este trabalho; o capítulo 5 mostra as soluções de recomendação multiobjetivo propostas para o problema apresentado, discorrendo sobre como os algoritmos utilizados foram adaptados e modelados para recomendação baseada em aspectos; o capítulo 6 mostra as fontes de dados usadas para esta pesquisa, descrevendo como estes dados foram limpos e estruturados. Traz também um estudo da propensão de usuários à diversidade, identificando os diferentes perfis de usuários que justificam a abordagem por aspectos proposta; enquanto os capítulos 7 e 8 mostram os resultados obtidos nos experimentos com os modelos propostos. Parte destes resultados foram publicados no ISMIR 2017 - The 18th International Society for Music Information Retrieval Conference, em Suzhou, China (OLIVEIRA et al., 2017); por fim, o capítulo 9 mostra as conclusões do trabalho e indica futuros caminhos a seguir na continuidade desta pesquisa.

## Capítulo 2

# Contextualização e Definição do Problema

Este capítulo apresenta uma contextualização do desenvolvimento de sistemas de recomendação e de sua evolução enquanto tópico de pesquisa. Também são apresentados os fundamentos sobre otimização multiobjetivo, técnica que serviu como base para o modelo proposto neste trabalho. Por último é apresentada a formulação do problema para o qual foi apresentado um modelo de solução e foram realizados experimentos neste trabalho.

### 2.1 Sistemas de Recomendação

Sistemas de Recomendação são ferramentas de software e algoritmos que têm o intuito de sugerir ao usuário de uma plataforma de serviços online itens com boa probabilidade de interessá-lo (RICCI; ROKACH; SHAPIRA, 2015). O tópico de Sistemas de Recomendação ganhou importância nos anos 90, devido ao surgimento e crescimento de plataformas de comércio eletrônico (AGGARWAL, 2016). Atualmente, além do *e-commerce*, sistemas de recomendação estão presentes nas principais plataformas de serviços disponibilizados via Web, tais como *streaming* de músicas, filmes e vídeos, redes sociais, plataformas de publicação de trabalhos científicos, sites de apoio a turismo, entre diversas outras aplicações.

Em um sistema de recomendação utiliza-se o termo "item" para designar o que o sistema recomenda para os usuários, por exemplo, filmes no Netflix, páginas no Facebook e vídeos no YouTube. Um sistema de recomendação coleta informações acerca das preferências dos

usuários para montar uma lista de itens com boas chances de se adequarem a estas preferências. Esta lista pode ser formada por itens de interesse geral, como os mais consumidos, mais recentes, etc., e ser destinada a todos os usuários, constituindo um sistema de recomendação não personalizado. Embora listas não personalizadas possam ser úteis em certos contextos, a elaboração de listas personalizadas com base em informações coletadas a partir do comportamento do usuário, como é feito ao se utilizar seu histórico de consumo, tendem a ser mais efetivas (RICCI; ROKACH; SHAPIRA, 2015).

As informações utilizadas por um sistema de recomendação para entender as preferências do usuário podem ser fornecidas por este de maneira explícita, na forma de uma nota ou avaliação para o item consumido, mas também de maneira implícita, como por exemplo, um clique em um item de catálogo online (BOBADILLA et al., 2013). Dependendo do domínio da aplicação, o ato de consumir um item já indica a sua relevância para o usuário. Por exemplo, em um serviço de *streaming* de músicas, cada música ouvida pode ser considerada um *feedback* positivo implícito do usuário.

Embora sistemas de recomendação tenham surgido com o objetivo inicial de auxiliar usuários com experiência pessoal insuficiente acerca dos itens disponíveis a fazer escolhas (RESNICK; VARIAN, 1997), a vasta disponibilidade de itens de grande parte dos serviços prestados na Web, e a consequente dificuldade de escolha mesmo para usuários experientes, tornou os sistemas de recomendação praticamente onipresentes.

### 2.1.1 Tipos Básicos de Sistemas de Recomendação

Aggarwal (2016) classifica os modelos básicos de sistemas de recomendação como baseados na interação entre usuários e itens, tais como compras, consumo e realização de avaliações; e baseados em atributos de usuários e itens, tais como perfis textuais do usuário, características dos itens e palavras-chave. A primeira categoria define os sistemas de Filtragem Colaborativa – *Collaborative Filtering* (CF) – em que usuários com perfis similares ao usuário-alvo (considerados os vizinhos mais próximos) são usados para a extração de recomendações de itens ainda não consumidos, e portanto potencialmente desconhecidos, pelo usuário-alvo. A segunda categoria define os Sistemas de Recomendação Baseados em Conteúdo – *Content-based Recommender Systems* (CBRS) – sistemas que recomendam itens similares àqueles que o usuário consumiu no passado.

Há abordagens de sistemas de recomendação, chamadas sistemas baseados em conhecimento (*Knowledge-based Recommender Systems*), que, ao invés de usar o histórico de avaliações ou comportamento do usuário, usam dados, normalmente enriquecidos com algum tipo conhecimento de domínio, obtidos em fontes externas, para criar a recomendação. Grande parte das abordagens do estado da arte, entretanto, são abordagens híbridas, sendo uma combinação entre estes diferentes tipos de sistemas.

### 2.1.2 Recomendações Baseadas em Relevância

Um item é dito relevante para um usuário se, dependendo do domínio do serviço, o item se adequa à preferência, ao desejo ou aos interesses do usuário. As pesquisas iniciais acerca de sistemas de recomendação foram baseadas em variantes de um algoritmo *k-Nearest Neighbors* (kNN) capaz de prever o quanto um usuário  $u$  iria gostar de um item  $i$  (KONSTAN; RIEDL, 2012). Com predição de avaliação do usuário como base, as avaliações dos sistemas eram feitas com base em acurácia (BREESE; HECKERMAN; KADIE, 1998), de forma a verificar se as avaliações previstas eram de fato relevantes para o usuário.

As avaliações com base em métricas de relevância quantificam a diferença entre os itens recomendados e os itens que o usuário efetivamente gostaria de ter recebido como recomendação, além de fornecer um valor numérico para o estabelecimento de uma classificação entre algoritmos candidatos (GUNAWARDANA; SHANI, 2015).

As métricas de relevância podem ser classificadas em métricas de avaliação de predição, tais como Erro Médio Absoluto – *Mean Absolute Error (MAE)* – e Raiz do Erro Quadrado Médio – *Root Mean Square Error (RMSE)*, que representam a magnitude do erro entre o *rating* previsto, cujas primeiras posições formam a lista de recomendações, e o *rating* real, que mostra os itens que efetivamente vieram a ser consumidos pelo usuário (SCHAFER et al., 2007). Há também métricas de avaliação da recomendação baseadas em noções de conjunto, tais como *Precision* e *Recall*, e métricas de avaliação da lista de recomendação como um ranking, ou seja, levam em consideração a posição dos itens na lista de recomendação, como por exemplo Ganho Cumulativo Descontado – *Discounted Cumulative Gain (DCG)* (BOBADILLA et al., 2013).



### 2.1.3 Avaliação Além da Acurácia

Diversos autores entendem que a geração de uma recomendação baseada em acurácia para garantir a relevância dos itens não resulta necessariamente em uma recomendação útil (VARGAS; CASTELLS, 2011; MCNEE; RIEDL; KONSTAN, 2006; ZHANG; HURLEY, 2008). A preocupação é que, apesar de relevante, a lista seja preenchida com itens muito semelhantes entre si, resultando em uma recomendação, a depender do domínio da aplicação, tediosa, preenchida por itens que o usuário poderia ter encontrado sem a ajuda de um sistema de recomendação. Uma recomendação com essas características poderia ter menos utilidade que uma recomendação que apresente itens relevantes porém mais diversos entre si.

Nesse contexto, objetivos como diversidade, novidade e serendipidade ganharam importância. Por diversidade, entende-se o quanto os itens que constam na lista de recomendações são dissemelhantes entre si. Uma recomendação diversa amplia o leque de possibilidades de escolha do usuário, aumentando o número de usuários que encontrarão ao menos um item que desperte seu interesse imediato na lista de recomendações (VARGAS; CASTELLS; VALLET, 2012). A busca por novidade, ou seja, por itens desconhecidos, pode ser requisito de alguns usuários, enquanto a serendipidade é o sentimento de surpresa positiva decorrente da descoberta de um item valioso (HERLOCKER et al., 2004).

Segundo Vargas e Castells (2011), diversidade e novidade, em particular, são dimensões chave para a utilidade de recomendações em domínios reais e um tópico fundamental para que as pesquisas na área continuem avançando. Com o surgimento de sistemas de recomendação com foco em outras métricas de utilidade que não a acurácia, foram propostas diversas outras métricas de avaliação. Ziegler et al. (2005), por exemplo, propôs o uso de *Intra-List Similarity* (ILS) para medir a similaridade entre os elementos de uma lista e a métrica derivada desta, *Intra-List Diversity* (ILD), que verifica a dissemelhança entre os itens da lista, conforme a equação 2.1,

$$ILD(R) = \sum_{i=1}^{N-1} \sum_{j=i+1}^N d(r_i, r_j) \quad (2.1)$$

onde  $R$  corresponde à lista de recomendação,  $N$  à quantidade de itens na lista,  $r_i$  e  $r_j$  aos itens recomendados nas posições  $i$  e  $j$ , respectivamente, e  $d(x, y)$  é uma função que mede a dissimilaridade entre os itens  $x$  e  $y$ . Outros trabalhos sugerem o uso do *coverage* como métrica de avaliação de diversidade, medindo, em termos percentuais, quantos dos valores

existentes para um atributo foram cobertos pela recomendação em relação ao total de valores disponíveis (KÜÇÜKTUNÇ et al., 2013; PARAMBATH; USUNIER; GRANDVALET, 2016). Por exemplo, considerando uma diversificação de gênero musical, caso existam 500 gêneros disponíveis, uma recomendação que incluía 80 gêneros possui um *coverage* de 16%.

Apesar da notável necessidade de considerar diversidade e novidade como métricas de utilidade, é arriscado um sistema de recomendação otimizar essas dimensões esquecendo a relevância, uma vez que o sistema pode não mais atender diretamente às preferências do usuário (RIBEIRO et al., 2012). Um exemplo que ilustra esse problema em potencial seriam recomendações completamente aleatórias, que teriam grandes chances de serem diversas em uma base de dados com muitos itens. Entretanto, a chance de uma recomendação gerada dessa forma não agradar ao usuário é obviamente muito alta. Existe um aparente conflito entre as métricas diversidade e acurácia, pois a otimização da diversidade pode levar a recomendações completamente alheias às preferências do usuário. Isso faz com que a otimização de alguma delas implique na deterioração da outra. Para tratar das duas métricas simultaneamente, têm surgido propostas de recomendação multiobjetivo.

## 2.2 Recomendação Multiobjetivo

Nessa categoria de problema, a otimização de um objetivo implica naturalmente em um impacto negativo em algum dos demais objetivos (GENG et al., 2015). Há duas formas de se considerar uma solução multiobjetivo: uma combinação de objetivos gerando um objetivo único ou uma solução que considera um vetor de objetivos, também chamada de otimização Pareto (ZITZLER; THIELE, 1999), que será detalhada a seguir.

### 2.2.1 Otimalidade de Pareto

Tem-se um problema de otimização multiobjetivo quando se deseja encontrar um elemento que atenda simultaneamente a múltiplos objetivos que possivelmente sejam conflitantes entre si. Em um problema desse tipo, a maximização do valor de um objetivo pode impactar negativamente em pelo menos um dos demais objetivos buscados.

Quando uma função multiobjetivo é utilizada, cada função objetivo produz um valor para cada solução candidata avaliada. Considerando-se duas soluções candidatas, diz-se que uma

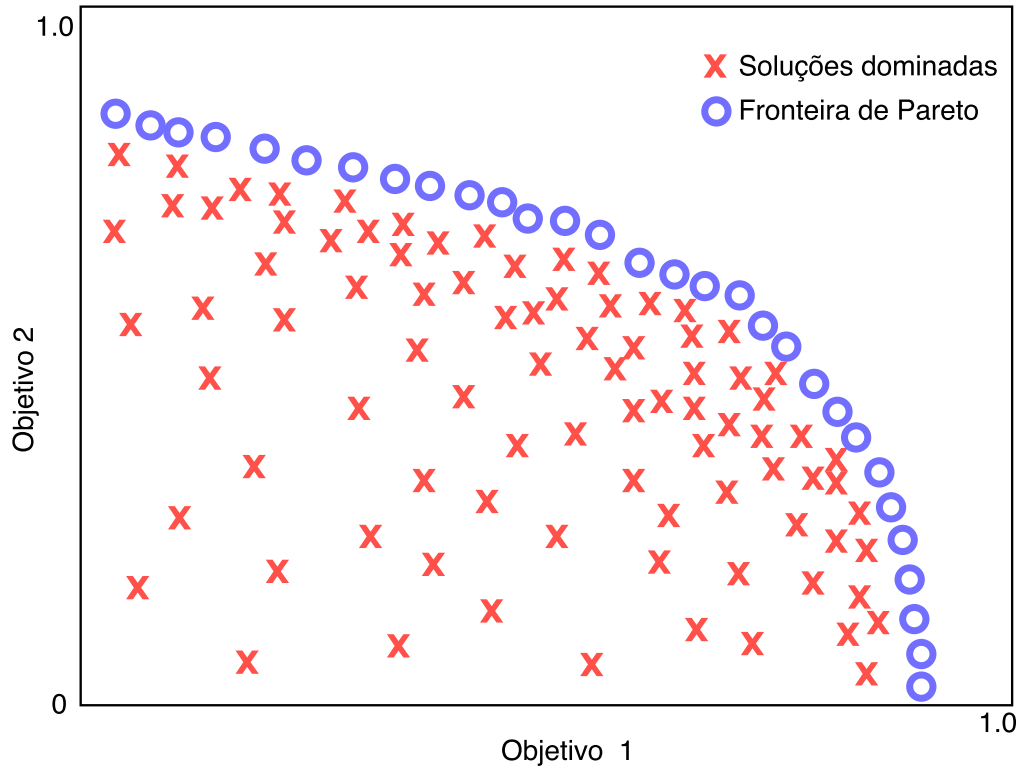
solução domina a outra quando ela, no mínimo, iguala todos os objetivos e supera a outra solução em ao menos um objetivo. Formalmente, dadas as soluções  $s_1$  e  $s_2$ ,  $n$  objetivos e função  $f(s_i) = \{y_{i,1}, y_{i,2}, \dots, y_{i,n}\}$ , diz-se que a solução  $s_1$  domina a solução  $s_2$  se

$$\forall i, y_{1,i} \geq y_{2,i} \quad (2.2)$$

$$\exists j \mid y_{1,j} > y_{2,j}. \quad (2.3)$$

Quando uma solução domina a outra, pode-se dizer que é uma solução melhor. Por outro lado, se não há relação de dominância entre duas soluções, nenhuma delas é a melhor. Quando uma solução não é dominada por nenhuma outra solução, ela é dita uma solução Pareto-ótima.

Figura 2.1: Fronteira de Pareto.



Fonte: Autoria própria.

O conjunto de todas as soluções Pareto-ótimas forma a fronteira de Pareto. A figura 2.1 ilustra a fronteira de Pareto. Os elementos marcados com  $x$  são as soluções dominadas, en-

quanto os elementos marcados com  $o$  são os não dominados, cujo conjunto forma a fronteira de Pareto. Para todos os elementos na fronteira de Pareto pode-se dizer que é impossível melhorar um objetivo sem piorar ao menos um dos demais objetivos (RIBEIRO et al., 2015). Então, no espaço de soluções, estes são os elementos que representam um melhor balanço entre os objetivos.

Determinar a Fronteira de Pareto é tido como uma tarefa que demanda um grande esforço computacional (CECCHINI et al., 2010; RIBEIRO et al., 2012). Uma abordagem força-bruta, por exemplo, envolveria uma enumeração exaustiva de todas as listas de recomendação possíveis para avaliação dos objetivos.

## 2.3 Formalização do Problema

O problema abordado neste trabalho pode ser definido da seguinte forma: dado um usuário alvo  $u \in U$  (onde  $U$  é o conjunto de usuários), seu histórico de consumo de itens  $I_u \subseteq I$  (onde  $I$  é o conjunto de itens) dois conjuntos disjuntos de aspectos musicais  $X$  e  $Y$ , fornecidos pelo usuário ou inferidos dos dados, deseja-se encontrar os  $n$  itens mais similares ao histórico do usuário  $I_u$  levando-se em consideração os aspectos definidos em  $X$  e, ao mesmo tempo, mais dissimilares entre si considerando-se os aspectos definidos em  $Y$ . Neste caso,  $X$  corresponde ao conjunto de aspectos que se deseja manter semelhança ao histórico do usuário, enquanto  $Y$  corresponde ao conjunto de aspectos que é do desejo do usuário que seja diversificado.

Itens podem ter diferentes tipos de metadados associados a si. Tais metadados são, a depender do contexto, conhecidos como atributos, dimensões, contextos ou informações adicionais. Por exemplo, se o item recomendável é um artista musical, pode-se imaginar cada possível gênero musical como sendo um atributo binário, onde são indicados positivamente os gêneros musicais aos quais a produção musical do artista se adequa, sem limites para a quantidade de gêneros associada ao artista. Mais formalmente, seja  $G$  o conjunto de gêneros e  $d : G \times I \rightarrow \{0, 1\}$  uma função que indica se o gênero  $g \in G$  descreve o item  $i \in I$ . O conjunto  $G$  é então um aspecto do item e representa o conjunto de atributos do tipo gênero musical.

De maneira mais genérica, sejam  $A = \{A_1, A_2 \dots, A_t\}$  o conjunto de  $t$  aspectos possí-

veis e  $f_j : A_j \times I \rightarrow \mathbb{R}$  uma função extratora de características para cada atributo  $a \in A_j$  que descreve o item  $i$ . Um item pode ser então representado como um vetor de características relacionadas ao aspecto  $j$ . Por exemplo,  $\vec{i} = (f_1(a_1, i), f_1(a_2, i), \dots, f_1(a_p, i))$  representa um vetor de características do item  $i$  onde  $a_1, a_2, \dots, a_p \in A_1$ .

A entrada para o algoritmo é um usuário  $u \in U$ , seu histórico de consumo de itens  $I_u$ , o tamanho  $n$  da lista de recomendação que se deseja produzir e dois conjuntos disjuntos de aspectos:  $X, Y \subseteq A$  onde  $X \cap Y = \emptyset$ , ou seja, nenhum aspecto pode ser selecionado para ser diversificado e manter semelhança com o histórico ao mesmo tempo. São também levadas em consideração duas funções de similaridade:  $g_a(R)$ , que retorna o ILS dos itens constantes da lista de recomendação  $R$  em relação ao aspecto  $a$ ; e  $h_a(R, I_u)$ , que retorna a similaridade entre  $R$  e os itens presentes no histórico do usuário  $I_u$ . Por fim, seja

$$diversity(R, X) = 1 - \frac{1}{|X|} \sum_{a \in X} g_a(R) \quad (2.4)$$

a ILD para todos os aspectos em  $X$ . Uma vez que a função  $g_a(R)$  mede a similaridade levando em consideração um aspecto, e deseja-se obter a dissimilaridade, que é a métrica oposta, subtrai-se a média das similaridades de 1. Seja também

$$affinity(R, Y) = \frac{1}{|Y|} \sum_{a \in Y} h_a(R, I_u) \quad (2.5)$$

a similaridade média entre  $R_u$  e  $I_u$  para todos os aspectos em  $Y$ . A similaridade aferida pela função *affinity* executa o papel, no contexto desse trabalho, de métrica de relevância, já que ela captura a concordância entre os itens recomendados e os itens no histórico do usuário em relação aos aspectos  $Y$ . Parte-se, para isso, da consideração de que o histórico do usuário é composto, em sua quase totalidade, por itens efetivamente relevantes, uma vez que o usuário os consumiu.

Assim, dados os elementos de entrada citados, deseja-se encontrar um conjunto  $R \setminus I_u \subseteq I$  de  $n$  itens (ou seja  $|R| = n$ ) que maximize, ao mesmo tempo, os objetivos definidos nas equações 2.4 e 2.5, o que pode ser resumido na equação 2.6, a seguir:

$$\arg \max_R (diversity(R, X), affinity(R, Y)) \quad (2.6)$$

## **2.4 Considerações Finais**

Este capítulo apresentou uma breve contextualização acerca de sistemas de recomendação e de sua evolução, desde o momento em que apenas a acurácia do sistema era considerada importante até a percepção da importância da diversidade e da novidade enquanto objetivos a serem buscados na recomendação. Foi apresentado também o referencial teórico necessário para a abordagem multiobjetivo de recomendações e, por fim, foi formalmente descrito o problema a ser tratado neste trabalho.

# Capítulo 3

## Modelos Base

Neste capítulo, são apresentados em detalhes os modelos que foram utilizados como base para desenvolver os recomendadores propostos nesta pesquisa. Entre as abordagens usadas para o tratamento desse tipo de problema, algoritmos evolucionários têm aparecido como alternativas eficientes (CUI et al., 2016; RIBEIRO et al., 2015; WANG et al., 2016; ZUO et al., 2015), bem como abordagens usando MAB (MARY; GAUDEL; PREUX, 2015; LI; KARATZOGLOU; GENTILE, 2016; LACERDA, 2017). Essas abordagens serão discutidas na sequência deste capítulo.

### 3.1 Algoritmos Evolucionários para Otimização Multiobjetivo

Algoritmos Evolucionários de objetivo único são heurísticas de otimização inspiradas na teoria da evolução de Charles Darwin, particularmente no princípio de sobrevivência do mais apto (BRANKE, 2016). Neste tipo de algoritmo, uma população inicial de indivíduos candidatos é gerada aleatoriamente e sofre mutações (mudanças aleatórias) e *crossing over* (troca de elementos entre indivíduos), gerando uma população descendente. Ao longo de gerações (iterações), busca-se obter indivíduos mais adaptados, ou seja, que otimizem o valor da função objetivo. Dessa forma, a população evolui, aproximando-se do valor ótimo da função objetivo.

A utilização de Algoritmos Evolucionários requer apenas a definição do espaço de busca,

a descrição de uma solução e a definição de uma função objetivo para avaliação da qualidade (adaptação ao ambiente) de uma solução. A função objetivo não precisa ser contínua ou diferenciável. Ao trabalhar, não com uma solução única que é evoluída para uma solução melhor, mas com um conjunto de soluções, Algoritmos Evolucionários evitam a ocorrência de mínimos locais. Com essas características, Algoritmos Evolucionários apresentam-se como uma ferramenta útil em domínios em que métodos de otimização exata não são aplicáveis (BRANKE, 2016).

Algoritmos Evolucionários podem ser estendidos para lidar com problemas multiobjetivo. Tais adaptações são chamadas de MOEA. O objetivo de um MOEA é evoluir sua população inicial no sentido de se aproximar da fronteira de Pareto. O que muda de um Algoritmo Evolucionário para um MOEA é a forma como são ordenados e selecionados os indivíduos para a próxima geração, uma vez que havendo mais de um objetivo, a seleção dos indivíduos mais aptos não é tão óbvia. Para este fim, deve-se levar em consideração a proximidade da fronteira de Pareto e a distribuição das soluções ao longo da fronteira.

### 3.1.1 NSGA-II

NSGA II é um MOEA proposto por Deb et al. (2002) como uma evolução do algoritmo *Non-dominated Sorting Genetic Algorithm* (SRINIVAS; DEB, 1994). Apesar da existência de vários MOEAs, como por exemplo SPEA2 (Strength Pareto Evolutionary Algorithm II) (ZITZLER; LAUMANN; THIELE, 2001) e o MOEA/D (Multiobjective Evolutionary Algorithm based on Decomposition) (ZHANG; LI, 2007), o NSGA II tem sido amplamente utilizado para a solução de problemas multiobjetivo (ZUO et al., 2015; OUNI et al., 2017). Há uma série de razões para o NSGA II obter resultados superiores a algoritmos concorrentes.

A cada iteração do algoritmo, um conjunto de possíveis soluções de tamanho  $N$ , chamado de população progenitora, dá origem a uma população descendente, também de tamanho  $N$ . Dentro desta população de tamanho  $2N$ , é utilizada uma abordagem elitista, o que faz com que os  $N$  indivíduos mais adaptados sejam mantidos para a geração seguinte, independente de serem da população progenitora ou da população descendente. Provou-se que o elitismo acelera a convergências de MOEAs para a fronteira de Pareto (ZITZLER; DEB; THIELE, 2000).



A seleção dos elementos para a iteração seguinte é o principal procedimento de um MOEA. No NSGA II, essa seleção é feita em duas etapas, o *Fast Non-dominated Sorting* e o *Crowding Distance Sorting*.

### Fast Non-dominated Sorting

O procedimento *Fast Non-dominated Sorting* consiste em separar as soluções candidatas em níveis de dominância. O conjunto de soluções da população que não são dominadas por nenhuma outra solução dentro da população formam a primeira faixa de dominância; o conjunto das soluções que são dominadas por soluções da primeira faixa, mas não são dominadas por nenhuma outra formam a segunda faixa; e assim sucessivamente. Formalmente, o conjunto  $F$  de soluções ( $|F| = 2N$ ) é dividido em  $k$  subgrupos disjuntos  $F = \{F_1 \cup F_2 \cup \dots \cup F_k\}$ , tal que  $\forall x \in F_s$  e  $\forall y \in F_t$ ,  $s < t \rightarrow x \gg^* y$ , onde  $\gg^*$  é o símbolo da dominância.

Num procedimento ineficiente, para determinar as faixas de dominância cada solução deve ser comparada a todas as outras soluções para determinar dominância, o que se repete na determinação de cada faixa de dominância, resultando numa complexidade de  $\mathcal{O}(MN^3)$ , onde  $M$  é o número de objetivos e  $N$  é o tamanho da população. O *Fast Non-dominated Sorting* é um procedimento otimizado, em que para cada solução  $p$  são mantidos um contador  $n_p$  de soluções que dominam  $p$  e um conjunto  $S_p$  de soluções dominadas por  $p$ . O cálculo destes elementos tem complexidade  $\mathcal{O}(MN^2)$ .

A primeira faixa de dominância é então formada pelas soluções que possuem  $n_p = 0$ . Para cada uma destas soluções, visita-se seu conjunto  $S_p$  e reduz-se seu valor  $n_p$  em 1. Assim, as soluções que após esse procedimento tiverem  $n_p = 0$  formam a próxima faixa de dominância. Estes passos se repetem até que todos os elementos façam parte de alguma faixa de dominância  $F_k$ . A figura 3.2 mostra a tarefa do *Non-dominated Sorting* de divisão em faixas de dominância. Todo o procedimento tem complexidade  $\mathcal{O}(MN^2)$ .

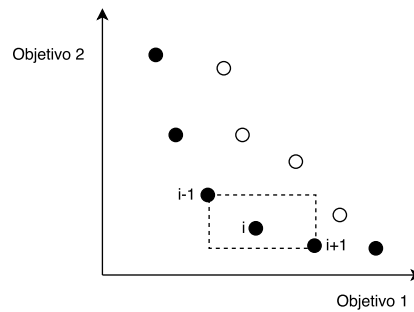
### Crowding Distance Sorting

Apois a geração da população descendente, tem-se uma população de  $2N$  soluções, da qual  $N$  soluções serão selecionadas para compor a população da próxima geração. As soluções são selecionadas pela faixa de dominância, acrescentando-se à próxima população todos os elementos da próxima faixa até que  $N$  elementos sejam selecionados. No entanto, a adição

de uma faixa de dominância pode resultar em um total de indivíduos maior que  $N$ . Nesse caso, deve haver uma seleção dentro da última faixa de dominância, apenas da quantidade de soluções necessárias para completar a população. Essa seleção é feita usando-se *Crowding Distance Sorting*.

Esta seleção é feita tentando-se preservar para a próxima população indivíduos distribuídos ao longo da fronteira de Pareto, de forma a preservar a diversidade de soluções, ao invés de concentrar-se em soluções cujos objetivos sejam valores próximos. O *Crowding Distance Sorting* estabelece uma métrica de estimativa de densidade chamada *Crowding Distance*, que mede o perímetro do cubóide formado quando são utilizados como vértices os objetivos das soluções mais próximas no espaço de objetivos. A figura 3.1 ilustra a formação do cubóide levando-se em consideração dois objetivos, porém a ideia pode ser estendida para qualquer número de objetivos.

Figura 3.1: Calculando o crowding distance para a  $i$ -ésima solução em sua faixa de dominância.



Fonte: Adaptado de Deb et al. (2002).

O NSGA II utiliza um valor chamado  $i_{distance}$  para aproximar a *crowding distance*. Para seu cálculo, as soluções dentro da faixa de dominância a ser avaliada são ordenadas de maneira crescente para cada objetivo e as distâncias entre as soluções são normalizadas de acordo com os valores máximo e mínimo daquele objetivo. A  $i_{distance}$  entre duas soluções é então a soma das distâncias individuais de cada objetivo. Dessa forma, duas soluções com  $i_{distance}$  pequeno são tidas como ocupando um espaço mais povoado, com o algoritmo tentando a selecionar elementos mais distantes entre si. A figura 3.2 ilustra uma situação em que o *Crowding Distance Sorting* é utilizado.

### Algoritmo

O algoritmo NSGA II é inicializado gerando randomicamente uma população inicial de soluções  $P_0$  de tamanho  $N$ . Esta população gera uma população descendente  $Q_0$ , também de tamanho  $N$ , através de *crossing over*, onde partes das soluções são permutadas, e mutações, onde há substituição de partes das soluções por itens aleatórios. A população  $R_0 = P_0 \cup S_0$ , de tamanho  $2N$ , é formada, de maneira a introduzir o elitismo no algoritmo, e é submetida ao procedimento *Fast Non-dominated Sorting*, sendo dividida em  $k$  faixas de dominância.

Quadro 3.1: Pseudocódigo do algoritmo NSGA II.

---

```

01: procedure NSGA-II ( $N, g$ )
02:    $P_0 = \text{criar\_população\_inicial}(N)$ 
03:   for  $t$  in 0 to  $g - 1$  do
04:      $Q_t = \text{criar\_nova\_geração}(P_t)$ 
05:      $R_t = P_t \cup Q_t$ 
06:      $F = \text{fast\_nondominated\_sort}(R_t)$ 
07:     while  $|P_t + 1| + |F_i| \leq N$  do
08:        $\text{crowding\_distance\_assignment}(F_i)$ 
09:        $P_{t+1} = P_{t+1} \cup F_i$ 
10:        $i = i + 1$ 
11:     end while
12:      $\text{ordenar}(F_i, \prec)$ 
13:      $P_{t+1} = P_{t+1} \cup F_i[N - P_{t+1}]$ 
14:   end for
15: end procedure

```

---

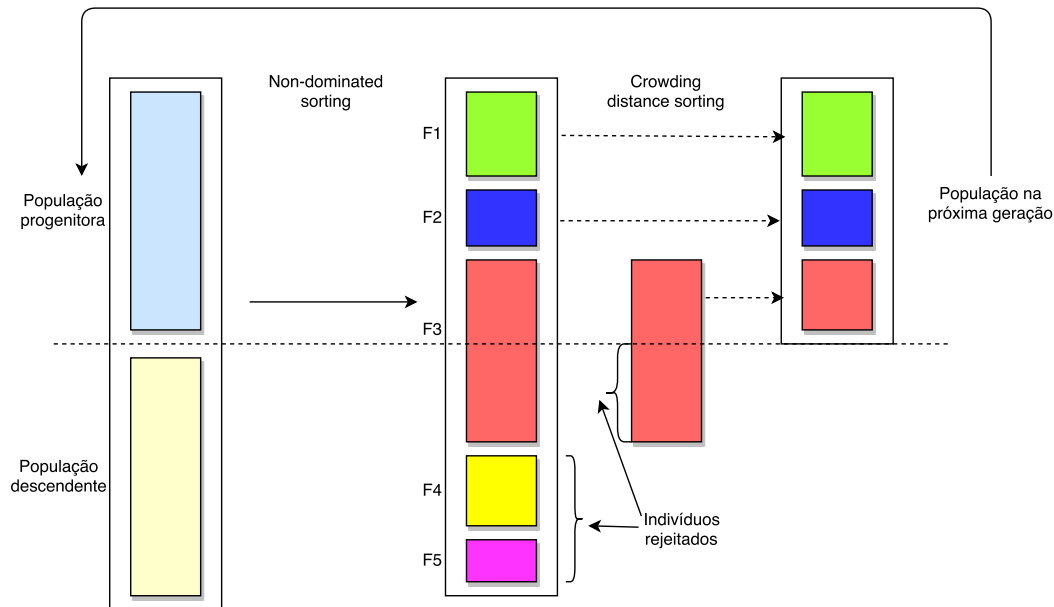
Fonte: adaptado de Deb et al. (2002).

Passam então a ser selecionadas as soluções para a população da próxima iteração. Os indivíduos nas faixas de dominância vão sendo adicionados até que a adição de uma faixa de dominância  $F_i$  extrapole o tamanho  $N$  da população. Então, o procedimento *Crowding Distance Sorting* é utilizado para selecionar as soluções necessárias para completar a população, privilegiando as soluções que diversifiquem melhor os valores de objetivos, fazendo com que as soluções selecionadas estejam preferencialmente espalhadas pela fronteira de Pareto.

O procedimento é então repetido ao longo de  $g$  gerações. Cada geração inicia com a população  $P_t$ , recebida da geração anterior, gerando uma nova população descendente  $S_t$ ,  $R_t = P_t \cup S_t$  e repetindo os passos descritos para a população  $R_0$ . Ao final das  $g$  gerações, tem-se uma população de  $N$  soluções representativa da fronteira de Pareto. Informações

adicionais são necessárias no sentido de determinar, dentre as  $N$  soluções, qual a solução selecionada como solução final. O quadro 3.1 demonstra o pseudocódigo do algoritmo NSGA II, enquanto a figura 3.2 ilustra o *loop* principal do algoritmo.

Figura 3.2: Passo principal do NSGA II.



Fonte: Adaptado de Deb et al. (2002).

## 3.2 Multi-Armed Bandits

MABs são uma política de Aprendizado por Reforço, modelada de acordo com o Processo de Decisão de Markov (MDP - *Markov Decision Process*), onde um agente tenta tomar decisões que maximizem os resultados de suas ações (LACERDA, 2017). São uma política geralmente usada como uma solução para o dilema *Exploration x Exploitation*, descrito a seguir.

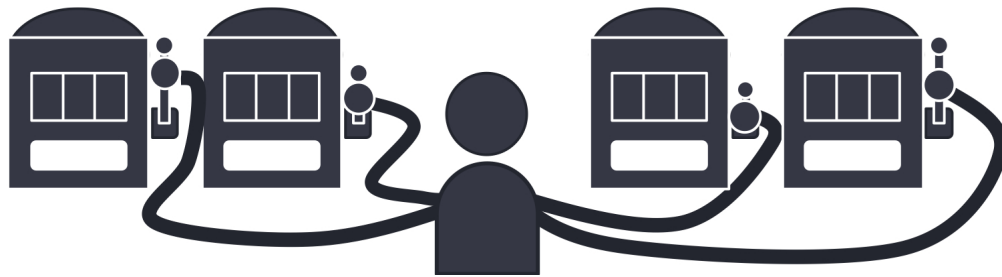
### 3.2.1 Exploration x Exploitation Dilemma

Políticas de Aprendizado por Reforço comumente deparam-se com uma questão denominada *Exploration x Exploitation Dilemma* (mantido em inglês, uma vez que os termos *exploration* e *exploitation* são traduzidos para o português usando mesma palavra: exploração). Trata-se da decisão, a ser tomada pelo algoritmo, entre usar a melhor solução conhecida para

obter ganhos (*exploit*) ou escolher uma solução cujo retorno é desconhecido, na esperança de encontrar ganhos ainda maiores (*explore*).

MABs são uma analogia para a solução deste problema, baseada em máquinas caça-níqueis, também chamadas de *one-armed bandit* (bandido de um braço). A operação dessas máquinas consiste em puxar o braço e esperar por uma possível recompensa. *Multi-Armed Bandits*, ou bandido de múltiplos braços, são uma modelagem proposta por Lai e Robbins (1984) e revisitada por Sutton, Barto et al. (1998), na qual uma máquina possui múltiplos braços, onde a distribuição de recompensas é desconhecida e possivelmente diferente para cada braço (Figura 3.3). O problema, então, é encontrar o braço que entrega a maior recompensa, após uma sequência de tentativas.

Figura 3.3: Multi-Armed Bandit.



Fonte: (CHAKRAVARTHI, 2019).

A maneira clássica de lidar com tomadas de decisão com efeitos desconhecidos é a realização de múltiplos testes A/B (MCDONALD, 2018). Neste caso, a execução dos testes seria puxar todos os braços, uma certa quantidade de vezes e, dessa forma, mapear a distribuição mais rentável através de testes estatísticos. Essa solução, apesar de efetiva, possui um problema bastante claro: o custo de se puxar todos os braços. Para evitar esse custo, é preciso balancear as tentativas entre *exploit* e *explore*.

Em MABs, na maior parte das vezes, o algoritmo irá escolher a solução tida como melhor solução, proporcionando aumento da recompensa obtida. Ao escolher um braço  $b_t \in B$ , uma recompensa  $r_{t,b}$  é obtida, enquanto nenhuma informação sobre recompensas nos outros braços é obtida (LI et al., 2010). A única forma de conhecer a recompensa fornecida por outro braço é puxá-lo, em uma tentativa subsequente. Por isso, eventualmente, um braço diferente daquele que é tido como melhor, é puxado e tem sua recompensa observada, podendo ser

então considerado o melhor braço. Dessa forma, o algoritmo tenta, após um certo número de iterações, aproximar-se da solução ótima, diminuindo o *regret* (arrependimento) médio. Por *regret* entende-se a diferença entre a recompensa obtida ao puxar um braço e a recompensa ótima.

Formalmente, considerando  $B = \{1, 2, \dots, M\}$  como o conjunto de  $M$  braços em um MAB e  $\vec{r} \in \{0, 1\}^M$  sendo o vetor de recompensas, tem-se o algoritmo 3.2, onde aparecem as rotinas *obtain-reward* e *positive-reward*, que dependem do contexto em que o MAB está sendo empregado. No contexto de recomendações, por exemplo, *obtain-reward* pode ser a exibição da recomendação ao usuário, enquanto *positive-reward* seria um clique do usuário na recomendação. Neste caso, trata-se de um MAB binário, ou seja, a escolha do braço implica em obter a recompensa ou não, porém, MABs podem ser generalizados para recompensas não binárias.

Quadro 3.2: Pseudocódigo do MAB.

Multi-Armed Bandit	
01:	<i>Initialize</i> $B(M)$
02:	<b>for</b> $t = 1$ to $T$ <b>do</b>
03:	$b(t) = \text{select-arm}(B)$
04:	<i>collect-reward</i> for $b(t)$
05:	<b>if</b> <i>positive-reward</i> <b>then</b>
06:	$r_t = 1$
07:	<b>else</b>
08:	$r_t = 0$
09:	<b>end if</b>
10:	<i>update-mab</i> ( $b(t), r_t, B$ )
11:	<b>end for</b>

Fonte: adaptado de Lacerda (2017).

A eficiência de um algoritmo em minimizar o *regret*, ou seja, aumentar a sua eficiência, está ligada a sua política de seleção de ações. Algumas das principais políticas em MAB serão discutidas na sequência.

### $\epsilon$ -greedy

Na política  $\epsilon$ -greedy, o parâmetro  $\epsilon$  determina a chance do algoritmo escolher aleatoriamente um braço, ou seja, de buscar uma alternativa à melhor opção conhecida (*explore*), enquanto a probabilidade de escolher a melhor opção conhecida (*exploit*) é de  $1 - \epsilon$ . Cada vez que

um braço é puxado, atualiza-se o valor de sua recompensa média, de forma que se uma nova melhor opção for encontrada, ela passara a ser o braço escolhido com probabilidade  $1 - \epsilon$  (SUTTON; BARTO et al., 1998).

Na política  $\epsilon$ -greedy, um maior valor de  $\epsilon$  realiza um maior número de explorações, permitindo que se encontre rapidamente a solução ótima, ou próxima disso. Entretanto, a continuidade da realização de buscas faz com que, um percentual  $\epsilon$  de vezes, se deixe de realizar a melhor ação. Essa penalização, entretanto, pode ser mitigada com a utilização de um valor decrescente para  $\epsilon$ , podendo chegar a zero, o que garante o uso da melhor opção todas as vezes, com o valor de *regret* tendendo a zero.

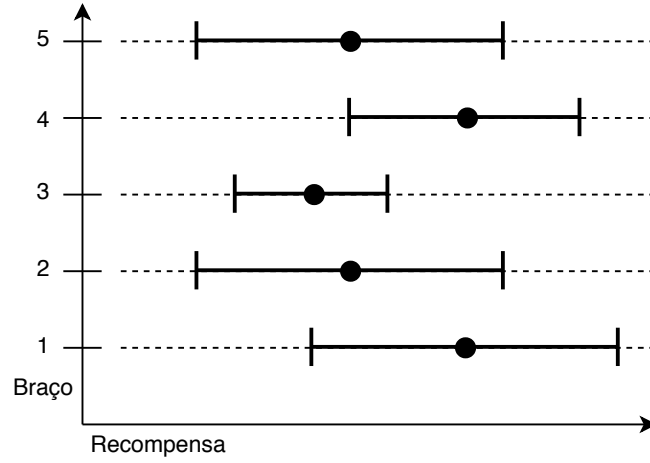
## UCB

A escolha de ações não ótimas é necessária em se tratando do dilema *Exploration x Exploitation*, devido à incerteza das recompensas. Entretanto, a política  $\epsilon$ -greedy o faz de maneira aleatória, sem realizar discriminação entre soluções potencialmente ótimas ou particularmente incertas (SUTTON; BARTO et al., 1998). A política *Upper Confidence Bound* (UCB) se baseia no intervalo de confiança gerado a partir da distribuição de recompensas já obtidas por um determinado braço. Com base no valor superior do intervalo de confiança, o UCB determina soluções promissoras, uma vez que recompensas prévias posicionaram o intervalo de confiança num valor mais elevado.

À medida que uma mesma solução é escolhida mais vezes, o intervalo de confiança encolhe, fazendo com que novas soluções tenham o valor superior do intervalo de confiança mais elevado, o que gera a exploração necessária de novas soluções. A Figura 3.4 ilustra uma situação em que as recompensas médias de cinco soluções estão mapeadas, junto com seus intervalos de confiança. Na situação colocada na figura, o braço 1, com o maior limite superior de intervalo de confiança, seria o escolhido. Após uma série de seleções, seu intervalo de confiança tende a encolher, fazendo com que o braço 4, o mais promissor entre os restantes, viesse a ser escolhido.

Formalmente, o braço considerado a melhor opção,  $B_n$ , na iteração  $n$  é escolhido com base na Equação 3.1

Figura 3.4: Política de Seleção de Ações Upper Confidence Bound.



Fonte: Autoria Própria.

$$B_t = \arg \max_b \left( Q_t(b) + c \sqrt{\frac{\log(t)}{k_t(b)}} \right), \quad (3.1)$$

onde  $Q_t(b)$  é a estimativa atual de recompensa para o braço  $b$ ,  $k_t(b)$  é o número de vezes que o braço  $b$  foi escolhido, e a constante  $c$  controla o grau de exploração.

### 3.2.2 Ranked Bandits

*Ranked Bandits* são uma forma de modelar MABs para o caso de se tentar encontrar a melhor solução, mas uma lista de soluções que, combinadas, produzam melhores índices. É o que ocorre em alguns cenários de recomendação, quando a geração de recompensa se dá não para um item recomendado, mas para uma lista de itens, como é o caso de quando se tenta produzir uma lista de recomendações diversificadas entre si. *Ranked Bandits* são particularmente adequados para a tarefa de diversificação de recomendações, uma vez que a diversidade de uma recomendação precisa da lista completa de itens recomendados para ser medida (RADLINSKI; KLEINBERG; JOACHIMS, 2008).

Assim, o modelo de Ranked Bandits para sistemas de recomendação é composto de tantas máquinas quantos forem os itens que se deseja recomendar, isto é, uma instância  $MAB_i$  para cada posição  $i$  da lista de recomendação, funcionando de maneira independente dos demais. Dessa forma, o algoritmo  $MAB_1$  é responsável por selecionar o item que será recomendado



na posição 1 da lista. Na sequência, o algoritmo  $MAB_2$  seleciona o item para a posição 2, a não ser que esse item já pertença à lista de recomendação. Nesse caso, um outro item é escolhido arbitrariamente para compor a lista naquela posição. Este processo é repetido para todos os  $k$  itens da lista.

Quadro 3.3: Pseudocódigo de Ranked Bandits.

Ranked Bandits	
01:	<i>Initialize</i> $B_1(M), \dots, B_k(M)$
02:	<b>for</b> $t = 1$ to $T$ <b>do</b>
03:	<b>for</b> $j = 1$ to $k$ <b>do</b>
04:	$\hat{b}_j(t) = \text{select-arm}(u_t, B_j)$
05:	<b>if</b> $j > 1$ <b>then</b>
06:	<b>if</b> $\hat{b}_j(t) \in \{b_1(t), \dots, b_{j-1}(t)\}$ <b>then</b>
07:	$b_j(t) = \text{random item}$
08:	<b>else</b>
09:	$b_j(t) = \hat{b}_j(t)$
10:	<b>end if</b>
11:	<b>end if</b>
12:	<b>end for</b>
13:	$r_t = \text{collect reward for } b_1(t), \dots, b_k(t)$
14:	<b>for</b> $j = 1, \dots, k$ <b>do</b>
15:	$\text{update-mab}(u_t, b_j(t), r_t, B_j)$
16:	<b>end for</b>
17:	<b>end for</b>

Fonte: adaptado de Radlinski, Kleinberg e Joachims (2008).

O Quadro 3.3 mostra o pseudocódigo para o algoritmo *Ranked Bandits*. Inicialmente,  $k$  máquinas são inicializadas. Então, um laço de  $T$  iterações é realizado (linha 02), onde em cada iteração cada uma das máquinas seleciona um item, de acordo com sua política de seleção de ações (linha 04). Os itens selecionados são adicionados na lista de recomendação, ou substituídos por um item aleatório, caso estejam previamente na lista (linha 06). Com a lista formada, obtêm-se a recompensa (linha 13), que é utilizada para atualização de cada máquina, novamente de acordo com sua política de seleção de ações (linha 15).

### 3.2.3 MAB para Recomendação Multiobjetivo

Uma maneira de tratar problemas em um cenário multiobjetivo é transformar os múltiplos objetivos em um objetivo único, através da escalarização de funções (EICHFELDER, 2009).

No caso de MAB, em um cenário multiobjetivo, cada braço  $b$  tem um vetor de saída,  $\vec{s}_b = \{s_b^1, \dots, s_b^O\}$ , onde  $O$  é um número fixo de objetivos. Uma função de escalarização é usada para computar um valor único a partir do vetor  $\vec{s}_b$ . Duas funções de escalarização foram consideradas neste trabalho.

### Escalarização Linear

Considerando se um vetor de pesos  $\vec{w} = (w^1, \dots, w^O)$ , onde  $\sum_{o=1}^O w^o = 1$ , cada componente  $s_b^o$  do vetor de saída de um braço  $b$  tem atribuído a si um peso  $w^o$ , retornando um valor escalar correspondente a sua média ponderada.

$$f_{\vec{w}}(\vec{s}_b) = \sum_{o=1}^O w^o s_b^o \quad (3.2)$$

### Escalarização Chebyshev

A função de Chebyshev usa um ponto de referência  $O$ -dimensional no espaço de objetivos, em um instante  $T$ ,  $z = (z^1, \dots, z^O)$ . Este ponto deve ser dominado pelo vetor de objetivos  $\vec{s}_b$ . A Equação 3.3 mostra a escalarização de Chebyshev (KASIMBEYLI et al., 2019).

$$f_{\vec{w}}(\vec{s}_b) = \min_{1 \leq j \leq O} w^j \times (s_j^b - z^j), \forall j \quad (3.3)$$

O ponto  $z$  corresponde ao menor objetivo dos vetor de recompensa ótimos em cada braço, subtraído de um valor  $\tau$  positivo constante (por exemplo,  $\tau = 0,001$ ). Considerando cada objetivo  $l$ , tem-se a Equação 3.4.

$$z^l = \min_{1 \leq j \leq O} s_j^l \quad (3.4)$$

## 3.2.4 Considerações Finais

Neste capítulo foram abordados os dois modelos de otimização para recomendações multiobjetivo usados neste trabalho: MOEAs e MABs. Ambos foram discutidos de maneira geral, uma vez que cada um integrou uma solução proposta para o problema da recomendação musical que simultaneamente busca maximizar diversidade e precisão. Nos capítulos

---

subsequentes, serão mostradas as soluções envolvendo estes modelos, seus resultados e avaliações.

## Capítulo 4

# Trabalhos Relacionados

Este capítulo apresenta uma seleção de trabalhos representativa do estado da arte no que se refere aos temas de contribuição deste trabalho. Tratar-se de trabalhos que contenham abordagens de diversidade em sistemas de recomendação, especificamente aqueles cujas soluções tentem aprimorar a diversidade nas recomendações oferecidas diminuindo o prejuízo da precisão do sistema. Mantendo a relação com este trabalho, serão apresentadas soluções multiobjetivo, notadamente as que utilizem algoritmos evolucionários e MAB. Por fim, são mostrados trabalhos que se relacionam com este de maneira geral.

### 4.1 Avaliação Além da Acurácia

Diversos trabalhos foram publicados em anos recentes propondo sistemas de recomendação preocupados em investigar outras métricas, além da acurácia. A ideia por trás de métricas como diversidade, novidade ou serendipidade é aumentar a satisfação do usuário com as recomendações, não apenas indicando itens que atendam ao seu gosto pessoal, mas trazendo itens que o usuário provavelmente não encontraria sozinho, dados os itens que o usuário tem consumido.

Ao oferecer recomendações diversas, tem-se a ideia de ampliar a gama de opções exibidas ao usuário, aumentando assim a chance de que alguma daquelas possibilidades adéque-se ao interesse momentâneo do usuário. Com a métrica novidade, tem-se a intenção de oferecer itens desconhecidos, porém próximos às preferências do usuário, enquanto serendipidade diz respeito à sensação de descoberta de um item valioso. Em todos os casos, as recomendações

realizadas devem ser de interesse do usuário, ou seja, o sistema de recomendação deve buscar essas métricas com o menor prejuízo possível à métrica de precisão do sistema.

O trabalho seminal de Ziegler et al. (2005) gerou os alicerces para uma série de trabalhos com foco em diversificação de sistemas de recomendação. Seu algoritmo, o *Topic Diversification*, baseia-se nos resultados do algoritmo de filtragem colaborativa, realizando um re-ranqueamento usando itens em posições que originalmente ficariam de fora da lista de recomendação, em uma abordagem gulosa, para obter uma lista diversificada. Dessa forma, a precisão estaria assegurada pelo fato de todas as recomendações virem de usuários semelhantes, por conta da filtragem colaborativa.

O MMR foi uma adaptação feita por Vargas e Castells (2011) do algoritmo *Topic Diversification* (TD) de Ziegler et al. (2005). Nesta adaptação, Vargas e Castells (2011) citam a semelhança do TD com o MMR usado em Recuperação da Informação, proposto por Carbonell e Goldstein (1998), e preferem manter este nome em sua adaptação.

Por ser um algoritmo de rerranqueamento, o TD precisa receber como entrada uma lista de recomendação com uma quantidade de itens substancialmente maior que o tamanho da lista de recomendação final, como por exemplo, uma lista de entrada de 50 ou 100 itens para gerar uma lista de saída de 10 itens. Isso porque, dentre os itens fornecidos nessa lista serão recomendados aqueles que proporcionarem a maior diversidade da lista final. Como a operação do algoritmo visa obter a maior diversidade possível, a manutenção da relevância da lista é feita com base em duas estratégias: (i) a lista fornecida como entrada é obtida com uso de um algoritmo com foco em relevância, como por exemplo a Filtragem Colaborativa e (ii) o primeiro item da lista de entrada é automaticamente selecionado como primeiro item da lista de saída.

A partir daí, a cada iteração, usando uma métrica de similaridade como distância do cosseno ou correlação de Pearson, é selecionado um item para compor a lista de recomendação de saída. Por se tratar de uma abordagem gulosa, é selecionado o item que possui a maior distância média para os itens que já constem da lista de recomendação final. Este procedimento se repete até que se complete a lista de saída.

Na adaptação proposta por (VARGAS; CASTELLS, 2011), cada item candidato a compor a lista de saída é avaliado por uma função objetivo que captura acurácia e diversidade. É feita uma combinação linear entre a avaliação (*rating*) predita pelo algoritmo *baseline* –

o que foi usado para gerar a lista de entrada – e a dissimilaridade em relação aos itens já posicionados na lista de saída – da mesma forma que o TD. Para a combinação linear, o MMR usa pesos iguais ( $\lambda = 0,5$ ). Com uso desta função objetivo, a lista de recomendação é preenchida da mesma forma que o faz o algoritmo TD.

O RxQuAD é também uma adaptação proposta por Vargas, Castells e Vallet (2012), do algoritmo *eXplicit Query Aspect Diversification* (xQuAD) (SANTOS; MACDONALD; OUNIS, 2010). O xQuAD é um algoritmo probabilístico de Recuperação da Informação, tendo sido originalmente desenvolvido para a diversificação de resultados de consultas realizadas em motores de busca. O algoritmo foi desenvolvido levando em consideração que uma consulta pode ser interpretada de diversas maneiras diferentes, cada uma dessas maneiras sendo considerada um aspecto da consulta. A ideia é realizar um reranqueamento do resultado original da consulta de modo a diversificar o resultado de acordo com os diferentes aspectos da consulta, otimizando a cobertura (*coverage*) dos aspectos da consulta e aumentando a chance de acertar nos primeiros itens retornados o aspecto que o usuário que emitiu a consulta esperava obter como resposta.

Para tanto, a consulta realizada é dividida em subconsultas, em que cada subconsulta procura atender a um aspecto diferente. O algoritmo utiliza a probabilidade do item  $d$  ser observado para a consulta  $q$ ,  $P(d|q)$ , somada à probabilidade de observar-se o item  $d$  ao invés dos itens já presentes na saída  $\bar{S}$ ,  $P(d, \bar{S}|q)$ , ambas as probabilidades tendo um *tradeoff* controlado pelo parâmetro  $\lambda$ :  $(1 - \lambda)P(d|q) + \lambda P(d, \bar{S}|q)$ . As probabilidades são calculadas com base no *feedback* implícito da comunidade para cada item.

No RxQuAD, a função objetivo em termos de probabilidade do item atender à consulta foi substituída por uma função objetivo de relevância  $g(d|S, q) = (1 - \lambda)P(r_d|q) + \lambda P(r_d, \neg r_S|q)$ , onde  $r_d$  é a relevância do item  $d$  para a consulta e  $\neg r_S$  representa o fato de não haver nenhum item relevante em  $S$ .

Para adaptar o RxQuAD à tarefa de recomendação, o histórico dos usuários é utilizado como consulta, tendo sido utilizados nos experimentos os dados do MovieLens <sup>1</sup> e do Last.fm. No experimento usando o MovieLens o gênero do filme foi usado como aspecto de diversificação da consulta, enquanto no Last.fm as tags de usuário foram usadas como aspecto.

---

<sup>1</sup><https://movielens.org/>

Diversos outros trabalhos foram publicados seguindo princípios semelhantes, mas baseados em técnicas diferentes, tais como aprendizado de máquina (ZHU et al., 2014; KAPOOR et al., 2015) e recuperação da informação (VARGAS; CASTELLS; VALLET, 2012).

Neste trabalho, aborda-se especificamente o desenvolvimento de sistemas de recomendação capazes de realizar recomendações diversificadas, sem que haja degradação na relevância dos itens recomendados. O MMR e o RxQuAD foram usados como *baselines* na avaliação dos resultados dos experimentos realizados.

## 4.2 Recomendações Multiobjetivo

Tendo em consideração o fato de que, ao explorar novas métricas em sistemas de recomendação, deve-se buscar manter o nível de acurácia do sistema, ou ao menos afetá-lo o mínimo possível, diversos trabalhos consideram tal tarefa como sendo naturalmente um problema multiobjetivo.

Ribeiro et al. (2012) desenvolveram uma técnica para sistemas de recomendação que busca atender a múltiplos objetivos, no caso, acurácia, novidade e diversidade. Os autores consideram que estes três objetivos podem ser conflitantes, então tratam o problema como sendo multiobjetivo. Esta técnica usa uma abordagem híbrida e os múltiplos algoritmos são selecionados com o uso de algoritmos evolucionários multiobjetivo. O problema reside no fato de que não é possível maximizar múltiplos objetivos conflitantes ao mesmo tempo, então a fronteira de Pareto é usada para determinar soluções Pareto-ótimas. Neste trabalho, buscou-se maximizar um dos objetivos, sem que os demais fossem sacrificados. Como resultados, os autores mostram que o modelo híbrido consegue alto desempenho em algum objetivo, ao mesmo tempo em que se mantém competitivo nos demais objetivos.

Ouni et al. (2017) buscam recomendar bibliotecas de software baseando-se nas bibliotecas já em uso de um determinado projeto. As bibliotecas candidatas devem maximizar três objetivos conflitantes: maximizar a co-utilização entre a biblioteca candidata e as bibliotecas já em uso, maximizar a similaridade semântica entre o código da biblioteca candidata e o código do sistema, e minimizar o número total de bibliotecas recomendadas. Utiliza-se o algoritmo genético NSGA II para geração de indivíduos não dominantes entre si, candidatos a biblioteca recomendada.

Wang et al. (2016) sugerem que sistemas de recomendação focados em acurácia tendem a recomendar itens populares, o que, em sua perspectiva, pode ser realizado sem um sistema de recomendação. Os autores propõem uma abordagem para recomendação de cauda longa (*long tail recommendations*), cuja ideia é obter uma recomendação que mescle itens populares e impopulares, sem prejuízo da acurácia. Para tanto, foi novamente utilizado o NSGA II como MOEA.

Zuo et al. (2015) propuseram um sistema de recomendação que equilibra os objetivos de acurácia e diversidade. Neste trabalho, ProbS (ZHOU et al., 2010) foi utilizado como métrica de acurácia, enquanto *coverage* foi usado como métrica de diversidade. O NSGA II também foi usado como MOEA. Para cuidar do problema de escala, *clusters* de usuários foram criados usando k-means.

Caldeira et al. (2018) desenvolveram um sistema multiobjetivo para recomendar uma composição de receitas formando uma refeição, otimizando simultaneamente aspectos nutricionais e harmonia entre as receitas. Utilizaram para tanto o NSGA II, desenvolvendo funções de similaridade para cada um dos componentes nutricionais, de forma a adequar a refeição dentro da faixa de recomendação da Organização Mundial de Saúde (OMS).

Assim como os trabalhos citados nesta seção, este trabalho considera o desenvolvimento de um sistema de recomendação focado em diversidade, mas sem prejuízo da acurácia, como sendo um problema de otimização multiobjetivo. O diferencial apresentado é o fato de aplicar acurácia e diversidade a diferentes grupos de aspectos, além do uso de métricas de similaridade adequadas a cada aspecto analisado.

### 4.3 Recomendação usando Multi-Armed Bandits

A utilização de MABs na tarefa de recomendação é tema recorrente nas comunidades de sistemas de recomendação e aprendizagem de máquina. Depois do trabalho original de Lai e Robbins (1984), Peter Auer realizou grande avanço na área, propondo a política UCB (AUER, 2000), entre outras políticas e variações (AUER; CESA-BIANCHI; FISCHER, 2002; AUER, 2002).

MABs vêm sendo aplicados em uma série de cenários nos quais ocorre uma mudança constante na base de itens a serem recomendados, devido à rápida capacidade de reação do



modelo a essas mudanças, tornando-se uma técnica referência no que se refere a sistemas de recomendação adaptativos (GENTILE; LI; ZAPPELLA, 2014; LI; KARATZOGLOU; GENTILE, 2016). Li et al. (2010) utilizam *contextual bandits* para a recomendação de notícias, cenário em que acontecem frequentes inserções e remoções de itens, devido à constante produção de notícias, resultando em uma base bastante dinâmica. Em seu modelo, utilizam o número de cliques dos usuários como recompensa para o algoritmo, e  $\epsilon$ -greedy e UCB1 como políticas de seleção de ação. Pereira et al. (2019) propõem um sistema de recomendações sequenciais que utiliza o *feedback* instantâneo do usuário para aprendizado contínuo do recomendador. Os autores fazem uso de *Counterfactual Dueling Bandits* (CDB), modelo que se mostra adequado a cenários em que há *feedback* escasso, como a recomendação de músicas sequenciais.

Alguns trabalhos têm incorporado componentes sociais em algoritmos MAB (BUCCA-PATNAM; ERYILMAZ; SHROFF, 2013; CESA-BIANCHI; GENTILE; ZAPPELLA, 2013; GENTILE; LI; ZAPPELLA, 2014). Dessa forma, pode-se tirar vantagem dos relacionamentos entre usuários para previsão de interesses, como por exemplo, recomendação para grupos de usuários com base em seus relacionamentos e reações a conteúdos veiculados em redes sociais (CARON et al., 2012; VALKO et al., 2014). Outra tentativa de prever o interesse dos usuários e adaptar recomendações a estes interesses usando MABs foi feita por Hariri, Mobasher e Burke (2015). Assim como no trabalho corrente, o feedback do usuário às recomendações feitas é usado para determinar estas mudanças de interesse.

Outra utilização comum de MABs pela comunidade científica é também propósito deste trabalho: a recomendação multiobjetivo. Radlinski, Kleinberg e Joachims (2008) propuseram o uso de *Ranked Bandits* para a geração de recomendações com foco em diversidade, em uma abordagem de recomendação online, garantindo assintoticamente o desempenho do algoritmo mesmo que os interesses do usuário mudem. Lacerda (2017) fez uso de *Ranked Bandits* com foco em recomendações multiobjetivo. Em outra abordagem, Zhong (2016) propõe uma versão multiobjetivo de MAB para a tarefa de classificação. Neste trabalho, propomos a utilização de *Ranked Bandits* para a geração de recomendações multiobjetivo, levando em consideração múltiplos aspectos musicais dos itens e mudanças de interesse do usuário.

Mais recentemente, surgiram trabalhos abordando melhorias na função de recompensa

(DRAGONE; MEHROTRA; LALMAS, 2019) e abordando maneiras de avaliar sistemas de recomendação offline simulando *feedback* de usuários online através de seu histórico (SANZ-CRUZADO; CASTELLS; LÓPEZ, 2019; JEUNEN, 2019)

## 4.4 Outras Abordagens

Ao invés de estabelecer o problema de diversidade como otimização bicritério entre relevância e diversidade, Küçüktunç et al. (2013) propuseram uma métrica híbrida, a *expanded relevance*, que avaliaria algoritmos segundo ambos os critérios. A métrica é uma combinação entre *normalized relevance* e *l-expansion ratio*. Para medir a relevância de um item, eles usaram os pesos gerados pelo algoritmo Personalized PageRank (HAVELIWALA, 2002). A cobertura do espaço de dados foi usada como métrica de diversidade, assumindo que uma recomendação diversa tende a atingir uma maior parte dos dados disponíveis no espaço de dados que uma recomendação alheia à diversidade, considerando-se a *feature* analisada. A métrica *expanded relevance* consiste da soma da relevância de todos os itens alcançada no nível  $l$  de profundidade.

E, finalmente, Pampalk e Goto (2007) propuseram uma interface gráfica onde os usuários podem ajustar suas recomendações de acordo com seu interesse, através do ajuste de aspectos musicais. Três aspectos são utilizados para que o usuário possa ajustar sua preferência: similaridade sonora, similaridade sociocultural baseada em web, que usa *Term Frequency - Inverse Document Frequency (TF-IDF)* como métrica, e similaridade de palavras usadas para descrição do artista (por exemplo "pop" ou "brasileiro"). As recomendações são feitas com base na similaridade entre esses aspectos.

Este trabalho relaciona-se com os trabalhos citados na medida que são propostas métricas específicas para a avaliação de acurácia e diversidade, bem como para cálculo de similaridade entre diferentes aspectos, entretanto, levando em consideração a separação destes aspectos ao aplicar os múltiplos objetivos analisados.

## **4.5 Considerações Finais**

Neste capítulo foram apresentados os trabalhos que foram referência para o desenvolvimento desta tese, desde o surgimento de propostas com foco em outras métricas que não a acurácia, porém usando estratégias diversas para evitar a diminuição da relevância, até trabalhos com foco simultâneo, ou multiobjetivo, em diversidade e acurácia.

## Capítulo 5

# Sistemas de Recomendação Multiobjetivo Baseados em Aspectos

Este capítulo discorre sobre os modelos propostos como solução para o problema formalizado na seção 2.3. Serão discutidos na sequência os procedimentos tomados para o desenvolvimento da solução, bem como os aspectos selecionados e as métricas utilizadas.

### 5.1 Diversificação Multiobjetivo Baseada em Aspectos

Neste trabalho, propõe-se abordagens multiobjetivo para sistemas de recomendação nos quais a busca por recomendações diversas não implique em prejuízo da precisão do sistema. Especificamente, busca-se, dentre os metadados de artistas musicais a serem recomendados, identificar diversos aspectos musicais que possivelmente caracterizam as preferências musicais dos usuários. O uso destes metadados permite o direcionamento da diversificação e relevância das recomendações a partir de aspectos compreensíveis para o usuário, facilitando a configuração do sistema.

Foram utilizados os aspectos contemporaneidade, localidade e gênero musical, mas diversos outros aspectos podem ser trabalhados, tais como letras das músicas, popularidade, dançabilidade, etc. A escolha dos aspectos utilizados se deu pelo senso de que tais aspectos são normalmente utilizados pelo usuário para caracterizar suas preferências musicais. Enquanto o gênero musical divide usuários de gosto eclético de usuários fãs de gêneros específicos, há também usuários que preferem ouvir música de seu próprio país, enquanto

outros estão mais abertos a conhecer artistas do mundo inteiro; há usuários que demonstram preferência por músicas contemporâneas, enquanto outros preferem músicas de determinada época ou são indiferentes a este aspecto.

Cada aspecto proposto é composto por um ou mais atributos do conjunto de atributos dos metadados trabalhados. É desejável, do ponto de vista do usuário, que as recomendações levem em consideração que aspectos seriam interessantes de serem diversificados e que aspectos o sistema de recomendação deve buscar semelhança ao que é comumente escutado por si.

Na abordagem proposta, pode-se separar os aspectos em dois grupos e aplicar um objetivo específico a cada um. O fato de haverem então dois objetivos diferentes aplicados a grupos específicos de aspectos justifica a escolha da função multiobjetivo. Para o grupo de aspectos a ser diversificado, aplica-se o ILD, que representa a diversidade entre os elementos de uma lista de recomendação ao comparar os itens par a par. Para o grupo de aspectos a ser mantido, compara-se cada item da lista aos itens no histórico do usuário, de forma a manter a proximidade nos itens recomendados com os itens historicamente consumidos, no que tange ao grupo de aspectos assim determinado.

## 5.2 Representação de Itens e Métricas de Similaridade

Com o intuito de calcular as funções objetivo definidas nas equações 2.4 e 2.5, é necessário calcular similaridades entre itens, levando em consideração o conjunto de aspectos usado como entrada. Para tanto, são definidas funções extratoras para cada aspecto tal que medidas de similaridade possam ser aplicadas.

### 5.2.1 Definição dos Aspectos

Quatro aspectos referentes a artistas musicais foram extraídos dos metadados. Nesta seção, o significado de cada um será definido.

- **Contemporaneidade ( $A_1$ ):** refere-se ao tempo de coexistência da carreira dos artistas, em relação ao tempo total de carreira de cada um, levando em consideração as datas de lançamento de seus respectivos primeiro e último álbum;

- **Localidade** ( $A_2$ ): para este aspecto, a granularidade utilizada foi país. Refere-se à localidade com a qual o artista se identifica primariamente. Geralmente é o país de nascimento, no caso de artistas solo, ou país de formação, no caso de bandas.
- **Gênero musical** ( $A_3$ ): refere-se, como se pode supor, aos gêneros musicais atribuídos ao artista. Um artista pode estar associado a diversos gêneros musicais, sem limite de quantidade.

Tais aspectos foram selecionados devido a duas razões principais: (i) estes aspectos são normalmente usados em trabalhos relacionados (não necessariamente em conjunto) como informações adicionais para enriquecer modelos de preferências de usuários; e (ii) são dados disponíveis publicamente em bases de dados como MusicBrainz e DBpedia, que serão melhor detalhadas na seção 6.1.

### 5.2.2 Métricas de Similaridade

Conforme definido na seção 2.3, a função  $g_a(R)$  calcula a similaridade entre os elementos na lista de recomendação par a par, enquanto a função  $h_a(R, I_u)$  calcula a similaridade entre os itens na lista de recomendação e no histórico do usuário; ambas as funções o fazem tendo em vista um aspecto específico, sendo usada a média dos aspectos especificados para o cálculo de cada objetivo. É necessário então determinar a métrica de comparação par a par para cada aspecto individualmente. As seções a seguir tratam dessas métricas.

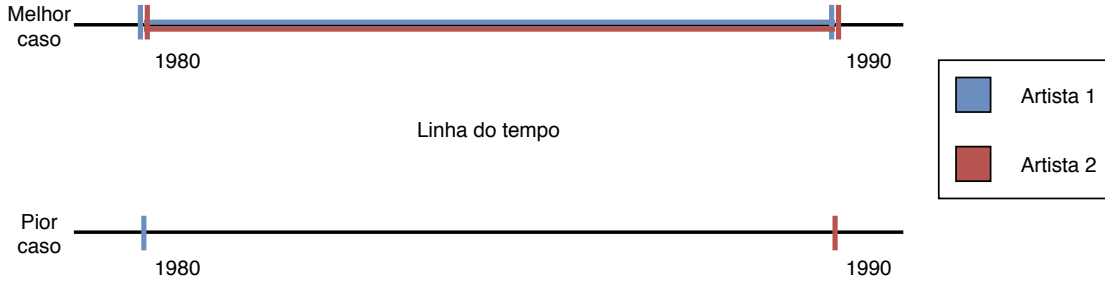
#### Contemporaneidade ( $A_1$ )

Cada item é representado por um vetor unidimensional de duas posições, sendo a primeira o ano de lançamento do primeiro álbum do artista, aqui chamado de  $a_1 \in A_{1_1}$ ; e a segunda o ano de lançamento de seu último álbum,  $a_2 \in A_{1_2}$ .

A medida de similaridade entre dois artistas de acordo com o aspecto contemporaneidade é determinada pelo número de anos em que a carreira coexiste (no caso de haver intersecção) ou pelo número de anos de distância entre as carreiras (no caso de não haver) em relação ao tempo total decorrido quando somadas as carreiras dos dois artistas. Esta medição resulta em um valor no intervalo  $[-1, 1]$ , com o melhor caso sendo aquele em que os artistas iniciaram e terminaram a carreira no mesmo ano, e o pior caso aquele em que ambas as carreiras iniciam

e terminam no mesmo ano, sendo anos diferentes. Ambas as situações estão representadas na Figura 5.1.

Figura 5.1: Melhor e pior caso para a métrica de contemporaneidade.



Fonte: Autoria própria.

Para que esta métrica resulte apenas valores dentro do intervalo  $[0, 1]$ , o mesmo das demais métricas, seu resultado é somando com 1 e dividido por dois, ficando então o pior caso com valor 0 e o melhor caso com valor 1.

Formalmente, a similaridade de dois itens  $i$  e  $j$  em relação à contemporaneidade é dada pela Equação 5.1.

$$sim_{A_1}(i, j) = \left( \frac{\min(a_2^i, a_2^j) - \max(a_1^i, a_1^j)}{\max(a_2^i, a_2^j) - \min(a_1^i, a_1^j)} + 1 \right) / 2 \quad (5.1)$$

Como exemplo, considere-se os artistas *Destiny's Child* e *David Bowie*. O primeiro álbum do *Destiny's Child* foi lançado em 1998, enquanto o último álbum foi lançado em 2004. Já *David Bowie* teve seu primeiro álbum lançado em 1967 e o último em 2016. Sendo  $i = \text{"Destiny's Child"}$  e  $j = \text{"David Bowie"}$ , tem-se o seguinte valor de contemporaneidade entre estes artistas:

$$\begin{aligned} sim_{A_1}(i, j) &= \left( \frac{\min(2004, 2016) - \max(1998, 1967)}{\max(2004, 2016) - \min(1998, 1967)} + 1 \right) / 2 = \\ &= \left( \frac{2004 - 1998}{2016 - 1967} + 1 \right) / 2 \cong 0,56 \end{aligned} \quad (5.2)$$

### Localidade ( $A_2$ )

Para o aspecto *Localidade*, a função extratora  $f_2$  é basicamente uma função identidade, ou seja, a função retorna exatamente o valor encontrado nos dados de entrada. Então, de maneira

similar ao aspecto  $A_1$ , itens são representados por vetores unidimensionais cujo componente é o valor nominal, ou seja, o nome do país.

Em se tratando de um atributo categórico, para computar similaridade de itens nesta forma de representação foi utilizada a métrica *Occurrence Frequency* (OF) (BORIAH; CHANDOLA; KUMAR, 2008), a qual, além de ser adequada para atributos categóricos, explora a frequência dos itens na base de dados como um todo. A métrica atribui o valor 1 para itens que tem o mesmo valor de atributo (*match*) e valores diferentes de 1 para valores diferentes (*mismatch*). Quando os diferentes valores comparados são pouco frequentes na base de dados, a métrica atribui um valor mais baixo do que quando os valores comparados são mais frequentes. Por exemplo, comparando-se um artista dos EUA com um artista da Inglaterra, que são os dois valores mais frequentes na base de dados, tem-se um valor de similaridade maior do que o resultado da comparação entre artistas dos EUA e da Costa Rica, uma vez que a Costa Rica tem muito menos artistas na base que os EUA.

A escolha desta métrica em detrimento de outras que levam em consideração a distância geográfica entre as localidades em questão foi feita com base nas considerações a seguir. Por conta de uma dominação cultural decorrente da política de nações economicamente dominantes, há um grande número de artistas oriundos de países como EUA e Inglaterra, inclusive mais semelhantes entre si do que em relação a países mais próximos, porém economicamente periféricos. Usar a distância geográfica faria com que uma lista contendo artistas de EUA e México fosse menos diversificada do que outra contendo artistas de EUA e Inglaterra.

Outra questão levada em conta foi o fato de que a quase totalidade dos usuários do serviço Last.fm são de países ocidentais, uma vez que usuários asiáticos geralmente utilizem serviços desenvolvidos para as especificidades de suas comunidades. Dessa forma, há a predominância de artistas ocidentais nos históricos dos usuários. Assim, recomendações que tentem maximizar a distância geográfica iriam privilegiar artistas asiáticos, como forma de aumentar a métrica, quando já se tem uma grande diversidade em países geograficamente próximos, como ocorre na Europa.

A equação a seguir define a métrica OF (que aplicada a este aspecto é chamada  $sim_{A_2}$ )



entre dois itens  $i, j$ , no que diz respeito à Localidade:

$$OF_{A_2}(i, j) = \begin{cases} 1 & \text{se } \vec{i} = \vec{j} \\ \frac{1}{1 + \log \frac{|I|}{freq_{A_2}(i)} \times \log \frac{|I|}{freq_{A_2}(j)}} & \text{caso contrário} \end{cases} \quad (5.3)$$

onde  $freq_{A_2}(i)$  retorna o número de artistas que possuem o mesmo valor de atributo (país, neste caso) que o item  $i$ , e  $|I|$  corresponde ao tamanho do conjunto de itens (número de artistas na base).

Tomando-se novamente os exemplos dos artistas *Destiny's Child* e *David Bowie*. Tratam-se de artistas de EUA e Reino Unido, respectivamente. Há na base de dados 9.476 ocorrências de EUA e 3.073 ocorrências de Reino Unido, sendo de 21.729 itens o tamanho da base. Sendo os artistas de localidades diferentes, tem-se :

$$\frac{1}{1 + \log \frac{21729}{9476} \times \log \frac{21729}{3073}} \cong 0,38$$

sendo 0,38 então o valor aproximado de similaridade entre os artistas no aspecto Localidade.

### Gênero musical ( $A_3$ )

Como um artista pode estar associado a múltiplos gêneros musicais, a função extratora de atributos para  $A_3$  é a função  $f_3 : A_3 \times I \rightarrow \{0, 1\}$ , que indica com o valor 1 os gêneros associados ao artista em questão, e com valor 0 os demais gêneros. Cada item é então representado por um vetor binário de gêneros. Para medir similaridade entre dois itens  $i, j$  em relação a este aspecto é usada a função similaridade do cosseno:

$$sim_{A_3}(i, j) = \cos(\vec{i}, \vec{j}). \quad (5.4)$$

Usando os mesmos artistas como exemplo, *Destiny's Child* é considerada do gênero *R&B, pop, soul, hip hop*, enquanto *David Bowie* dos gêneros *Rock, pop, glam rock, art rock, blue-eyed soul, música eletrônica, música experimental*. Considerando os gêneros citados na ordem  $G=R\&B, pop, soul, hip hop, rock, glam rock, art rock, blue-eyed soul, música eletrônica, música experimental$ , e desconsiderando todos os outros gêneros não citados, que não influenciarão no cálculo da distância do cosseno, tem-se a diferença entre os vetores:

- $V_1 = (1, 1, 1, 1, 0, 0, 0, 0, 0, 0)$  para *Destiny's Child*
- $V_2 = (0, 1, 0, 0, 1, 1, 1, 1, 1, 1)$  para *David Bowie*

a distância do cosseno é de 0,19, aproximadamente. Este é o valor de similaridade entre os artistas citados para o aspecto Gênero Musical.

Assim, pode-se finalmente instanciar as funções  $g_a(\cdot)$  and  $h_a(\cdot)$  introduzidas na seção 2.3. Para um dado aspecto  $a \in A$ , uma lista de recomendação  $R$  e o histórico do usuário  $I_u$ , tem-se:

$$g_a(R) = \sum_{(i,j) \in R \times R | i \neq j} sim_a(i, j) \quad (5.5)$$

$$h_a(R, I_u) = \sum_{(i,j) \in R \times I_u | i \neq j} sim_a(i, j) \quad (5.6)$$

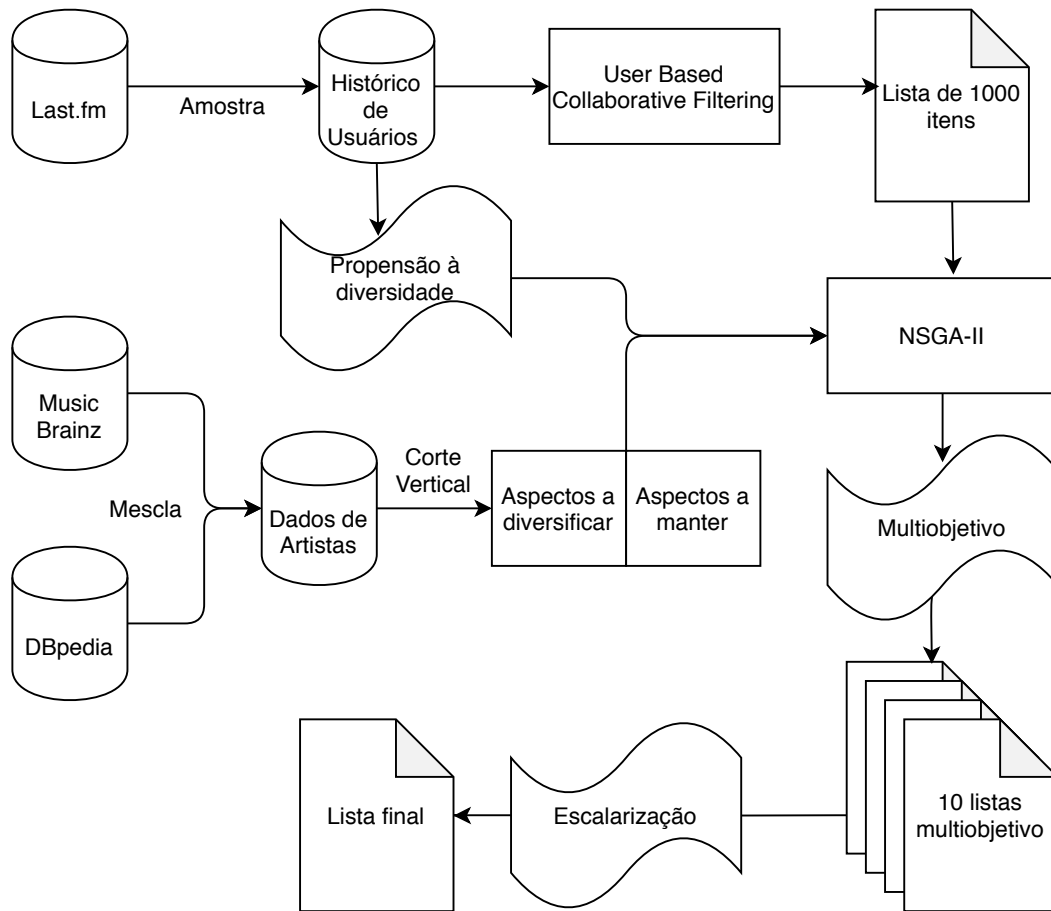
### 5.3 Modelo MOAD

O MOAD, primeiro modelo desta pesquisa, faz uso de um algoritmo MOEA, especificamente o NSGA II, descrito na seção 3.1.1, para gerar recomendações multiobjetivo. O fluxo do modelo é mostrado na Figura 5.2.

Neste modelo, cada usuário tem suas recomendações geradas a partir de uma lista obtida com uso de Filtragem Colaborativa. Este passo é realizado com o algoritmo UBCF, considerando-se um total de 100 vizinhos. Esta lista deve conter uma quantidade de itens que dê ao modelo opções para formar recomendações que otimizem os objetivos em questão, mas eliminando itens posicionados na cauda-longa, ou seja, os artistas mais impopulares da base de dados. Procura-se com esta política pré-selecionar indivíduos possivelmente mais próximos do ótimo, uma vez que o UBCF é bastante utilizado na comunidade de sistemas de informação para gerar recomendações relevantes e normalmente com mais diversidade do que abordagens alternativas, tais como recomendações *Content-based*. Adicionalmente, ao não recomendar artistas da cauda-longa evita-se uma sensação de estranhamento gerada no caso de recomendações compostas apenas de artistas com os quais não se tem familiaridade alguma. Neste trabalho, esta pré-seleção utiliza uma lista de 1.000 artistas.

Esta lista é fornecida juntamente com a propensão à diversificação do usuário ao NSGA II. Para a execução do algoritmo, considera-se como indivíduos listas de recomendação, do

Figura 5.2: Fluxo do MOAD.



Fonte: Autoria própria.

tamanho final a ser gerado pelo modelo, neste caso, 10 itens. Isto se deve ao fato de que, em um algoritmo evolucionário, indivíduos são avaliados quanto à função objetivo proposta, no caso, a função multiobjetivo composta por *Diversidade* e *Afinidade*. As lista de recomendação que atingem melhores valores nesta função são consideradas os indivíduos mais adaptados, sendo mantidos como próxima geração, para a próxima iteração do algoritmo.

Na utilização da função multiobjetivo tem-se a partição dos metadados trabalhados em aspectos a serem diversificados ou não, sendo usadas as métricas de similaridade propostas na seção 5.2.2, realizando comparações dos itens da lista de recomendação par a par para a obtenção da *Diversidade*, ou dos itens recomendados em relação ao itens no histórico de usuário, para obtenção da *Afinidade*.

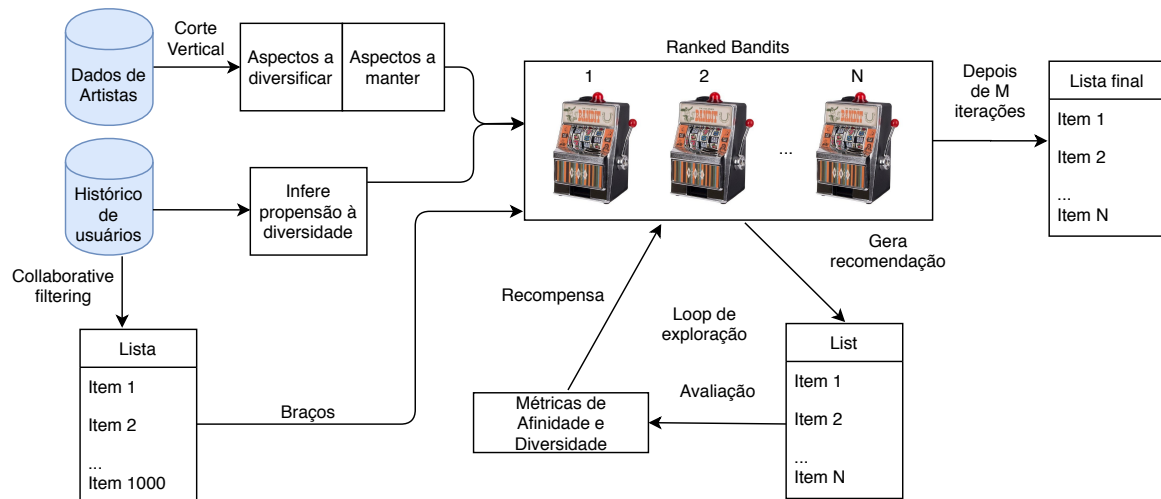
Por fim, o NSGA II, ao fim do ciclo de gerações, resulta em uma população de indivíduos, considerados os mais adaptados. Dentre estes indivíduos, alguns encontram-se na primeira

camada de dominância. Dentre estes, utilizando-se a função de escalarização de objetivos, determina-se a lista de recomendações final. Foi utilizada para este fim a média aritmética dos objetivos, sem ponderação entre estes.

## 5.4 Modelo *Ranked Bandits*

A segunda abordagem proposta para a realização de recomendações multiobjetivo é baseada no modelo de *Ranked Bandits*, onde múltiplas cópias do algoritmo MAB são executadas em paralelo, cada uma sendo responsável pela recomendação de um item para compor a lista final de recomendação. A Figura 5.3 mostra o fluxo do modelo.

Figura 5.3: Fluxo dos *Ranked Bandits*.



Fonte: Autoria própria.

Assim como no modelo MOAD, o histórico dos usuários é usado para a obtenção de uma lista de 1.000 itens, com o algoritmo UBCF, ajustado para 100 vizinhos. No caso de MAB, para trabalhar com um número muito alto de itens ter-se-ia o mesmo número de braços, o que implicaria na necessidade de um número maior de iterações para a aproximação da recomendação ótima. Soma-se a isso o fato dessa pré-seleção de itens fazer com que itens menos populares não façam parte da recomendação.

A inferência acerca das propensões à diversificação de cada usuário também são feitas no modelo *Ranked Bandits*, uma vez que a função de recompensa para cada iteração do modelo é a função multiobjetivo que inclui *Diversidade* e *Afinidade*, assim como no MOAD. Dessa

forma, também é necessário definir subconjuntos de aspectos a serem diversificados ou não.

O modelo executa  $N$  MABs independentes, onde cada máquina possui 1.000 braços, um para recomendar cada artista entre os 1.000 artistas selecionados pelo algoritmo UCBF. Cada máquina, sequencialmente, inclui na lista de recomendação o item equivalente ao braço selecionado como melhor opção pela política de seleção de ações do MAB. Neste modelo utilizou-se a política  $\epsilon$ -greedy, uma vez que a política UCB não se mostrou eficiente neste modelo para explorar melhores soluções. Se alguma máquina tentar incluir na lista um item previamente adicionado por outra máquina, um novo item é escolhido aleatoriamente para substituí-lo. Dessa forma, compõe-se uma lista de recomendações de tamanho  $N$ .

Tendo sido gerada a lista de recomendação, realiza-se sua avaliação com uso da função multiobjetivo, dividindo-se os metadados em dois conjuntos de aspectos, de acordo com a propensão à diversidade aferida para o usuário, a partir dos quais são obtidos os valores de *Afinidade* e *Diversidade* da lista. Com uso de escalarização linear, tem-se um valor único, usado como recompensa para todas as  $N$  máquinas, que atualizam suas políticas de seleção de ações, completando o loop de exploração. Ao fim de  $M$  iterações, tem-se uma lista de recomendação final.

## 5.5 Aplicação do *Ranked Bandits* em Recomendações Online

O modelo *Ranked Bandits* foi proposto com o intuito de, além de otimizar simultaneamente objetivos antagônicos, seja escalável, para que possa fornecer recomendações com certa frequência (podendo ser diárias, semanais, etc.) em um quadro que inclua grande volume de itens e de usuários. O modelo foi então adaptado, como reportado nesta seção, para execução online, considerando a atualização do histórico do usuário ao longo do tempo.

### 5.5.1 Propensão à Diversidade Diária

Conforme demonstrado na seção 6.4, muitos usuários mudam suas propensões à diversidade no decorrer dos dias de uso. Faz-se então necessária a definição de uma forma de aferição dessa propensão para cada dia para o qual se deseja gerar uma recomendação.

Nesse sentido, definiu-se o *Previous Day Intra-List Diversity* (PD-ILD). Considerando-se que, dentre os perfis de propensão possíveis, um perfil já verificado no consumo do usuário tem maior chance de ser mantido, a propensão a ser usada para definir o conjunto de aspectos a ser diversificado em cada recomendação foi baseada no ILD do dia anterior. Assim, considerando-se a definição de ILD da Equação 6.1, o PD-ILD é definido na Equação 5.7:

$$prop_a^{u,d} = \begin{cases} 1, & \text{se } ild_a^{u,d-1} \geq \mu_a + \sigma_a \\ 0, & \text{se } ild_a^{u,d-1} < \mu_a + \sigma_a \end{cases} \quad (5.7)$$

onde  $\mu$  e  $\sigma$  são respectivamente média e desvio padrão do ILD, específicos do aspecto em questão, para todos os usuários da base analisada, e  $d$  refere-se ao dia para o qual será feita a recomendação. Sendo o desvio padrão uma medida da dispersão dos ILDs para a amostra de usuário, usuários com ILDs acima da soma de média e desvio padrão demonstram um desvio mais alto que o esperado para a amostra, sendo então definidos como usuários com propensão à diversidade naquele aspecto.

### 5.5.2 Execução do Modelo

Na execução do modelo online, onde recomendações são feitas para cada dia com base em uma propensão à diversidade verificada diariamente, é provável que um mesmo usuário mude de propensão algumas vezes, voltando inclusive a demonstrar propensão a um cenário que já foi verificado em recomendações anteriores.

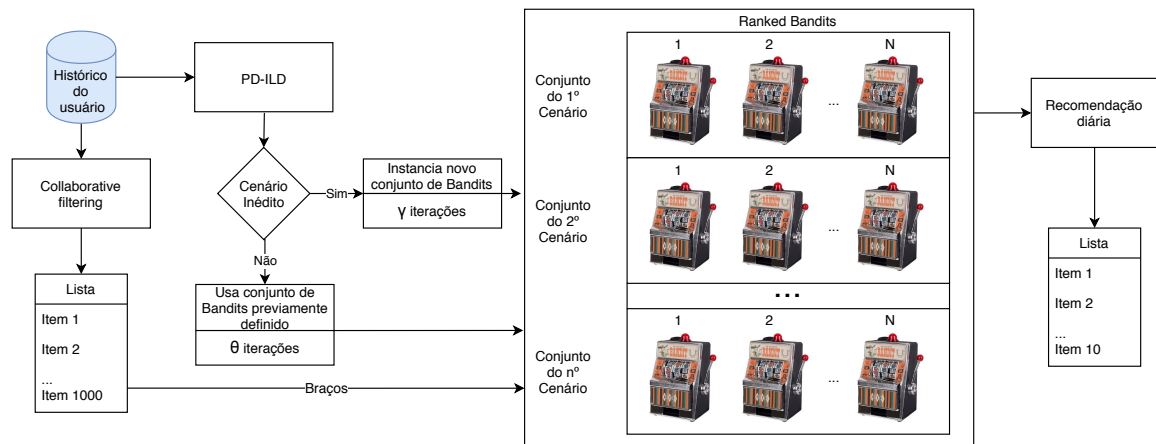
Em cada execução do MAB é mantida a quantidade de vezes que um cada item foi selecionado, além da recompensa média obtida com o uso daquele item. Conforme definido anteriormente, *Ranked Bandits* mantêm múltiplas cópias de MABs, uma para cada posição a ser preenchida por um item na lista de recomendação.

No modelo de recomendações online, então, múltiplos conjuntos de *Ranked Bandits* são mantidos, com o intuito de usar cada um deles para um cenário de propensão diferente. Dessa forma, quando o usuário muda sua propensão para um cenário que já foi processado anteriormente, é utilizado o conjunto de *Ranked Bandits* que manteve os números de seleções e recompensas de itens obtidos no processamento daquele cenário. Com isso, parte-se de um ponto em que já houve um número de iterações do algoritmo em que diversas explorações foram realizadas, sendo possível continuar deste ponto.

É necessária então a definição de quantas iterações são necessárias quando a recomendação é feita pela primeira vez em um determinado cenário de propensão, fazendo com que um novo conjunto de MABs seja executado pela primeira vez. Deve-se definir também quantas execuções são necessárias ao se fazer uma recomendação que retoma um cenário já visitado previamente. Dois parâmetros determinam essas informações dentro do modelo:  $\gamma$ , para o número de execuções na primeira vez que o cenário é definido para o usuário, e  $\theta$ , para cenários já executados.

Assim como no modelo de recomendação offline, a recomendação parte de um conjunto de itens gerados por Filtragem Colaborativa. Recomendações são geradas a partir do segundo dia de histórico do usuário, uma vez que se utiliza o primeiro para identificar o perfil de propensão a ser utilizado, através do PD-ILD. A cada dia, os artistas presentes no histórico em qualquer dia anterior não são considerados para recomendação. Definido então o PD-ILD do dia anterior, caso o cenário de propensão já tenha sido utilizado para recomendação, são realizadas  $\theta$  iterações, usando o conjunto de *bandits* já instanciado e utilizado nas recomendações passadas neste mesmo cenário. Caso seja um cenário ainda não utilizado para este usuário, é instanciado um novo conjunto de *bandits*, no qual são realizadas  $\gamma$  iterações. Seja qual for o caso, tem-se uma lista de recomendações para o dia em questão. A Figura 5.4 sintetiza a adaptação completa do modelo definido para execução online.

Figura 5.4: Modelo *Ranked Bandits* para execução online.



Fonte: Autoria própria.

## 5.6 Considerações Finais

Neste capítulo foram apresentadas as soluções multiobjetivo baseadas em aspectos para o problema proposto. Foram discutidas a forma de representação de um item e as métricas usadas nas comparações entre itens. Acerca dos modelos propostos, mostrou-se como os algoritmos utilizados foram adaptados para o problema em questão, e de que forma os modelos atingem suas soluções.



## Capítulo 6

# Tratamento e Análise dos Dados

Neste capítulo serão apresentadas as fontes e os tratamentos de limpeza e estruturação realizados nos dados utilizados para a avaliação. Além disso, serão mostradas análises estatísticas exploratórias sobre os dados históricos dos usuários, com o intuito de verificar a existência de variação no padrão de consumo, com a diversidade variando entre os aspectos de maneira diferente a cada usuário.

### 6.1 Coleta de Dados e Preparação

Para a avaliação dos modelos propostos foram utilizadas três bases de dados públicas: Last.fm, Music Brainz e DBpedia. Last.fm é uma rede social na qual os usuários compartilham dados acerca de seus hábitos musicais. Cada música executada é registrada, gerando uma informação chamada de *scrobble*, que contém dados da faixa, tais como título, artista e álbum, além de um registro temporal (*timestamp*) do momento de execução da música. Em particular, foi utilizada uma base de dados compilada a partir dos dados obtidos na ferramenta Last.fm, publicada e disponibilizada por Schedl (2016)

A extração de aspectos foi feita sobre metadados extraídos da Music Brainz, uma enciclopédia musical que disponibiliza dados acerca de artistas, gravações, álbuns, etc. Além do *dump* completo da base de dados, o Music Brainz disponibiliza o esquema da base e dicionário completo dos dados. Para a composição dos aspectos definidos para o problema, os seguintes atributos foram extraídos:

- *Contemporaneidade*

Foram utilizadas informações da tabela *release*, que contém informações dos álbuns lançados. Esta tabela possui ligação com a tabela *artist* através da tabela *artist\_credit*. Nas tabelas *release\_country* e *release\_unknown\_country* há a data de lançamento de cada álbum. Entre todos os álbuns de cada artista, foi selecionado o menor ano, obtendo-se o ano de lançamento do primeiro álbum do artista, e o maior, referente ao lançamento do último álbum, informações que compõem o aspecto *Contemporaneidade*.

- *Localidade*

Para obtenção da *Localidade* do artista foi usada a tabela *area*, acessível a partir de chave estrangeira na tabela *artist*. A tabela *area* contém o nome do país de origem do artista, ou o país com o qual o grupo possui uma maior identificação.

Como o Music Brainz não possui dados acerca do gênero musical dos artistas, o DBpedia foi utilizado para obter tais informações. A versão utilizada foi a Data Set 3.0 e os dados sobre gênero musical foram obtidos na tabela *MusicalArtist*, no atributo *genre\_label*. O atributo é do tipo texto, mas com separadores, no formato  $G = \{g_1|g_2|...|g_n\}$ , a partir do qual foi gerado um vetor binário, onde cada gênero é um elemento distinto do vetor.

Depois de preencher os dados de artistas com os aspectos acima mencionados, gêneros associados a menos que 5 artistas foram removidos, bem como artistas sem dados para algum dos aspectos observados. Finalmente, uma amostra de 1.000 usuários da base de dados do Last.fm foi selecionada aleatoriamente. Este número de usuários é condizente com o tamanho de outras bases do Last.fm conhecidas na comunidade de sistemas de recomendação, como por exemplo *Last-fm - 1K users dataset*<sup>1</sup> (CELMA, 2010). Nessa amostra, 3 usuários não tinham histórico, sendo então descartados.

A tabela 6.1 mostra algumas estatísticas acerca da base de dados constituída para a avaliação da abordagem deste trabalho.

A base de dados resultante foi dividida em duas partes, para treino e teste, na razão de 80% dos dados para o conjunto de treino e 20% dos dados para o conjunto de teste. Essa divisão foi feita de maneira cronológica por usuário, ou seja, para cada usuário, os primeiros 80% do seu histórico de consumo fazem parte do conjunto de treino, enquanto os 20% finais

---

<sup>1</sup><<http://www.dtic.upf.edu/~ocelma/MusicRecommendationDataset/lastfm-1K.html>>

Tabela 6.1: Estatísticas acerca da base de dados utilizada

<b>Estatística</b>	<b>Valor</b>
Número de artistas	21.729
Artistas por usuário	140
Contemporaneidade	1212 a 2017
Localidades	437
Gêneros musicais	847

Fonte: Autoria própria.

formam o conjunto de teste.

## 6.2 Verificação da Propensão a Diversidade

A realização de recomendações levando em consideração diferentes aspectos musicais, buscando diversificar alguns aspectos enquanto outros são mantidos constantes, leva em consideração a percepção de que usuários consomem música de maneira bastante variada. Tal variação na forma de ouvir música leva à definição de diferentes perfis de usuários, onde cada perfil indica a preferência do usuário no que se refere a cada um dos aspectos trabalhados.

Para validar a percepção de que existe essa separação de usuários que consomem música com diferentes gostos em relação a diversidade, examinou-se a existência de usuários com os diferentes perfis possíveis através dos seus históricos de consumo. Considerando o histórico de cada usuário como uma lista de artistas escutados, foi feita uma verificação, para cada aspecto separadamente, da diversidade entre esses artistas, usando-se as funções de medida de similaridade definidas na seção 5.2.2, para gerar um valor de ILD do usuário em relação àquele aspecto.

Com o intuito de perceber o quão diversificado para um determinado aspecto é o gosto do usuário num dado momento, e não ao longo de todo o seu histórico, optou-se por dividir o histórico em consumos diários, sendo o ILD do usuário formado então pela média dos seus ILDs diários. Tal escolha se justifica pelo fato de que uma recomendação é realizada para consumo em um dado momento. Adicionalmente, o cálculo do ILD uma operação com alto custo computacional, notadamente no caso de usuários com grande histórico de itens escutados. Isso porque a obtenção do valor de ILD exige comparação de elementos dois a dois, para todos os elementos que constam da lista em análise, no caso, histórico dos

usuários. A Equação 6.1 define formalmente o cálculo realizado.

$$ild_a^u = 1 - \frac{1}{|D|} \sum_{d=0}^D g_a(I_u^d), \quad (6.1)$$

onde  $ild_a^u$  corresponde ao ILD do usuário  $u$  para o aspecto  $a$ ,  $D$  é o conjunto de dias nos quais o histórico  $I_u$  possui alguma entrada e  $g_a$  é a função de similaridade para o aspecto  $a$ .

De posse dos ILDs de cada usuário, obteve-se o ILD médio ( $\mu$ ) e o desvio padrão ( $\sigma$ ), considerando-se toda a base de usuários. Assim, as propensões à diversidade de um usuário em relação a cada aspecto são definidas como:

$$prop_a^u = \begin{cases} 1, & \text{se } ild_a^u \geq \mu + \sigma \\ 0, & \text{se } ild_a^u < \mu + \sigma \end{cases} \quad (6.2)$$

### 6.3 Cenários de Propensão

Levando em consideração que três aspectos musicais são abordados neste trabalho, sendo eles contemporaneidade, localidade e gênero musical, oito cenários são possíveis, combinando-se as duas possibilidades de propensão a diversidade (positiva ou negativa) entre si. A Tabela 6.2 mostra essas combinações e os cenários resultantes.

Tabela 6.2: Cenários avaliados

#	Diversificar	Conservar afinidade
1	–	Contemp., Localidade, Gênero musical
2	Contemporaneidade	Localidade, Gênero musical
3	Localidade	Contemporaneidade, Gênero musical
4	Gênero musical	Contemporaneidade, Localidade
5	Contemporaneidade, Localidade	Gênero musical
6	Contemporaneidade, Gênero musical	Localidade
7	Localidade, Gênero musical	Contemporaneidade
8	Contemp., Localidade, Gênero musical	–

Fonte: Autoria própria.

## 6.4 Distribuição de Usuários

Uma análise da distribuição de usuários nestes oito cenários possíveis foi feita tanto para a base de dados de 1.000 usuários usada na validação dos modelos e descrita na seção 6.1, como sobre um subconjunto da base *Music Listening Histories Dataset* (MLHD)<sup>2</sup>, gerada por Vigliensoni e Fujinaga (2017) a partir de 27 bilhões de registros de consumo musical extraídos do Last.fm<sup>3</sup>. Esta segunda análise foi possível devido ao menor esforço computacional necessário para a análise.

### 6.4.1 Histórico Completo

Para um subconjunto de 10.061 usuários, correspondentes aos arquivos de índice 000 a 009 da base de dados MLHD, obteve-se a distribuição exibida na Figura 6.1, considerando-se o histórico completo para a distribuição em perfis, ou seja, a média de ILDs incluindo todos os dias em que o usuário ouviu algum item. Os cenários no eixo  $x$  da figura correspondem aos descritos na Tabela 6.2. Percebe-se que a quase totalidade dos usuários analisados ouve itens ou de maneira conservadora (sem diversidade), ou diversificando, no máximo, um dos aspectos considerados, o que corresponde aos cenários 1 a 4. Entretanto, a necessidade de se levar em conta a diversidade orientada a aspectos se justifica pelo fato de que, excetuando-se o cenário 1, onde não há diversidade em aspecto algum, tem-se um total de 7.566 usuários, ou cerca de 75% dos usuários analisados.

Aplicando-se a mesma análise para o conjunto de 1.000 usuários usado nas avaliações dos Capítulos 7 e 8, têm-se os resultados mostrados na Figura 6.2. A distribuição de cenários de propensão para esta amostra é proporcional à distribuição para os 10 mil usuários da base MLHD, apresentada na Figura 6.1.

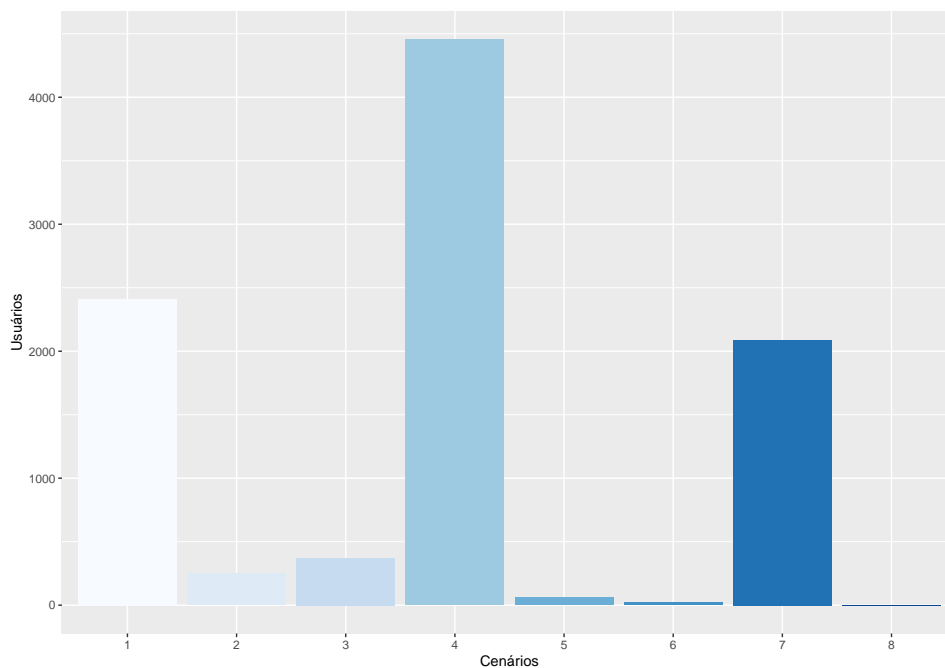
### 6.4.2 Histórico Particionado em Dias

Outra análise realizada diz respeito às mudanças de perfil ao longo do histórico do usuário. Nesta verificação deseja-se saber em quantos diferentes perfis de propensão o usuário se enquadra ao longo dos dias do seu histórico. A Figura 6.3 mostra, no eixo  $x$ , a quantidade

<sup>2</sup>Disponível em [https://ddmal.music.mcgill.ca/research/The\\_Music\\_Listening\\_Histories\\_Dataset\\_\(MLHD\)/](https://ddmal.music.mcgill.ca/research/The_Music_Listening_Histories_Dataset_(MLHD)/)

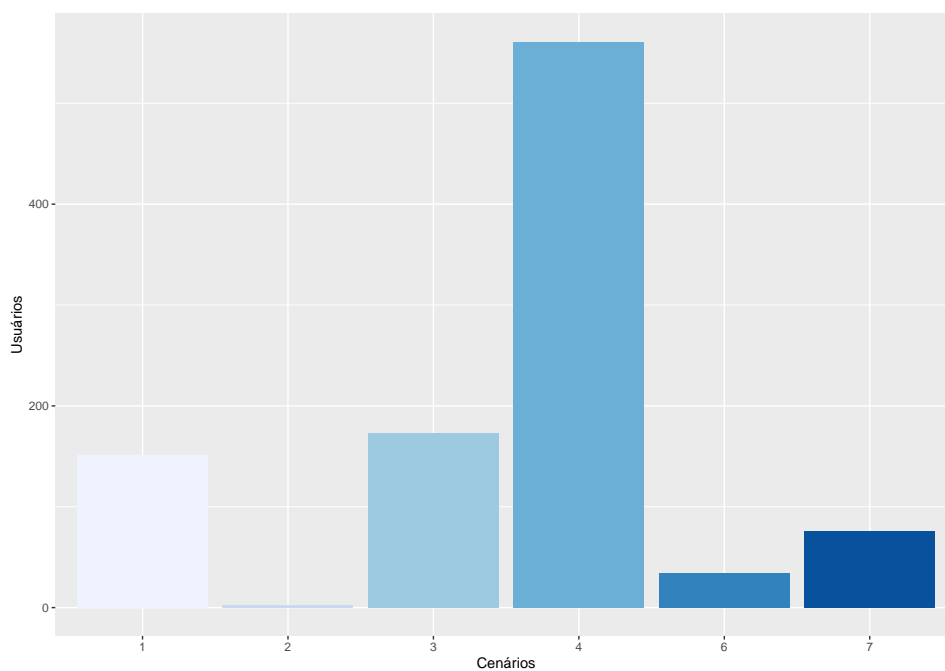
<sup>3</sup><https://www.last.fm/>

Figura 6.1: Distribuição de usuários nos perfis de propensão a diversidade considerando histórico completo.



Fonte: Autoria própria.

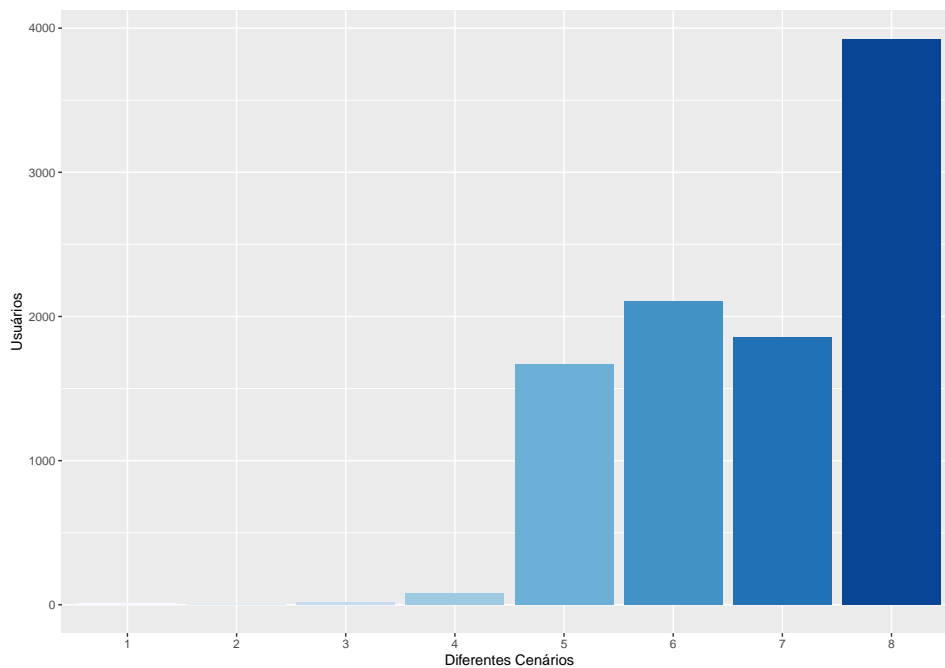
Figura 6.2: Distribuição de usuários avaliados nos perfis de propensão a diversidade considerando histórico completo.



Fonte: Autoria própria.

de diferentes cenários de propensão ocupados por um usuário, enquanto o eixo *y* mostra quantos usuários se enquadram naquela quantidade de cenários. Verifica-se, com auxílio da Figura 6.4, que quase a totalidade dos usuários variou ao menos 5 cenários de propensão ao longo do seu histórico, o que reforça a necessidade de uma forma de recomendação online, em que o sistema de recomendação possa se adaptar às constantes mudanças de perfil de propensão do usuário.

Figura 6.3: Distribuição de usuários nos perfis de propensão a diversidade considerando histórico completo.

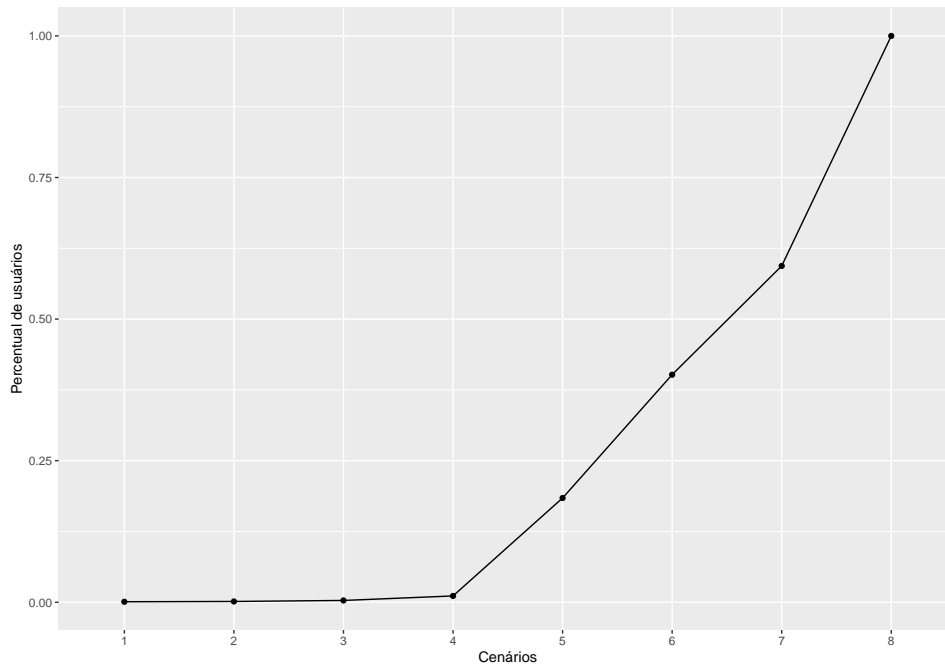


Fonte: Autoria própria.

## 6.5 Considerações Finais

Neste capítulo foram demonstradas a fonte dos dados usados na avaliação dos modelos propostos, além de como esses dados foram extraídos e tratados para servirem de histórico de usuários e metadados acerca dos itens a serem recomendados pelos modelos. Foi realizado também um estudo sobre as propensões de usuários a consumir itens da base de dados, artistas musicais no caso, de maneira diversificada ou não, separando-se essa propensão por aspecto considerado. Esta análise demonstra a veracidade do senso comum de que dife-

Figura 6.4: Distribuição acumulada de usuários de acordo com o número de mudanças de cenários de propensão.



Fonte: Autoria própria.

rentes usuários possuem diferentes propensões a diversidade, e que este comportamento se diferencia pra um mesmo usuário, considerando-se diferentes aspectos musicais. Alia-se a isto o fato de que, ao longo do tempo, um mesmo usuário pode variar em termos de sua propensão à diversificar diferentes aspectos. Esta análise dá então justificativa pra o modelo de recomendação proposto nos capítulos subsequentes.



## Capítulo 7

### Avaliação Offline

Neste capítulo, é discutido o protocolo de avaliação e os resultados obtidos pelos modelos propostos, em face a algoritmos semelhantes da literatura. Especificamente, será abordada nesse capítulo a avaliação *offline*, ou seja, avaliação feita a partir de dados previamente coletados. A razão para se fazer este tipo de avaliação é a dificuldade de se ter uma amostra significativa de usuários usando o sistema de recomendações e reagindo às recomendações feitas, de maneira a permitir uma avaliação *online*.

Foram desenvolvidos então protótipos para as duas abordagens propostas neste trabalho. A primeira delas, o MOAD, usa algoritmos evolucionários, mais precisamente MOEA, enquanto a outra abordagem usa *Ranked Bandits*. Os protótipos foram construídos respectivamente em linguagem R (R Development Core Team, 2008) e Python (ROSSUM, 1995), usando Pandas (MCKINNEY, 2010). Utilizando o sistema desenvolvido, foram executadas simulações de recomendações para um grupo de usuários reais, utilizando-se bases de dados públicas para a coleta de metadados, conforme exposto no Capítulo 6. O experimento inclui históricos de consumo dos usuários e metadados de artistas reais.

Espera-se, com os experimentos descritos neste capítulo, poder verificar a efetividade dos modelos multiobjetivo em atuar em conjuntos de aspectos específicos, diversificando um conjunto de aspectos sem que haja degradação da relevância no conjunto de aspectos não selecionado para diversificação, garantindo a relevância dos resultados em relação à configuração realizada.

## 7.1 Determinação de Propensão à Diversidade

O primeiro passo para a validação dos modelos propostos é a inferência do perfil de consumo do usuário, em relação a diversificação ou não de cada um dos aspectos analisados, de forma que se possa atribuir o usuário a um dos cenários de diversificação estabelecidos na Tabela 6.2. Esta inferência é feita considerando-se a diversidade em cada aspecto verificada no conjunto de treino, ou seja, parte do histórico de itens ouvidos pelo usuário.

Como se deseja verificar a diversidade de cada aspecto, utilizou-se o ILD, aplicado a cada aspecto, usando as medidas de similaridade  $sim_{A_1}, \dots, sim_{A_3}$ , definidas na seção 5.2.2. Entretanto, como o ILD mede dissemelhança, então as métricas são invertidas para métricas de dissemelhança, uma vez que todas retornam valores no intervalo  $[0, 1]$ . O ILD é calculado então, para cada usuário  $u$  e aspecto  $a$ , com base na Equação 7.1.

$$ILD(I_u^a) = \frac{\left( \sum_{i=1}^{N-1} \sum_{j=i+1}^N 1 - sim^a(I_u^i, I_u^j) \right)}{N(N-1)}, \quad (7.1)$$

onde  $N = |I_u|$ .

Entretanto, alguns históricos de usuários podem alcançar um número bastante alto de itens. Na base analisada, o maior histórico, considerando apenas o conjunto de treino, possui 1.407 artistas. Calcular o ILD nestes casos é uma operação bastante custosa. Alternativamente, preferiu-se dividir o histórico do usuário em dias, calculando-se o ILD individualmente para cada dia e efetuando-se a média, para obter o ILD médio do usuário.

Para determinar o cenário de propensão de um usuário, para cada aspecto, compara-se o seu ILD médio com um limiar definido como sendo o ILD médio somado ao desvio padrão de toda a base de dados. Dessa forma, tem-se a propensão à diversidade de cada usuário, para cada aspecto, o que serve como entrada para os modelos de recomendação multiobjetivo avaliados neste capítulo.

Tabela 7.1: ILD e desvio padrão da base analisada, por aspecto.

	ILD	Desvio padrão	ILD + Desvio Padrão
Contemporaneidade	0.2442286	0.04717814	0.2914068
Localidade	0.3159195	0.12756527	0.4434848
Gênero Musical	0.8239307	0.11078188	0.9347125

Fonte: Autoria própria.

Na base de dados trabalhada, tem-se os valores de ILD e desvio padrão por aspecto mostrados na Tabela 7.1. Considerando então, por exemplo, o usuário com identificador 29970410, que possui os ILDs médios  $I = \{0.2597892, 0.4569470, 0.8765371\}$  para *Contemporaneidade*, *Localidade* e *Gênero Musical*, respectivamente. Em comparação com os limiares estabelecidos na última coluna da tabela 7.1, tem-se um usuário com diversidade alta para *Localidade* e baixa nos demais aspectos, o que o associa ao cenário de propensão 3 da Tabela 6.2.

## 7.2 Algoritmos *Baselines*

Para avaliar o desempenho do MOAD e do *Ranked Bandits*, alguns algoritmos foram selecionados para servir como *baseline*. A relação de *baselines* é formada por algoritmos não personalizados (*random* e *popular*), algoritmos que não buscam recomendações diversas (UBCF) e algoritmos focados em diversificação (MMR e RxQuAD). Todos os algoritmos foram configurados para produzir uma lista de 10 recomendações.

Vale ressaltar que, até onde se estendeu esta pesquisa, não foram encontrados métodos compatíveis com a proposta do MOAD e do *Ranked Bandits* de realizar a diversificação por aspectos. Dessa forma, os *baselines* utilizados estão sendo adaptados para a execução em cenários diferentes daqueles para os quais foram desenvolvidos. Suas escolhas, notadamente do MMR e do RxQuAD, se devem ao fato de que são algoritmos reconhecidamente eficazes em realizar recomendações diversas com baixa degradação da acurácia, no que se assemelham ao MOAD e ao *Ranked Bandits*. É, entretanto, de se esperar que estes obtenham resultados superiores, uma vez que estão sendo executados dentro do contexto para o qual foram desenvolvidos, o que não ocorre com os algoritmos *baselines*, que não foram desenvolvidos para a otimização de dois objetivos separados por grupo de aspectos.

### 7.2.1 *Random*

O algoritmo *Random* simplesmente escolhe itens aleatórios para compor a lista de recomendação.

### 7.2.2 Mais Popular

O algoritmo Mais Popular gera listas de recomendação baseado nos itens mais frequentes na base de histórico considerando todos os usuários. O Mais Popular não recomenda itens presentes no histórico do usuário que está recebendo a recomendação.

### 7.2.3 UBCF

O algoritmo baseia-se no algoritmo kNN para encontrar os  $k$  vizinhos mais próximos do usuário. As recomendações são retiradas dos históricos de consumo destes vizinhos, com base na intuição de que usuários com preferências similares tendem a se interessar pelos mesmos itens. O UBCF tende a recomendar itens com alta relevância, entretanto, não há garantias que diversidade vai ser atingida, com uma propensão muitas vezes à relevância em detrimento da diversidade. Nesta execução, utilizou-se o UBCF para 5 vizinhos e método Similaridade do Cosseno para medir a similaridade entre vizinhos.

### 7.2.4 MMR

O algoritmo MMR foi tratado na seção 4.1. Por se tratar de um algoritmo de reranqueamento MMR precisa receber como entrada uma lista de recomendações substancialmente maior que o tamanho da lista final, nesta execução foi gerada uma lista de 50 elementos usando o UBCF, assim como realizado nos experimentos de Ziegler et al. (2005), para que houvesse a seleção dos 10 elementos componentes da lista final.

### 7.2.5 RxQuAD

O RxQuAD foi tratado na seção 4.1. Assim como o MMR, realiza também um reranqueamento, necessitando receber uma lista de recomendação de entrada em quantidade substancialmente maior que a lista final a ser produzida. Nesta execução recebeu os mesmos 50 itens que o MMR, gerando uma lista de saída de 10 itens.

Os resultados dos algoritmos random, popular e UBCF foram obtidos com o uso do pacote *recommenderlab* (HAHSLER, 2015), da linguagem R. O MMR e o RxQuAD foram executados com auxílio da ferramenta RankSys (VARGAS; CASTELLS; VALLET, 2011,

2012), onde já se encontram implementados.

## 7.3 Avaliação do MOAD

Nesta seção definem-se o protocolo de avaliação do modelo MOAD e como foram decididos os parâmetros a serem fornecidos na sua execução.

### 7.3.1 Protocolo de Avaliação

O protocolo de avaliação consiste das seguintes etapas: a partir da amostra da base de dados Last.fm, é realizado o procedimento de inferência da propensão à diversidade de cada usuário, o que é uma das informações de entrada para o MOAD. Na sequência, para cada um dos usuários, é gerada uma lista de 1.000 itens usando o algoritmo UBCF, com uso do pacote RecommenderLab (HAHSLER, 2015) ajustado para utilizar o histórico de 100 vizinhos. Com isso, espera-se evitar recomendações de artistas completamente desconhecidos, além de se reduzir o espaço de busca, levando o algoritmo a convergir mais rapidamente.

Os metadados acerca dos 1.000 artistas selecionados, representando os aspectos, são divididos em dois grupos, respeitando a configuração de aspectos a diversificar e a manter afinidade estabelecidos pela propensão do usuário. Estes aspectos, juntamente com os históricos de usuários da amostra da base Last.fm são submetidos ao algoritmo NSGA II, que irá gerar populações aleatórias de tamanho 10 e avaliá-las de acordo com os objetivos *Diversity* e *Affinity*, medidos no conjunto de atributos apropriado. Ao final das iterações, será fornecido um total de 10 listas de recomendação próximas à fronteira de Pareto.

Durante a execução das simulações do MOAD, o NSGA II foi executado com população inicial de 10 indivíduos, cada um correspondendo a uma lista de recomendação composta por 10 artistas. Os parâmetros foram ajustados para 20 gerações, probabilidade de mutação de 0,1 e probabilidade de *crossing over* de 0,9. A escolha destes valores será detalhada na subseção 7.3.2. Das 10 listas de recomendação fornecidas, apenas uma é selecionada para ser a lista final de recomendação.

Para verificar a variabilidade do MOAD ao longo de diferentes execuções, a simulação de cada cenário foi executada 8 vezes. Este número foi escolhido pela facilidade de rodar todas as execuções em paralelo em uma máquina de oito núcleos. Para cada cenário, foi executado

um teste de Kruskal-Wallis, que não exige dados normais e cuja hipótese nula indica que os resultados obtidos em cada execução seriam populações idênticas.

Como indicado na seção 2.3, o problema analisado consiste em otimizar dois objetivos simultaneamente. Para avaliar os resultados foram calculados *Diversity* e *Affinity* para as listas de recomendações finais de cada usuário no conjunto de dados experimental. Como mencionado anteriormente, para computar o *Affinity*, o conjunto de testes de cada usuário foi usado como histórico de consumo. Tratam-se de itens cuja relevância é atestada pelo fato de que o usuário efetivamente os consumiu, porém, com a divisão cronológica do histórico em conjuntos de treino e teste, o conjunto de testes representa de certa forma as preferências futuras do usuário, o que é um indicativo do que deve ser recomendado por um sistema de recomendação.

### 7.3.2 Ajuste de Parâmetros do NSGA II

O NSGA II requer o fornecimento de alguns parâmetros para executar, que são número de gerações, tamanho da população e probabilidade de mutação. Para a determinação de valores adequados à abordagem proposta, foi extraída uma subamostra de 30 usuários da base experimental do Last.fm e foi determinado um cenário fixo para a realização de algumas execuções. Foi utilizado o quarto cenário da tabela 6.2 e os valores padrão exibidos na tabela 7.2.

Tabela 7.2: Valores padrão para os parâmetros do NSGA II.

Parâmetro	Valor padrão
Tamanho da população ( $N$ )	10
Número de gerações ( $nGen$ )	10
Probabilidade de mutação ( $mProb$ )	0,1
Probabilidade de <i>crossing over</i> ( $cProb$ )	0,9

Fonte: Autoria própria.

Para ajustar um parâmetro em particular, os demais parâmetros foram fixados nos seus valores padrão, enquanto o parâmetro alvo teve seu valor incrementado até que não fossem encontradas mudanças significativas nas métricas de avaliação, com o que chegou-se ao valor ideal para as simulações.

Devido à não normalidade dos dados, o teste não paramétrico de Wilcoxon foi utilizado

para verificar a significância das alterações de parâmetros. A tabela 7.3 mostra o p-valor obtido no teste de Wilcoxon para cada parâmetro avaliado.

Tabela 7.3: *p-valor* resultante dos testes de Wilcoxon para a determinação dos parâmetros do NSGA II.

Parâmetro	Valores	Objetivo 1	Objetivo 2
Gerações	10-20	0.6274	9.536e-10
	20-30	0.1053	0.0006884
	30-50	0.8637	9.037e-07
Tamanho da população	5-10	0.9343	0.01147
	10-20	0.2008	0.3066
	20-30	0.4949	0.7973
Probabilidade de mutação	0.1-0.5	0.9773	0.9976
	0.5-0.9	0.4764	0.9985

Fonte: Autoria própria.

Na tabela 7.3, a coluna *Valores* mostra os valores diferentes que o parâmetro recebeu em cada comparação. Como se pode verificar, aumentar a probabilidade de mutação não resulta em incremento dos objetivos e uma população de tamanho 10 mostrou-se suficiente. O número de gerações, entretanto, influencia no objetivo 2 a cada vez que é incrementado, de acordo com o teste de Wilcoxon. Com essas análises, decidiu-se executar as simulações com tamanho de população de 10 indivíduos, probabilidade de mutação de 0,1 e 20 gerações. Não foram executadas simulações de 50 ou mais gerações, já que 20 gerações mostrou-se suficiente para o alcance de bons resultados.

No que diz respeito às métricas de avaliação, foram utilizadas *diversity* e *affinity*, definidas nas equações 2.4 e 2.5, respectivamente. *diversity* é definida em termos do ILD, enquanto *affinity* tenta estimar a relevância das recomendações levando em consideração os aspectos mantidos contantes, medindo a similaridade entre os itens que compõem a lista de recomendações e os itens presentes no histórico do usuário. A intuição é que itens consumidos pelo usuário são considerados relevantes, então itens semelhantes a estes também o seriam. Enquanto *diversity* é verificada na lista de recomendação final, *affinity* é verificada em relação ao conjunto de teste.

## 7.4 Avaliação Ranked Bandits

Nesta seção, o segundo modelo é avaliado ainda de maneira offline, assim como o modelo anterior. São analisados os parâmetros definidos para a execução da avaliação e suas escolhas são justificadas.

### 7.4.1 Ajuste de Parâmetros

Apenas dois parâmetros devem ser fornecidos para um MAB com política de seleção de ações  $\epsilon$ -greedy: o valor de  $\epsilon$  e o número de iterações. Executou-se o modelo completo usando os valores de  $\epsilon$  0,05, 0,1, 0,2 e 0,3, sempre pra um total de 500 iterações por usuário.

Para verificar o melhor valor para  $\epsilon$ , em cada uma das execuções do algoritmo obteve-se a média do valor escalarizado dos dois objetivos analisados durante as 100 últimas iterações. Como o algoritmo realiza constantes explorações, usar uma única execução do algoritmo provavelmente resultará em um valor que não é representativo do seu desempenho de maneira geral, tendo em vista que a exploração realizada pode resultar numa recompensa menor, fazendo com que o item selecionado não seja aproveitado nas próximas iterações. A utilização da média reduz o impacto de explorações com resultados inferiores.

Foram geradas distribuições contendo as médias obtidas para todos os usuários da amostra, sendo uma distribuição para cada valor de  $\epsilon$ . As três distribuições foram comparadas usando o teste não paramétrico de Kruskal-Wallis, por se tratarem de distribuições não normais. Neste teste, a hipótese nula foi rejeitada, o que atesta a hipótese alternativa de que há ao menos uma distribuição não equivalente às demais. Para identificar a melhor delas então, executaram-se testes de Wilcoxon usando as distribuições duas a duas, pareadas, uma vez que valores na mesma posição na distribuição provêm de um mesmo usuário. A Tabela 7.4 mostra os p-valores obtidos.

Tabela 7.4:  $p$ -valor do teste de Wilcoxon para diferentes valores de  $\epsilon$  no algoritmo *Ranked Bandits*  $\epsilon$ -greedy. .

Comparação \ Alternativa	less	two-sided	greater
0,05 x 0,1	5.706e-08	1.141e-07	1.0
0,1 x 0,2	1.0	<2.2e-16	<2.2e-16
0,2 x 0,3	1.0	<2.2e-16	<2.2e-16

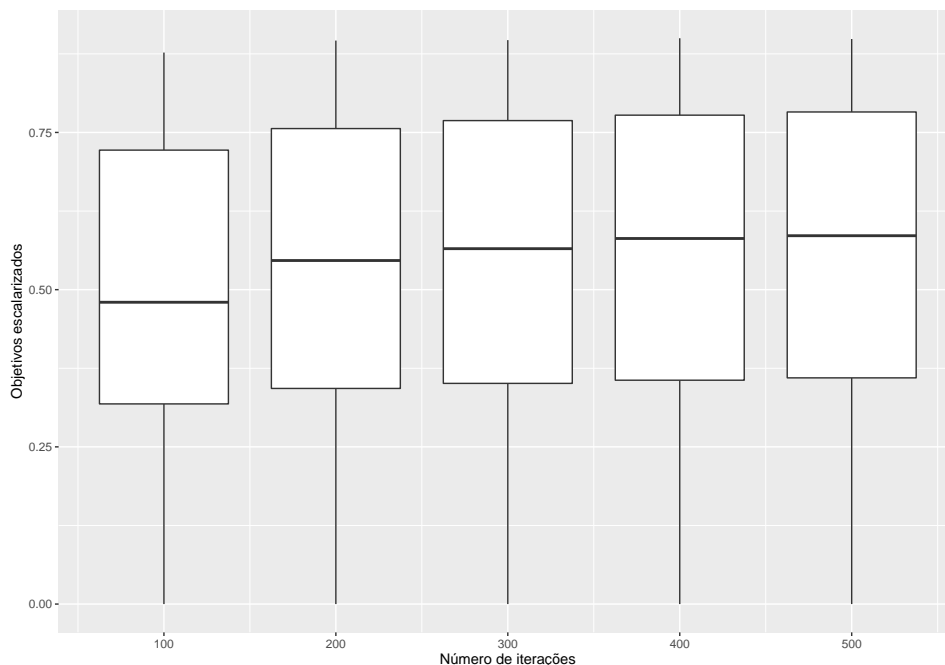
Fonte: Autoria própria.



Verifica-se, nos resultados dos testes, que utilizando-se  $\epsilon = 0,1$  os resultados obtidos foram superiores aos atingidos com  $\epsilon = 0,05$  (p-valor  $< 0,05$  no teste *less*), devido ao fato de que o valor menor de  $\epsilon$  faz com que o algoritmo explore menos, precisando de mais iterações pra atingir os mesmos resultados. Também utilizando-se  $\epsilon = 0,1$  têm-se resultados superiores aos atingidos com  $\epsilon = 0,2$  (p-valor  $< 0,05$  no teste *greater*), pois o excesso de explorações com  $\epsilon = 0,2$  faz com que os bons resultados não sejam aproveitados na mesma proporção. Percebe-se também, com a comparação feita com os resultados usando  $\epsilon = 0,2$  e  $\epsilon = 0,3$ , que os resultados seguem piorando à medida que se aumenta o valor de  $\epsilon$ . Dessa forma, o valor de  $\epsilon = 0,1$  foi fixado para todos os experimentos.

Para verificar o número de iterações ideal, todos os usuários da amostra foram submetidos a um ciclo de 500 recomendações. A Figura 7.1 mostra as distribuições das médias dos objetivos *Diversity* e *Affinity* escalarizados, medidos a cada 100 iterações. Embora tenham sido realizados testes de Wilcoxon, que mostram que há diferença significativa entre todos os grupos, graficamente percebe-se que a partir de 100 iterações há uma evolução muito pequena, não justificando a realização de mais de 100 iterações.

Figura 7.1: Comparativo da evolução da média dos objetivos escalarizados a cada 100 iterações.



Fonte: Autoria própria.

## 7.5 Resultados

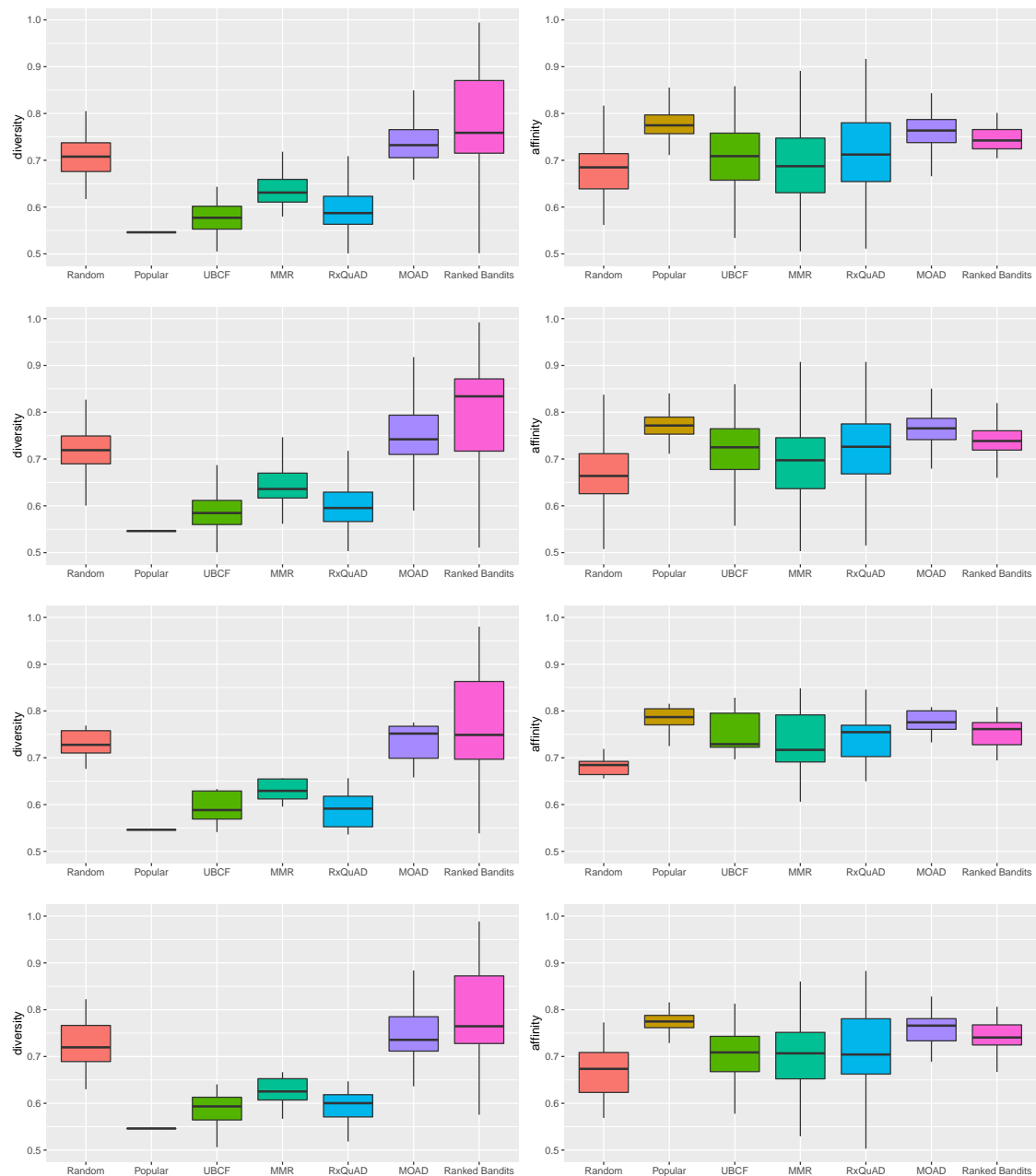
Em cada cenário de propensão, ocorre uma divisão dos metadados de artistas em dois grupos, de acordo com a sua configuração. Os algoritmos *baseline* ao serem executados receberam apenas os aspectos a serem diversificados, uma vez que nenhum deles faz divisão de aspectos e não se desejava induzi-los a diversificar os aspectos configurados para manterem similaridade com o histórico do usuário. O MOAD e os *Ranked Bandits* receberam todos os aspectos e realizaram a avaliação de cada objetivo no grupo de aspectos apropriado.

Por exemplo, considere-se um usuário cuja propensão à diversidade coincida com o cenário 5 da tabela 6.2. Todos os *baselines* executarão sobre os aspectos Contemporaneidade e Localidade (anos de lançamento do primeiro e do último álbum do artista e país de origem). O MOAD e o *Ranked Bandits* usarão os mesmos aspectos nas comparações entre os itens realizadas para verificação do objetivo *Diversidade*, usando o aspecto restante (Gênero musical) para as comparações entre itens da recomendação e itens do histórico do usuário feitas para o objetivo *Afinidade*.

Por fim, com o intuito de verificar os resultados obtidos pelas abordagens propostas em frente aos algoritmos *baseline*, foi feito um teste estatístico para verificar a significância da diferença dos resultados obtidos em relação a cada *baseline*. Por conta da não normalidade dos dados, o teste de Wilcoxon foi realizado, de forma pareada, comparando então os resultados de um mesmo usuário nos diferentes algoritmos.

Uma vez que diferentes cenários de propensão produzem resultados de Diversidade e Afinidade usando métricas diferentes, obtidas sobre grupos de aspectos não coincidentes, há então diferenças nas grandezas dos valores obtidos, fazendo com que não haja sentido em se comparar resultados de usuários com diferentes propensões à diversidade. Os resultados obtidos foram então separados por cenário, sendo então reportados os cenários 3, 4, 6 e 7, aqueles que foram obtidos a partir de um número significativo de usuários da amostra, conforme indicado na Figura 6.2. Resultados para o cenário 1 não são exibidos porque os *baselines* não foram executados para usuários neste cenário, uma vez que não há atributos a serem diversificados neste cenário.

Figura 7.2: Resultados para métricas Diversidade e Afinidade, todos os algoritmos, cenários 3, 4, 6 e 7.



Fonte: Autoria própria.

### 7.5.1 Resultados MOAD

A Figura 7.2 mostra graficamente um comparativo dos resultados obtidos pelos modelos propostos e pelos *baselines*. Para facilitar a visualização, o eixo Y está limitado ao intervalo  $[0.5, 1]$  e os *outliers* foram removidos. Observa-se o desempenho superior aos *baselines* do MOAD em todos os cenários na métrica *Diversidade*, verificado com testes de Wilcoxon. Cabe ressaltar que, naturalmente, o *baseline* Aleatório é um concorrente forte em diversidade, pois tende a ser uma amostra representativa dos dados como um todo.

Na métrica *Afinidade*, ainda que não obtenha resultados superiores em todos os cenários, está sempre próximo dos valores mais elevados, mostrando que a abordagem proposta com o uso de algoritmos evolucionários possibilita a busca de valores ótimos simultaneamente em dois objetivos antagônicos. Os resultados dos testes de Wilcoxon com opção *greater* do MOAD em relação aos demais algoritmos são mostrados nas Tabelas 7.5 e 7.6, onde o p-valor abaixo de 0,05 indica a rejeição da hipótese nula, de que não há diferença significativa entre os resultados, prevalecendo a hipótese de que o MOAD mostra resultados superiores. Os resultados destacados em negrito correspondem a esta situação.

Tabela 7.5: *p*-valor do teste de Wilcoxon com opção *greater* na métrica *Diversidade*, favorável ao MOAD, para diferentes cenários de recomendação.

	Random	Popular	UBCF	MMR	RxQuAD	Ranked Bandits
Cenário 3	<b>2.2e-16</b>	<b>2.2e-16</b>	<b>2.2e-16</b>	<b>2.2e-16</b>	<b>2.2e-16</b>	0.1561
Cenário 4	<b>1.223e-06</b>	<b>6.059e-13</b>	<b>2.2e-16</b>	<b>2.2e-16</b>	<b>2.486e-11</b>	0.1161
Cenário 6	<b>2.032e-12</b>	<b>2.2e-16</b>	<b>2.2e-16</b>	<b>2.2e-16</b>	<b>2.2e-16</b>	0.261
Cenário 7	<b>2.2e-16</b>	<b>2.2e-16</b>	<b>2.2e-16</b>	<b>3.211e-15</b>	<b>2.2e-16</b>	0.2294

Fonte: Autoria própria.

Tabela 7.6: *p*-valor do teste de Wilcoxon com opção *greater* na métrica *Afinidade*, favorável ao MOAD, para diferentes cenários de recomendação.

	Random	Popular	UBCF	MMR	RxQuAD	Ranked Bandits
Cenário 3	<b>0.0008371</b>	0.7561	<b>0.04687</b>	<b>0.02513</b>	0.05059	0.08269
Cenário 4	<b>2.341e-14</b>	0.8413	<b>7.52e-05</b>	<b>4.619e-08</b>	<b>0.0002945</b>	<b>7.966e-06</b>
Cenário 6	<b>2.2e-16</b>	0.9999	<b>2.2e-16</b>	<b>2.2e-16</b>	<b>2.2e-16</b>	0.2498
Cenário 7	<b>2.2e-16</b>	0.2274	0.368	<b>6.903e-05</b>	<b>0.002699</b>	0.1371

Fonte: Autoria própria.

A grande maioria dos resultados em *Diversidade* mostra um p-valor inferior a 0,05, in-

dicando que os resultados obtidos pelo MOAD são significativamente melhores do que os obtidos por todos os *baselines*. Nas comparações em relação ao *Ranked Bandits*, não foi possível rejeitar a hipótese nula em nenhum dos cenários analisados, de forma que não há resultado superior entre esses dois algoritmos.

No caso do objetivo *Afinidade*, o MOAD obteve resultados inferiores ao *baseline* Mais Popular. Os resultados obtidos pelo *baseline* Mais Popular reforçam que os objetivos analisados são antagônicos, pois, como esperado, há baixa Diversidade neste *baseline*, enquanto há alta Afinidade. Uma explicação para estes resultados em Afinidade pode ser o fato de que há, no Last.fm, uma predominância de usuários com preferência pelo gênero Rock e seus derivados, bem como de artistas ingleses e americanos.

O MOAD foi, entretanto, superior na grande maioria dos outros cenários, mas com resultados bem próximos àqueles que o teste de Wilcoxon não indicaram sua superioridade.

Pode-se concluir que, ao conseguir resultados superiores a todos os *baselines* em *Diversidade* e, no mínimo, competitivos em *Afinidade*, que o MOAD atingiu o objetivo de otimizar simultaneamente as duas métricas. Mesmo no caso do *baseline* Mais Popular, que conseguiu melhores resultados em *Afinidade*, o MOAD foi superior considerando-se os objetivos de maneira agregada, devido aos baixos resultados de *Diversidade* do *Mais Popular*, o que condiz com o esperado pra este *baseline*.

De maneira geral, verifica-se que, além do sucesso obtido pelo MOAD em otimizar ambas as métricas, há pouca variabilidade nos resultados, o que pode ser verificado pelos *box-plots* estreitos, indicando que todos os usuários recebem recomendações com bons índices de Diversidade e Afinidade.

### 7.5.2 Resultados Ranked Bandits

De maneira semelhante ao MOAD, o modelo *Ranked Bandits* também superou em *Diversidade* todos os *baselines*, em todos os cenários, como pode ser verificado na Figura 7.2 e na Tabela 7.7, com todos os resultados de p-valor abaixo de 0,05, indicando a rejeição da hipótese nula de que os resultados dos *Ranked Bandits* não seriam superiores. Ainda considerando a Tabela 7.7, verifica-se que assim como o teste de Wilcoxon não rejeita a hipótese nula que permite admitir superioridade do MOAD em relação aos *Ranked Bandits*, o mesmo ocorre com a suposição inversa, mostrando que os resultados de ambos os modelos propostos

são equivalentes.

Em relação a *Afinidade*, conforme reportado na Tabela 7.8, os resultados do modelo *Ranked Bandits* não se mostraram superiores aos dos *baselines*, com exceção do *Random*. Entretanto, observando graficamente na Figura 7.2, os resultados mostram-se próximos aos dos *baselines*.

Em uma comparação direta do MOAD com os *Ranked Bandits*, embora em nenhum cenário se possa dizer que os resultados de um modelo são superiores aos do outro, o que ocorre tanto para *Diversidade* como para *Afinidade*, nota-se que há uma ligeira vantagem em *Diversidade* para o *Ranked Bandits*, assim como o há em *Afinidade* para o MOAD. Isso se explica pelo fato de que, enquanto o MOAD considera os dois objetivos individualmente, considerando a dominância entre as soluções, o *Ranked Bandits* considera os dois objetivos linearmente escalarizados. Como as métricas de *Diversidade* atingem valores absolutos ligeiramente maiores, o modelo de *Ranked Bandits* tende a optar por soluções que privilegiem este objetivo, que resultam em um valor escalarizado maior. Substituindo-se a função linear de escalarização, pode-se encontrar resultados diferentes nesse sentido. Observa-se entretanto que, de maneira geral, os dois modelos apresentaram resultados que permitem concluir que ambos os objetivos são otimizados.

A característica de *Ranked Bandits* que o destaca do MOAD é o tempo de processamento. Na execução da função multiobjetivo para uma lista de recomendação, itens presentes na lista devem ser comparados usando-se as funções de similaridade da seção 5.2.2, ou dois a dois, no caso do objetivo *Diversidade*, ou com os itens no histórico do usuário, no caso de *Afinidade*. Essas comparações podem requerer um grande poder de processamento, para um grande número de usuários.

No caso do MOAD, executado com população de tamanho 10, com 20 gerações, tem-se um total de 400 chamadas à função multiobjetivo por usuário, enquanto com *Ranked Bandits*, com bons resultados sendo exibidos com 100 iterações, são 100 chamadas à função multiobjetivo realizadas. Com isso, tem-se um modelo que requer menos tempo de execução, capaz de reagir rapidamente a mudanças no interesse do usuários, notadamente a mudanças no perfil de propensão à diversidade. Isso o torna um modelo adequado para a realização de recomendações online, conforme discutido no capítulo 8, onde será demonstrada a diferença esforço computacional entre os dois modelos.

Tabela 7.7: *p*-valor do teste de Wilcoxon com opção *greater* na métrica *Diversidade*, favorável aos Ranked Bandits, para diferentes cenários de recomendação.

	Random	Popular	UBCF	MMR	RxQuAD	MOAD
Cenário 3	<b>2.2e-16</b>	<b>2.2e-16</b>	<b>2.2e-16</b>	<b>2.2e-16</b>	<b>2.2e-16</b>	0.8441
Cenário 4	<b>2.2e-16</b>	<b>2.2e-16</b>	<b>2.2e-16</b>	<b>2.2e-16</b>	<b>2.2e-16</b>	0.884
Cenário 6	<b>0.001454</b>	<b>5.879e-10</b>	<b>2.472e-11</b>	<b>6.354e-08</b>	<b>2.508e-10</b>	0.7429
Cenário 7	7.2	7.2	7.2	7.2	7.2	0.7717

Fonte: Autoria própria.

Tabela 7.8: *p*-valor do teste de Wilcoxon com opção *greater* na métrica *Afinidade*, favorável aos Ranked Bandits, para diferentes cenários de recomendação.

	Random	Popular	UBCF	MMR	RxQuAD	MOAD
Cenário 3	<b>0.003549</b>	0.9849	0.2464	0.1125	0.1701	0.9175
Cenário 4	<b>3.816e-07</b>	1	0.6314	<b>0.01305</b>	0.4011	1
Cenário 6	<b>0.02051</b>	0.7832	0.5749	0.4946	0.559	0.7544
Cenário 7	<b>0.02834</b>	0.968	0.2544	0.21	0.3445	0.8638

Fonte: Autoria própria.

### 7.5.3 Exemplo de Recomendação

Nesta seção tem-se uma recomendação de exemplo, para um usuário real da base experimental do Last.fm. Foram geradas recomendações em quatro algoritmos em uma simulação do cenário 3 da tabela 6.2, ou seja, o sistema estaria configurado para diversificar o aspecto *Localidade* e conservar os demais aspectos, como faria um usuário interessado em ouvir artistas de diferentes países dentro de suas preferências musicais para os demais aspectos.

A Tabela 7.9 mostra as recomendações obtidas nas simulações executadas. O UBCF, que não leva em conta a necessidade de diversificação, recomendou artistas de 5 países diferentes. O MMR, que recebeu os metadados relativos a *Localidade* para diversificação, recomenda artistas de 6 países diferentes. O MOAD e o *Ranked Bandits* recomendam, respectivamente, artistas de 9 e 8 países para o usuário exemplificado. Em termos de ganho percentual, MOAD e o *Ranked Bandits* trazem um ganho de diversidade entre 33% e 50%, em relação ao MMR.

### 7.5.4 Benefícios Colaterais

As abordagens MOAD e *Ranked Bandits* foram desenvolvidas para maximizar os objetivos *Diversidade* e *Afinidade*, permitindo a obtenção de uma recomendação diversa, mas que

mantivesse proximidade com itens presentes no histórico do usuário, aumentando as chances de que, apesar de diversas, as recomendações sejam relevantes para o usuário. Entretanto, verificou-se que tais abordagens indicaram ganhos também em relação a outras métricas analisadas.

### Novidade

Uma dos efeitos observados em ambas as abordagens propostas é a presença de Novidade na recomendação. Entende-se por novidade a recomendação de itens não familiares ao usuário, ou seja, artistas previamente desconhecidos, mas que encaixam nas preferências musicais do usuário.

Uma métrica de novidade, *Mean Self-information* (MSI), proposta por Zhou et al. (2010) e usada em vários outros trabalhos (VARGAS; CASTELLS, 2011; ZHANG et al., 2012), considera como novidade itens posicionados na cauda longa, ou seja, artistas escutados por menores quantidades de usuários. Para tanto, vale-se do índice de popularidade do artista  $i$  ( $p_i$ ), que é a razão entre o número de usuários que escutaram o artista e o número total de usuários. Assim, o índice MSI é calculado pela Equação 7.2.

$$MSI = -\frac{1}{|R|} \sum_{i \in R} \log_2 p_i \quad (7.2)$$

onde  $R$  é a lista de artistas recomendados.

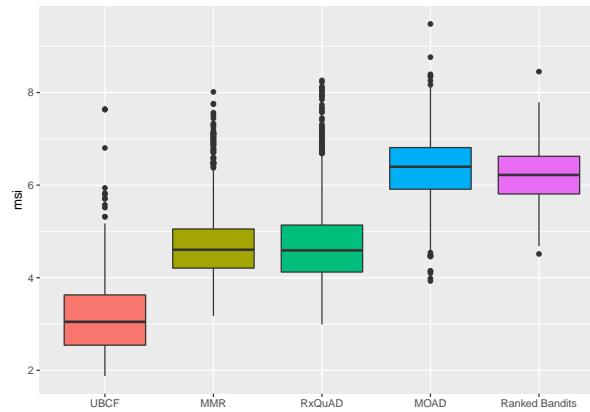
A Figura 7.3 mostra as distribuições de valores de MSI alcançados para cada usuário, usando as abordagens propostas e os *baselines*. Tanto o MOAD como os *Ranked Bandits* alcançaram valores substancialmente mais altos de MSI que os *baselines*, indicando que são capazes de introduzir novos itens aos usuários, ainda que respeitando suas propensões de diversidade e levando em consideração a proximidade com o histórico do usuário, apesar da filtragem inicial de artistas a serem recomendados, que já retira artistas da cauda longa.

### Acurácia

Utilizando-se da divisão dos históricos de usuários em partições de treino e teste, foi possível verificar a acurácia de cada um dos algoritmos, através da relação de itens recomendados presentes no conjunto de testes do usuário. A métrica de acurácia é definida como sendo a



Figura 7.3: Resultados de MSI de todos os usuários da amostra testada.

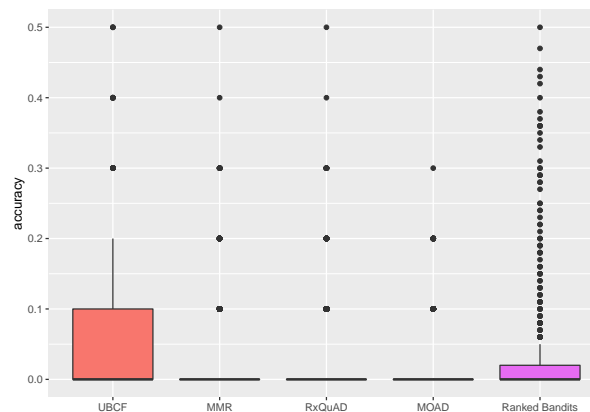


Fonte: Autoria própria.

razão entre o número de itens recomendados identificados no conjunto de testes e o número total de itens recomendados.

Considerando-se para o *Ranked Bandits* a média das últimas 50 recomendações, tem-se o resultado mostrado na Figura 7.4. Percebe-se um incremento na acurácia obtido pela abordagem *Ranked Bandits* em relação a outros algoritmos que buscam diversificar seus resultados. Apenas o UBCF obteve maior acurácia, sendo o único dos algoritmos testados que não diversifica sua recomendação.

Figura 7.4: Resultados de acurácia de todos os usuários da amostra testada.



Fonte: Autoria própria.

## 7.6 Considerações Finais

Neste capítulo foram apresentados o modelo MOAD e o *Ranked Bandits*, abordagens multi-objetivo para sistemas de recomendação musical com foco em diversificação e minimizando perda de acurácia. Foram discutidos os procedimentos a avaliação dos experimentos realizados. Os resultados obtidos foram exibidos juntamente com uma discussão acerca destes resultados, ressaltando situações em que um modelo mostra-se mais vantajoso que o outro.

Tabela 7.9: Top-10 recomendações para um usuário real do Last.fm.

Artista		Inglaterra	Sérvia	África do Sul	EUA	Irlanda do Norte	França	Suécia	Alemanha	Canadá	Finlândia	País de Gales	Polônia	Irlanda	Brasil	Itália	Chile	Cuba
UBCF	Pink Floyd	X																
	Bajaga i Instruktori		X															
	Die Antwoord			X														
	R.E.M.				X													
	System of a Down				X													
	Radiohead	X																
	Depeche Mode	X																
	Buckcherry				X													
	Therapy?					X												
	Judas Priest	X																
MMR	R.E.M.				X													
	Buckcherry				X													
	Therapy?					X												
	Between the Buried and Me				X													
	Kunt and the Gang	X																
	The Faceless				X													
	Benighted						X											
	Crucified Barbara							X										
	Anti-Nowhere League	X																
	Moderat								X									
MOAD	Arcade Fire									X								
	Scandinavian Music Group										X							
	Los Campesinos!											X						
	Crucified Barbara							X										
	Farben Lehre												X					
	Aphex Twin													X				
	Nação Zumbi														X			
	Jaya the Cat				X													
	Die Toten Hosen								X									
	Alghazanth										X							
Ranked Bandits	Nothingface				X													
	Leftöver Crack				X													
	Raul Seixas														X			
	Dubmood							X										
	Peter Gabriel	X																
	Andrea Bocelli															X		
	Poema Arcano																X	
	Paatos							X										
	Silvio Rodríguez																	X
	Mono Inc.								X									

Fonte: Autoria Própria.

## Capítulo 8

### Avaliação Online

Neste capítulo, aborda-se a avaliação dos modelos de recomendação multiobjetivo propostos de maneira online, ou seja, realizando recomendações em períodos de tempo subsequentes, levando em consideração o *feedback* do usuário ao aprovar ou não as recomendações, ao mesmo tempo em que novas recomendações se adaptam a mudanças de interesse do usuário em termos dos aspectos que ele deseja diversificar ou conservar. Esse tipo de avaliação de assemelha em maior grau a um cenário realístico de operação de um sistema de recomendação.

Para a realização da avaliação online, num cenário ideal, ter-se-ia acesso, além do histórico de itens consumidos por um usuário, às reações do usuário à medida que novos itens vão sendo a ele recomendados. Por reações, entende-se as interações dos usuários com os itens recomendados, por exemplo, ouvindo-o por inteiro, parcialmente ou por poucos segundos, ignorá-lo por completo ou marcá-lo como interessante ou desinteressante (através de "curtidas" ou algo equivalente). Para se ter acesso a tais reações, é necessário que se tenha um serviço através do qual o usuário acesse itens, além de se ter registro não só do histórico de itens consumidos, mas dessas reações à medida em que elas acontecem.

Em face à dificuldade de se ter acesso ao registro de atividades de usuários em um serviço de músicas nos termos apresentados, muitos trabalhos realizam experimentos que simulam a existência de interações entre o sistema de recomendação analisado e os usuários testados. Na seção 8.1, será descrita a maneira como o histórico completo de usuários foi trabalhado para simular a operação de um sistema de recomendação sobre um serviço online de acesso a itens musicais.

## 8.1 Protocolo de Avaliação Online de Recomendações

O protocolo de avaliação executado nesta simulação consiste em realizar as etapas descritas no modelo retratado na figura 5.4. As subseções a seguir mostram como isso foi feito.

### 8.1.1 Divisão do Histórico

Cada registro do histórico do usuário é composto de um artista que foi escutado em um determinado instante de tempo, medido em segundos decorridos a partir de um marco inicial. Convertendo-se esse instante para dias decorridos a partir do mesmo marco inicial, pode-se facilmente agrupar os itens nos históricos do usuários por dias.

A escolha de um dia como grão para a realização de recomendações baseia-se no senso de que, utilizando-se um grão menor, há uma maior chance de concentrar itens consumidos sob um interesse comum do usuário, ou seja, ao longo de dias diferentes há uma maior chance de mudança no interesse do usuário no que diz respeito a diversificação de aspectos musicais. Uma possibilidade alternativa seria a utilização de uma técnica de identificação de sessões; essa configuração não foi conduzida neste experimento, entretanto, pelo fato de que a existência de múltiplas sessões diárias resultaria num aumento substancial dos requerimentos de memória e processamento.

### 8.1.2 Execução

Nesta avaliação, executou-se a abordagem de *Ranked Bandits* com os valores de parâmetros definidos na Tabela 8.1. O valor de  $\epsilon$  foi mantido em relação à avaliação offline, uma vez que testes demonstraram que este é o valor que resulta no melhor equilíbrio *exploit x explore*. Quanto aos parâmetros que determinam número de iterações, optou-se por valores baixos, com  $\gamma = 100$  iterações para cenários processados pela primeira vez, o que garante resultados semelhantes aos conseguidos com até 500 iterações, conforme Figura 7.1; e  $\theta = 1$  iteração em cenários previamente processados. O intuito ao selecionar estes valores foi mostrar um quadro em que pouco processamento é requerido para atingir resultados satisfatórios.

Tabela 8.1: Valores dos parâmetros usado na avaliação online do *Ranked Bandits*

Parâmetro	Valor
$\epsilon$	0,1
$\gamma$	100
$\theta$	1

Fonte: Autoria própria.

### 8.1.3 *Baselines*

Como *baselines* para esta avaliação, foram executados os modelos MMR e RxQuAD, que são os modelos que realizam diversificação já usados no Capítulo 7. Os atributos passados para ambos os modelos são os referentes aos aspectos a serem diversificados, de acordo com o cenário de propensão estabelecido através do PD-ILD, mesma informação fornecida para o modelo *Ranked Bandits*. Da mesma forma, quando se gera uma recomendação para determinado dia, os dados referentes ao consumo do usuário em dias anteriores ao que se está gerando a recomendação são considerados histórico do usuário.

O modelo MOAD não foi executado nesta avaliação, uma vez que sua execução neste protocolo de recomendação online não é viável. Para fornecer uma recomendação, o MOAD precisa executar por completo, o que tem que se repetir em cada dia no histórico do usuário. Cada execução do MOAD com 20 gerações de 10 indivíduos requer 400 execuções da função multiobjetivo, que é a operação mais custosa computacionalmente do processo, considerando que o seu tempo médio de execução fica entre 0,2 e 0,3 segundo. Para analisar o custo computacional, será utilizado o pior caso dentre os 997 usuários da base de dados usada para a avaliação, o usuário com identificador 3423731, com um histórico de 1.697 dias de uso. Os dados de tempo foram computados usando um mesmo computador para ambos os modelos, sem processamento paralelo. Dados comparativos entre os modelos MOAD e *Ranked Bandits* podem ser vistos na Tabela 8.2.

Tabela 8.2: Comparativo de custos computacionais entre os modelos MOAD e *Ranked Bandits* para pior caso.

	MOAD	<i>Ranked Bandits</i>
Nº de execuções da função multiobjetivo	678.800	2.489
Tempo de execução	$\cong$ 20 horas	12,5 minutos

Fonte: Autoria própria.

## 8.2 Resultados

Foram executados então os *baselines* citados na seção 8.1.3 e o *Ranked Bandits*. Nestes resultados há múltiplos usuários, com recomendações para cada um de seus dias de consumo (com exceção do primeiro dia). Os usuários possuem históricos com diferentes quantidades de dias, situados em diferentes momentos na linha do tempo. Com o intuito de comparar os resultados desses usuários, suas linhas temporais foram normalizadas para a faixa  $[0, 1]$ , sendo 0 o primeiro dia e 1 o último dia que consta no seu histórico. Essas linhas temporais foram divididas em 20 partes, de forma que se pode verificar a evolução dos resultados dos modelos ao longo do tempo em períodos relativamente equivalentes para todos os usuários.

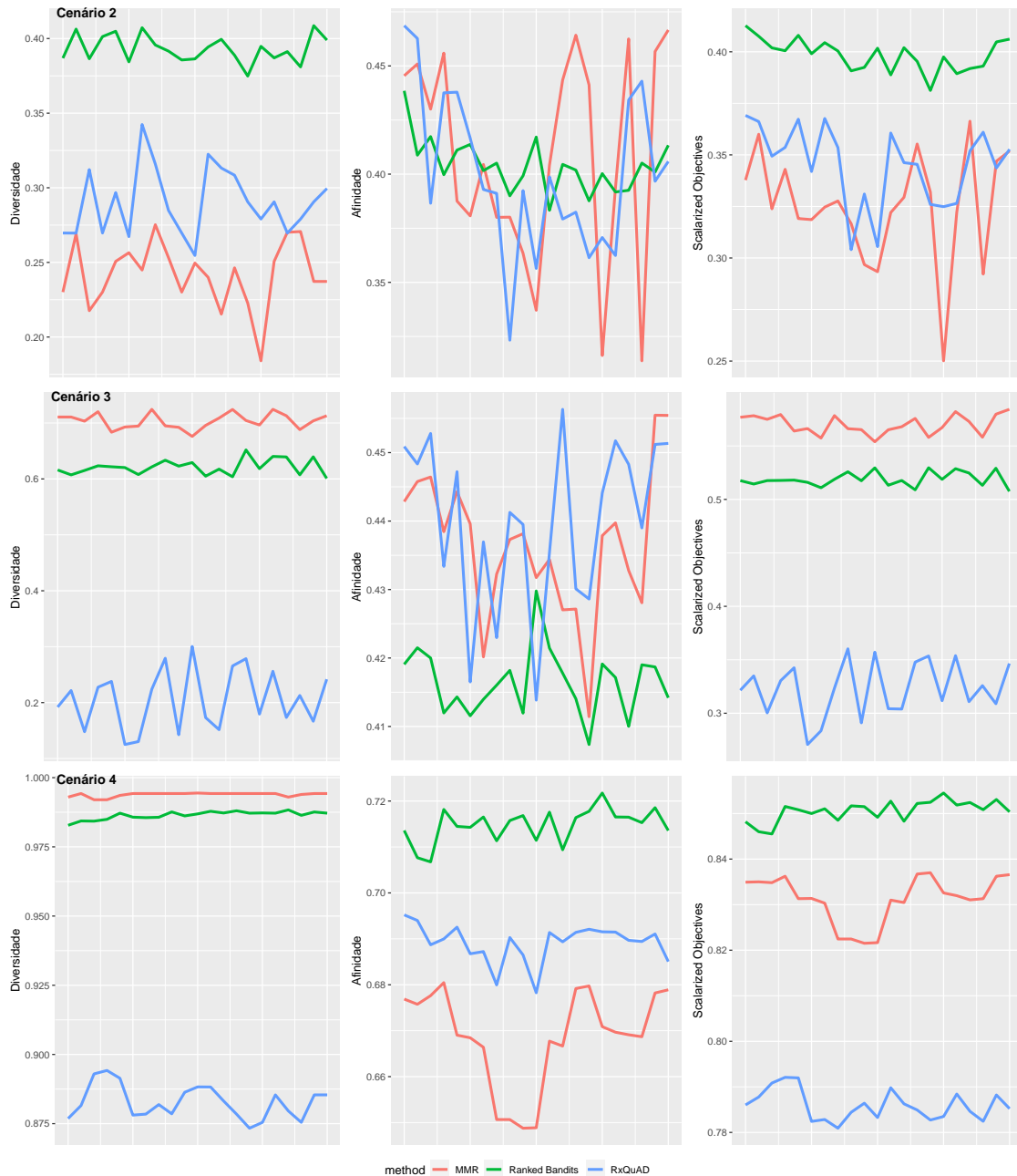
Quando um usuário demonstra diferentes propensões à diversidade ao longo dos dias, diferentes aspectos são usados para a determinação dos objetivos *Diversidade* e *Afinidade*, o que resulta em resultados de grandezas diferentes. Por exemplo um usuário no cenário 2, ou seja diversificando o aspecto Contemporaneidade, pode receber uma recomendação com valor de Diversidade 0,7. Se este mesmo usuário mudar para o cenário de propensão 4, passando a diversificar o aspecto Gênero Musical, ele pode receber uma recomendação com valor de Diversidade 0,9. Isso não significa que o modelo teve melhor performance no segundo caso que no primeiro, mas que as recomendações realizaram comparações sobre conjuntos de metadados diferentes, o que possibilita a obtenção de valores ótimos em diferentes faixas. Não há sentido em comparar, então, resultados de dias em que a propensão à diversidade são diferentes. Por isso, a exibição de resultados foi feita separando-se os dias pelo cenário de propensão à diversidade apresentados, conforme a Tabela 6.2.

A Figura 8.1 mostra os resultados para os cenários de diversificação com um aspecto a ser diversificado, enquanto a Figura 8.2 mostra resultados em cenários em que mais de um aspecto são diversificados (essa separação foi feita apenas porque não foi possível condensar todos os resultados em uma página). Como em cada ponto da linha do tempo há 997 resultados de usuário a serem exibidos, para que o gráfico não tenha excesso de informações e tenha sua compreensão dificultada, as medianas dos resultados formam a linha mostrada ao longo do gráfico. Ressalte-se que, no caso do cenário 8, não há resultados a serem exibidos para afinidade, uma vez que todos os aspectos são usados para a métrica de diversidade.

Além dos gráficos dos objetivos *Diversidade* e *Afinidade*, foi incluído um terceiro gráfico

com o valor escalarizado dos objetivos. Assim como na avaliação offline, foi utilizada a média aritmética como função de escalarização. A partir deste gráfico pode-se verificar o ganho proporcionado ao se maximizar simultaneamente ambos os objetivos.

Figura 8.1: Resultados para recomendação online em cenários com diversificação de um aspecto (resultados do *Ranked Bandits* em verde).



Fonte: Autoria própria.

De maneira geral, o que se pode perceber é um ganho do modelo *Ranked Bandits* em relação aos baselines na maioria dos cenários. O comportamento mais comumente observado



foi o *Ranked Bandits* atingindo um valor destacado em *Diversidade*. Em alguns cenários, *Ranked Bandits* mantém desempenho competitivo em relação aos *baselines* em *Afinidade*, o que o deu vantagem quando aglomerados os objetivos, caso dos cenários 2 e 7. No cenário 6, entretanto, *Ranked Bandits* obtiveram larga vantagem nos dois objetivos. Essa vantagem é observada em *Diversidade* no cenário 8, único objetivo observado para este cenário.

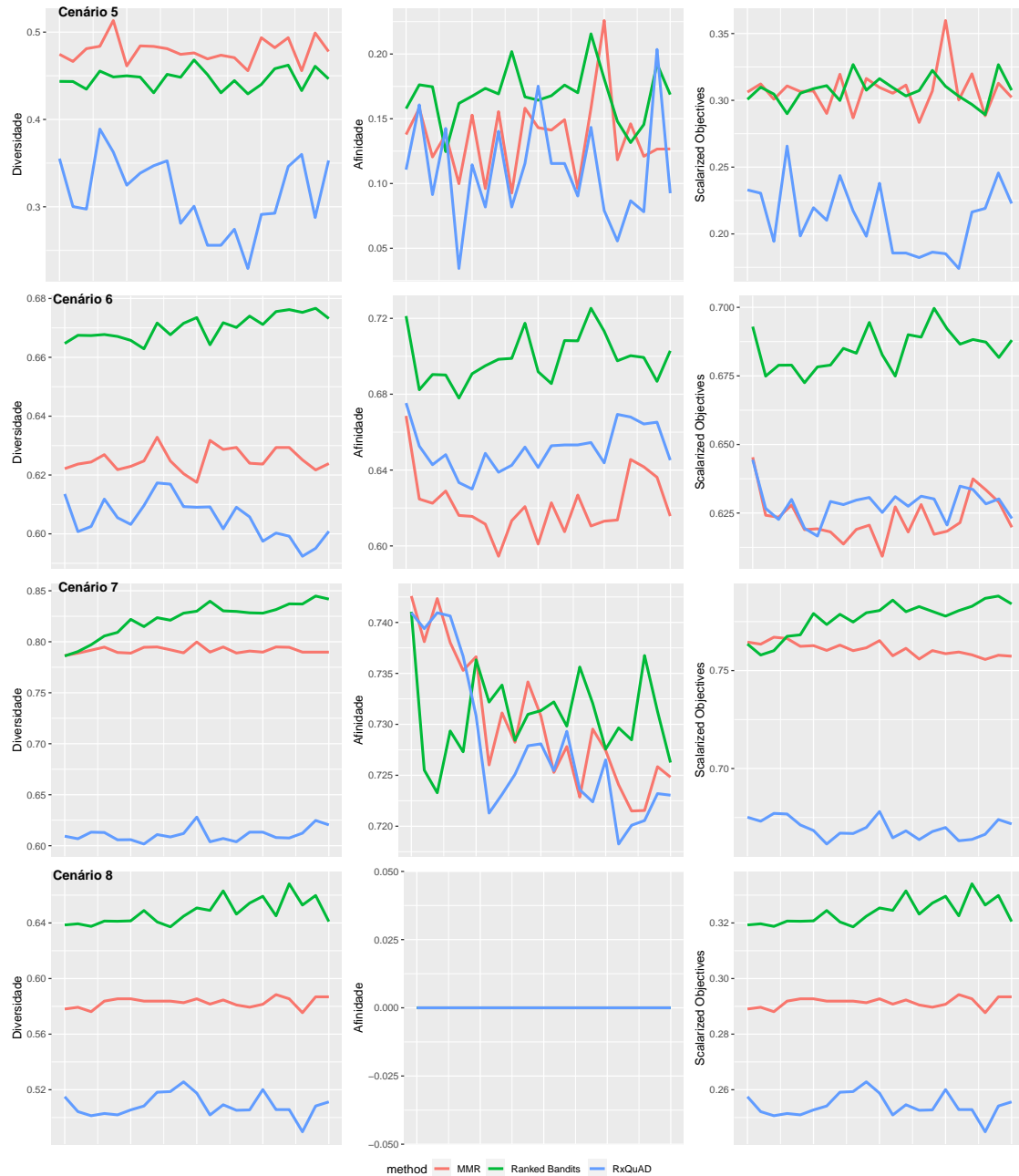
Cabe destacar o desempenho superior em *Diversidade* do modelo MMR nos cenários 3 e 4. Tratam-se de cenários em que apenas um aspecto é diversificado, sendo aspectos compostos de atributos discretos (Localidade e Gênero musical), o que faz com que o coeficiente de similaridade de Jaccard usado pelo modelo como função de similaridade tenha bom desempenho. Entretanto, o que se observa no cenário 4 é que, trabalhando próximo do valor ótimo de diversidade no cenário 4, o modelo *Ranked Bandits* conseguiu se destacar em *Afinidade*, o que demonstra que, por se tratarem de objetivos antagônicos, o desempenho em um dos objetivos é limitado pela necessidade de desempenhar bem o outro objetivo.

No cenário 5, cabe observar que o modelo *Ranked Bandits* desempenha melhor no objetivo *Afinidade*, o que faz com que o seu desempenho ao considerar ambos os objetivos seja semelhante ao do MMR. Um teste de Wilcoxon com opção *two sided* obteve p-valor igual a 0.7859, não rejeitando a hipótese nula de que as distribuições seriam iguais. Este teste mostrou que não há diferença entre os resultados obtidos pelos dois modelos nos objetivos agregados.

A Tabela 8.3 mostra os resultados do teste de Wilcoxon em favor do *Ranked Bandits*, testando-se sempre as medianas da distribuição de resultados de todos os usuários naquele período de tempo. Os destaques em negrito foram aqueles nos quais o teste demonstrou superioridade dos resultados obtidos pelo *Ranked Bandits*.

Outro ganho a se considerar é o fato de que o modelo *Ranked Bandits* possui, na maioria dos cenários, bem menos oscilação ao longo da linha do tempo que o observado nos *baselines*, o que se constitui num ganho colateral significativo, uma vez que as recomendações realizadas mantêm o nível de adequação e diversidade com o decorrer dos dias.

Figura 8.2: Resultados para recomendação online em cenários com diversificação de mais de um aspecto.



Fonte: Autoria própria.

Tabela 8.3: *p*-valor do teste de Wilcoxon com opção *greater* na métrica favoráveis ao *Ranked Bandits*, para diferentes cenários de recomendação da avaliação online.

Cenário	MMR			RxQuAD		
	Diversidade	Afinidade	Obj. Escal.	Diversidade	Afinidade	Obj. Escal.
2	<b>4.76e-07</b>	0.730	<b>4.76e-07</b>	<b>4.76e-07</b>	0.341	<b>4.76e-07</b>
3	1	1	1	<b>4.76e-07</b>	0.999	<b>4.76e-07</b>
4	0.999	<b>4.76e-07</b>	<b>4.76e-07</b>	<b>3.20e-05</b>	<b>4.76e-07</b>	<b>4.76e-07</b>
5	1	<b>1.76e-04</b>	0.405	<b>4.76e-07</b>	<b>9.05e-06</b>	<b>4.76e-07</b>
6	<b>4.76e-07</b>	<b>4.76e-07</b>	<b>4.76e-07</b>	<b>4.76e-07</b>	<b>4.76e-07</b>	<b>4.76e-07</b>
7	<b>9.53e-07</b>	0.205	<b>1.57e-05</b>	<b>4.76e-07</b>	0.0516	<b>4.76e-07</b>
8	<b>4.76e-07</b>	-	<b>4.76e-07</b>	<b>4.76e-07</b>	-	<b>4.76e-07</b>

### 8.3 Considerações Finais

Neste capítulo foi abordada a validação do cenário online de recomendações, visando demonstrar os resultados obtidos em uma situação próxima da real, com recomendações sendo feitas ao longo de dias em que o usuário utiliza o serviço. Pôde-se observar o desempenho satisfatório do modelo *Ranked Bandits*, sendo superior aos *baselines* na maioria dos cenários de propensão à diversidade, principalmente quando considerados simultaneamente os objetivos de Diversidade e Afinidade.

Como o modelo *Ranked Bandits* utiliza informações de suas execuções passadas para cada nova iteração, após um esforço inicial com um número mais alto de iterações para atingir recomendações apropriadas, poucas iterações são exigidas para se manter este desempenho, o que permite que o uso do modelo seja escalado para um número mais alto de usuários.

## Capítulo 9

### Conclusões

Este trabalho consiste de uma abordagem para o desenvolvimento de um sistema de recomendações musicais, na qual considerou-se que a diversidade da lista de recomendação produzida é um objetivo tão importante quanto a relevância dos itens para o usuário. Levando em consideração a possibilidade de que a otimização da diversidade acarrete uma degradação da relevância da lista obtida, utilizou-se uma abordagem multiobjetivo para garantir que ambos os objetivos sejam otimizados simultaneamente.

Foi proposto inicialmente o MOAD, modelo que utilizou um algoritmo evolucionário multiobjetivo para a geração de recomendações de artistas musicais. Com o intuito de comparar itens entre si para a geração dessas recomendações, foram utilizados metadados acerca dos artistas, tendo sido estes metadados divididos em aspectos musicais, com o intuito de capturar características que formam o gosto musical do usuário e possibilitar que a diversificação fosse aplicada especificamente aos aspectos que interessam ao usuário.

Experimentos realizados com o MOAD e algoritmos de recomendação diversificada da literatura atestaram que a solução deste trabalho mostrou-se competitiva em todos os cenários avaliados, tendo obtido resultados superiores ao menos em um dos objetivos em cada cenário, ou em ambos os objetivos em alguns deles.

Entretanto, o MOAD não se mostrou adequado à realização de recomendações online, aplicação que se aproxima de um cenário mais realístico de um sistema de recomendações. Nesse sentido, foi proposto o modelo *Ranked Bandits*. Este modelo é baseado em Multi-Armed Bandits, porém, adaptado para fornecer não um item recomendado, mas uma lista de itens, o que permite uma avaliação das recomendações quanto à diversidade. Seguindo a

motivação exposta neste trabalho, os *Ranked Bandits* foram adaptados para a realização de recomendação multiobjetivo.

O modelo *Ranked Bandits* se mostrou competitivo em relação aos *baselines* testados, obtendo resultados superiores na maioria dos cenários quando agregados os dois objetivos buscados. O modelo mostrou capacidade de adaptação a mudanças de interesse do usuário no que se refere a diversificação em aspectos, mantendo recomendações que atingem valores elevados nas métricas analisadas de maneira constante.

Com os modelos desenvolvidos, pode-se dizer que os objetivos propostos neste trabalho foram alcançados. Mostrou-se que existe variação de interesse dos usuários no sentido de diversificar suas recomendações, interesses que foram levados em consideração a cada recomendação diária gerada, com o modelo se adaptando em um tempo de execução que permite escalar o modelo para um grande número de usuários.

Ambos os modelos, entretanto, são dependentes da existência de metadados acerca dos artistas, para a formação dos aspectos, o que pode ser uma limitação, principalmente se tratando de artistas menos conhecidos. Outra limitação, bastante comum em sistemas de recomendação, é a dependência de históricos dos usuários, fazendo com que outras estratégias de recomendação precisem ser utilizadas para novos usuários, em um problema conhecido como *cold start*.

## 9.1 Trabalhos Futuros

Este trabalho desperta várias possibilidades de continuação, no sentido de aprimorar seus resultados ou até mesmo de acrescentar novas funcionalidades. De maneira geral, assim como os modelos propostos foram aplicados no domínio musical, com algumas adaptações pode-se estender os modelos a outros domínios, especificamente gerando novas funções de similaridades, apropriadas aos aspectos de interesse do domínio em questão. Ainda no domínio musical, os modelos propostos podem ser ampliados com a inclusão de novos objetivos, enquanto a validação destes modelos pode ser enriquecida com a aplicação de novas métricas.

Entre as tentativas de aprimorar os resultados alcançados, uma adaptação de modelos mais eficientes de MAB, tais como UCB ou *Thompson Sampling* pode resultar em um mo-

delo mais eficiente. É necessário que estas políticas de seleção de itens sejam adaptadas para *Ranked Bandits* e recompensas não binárias. Primeiros esforços nesse sentido não foram bem sucedidos, mas essa é uma linha de pesquisa bastante promissora no sentido de obter resultados mais apurados.

Pelo fato de que a adaptação de MABs para recomendações multiobjetivo é dependente da escalarização de suas funções objetivo, uma pesquisa acerca de funções de escalarização pode trazer ganhos que permitam uma aproximação à fronteira de Pareto, permitindo uma otimização mais eficiente e equilibrada de ambos os objetivos.

Uma sequência que surge naturalmente no sentido de aplicar os modelos propostos é aplicá-los a um serviço real e submetê-lo a usuários reais, colhendo suas interações com o sistema e utilizando-as como recompensa para seleção de itens. Dados o esforço e custos envolvidos neste tipo de avaliação, preferiu-se neste trabalho avaliar os modelos em cima de métricas, passo que é necessário para que se tenha um modelo sólido, que faça sentido submeter a um experimento envolvendo pessoas. Esta é, então, a sequência esperada deste trabalho.

# Referências Bibliográficas

AGGARWAL, C. C. **Recommender systems**. [S.l.]: Springer, 2016.

AUER, P. Using upper confidence bounds for online learning. In: IEEE. **Proceedings 41st Annual Symposium on Foundations of Computer Science**. [S.l.], 2000. p. 270–279.

AUER, P. Using confidence bounds for exploitation-exploration trade-offs. **Journal of Machine Learning Research**, v. 3, n. Nov, p. 397–422, 2002.

AUER, P.; CESA-BIANCHI, N.; FISCHER, P. Finite-time analysis of the multiarmed bandit problem. **Machine learning**, Springer, v. 47, n. 2-3, p. 235–256, 2002.

BOBADILLA, J.; ORTEGA, F.; HERNANDO, A.; GUTIÉRREZ, A. Recommender systems survey. **Knowledge-Based Systems**, Elsevier B.V., v. 46, p. 109–132, 2013. ISSN 09507051. Disponível em: <<http://dx.doi.org/10.1016/j.knosys.2013.03.012>>.

BORIAH, S.; CHANDOLA, V.; KUMAR, V. Similarity measures for categorical data: A comparative evaluation. In: SIAM. **Proceedings of the 2008 SIAM International Conference on Data Mining**. [S.l.], 2008. p. 243–254.

BRANKE, J. Mcd and multiobjective evolutionary algorithms. In: **Multiple Criteria Decision Analysis**. [S.l.]: Springer, 2016. p. 977–1008.

BREESE, J. S.; HECKERMAN, D.; KADIE, C. Empirical analysis of predictive algorithms for collaborative filtering. In: MORGAN KAUFMANN PUBLISHERS INC. **Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence**. [S.l.], 1998. p. 43–52.

BUCCAPATNAM, S.; ERYILMAZ, A.; SHROFF, N. B. Multi-armed bandits in the presence of side observations in social networks. In: IEEE. **52nd IEEE Conference on Decision and Control**. [S.l.], 2013. p. 7309–7314.

BURKE, R. Hybrid recommender systems: Survey and experiments. **User modeling and user-adapted interaction**, Springer, v. 12, n. 4, p. 331–370, 2002.

CALDEIRA, J.; OLIVEIRA, R. S.; MARINHO, L.; TRATTNER, C. Healthy menus recommendation: optimizing the use of the pantry. In: **Proceedings of the 3rd International Workshop on Health Recommender Systems (HealthRecSys' 18) co-located with the 12th ACM Conference on Recommender Systems (ACM RecSys 2018)(CEUR Workshop Proceedings)**. [S.l.: s.n.], 2018.

- CARBONELL, J.; GOLDSTEIN, J. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In: **ACM. Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval**. [S.l.], 1998. p. 335–336.
- CARON, S.; KVETON, B.; LELARGE, M.; BHAGAT, S. Leveraging side observations in stochastic bandits. **arXiv preprint arXiv:1210.4839**, 2012.
- CECCHINI, R. L.; LORENZETTI, C. M.; MAGUITMAN, A. G.; BRIGNOLE, N. B. Multiobjective evolutionary algorithms for context-based search. **Journal of the Association for Information Science and Technology**, Wiley Online Library, v. 61, n. 6, p. 1258–1274, 2010.
- CELMA, O. **Music Recommendation and Discovery in the Long Tail**. [S.l.]: Springer, 2010.
- CESA-BIANCHI, N.; GENTILE, C.; ZAPPELLA, G. A gang of bandits. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2013. p. 737–745.
- CHAKRAVARTHI, S. **Multi-Armed Bandits 101**. 2019. [Online; posted 16-August-2019]. Disponível em: <<https://medium.com/opex-analytics/multi-armed-bandits-101-6f4ac62b6bd6>>.
- CUI, L.; OU, P.; FU, X.; WEN, Z.; LU, N. A novel multi-objective evolutionary algorithm for recommendation systems. **Journal of Parallel and Distributed Computing**, Elsevier, 2016.
- DEB, K.; PRATAP, A.; AGARWAL, S.; MEYARIVAN, T. A fast and elitist multiobjective genetic algorithm: Nsga-ii. **IEEE transactions on evolutionary computation**, IEEE, v. 6, n. 2, p. 182–197, 2002.
- DRAGONE, P.; MEHROTRA, R.; LALMAS, M. Deriving user-and content-specific rewards for contextual bandits. In: **The World Wide Web Conference**. [S.l.: s.n.], 2019. p. 2680–2686.
- EICHFELDER, G. An adaptive scalarization method in multiobjective optimization. **SIAM Journal on Optimization**, SIAM, v. 19, n. 4, p. 1694–1718, 2009.
- GENG, B.; LI, L.; JIAO, L.; GONG, M.; CAI, Q.; WU, Y. Nnia-rs: A multi-objective optimization based recommender system. **Physica A: Statistical Mechanics and its Applications**, Elsevier, v. 424, p. 383–397, 2015.
- GENTILE, C.; LI, S.; ZAPPELLA, G. Online clustering of bandits. In: **International Conference on Machine Learning**. [S.l.: s.n.], 2014. p. 757–765.
- GUNAWARDANA, A.; SHANI, G. Evaluating recommender systems. In: **Recommender Systems Handbook**. [S.l.]: Springer, 2015. p. 265–308.
- HAHSLER, M. **recommenderlab: A framework for developing and testing recommendation algorithms**. [S.l.], 2015.



- HARIRI, N.; MOBASHER, B.; BURKE, R. Adapting to user preference changes in interactive recommendation. In: **Twenty-Fourth International Joint Conference on Artificial Intelligence**. [S.l.: s.n.], 2015.
- HAVELIWALA, T. H. Topic-sensitive pagerank. In: **ACM. Proceedings of the 11th international conference on World Wide Web**. [S.l.], 2002. p. 517–526.
- HERLOCKER, J. L.; KONSTAN, J. A.; TERVEEN, L. G.; RIEDL, J. T. Evaluating collaborative filtering recommender systems. **ACM Transactions on Information Systems (TOIS)**, v. 22, n. 1, p. 5–53, 2004. ISSN 1046-8188. Disponível em: <<http://portal.acm.org/citation.cfm?id=963770.963772>>.
- JEUNEN, O. Revisiting offline evaluation for implicit-feedback recommender systems. In: **Proceedings of the 13th ACM Conference on Recommender Systems**. [S.l.: s.n.], 2019. p. 596–600.
- KAPOOR, K.; KUMAR, V.; TERVEEN, L.; KONSTAN, J. A.; SCHRATER, P. I like to explore sometimes: Adapting to dynamic user novelty preferences. In: **ACM. Proceedings of the 9th ACM Conference on Recommender Systems**. [S.l.], 2015. p. 19–26.
- KASIMBEYLI, R.; OZTURK, Z. K.; KASIMBEYLI, N.; YALCIN, G. D.; ERDEM, B. I. Comparison of some scalarization methods in multiobjective optimization. **Bulletin of the Malaysian Mathematical Sciences Society**, Springer, v. 42, n. 5, p. 1875–1905, 2019.
- KONSTAN, J. A.; RIEDL, J. Recommender systems: from algorithms to user experience. **User Modeling and User-Adapted Interaction**, Springer, v. 22, n. 1, p. 101–123, 2012.
- KÜÇÜKTUNÇ, O.; SAULE, E.; KAYA, K.; ÇATALYÜREK, Ü. V. Diversified recommendation on graphs: pitfalls, measures, and algorithms. In: **ACM. Proceedings of the 22nd international conference on World Wide Web**. [S.l.], 2013. p. 715–726.
- LACERDA, A. Multi-objective ranked bandits for recommender systems. **Neurocomputing**, Elsevier, v. 246, p. 12–24, 2017.
- LAI, T. L.; ROBBINS, H. Asymptotically optimal allocation of treatments in sequential experiments. **Design of Experiments: Ranking and Selection**, New York: Dekker, p. 127–142, 1984.
- LI, L.; CHU, W.; LANGFORD, J.; SCHAPIRE, R. E. A contextual-bandit approach to personalized news article recommendation. In: **Proceedings of the 19th international conference on World wide web**. [S.l.: s.n.], 2010. p. 661–670.
- LI, S.; KARATZOGLOU, A.; GENTILE, C. Collaborative filtering bandits. In: **Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval**. [S.l.: s.n.], 2016. p. 539–548.
- MARY, J.; GAUDEL, R.; PREUX, P. Bandits and recommender systems. In: **SPRINGER. International Workshop on Machine Learning, Optimization and Big Data**. [S.l.], 2015. p. 325–336.

- MCDONALD, C. **Solving multiarmed bandits: A comparison of epsilon-greedy and Thompson sampling**. 2018. [Online; posted 30-September-2018]. Disponível em: <<https://towardsdatascience.com/solving-multiarmed-bandits-a-comparison-of-epsilon-greedy-and-thompson-sampling-d97167ca9a50>>.
- MCKINNEY, W. Data structures for statistical computing in python. In: WALT, S. van der; MILLMAN, J. (Ed.). **Proceedings of the 9th Python in Science Conference**. [S.l.: s.n.], 2010. p. 51 – 56.
- MCNEE, S. M.; RIEDL, J.; KONSTAN, J. A. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In: ACM. **CHI'06 extended abstracts on Human factors in computing systems**. [S.l.], 2006. p. 1097–1101.
- NOIA, T. D.; OSTUNI, V. C.; ROSATI, J.; TOMEIO, P.; SCIASCIO, E. D. Adaptive diversity in recommender systems. In: **IIR**. [S.l.: s.n.], 2015.
- OLIVEIRA, R. S.; NÓBREGA, C.; MARINHO, L. B.; ANDRADE, N. A multiobjective music recommendation approach for aspect-based diversification. In: **Proceedings of the 18th International Society for Music Information Retrieval Conference**. [S.l.: s.n.], 2017.
- OUNI, A.; KULA, R. G.; KESSENTINI, M.; ISHIO, T.; GERMAN, D. M.; INOUE, K. Search-based software library recommendation using multi-objective optimization. **Information and Software Technology**, Elsevier, v. 83, p. 55–75, 2017.
- PAMPALK, E.; GOTO, M. Musicsun: A new approach to artist recommendation. In: **ISMIR**. [S.l.: s.n.], 2007. p. 101–104.
- PARAMBATH, S. A. P.; USUNIER, N.; GRANDVALET, Y. A coverage-based approach to recommendation diversity on similarity graph. In: ACM. **Proceedings of the 10th ACM Conference on Recommender Systems**. [S.l.], 2016. p. 15–22.
- PEREIRA, B. L.; UEDA, A.; PENHA, G.; SANTOS, R. L.; ZIVIANI, N. Online learning to rank for sequential music recommendation. In: **Proceedings of the 13th ACM Conference on Recommender Systems**. [S.l.: s.n.], 2019. p. 237–245.
- R Development Core Team. **R: A Language and Environment for Statistical Computing**. Vienna, Austria, 2008. ISBN 3-900051-07-0. Disponível em: <<http://www.R-project.org>>.
- RADLINSKI, F.; KLEINBERG, R.; JOACHIMS, T. Learning diverse rankings with multi-armed bandits. In: **Proceedings of the 25th international conference on Machine learning**. [S.l.: s.n.], 2008. p. 784–791.
- RESNICK, P.; VARIAN, H. R. Recommender systems. **Communications of the ACM**, ACM, v. 40, n. 3, p. 56–58, 1997.
- RIBEIRO, M. T.; LACERDA, A.; VELOSO, A.; ZIVIANI, N. Pareto-efficient hybridization for multi-objective recommender systems. In: ACM. **Proceedings of the sixth ACM conference on Recommender systems**. [S.l.], 2012. p. 19–26.

RIBEIRO, M. T.; ZIVIANI, N.; MOURA, E. S. D.; HATA, I.; LACERDA, A.; VELOSO, A. Multiobjective pareto-efficient approaches for recommender systems. **ACM Transactions on Intelligent Systems and Technology (TIST)**, ACM, v. 5, n. 4, p. 53, 2015.

RICCI, F.; ROKACH, L.; SHAPIRA, B. Recommender systems handbook. In: **Recommender systems handbook**. [S.l.]: Springer, 2015.

ROSSUM, G. van. **Python tutorial**. Amsterdam, 1995.

SANTOS, R. L.; MACDONALD, C.; OUNIS, I. Exploiting query reformulations for web search result diversification. In: **ACM. Proceedings of the 19th international conference on World wide web**. [S.l.], 2010. p. 881–890.

SANZ-CRUZADO, J.; CASTELLS, P.; LÓPEZ, E. A simple multi-armed nearest-neighbor bandit for interactive recommendation. In: **Proceedings of the 13th ACM Conference on Recommender Systems**. [S.l.: s.n.], 2019. p. 358–362.

SCHAFER, J. B.; FRANKOWSKI, D.; HERLOCKER, J.; SEN, S. Collaborative filtering recommender systems. In: **The adaptive web**. [S.l.]: Springer, 2007. p. 291–324.

SCHEDL, M. The lfm-1b dataset for music retrieval and recommendation. In: **Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval**. New York, NY, USA: ACM, 2016. (ICMR '16), p. 103–110. ISBN 978-1-4503-4359-6. Disponível em: <<http://doi.acm.org/10.1145/2911996.2912004>>.

SRINIVAS, N.; DEB, K. Multiobjective optimization using nondominated sorting in genetic algorithms. **Evolutionary Computation**, v. 2, n. 3, p. 221–248, 1994.

SUTTON, R. S.; BARTO, A. G. et al. **Introduction to reinforcement learning**. [S.l.]: MIT press Cambridge, 1998. v. 135.

VALKO, M.; MUNOS, R.; KVETON, B.; KOCÁK, T. Spectral bandits for smooth graph functions. In: **International Conference on Machine Learning**. [S.l.: s.n.], 2014. p. 46–54.

VARGAS, S.; CASTELLS, P. Rank and relevance in novelty and diversity metrics for recommender systems. In: **ACM. Proceedings of the fifth ACM conference on Recommender systems**. [S.l.], 2011. p. 109–116.

VARGAS, S.; CASTELLS, P.; VALLET, D. Intent-oriented diversity in recommender systems. In: **Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval**. New York, NY, USA: ACM, 2011. (SIGIR '11), p. 1211–1212. ISBN 978-1-4503-0757-4. Disponível em: <<http://doi.acm.org/10.1145/2009916.2010124>>.

VARGAS, S.; CASTELLS, P.; VALLET, D. Explicit relevance models in intent-oriented information retrieval diversification. In: **Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval**. New York, NY, USA: ACM, 2012. (SIGIR '12), p. 75–84. ISBN 978-1-4503-1472-5. Disponível em: <<http://doi.acm.org/10.1145/2348283.2348297>>.

- VIGLIENSONI, G.; FUJINAGA, I. The music listening histories dataset. In: **ISMIR**. [S.l.: s.n.], 2017. p. 96–102.
- WANG, S.; GONG, M.; LI, H.; YANG, J. Multi-objective optimization for long tail recommendation. **Knowledge-Based Systems**, Elsevier, v. 104, p. 145–155, 2016.
- ZHANG, M.; HURLEY, N. Avoiding monotony: improving the diversity of recommendation lists. In: **ACM. Proceedings of the 2008 ACM conference on Recommender systems**. [S.l.], 2008. p. 123–130.
- ZHANG, Q.; LI, H. Moea/d: A multiobjective evolutionary algorithm based on decomposition. **IEEE Transactions on evolutionary computation**, IEEE, v. 11, n. 6, p. 712–731, 2007.
- ZHANG, Y. C.; SÉAGHDHA, D. Ó.; QUERCIA, D.; JAMBOR, T. Auralist: introducing serendipity into music recommendation. In: **Proceedings of the fifth ACM international conference on Web search and data mining**. [S.l.: s.n.], 2012. p. 13–22.
- ZHONG, H. **Bandit feedback in Classification and Multi-objective Optimization**. Tese (Doutorado) — Ecole centrale de Marseille, 2016.
- ZHOU, T.; KUSCSIK, Z.; LIU, J.-G.; MEDO, M.; WAKELING, J. R.; ZHANG, Y.-C. Solving the apparent diversity-accuracy dilemma of recommender systems. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 107, n. 10, p. 4511–4515, 2010.
- ZHU, Y.; LAN, Y.; GUO, J.; CHENG, X.; NIU, S. Learning for search result diversification. In: **ACM. Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval**. [S.l.], 2014. p. 293–302.
- ZIEGLER, C.-N.; MCNEE, S. M.; KONSTAN, J. A.; LAUSEN, G. Improving recommendation lists through topic diversification. In: **ACM. Proceedings of the 14th international conference on World Wide Web**. [S.l.], 2005. p. 22–32.
- ZITZLER, E.; DEB, K.; THIELE, L. Comparison of multiobjective evolutionary algorithms: Empirical results. **Evolutionary computation**, MIT Press, v. 8, n. 2, p. 173–195, 2000.
- ZITZLER, E.; LAUMANN, M.; THIELE, L. Spea2: Improving the strength pareto evolutionary algorithm. **TIK-report**, Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische Informatik und Kommunikationsnetze (TIK), v. 103, 2001.
- ZITZLER, E.; THIELE, L. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. **IEEE transactions on Evolutionary Computation**, IEEE, v. 3, n. 4, p. 257–271, 1999.
- ZUO, Y.; GONG, M.; ZENG, J.; MA, L.; JIAO, L. Personalized recommendation based on evolutionary multi-objective optimization [research frontier]. **IEEE Computational Intelligence Magazine**, IEEE, v. 10, n. 1, p. 52–62, 2015.