

UNIVERSIDADE FEDERAL DE UBERLÂNDIA  
FACULDADE DE CIÊNCIA DA COMPUTAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO



**PREFREC: UMA METODOLOGIA PARA  
DESENVOLVIMENTO DE SISTEMAS DE RECOMENDAÇÃO  
UTILIZANDO ALGORITMOS DE MINERAÇÃO DE  
PREFERÊNCIAS**

CLEIANE GONÇALVES OLIVEIRA

Uberlândia - Minas Gerais  
2014



UNIVERSIDADE FEDERAL DE UBERLÂNDIA  
FACULDADE DE CIÊNCIA DA COMPUTAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO



CLEIANE GONÇALVES OLIVEIRA

**PREFREC: UMA METODOLOGIA PARA  
DESENVOLVIMENTO DE SISTEMAS DE RECOMENDAÇÃO  
UTILIZANDO ALGORITMOS DE MINERAÇÃO DE  
PREFERÊNCIAS**

Dissertação de Mestrado apresentada à Faculdade de Ciência da Computação da Universidade Federal de Uberlândia, Minas Gerais, como parte dos requisitos exigidos para obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Banco de Dados.

Orientadora:

Prof<sup>a</sup>. Dr<sup>a</sup>. Sandra Aparecida de Amo

Uberlândia, Minas Gerais  
2014

Dados Internacionais de Catalogação na Publicação (CIP)  
Sistema de Bibliotecas da UFU

---

O48p Oliveira, Cleiane Gonçalves, 1986-  
PrefREC: uma metodologia para desenvolvimento de sistemas de  
recomendação utilizando algoritmos de mineração de preferências /  
Cleiane Gonçalves Oliveira. - 2014.  
100 p. : il.

Orientadora: Sandra Aparecida de Amo.  
Dissertação (mestrado) – Universidade Federal de Uberlândia, Pro-  
grama de Pós-Graduação em Ciência da Computação.  
Inclui bibliografia.

1. Computação - Teses. 2. Banco de dados - Teses. 3. Mineração de  
dados (Computação) - Teses. I. Amo, Sandra Aparecida de. II. Univer-  
sidade Federal de Uberlândia. Programa de Pós-Graduação em Ciência  
da Computação. III. Título.

---

CDU: 681.3

UNIVERSIDADE FEDERAL DE UBERLÂNDIA  
FACULDADE DE CIÊNCIA DA COMPUTAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Os abaixo assinados, por meio deste, certificam que leram e recomendam para a Faculdade de Ciência da Computação a aceitação da dissertação intitulada **“PrefRec: Uma Metodologia para Desenvolvimento de Sistemas de Recomendação utilizando Algoritmos de Mineração de Preferências”** por **Cleiane Gonçalves Oliveira** como parte dos requisitos exigidos para a obtenção do título de **Mestre em Ciência da Computação**.

Uberlândia, 7 de Fevereiro de 2014

Orientadora:

---

Prof<sup>a</sup>. Dr<sup>a</sup>. Sandra Aparecida de Amo  
Universidade Federal de Uberlândia

Banca Examinadora:

---

Prof. Dr. Humberto Luiz Razente  
Universidade Federal de Uberlândia

---

Prof. Dr. Caetano Traina Júnior  
Universidade de São Paulo



# Dedicatória

*Aos meus pais Alcindio e Maria e minhas irmãs Ariane e Rosiane.  
Ao meu marido Leandro.*



# Agradecimentos

Agradeço a Deus primeiramente. Sem a sua presença não haveria nada para aqui agradecer.

Agradeço muito aos meus pais que sempre me incentivaram, me deram apoio e lutaram para que eu conquistasse tudo que tenho hoje.

À minha irmã Ariane por tantas vezes escutar meus lamentos, me motivar a seguir adiante e compreender tantos momentos de ausência.

À minha irmã Rosiane pelo exemplo de dinamismo e por me mostrar que ainda há muito o que fazer.

Ao meu marido Leandro sem o qual eu não seria capaz de concluir esse mestrado. Obrigada pela colaboração durante todo o processo, principalmente na coleta de dados nos sábados, domingos e madrugadas nos laboratórios do Instituto.

À minha avó Clemêncio por sempre ser meu exemplo de dedicação e serviço aos outros.

A todos os meus tios, tias, primos e demais parentes que me acompanharam.

Aos muitos amigos que tive que me afastar, aos novos amigos que encontrei em Uberlândia e que me deram muito apoio durante toda a caminhada.

Às amigas Juliete e Geycy por compartilharem tantos momentos ímpares durante essa trajetória. Aos demais amigos do mestrado que sempre reforçaram minha esperança.

À família GPP-Uberlândia que foi peça fundamental para a minha caminhada e me permitiu sentir acolhida em uma terra tão distante da minha casa.

Ao Instituto Federal do Norte de Minas Gerais - Campus Januária que me proporcionou dedicação exclusiva por maior parte do tempo da pesquisa e pela disponibilização de computadores para testes.

A tantas pessoas que direta ou indiretamente colaboraram para que eu alcançasse esse objetivo. Em especial aos professores da FACOM e ao secretário Erisvaldo pela atenção.

E principalmente à professora Sandra de Amo pelo profissionalismo, apoio, paciência e orientação para a realização deste trabalho.



*"A mente que se abre a uma nova ideia jamais voltará ao seu tamanho original."*  
*Albert Einstein*



# Resumo

A enorme quantidade de informação disponível na web tem dificultado os usuários a selecionarem itens que atendam suas necessidades. Os Sistemas de Recomendação surgem como ferramentas indispensáveis neste cenário de sobrecarga de informação, a fim de filtrar o que é de interesse do usuário e permitir que ele tenha uma experiência diferenciada com os atuais sistemas de informação.

Apresentamos nessa dissertação uma metodologia para o desenvolvimento de Sistemas de Recomendação, utilizando algoritmos de mineração de preferências: a PrefRec. Objetivamos com essa proposta a construção de Sistemas de Recomendação que tenham bons valores de acurácia e que permitam uma interação com o usuário mais satisfatória, a partir das medidas de validação de acurácia, cobertura, novidade e *serendipity*.

A utilização de algoritmos de mineração de preferências objetiva compreender as preferências dos usuários sobre as características dos itens, alcançando recomendações mais acuradas. No estudo de caso implementado, o Sistema de Recomendação XPrefRec, aplicamos um algoritmo minerador de um tipo especial de preferências contextuais, a fim de definir qual a preferência do usuário diante de determinado contexto.

Analisamos ainda quais são os fatores da metodologia proposta que influenciam na performance do Sistema de Recomendação, e realizamos a comparação do desempenho desse sistema com o estado da arte em relação a Sistemas de Recomendação híbridos.

**Palavras chave:** sistemas de recomendação, mineração de preferências, preferências contextuais



# Abstract

The huge amount of information available on the web has bothered users to select items that meet their needs . The Recommender Systems emerged as indispensable tools in this information overload scenario in order to filter out what is of interest to the user and allow him to have a differentiated experience with existing information systems.

We present a methodology for developing recommendation systems, using mining algorithms preferences: PrefRec. We aim at building Recommender Systems that have good values of accuracy and allowing a more satisfying interaction to the user, from the validation measures of accuracy, coverage, novelty and serendipity.

The use of algorithms for mining preferences objectively understand the user's preferences about the characteristics of the items, achieving more accurate recommendations. In the case study implemented, the Recommendation System XPrefRec, we apply a mining algorithm from a special type of contextual preferences in order to define the user preferences on a certain context.

We also analyzed what are the factors that influence the methodology proposed in the Recommendation System performance and presents the comparison of the performance of this system with the state of the art regarding Hybrid Recommender Systems.

**Keywords:** recommender systems, preferences mining, contextual preferences



# Sumário

<b>Lista de Figuras</b>	<b>xvii</b>
<b>1 Introdução</b>	<b>19</b>
<b>2 Conceitos preliminares</b>	<b>23</b>
2.1 Sistemas de Recomendação . . . . .	23
2.1.1 Abordagens . . . . .	25
2.1.2 Desafios em Sistemas de Recomendação . . . . .	30
2.1.3 Medidas de Qualidade em Sistemas de Recomendação . . . . .	32
2.2 Mineração de Preferências . . . . .	35
2.2.1 Modelo de Preferências Qualitativas . . . . .	37
2.2.2 Mineração de Preferências Não Contextuais . . . . .	38
2.2.3 Mineração de Preferências Contextuais . . . . .	38
<b>3 Trabalhos Correlatos</b>	<b>45</b>
3.1 Abordagens de Filtragem Colaborativa . . . . .	45
3.2 Abordagens Baseadas em Conteúdo . . . . .	48
3.3 Abordagens Híbridas . . . . .	49
3.4 Abordagens que Utilizam Mineração de Dados . . . . .	51
<b>4 A Metodologia PrefRec para Sistemas de Recomendação</b>	<b>53</b>
4.1 Etapa 1 - Representação dos Usuários . . . . .	54
4.2 Etapa 2 - Agrupamento dos Usuários . . . . .	56
4.3 Etapa 3 - Agregação de Preferências Individuais . . . . .	59
4.3.1 Média Aritmética . . . . .	60
4.3.2 Média Ponderada . . . . .	60
4.3.3 Quantificadores Fuzzy . . . . .	61
4.4 Etapa 4 - Mineração de Preferências . . . . .	62
4.5 Etapa 5 - Recomendação . . . . .	64
<b>5 Validação da Metodologia PrefRec</b>	<b>67</b>
5.1 Dados de Teste . . . . .	67

5.2	Medidas de Validação . . . . .	68
5.2.1	Acurácia . . . . .	68
5.2.2	Cobertura . . . . .	71
5.2.3	Novidade . . . . .	72
5.2.4	Serendipity . . . . .	73
5.3	Fatores que Influenciam no Desempenho da Metodologia PrefRec . . . . .	74
5.4	Comparação com Outros Trabalhos . . . . .	76
<b>6</b>	<b>Resultados Experimentais</b>	<b>79</b>
6.1	Fatores que Influenciam no Desempenho da Metodologia PrefRec . . . . .	79
6.1.1	Grupo de Teste 1 . . . . .	79
6.1.2	Grupo de Teste 2 . . . . .	81
6.1.3	Grupo de Teste 3 . . . . .	82
6.2	Comparação com Outros Trabalhos . . . . .	83
6.2.1	Acurácia . . . . .	83
6.2.2	Cobertura . . . . .	84
6.2.3	Novidade . . . . .	85
6.2.4	Serendipity . . . . .	85
6.2.5	Tempo de Execução . . . . .	86
<b>7</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>89</b>
	<b>Referências Bibliográficas</b>	<b>93</b>

# Lista de Figuras

1.1	Capítulos da dissertação	22
2.1	Categorias da Filtragem Colaborativa	28
2.2	(a) Uma instância I e (b) um banco de dados de preferência P [de Amo et al. 2012a]	40
2.3	Rede de preferência PNet1 [de Amo et al. 2012a]	40
3.1	Clusterização dos usuários em [Goldberg et al. 2001] com 4 níveis de recursão formando 40 clusters.	46
3.2	Grafo baseado em usuários e itens em [Huang et al. 2004]	47
4.1	Arquitetura Geral da Metodologia PrefRec	54
4.2	Algoritmo Cure com diferentes fatores de retração. Fonte: [Guha et al. 2001]	58
4.3	Clusters baseados em densidade. Fonte: [Ester et al. 1996]	59
4.4	Exemplos de $M_{pref}'s$	60
4.5	Quantificadores <i>fuzzy</i> e seus parâmetros. Fonte: [Chiclana et al. 2001]	61
4.6	$M_{pref}'s$ calculadas com quantificadores <i>fuzzy</i>	62
4.7	Entrada e Saída do Minerador de Preferências na metodologia PrefRec	63
4.8	Algoritmo <b>Order-By-Preferences</b> adaptado de [Cohen et al. 1999]	64
5.1	Transformação de avaliações para pares	69



# Capítulo 1

## Introdução

O crescimento de mercados *online*, a introdução de novos serviços virtuais, o crescimento explosivo da internet e a variedade de informação disponível na Web tem tido um impacto considerável nos hábitos dos consumidores, provendo a eles acesso a uma enorme variedade de produtos e informações.

Enquanto a liberdade de compra tem feito do comércio online uma indústria multibilionária, também fez com que seja mais difícil para os consumidores selecionar produtos que melhor atendam suas necessidades. A grande quantidade de informação frequentemente sobrecarrega os usuários, fazendo com que a disponibilidade de escolhas, no lugar de produzir um benefício, comece a diminuir o bem-estar dos usuários. Os usuários começam a encontrar muita dificuldade para chegar às escolhas mais apropriadas dentre a imensidão de variedades de itens (produtos e serviços) que estes *web sites* oferecem.

Diante deste cenário, novos sistemas de informação começaram a surgir com o objetivo de filtrar informações úteis aos usuários. Ao tentar fazer alguma escolha, o usuário só recebe as informações que lhe são pertinentes. A estes sistemas dá-se o nome de Sistemas de Recomendação que podem ser definidos como ferramentas e técnicas que fornecem sugestões de itens a serem utilizados pelos usuários.

Os Sistemas de Recomendação surgem como uma ferramenta indispesável nos dias atuais devido a grande quantidade de informação disponibilizada. Além de servir como um filtro que pode ser muito eficiente, ele permite uma experiência diferenciada do usuário com o sistema de informação, que o deixa mais satisfeito com as transações que ele realiza. Nos últimos tempos, tem se constatado que, quanto mais personalizado é o atendimento ao cliente, tanto mais fiel ele se torna à empresa. Além disso, as recomendações quando bem feitas podem aumentar a quantidade de itens vendidos, além de colaborar na venda de uma maior diversidade de produtos. Enfim, Sistemas de Recomendação trazem vantagens não só para o usuário que alcança uma experiência mais satisfatória, mas também para os empreendimentos, que descobrem mais sobre as preferências dos seus clientes e assim podem ter mais sucesso.

Pode-se encontrar atualmente Sistemas de Recomendação em diversos cenários como:

recomendação de entretenimento ([www.last.fm](http://www.last.fm); [www.youtube.com](http://www.youtube.com); [www.netflix.com](http://www.netflix.com)), conteúdo ([www.yahoo.com](http://www.yahoo.com)), e-commerce ([www.amazon.com](http://www.amazon.com); [www.submarino.com.br](http://www.submarino.com.br)) e serviços ([www.tripadvisor.com](http://www.tripadvisor.com)). Também há trabalhos de recomendação de colaboradores de pesquisa ([Brandão e Moro 2012]), de amizades ([www.facebook.com](http://www.facebook.com)), entre outras tantas aplicações.

Os Sistemas de Recomendação utilizam-se de várias técnicas para conseguir recomendar. Aqueles denominados Baseados em Conteúdo avaliam, por exemplo, o histórico de compras dos usuários e recomendam itens semelhantes ao que ele já mostrou interesse. Já os sistemas de Filtragem Colaborativa recomendam itens ao usuário ativo (usuário ao qual se deseja recomendar) a partir de outros usuários que tenham preferências semelhantes, isto é, que possuam um histórico de compras/avaliações similares. Os itens já avaliados pelos usuários semelhantes são recomendados ao usuário ativo. Essa abordagem parte do princípio de que se duas pessoas concordaram no passado (compraram ou avaliaram itens de forma semelhante), a probabilidade delas continuarem concordando no futuro é grande. Essa abordagem tem se tornado a mais popular em Sistemas de Recomendação a partir dos bons resultados que ela obtém.

Para alcançar boas recomendações, os Sistemas de Recomendação enfrentam diversos desafios. Como as recomendações são extraídas a partir de dados já existentes, um primeiro desafio é a disponibilidade desses dados. Quanto mais informação, mais acurada é a recomendação. Pouca informação gera recomendações pobres. Porém, ao mesmo tempo que é necessária muita informação para boas recomendações, o desempenho de qualquer sistema pode ser comprometido por essa quantidade de dados, isto é, o sistema precisa ser escalável em relação ao aumento dos dados com o tempo. Outros desafios específicos são: como recomendar para um usuário que acabou de se cadastrar no sistema? Como recomendar um item que acabou de ser incluído no banco de dados?

Tentando minimizar estes desafios, muitos Sistemas de Recomendação incorporam abordagens híbridas entre as técnicas Baseadas em Conteúdo e de Filtragem Colaborativa, agregando as vantagens que cada uma apresenta. Além disso, outro recurso é conhecer com detalhe as preferências dos usuários para conseguir recomendar itens realmente úteis a ele.

Neste contexto, apresentamos nesta dissertação uma metodologia para desenvolvimento de Sistemas de Recomendação utilizando algoritmos de mineração de preferências: a PrefRec. Objetivamos com essa proposta a construção de Sistemas de Recomendação híbridos, entre as abordagens Baseada em Conteúdo e de Filtragem Colaborativa, capazes de recomendar itens ainda não avaliados, que tenham bons valores de acurácia e que permitam uma interação mais satisfatória com o usuário.

Para alcançar tais características, a metodologia PrefRec propõe a construção de um modelo de recomendação em um processamento *offline*, para ser utilizado na recomendação *online* ao usuário ativo. A construção do modelo de recomendação se dá a partir

da colaboração entre usuários já cadastrados no sistema. Os usuários são organizados em grupos para definição de perfis de preferências. Cada perfil é definido a partir da mineração de preferências dos usuários pertencentes ao grupo.

A utilização de algoritmos de mineração de preferências visa alcançar um Sistema de Recomendação mais acurado a partir do melhor conhecimento sobre o que os usuários preferem. Além disso, o modelo de preferência minerado deve ser capaz de recomendar os itens a partir das características que eles apresentam, minimizando o problema de recomendação de itens recém inseridos no sistema.

No estudo de caso realizado neste trabalho, propomos a aplicação de uma maneira específica de representação de preferências: as *regras de preferências contextuais*. Estas regras permitem que, dados dois itens, seja possível inferir qual deles é preferido a partir das características que eles apresentam. Além disso, esta forma de representar preferências permite que seja incluído um contexto, isto é, quais características dos itens determinam a preferência do usuário. Uma *regra de preferência contextual* pode ser, por exemplo, a afirmativa: "entre dois filmes de comédia, prefiro os filmes com o ator Jim Carrey aos com o ator Tom Hanks". Nesta afirmativa, o gênero comédia é o contexto sobre o qual acontece a preferência de filmes com o ator Jim Carrey.

Diante deste cenário, os objetivos deste trabalho de dissertação são:

- Desenvolver uma metodologia de Sistema de Recomendação que utiliza algoritmos de mineração de preferências;
- Utilizar a metodologia para implementar o sistema XPrefRec que utiliza um algoritmo minerador de um tipo especial de preferências contextuais. Pelo que sabemos, não existe sistema de recomendação baseado nesta representação de preferências do usuário;
- Estudar os fatores que influenciam o desempenho do Sistema de Recomendação;
- Analisar a performance da metodologia proposta em comparação ao estado da arte em relação a Sistemas de Recomendação de Filtragem Colaborativa e híbrido, através de diferentes medidas de validação: precisão, revocação, cobertura, novidade e *serendipity* (Seção 5.2).

As contribuições que trazemos nesse trabalho são:

- Definição de uma metodologia para Sistemas de Recomendação que utiliza mineração de preferências com desempenho superior ao estado da arte - Capítulo 4 e Capítulo 6;
- Nova maneira de representar os usuários em Sistemas de Recomendação - Capítulo 4;
- Estudo de alternativas de agregação de preferências em um grupo de usuários, Capítulo 5 e Capítulo 6;

- Estudo de fatores que influenciam no desempenho da metodologia proposta, Capítulo 6.

Esta dissertação está organizada como segue. No Capítulo 2 introduzimos os conceitos preliminares necessários para a compreensão da metodologia PrefRec e dos trabalhos correlatos. O Capítulo 3 apresenta os trabalhos correlatos em Sistemas de Recomendação. O Capítulo 4 apresenta a metodologia PrefRec proposta. No Capítulo 5 discutimos a validação da metodologia. O Capítulo 6 apresenta os resultados experimentais obtidos. O Capítulo 7 apresenta as conclusões e os trabalhos futuros. A figura 1.1 sugere a sequência de leitura deste trabalho.



Figura 1.1: Capítulos da dissertação

# Capítulo 2

## Conceitos preliminares

Neste capítulo são destacados alguns conceitos necessários para o entendimento do trabalho. Também são apresentadas algumas das técnicas que são utilizadas na implementação da metodologia PrefRec. Na seção 2.1 sintetizamos os conceitos sobre Sistemas de Recomendação, as abordagens existentes, os desafios enfrentados e como eles são avaliados. Na seção 2.2 discutimos como representar preferências para, assim, apresentar mineradores capazes de minerar o formato de preferência desejado.

### 2.1 Sistemas de Recomendação

[Ricci et al. 2011] define Sistemas de Recomendação como ferramentas e técnicas que fornecem sugestões de itens a usuários. As sugestões são relacionadas a diversos processos de tomada de decisão como: itens de compra em um supermercado, músicas para escutar em uma rádio, notícias para ler em um site jornalístico. Os Sistemas de Recomendação são direcionados para auxiliar os usuários a lidarem com um grande número de alternativas que uma loja de departamentos, por exemplo, pode oferecer.

Um famoso caso real é o site da Amazon.com, que emprega Sistemas de Recomendação para personalizar a loja virtual para cada cliente ([Jannach 2008]). Esta empresa também fornece médias de avaliações dos usuários para os livros apresentados e uma lista de outros livros que são comprados por usuários que adquiriram um livro específico. O Netflix ([www.Netflix.com](http://www.Netflix.com)), empresa de aluguéis de filmes pela internet, calcula recomendações para cada usuário, além de mostrar os itens mais populares e premiados. O site da Microsoft fornece sugestões de downloads de ferramentas a partir dos itens já pesquisados pelo usuário ([Shani e Gunawardana 2011]).

Os Sistemas de Recomendação surgem como uma das principais soluções para a sobrecarga de informação existente, permitindo diminuir o esforço de busca dos usuários ([Desrosiers e Karypis 2011] e [Su e Khoshgoftaar 2009]). Além disso, também prometem uma alta fidelidade por parte dos clientes, aumento das vendas, mais receita publicitária e alcance de mais benefícios a partir de promoções específicas para determinados segmentos

de clientes. Além dessas vantagens, [Ricci et al. 2011] enumera outros benefícios que os Sistemas de Recomendação podem trazer para uma empresa ou organização:

- Aumentar o número de itens vendidos: o primeiro objetivo para implantar um Sistema de Recomendação é aumentar a taxa de conversão, isto é, o número de usuários que aceitam as recomendações e as consomem, comparada ao número de simples visitantes que somente navegam pela informação disponível. Este objetivo é alcançado porque os itens recomendados tendem a serem adequados às necessidades e desejos dos usuários.
- Vender itens mais diversos: os Sistemas de Recomendação permitem o usuário selecionar itens que podem ser difíceis de encontrar sem uma recomendação precisa. Em um cenário de recomendações de filme, por exemplo, a empresa está interessada em alugar todos os DVDs no catálogo, não somente os mais populares. Isso pode ser difícil sem um Sistema de Recomendação, uma vez que a empresa não pode correr o risco de fazer propagandas de filmes que não são do gosto do usuário em particular. Além disso, um Sistema de Recomendação sugere ou faz propaganda de filmes não populares para os usuários certos.
- Aumentar a satisfação do usuário: o usuário irá encontrar recomendações interessantes e relevantes, fazendo com que goste de usar o sistema. A combinação de eficientes e acuradas recomendações com uma interface apropriada pode aumentar a avaliação subjetiva do usuário em relação ao sistema. Isto pode aumentar o uso do sistema e a probabilidade das recomendações serem aceitas.
- Aumentar a fidelidade do usuário: um usuário tende a ser leal a um web site que, quando visitado, reconhece suas preferências e o trata como um importante visitante. Esta é a característica comum de um Sistema de Recomendação uma vez que calculam recomendações aproveitando as informações adquiridas dos usuários em suas interações prévias. Consequentemente, quanto mais o usuário interage com o sistema, melhor se consegue representar as preferências do usuário e mais a recomendação feita pelo Sistema de Recomendação pode ser eficiente, acertando as preferências do usuário.
- Melhor entendimento do que o usuário quer: o Sistema de Recomendação consegue descrever as preferências dos usuários, ou coletada explicitamente ou predita pelo sistema. A empresa pode então decidir reusar o conhecimento para outros objetivos, como melhorar o gerenciamento dos itens em estoque ou em produção.

Diante de tantos benefícios, os Sistemas de Recomendação vem sendo empregados em uma grande variedade de aplicações tais como:

- recomendações de livros e cds ( [Linden et al. 2003]; [Mooney e Roy 1999]),

- músicas (www.last.fm, [Shardanand e Maes 1995]),
- filmes ( [Miller et al. 2003], [NCBI 2009] ),
- notícias ( [Billsus et al. 2002] ),
- páginas web ( [Mobasher et al. 2002] ),
- piadas ( [Goldberg et al. 2001] ),
- serviços: pacotes de viagens (www.tripadvisor.com), consultorias, vendas e aluguéis de imóveis,
- e-commerce ( [Linden et al. 2003] ),
- televisão digital ( [Blanco-Fernández et al. 2008] ),
- colaborações entre pesquisadores para formação de novos grupos e colaborações em pesquisas ( [Brandão e Moro 2012] ).

### 2.1.1 Abordagens

Os Sistemas de Recomendação são desenvolvidos sob diversas abordagens. A seguir apresentamos as mais comuns e as suposições que elas assumem para prover recomendação. No capítulo 3 apresentamos trabalhos que exploram essas abordagens.

#### Filtragem Colaborativa

A abordagem de Sistema de Recomendação denominada Filtragem Colaborativa foi desenvolvida a partir de uma simples observação: as pessoas frequentemente usam recomendações dadas por outras pessoas em suas atividades rotineiras. Por exemplo: ao selecionar um livro para ler, procura-se recomendações de quem já leu aquele livro; empregadores contam com cartas de recomendações para escolher novos funcionários; ao escolher um filme para assistir as pessoas tendem a ler comentários e críticas em jornais ou sites ( [Ricci et al. 2011] ).

A técnica de Filtragem Colaborativa recomenda para um usuário ativo (a quem se deseja recomendar) os itens que outros usuários, com gostos similares a ele, gostaram no passado. A similaridade entre dois usuários é calculada baseada na similaridade do histórico de avaliação/compra dos usuários. A ideia chave é que a nota do usuário ativo  $u$  para um novo item  $i$  é provavelmente similar a nota de outro usuário  $v$ , se  $u$  e  $v$  avaliaram itens no passado de forma semelhante ( [Desrosiers e Karypis 2011] ).

A abordagem tradicional de Filtragem Colaborativa foi introduzida em [Resnick et al. 1994], sendo este o trabalho considerado como principal precursor dos Sistemas de Recomendação. Os dados são representados pela matriz usuário×item, apresentada na tabela 2.1. A matriz apresenta as avaliações feitas pelos usuários aos itens. A partir

dela é possível calcular recomendações aos usuários. Os espaços com "\*" indicam que o valor da avaliação não foi informado.

	$i_1$	$i_2$	$i_3$	$i_4$	$i_5$	$i_6$
Paulo	5	2	4	5	*	*
João	*	*	*	4	2	5
Márcia	*	3	5	3	2	4
Carlos	3	2	*	2	3	*
Ana	5	4	3	4	3	3
Mauro	1	*	2	5	*	4
Joaquim	3	1	*	4	5	3

Tabela 2.1: Matriz usuário×item

Como exemplo vamos calcular recomendações para o usuário João. O primeiro passo é encontrar quais são os usuários que possuem gosto semelhante ao dele. Para tal, calcula-se a similaridade de João com os outros usuários a partir das notas dadas aos itens em comum. A medida de similaridade mais amplamente utilizada é o *Pearson Correlation Coefficient* (PCC), dado pela fórmula:

$$sim(u, v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)^2 \sum_{i \in I_{uv}} (r_{vi} - \bar{r}_v)^2}}$$

Onde  $I_{uv}$  representa o conjunto de itens que os usuários  $u$  e  $v$  avaliaram em comum.  $\bar{r}_u$  é a média das notas dadas pelo usuário  $u$  aos itens que ele avaliou em comum com o usuário  $v$ . Assim, a similaridade entre o usuário João e os demais usuários, a partir dos dados da tabela 2.1, é:

- $sim(\text{João}, \text{Paulo}) = 0$ ;
- $sim(\text{João}, \text{Márcia}) = 0.98$ ;
- $sim(\text{João}, \text{Carlos}) = -1$ ;
- $sim(\text{João}, \text{Ana}) = 0.19$ ;
- $sim(\text{João}, \text{Mauro}) = -1$ ;
- $sim(\text{João}, \text{Joaquim}) = -0.98$ .

O valor de similaridade retornado pelo PCC se encontra no intervalo de  $[-1, 1]$ . Valores próximos a 1 indicam que os dois usuários tendem a concordar, e valores próximos a -1 indicam que os dois usuários tendem a discordar um do outro. Valores igual a 0 indicam que o histórico de avaliações dos dois usuários não são correlacionados. Caso dois usuários não tenham avaliado itens em comum não é possível calcular a similaridade entre eles.

Para predizer a nota a um determinado item que João ainda não avaliou é utilizada a fórmula abaixo:

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in U, v \neq u} (r_{vi} - \bar{r}_v) * sim(u, v)}{\sum_{v \in U, v \neq u} |sim(u, v)|}$$

Na fórmula  $U$  é o conjunto de usuários existentes no banco de dados do sistema. A predição das notas para os itens que João ainda não avaliou, a partir dos dados da tabela 2.1, para o usuário João é:

- $i_1 = 1.35$ ;
- $i_2 = 1.13$ ;
- $i_3 = 2.02$ .

Os itens com maiores valores preditos são os recomendados ao usuário João. Construindo um ranking, os itens seriam recomendados na seguinte ordem:  $i_3$ ,  $i_1$  e  $i_2$ .

Observa-se que a abordagem de Filtragem Colaborativa não é capaz de recomendar um item que ainda não recebeu avaliação. Da mesma forma, se o usuário ativo não avaliou itens em comum com outros usuários a abordagem não consegue recomendar outros itens. Entretanto, a abordagem de Filtragem Colaborativa apresenta bons resultados na literatura e permite a recomendação de uma diversidade de itens já avaliados pelos usuários similares ao usuário ativo.

Diversos outros trabalhos partiram dessa ideia inicial e propuseram outros recursos para aprimorar as recomendações. Assim, definiram-se categorias (figura 2.1) dentro da abordagem de Filtragem Colaborativa para melhor classificar e compreender as propostas desenvolvidas.

A ideia inicial de [Resnick et al. 1994] é categorizada como *user-based* e *memory-based*. Denomina-se *user-based* por usar as avaliações de outros usuários para o cálculo da recomendação. Por outro lado, existe a abordagem *item-based*, [Sarwar et al. 2001], que calcula a predição de um item encontrando itens similares a partir das avaliações que eles receberam. *Memory-based* se refere ao cálculo das predições, baseando-se em uma coleção inteira de itens previamente avaliados pelos usuários. Em contrapartida, existe a abordagem *model-based*, que aplica algoritmos de mineração de dados para construir um modelo a partir dos dados disponíveis, o qual é então utilizado para realizar predições de avaliação.

## Baseado em Conteúdo

[Lops et al. 2011] apresenta que os Sistemas de Recomendação Baseados em Conteúdo analisam as características dos itens previamente avaliados pelo usuário, e constróem seu respectivo perfil de interesse. O perfil é uma representação estruturada das preferências do usuário, usada para recomendar novos itens interessantes. O processo básico de recomendação consiste em combinar os atributos do perfil com os atributos dos itens. O

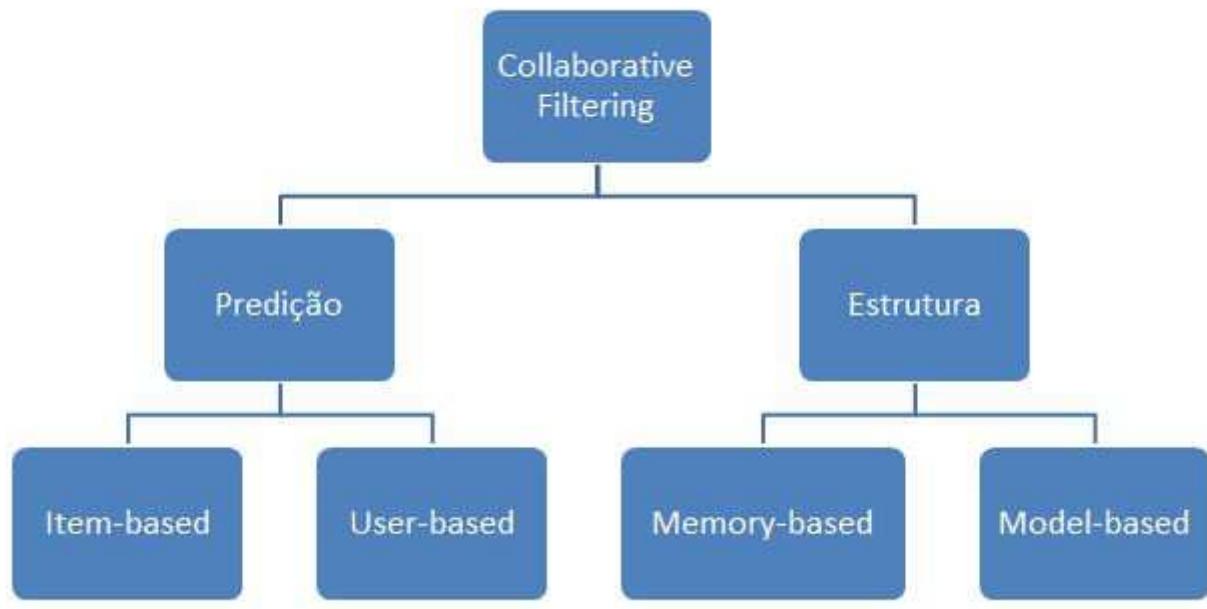


Figura 2.1: Categorias da Filtragem Colaborativa

resultado é um julgamento de relevância que representa o nível de interesse do usuário naquele item. Se o perfil reflete bem as preferências do usuário, ele vai ser de grande eficácia para o processo de recomendação.

Se um usuário avaliou positivamente um filme do gênero comédia, por exemplo, o sistema aprende a recomendar outros filmes do mesmo gênero. Essa abordagem de Sistemas de Recomendação parte da observação de que, se o usuário tem preferência sobre determinada característica hoje, ele tende a manter essa preferência durante algum tempo ([Desrosiers e Karypis 2011]).

A abordagem Baseada em Conteúdo, como observa [Burke 2007], trata a tarefa de recomendação como um problema de classificação específico para cada usuário. Um classificador é treinado a partir dos itens que o usuário informou serem preferidos ou não, e as classes são as notas informadas pelo usuário. Um dos métodos mais utilizados e com bons resultados é o classificador bayesiano *naive bayes*. Este classificador é uma abordagem probabilística de aprendizado e gera um modelo baseado nos atributos dos itens avaliados previamente.

O classificador estima a probabilidade  $P(c|i)$  do item  $i$  pertencer a classe  $c$ . Considerando que o item  $i$ , na abordagem Baseada em Conteúdo, é um conjunto de atributos  $(a_1, a_2, \dots, a_n)$ , a probabilidade  $P(c|i)$  é dada pela fórmula:

$$P(c|i) = \frac{P(c)}{P(i)} \prod_{k=1}^n P(a_k|c)$$

As probabilidades  $P(c)$  e  $P(a_k|c)$  são estimadas a partir dos dados de treinamento e indicam, respectivamente, a probabilidade dos itens observados na classe e a probabilidade

de observar aquele atributo  $a_k$  na classe  $c$ . A probabilidade  $P(i)$  pode ser ignorada uma vez que ela é constante para qualquer item.

O classificador *naive bayes* assume que todos os atributos do item  $i$  observado são independentes uns dos outros. O perfil do usuário é então o conjunto das probabilidades de cada característica em cada classe existente. No momento da recomendação, para verificar a probabilidade do novo item ser de determinada classe, procura-se no perfil do usuário a probabilidade de cada característica que o item possui para aquela classe. O resultado é a multiplicação dos valores de cada característica. Para classificar o item  $i$  é escolhida então a classe com a maior probabilidade. A classe corresponde à nota predita para aquele item.

A abordagem Baseada em Conteúdo possui a capacidade de recomendar itens que nunca foram avaliados por algum usuário: basta que o item corresponda ao perfil do usuário para que ele seja recomendado. Já na abordagem Filtragem Colaborativa o item depende de já ter sido avaliado por outro usuário similar ao usuário ativo. Em contrapartida, a abordagem Baseada em Conteúdo só recomenda itens semelhantes aos já avaliados pelo usuário ativo, especializando as recomendações. A abordagem de Filtragem Colaborativa, por sua vez, é capaz de recomendar itens diversos a partir do que os usuários similares ao usuário ativo já recomendaram.

## Outras categorias

Muitas outras tentativas de Sistemas de Recomendação foram desenvolvidas buscando sempre aprimorar as recomendações. Uma proposta é o Sistema de Recomendação demográfico, que explora os dados do perfil pessoal do usuário para recomendar. Assume-se que diferentes recomendações devem ser geradas para diferentes nichos demográficos. Como exemplo, temos *web sites* direcionados para determinados idiomas, ou sugestões customizadas de acordo com a idade do usuário. Pesquisas com essa abordagem podem ser vistas em [Krulwich 1997], [Pazzani 1999] e [Mahmood e Ricci 2007].

Outra abordagem que vem sendo muito pesquisada é a *community-based*, que se baseia nas preferências dos amigos dos usuários. Com o advento das redes sociais, procura-se saber o quanto uma ligação de um usuário ao outro influencia na sua preferência. Essa abordagem parte da ideia que as pessoas tendem a confiar mais em recomendações dos seus amigos do que em recomendações de usuários similares porém anônimos ( [Ricci et al. 2011]). Trabalhos com essa abordagem são os apresentados em [Groh e Ehmig 2007] e [Guy et al. 2009].

Outra técnica muito utilizada é a hibridação entre as categorias existentes. Uma técnica A é combinada com outra técnica B no sentido de usar as vantagens de A para consertar as desvantagens de B. A metodologia proposta neste trabalho propõe uma abordagem híbrida entre as técnicas de Filtragem Colaborativa e Baseada em Conteúdo.

### 2.1.2 Desafios em Sistemas de Recomendação

Nesta seção enumeramos os principais problemas/desafios que as abordagens apresentadas encontram para prover boas recomendações. Estes problemas são objeto de muitas pesquisas que buscam maneiras de minimizá-los ou até eliminá-los.

#### Análise de Conteúdo Limitado

Este problema impacta a abordagem Baseada em Conteúdo ( [Lops et al. 2011]). Tal abordagem tem um limite natural no número e tipo de características que são associadas aos itens recomendados. Por exemplo, para *web pages* a técnica de extração de característica do texto ignora completamente a qualidade de estética e informação multi-mídia adicional. A atribuição de características aos itens pode ser insuficiente para definir aspectos distintos dos itens que se tornam necessários para eliciar o interesse do usuário.

#### *Over Specialization*

Também é um problema específico da abordagem Baseada em Conteúdo. A abordagem não tem nenhum método para recomendar algo inesperado. O sistema sugere itens que são semelhantes ao perfil do usuário e produz recomendações com um grau limitado de novidade. Por exemplo, se um usuário avaliou somente filmes de um mesmo diretor, o sistema recomendará com maior intensidade filmes daquele mesmo diretor. Uma alternativa usada por alguns trabalhos é adicionar recomendações aleatórias, mas sem predição se será do gosto do usuário.

#### Esparsidade dos Dados

[Adomavicius e Tuzhilin 2005] afirmam que em qualquer sistema de recomendação o número de avaliações já obtido é geralmente muito pequeno comparado ao número de avaliações necessárias para realizar a predição. O sucesso de um Sistema de Recomendação de Filtragem Colaborativa, em específico, depende da disponibilidade de uma massa crítica de usuários. Por exemplo, em um Sistema de Recomendação de filmes, podem existir muitos filmes que foram avaliados por um pequeno grupo de pessoas e que raramente serão recomendados, ainda que as notas sejam altas. Também para um usuário que tem preferências que não são comparadas com o resto da população serão geradas recomendações pobres.

Da mesma forma, a abordagem Baseada em Conteúdo sofre com os poucos dados disponíveis. Se um usuário não interage com o sistema e não disponibiliza suas preferências de forma explícita ou implícita, dificilmente o Sistema de Recomendação conseguirá calcular boas recomendações. A esparsidade dos dados provoca os dois problemas apresentados a seguir.

### ***Cold-start de Item***

Novos itens são adicionados regularmente aos sistemas. Até que um novo item seja avaliado por um expressivo número de usuários, a técnica de Filtragem Colaborativa não conseguirá recomendá-lo, como afirma [Adomavicius e Tuzhilin 2005]. No exemplo dado na tabela 2.1, se nenhum usuário tivesse avaliado o item  $i_3$  ele nunca seria recomendado para o usuário João.

Entretanto, a abordagem Baseada em Conteúdo é capaz de recomendar itens novos, uma vez que ela explora as características do item para recomendar. Basta que o item possua características que combinem com o perfil do usuário para que ele possa ser recomendado. Desta forma, muitos trabalhos utilizam a hibridação entre as duas abordagens para minimizar o problema de *cold-start* de item. A metodologia proposta nesta dissertação também sugere essa hibridação para o tratamento de *cold-start* de item.

### ***Cold-start de Usuário***

Para receber recomendações, tanto na abordagem de Filtragem Colaborativa como na Baseada em Conteúdo, os usuários deverão ter avaliado um número suficiente de itens. Só a partir daí o Sistema de Recomendação conseguirá aprender as preferências dos usuários e então conseguir recomendar itens. Com pouca informação o sistema não é capaz de fornecer recomendações confiáveis.

### **Escalabilidade**

Quando o número de usuários e itens aumenta exageradamente, algoritmos de Filtragem Colaborativa tradicionais vão sofrer com problemas sérios de escalabilidade. Considerando uma base com 10 milhões de usuários e milhões de itens, é necessário calcular a similaridade do usuário ativo com toda a base de usuários, além de predizer a avaliação para os milhões de itens que ele ainda não avaliou. Da mesma forma, a técnica Baseada em Conteúdo também precisa ser escalável para a quantidade de características que um item pode possuir, assim como a quantidade de itens que precisam ser comparados ao perfil do usuário para serem recomendados.

Os sistemas precisam reagir imediatamente às requisições *online* e fazer recomendações para todos os usuários, independentemente do seu histórico de avaliação. Técnicas de redução de dimensionalidade da matriz usuário×item são aplicadas como alternativa. Algoritmos *model-based*, usando clusterização, tratam o problema de escalabilidade procurando grupos de usuários para recomendação, diminuindo a quantidade de comparações que precisam ser realizadas ( [Su e Khoshgoftaar 2009]).

### 2.1.3 Medidas de Qualidade em Sistemas de Recomendação

[Herlocker et al. 2004] afirma que avaliar Sistemas de Recomendação e seus algoritmos é inherentemente difícil por muitas razões. Primeiro, diferentes algoritmos podem ser melhores ou piores em diferentes conjuntos de dados. Muitos algoritmos de Filtragem Colaborativa tem sido desenvolvidos especificamente para conjuntos de dados onde existem mais usuários do que itens. Tais algoritmos podem ser totalmente inapropriados em um domínio onde existem mais itens do que usuários. Diferenças similares existem para densidade de avaliações, escala de avaliações e outras propriedades de bancos de dados.

A segunda razão é que o objetivo para o qual a avaliação é realizada pode diferir. Desde cedo o trabalho de avaliação foca especificamente na acurácia dos algoritmos em predizer avaliações. Porém, quando os Sistemas de Recomendação são usados para suporte a decisão, pode ser mais valioso medir quantas vezes o sistema leva o usuário a fazer escolhas erradas.

Existe um emergente entendimento que uma boa acurácia na recomendação sozinha não permite aos usuários dos Sistemas de Recomendação uma experiência efetiva e satisfatória. Sistemas de recomendação devem prover não somente acurácia, mas também utilidade. Um Sistema de Recomendação pode alcançar alta acurácia por computar somente previsões para itens fáceis de predizer - mas esses são os itens que os usuários provavelmente não estão precisando.

[Shani e Gunawardana 2011] reforça que o uso de um Sistema de Recomendação pelas pessoas é muito mais do que uma antecipação exata de suas preferências. O usuário pode estar interessado em descobrir novos itens, em explorar rapidamente itens diversos, em respostas rápidas do sistema e outras propriedades de interação com o mecanismo de recomendação.

A seguir apresentamos várias medidas que são usadas para avaliar Sistemas de Recomendação.

#### Acurácia

[Shani e Gunawardana 2011] afirmam que a suposição básica em um Sistema de Recomendação é que um sistema que fornece previsões mais acuradas será preferido pelo usuário. A acurácia da previsão é tipicamente independente da interface do usuário e pode ser medida em um experimento offline.

Em algumas aplicações deseja-se predizer a nota que um usuário daria a um item. Em tais casos, deseja-se medir a acurácia dos sistemas em predizer notas. As medidas mais utilizadas são a *Root Mean Squared Error* (RMSE) e a *Mean Absolute Error* (MAE).

Em outras aplicações, o Sistema de Recomendação tenta recomendar aos usuários itens que ele pode usar, sem predizer notas. [Herlocker et al. 2004] destaca que métricas de classificação podem ser utilizadas para avaliar, uma vez que medem a frequência com

que o Sistema de Recomendação fez decisões corretas ou incorretas sobre um item. As métricas mais populares para esse cenário são precisão e revocação.

Precisão é definida como a razão entre o número dos itens relevantes recomendados e o número total de itens recomendados. Representa a probabilidade que um item recomendado é relevante ao usuário. Revocação é definida como a razão entre os itens relevantes recomendados e o número total de itens relevantes disponíveis. Revocação representa a probabilidade que um item relevante será recomendado. Se um algoritmo tem uma precisão de 70%, então o usuário pode esperar que, em média, 7 de cada 10 itens recomendados serão relevantes. Se um algoritmo tem revocação de 80% significa que 8 a cada 10 itens relevantes serão recomendados.

Um dos desafios primários do uso de precisão e revocação para comparar diferentes algoritmos é que precisão e revocação devem ser consideradas juntas para avaliar completamente o desempenho do algoritmo. Muitas abordagens tem combinado as duas métricas na métrica F1.

$$F1 = \frac{2 \times \text{precisão} \times \text{revocação}}{\text{precisão} + \text{revocação}}$$

A métrica de precisão sozinha pode ser apropriada se o usuário não necessita de uma lista completa de todos os itens potencialmente relevantes, tal como a tarefa de encontrar bons itens. Se a tarefa é encontrar todos os itens relevantes em uma área, então revocação passa a ser importante.

## Cobertura

Como a acurácia da predição de um Sistema de Recomendação em muitos casos aumenta com a quantidade de dados, alguns algoritmos podem prover recomendação com alta qualidade, mas somente para uma pequena porção dos itens onde se tem grande quantidade de dados ([Shani e Gunawardana 2011]).

Na maioria das vezes, o termo cobertura se refere a proporção dos itens que o Sistema de Recomendação pode recomendar. Cobertura também pode ser a proporção dos usuários para o qual o sistema pode recomendar itens. Outro significado de cobertura é o comportamento do Sistema de Recomendação em relação ao problema de *cold-start*. *Cold start* pode ser considerado como um sub problema da cobertura porque mede a cobertura do sistema sobre um específico conjunto de itens e/ou usuários.

Destacando essa última definição de cobertura para *cold-start* de itens, é possível definir um parâmetro para definir o conjunto de itens "frios". Itens frios podem ser aqueles com nenhuma avaliação ou evidência de uso, ou itens que existem no sistema por menos de um certo período de tempo, ou ainda itens que tem menos que uma determinada quantidade de avaliações.

Definido o conjunto de itens frios, mede-se a acurácia do Sistema de Recomendação

em recomendá-los. Bons resultados indicarão que o Sistema de Recomendação alcança boa cobertura, isto é, consegue recomendar de forma precisa os itens frios.

## Novidade

Recomendações novas são recomendações para itens que o usuário não conhecia e não conheceria na ausência de um sistema de recomendação ([Shani e Gunawardana 2011]). Um método para avaliar recomendações novas usa a suposição de que itens populares são menos prováveis de serem novidades. Assim, novidade pode ser considerada como uma métrica de acurácia onde o sistema não dá o mesmo peso para previsões corretas de itens populares, pesando mais itens não populares ([Ziegler et al. 2005]).

Considerando que o Sistema de Recomendação recomenda um conjunto de itens, calcula-se a popularidade de cada item recomendado para verificar como o sistema recomenda itens novos. Uma taxa alta de novidade indica que o sistema tende a recomendar os itens mais populares. Entretanto, a medida de novidade deve ser tão baixa quanto possível para um bom desempenho de novidade, indicando que o algoritmo é capaz de recomendar itens que não são tão populares ([Fouss e Saerens 2008]).

A medida de novidade implica em avaliar se o Sistema de Recomendação, além de ser acurado, consegue ter outras características que podem colaborar na satisfação do usuário com o sistema. Os itens populares são fáceis de serem alcançados pelos usuários por outros mecanismos de divulgação. Itens não tão populares tendem a ter um acesso mais difícil até o usuário final.

## *Serendipity*

*Serendipity* é a medida de quão surpreendentes as recomendações são. Por exemplo, se o usuário avaliou positivamente muitos filmes onde um determinado ator aparece, recomendar um novo filme daquele ator pode ser novo, porque o usuário pode não conhecê-lo ainda, mas não é surpreendente. Recomendações aleatórias podem ser muito surpreendentes mas é necessário balancear acurácia e *serendipity* ([Shani e Gunawardana 2011]).

Pode-se pensar a métrica *serendipity* como a quantidade de informação relevante que é nova para o usuário na recomendação. Por exemplo, se seguindo uma recomendação de filme de sucesso o usuário fica sabendo de um novo ator que ele gosta, esta pode ser considerada como uma recomendação *serendipity*.

Para calcular a *serendipity* de um Sistema de Recomendação, [Shani e Gunawardana 2011] definem uma métrica que calcula a distância do item recomendado ao conjunto de itens que o usuário já avaliou. Assim, quanto maior a distância mais recomendações surpreendentes são feitas pelo sistema. Quanto menor a distância menos recomendações surpreendentes são feitas.

Como exemplo tem-se uma aplicação de recomendação de livros. O objetivo é recomen-

dar livros de autores que o leitor é pouco familiar. Define-se uma métrica de distância entre o livro  $b$  e o conjunto de livros  $B$  (os livros que o usuário já leu).

Dado  $c_{B,w}$  o número de livros escritos pelo escritor  $w$  em  $B$ . Dado  $c_B = \max_w c_{B,w}$  o número máximo de livros de um simples escritor em  $B$ . A distância entre  $b$  e  $B$  é dada por:

$$d(b, B) = \frac{(1 + c_B - c_{B,w(b)})}{1 + c_B},$$

onde  $w(b)$  é o autor do livro  $b$ .

Essa distância é calculada para cada atributo do item recomendado. Pode-se verificar qual a característica do item que apresenta mais surpresa ao usuário, da mesma forma como se pode calcular qual o item mais surpreendente somando os valores encontrados para cada atributo do item. Dentro de um conjunto de itens recomendados é possível calcular, então, o quanto o sistema se comporta em relação à recomendações surpreendentes.

Análogo à medida de novidade, a medida de *serendipity* visa avaliar outra característica do Sistema de Recomendação, que pode colaborar com a satisfação do usuário na sua interação com o sistema.

## Outras Medidas

Muitas outras medidas para avaliação de Sistemas de Recomendação são utilizadas na literatura. Apresentamos brevemente algumas delas:

- Confiança: a probabilidade do valor predito ser realmente verdade. Se um sistema apresenta uma baixa taxa de confiança, o usuário pode tender a pesquisar mais sobre o item antes de fazer uma decisão ( [Shani e Gunawardana 2011] ).
- Robustez: a habilidade do algoritmo em fazer boas previsões na presença de dados ruidosos ( [Fouss e Saerens 2008] ).
- Diversidade: capacidade do Sistema de Recomendação sugerir itens dissimilares aos itens que o usuário já conhece ( [Ziegler et al. 2005] ).
- Tempo de recomendação: o tempo necessário para a fase de treinamento do sistema e o tempo necessário para realizar recomendações ( [Fouss e Saerens 2008] ).
- Significância estatística: a probabilidade de que os resultados foram obtidos devido à sorte ( [Shani e Gunawardana 2011] ).

## 2.2 Mineração de Preferências

Qualquer Sistema de Recomendação necessita aprender as preferências dos usuários para poder recomendar itens úteis. O primeiro passo é definir o modelo com que essas

preferências serão representadas para então conseguir minerá-las a partir do conjunto de dados disponível.

Em [Stefanidis et al. 2011] são apresentados diversos modelos de preferências utilizados na literatura. Os autores então usam critérios para categorizá-los. Entre esses critérios tem-se os de formulação e contexto.

Em relação ao critério de formulação das preferências, estas podem ser expressas de forma qualitativa ou quantitativa. Na abordagem qualitativa, as preferências entre dois itens são expressas diretamente, tipicamente, usando uma relação de preferência binária. Como exemplo de preferência qualitativa podemos apresentar a afirmativa: "O usuário prefere filmes com o ator Tom Hanks", que permite afirmar que esse usuário prefere o filme  $f_2$  ao filme  $f_3$  da tabela 2.2, ou  $f_2 > f_3$ .

Na abordagem quantitativa, as preferências são expressas através de notas atribuídas aos itens. Nesse caso, um item  $i_i$  é preferido ao item  $i_j$ , se e somente se, a nota de  $i_i$  é maior do que a de  $i_j$ . Por exemplo, o usuário poderia informar a nota 5 para o filme  $f_1$  e nota 3 para o filme  $f_3$ , sendo então  $f_1$  preferido a  $f_3$ . São diversos os trabalhos que utilizam essa representação, como o [Luo et al. 2008], [Melville et al. 2002] e [Goldberg et al. 2001].

Tabela 2.2: Filmes

Id	Gênero	Autor principal
$f_1$	Mistério	<i>Tom Hanks</i>
$f_2$	Drama	<i>Tom Hanks</i>
$f_3$	Drama	<i>Jim Carrey</i>

Em relação ao critério de contexto, as preferências podem ser contextuais ou não contextuais. As preferências contextuais são aquelas definidas sob algumas circunstâncias. Uma preferência contextual é um par  $(C, P)$  onde  $C$  define o contexto e  $P$  define a preferência. O contexto  $C$  especifica as condições sobre as quais se tem a preferência  $P$ . Um exemplo de preferência contextual seria "Entre os filmes de drama o usuário prefere aqueles com o ator Tom Hanks". Nesse exemplo o contexto é o filme ser do gênero drama e a preferência é a escolha do ator Tom Hanks em relação aos outros atores. A partir desta preferência também é possível afirmar que o usuário prefere o filme  $f_2$  ao filme  $f_3$  da tabela 2.2.

As preferências não contextuais, por sua vez, são aquelas que não apresentam condição para definir a preferência sobre os itens. Por exemplo: "prefiro filmes de mistério aos de drama", onde, entre os filmes da tabela 2.2, o  $f_1$  seria preferido aos outros dois.

Neste trabalho de dissertação nosso foco é um estudo de caso que utilizará um algoritmo de mineração de preferências contextuais qualitativas. Entretanto, a metodologia PrefRec pode ser utilizada com qualquer algoritmo de mineração de preferências, a ser discutido no capítulo 4.

### 2.2.1 Modelo de Preferências Qualitativas

[Stefanidis et al. 2011] discute que, em termos de poder de expressividade, a especificação qualitativa de preferências é mais ampla do que a quantitativa, uma vez que nem todas as relações de preferências podem ser expressas por funções de *score* ou através de níveis de interesse ou notas. Dados os filmes da tabela 2.2, como exemplo, um usuário prefere um filme a outro, se e somente se, são da mesma categoria e o primeiro filme é estrelado por Tom Hanks. Entre os itens da tabela 2.2 o usuário prefere  $f_2$  a  $f_3$ , e  $f_1$  não pode ser comparado com nenhum outro. Essa preferência não pode ser expressa quantitativamente: poderia ser dado um score maior para  $f_2$  em relação a  $f_3$ , porém nenhum valor poderia ser atribuído a  $f_1$ , por não poder ser comparado com nenhum outro.

As preferências qualitativas retornam os objetos mais preferidos sem distinguir o quanto um objeto é melhor que outro. Se torna mais fácil para o usuário expressar a relação de preferências entre pares de objetos do que determinar notas para todos eles. De qualquer forma, o usuário ao atribuir uma nota a um item o faz a partir da comparação deste com outro existente. Em contrapartida, implementar preferências qualitativas para atingir uma ordem envolvendo todos os itens do cenário é claramente complexo.

A respeito das preferências qualitativas contextuais, [Dey 2001] ressalta que o contexto é alguma informação que pode ser usada para caracterizar a situação em que o usuário se encontra e que é considerado relevante para a interação entre o usuário e a aplicação. O contexto das preferências pode ser interno ou externo ao banco de dados. O contexto interno pode ser, por exemplo, o ano de lançamento de um filme. Um contexto externo pode ser a preferência por suspense depois da meia noite. Tal informação depende do tempo e não das características do filme.

[Wilson 2004] apresenta uma definição formal para preferências qualitativas contextuais denominando-as por *regras de preferências contextuais*.

**Regra de preferência contextual.** Uma regra de preferência contextual (*cp-rule*) é uma instrução no formato  $\varphi : u \rightarrow (X = x_1) > (X = x_2)$ . Sendo que  $u$  é uma fórmula da forma  $(A_{i1} = a_{i1}) \wedge \dots \wedge (A_{ik} = a_{ik})$  e é chamada condição da *cp-rule*. Uma teoria de preferências contextuais (*cp-teoria*) é um conjunto  $\Gamma$  de *cp-rules*.

Para o usuário x, por exemplo, poderia ser obtido a *cp-teoria*  $\Gamma = \{\varphi_1, \varphi_2\}$ , onde  $\varphi_1 : (\text{ator} = \text{Tom Hanks}) \rightarrow (\text{categoria} = \text{Mistério} > \text{categoria} = \text{Drama})$ , e  $\varphi_2 : (\text{categoria} = \text{Drama}) \rightarrow (\text{ator} = \text{Jim Carrey} > \text{ator} = \text{Tom Hanks})$ . Assim, entre os filmes apresentados na tabela 2, a *cp-rule*  $\varphi_1$  infere que o filme  $f_1$  será preferido ao filme  $f_2$ , e  $\varphi_2$  infere que o filme  $f_3$  será preferido ao filme  $f_2$ .

As regras de preferências contextuais permitem expressar com riqueza de detalhes as preferências do usuário. Além de saber sobre qual característica o usuário tem preferência, permitem informar sob qual condição essa preferência acontece. É fácil observar que em nossas escolhas cotidianas, dependendo das circunstâncias, nossas preferências mudam.

Assim, as regras de preferências contextuais permitem representar com maior qualidade o comportamento do usuário.

As regras de preferências contextuais podem ser aplicadas aos Sistemas de Recomendação como recursos para tratamento do problema de *cold-start* de item. Uma vez que se conhece as preferências do usuário representadas por regras de preferências contextuais, é possível comparar itens a partir dos atributos que eles apresentam e predizer quais deles serão preferidos pelos usuários. Não há a necessidade dos itens já terem sido avaliados para serem recomendados ao usuário ativo.

No estudo de caso apresentado neste trabalho (capítulo 4), utilizamos um algoritmo que minera regras de preferências contextuais no intuito de tratar o problema de *cold-start* de item. Entretanto, qualquer outro modelo de preferência pode ser utilizado, como é discutido naquele capítulo.

### 2.2.2 Mineração de Preferências Não Contextuais

Definido o modelo de preferência a ser utilizado, é necessário um algoritmo que minere as preferências a partir dos dados disponíveis. Nesta seção apresentamos propostas de mineração de preferências qualitativas, porém não contextuais.

No trabalho apresentado em [Jiang et al. 2008] são usadas amostras de preferências fornecidas pelo usuário para inferir uma ordem sobre qualquer par de tuplas no banco de dados. Os exemplos fornecidos pelo usuário são divididos em duas classes: superior e inferior. Cada classe contém informação sobre algumas das tuplas preferidas e das não-preferidas respectivamente. O modelo de preferência aplicado é o modelo Pareto. As regras de preferências obtidas são sobre os valores dos atributos e não dependem dos valores de outros atributos.

Outros dois trabalhos são os apresentados em [Holland et al. 2003] [Chomicki 2003]. O primeiro também utiliza o modelo de preferência Pareto através de construtores especiais de preferências. As preferências são mineradas a partir de logs web e assumem a ideia de que se um item frequentemente aparece, ele é preferido pelo usuário. O segundo trabalho utiliza fórmulas lógicas que são incorporadas à álgebra relacional.

### 2.2.3 Mineração de Preferências Contextuais

A metodologia de Sistemas de Recomendação aqui proposta tem como um dos objetivos aplicar a mineração de preferências como recurso para obter melhores recomendações. Diante das vantagens apresentadas em relação às regras de preferências contextuais, este modelo é definido como foco para o estudo de caso. Assim, deseja-se um minerador de preferências que minere esse modelo de preferência para o nosso estudo de caso.

Os algoritmos de mineração de preferências contextuais apresentados nesta seção são o Profminer ( [de Amo et al. 2012b]) e o CprefMiner ( [de Amo et al. 2012a]).

O Profminer descobre perfis de usuário especificados por um conjunto de regras de preferências. Uma técnica baseada na descoberta de regras de associação é proposta mostrando alta precisão, porém, baixa revocação.

O CPrefMiner, por sua vez, extrai uma rede bayesiana do conjunto das avaliações do usuário que permite comparar, dados dois itens, qual deles é preferido a partir das características que eles apresentam. A principal vantagem do CPrefMiner sobre o ProfMiner é que ele produz um modelo compacto de preferência, que induz uma ordem parcial sobre o conjunto de itens ( [de Amo et al. 2012a] ).

No presente trabalho nosso foco é um estudo de caso que utilizará o algoritmo de mineração de preferências contextuais qualitativas CPrefMiner. A seguir é descrito com mais detalhes este minerador de preferências e como ele é capaz de comparar dois itens.

Um minerador de preferências é uma ferramenta estratégica para evitar o inconveniente do usuário informar suas preferências explicitamente. Tal processo seria tedioso e tomaria muito tempo causando indisposição do usuário para fornecer tais informações. Diante disso, o desenvolvimento de técnicas de mineração de preferências que permitam a inferência automática das preferências do usuário se torna muito relevante ( [de Amo et al. 2013a] ).

O objetivo principal do método de mineração é a habilidade de fornecer uma *relação de preferência* sobre um conjunto de dados. Uma *relação de preferência* sobre um conjunto finito de objetos  $A = a_1, a_2, \dots, a_n$  é uma ordem parcial estrita sobre  $A$ , isto é, uma relação binária  $R \subseteq A \times A$  satisfazendo as propriedades de transitividade e irreflexividade. Tipicamente, uma ordem parcial estrita é representada pelo símbolo  $>$ . Então se  $>$  é a relação de preferência, nós denotamos por  $a_1 > a_2$  o fato de  $a_1$  ser preferido a  $a_2$ .

Seja  $R(A_1, A_2, \dots, A_n)$  um esquema relacional. Seja  $Tup(R)$  o conjunto de todas as tuplas sobre  $R$ . Um *banco de dados de preferência* sobre  $R$  é um conjunto finito  $\mathcal{P} \subseteq Tup(R) \times Tup(R)$  que seja consistente, isto é, se  $(u, v) \in \mathcal{P}$  então  $(v, u) \notin \mathcal{P}$ . O par  $(u, v)$ , chamado de bitupla, representa o fato que o usuário prefere a tupla  $u$  à tupla  $v$ .

A figura 2.2 (a) apresenta uma instância  $I$  sobre  $R$  e em (b) ilustra um banco de dados de preferência sobre  $R$  fornecida pelo usuário indicando suas preferências sobre as tuplas de  $I$ .

O objetivo do CPrefMiner é extrair um *modelo de preferência contextual* a partir de um banco de dados de preferência fornecida pelo usuário. O modelo de preferência contextual é especificado, no CPrefMiner, por uma *Rede de Preferência Bayesiana*.

Uma *Rede de Preferência Bayesiana* ou (BPN) sobre um esquema relacional  $R(A_1, \dots, A_N)$  é um par  $(G, \theta)$ , onde  $G$  é um grafo acíclico direcionado onde os nós são atributos em  $R$  e as arestas representam dependências entre os atributos.  $\theta$  é o mapeamento que associa cada nó de  $G$  a uma tabela de probabilidades condicionais, isto é, um conjunto finito de probabilidade condicionais da forma  $P[E_2|E_1]$  onde  $E_1$  é um evento da forma  $(A_{i_1} = a_{i_1}) \wedge \dots \wedge (A_{i_k} = a_{i_k})$ , tal que  $\forall j \in 1, \dots, k; a_{i_j} \in \text{dom}(A_{i_j})$ , e  $E_2$  é um

Id	A	B	C	D	
$t_1$	$a_1$	$b_1$	$c_1$	$d_1$	$(t_1, t_2)$
$t_2$	$a_1$	$b_1$	$c_1$	$d_2$	$(t_1, t_3)$
$t_3$	$a_2$	$b_1$	$c_1$	$d_2$	$(t_4, t_5)$
$t_4$	$a_1$	$b_2$	$c_1$	$d_2$	$(t_4, t_2)$
$t_5$	$a_2$	$b_1$	$c_2$	$d_1$	$(t_5, t_6)$
$t_6$	$a_3$	$b_1$	$c_1$	$d_1$	$(t_3, t_5)$
					$(t_4, t_1)$

(a)

(b)

Figura 2.2: (a) Uma instância I e (b) um banco de dados de preferência P [de Amo et al. 2012a]

evento da forma " $(B = b_1)$  é preferido a  $(B = b_2)$ ", ou simplesmente,  $b_1 > b_2$ , onde  $B$  é um atributo de  $R$ ,  $B \neq A_{i_j}, \forall j \in 1, \dots, k$  e  $b_1$  e  $b_2 \in \text{dom}(B)$ , sendo  $b_1 \neq b_2$ .

A figura 2.3 abaixo apresenta um exemplo de rede de preferência sobre  $R(A, B, C, D)$ .

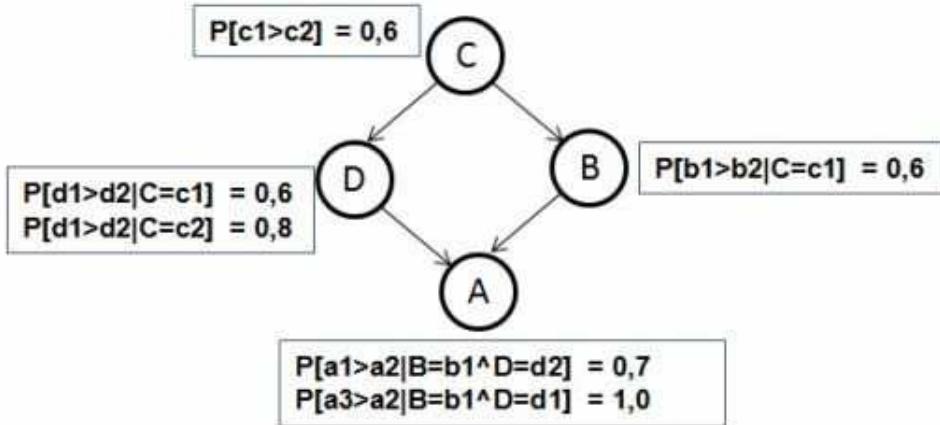


Figura 2.3: Rede de preferência PNet1 [de Amo et al. 2012a]

Cada probabilidade condicional em uma BPN representa uma regra de preferência contextual probabilística (*cp – rule*), onde o evento  $E_1$  é o contexto e o evento  $E_2$  é a preferência. A regra de preferência contextual probabilística associada ao nó  $X$  no grafo  $G$  representa o nível de confiança de preferir alguns valores de  $X$  em relação a outros, dependendo dos valores assumidos por seus pais no grafo. Por exemplo, como apresentado na figura 2.3,  $P[d1 > d2 | C = c1] = 0.6$  significa que a probabilidade da tupla com  $D = d1$  ser preferida a  $D = d2$  é 60%, dado que  $C = c1$ .

BPNs são usadas para comparar pares de tuplas. Como exemplo, usando a BPN da figura 2.3, e dadas as duas tuplas  $u_1 = (a_1, b_1, c_1, d_1)$  e  $u_2 = (a_2, b_2, c_1, d_2)$ , realiza-se a comparação dessas duas tuplas seguindo os seguintes passos:

- Seja  $\Delta(u_1, u_2)$  o conjunto de atributos onde  $u_1$  difere de  $u_2$ . No exemplo,  $\Delta(u_1, u_2) = \{A, B, D\}$ .

- Seja o  $\min(\Delta(u_1, u_2)) \subseteq \Delta$  tal que os atributos em  $\min(\Delta)$  não tem ancestrais em  $\Delta$  (de acordo com o grafo  $G$ ). No exemplo  $\min(\Delta(u_1, u_2)) = \{B, D\}$ . No sentido de  $u_1$  ser preferida a  $u_2$  é necessário e suficiente que  $u_1[D] > u_2[D]$  e  $u_1[B] > u_2[B]$ .
- Calcula-se a probabilidade  $p_1$  que é a probabilidade de  $u_1 > u_2 = P[d1 > d2|C = c1] * P[b1 > b2|C = c1] = 0.6 * 0.6 = 0.36$ ;
- Calcula-se a probabilidade  $p_3$  que é a probabilidade de  $u_2 > u_1 = P[d2 > d1|C = c1] * P[b2 > b1|C = c1] = 0.4 * 0.4 = 0.16$ ;
- Calcula-se a probabilidade  $p_2$  que é a probabilidade de  $u_1$  e  $u_2$  serem incomparáveis  $= P[d1 > d2|C = c1] * P[b2 > b1|C = c1] + P[d2 > d1|C = c1] * P[b1 > b2|C = c1] = 0.6 * 0.4 + 0.4 * 0.6 = 0.48$ ;

Para comparar  $u_1$  e  $u_2$  foca-se somente em  $p_1$  e  $p_3$  (ignorando o nível de incomparabilidade  $p_2$ ) e selecionamos o maior deles. No exemplo, como  $p_1 > p_3$  infere-se que  $u_1$  é preferido a  $u_2$ . Se  $p_1 = p_3$  conclui-se que  $u_1$  e  $u_2$  são incomparáveis.

O algoritmo CPrefMiner consiste na construção da rede bayesiana em duas fases: (1) a construção do grafo e (2) o cálculo do conjunto dos parâmetros  $\theta$  representando as probabilidades condicionais do modelo. O trabalho adota uma abordagem baseada em algoritmo genético para a aprendizagem da estrutura.

Resumindo, a partir de um banco de dados de preferência, o CPrefMiner gera uma rede bayesiana de preferência que é capaz de comparar, dados dois itens, qual deles é preferido. O CPrefMiner trabalha com a representação qualitativa das preferências a partir do modelo de regras de preferências contextuais. Diferente das redes bayesianas *naive bayes*, amplamente utilizadas na abordagem Baseada em Conteúdo (seção 2.1.1), a BPN gerada pelo CPrefMiner leva em consideração a dependência entre os atributos: a preferência por um atributo pode depender do valor de outro atributo.

Vale ressaltar que essa versão do CPrefMiner trabalha com atributos monovalorados dos itens, isto é, tendo como exemplo um filme, se este possuir mais de um gênero é dado um nome ao conjunto de gêneros que ele possui.

## Técnica de Otimização de Algoritmos de Mineração de Preferências

No artigo [de Amo et al. 2013b] é introduzida a técnica de Range Voting para otimização de algoritmos de mineração de preferências. Esta técnica pode ser aplicada a qualquer minerador de preferências que produza uma função  $PREF : \mathcal{I} \times \mathcal{I} \rightarrow [0, 1]$ , onde  $\mathcal{I}$  é o conjunto de itens a serem recomendados.

A função  $PREF$  associa a um par  $(a, b)$  o valor 0 (zero) caso o minerador de preferência infira que o item  $b$  seja preferido ao item  $a$ ; o valor 1 (um) caso o item  $a$  seja preferido ao item  $b$ ; ou indiferente quando o minerador não é capaz de comparar os dois itens.

Um minerador de preferências pode ter baixa revocação quando não consegue comparar muitos pares de itens (bituplas). Para melhorar o seu desempenho, a técnica de

votação *Range Voting* pode ser associada ao minerador, a fim de tentar inferir preferências para os pares de itens que o minerador não foi capaz de comparar, assim como confirmar ou retificar as preferências inferidas pelo minerador.

A técnica de *Range Voting* consiste na ideia de votação por notas, isto é, dado um conjunto de candidatos calcular qual deles seria "eleito" a partir das notas dadas a cada um. O candidato com maior nota ganha a eleição. No nosso contexto, os candidatos são os itens e os votos são os valores retornados pela função  $PREF$  do minerador de preferência que está sendo utilizado. A técnica de *Range Voting* é aplicada, então, a cada par possível dentro do conjunto  $\mathcal{I}$ .

Dados dois itens  $(t, u)$ , a técnica de *Range Voting* verifica qual deles foi mais vezes preferido em relação aos outros itens do conjunto  $\mathcal{I}$  a partir da função  $\mathcal{P}$  abaixo. O resultado dessa função é dado dentro do intervalo  $[0,1]$ . Caso  $\mathcal{P} > 0.5$ , então infere-se que  $t > u$ . Caso  $\mathcal{P} < 0.5$ ,  $u > t$ . Caso  $\mathcal{P} = 0.5$ ,  $(t, u)$  são incomparáveis pela técnica de *Range Voting*.  $T$  é o conjunto  $\mathcal{I}$  excetuando os itens que estão sendo comparados  $(t, u)$ .

$$\mathcal{P}(t, u) = \frac{\sum_{v \in T} (PREF(t, v) - PREF(u, v) + 1)}{2 \cdot |T|}$$

Para exemplificar, seja o conjunto de itens  $I = \{i_1, i_2, i_3, i_4, i_5\}$ . Um determinado minerador de preferências retornou os seguintes resultados da tabela 2.3:

Tabela 2.3: Preferências inferidas pelo minerador

$PREF(i_1, i_2) = 1$	$(i_1 > i_2)$
$PREF(i_1, i_3) = 1$	$(i_1 > i_3)$
$PREF(i_1, i_4) = Indiferente$	$(i_1 \neq i_4)$
$PREF(i_1, i_5) = 0$	$(i_1 < i_5)$
$PREF(i_2, i_3) = 1$	$(i_2 > i_3)$
$PREF(i_2, i_4) = 0$	$(i_2 < i_4)$
$PREF(i_2, i_5) = 0$	$(i_2 < i_5)$
$PREF(i_3, i_4) = 1$	$(i_3 > i_4)$
$PREF(i_3, i_5) = Indiferente$	$(i_3 \neq i_5)$
$PREF(i_4, i_5) = Indiferente$	$(i_4 \neq i_5)$

Tomemos o par  $(i_3, i_5)$  da tabela 2.3 que o minerador não conseguiu comparar. Para saber qual entre os dois itens é preferido aplicamos a função  $\mathcal{P}$ :

$$\mathcal{P}(i_3, i_5) = (PREF(i_3, i_1) - PREF(i_5, i_1) + 1) = 0 - 1 + 1 = 0$$

$$\mathcal{P}(i_3, i_5) = 0 + (PREF(i_3, i_2) - PREF(i_5, i_2) + 1) = 0 + 0 - 1 + 1 = 0$$

$$\mathcal{P}(i_3, i_5) = 0 + (PREF(i_3, i_4) - PREF(i_5, i_4) + 1) = 0 + 1 - 0.5 + 1 = 2.5$$

$$\mathcal{P}(i_3, i_5) = 2.5/10 = 0.25$$

Como  $\mathcal{P}(i_3, i_5) < 0.5$  infere-se que  $i_5 > i_3$  a partir da técnica de *Range Voting*. Vale

observar que para calcular  $\mathcal{P}(i_3, i_5)$  foi necessário saber o valor de  $PREF(i_5, i_4)$  que o minerador não foi capaz de comparar. Quando isso acontece é dado o valor de 0.5 para a função  $PREF$ . A tabela 2.4 apresenta o resultado da técnica Range Voting aplicada a todos os pares da tabela 2.3.

Tabela 2.4: Preferências inferidas com a técnica de Range Voting

$\mathcal{P}(i_1, i_2) = 0.35$	$(i_1 < i_2)$
$\mathcal{P}(i_1, i_3) = 0.30$	$(i_1 < i_3)$
$\mathcal{P}(i_1, i_4) = 0.35$	$(i_1 < i_4)$
$\mathcal{P}(i_1, i_5) = 0.35$	$(i_1 < i_5)$
$\mathcal{P}(i_2, i_3) = 0.15$	$(i_2 < i_3)$
$\mathcal{P}(i_2, i_4) = 0.30$	$(i_2 < i_4)$
$\mathcal{P}(i_2, i_5) = 0.20$	$(i_2 < i_5)$
$\mathcal{P}(i_3, i_4) = 0.15$	$(i_3 < i_4)$
$\mathcal{P}(i_3, i_5) = 0.15$	$(i_3 < i_5)$
$\mathcal{P}(i_4, i_5) = 0.20$	$(i_4 < i_5)$

Nesse exemplo, a técnica Range Voting consegue inferir preferências para os pares não comparados pelo minerador, além de retificar as inferências do minerador nos pares  $(i_1, i_2)$ ,  $(i_1, i_3)$ ,  $(i_2, i_3)$ ,  $(i_3, i_4)$ . Observemos, por exemplo, que o minerador inferiu na tabela 2.3 que  $i_2 > i_3$  e que  $i_3 > i_4$ . Por transitividade, poderíamos afirmar que  $i_2 > i_4$ , porém, o minerador inferiu o contrário. No exemplo, a técnica Range Voting infere que  $i_2 < i_3$  e  $i_3 < i_4$ , mantendo a transitividade de que  $i_2 < i_4$ .

Pode-se entender que a técnica de Range Voting utiliza as inferências feitas pelo minerador para reforçar as preferências entre os pares de itens. Dessa forma, consegue predizer inferências para os pares que o minerador não foi capaz de comparar, além de corrigir outras inferências feitas. As características desta técnica de votação permitem que a precisão do minerador seja melhor, além de diminuir a quantidade de pares incomparáveis, aumentando a revocação do minerador de preferências ([de Amo et al. 2013b]).

Esta técnica pode ser aplicada a qualquer algoritmo de mineração de preferências. Assim, sempre que se testar um algoritmo A, pode-se testar sua versão otimizada A\*.



# Capítulo 3

## Trabalhos Correlatos

A pesquisa na área de Sistemas de Recomendação é relativamente recente, comparada com outras áreas. Os trabalhos considerados como precursores, [Resnick et al. 1994] e [Goldberg et al. 1992], datam dos últimos 20 anos. Nesse período muita pesquisa foi desenvolvida, buscando sempre alcançar recomendações mais acuradas.

Neste capítulo apresentamos diversos trabalhos de Sistemas de Recomendação e os separamos nas seguintes subseções: Filtragem Colaborativa, Baseada em Conteúdo, abordagens híbridas e trabalhos que utilizam técnicas de mineração de dados.

### 3.1 Abordagens de Filtragem Colaborativa

O trabalho inicial [Resnick et al. 1994] foi aplicado no cenário de notícias, onde o usuário começava a receber centenas delas e nem todas eram do seu interesse. O GroupLens, nome dado ao Sistema de Recomendação, conseguia filtrar entre as notícias quais seriam as de interesse do usuário ativo, através da predição de notas para os itens, usando dados de outros usuários. Mas como já discutido na seção 2.1.1, a abordagem proposta sofre com os problemas de esparsidade dos dados, *cold-start* de item e usuário e também com a escalabilidade da aplicação.

Em [Herlocker et al. 1999] é feito um estudo sobre várias características do sistema GroupLens. A principal contribuição foi a definição da significância ponderada. Este conceito define que, para um usuário ser similar a outro, deve-se levar em conta a quantidade de itens que eles avaliaram em comum. Na medida de similaridade PCC (seção 2.1.1), se dois usuários avaliaram de forma igual um único item, eles são altamente correlacionados. Com isto, pode-se gerar recomendações pobres. Assim, neste trabalho é definido um parâmetro para informar quantos itens no mínimo um usuário deve ter avaliado em comum com o usuário ativo para ser considerado como semelhante e influenciar as recomendações.

Outra contribuição apresentada por [Herlocker et al. 1999] é relacionada a seleção dos vizinhos. Chegou-se a conclusão de que o aumento do número de vizinhos nem sempre

aumenta a acurácia da predição. Assim, devem ser selecionados os melhores vizinhos, os mais similares, para realizar a recomendação.

Outro trabalho de Filtragem Colaborativa é o apresentado em [Goldberg et al. 2001] que já explora a ideia *model-based*. O cenário daquele trabalho é o de recomendação de piadas e procura-se tratar, principalmente, os problemas de esparsidade dos dados e de *cold-start* de usuário. Para isso, é solicitado que todo novo usuário ao entrar no sistema avalie um determinado conjunto de itens (*gauge set*). Desta forma, o sistema obtém uma matriz usuário×item densa.

Cada usuário é representado pelo seu vetor de avaliação reduzido a duas coordenadas ( $x, y$ ) através da técnica de redução de dimensionalidade PCA (*Principal Component Analysis*). Os usuários são projetados como pontos em um plano cartesiano e clusterizados a partir da técnica de recursão retangular, como apresentado na figura 3.1.

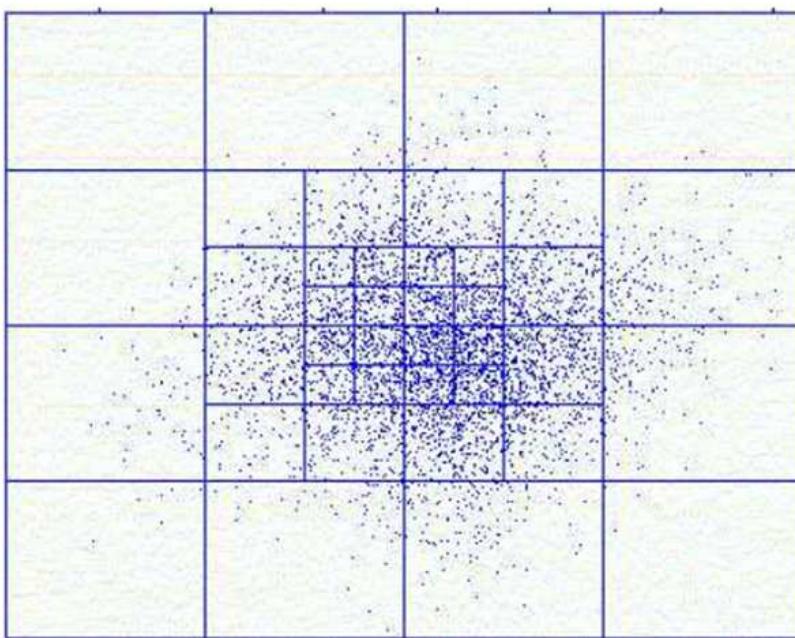


Figura 3.1: Clusterização dos usuários em [Goldberg et al. 2001] com 4 níveis de recursão formando 40 clusters.

Cada quadrado na figura representa um cluster e cada ponto um usuário. Para cada cluster identifica-se quais foram os itens avaliados pelos seus usuários, construindo uma lista de recomendação de itens. Um novo usuário, ao acessar o sistema, informa suas avaliações sobre os itens do *gauge set*, tem seu vetor de avaliações reduzido e é projetado nas dimensões  $x$  e  $y$ . Assim, ele é associado a algum cluster e recebe deste a lista de recomendação de itens já calculada.

Este trabalho alcança em seus experimentos resultados melhores que a abordagem tradicional [Resnick et al. 1994]. Porém, a definição dos itens que pertencem ao *gauge set* se torna uma tarefa difícil já que é a partir das notas dadas a estes itens que o usuário será representado no plano cartesiano. Além disso, um item só será recomendado ao

usuário ativo se já tiver sido avaliado pelos outros usuários, permanecendo o problema de *cold-start* de item.

Já no trabalho de [Huang et al. 2004] procura-se tratar o problema de esparsidade dos dados procurando mais vizinhos para o usuário ativo. O recurso utilizado é a representação dos usuários e itens por um grafo bipartido como apresentado na figura 3.2

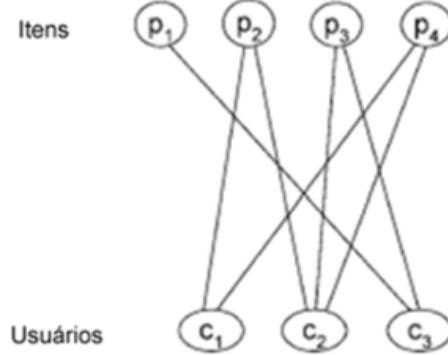


Figura 3.2: Grafo baseado em usuários e itens em [Huang et al. 2004]

O grafo é bipartido em dois conjuntos de nós: um conjunto representa os usuários e o outro os itens. As arestas entre os nós só existem entre os dois conjuntos e representam a avaliação daquele usuário para aquele item.

Um item pode ser recomendado a um determinado usuário caso exista um caminho no grafo que os conecte. Como na figura 3.2 o usuário  $c_1$  não avaliou o item  $p_3$ , mas existe um caminho no grafo que conecta os dois,  $p_3$  pode ser recomendado ao usuário  $c_1$ . A proposta do trabalho é encontrar caminhos no grafo de tamanhos diferentes de três para encontrar mais itens a serem recomendados ao usuário.

Quanto maior o caminho entre um item e um usuário, menor a nota que será predita ao item. Quanto menor o caminho entre os dois, maior a nota que será predita. Quanto maior a quantidade de caminhos conectando um item a um usuário, mais forte é a associação entre eles e assim o item será um forte candidato a ser recomendado para o usuário.

Mesmo tratando o problema de esparsidade dos dados o trabalho ainda sofre com os problemas de *cold-start* de item e usuário. Por ser uma abordagem *memory-based*, também vai sofrer do problema de escalabilidade.

O trabalho apresentado em [Ma et al. 2007] também trata o problema de esparsidade dos dados, aplicando uma abordagem híbrida entre *item-based* e *user-based*. O primeiro passo, *offline*, é predizer os dados faltosos: para tentar predizer a nota faltosa do usuário  $u_2$  para o item  $i_1$  define-se o conjunto de usuários similares a  $u_2$  e o conjunto de itens similares a  $i_1$ . A partir dos valores dos usuários e itens similares informados é calculada a predição da nota faltosa.

No momento da recomendação ao usuário ativo tenta-se predizer a nota que ele daria a um determinado item a partir dos usuários similares ao usuário ativo, e dos itens similares ao item em questão a partir dos dados já existentes e preditos no primeiro passo. Caso

não haja usuários nem itens similares é predita uma nota baseada na média das notas que o usuário ativo já informou com a média das notas que o item já recebeu. O problema de falta de dados é minimizado, porém, são mantidos os problemas de *cold-start* de item e de usuário.

Em [Luo et al. 2008] é proposta uma nova definição de similaridade entre os usuários para alcançar mais acurácia nas recomendações. Eles definem que dois usuários são similares se, em relação a avaliação média daquele item, eles avaliaram de forma semelhante. Essa similaridade é denominada similaridade local.

A partir dessa similaridade é construído um grafo onde os nós são os usuários e as arestas indicam que um usuário é similar localmente a outro, sendo o peso da aresta o valor da similaridade entre os dois. A partir do grafo são calculados usuários similares globalmente, isto é, similares por transitividade. O valor da similaridade é calculado a partir dos pesos das arestas que ligam os dois usuários.

Encontrados os usuários similares localmente e globalmente o usuário ativo possui mais vizinhos que vão colaborar para a recomendação, minimizando o problema de esparsidade de dados. Os dois tipos de vizinhos são usados no cálculo da predição, porém, são dados pesos diferentes aos usuários similares localmente e globalmente. Observou-se nos experimentos que quanto mais os dados são esparsos, mais a similaridade global auxilia na predição. Quando há pouca esparsidade, os usuários similares localmente já alcançam bons resultados.

A proposta alcança resultados mais acurados, porém, ainda sofre com o problema de *cold-start* de item e usuário. O problema da esparsidade é minimizado, mas ainda pode afetar o sistema se poucos itens tiverem sido avaliados: poucos itens serão recomendados. Tratando-se de uma abordagem *memory-based*, acaba sofrendo também do problema de escalabilidade.

## 3.2 Abordagens Baseadas em Conteúdo

A abordagem Baseada em Conteúdo também vem sendo explorada nos últimos 20 anos. O trabalho apresentado em [Lieberman 1995] apresenta um Sistema de Recomendação denominado *Letizia* implementado como uma extensão de um navegador web que monitora o comportamento de navegação do usuário e constrói um modelo personalizado, que consiste nas palavras chaves relacionadas aos interesses do usuário.

De forma semelhante, o sistema *Personal WebWatcher*, apresentado em [Mladenic 1999], aprende os interesses individuais do usuário a partir das páginas web que ele visita e dos documentos ligados a essas páginas. Entre esses documentos, os visitados são considerados como exemplos positivos, e os não visitados como exemplos negativos de preferência do usuário.

O sistema *YourNews* ([Ahn et al. 2007]), por sua vez, mantém perfis separados de

interesse para oito diferentes tópicos de notícias (Nacionais, Mundo, Negócios, etc.). O perfil de interesse do usuário para cada tópico é representado como um vetor de pesos para os termos extraídos das notícias vistas pelo usuário. Um número específico de artigos vistos para cada tópico pelo usuário são coletados, e os 100 (cem) termos mais vistos são extraídos para gerar o vetor final do perfil do usuário.

Ambos os trabalhos representam o perfil do usuário baseado nas palavras chaves extraídas dos documentos analisados. Segundo [Lops et al. 2011] essa representação para ambos itens e perfis alcança Sistemas de Recomendação acurados desde que se tenha disponível quantidade suficiente de informação sobre os interesses do usuário. O problema dessa técnica é a denominada "falta de inteligência" quando características avançadas são necessárias, por exemplo, quando o usuário procura por sinônimos ou referências. O próprio autor dá o seguinte exemplo: se o usuário gosta de artes "impressionistas francesas", o Sistema de Recomendação que utiliza palavras chaves só retornará itens que apresentem exatamente essas palavras. Itens relacionados a Claude Monet ou Renoir não aparecerão entre os itens recomendados, mesmo sendo de interesse do usuário.

Dessa forma, outros trabalhos procuram analisar semânticamente as palavras chaves a partir de ontologias. O trabalho apresentado em [Blanco-Fernández et al. 2008], por exemplo, propõe um Sistema de Recomendação para televisão digital. Este sistema aplica técnicas de raciocínio para comparar as preferências do usuário com os itens (programas de televisão) de uma forma mais flexível. Os programas de televisão são descritos por metadados que descrevem detalhadamente os seus principais atributos. O conhecimento sobre os itens e os perfis do usuário são representados usando a ontologia OWL, própria para definir ontologias na Web. Os perfis baseados em ontologia fornecem uma representação formal das preferências do usuário, sendo capaz de raciocinar sobre elas e descobrir conhecimento extra sobre os seus interesses e assim prover recomendações.

Outros trabalhos de abordagem Baseada em Conteúdo tentam minimizar o problema de *over specialization*. O trabalho apresentado em [Billsus e Pazzani 2000] descarta os itens a serem recomendados se eles forem muito similares ao que o usuário já avaliou. [Iaquinta et al. 2008], por sua vez, recomenda itens que são pouco similares ao perfil do usuário. Os autores assumem que, quanto menor a probabilidade de um usuário conhecer um item, mais alta é a probabilidade daquele item ser uma recomendação surpreendente.

### 3.3 Abordagens Híbridas

Como afirmado em [Amatriain 2013], na prática, a maioria dos Sistemas de Recomendação avançados usados na indústria são baseados em algum tipo de hibridação, e raramente são puramente Filtragem Colaborativa ou Baseada em Conteúdo. Os problemas apresentados em um tentam ser minimizados pelas vantagens apresentadas no outro. Nesta seção apresentamos alguns trabalhos de abordagem híbrida.

Em [Degemmis et al. 2007] os usuários são representados pelos perfis construídos a partir da abordagem Baseada em Conteúdo. Os usuários são então clusterizados a partir da similaridade entre os seus perfis, não precisando ter avaliado itens em comum para pertencerem ao mesmo grupo. Para recomendar para um usuário ativo, o seu perfil baseado em conteúdo também é construído e comparado ao centróide dos grupos existentes. Os usuários do grupo que o usuário ativo for mais similar serão os que colaborarão no momento da recomendação. Além disso, ele incorpora o uso de conhecimento linguístico no processo de aprendizagem semântica do perfil do usuário. Perfis semânticos são obtidos integrando algoritmo de *naive bayes* e preferências explícitas dos usuários, com uma estratégia baseada exclusivamente em conhecimento lexical armazenado em um banco de dados denominado WordNet.

Ao clusterizar os usuários, os autores conseguem minimizar o problema de escalabilidade. A maneira de calcular a similaridade entre os usuários demonstra, a partir dos experimentos realizados, bons resultados de acurácia. Entretanto, os problemas de *cold-start* permanecem, uma vez que a abordagem Baseada em Conteúdo é utilizada para identificar os vizinhos e não para recomendar os itens.

Em [Ko e Lee 2002] é apresentado outro Sistema de Recomendação em que as duas abordagens são feitas paralelamente e são mescladas no momento da recomendação. O perfil do usuário é criado usando o método Baseada em Conteúdo, de modo que o usuário faz uma consulta e os itens encontrados são usados para construir seu perfil. Na técnica Filtragem Colaborativa é utilizado um classificador bayesiano para diminuir as dimensões dos itens e um algoritmo genético para clusterizar os usuários. Para cada cluster é construída uma lista de recomendação dos itens que os usuários do grupo avaliaram.

As preferências de um novo usuário são aprendidas usando o perfil do usuário criado no Baseada em Conteúdo e nos perfis de grupos criados no Filtragem Colaborativa. O usuário, primeiramente, deve ser classificado em algum grupo a partir da similaridade do seu vetor de avaliação com os vetores dos usuários de determinado grupo. É recomendado ao usuário a lista de recomendação do grupo que ele for mais similar e os itens que foram recomendados pela técnica Baseada em Conteúdo.

Em [Melville et al. 2002] é apresentado o *Content-boosted collaborative filtering* (CBCF). Este trabalho também apresenta um Sistema de Recomendação híbrido entre as abordagens de Filtragem Colaborativa e Baseada em Conteúdo. Em um primeiro momento, por meio de um processamento *offline*, é aplicada a abordagem Baseada em Conteúdo para predição dos dados faltantes, buscando obter uma matriz usuário×item densa.

Para cada usuário obtém-se o conjunto de itens já avaliados e aplica-se o algoritmo *naive bayes* a fim de predizer as notas para os itens que ele ainda não avaliou. Como apresentado na seção 2.1.1, o *naive bayes* calcula a probabilidade de determinado valor de atributo acontecer em determinada classe (nota). Dado um item ainda não avaliado pelo

usuário, calcula-se a probabilidade dos valores de atributos que o item apresenta pertencer a alguma classe. A classe com maior probabilidade é a classe (nota) predita para aquele item. Caso exista algum valor de atributo no item a ser predito que não existe entre os itens já avaliados, é atribuído um valor de probabilidade mínima. Qualquer item consegue receber um valor predito.

Ao final da tarefa *offline*, onde aplica-se a abordagem Baseada em Conteúdo, é obtida uma matriz densa de usuário×item. No momento da recomendação, dado o vetor de avaliações do usuário ativo, é calculado quais são seus vizinhos mais similares. Os vetores dos vizinhos similares são formados por valores reais e valores preditos pela abordagem Baseada em Conteúdo.

Ao cálculo de recomendação, os autores acrescentam algumas contribuições para alcançar melhores resultados. O valor de similaridade do vizinho depende da quantidade de valores preditos e reais que ele possui. Quanto mais valores preditos, menor a sua similaridade. O usuário ativo também é considerado como um vizinho, e também são preditas as avaliações a partir da abordagem Baseada em Conteúdo para o item que se está tentando recomendar no momento, a partir das avaliações já conhecidas.

Observa-se que o CBCF consegue tratar o problema de *cold-start* de item, uma vez que qualquer item terá alguma avaliação, ou real ou predita pela abordagem Baseada em Conteúdo. Quando um novo item é inserido no sistema são preditas as avaliações em relação a ele para cada usuário do sistema, até que existam avaliações reais. A partir desse momento ele já pode ser recomendado, por possuir avaliações de todos os usuários do sistema.

O CBCF consegue predizer avaliações para qualquer item a partir da abordagem Baseada em Conteúdo, para quando for aplicada a Filtragem Colaborativa o Sistema de Recomendação ser capaz de recomendar o item.

O CBCF se torna pouco escalável por aplicar a categoria *memory-based* de Filtragem Colaborativa, isto é, no momento da recomendação é necessário calcular a similaridade do usuário ativo com todos os outros usuários do sistema, além de predizer para o próprio usuário ativo, que também se torna um vizinho, a predição para aquele item através da abordagem Baseada em Conteúdo.

## 3.4 Abordagens que Utilizam Mineração de Dados

Em [Amatriain et al. 2011] é discutido que a maioria dos Sistemas de Recomendação possui como núcleo um algoritmo que pode ser entendido como uma técnica de mineração de dados. Nesta seção vamos ressaltar a utilização desta técnica em diferentes trabalhos.

Uma das técnicas de mineração utilizadas é a técnica de classificação. Um classificador é um mapeamento entre o conjunto de características a um conjunto de classes. Um Sistema de Recomendação para um restaurante, por exemplo, pode ser implementado por

um algoritmo de classificação que classifica restaurantes em duas categorias (bom, ruim) baseado no número de características que o descrevem. Entre os algoritmos de classificação são utilizados os algoritmos *kNN*, árvores de decisão, classificadores bayesianos, redes neurais e *svm*. O mais utilizado para abordagens de Filtragem Colaborativa é o *k-Nearest Neighbor* (*kNN*), uma vez que ele captura de forma transparente a suposição de opiniões semelhantes: ele opera encontrando, para cada usuário (ou item), um número de usuários similares (itens) cujos perfis podem ser usados diretamente para calcular recomendações.

O trabalho apresentado em [Bouza et al. 2008] constrói uma árvore de decisão usando informação semântica disponível para os itens. A árvore é construída depois do usuário ter avaliado somente dois itens. As características de cada item são usadas para construir um modelo que explica as avaliações do usuário. Mesmo sendo a abordagem interessante a partir de uma perspectiva teórica, a precisão do sistema é pior do que a recomendação por média de avaliações.

Já os classificadores bayesianos trazem benefícios por serem robustos para isolar ruídos e atributos irrelevantes, e lidam com valores faltantes ignorando a instância durante o cálculo da probabilidade. Em [Pronk et al. 2007] é utilizado um classificador *naive bayes* como base para incorporar controle do usuário e aumentar performance, especialmente em situações *cold-start*. Para isso ele propõe manter dois perfis para cada usuário: um aprendido do histórico de avaliações e outro explicitamente criado pelo usuário. A mistura dos dois classificadores pode ser controlada de tal maneira que o perfil definido pelo usuário é utilizado nos primeiros estágios, quando não se tem muita informação, e os classificadores aprendidos são usados em estágios posteriores. Classificadores bayesianos são bastante utilizados em abordagens com características Baseada em Conteúdo, como os já mencionados [Melville et al. 2002] e [Ko e Lee 2002].

Uma das técnicas utilizadas para tratar o problema de escalabilidade é a clusterização. A abordagem de Filtragem Colaborativa envolve muitos cálculos de similaridade para identificar os usuários mais similares ao usuário ativo. A clusterização agrupa os usuários existentes diminuindo a quantidade de operações de similaridade e aumentando a eficiência. Porém, não se pode afirmar da mesma forma que vá melhorar a acurácia do algoritmo.

[Xue et al. 2005], apresenta um uso típico de clusterização, utilizando o algoritmo *k-means* como uma etapa de pré-processamento, para ajudar a encontrar os usuários similares ao usuário ativo. O método possui um desempenho pouco melhor do que o padrão de Filtragem Colaborativa apresentada em [Resnick et al. 1994] que é baseado em *kNN*. [Goldberg et al. 2001], também usa a clusterização para diminuir a quantidade de comparações e obter a lista de itens que será recomendada ao usuário ativo.

Enfim, o que pode ser observado é que as técnicas de mineração de dados atuam como importante fator no campo de Sistemas de Recomendação ao conseguirem tratar uma grande quantidade de dados e, assim, prover recursos para melhores recomendações.

# Capítulo 4

## A Metodologia PrefRec para Sistemas de Recomendação

Apresentamos neste capítulo a Metodologia PrefRec para desenvolvimento de Sistemas de Recomendação, utilizando algoritmos de mineração de preferências. A metodologia propõe uma abordagem híbrida entre as categorias de Sistemas de Recomendação Baseada em Conteúdo e Filtragem Colaborativa, no intuito de agregar as vantagens que cada uma traz.

A primeira etapa da metodologia PrefRec consiste em representar os usuários já existentes no banco de dados para que estes possam colaborar na recomendação ao usuário ativo. A segunda etapa procura usuários com preferências semelhantes e os organiza em grupos. Na terceira etapa uma representação do gosto de todos os seus usuários é calculada para cada grupo, isto é, um consenso. A partir deste consenso, na quarta etapa, aplica-se um minerador de preferências que permita obter as preferências do grupo sobre as características dos itens a serem recomendados.

Da primeira à quarta etapa, o processamento é feito *offline* e tem como resultado um modelo de recomendação composto por *perfis de preferências*. Um *perfil de preferência* é um par  $\theta = (C, M)$  onde  $C$  é a representação do consenso de um grupo (Etapa 3) e  $M$  é o seu respectivo modelo de preferências, que, dados dois objetos, permite inferir qual dos dois é preferido a partir das características que eles apresentam.

A quinta etapa é realizada *online*. Ela calcula a representação do usuário ativo a partir do seu histórico de avaliações e o associa a um dos perfis do modelo de recomendação calculado no processamento *offline*. O usuário é associado ao perfil cujo consenso  $C$  é o mais similar à sua representação. O modelo de preferências  $M$  do perfil escolhido é então aplicado sobre o conjunto de itens, para inferir quais serão os preferidos pelo usuário ativo a partir das características que eles apresentam. A figura 4.1 apresenta a sequência das etapas da metodologia PrefRec.

As ideias da abordagem de Filtragem Colaborativa são empregadas na metodologia PrefRec ao explorar as informações de outros usuários para construção de perfis de prefe-

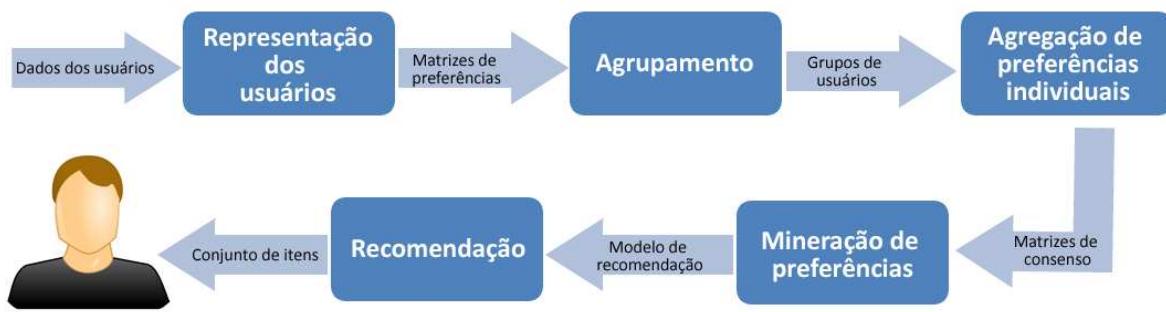


Figura 4.1: Arquitetura Geral da Metodologia PrefRec

rências. As ideias da técnica Baseada em Conteúdo são aplicadas ao propormos o uso de mineradores de preferências que obtenham um modelo capaz de comparar itens a partir das suas características.

As características aplicadas da abordagem Baseada em Conteúdo permitem que a metodologia PrefRec consiga recomendar itens que nunca foram avaliados, a partir das suas características, minimizando o problema de *cold-start* de item. Da mesma forma, a colaboração dos usuários já existentes no sistema permite a recomendação de itens diversos ao usuário ativo. A organização dos usuários em grupos colabora para obtenção de um sistema escalável diminuindo a quantidade de comparações no momento da recomendação, a fim de encontrar usuários semelhantes.

Nas seções seguintes detalhamos cada etapa da metodologia PrefRec e elicitamos algumas técnicas que podem ser empregadas em cada uma delas.

## 4.1 Etapa 1 - Representação dos Usuários

Para que seja possível recomendar algo para algum usuário é necessário que se tenha informações prévias à recomendação que possam ser exploradas. Desta forma, a metodologia PrefRec assume que está disponível:

- um conjunto  $\mathcal{U}$  de usuários representados somente pelo seu número de identificação;
- um conjunto  $\mathcal{I}$  de itens e seus atributos  $\{item_{id}, attr_1, attr_2, \dots, attr_k\}$ ;
- um conjunto  $\mathcal{R}$  de avaliações dos usuários aos itens onde cada elemento é da forma  $\{u_i, i_j, r_{ij}\}$ , onde  $r_{ij}$  é a avaliação dada ao item  $i_j$  pelo usuário  $u_i$ .

Nos Sistemas de Recomendação de Filtragem Colaborativa, os usuários são geralmente representados pelo seu histórico de avaliações (seção 2.1.1). A forma tradicional é associar a cada usuário um vetor de avaliações. Cada posição deste vetor representa um item e o seu valor representa a nota dada pelo usuário àquele item. Como exemplo temos a tabela 4.1 que apresenta a representação de um usuário  $u_i$  a partir do seu vetor de avaliações. O usuário  $u_i$  avaliou o item  $i_1$  com nota 5, o item  $i_2$  com nota 3 e o item  $i_4$  com nota 1. O

caractere \* que aparece relacionado ao item  $i_3$  indica que o usuário  $u_i$  não informou sua avaliação a este item. Os vetores definidos são usados então para identificar a similaridade entre os usuários: usuários que deram notas parecidas para os mesmos itens serão mais semelhantes (medida de similaridade PCC, seção 2.1.1).

Item	Avaliação
$i_1$	5
$i_2$	3
$i_3$	*
$i_4$	1

Tabela 4.1: Vetor de avaliações do usuário  $u_i$

A metodologia PrefRec propõe uma nova maneira de representação dos usuários em Sistemas de Recomendação: a matriz de preferência  $\mathcal{M}_{pref}$ . Uma  $\mathcal{M}_{pref}$  é uma matriz de dimensão  $n \times n$ , sendo  $n$  a quantidade de itens no sistema. Cada posição  $(a, b)$  da  $\mathcal{M}_{pref}$  de um determinado usuário  $u_i$  contém um valor entre 0 e 1 que representa o quanto o usuário  $u_i$  prefere o item  $i_a$  ao item  $i_b$ , isto é, o *grau de preferência* do usuário ao item  $i_a$  em relação ao item  $i_b$ . Como exemplo temos a  $\mathcal{M}_{pref}$  do usuário  $u_1$  apresentada na tabela 4.2 abaixo.

	$i_1$	$i_2$	$i_3$	$i_4$
$i_1$	0.50	0.63	*	0.83
$i_2$	0.37	0.50	*	0.75
$i_3$	*	*	0.50	*
$i_4$	0.17	0.25	*	0.50

Tabela 4.2:  $\mathcal{M}_{pref}$  de avaliações do usuário  $u_i$

O valor 0.83, localizado na linha do item  $i_1$  e na coluna do item  $i_4$ , indica que o usuário prefere o item  $i_1$  ao item  $i_4$  com um grau de preferência de 0.83. Este valor é calculado utilizando-se uma *relação de preferência fuzzy* ([Chiclana et al. 2001]) dada pela fórmula:  $r(r_a, r_b) = \frac{r_a}{r_b}$ , onde  $r_x$  é a nota dada para o item  $i_x$ . Esta razão é normalizada utilizando-se a função  $h(x) = \frac{x}{x+1}$  que produz um número no intervalo  $[0,1]$ . O valor final é dado por  $f(a, b) = h(r(r_a, r_b))$ . A função  $f$  assegura as seguintes propriedades importantes quando se trata de graus de preferência:

- $f(z, y) + f(y, z) = 1$ , condição de reciprocidade;
- $f(z, z) = 1/2$ , indica a indiferença do usuário entre dois itens que ele tenha preferências iguais;
- $f(z, 0) = 1$ , indica que entre um item que não satisfaz o usuário e outro que satisfaz o usuário positivamente, este deve ser preferido com valor máximo de preferência;
- $f(z, y) > 1/2$ , se  $r_z > r_y$ , indica que o usuário tem maior preferência para aquele item que ele deu uma melhor avaliação.

Os valores apresentados na tabela 4.2 são calculados a partir dos dados da tabela 4.1: calcula-se a *relação de preferência fuzzy* entre cada par de item. Como o item  $i_3$  não recebeu avaliação, ele não pode ser comparado a nenhum outro item, gerando valores \* (ausentes) na  $M_{pref}$  do usuário  $u_i$ .

É fácil observar que ao avaliarmos não avaliamos um item isoladamente, mas o comparamos com outro item para saber se ele é melhor ou não. Por exemplo, um determinado filme recebe uma nota melhor porque o consideramos melhor do que os outros filmes que já assistimos. Da mesma forma, um filme recebe uma nota baixa se o achamos inferior aos outros filmes que já conhecemos. A matriz de preferência tem como objetivo representar esse comportamento do usuário comparando os itens que ele avaliou e identificando, para cada par de itens, qual deles é o mais preferido. Assim, temos uma representação mais fiel do comportamento do usuário e que será a base para o desenvolvimento do sistema de recomendação.

As matrizes de preferências são usadas para determinar a semelhança entre os usuários dentro da metodologia PrefRec. Um usuário será semelhante ao outro se suas matrizes de preferências forem similares, isto é, se eles possuírem graus de preferências semelhantes para os mesmos pares de itens.

Diversos autores, como [Nurmi 1981] e [Sen 1970], tem enfatizado a importância da informação referente ao grau de preferência do usuário entre dois itens  $u$  e  $v$  ao se agregar preferências individuais em uma preferência coletiva (Etapa 3 da metodologia PrefRec). A consideração dos graus de preferências já é bem aceito na literatura em outras áreas de conhecimento, como os estudos sociais e econômicos, quando se estuda a agregação de preferências de um grupo social. Entretanto, esta representação ainda não é utilizada em Sistemas de Recomendação, sendo proposta neste trabalho como uma abordagem promissora a ser utilizada em conjunto com a mineração de preferências.

A tabela 4.3 abaixo apresenta quais são os dados de entrada e saída da Etapa 1 da metodologia PrefRec discutida nesta seção.

Etapa 1 - Representação dos usuários	
Entrada	um conjunto $\mathcal{U}$ de usuários;
	um conjunto $\mathcal{I}$ de itens;
	um conjunto $\mathcal{R}$ de avaliações dos usuários aos itens.
Saída	Uma $M_{pref}$ para cada usuário do conjunto $\mathcal{U}$

Tabela 4.3: Etapa 1 da metodologia PrefRec

## 4.2 Etapa 2 - Agrupamento dos Usuários

A Etapa 2 da metodologia PrefRec organiza os usuários em grupos a partir das suas respectivas  $M_{pref}$ 's calculadas na etapa 1. Usuários do mesmo grupo possuem preferências em comum, isto é,  $M_{pref}$ 's semelhantes. A definição de grupos entre os usuários

visa encontrar *perfis de preferências*. Um *perfil de preferência* é um par  $\theta = (C, M)$  onde  $C$  é a representação do consenso de um grupo (Etapa 3) e  $M$  é o seu respectivo modelo de preferências sobre os atributos dos itens do conjunto  $\mathcal{I}$  (Etapa 4).

Esta etapa da metodologia implica diretamente na escalabilidade do Sistema de Recomendação em relação ao momento de recomendação. O usuário ativo será comparado à quantidade de perfis existentes e não a todos os usuários do conjunto  $\mathcal{U}$ . A quantidade de perfis é consideravelmente menor do que a quantidade de usuários. Além disso, o aumento da quantidade de usuários e itens implicará em um aumento da construção do modelo de recomendação que é feito em um processamento *offline* na metodologia PrefRec.

O agrupamento de usuários aumenta a eficiência do Sistema de Recomendação em relação à escalabilidade. Porém, não se pode afirmar que irá melhorar a acurácia do algoritmo. Desta forma, a metodologia PrefRec busca equilibrar escalabilidade e acurácia a partir das técnicas de agregação e mineração de preferências (Etapa 3 e 4).

Para organizar os usuários em grupos é aplicada a técnica de mineração de dados denominada clusterização. Clusterização é o processo de agrupar um conjunto de objetos em vários grupos de tal forma que os objetos do mesmo grupo tenham alta similaridade e sejam muito dissimilares em relação aos objetos de outros grupos ([Han et al. 2011]).

Para efetuar a tarefa de clusterização é preciso definir uma medida de similaridade entre os usuários: um usuário é similar a outro se eles possuem gostos semelhantes, isto é,  $M_{pref}'s$  similares. A definição de similaridade entre  $M_{pref}'s$  é dada pela função de distância Cosseno que é muito utilizada na literatura. Dadas duas  $M_{pref}'s$   $A$  e  $B$ , e sendo  $n$  a quantidade de itens, a similaridade entre elas é dada por:

$$sim(A, B) = \frac{\sum_{i=1}^n \sum_{j=1}^n a_{ij} \cdot b_{ij}}{\sqrt{\sum_{i=1}^n \sum_{j=1}^n a_{ij}^2} \cdot \sqrt{\sum_{i=1}^n \sum_{j=1}^n b_{ij}^2}}$$

Existem diversas outras medidas de similaridades que podem ser utilizadas. O estudo delas e o impacto na metodologia PrefRec é sugerido como trabalho futuro.

Como alternativas de algoritmos de clusterização escolhemos três algoritmos de paradigmas distintos: o k-means, o Cure e o DBScan. O k-means se baseia no método de particionamento, o Cure se baseia no paradigma hierárquico e o DBScan utiliza a ideia de clusterização por densidade.

O *k*-means agrupa os objetos em  $k$  clusters. Na primeira iteração do algoritmo são definidos  $k$  objetos iniciais. Os demais objetos são associados aos objetos aos quais eles forem mais semelhantes. Para a próxima iteração são calculados os centróides (ponto médio) de cada grupo. Os objetos são comparados a todos os centróides e realocados para os grupos que forem mais semelhantes. O cálculo do centróide e a realocação dos objetos são feitos diversas vezes até que o critério de parada seja atendido. O *k*-means obtém clusters esféricos e é sensível a dados ruidosos [Han et al. 2011].

O algoritmo Cure ([Guha et al. 2001]) adota o método de clusterização hierárquico.

Em um primeiro momento, cada objeto é considerado como um cluster. A cada iteração do algoritmo pares de clusters são aglutinados até que um número  $k$  dado de clusters seja alcançado. Como parâmetros, além da quantidade  $k$  de clusters que se deseja, o Cure também solicita a quantidade  $c$  de representantes de cada cluster e um fator de retração  $\alpha$ . Os representantes de um cluster são pontos de regiões bem distintas que tentam capturar o formato do cluster. Eles que são usados para o cálculo da distância entre dois clusters. Já o fator de retração é utilizado para aproximar os representantes do centro do cluster, para diminuir a influência dos ruídos. A figura 4.2 apresenta clusters obtidos com o algoritmo Cure com diferentes valores de retração. O Cure consegue obter clusters de formatos esféricos e necessita que se defina a quantidade de clusters que se deseja alcançar.

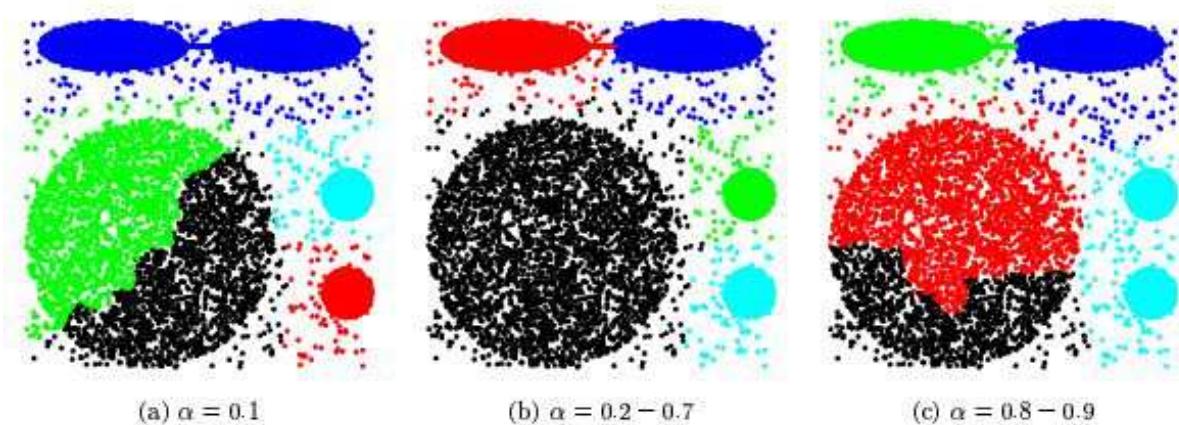


Figura 4.2: Algoritmo Cure com diferentes fatores de retração. Fonte: [Guha et al. 2001]

O algoritmo DBScan foi introduzido em [Ester et al. 1996] e utiliza o método baseado em densidade, isto é, dado um objeto calcula-se a quantidade de objetos próximos (similares) a ele. Diferente de outros algoritmos, o DBScan consegue encontrar clusters de formatos arbitrários, como os da figura 4.3. Outra característica deste algoritmo é que ele é robusto em relação a dados ruidosos, excluindo-os dos agrupamentos. Para definir uma região densa o DBScan recebe como parâmetro o valor do raio, para se definir uma esfera ao redor de um determinado ponto, e a quantidade mínima de pontos ( $minPt$ ) que deve existir dentro desse raio. Assim, o DBScan não necessita que seja informado a quantidade de grupos que se deseja criar. O DBScan define três tipos de pontos: pontos *core* são aqueles que tem mais de um número especificado ( $minPt$ ) de vizinhos dentro de um determinado raio; pontos *border* são os que tem menos vizinhos que o número especificado, mas pertence a vizinhança de um ponto *core*; e pontos *noise*, que são aqueles que não são nem *core* nem *border*. O algoritmo iterativamente remove os pontos *noise* e faz a clusterização com os pontos restantes.

A tabela 4.4 a seguir apresenta quais são os dados de entrada e saída da Etapa 2 da metodologia PrefRec discutida nesta seção.



Figura 4.3: Clusters baseados em densidade. Fonte: [Ester et al. 1996]

Etapa 2 - Agrupamento dos usuários	
Entrada	Uma $M_{pref}$ para cada usuário do conjunto $\mathcal{U}$
Saída	Grupos de usuários

Tabela 4.4: Etapa 2 da metodologia PrefRec

### 4.3 Etapa 3 - Agregação de Preferências Individuais

Para cada grupo de usuários calculado na etapa 2 é construído um perfil de preferência. Na etapa 3 busca-se definir o consenso  $C$  de cada perfil que agregue as preferências individuais dos usuários de cada grupo. O consenso  $C$  do perfil de preferência deve agrregar tanto as preferências comuns dos usuários do grupo quanto as preferências específicas de cada um, no sentido de equilibrar a escalabilidade e acurácia no Sistema de Recomendação.

A agregação de preferências é de fundamental importância dentro da metodologia PrefRec. O consenso  $C$  será utilizado para mineração das preferências do grupo (item  $M$  do perfil), além de ser comparado ao usuário ativo no momento da recomendação, para identificar qual perfil é mais semelhante ao usuário ativo. Assim, a agregação das preferências dos usuários de um grupo implica diretamente na qualidade das recomendações do Sistema de Recomendação.

A etapa 3, então, obtém uma  $M_{pref}$  consensual para cada grupo a partir da agregação das  $M_{pref}'s$  dos seus respectivos usuários. A  $M_{pref}$  consensual corresponde ao consenso  $C$  do perfil de preferência de cada grupo. A tarefa de agregação é estudada por outras áreas e vem sendo abordada em muitos trabalhos como em [Llamazares et al. 2013] e [García-Lapresta e Meneses 2005].

Cada posição  $(i, j)$  da  $M_{pref}$  consensual deve representar o valor preferido de acordo com a maioria dos usuários. O valor consensual só é calculado se mais da metade dos usuários do grupo informaram algum valor para esta posição, caso contrário, tal posição apresenta o símbolo \* (ausência de valor).

Para calcular a  $M_{pref}$  consensual de cada grupo propomos o estudo de cinco alternativas exemplificadas nas subseções a seguir.

### 4.3.1 Média Aritmética

A primeira alternativa de agregação de preferências, e a mais simples, é o cálculo da média aritmética para cada posição da  $M_{pref}$  consensual. Como exemplo, as tabelas 4.4(a), 4.4(b) e 4.4(c) abaixo apresentam as  $M_{pref}$ 's dos usuários  $u_1$ ,  $u_2$  e  $u_3$ , respectivamente, que pertencem a um mesmo grupo. A tabela 4.5 apresenta a  $M_{pref}$  consensual deste grupo, calculada pela média aritmética. A posição  $(i_1, i_2)$  da  $M_{pref}$  consensual, por exemplo, é calculada a partir dos valores  $\{0.63, 0.40, 0.52\}$  das  $M_{pref}$ 's dos usuários, obtendo o valor 0.52. As posições  $(i_3, i_4)$  e  $(i_4, i_3)$  da  $M_{pref}$  consensual aparecem com valor \* porque a maioria dos usuários, 2 usuários no exemplo, não possuem valor para essas posições.

	$i_1$	$i_2$	$i_3$	$i_4$
$i_1$	0.50	0.63	*	0.83
$i_2$	0.37	0.50	*	0.75
$i_3$	*	*	0.50	*
$i_4$	0.17	0.25	*	0.50

(a)  $M_{pref}$  do usuário  $u_1$

	$i_1$	$i_2$	$i_3$	$i_4$
$i_1$	0.50	0.40	0.20	0.45
$i_2$	0.60	0.50	0.38	0.90
$i_3$	0.80	0.62	0.50	0.60
$i_4$	0.55	0.10	0.40	0.50

(b)  $M_{pref}$  do usuário  $u_2$

	$i_1$	$i_2$	$i_3$	$i_4$
$i_1$	0.50	0.52	0.30	*
$i_2$	0.48	0.50	0.42	*
$i_3$	0.70	0.58	0.50	*
$i_4$	*	*	*	0.50

(c)  $M_{pref}$  do usuário  $u_3$

Figura 4.4: Exemplos de  $M_{pref}$ 's

	$i_1$	$i_2$	$i_3$	$i_4$
$i_1$	0.50	0.52	0.25	0.64
$i_2$	0.48	0.50	0.40	0.83
$i_3$	0.75	0.60	0.50	*
$i_4$	0.36	0.18	*	0.50

Tabela 4.5:  $M_{pref}$  consensual pela média aritmética

### 4.3.2 Média Ponderada

Uma outra alternativa para agregação de preferências é o cálculo de cada posição da  $M_{pref}$  consensual a partir de uma média ponderada, sendo o peso de cada usuário a medida do seu coeficiente de silhueta. O coeficiente de silhueta mede quanto um objeto (no caso, o usuário) está próximo dos elementos do seu grupo e distante dos elementos de outro grupo ([Han et al. 2011]). Assim, aqueles usuários que possuem altos coeficientes de silhueta, isto é, que representam bem o seu grupo, possuem maior peso no cálculo da  $M_{pref}$  consensual.

Considerando os seguintes valores de coeficiente de silhueta para os usuários do nosso exemplo:  $u_1 = 0.87$ ,  $u_2 = 0.5$ ,  $u_3 = -0.3$ ; a tabela 4.6 abaixo apresenta a  $M_{pref}$  consensual do grupo pela média ponderada pelo coeficiente de silhueta.

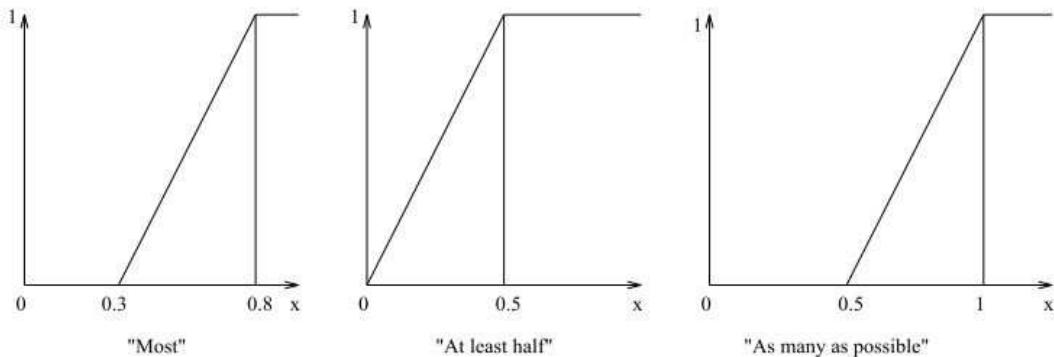
	$i_1$	$i_2$	$i_3$	$i_4$
$i_1$	0.50	0.55	0.05	0.69
$i_2$	0.45	0.50	0.32	0.80
$i_3$	0.95	0.68	0.50	*
$i_4$	0.31	0.20	*	0.50

Tabela 4.6:  $M_{pref}$  consensual pela média ponderada

### 4.3.3 Quantificadores Fuzzy

O problema da agregação de preferências é apresentar um valor que corresponda à opinião da maioria dos usuários que participam do grupo. Um conceito mais suave de maioria é a maioria *fuzzy*, que é expressa por um quantificador *fuzzy*. Os quantificadores *fuzzy* representam termos da linguagem natural que expressam quantidade, como "pelo menos a metade", "o maior número possível", entre outros ([Chiclana et al. 2001]).

A figura 4.5 apresenta três quantificadores *fuzzy*: *most*, *at least half* e *as many as possible*. Cada quantificador apresenta dois parâmetros,  $a$  e  $b$ , que são usados para definir pesos para o intervalo de valores a serem utilizados na agregação. O quantificador *most*, por exemplo, apresenta os parâmetros  $a = 0.3$  e  $b = 0.8$ .

Figura 4.5: Quantificadores *fuzzy* e seus parâmetros. Fonte: [Chiclana et al. 2001]

Para calcular um valor de agregação  $v$  dentro de um vetor  $V$  de valores utilizando quantificadores *fuzzy*, deve-se:

- Ordenar os valores de  $V$  de forma decrescente;
- Calcular o peso de cada posição do vetor a partir dos parâmetros  $a$  e  $b$  do quantificador escolhido;
- Calcular a média ponderada com os valores do vetor e seus respectivos pesos.

O peso de cada posição  $i$  ( $i = 1, \dots, l$ ) é calculado por

$$w_i = Q\left(\frac{i}{l}\right) - Q\left(\frac{i-1}{l}\right), Q(r) = \begin{cases} 0 & \text{se } r < a \\ \frac{r-a}{b-a} & \text{se } a \leq r \leq b \\ 1 & \text{if } r > b \end{cases}$$

Como exemplo, considerando  $V = \{0.6, 0.5, 0.4, 0.3\}$ , já ordenado de forma decrescente, os quantificadores da figura 4.5 obtém os seguintes pesos para cada posição de  $V$ :

- *Most* -  $w_{Most} = \{0.0, 0.4, 0.5, 0.1\}$
- *At least half* -  $w_{Least} = \{0.5, 0.5, 0.0, 0.0\}$
- *As many as possible* -  $w_{Many} = \{0.0, 0.0, 0.5, 0.5\}$

Pode-se observar que cada quantificador vai ponderar cada posição de forma diferente. Como o conjunto de valores,  $V$ , sempre é ordenado em forma decrescente, o quantificador *most* pondera mais os valores centrais, o *at least half* pondera com maior peso os maiores valores e o *as many as possible* os menores valores. Como resultado do exemplo dado, o valor  $v$  para cada quantificador citado acima, respectivamente, seria  $v_{least} = 0.55$ ,  $v_{most} = 0.43$  e  $v_{many} = 0.35$ . As tabelas 4.6(a), 4.6(b) e 4.6(c) abaixo mostram as  $Mpref$ 's consensuais dos usuários  $u_1, u_2$  e  $u_3$  obtidas para cada quantificador.

	$i_1$	$i_2$	$i_3$	$i_4$
$i_1$	0.50	0.50	0.21	0.48
$i_2$	0.46	0.50	0.38	0.76
$i_3$	0.71	0.58	0.50	*
$i_4$	0.20	0.11	*	0.50

(a)  $Mpref_{Most}$ 

	$i_1$	$i_2$	$i_3$	$i_4$
$i_1$	0.50	0.59	0.27	0.70
$i_2$	0.56	0.50	0.41	0.85
$i_3$	0.77	0.61	0.50	*
$i_4$	0.42	0.20	*	0.50

(b)  $Mpref_{Least}$ 

	$i_1$	$i_2$	$i_3$	$i_4$
$i_1$	0.50	0.44	0.20	0.45
$i_2$	0.41	0.50	0.38	0.75
$i_3$	0.70	0.58	0.50	*
$i_4$	0.17	0.10	*	0.50

(c)  $Mpref_{Many}$ Figura 4.6:  $Mpref$ 's calculadas com quantificadores fuzzy

A tabela 4.7 abaixo apresenta quais são os dados de entrada e saída da Etapa 3, da metodologia PrefRec discutida nesta seção.

Etapa 3 - Agregação de preferências	
Entrada	Grupos de usuários
Saída	Uma $Mpref$ consensual para cada grupo

Tabela 4.7: Etapa 3 da metodologia PrefRec

## 4.4 Etapa 4 - Mineração de Preferências

A última etapa *offline* da metodologia PrefRec aplica um minerador de preferências sobre cada  $Mpref$  consensual obtida na etapa 3. Nesta etapa, calcula-se o modelo de preferências  $M$  de cada perfil de preferência,  $\theta = (C, M)$ , a partir da  $Mpref$  consensual  $C$ .

O modelo de preferências  $M$  traduz a agregação das preferências individuais de um grupo de usuários que possuem preferências semelhantes. Esse modelo é utilizado para calcular as recomendações ao usuário ativo a partir do momento que se identifica sua similaridade com algum grupo. Desta forma, o usuário ativo receberá recomendações de itens que o grupo, ao qual ele é semelhante, informou preferências.

O modelo de preferências  $M$ , empregado na metodologia PrefRec, é um modelo que corresponde a qualquer estrutura compacta que permita prever, dados dois itens, qual deles é preferido pelo usuário a partir das suas características. Ao associar um usuário ativo a algum perfil na etapa de recomendação, é este modelo de preferências  $M$  que será aplicado sobre os itens, para assim, poder recomendá-los.

O minerador de preferências a ser utilizado na metodologia PrefRec deve ser capaz de receber como entrada uma  $M_{pref}$  consensual e calcular como saída um modelo de preferências  $M$ . A figura 4.7 ilustra esse processo.



Figura 4.7: Entrada e Saída do Minerador de Preferências na metodologia PrefRec

Caso o minerador a ser empregado não trabalhe com esses formatos de entrada e saída, é necessário que se implemente um conversor que transforme a  $M_{pref}$  consensual nos dados de entrada do minerador em questão. Da mesma forma, é necessário que exista um conversor dos dados de saída do minerador para o modelo de preferências  $M$ .

A aplicação de um modelo de preferências que permita comparar itens a partir das suas características colabora para a minimização do problema de *cold-start* de item discutido na seção 2.1.2. Para um item ser recomendado, basta que o modelo de preferências seja capaz de compará-lo a partir das suas características, independentemente da quantidade de avaliações que o item já tenha recebido.

Qualquer minerador de preferências pode ser aplicado à metodologia PrefRec desde que atenda ao critério apresentado e ilustrado na figura 4.7. Entre os mineradores apresentados na seção 2.2, as alternativas para estudo de caso da metodologia PrefRec são o CprefMiner e o CprefMiner\* (CprefMiner + técnica de otimização *Range Voting*). O modelo de preferência  $M$  gerado por essas duas alternativas é formado por um conjunto de regras de preferências contextuais (seção 2.2) que permitem, dados dois itens, que seja inferido qual dos dois será preferido pelo usuário a partir das suas características. Assim, não é necessário nenhum tratamento de conversão nos dados de saída desses mineradores.

Ao final dessa etapa temos construído o nosso *modelo de recomendação*, composto por um conjunto de perfis  $\theta$ , sendo cada um composto por sua  $M_{pref}$  consensual  $C$  e seu modelo de preferências  $M$ .

A tabela 4.8 a seguir apresenta quais são os dados de entrada e saída da Etapa 4 da metodologia PrefRec discutida nesta seção.

Etapa 4 - Mineração de preferências	
Entrada	Uma $M_{pref}$ consensual $C$ para cada grupo
Saída	Um modelo de recomendação

Tabela 4.8: Etapa 4 da metodologia PrefRec

## 4.5 Etapa 5 - Recomendação

A última etapa da metodologia PrefRec é o ato da recomendação realizada *online*, ao usuário ativo. Tem-se à disposição o modelo de recomendação construído nas etapas anteriores, e dados do histórico de avaliações do usuário ativo.

O primeiro passo da etapa de recomendação é calcular a  $M_{pref}$  do usuário ativo a partir dos dados disponíveis sobre seu histórico de avaliações. Feito isso, calcula-se a similaridade da  $M_{pref}$  do usuário ativo para as  $M_{pref}'s$  ( $C$ ) de cada perfil do modelo de recomendação. O usuário será associado ao perfil que possuir o  $C$  mais similar à sua  $M_{pref}$ . Denominamos o perfil escolhido como perfil ativo.

A comparação da  $M_{pref}$  do usuário ativo com os  $C$  dos perfis de preferência é feita em uma quantidade bem menor em relação à quantidade de usuários existentes no sistema. O modelo de preferências  $M$  do perfil ativo é então aplicado sobre o conjunto de itens do conjunto  $\mathcal{I}$ , excetuando aqueles que o usuário ativo já avaliou. Os itens que o modelo de preferências indica como preferidos são recomendados ao usuário ativo. Como o perfil representa as preferências de um conjunto de usuários, o usuário ativo receberá recomendações de itens diversos a partir das preferências desses usuários. Busca-se, assim, que o usuário ativo tenha uma interação satisfatória com o Sistema de Recomendação.

---

### Algoritmo Order By Preferences

---

```

1:  $V = X$ 
2: for all  $v \in V$  do
3:    $\pi(v) = \sum_{u \in V} PREF(v, u) - \sum_{u \in V} PREF(u, v)$ 
4: end for
5: while  $V \neq \{\}$  do
6:    $t = argmax_{u \in V} \pi(u)$ 
7:    $\rho(t) = |V|$ 
8:    $V = V - \{t\}$ 
9:   for all  $v \in V$  do
10:     $\pi(v) = \pi(v) + PREF(t, v) - PREF(v, t)$ 
11:   end for
12: end while

```

Figura 4.8: Algoritmo **Order-By-Preferences** adaptado de [Cohen et al. 1999]

O modelo de preferências  $M$  é capaz de comparar pares de itens a partir das suas características. Podemos associar uma função  $PREF(x, y)$  ao modelo de preferências que retorna o valor 1 caso o modelo infira que o item  $x$  é preferido ao item  $y$ . O valor -1

é retornado caso seja inferido que o item  $y$  é preferido ao item  $x$ . Caso o modelo não consiga comparar os dois itens, a função  $PREF(x, y)$  retorna 0.

Diante disso, é possível construir um *ranking* de itens a partir da inferência do modelo de preferências sobre pares de itens, aplicando o algoritmo *Order By Preferences*, adaptado de [Cohen et al. 1999] (figura 4.8). A entrada deste algoritmo é o conjunto  $X$  de itens a serem recomendados. A saída é a ordenação desses itens aplicando a função  $PREF$ . Exemplificamos a construção de um *ranking* de itens na seção 5.2.3.

A tabela 4.9 abaixo apresenta quais são os dados de entrada e saída da Etapa 5 da metodologia PrefRec discutida nesta seção.

Etapa 5 - Recomendação	
Entrada	Um modelo de recomendação Um usuário ativo e sua respectiva $M_{pref}$
Saída	Um <i>ranking</i> de itens

Tabela 4.9: Etapa 5 da metodologia PrefRec



# Capítulo 5

## Validação da Metodologia PrefRec

Neste capítulo apresentamos como validamos a metodologia PrefRec. Primeiro apresentamos quais dados utilizamos e quais foram as medidas de validação escolhidas. A seguir, definimos grupos de testes de forma empírica, visando aplicar as alternativas propostas para cada etapa no capítulo 4 e identificar os fatores dentro da metodologia que influenciam seu desempenho. Por fim, definimos com quais trabalhos da literatura a metodologia PrefRec será comparada.

### 5.1 Dados de Teste

Os dados utilizados para a validação da metodologia PrefRec são os dados reais disponibilizados pelo projeto GroupLens ([www.grouplens.org](http://www.grouplens.org)). Os dados pertencem a um cenário de avaliação de filmes, tendo disponível a identificação do usuário (IdUsuario), a identificação do filme (IdFilme) e a nota dada pelo usuário ao filme (Nota). Os filmes contidos nessa base de dados são os apresentados no site MovieLens ([www.movieLens.umn.edu](http://www.movieLens.umn.edu)) que é um Sistema de Recomendação de filmes. Para a coleta de dados sobre os filmes foi feito um *crawler* para obter as seguintes características sobre eles: Gênero (até 5 valores), Atores (até 5 atores), Idioma, Ano, Diretor.

A base utilizada disponibiliza 10.000.054 avaliações de 69.878 usuários, sobre 10.677 filmes. Entre essa quantidade de dados foram escolhidos usuários que avaliaram uma quantidade expressiva de filmes. Assim, foram selecionados 296 usuários que avaliaram no mínimo 13% dos itens da base, isto é, avaliaram no mínimo 1388 filmes. O objetivo dessa seleção é ter usuários com experiência no sistema, isto é, *experts* sobre os itens do cenário. A esse conjunto damos o nome de  $\bar{U}$ .

Para a seleção dos itens escolheu-se aqueles que foram avaliados por uma quantidade expressiva de usuários de  $\bar{U}$ . Desse modo, definimos um parâmetro de 80%, onde obtivemos 262 itens, o conjunto  $\bar{I}$ . Cada item do conjunto  $\bar{I}$  foi avaliado por no mínimo 237 usuários de  $\bar{U}$ . As avaliações disponíveis dos usuários para os itens selecionados somam a quantidade de 67.971 avaliações. Este valor indica que não se tem disponível todas as

avaliações dos 296 usuários para os 262 filmes, isto é, este conjunto de dados apresenta 12% de esparsidade. A este banco damos o nome de  $BD_{100}$ .

Sendo a esparsidade dos dados um problema em Sistemas de Recomendação, estruturamos outros 5 bancos aumentando a esparsidade dos dados de avaliações, isto é, retirando avaliações do conjunto original. Assim, para o  $BD_{90}$  retiramos 10% das avaliações do  $BD_{100}$  de forma estratificada: para cada usuário, tiramos, aleatoriamente, 10% das avaliações que ele deu para cada nota. A tabela abaixo apresenta os bancos gerados, a quantidade de avaliações e o valor da esparsidade de cada um. O objetivo da construção desses outros bancos de dados é avaliar o comportamento da metodologia PrefRec diante de dados esparsos.

Tabela 5.1: Bancos de validação

BD	Avaliações	Esparsidade
$BD_{100}$	67.971	12.35%
$BD_{90}$	61.143	21.16%
$BD_{80}$	54.423	29.82%
$BD_{70}$	47.464	38.80%
$BD_{60}$	40.831	47.35%
$BD_{50}$	33.344	57.00%

## 5.2 Medidas de Validação

Para validar a metodologia PrefRec, aplicamos a técnica de *k-cross-validation*, para  $k = 5$ . Primeiramente, dividimos o conjunto de usuários em 5 partes, sendo que a cada iteração quatro partes são usadas para treinamento e a outra para teste. Para cada usuário de teste também foi realizado o *5-cross-validation*, dividindo o conjunto de itens em 5 partes, de tal forma que os itens avaliados por esse usuário estavam divididos de forma estratificada em relação às notas recebidas.

A seguir, apresentamos as medidas de validação empregadas no nosso estudo de caso.

### 5.2.1 Acurácia

Como discutido na seção 2.1.3, a medida de acurácia tem como objetivo identificar quanto o Sistema de Recomendação é capaz de predizer corretamente as preferências dos usuários. Essa capacidade é o mínimo que se espera de um bom Sistema de Recomendação.

Propomos medir a acurácia da metodologia PrefRec a partir das medidas de precisão e revocação, que são baseadas na quantidade de acertos e erros que o sistema comete. O modelo de recomendação da metodologia PrefRec aplica um modelo de preferências que é capaz de comparar dois itens a partir das suas características. Nesse sentido, para cada usuário teste são definidos pares de itens a partir das avaliações deste usuário que se tem

disponíveis. O modelo de preferências é aplicado sobre estes pares de itens, para verificar se ele infere de forma correta a preferência do usuário teste. Como exemplo, temos as avaliações do usuário  $u$  para os itens apresentados na tabela 5.1(a). Na tabela 5.1(b) temos os pares gerados a partir das avaliações disponíveis. Observa-se que os itens que receberam notas iguais não geram pares.

Item	Avaliação
$i_1$	4
$i_2$	5
$i_3$	3
$i_4$	1
$i_5$	1
$i_6$	4
$i_7$	2

(a) Avaliações do usuário  $u$

$i_1 > i_3$	$i_3 > i_4$
$i_1 > i_4$	$i_3 > i_5$
$i_1 > i_5$	$i_3 > i_7$
$i_1 > i_7$	$i_6 > i_3$
$i_2 > i_1$	$i_6 > i_4$
$i_2 > i_3$	$i_6 > i_5$
$i_2 > i_4$	$i_6 > i_7$
$i_2 > i_5$	$i_7 > i_4$
$i_2 > i_6$	$i_7 > i_5$
$i_2 > i_7$	

(b) Pares

Figura 5.1: Transformação de avaliações para pares

O modelo de recomendação recebe os pares formados e infere a preferência do usuário para cada um deles, ou informa que não é possível comparar os dois itens. Ao modelo o par  $(i_1 > i_3)$  é repassado no formato  $(i_1, i_3)$  com ausência da informação de qual é a preferência correta. Essa informação é usada posteriormente para verificar se o modelo acertou ou não. As medidas de precisão e revocação são então calculadas a partir dos erros e acertos do modelo de recomendação sobre os pares dados.

A medida de acurácia é dividida na nossa validação em: acurácia fraca, acurácia forte e acurácia global. Cada tipo de acurácia possui sua respectiva medida de precisão e revocação. A cada iteração do cross-validation é definido um conjunto de itens de teste e um conjunto de itens de treinamento. Para cada um desses conjuntos, conhecendo as notas dadas pelo usuário, são calculados seus respectivos pares, que são denominados pares-treinamento e pares-teste.

A acurácia forte mede a acurácia do modelo de recomendação em relação aos pares-teste. Os pares de teste são formados por itens cujas avaliações não participam da fase de treinamento. A acurácia fraca mede a acurácia do modelo de recomendação em relação aos pares-treinamento. Os itens que compõem os pares de treinamento, assim como suas avaliações, foram utilizados para gerar o modelo de recomendação, portanto, são informações já conhecidas pelo sistema. A acurácia global, por sua vez, une os dados obtidos na acurácia fraca e na acurácia forte para verificar, globalmente, como o modelo se comporta.

Abaixo, apresentamos as fórmulas para as medidas de acurácia:

$$precisão_{fraca} = p_f = \frac{acertos_{fraco}}{acertos_{fraco} + erros_{fraco}}$$

$$revocacão_{fraca} = r_f = \frac{acertos_{fraco}}{|pares_{treinamento}|}$$

$$precisão_{forte} = p_F = \frac{acertos_{forte}}{acertos_{forte} + erros_{forte}}$$

$$revocacão_{forte} = r_F = \frac{acertos_{forte}}{|pares_{teste}|}$$

$$precisão_{global} = p_G = \frac{acertos_{forte} + acertos_{fraco}}{acertos_{forte} + acertos_{fraco} + erros_{forte} + erros_{fraco}}$$

$$revocacão_{global} = r_G = \frac{acertos_{forte} + acertos_{fraco}}{|pares_{treinamento}| + |pares_{teste}|}$$

Como exemplo do cálculo da acurácia vamos considerar que, para treinamento, foram definidos os itens  $i_1, i_2, i_3, i_4$  da tabela 5.1(a) e o restante para teste. Assim, os pares de treinamento obtidos são os listados na primeira coluna da tabela 5.2. Para o modelo de recomendação são repassados os pares formatados, como na segunda coluna da mesma tabela. Na terceira coluna são apresentadas as inferências do modelo de recomendação para cada par de treinamento. Para o primeiro par, o modelo infere corretamente qual item do par é o preferido do usuário; para o segundo par, ele já foi indiferente, não conseguindo predizer qual seria preferido; para o quarto par, ele infere erroneamente a preferência do usuário, e assim sucessivamente. Da mesma forma, apresentamos os pares de teste na tabela 5.3 e as inferências feitas pelo modelo de recomendação.

Tabela 5.2: Conjunto de treinamento

Pares treinamento	Pares formatados	Predição
$i_1 > i_3$	$i_1, i_3$	$i_1 > i_3$
$i_1 > i_4$	$i_1, i_4$	Indiferente
$i_2 > i_1$	$i_2, i_1$	$i_2 > i_1$
$i_2 > i_3$	$i_2, i_3$	$i_2 < i_3$
$i_2 > i_4$	$i_2, i_4$	$i_2 > i_4$
$i_3 > i_4$	$i_3, i_4$	$i_3 > i_4$

Tabela 5.3: Conjunto de teste

Pares teste	Pares formatados	Predição
$i_6 > i_5$	$i_6, i_5$	$i_6 > i_5$
$i_6 > i_7$	$i_6, i_7$	$i_6 < i_7$
$i_7 > i_5$	$i_7, i_5$	Indiferente

Nesse exemplo os valores da acurácia fraca, forte e global seriam:

$$precisão_{fraca} = \frac{4}{4+1} = 80\%$$

$$revocacão_{fraca} = \frac{4}{6} = 67\%$$

$$precisão_{forte} = \frac{1}{1+1} = 50\%$$

$$revocacão_{forte} = \frac{1}{3} = 33\%$$

$$\text{precisão}_{\text{global}} = \frac{4+1}{4+1+1+1} = 71\%$$

$$\text{revocacão}_{\text{global}} = \frac{4+1}{6+3} = 55\%$$

### 5.2.2 Cobertura

Assumimos como cobertura a definição proposta por [Shani e Gunawardana 2011] de ser a medida do comportamento do Sistema de Recomendação em relação ao problema de *cold-start* de item. Propomos medir a cobertura do sistema sobre um conjunto específico de itens e a acurácia do sistema em relação a eles. Esse conjunto específico de itens, itens-cobertura, é composto por itens que não fazem parte do conjunto  $\bar{I}$ , e que o modelo de recomendação não teve acesso às suas avaliações, isto é, são itens que nunca foram avaliados do ponto de vista do modelo de recomendação. A diferença do conjunto de itens-cobertura para os itens de teste é que, esses últimos são itens avaliados por pelo menos 80% dos usuários de  $\bar{U}$ , enquanto que os itens cobertura podem ter sido avaliados por uma quantidade menor de usuários, ou por usuários que não fazem parte do conjunto  $\bar{U}$ .

Para cada usuário teste obtém-se os itens que ele avaliou da base de dados original e que não fazem parte do conjunto  $\bar{I}$ . São escolhidos 10% dos itens que receberam nota 1, mais 10% dos itens que receberam nota 2, e assim sucessivamente para todas as notas dadas pelo usuário teste.

Análogo ao cálculo da acurácia, são calculados os pares-cobertura a partir dos itens-cobertura. O objetivo de medir a cobertura é saber se o Sistema de Recomendação consegue prever a preferência do usuário sobre itens ainda não avaliados. Dessa forma, calculamos os acertos e os erros do modelo de recomendação em relação aos pares-cobertura e obtemos as seguintes medidas:

$$\text{precisão}_{\text{cobertura}} = p_C = \frac{\text{acertos}_{\text{cobertura}}}{\text{acertos}_{\text{cobertura}} + \text{erros}_{\text{cobertura}}}$$

$$\text{revocacão}_{\text{cobertura}} = r_C = \frac{\text{acertos}_{\text{cobertura}}}{|\text{pares}_{\text{cobertura}}|}$$

Exemplificando, temos os itens da tabela 5.4 compondo o conjunto itens-cobertura. A partir deles são calculados os pares-cobertura, e sobre eles o modelo infere quais itens são preferido como apresentado na tabela 5.5.

Nesse exemplo, os valores de precisão e revocação da cobertura são:

$$\text{precisão}_{\text{cobertura}} = \frac{5}{5+2} = 71\%$$

$$\text{revocacão}_{\text{cobertura}} = \frac{5}{9} = 55\%$$

Tabela 5.4: Avaliações do usuário  $u$  para itens de cobertura

Item	Avaliação
$i_8$	3
$i_9$	2
$i_{10}$	5
$i_{11}$	5
$i_{12}$	4

Tabela 5.5: Conjunto de cobertura

Pares cobertura	Pares formatados	Predição
$i_8 > i_9$	$i_8, i_9$	$i_8 > i_9$
$i_{10} > i_8$	$i_{10}, i_8$	$i_{10} > i_8$
$i_{10} > i_9$	$i_{10}, i_9$	Indiferente
$i_{10} > i_{12}$	$i_{10}, i_{12}$	$i_{10} < i_{12}$
$i_{11} > i_8$	$i_{11}, i_8$	$i_{11} > i_8$
$i_{11} > i_9$	$i_{11}, i_9$	$i_{11} > i_9$
$i_{11} > i_{12}$	$i_{11}, i_{12}$	$i_{11} < i_{12}$
$i_{12} > i_8$	$i_{12}, i_8$	Indiferente
$i_{12} > i_9$	$i_{12}, i_9$	$i_{12} > i_9$

### 5.2.3 Novidade

Como discutido na seção 2.1.3, um Sistema de Recomendação deve trazer outros benefícios para o usuário, além da acurácia. São vários os fatores que permitem uma maior satisfação do usuário no momento da recomendação. Entre as medidas apresentadas definimos verificar como a metodologia PrefRec se comporta em relação a recomendação de itens novos e surpreendentes. A verificação de outras medidas de validação é sugerido como trabalho futuro.

Em relação à medida de novidade, assumimos a proposta de [Ziegler et al. 2005], ao definir novidade como a métrica que pontua melhor as recomendações de itens pouco populares. Os itens populares facilmente chegariam ao conhecimento do usuário sem a ajuda de um Sistema de Recomendação. Recomendar itens da preferência do usuário e não tão populares seria de melhor auxílio ao usuário.

Na validação da metodologia PrefRec, a popularidade de um item é definida pela quantidade de avaliações que ele possui entre os usuários do conjunto  $\bar{U}$ . Usamos o mesmo conjunto de itens usados para a medida de cobertura. Como a quantidade de itens-cobertura se torna grande, definimos um *ranking* de itens e escolhemos os 50 primeiros para calcular a novidade. A construção do *ranking* é feita a partir da aplicação do algoritmo *Order by preferences*, apresentado na seção 4.5.

A partir dos itens-cobertura da tabela 5.4 e das inferências feitas pelo modelo de recomendação, apresentadas na tabela 5.5, o *ranking* destes itens é o apresentado na tabela 5.6 na coluna Ranking, sendo o item  $i_{12}$  o mais preferido e o  $i_{10}$  o menos preferido.

As inferências feitas pelo modelo de recomendação correspondem a função PREF do algoritmo *Order by preferences*. Por exemplo, o modelo inferiu que  $i_8 > i_9$ , assim a função  $PREF(i_8, i_9) = 1$ . Na coluna Avaliações, da mesma tabela, aparecem as quantidades de avaliações que cada item recebeu, como exemplo.

Tabela 5.6: Ranking de itens

Ranking	Avaliações
$i_{12}$	200
$i_{11}$	50
$i_{10}$	102
$i_8$	78
$i_9$	140

Para cada item, é calculada a quantidade de avaliações que ele recebeu dos usuários do conjunto  $\bar{U}$ . A medida de novidade da lista é a média da quantidade de avaliações de cada item. No exemplo da tabela 5.6 a novidade é de valor 114. A medida de novidade do usuário é a média dos valores de novidade obtido em cada iteração. A medida de novidade do Sistema de Recomendação será a média dos valores de novidade de cada usuário.

Quanto menor o valor obtido na métrica novidade significa que o Sistema de Recomendação consegue recomendar itens poucos populares ao usuário, isto é, recomenda itens que são novidade ao usuário ativo. Para verificar o comportamento do sistema, comparamos a metodologia PrefRec com outro trabalho da literatura.

#### 5.2.4 Serendipity

Essa métrica calcula quão surpreendentes são os itens recomendados ao usuário. Para mensurar esta medida, calcula-se a semelhança do item recomendado ao conjunto de itens que o usuário já conhece ([Shani e Gunawardana 2011]). A métrica utilizada é a descrita na seção 2.1.3, que calcula a distância de cada atributo do item recomendado ao conjunto de itens que o usuário já conhece.

Na validação da metodologia PrefRec, em relação à medida *serendipity*, também utilizamos o *ranking* de 50 itens recomendados a partir do conjunto de itens-cobertura construído para cada usuário teste. O conjunto de itens que esse usuário já conhece é o conjunto de itens-treinamento utilizado para construir o modelo de recomendação. Assim, calcula-se a distância de cada atributo de cada item recomendado no *ranking* para o conjunto de itens-treinamento. Cada item, na base de dados  $BD_{100}$ , possui até 12 atributos, sendo no máximo 5 caracterizando o seu gênero; 4 caracterizando os atores, 1 para idioma, 1 para diretor e 1 para ano.

O valor de *serendipity* de um item será a média dos valores obtidos para cada um dos seus atributos. O valor de *serendipity* de um *ranking* é a média dos valores de *serendipity* dos seus itens. Da mesma forma, como na novidade, o valor de *serendipity* alcançado

para um usuário é a média obtida em todas as iterações de teste. Por fim, o valor de *serendipity* do Sistema de Recomendação é a média obtida de todos os usuários.

Observamos que, no caso dos atributos multivalorados gênero e atores principais, a distância de cada valor apresentado pelo item é calculada em relação a todo o conjunto de valores de gênero e o conjunto de valores de atores principais. A ordem com que os valores aparecem no item ou no conjunto de itens já conhecidos pelo usuário não influencia no cálculo da distância.

Análogo a medida de novidade, é necessário comparar a metodologia PrefRec com outra abordagem para verificar seu comportamento. Quanto maior for o valor alcançado de *serendipity*, mais surpreendentes são as recomendações realizadas.

### 5.3 Fatores que Influenciam no Desempenho da Metodologia PrefRec

Buscamos identificar, nesta etapa da validação, quais são os fatores que influenciam no desempenho da metodologia PrefRec. No capítulo 4 elicitamos alternativas de técnicas que poderiam ser aplicadas à metodologias e que são enumeradas na tabela 5.7. As etapas 1 e 5 não possuem alternativas e são realizadas como descritas no capítulo 4.

Tabela 5.7: Alternativas para a metodologia PrefRec

Etapa 2 - Agrupamento dos usuários	K-Means
	DBScan
	Cure
Etapa 3 - Agregação de preferências	Média aritmética
	Média ponderada pelo coeficiente de silhueta
	Quantificador fuzzy <i>at least half</i>
	Quantificador fuzzy <i>as many as possible</i>
	Quantificador fuzzy <i>most</i>
Etapa 4 - Mineração de preferências	Cprefminer
	Cprefminer*

A fim de identificar quais das etapas e quais das alternativas (fatores) influenciam no desempenho da metodologia PrefRec, definimos grupos de teste onde, em cada grupo, instanciamos alguma das alternativas de determinada etapa. A definição da sequência dos grupos e quais alternativas instanciar é feito de forma empírica. A melhor alternativa de um grupo de teste é aplicada como padrão para o grupo de teste subsequente. Uma alternativa é dita melhor que outra a partir dos valores de acurácia e cobertura obtidos.

Para realizar os testes, implementamos a metodologia obtendo o Sistema de Recomendação ao qual denominamos XPrefRec. A partir do XPrefRec é possível mensurar as medidas de validação. Ao final dos grupos de testes temos o sistema implementado com

as melhores alternativas para cada etapa da metodologia PrefRec.

O primeiro grupo de teste instancia as alternativas da etapa 2: qual algoritmo de clusterização obtém melhor resultado de acurácia e cobertura. Para as outras etapas, 3 e 4, assume-se as alternativas de média aritmética e o minerador CprefMiner como alternativas padrões. A configuração do primeiro grupo de teste é a descrita na tabela 5.8.

Tabela 5.8: Grupo de Teste 1

Etapa 2 - Agrupamento dos usuários	K-Means
	DBSCAN
	CURE
Etapa 3 - Agregação de preferências	Média aritmética
Etapa 4 - Mineração de preferências	Cprefminer

A melhor alternativa do grupo de teste 1 é utilizada para o grupo de teste 2. Neste outro grupo são testadas quais das alternativas de agregação de preferências obtém os melhores resultados (tabela 5.9). A melhor alternativa do grupo de teste 2 é aplicada no grupo de teste 3.

Tabela 5.9: Grupo de Teste 2

Etapa 2 - Agrupamento dos usuários	Melhor alternativa do grupo 1
Etapa 3 - Agregação de preferências	Média aritmética
	Média ponderada pelo coeficiente de silhueta
	Quantificador fuzzy at least half
	Quantificador fuzzy as many as possible
Etapa 4 - Mineração de preferências	Quantificador fuzzy most
	Cprefminer

O grupo de teste 3, por sua vez, verificará qual das alternativas de minerador de preferências levantados tem melhor resultado. As alternativas são o CprefMiner e o Cprefminer\*, que é a versão original do algoritmo acrescido da técnica de otimização *Range Voting*. A configuração do grupo de teste 3 é a apresentada na tabela 5.10.

Tabela 5.10: Grupo de Teste 3

Etapa 2 - Agrupamento dos usuários	Melhor alternativa do grupo 1
Etapa 3 - Agregação de preferências	Melhor alternativa do grupo 2
Etapa 4 - Mineração de preferências	Cprefminer
	Cprefminer*

Ao final dos três grupos de testes temos a configuração do XPrefRec com as melhores alternativas elicitadas. A cada resultado de teste será possível observar qual alternativa obtém melhores resultados que outras, e qual etapa da metodologia PrefRec mais influencia no desempenho do sistema XPrefRec.

## 5.4 Comparação com Outros Trabalhos

Para verificar a qualidade dos resultados obtidos pela metodologia PrefRec é necessário compará-la com outra(s) abordagens da literatura. Entre os trabalhos apresentados no capítulo 3 definimos o de [Resnick et al. 1994] e [Melville et al. 2002] como *baseline* de comparação.

O trabalho de [Resnick et al. 1994], que trata da abordagem tradicional de Filtragem Colaborativa em Sistemas de Recomendação (explanado nas seções 2.1.1 e 3.1), é empregado para servir de valores mínimos que o sistema XPrefRec deve alcançar em relação às medidas de acurácia.

Já o trabalho de [Melville et al. 2002] é usado como estado da arte para todas as medidas de validação definidas neste capítulo. Este trabalho propõe o Sistema de Recomendação CBCF e é detalhado na seção 3.3. A definição deste trabalho como estado da arte se dá pelo conjunto de características em comum que ele apresenta em relação à metodologia PrefRec:

- Abordagem híbrida entre a abordagem Baseada em Conteúdo e Filtragem Colaborativa;
- Tratamento do problema de *cold-start* de item;
- Mineração de preferências sobre os atributos a partir de um classificador bayesiano.

Tais características ressaltam pontos importantes que definem os bons resultados de ambos trabalhos. Porém, a metodologia PrefRec difere do sistema CBCF em vários pontos:

- O CBCF utiliza a abordagem Baseada em Conteúdo para predizer avaliações faltantes, enquanto o PrefRec a utiliza para construir o modelo de preferências e utilizá-lo no momento da recomendação;
- O CBCF compara o usuário ativo com todos os usuários do sistema, enquanto que o PrefRec clusteriza os usuários diminuindo significativamente a quantidade de comparações no momento da recomendação;
- O CBCF representa os usuários a partir do seu vetor de avaliações, enquanto que o PrefRec representa os usuários por sua matriz de preferência, que apresenta o quanto o usuário prefere um item ao outro;
- O CBCF tenta predizer notas, o PrefRec prediz uma relação de preferência;
- O CBCF é capaz de calcular a recomendação para qualquer item a partir da definição de probabilidades mínimas de utilidade. O PrefRec pode ter situações em que o modelo de recomendação não consegue inferir a preferência sobre dois itens;

- O CBCF aplica um classificador bayesiano *naive bayes*, que considera os atributos como eventos independentes. No estudo de caso da PrefRec, o algoritmo de mineração de preferências aplicado utiliza uma rede bayesiana de preferência, que considera a dependência entre os atributos.

Como o sistema CBCF, assim como a abordagem de Filtragem Colaborativa tradicional, prediz notas para os itens, transformamos os resultados gerados em pares para calcular as medidas de acurácia, cobertura, novidade e *serendipity*, como explanadas na seção 5.2. Dado um par  $(i_1, i_2)$ , por exemplo, verificamos se o CBCF prediz uma nota maior para  $i_1$  do que para  $i_2$  e verificamos a quantidade de acertos e erros que ele alcança.



# Capítulo 6

## Resultados Experimentais

Neste capítulo apresentamos os resultados obtidos a partir da validação da metodologia PrefRec, apresentada no capítulo 5. Referenciamos a metodologia a partir do sistema implementado XPrefRec.

### 6.1 Fatores que Influenciam no Desempenho da Metodologia PrefRec

Os três grupos de teste definidos no capítulo 5 buscam identificar as melhores alternativas entre as propostas para a metodologia PrefRec. Para cada grupo são calculadas as medidas de acurácia (forte, fraca e global) e cobertura. Os testes são realizados sobre o banco de dados  $BD_{100}$ .

O valor mínimo de acurácia a ser alcançado pelo XPrefRec é o obtido pela abordagem de Filtragem Colaborativa tradicional, apresentada em [Resnick et al. 1994]. Realizou-se o teste dessa abordagem sobre o banco  $BD_{100}$ , obtendo os resultados apresentados na tabela 6.1. Observa-se que a abordagem consegue um mínimo de 70% de acerto nas suas recomendações a partir das medidas de precisão e revocação apresentadas na seção 5.2.1. A medida de cobertura não é apresentada uma vez que essa abordagem não é capaz de recomendar itens que nunca foram avaliados por outro usuário.

Tabela 6.1: Resultados de acurácia para o Filtragem Colaborativa tradicional

Acurácia					
$p_f$	$r_f$	$p_F$	$r_F$	$p_G$	$r_G$
70.30%	70.30%	71.74%	71.74%	70.35%	70.35%

#### 6.1.1 Grupo de Teste 1

No grupo de teste 1 buscou-se verificar qual o melhor algoritmo de clusterização. Foram definidas como alternativas o algoritmo DBSCAN, categorizado como algoritmo de

clusterização por densidade; o Cure, sendo um algoritmo de clusterização hierárquico e o K-Means, que é um algoritmo de particionamento.

O algoritmo K-Means necessita que seja passada como parâmetro a quantidade  $k$  de clusters que se deseja encontrar. Para tal, a definição de  $k$  foi feita de forma iterativa: o valor de  $k$  é assumido como a quantidade de usuários de treinamento e diminuído a cada iteração, até encontrar uma clusterização com coeficiente de silhueta de no mínimo 80%. Os elementos iniciais para formação dos clusters foram definidos como os usuários que possuíam a maior quantidade de usuários mais próximos a ele. Nesse sentido, foi necessário calcular uma matriz de distâncias entre todos os usuários. Cada execução do algoritmo  $k$ -Means compreende várias comparações entre os objetos e os centróides dos clusters, para alocá-los no cluster mais próximo. Este processo é feito iterativamente, até que não se alterem os usuários do cluster, e a uma quantidade máxima de 10 iterações.

Os testes realizados com o algoritmo  $k$ -Means apresentaram um alto custo computacional e foram demasiadamente demorados para a realização dos testes de validação (mais de 15 dias de teste). Um teste de validação completo necessita realizar 5 iterações para cada um dos 296 usuários, totalizando 1480 iterações. Em cada iteração, o algoritmo  $k$ -Means obteve mais de 150 clusters, sendo que para cada um foi necessário calcular um perfil. Este valor implicou, no momento da recomendação, na comparação do usuário ativo com mais de 150 perfis, o que não tornou o XPrefRec escalável. Assim, excluímos a alternativa K-Means do grupo de teste 1 e consideramos apenas os algoritmos DBScan e Cure.

O algoritmo Cure também necessita do parâmetro  $k$  de clusters que se deseja encontrar. A definição de  $k$  também foi feita iterativamente como proposto para o algoritmo k-means. O Cure também encontra uma quantidade de mais de 150 clusters, porém, possui um custo computacional muito inferior ao k-means porque realiza uma quantidade bem inferior de comparações. Para o parâmetro de quantidade de representantes foi definido o valor 3, e para o valor de retração foi configurado 0.5.

O algoritmo DBScan não necessita da definição de quantidade de clusters, porém, precisa dos parâmetros de definição de densidade dos agrupamentos,  $minPt$ , e do raio ao qual se aplicará a densidade. A definição destes parâmetros pode ser considerada como limitação do algoritmo, porém, diversos estudos já foram feitos apresentando valores eficientes para  $minPt$  e técnicas para calcular automaticamente o raio ([Tan et al. 2005]). Seguindo as orientações deste autor, foi configurado o valor do parâmetro  $minPt=4$  e o valor do raio calculado automaticamente: dado o valor de  $minPt$ , procura-se a distância de cada ponto para o seu  $minPt^{\circ}$  vizinho mais próximo. Os valores são colocados em ordem crescente e espera-se que em uma dada região haverá um aumento considerável no valor das distâncias. Se encontrado, este é o valor apropriado para o raio. Caso não exista essa região, o raio é definido como o valor médio dessas distâncias.

Uma característica interessante observada na implementação do algoritmo DBScan é

que ele obtém uma quantidade mínima de clusters, sendo em média 2 clusters, favorecendo a escalabilidade do sistema XPrefRec no momento da recomendação. Além disso, ele exclui alguns usuários que considera como ruído dentro da base de dados.

A tabela 6.2 abaixo apresenta os resultados obtidos com esses dois últimos algoritmos. É possível observar que o DBScan obtém, na maioria das medidas, melhores resultados (em negrito) que o Cure. A característica do algoritmo de obter clusters de formatos variados favorece o aumento da acurácia em relação ao Cure, que obtém clusters esféricos.

Tabela 6.2: Resultados do Grupo de Teste 1

BD <sub>100</sub>	Acurácia						Cobertura	
	$p_f$	$r_f$	$p_F$	$r_F$	$p_G$	$r_G$	$p_C$	$r_C$
DBScan	<b>76.32%</b>	<b>74.50%</b>	<b>69.41%</b>	<b>56.93%</b>	<b>76.10%</b>	<b>73.85%</b>	63.16%	<b>49.96%</b>
CURE	75.15%	72.27%	70.99%	55.40%	75.02%	71.64%	<b>63.22%</b>	48.57%

Em comparação com os valores da abordagem tradicional, tabela 6.1, a alternativa DBScan consegue superar nas medidas de acurácia fraca e global, mostrando que a metodologia consegue alcançar um equilíbrio entre escalabilidade e acurácia.

Observa-se também que os valores obtidos na medida de cobertura não são interessantes. A revocação se encontra abaixo de 50%, o que não apresenta um desempenho bom do sistema, no primeiro grupo de teste, em relação ao problema de *cold-start* de item. O valor obtido na revocação forte de 56.93% também não é tão interessante e não supera o apresentado pela abordagem padrão de Filtragem Colaborativa.

Após a obtenção do melhor resultado com o algoritmo DBScan, testou-se uma variação no valor de  $minPt$ , para verificar se densidades menores retornariam resultados mais interessantes a partir da criação de mais clusters. Como pode ser visto na tabela 6.3, valores menores que  $minPt=4$  não retornam recomendações mais acuradas, ratificando o que é sugerido em [Tan et al. 2005], para a definição do valor 4 para o parâmetro  $minPt$ .

Tabela 6.3: Variação do valor de  $minPt$  no algoritmo DBScan

BD <sub>100</sub>	Acurácia						Cobertura	
	$p_f$	$r_f$	$p_F$	$r_F$	$p_G$	$r_G$	$p_C$	$r_C$
$Minpt = 4$	<b>76.32%</b>	<b>74.50%</b>	<b>69.41%</b>	<b>56.93%</b>	<b>76.10%</b>	<b>73.85%</b>	<b>63.16%</b>	<b>49.96%</b>
$Minpt = 3$	74.54%	73.15%	67.93%	55.75%	73.89%	71.15%	62.47%	49.12%
$Minpt = 2$	74.63%	72.99%	67.83%	55.53%	73.96%	70.98%	62.57%	49.25%
$Minpt = 1$	74.32%	73.25%	67.73%	55.70%	73.68%	71.23%	62.49%	49.13%

Assim, fixamos como melhor alternativa do grupo de teste 1, o algoritmo DBScan com o parâmetro  $minPt=4$  e o cálculo do raio de forma automática.

### 6.1.2 Grupo de Teste 2

O grupo de teste 2 testa as cinco alternativas de agregação de preferências, para verificar qual consegue melhor representar os clusters, para alcançar recomendações mais

acuradas. A tabela 6.4 apresenta os resultados obtidos. As alternativas são as apresentadas na seção 4.3 e ordenadas da seguinte forma:

- Alternativa 1: Média aritmética;
- Alternativa 2: Média ponderada pelo coeficiente de silhueta;
- Alternativa 3: Quantificador fuzzy *Most*;
- Alternativa 4: Quantificador fuzzy *At least half*;
- Alternativa 5: Quantificador fuzzy *As many as possible*.

Tabela 6.4: Resultados do Grupo de teste 2

$BD_{100}$	Acurácia						Cobertura	
	$p_f$	$r_f$	$p_F$	$r_F$	$p_G$	$r_G$	$p_C$	$r_C$
Alternativa 1	<b>76.32%</b>	<b>74.50%</b>	<b>69.41%</b>	<b>56.93%</b>	<b>76.10%</b>	<b>73.85%</b>	63.16%	49.96%
Alternativa 2	76.19%	74.36%	69.36%	56.92%	75.97%	73.71%	<b>63.22%</b>	<b>50.12%</b>
Alternativa 3	63.74%	61.08%	66.34%	51.23%	63.82%	60.69%	56.92%	43.50%
Alternativa 4	63.37%	60.77%	66.65%	51.19%	63.47%	60.41%	57.83%	43.79%
Alternativa 5	63.73%	60.89%	66.89%	51.23%	63.83%	60.53%	59.05%	44.11%

Observa-se que as alternativas 1 e 2 obtiveram os melhores resultados de acurácia e cobertura, não sendo a diferença entre as duas alternativas tão significativa. Observa-se também um valor um pouco superior da média ponderada nas medidas de cobertura. Os quantificadores fuzzy, entretanto, não obtiveram resultados tão interessantes. Como melhor alternativa, e agregando o fato de ser de fácil implementação, a alternativa 1 de média aritmética é definida como a melhor alternativa entre as propostas para obtenção de consenso.

Os valores obtidos no grupo de teste 2 se mantém idênticos ao do grupo de teste 1, mostrando que a etapa de obtenção de consenso, com as alternativas testadas, não são influentes no desempenho do sistema XPrefRec.

### 6.1.3 Grupo de Teste 3

O grupo de teste 3 testa as alternativas de algoritmos de mineração de preferências. A primeira alternativa é o algoritmo Cprefminer e a segunda é o algoritmo Cprefminer acrescido da técnica de otimização *Range Voting*, o CprefMiner\*. Os resultados obtidos são apresentados na tabela 6.5.

Tabela 6.5: Resultados do Grupo de Teste 3

$BD_{100}$	Acurácia						Cobertura	
	$p_f$	$r_f$	$p_F$	$r_F$	$p_G$	$r_G$	$p_C$	$r_C$
CprefMiner	76.32%	74.50%	69.41%	56.93%	76.10%	73.85%	63.16%	49.96%
CprefMiner*	<b>89.56%</b>	<b>88.89%</b>	<b>78.87%</b>	<b>78.24%</b>	<b>89.16%</b>	<b>88.49%</b>	<b>79.30%</b>	<b>78.79%</b>

Observa-se uma melhora significativa nos resultados apresentados. Todas as medidas se encontram acima de 78%, o que pode ser considerado como um bom resultado. A medida de cobertura melhora significativamente em relação aos dados do grupo de teste 2. Todos os valores se encontram acima dos valores obtidos pela abordagem de Filtragem Colaborativa apresentados na tabela 6.1.

É possível observar que entre as alternativas testadas, o minerador de preferência é o fator que mais influencia no desempenho do sistema XPrefRec. Quanto mais acurado o minerador de preferências é, mais acuradas serão as recomendações feitas pelo Sistema de Recomendação. Assim, esse resultado sugere enfatizar que é importante a pesquisa em mineradores de preferências cada vez mais acurados, para serem empregados em Sistemas de Recomendação baseados em mineração de preferências.

A configuração final do Sistema XPrefRec é o apresentado na tabela 6.6 .

Tabela 6.6: Sistema XPrefRec

Etapa 2 - Agrupamento dos usuários	DBScan
Etapa 3 - Agregação de preferências	Média aritmética
Etapa 4 - Mineração de preferências	Cprefminer*

## 6.2 Comparação com Outros Trabalhos

Como explanado no capítulo 5, a proposta apresentada por [Melville et al. 2002], o CBCF, foi implementada para servir de comparação com o sistema XPrefRec. Além de comparar as medidas de acurácia, cobertura, novidade e serendipity, nós testamos os dois sistemas nos outros bancos definidos na tabela 5.1, para verificar o comportamento dos dois em relação à esparsidade dos dados. Apresentamos a seguir os resultados obtidos por medida de validação.

### 6.2.1 Acurácia

Como ressaltado na seção 2.1.3, o mínimo que se espera de um Sistema de Recomendação é que ele consiga acertar as preferências do usuário. As tabelas 6.7 e 6.8 apresentam os resultados obtidos pelos dois sistemas, XPrefRec e CBCF, em relação à acurácia que eles alcançam em cada banco de dados.

Claramente se observa o melhor desempenho do sistema XPrefRec em relação ao sistema CBCF. Este último supera os valores obtidos pela abordagem de Filtragem Colaborativa no  $BD_{100}$ , excetuando a medida de acurácia forte.

Em relação à esparsidade dos dados também é possível observar que o XPrefRec consegue um comportamento melhor que o sistema CBCF. Enquanto que, na ausência de 57% dos dados de avaliação,  $BD_{50}$ , o XPrefRec ainda alcança resultados acima dos 66%, o CBCF fica abaixo do 40% na acurácia forte e global.

Tabela 6.7: Comparação XPrefRec e CBCF - Acurácia (Valores fortes e fracos)

Bancos	$p_f$		$r_f$		$p_F$		$r_F$	
	XPrefRec	CBCF	XPrefRec	CBCF	XPrefRec	CBCF	XPrefRec	CBCF
$BD_{100}$	<b>89.56%</b>	73.20%	<b>88.89%</b>	73.20%	<b>78.87%</b>	70.14%	<b>78.24%</b>	70.54%
$BD_{90}$	<b>88.04%</b>	72.52%	<b>85.41%</b>	72.52%	<b>78.45%</b>	69.48%	<b>76.55%</b>	69.48%
$BD_{80}$	<b>83.45%</b>	70.95%	<b>83.25%</b>	70.95%	<b>77.75%</b>	64.74%	<b>77.35%</b>	64.74%
$BD_{70}$	<b>80.45%</b>	65.96%	<b>78.45%</b>	65.96%	<b>76.60%</b>	62.41%	<b>75.54%</b>	62.41%
$BD_{60}$	<b>77.25%</b>	65.29%	<b>68.55%</b>	65.29%	<b>73.76%</b>	51.88%	<b>69.41%</b>	51.88%
$BD_{50}$	<b>75.96%</b>	59.36%	<b>66.07%</b>	59.36%	<b>75.00%</b>	37.14%	<b>68.78%</b>	37.14%

Tabela 6.8: Comparação XPrefRec e CBCF - Acurácia (Valores globais)

Bancos	$p_G$		$r_G$		F1	
	XPrefRec	CBCF	XPrefRec	CBCF	XPrefRec	CBCF
$BD_{100}$	<b>89.16%</b>	70.64%	<b>88.49%</b>	70.64%	<b>88.82%</b>	70.64%
$BD_{90}$	<b>87.71%</b>	69.59%	<b>85.10%</b>	69.59%	<b>86.39%</b>	69.59%
$BD_{80}$	<b>83.32%</b>	64.88%	<b>83.12%</b>	64.88%	<b>83.22%</b>	64.88%
$BD_{70}$	<b>80.38%</b>	62.47%	<b>78.39%</b>	62.47%	<b>79.37%</b>	62.47%
$BD_{60}$	<b>77.20%</b>	52.07%	<b>68.56%</b>	52.07%	<b>72.62%</b>	52.07%
$BD_{50}$	<b>75.95%</b>	37.39%	<b>66.10%</b>	37.39%	<b>70.68%</b>	37.39%

Pode-se concluir que o XPrefRec é um Sistema de Recomendação mais acurado que o sistema CBCF e que apresenta um bom nível de acurácia mesmo com alta esparsidade dos dados. Pela medida de acurácia, podemos concluir que o XPrefRec, construído a partir da metodologia PrefRec, é uma alternativa promissora para implementação de Sistemas de Recomendação.

### 6.2.2 Cobertura

A medida de cobertura visa avaliar como os sistemas se comportam em relação ao problema de *cold-start* de itens. Ambos os sistemas são capazes de recomendar itens que ainda não possuem nenhuma avaliação. A tabela 6.9 apresenta os resultados obtidos pelo sistema XPrefRec e CBCF na medida de cobertura.

Tabela 6.9: Comparação XPrefRec e CBCF - Cobertura

Bancos	$p_G$		$r_G$	
	XPrefRec	CBCF	XPrefRec	CBCF
$BD_{100}$	<b>79.30%</b>	61.19%	<b>78.79%</b>	61.16%
$BD_{90}$	<b>78.74%</b>	61.60%	<b>76.90%</b>	61.57%
$BD_{80}$	<b>73.28%</b>	61.25%	<b>73.07%</b>	61.23%
$BD_{70}$	<b>72.84%</b>	64.72%	<b>66.21%</b>	64.72%
$BD_{60}$	<b>73.19%</b>	60.89%	<b>45.83%</b>	60.87%
$BD_{50}$	<b>74.63%</b>	51.00%	<b>40.40%</b>	50.98%

Observa-se novamente o melhor desempenho do XPrefRec na maioria dos bancos testados. Diferente da medida de acurácia, nos bancos mais esparsos, o CBCF obteve melhores

resultados na revocação da cobertura. Porém, a aplicação do CBCF no banco  $BD_{50}$ , é inviabilizada pelos resultados alcançados na medida de acurácia.

O melhor desempenho do XPrefRec se deve à utilização das regras de preferências contextuais mineradas pelo CprefMiner\*, que utiliza-se de uma rede de preferência bayesiana, que considera a dependência entre as características dos itens. O CBCF utiliza um classificador *naive bayes*, o qual assume que as características são independentes umas das outras no cálculo das probabilidades de recomendação.

### 6.2.3 Novidade

A medida de novidade visa avaliar se os itens recomendados serão itens novos aos usuários. A definição de novo, dada por [Shani e Gunawardana 2011], é o item que não é tão popular, aquele que não foi avaliado por muitos usuários. Os resultados obtidos pelos dois sistemas são expostos na tabela 6.10. As medidas de novidade e *serendipity* foram calculadas sobre os bancos  $BD_{100}$ ,  $BD_{70}$  e  $BD_{50}$ .

Tabela 6.10: Comparação XPrefRec e CBCF - Novidade

Bancos	Novidade	
	XPrefRec	CBCF
$BD_{100}$	133.70	<b>121.73</b>
$BD_{70}$	158.25	<b>140.02</b>
$BD_{50}$	152.27	<b>140.51</b>

Quanto menor o valor da novidade mais o sistema recomenda itens menos populares. Os resultados apresentam um melhor desempenho do sistema CBCF. Porém, vale ressaltar que os itens recomendados pelo CBCF, mesmo que menos populares, acertam menos as preferências dos usuários, isto é, possuem menor acurácia do que o sistema XPrefRec.

Considerando que o conjunto de usuários utilizados para o cálculo da novidade (usuários de treinamento) tem em média 237 usuários a cada iteração da validação, observa-se que ambos os sistemas recomendam, em média, itens avaliados por mais de 50% dos usuários desse conjunto. Lembrando que os usuários de treinamento são usuários que avaliaram um grande conjunto de itens, assumimos que este valor de 50% não corresponde a um valor de novidade tão interessante. Mesmo o CBCF alcançando melhores resultados, também consideramos o desempenho do XPrefRec em relação à medida de novidade satisfatório.

### 6.2.4 Serendipity

Neste teste avaliamos se os itens recomendados pelos sistemas são surpreendentes ao usuário. A definição de surpreendente mede o quão distantes as características dos itens recomendados são dos itens que o usuário ativo já avaliou.

A tabela 6.11 apresenta o valor médio de *serendipity* obtido para cada atributo dos itens. Os atributos 1, 2 e 3 correspondem às características de diretor, idioma e ano, respectivamente. Os atributos de 4 a 8 correspondem às características de gênero, e de 9 a 12 aos atores que atuam nos filmes.

Tabela 6.11: Comparação XPrefRec e CBCF - Serendipity

Atributos	$BD_{100}$		$BD_{70}$		$BD_{50}$	
	XPrefRec	CBCF	XPrefRec	CBCF	XPrefRec	CBCF
1	0.79	0.74	0.79	0.75	0.79	0.75
2	0.21	0.31	0.21	0.30	0.18	0.18
3	0.45	0.64	0.32	0.57	0.31	0.51
4	0.80	0.83	0.78	0.82	0.78	0.81
5	0.57	0.66	0.51	0.65	0.51	0.59
6	0.34	0.44	0.34	0.41	0.32	0.32
7	0.11	0.16	0.09	0.15	0.08	0.12
8	0.44	0.45	0.44	0.44	0.43	0.43
9	0.89	0.89	0.86	0.86	0.86	0.86
10	0.91	0.92	0.90	0.91	0.90	0.91
11	0.93	0.94	0.93	0.93	0.93	0.93
12	0.94	0.94	0.94	0.94	0.94	0.94
Média	0.62	<b>0.66</b>	0.59	<b>0.64</b>	0.59	<b>0.61</b>

Quanto maior o valor de *serendipity* mais surpreendentes são os itens recomendados pelo sistema. Nos três bancos testados, o sistema CBCF consegue melhores resultados. O XPrefRec, entretanto, alcança valores bem próximos.

Esse resultado permite observar quais das características dos itens são mais surpreendentes aos usuários. Observa-se que os atributos de 9 a 12, que correspondem aos atores dos filmes, são os que apresentam maior valor de *serendipity*. Isso se deve ao fato do domínio dos atores ser muito maior, fazendo com que existam poucos filmes com o mesmo ator dentro do conjunto de filmes que o usuário já avaliou. Da mesma forma acontece para o atributo 1, que corresponde ao diretor do filme. O atributo 4, que corresponde à primeira descrição do gênero do filme, também aparece com alto valor de *serendipity*. Pode-se afirmar que o primeiro gênero informado para o filme tenta classificá-lo melhor, diferenciando-o dos demais, enquanto os outros quatro gêneros que o descrevem o associa com os demais filmes.

### 6.2.5 Tempo de Execução

Também contabilizamos quanto tempo os sistemas testados levam para realização de suas subtarefas. A tabela 6.12 apresenta os resultados obtidos.

A tarefa de construção do modelo compreende, no XPrefRec, construir o modelo de recomendação (Etapas de 1 a 4). Para o CBCF, a construção do modelo é a predição dos

Tabela 6.12: Tempo de execução

Tempo	XprefRec	CBCF
Construção do modelo	6 m 19 s	54 min 14 s
Comparação com os colaboradores	19 ms	3 s 386 ms
Cálculo da recomendação	69s 627 ms	16 min 27 s

valores de avaliação faltantes na matriz usuário×item, a partir da abordagem Baseada em Conteúdo. O XPrefRec possui um tempo consideravelmente menor em relação ao CBCF.

O tempo de comparação com os colaboradores compreende, no CBCF, a comparação do usuário ativo com todos os usuários do sistema, e para o XPrefRec a comparação com todos os perfis de preferência. Os resultados apresentados confirmam ser o XPrefRec um sistema escalável e eficaz, realizando menos comparações no momento da recomendação.

O cálculo da recomendação compreende a construção do *ranking* com 50 itens em ambas as abordagens. A essa tarefa é dado o conjunto itens-cobertura, que compreende uma média de 1300 itens. Mais uma vez o XPrefRec apresenta melhores tempos em relação ao CBCF. Entretanto, mesmo sendo um tempo menor, ainda não é um tempo razoável para um sistema *online*. O tempo de 69 segundos é demasiado grande para o usuário poder receber recomendações a partir de um conjunto de 1300 itens. Em um sistema real, este conjunto provavelmente é muito maior.



# Capítulo 7

## Conclusões e Trabalhos Futuros

Neste trabalho de dissertação apresentamos a PrefRec: uma metodologia para desenvolvimento de Sistemas de Recomendação utilizando algoritmos de mineração de preferências. Objetivamos com essa proposta a construção de Sistemas de Recomendação capazes de recomendar itens ainda não avaliados, que tenham bons valores de acurácia e que permitam uma interação com o usuário mais satisfatória.

A metodologia propõe uma sequência de etapas *offline*, que gera um modelo de recomendação composto por perfis de preferências. Os perfis, por sua vez, são compostos por um modelo de preferência e uma matriz de preferências. No momento da recomendação, realizada *online*, é associado um dos perfis de preferência ao usuário ativo, para que a partir do seu respectivo modelo de preferência, este perfil compare os itens e os recomende. Além de apresentarmos uma nova metodologia para desenvolvimento de Sistemas de Recomendação, também acrescentamos contribuições ao propor uma nova maneira de representação dos usuários, através dos graus de preferências, e a utilização de um modelo de preferência que seja capaz de inferir, dados dois itens, qual é o preferido pelo usuário a partir das características que eles apresentam sem nunca terem sido avaliados.

A validação da metodologia através da implementação do sistema XPrefRec possibilitou analisar quais fatores, dentre as etapas da metodologia, mais influencia na acurácia do Sistema de Recomendação. Alternativas de agregação de preferências foram testadas, assim, como algoritmos de clusterização de diferentes paradigmas. Porém, o fator que mais impactou no aumento da acurácia da recomendação foi o minerador de preferências. O minerador CPrefMiner, utilizado no nosso estudo de caso, demonstrou alto desempenho quando associado a técnica de otimização *Range Voting*, obtendo os melhores resultados de acurácia e cobertura.

Os resultados experimentais também demonstram que, comparado ao trabalho que definimos como *baseline* - o CBCF - o XPrefRec também alcançou melhores resultados em relação a acurácia e cobertura das recomendações. Em relação às medidas de novidade e *serendipity*, que influenciam em uma interação do usuário mais satisfatória, o XPrefRec apresenta resultados um pouco inferiores, porém, sem prejuízo da relevância da

metodologia, uma vez que acerta mais a preferência do usuário quando recomenda.

A metodologia PrefRec também alcança Sistemas de Recomendação escaláveis: uma grande quantidade de usuário, itens e avaliações, não interfere em grande tempo no cálculo das recomendações. A definição de perfis de preferências a partir da clusterização dos usuários permite que o usuário ativo seja comparado no momento da recomendação, com uma pequena quantidade de perfis. No estudo de caso, quando aplicamos o algoritmo de clusterização DBScan, essa comparação é feita, geralmente, duas vezes.

Por fim, concluímos que a metodologia PrefRec, com todas as suas características, é uma metodologia promissora para desenvolvimento de Sistemas de Recomendação, alcançando bons valores de acurácia e cobertura, e valores aceitáveis de novidade e *serendipity*.

## Trabalhos Futuros

Durante o desenvolvimento da pesquisa, várias questões foram levantadas para otimização e validação da metodologia PrefRec, que podem proporcionar pesquisas futuras.

Uma das questões levantadas é a validação da metodologia em bancos de dados de outros domínios, que não o de recomendação de filmes, para verificar sua eficiência em cenários diferentes. A variação da quantidade de usuários, itens e também avaliações se torna relevante para verificar o desempenho da metodologia.

Para melhor justificar a escolha da metodologia PrefRec em relação a outros Sistemas de Recomendação, é pertinente calcular outras medidas de validação como robustez, confiança, diversidade e a outra definição apresentada para a medida de cobertura: entre os itens do banco de dados quantos o sistema consegue recomendar. Isto se torna interessante uma vez que a empresa que aplica o Sistema de Recomendação como ferramenta de inteligência, objetiva vender todos os seus produtos, e não só uma parte deles. Além de outras medidas de validação, também é necessário comparar a metodologia PrefRec com outros trabalhos da literatura.

Todos os testes apresentados nos resultados experimentais foram realizados *offline*. A validação da metodologia com testes *online*, isto é, com usuário reais, se torna interessante para que eles informem o quanto ficam satisfeitos em relação às recomendações que o Sistema de Recomendação calcula, e quais são os fatores que favorecem a interação usuário/sistema.

Em relação ao problema de *cold-start* de usuário, não tratado neste trabalho, é relevante verificar a quantidade de avaliações que o usuário necessita fazer para começar a receber boas recomendações. Nos testes realizados, utilizamos todas as avaliações disponíveis para os itens de treinamento, em cada iteração do *cross-validation*.

Entre as diversas alternativas apresentadas para a metodologia PrefRec pode-se tam-

bém verificar a utilização e o impacto de outras tantas alternativas como: medidas de distâncias diferenciadas entre matrizes de preferências; diferentes representações de usuários como o vetor de avaliações usado tradicionalmente; outras alternativas para agregação de preferências, além de outros mineradores de preferências como o ProfMiner, apresentado na seção XX, e o CprefMiner com a versão que explora valores multivalorados.

Porém, a definição de quais são as melhores alternativas a serem usadas na metodologia PrefRec é um problema difícil, devido a grande quantidade de alternativas que podem ser aplicadas. Desta forma, sugere-se como principal trabalho futuro aplicar uma análise factorial, para tornar a metodologia PrefRec proposta mais rigorosa.

A análise factorial, apresentada em [Jain 1991] e [Montgomery 2006] , é umas das técnicas mais aplicadas para esta situação. A ideia é obter o máximo de informação sobre alternativas com o número mínimo de experimentos. Além disso, permite determinar se o efeito de uma alternativa é significante ou não. Existem trabalhos recentes que exploram esta técnica como os apresentados em [de Lima et al. 2010] e [Brandão et al. 2013]. Este último faz a análise dos fatores impactantes em um Sistema de Recomendação de colaborações acadêmicas.



# Referências Bibliográficas

- [Adomavicius e Tuzhilin 2005] Adomavicius, G. e Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749.
- [Ahn et al. 2007] Ahn, J.-w., Brusilovsky, P., Grady, J., He, D., e Syn, S. Y. (2007). Open User Profiles for Adaptive News Systems: Help or Harm? In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pp. 11–20, Banff, Alberta, Canada, New York, NY, USA. ACM.
- [Amatriain 2013] Amatriain, X. (2013). Mining Large Streams of User Data for Personalized Recommendations. *SIGKDD Explor. Newsl.*, 14(2):37–48.
- [Amatriain et al. 2011] Amatriain, X., Jaimes, A., Oliver, N., e Pujol, J. M. (2011). Data Mining Methods for Recommender Systems. In *Recommender Systems Handbook*, pp. 39–71.
- [Billsus et al. 2002] Billsus, D., Brunk, C. A., Evans, C., Gladish, B., e Pazzani, M. (2002). Adaptive Interfaces for Ubiquitous Web Access. *Commun. ACM*, 45(5):34–38.
- [Billsus e Pazzani 2000] Billsus, D. e Pazzani, M. J. (2000). User Modeling for Adaptive News Access. *User Model. User-Adapt. Interact.*, 10(2-3):147–180.
- [Blanco-Fernández et al. 2008] Blanco-Fernández, Y., Arias, J. J. P., Gil-Solla, A., Cabrer, M. R., e Nores, M. L. (2008). Providing entertainment by content-based filtering and semantic reasoning in intelligent recommender systems. *IEEE Trans. Consumer Electronics*, 54(2):727–735.
- [Bouza et al. 2008] Bouza, A., Reif, G., Bernstein, A., e Gall, H. (2008). SemTree: Ontology-Based Decision Tree Algorithm for Recommender Systems. In Bizer, C. e Joshi, A. (editores), *International Semantic Web Conference (Posters and Demos)*, volume 401 de *CEUR Workshop Proceedings*. CEUR-WS.org.
- [Brandão e Moro 2012] Brandão, M. A. e Moro, M. M. (2012). Recomendação de colaboração em redes sociais acadêmicas baseada na afiliação dos pesquisadores. In *SBBD*.
- [Brandão et al. 2013] Brandão, M. A., Moro, M. M., e Almeida, J. M. (2013). Análise de fatores impactantes na recomendação de colaborações acadêmicas utilizando projeto fatorial. In *SBBD*.
- [Burke 2007] Burke, R. (2007). Hybrid Web Recommender Systems. In *The Adaptive Web: Methods and Strategies of Web Personalization*, chapter 12, pp. 377–408. Berlin.

- [Chiclana et al. 2001] Chiclana, F., Herrera, F., e Herrera-Viedma, E. (2001). Integrating multiplicative preference relations in a multipurpose decision-making model based on fuzzy preference relations. *Fuzzy Sets and Systems*, 122(2):277–291.
- [Chomicki 2003] Chomicki, J. (2003). Preference Formulas in Relational Queries. *ACM Trans. Database Syst.*, 28(4):427–466.
- [Cohen et al. 1999] Cohen, W. W., Schapire, R. E., e Singer, Y. (1999). Learning to Order Things. *J. Artif. Int. Res.*, 10(1):243–270.
- [de Amo et al. 2012a] de Amo, S., Bueno, M. L. P., Alves, G., e da Silva, N. F. F. (2012a). CPrefMiner: An Algorithm for Mining User Contextual Preferences Based on Bayesian Networks. In *ICTAI*, pp. 114–121.
- [de Amo et al. 2013a] de Amo, S., Bueno, M. L. P., Alves, G., e da Silva, N. F. F. (2013a). Mining User Contextual Preferences. *JIDM*, 4(1):37–46.
- [de Amo et al. 2012b] de Amo, S., Diallo, M. S., Diop, C. T., Giacometti, A., Li, H. D., e Soulet, A. (2012b). Mining Contextual Preference Rules for Building User Profiles. In Cuzzocrea, A. e Dayal, U. (editores), *DaWaK*, volume 7448 de *Lecture Notes in Computer Science*, pp. 229–242. Springer.
- [de Amo et al. 2013b] de Amo, S., Diallo, M. S., Diop, C. T., Giacometti, A., Li, H. D., e Soulet, A. (2013b). Contextual Preference Mining for User Profile Construction. *to appear*.
- [de Lima et al. 2010] de Lima, E. B., Pappa, G. L., de Almeida, J. M., Gonçalves, M. A., e Jr., W. M. (2010). Tuning Genetic Programming parameters with factorial designs. In *IEEE Congress on Evolutionary Computation*, pp. 1–8. IEEE.
- [Degemmis et al. 2007] Degemmis, M., Lops, P., e Semeraro, G. (2007). A Content-collaborative Recommender That Exploits WordNet-based User Profiles for Neighborhood Formation. *User Modeling and User-Adapted Interaction*, 17(3):217–255.
- [Desrosiers e Karypis 2011] Desrosiers, C. e Karypis, G. (2011). A Comprehensive Survey of Neighborhood-based Recommendation Methods. In Ricci, F., Rokach, L., Shapira, B., e Kantor, P. B. (editores), *Recommender Systems Handbook*, pp. 107–144. Springer.
- [Dey 2001] Dey, A. K. (2001). Understanding and Using Context. *Personal Ubiquitous Comput.*, 5(1):4–7.
- [Ester et al. 1996] Ester, M., peter Kriegel, H., S, J., e Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. pp. 226–231. AAAI Press.
- [Fouss e Saerens 2008] Fouss, F. e Saerens, M. (2008). Evaluating Performance of Recommender Systems: An Experimental Comparison. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '08, pp. 735–738, Washington, DC, USA. IEEE Computer Society.

- [García-Lapresta e Meneses 2005] García-Lapresta, J. L. e Meneses, L. C. (2005). Individual-valued preferences and their aggregation: consistency analysis in a real case. *Fuzzy Sets and Systems*, 151(2):269–284.
- [Goldberg et al. 1992] Goldberg, D., Nichols, D., Oki, B. M., e Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70.
- [Goldberg et al. 2001] Goldberg, K., Roeder, T., e Perkins, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4:133–151.
- [Groh e Ehmig 2007] Groh, G. e Ehmig, C. (2007). Recommendations in Taste Related Domains: Collaborative Filtering vs. Social Filtering. In *Proceedings of the 2007 International ACM Conference on Supporting Group Work*, GROUP '07, pp. 127–136, New York, NY, USA. ACM.
- [Guha et al. 2001] Guha, S., Rastogi, R., e Shim, K. (2001). Cure: An Efficient Clustering Algorithm for Large Databases. *Inf. Syst.*, 26(1):35–58.
- [Guy et al. 2009] Guy, I., Zwerdling, N., Carmel, D., Ronen, I., Uziel, E., Yoge, S., e Ofek-Koifman, S. (2009). Personalized Recommendation of Social Software Items Based on Social Relations. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pp. 53–60, New York, NY, USA. ACM.
- [Han et al. 2011] Han, J., Kamber, M., e Pei, J. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition.
- [Herlocker et al. 1999] Herlocker, J. L., Konstan, J. A., Borchers, A., e Riedl, J. (1999). An Algorithmic Framework for Performing Collaborative Filtering. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pp. 230–237, New York, NY, USA. ACM.
- [Herlocker et al. 2004] Herlocker, J. L., Konstan, J. A., Terveen, L. G., John, e Riedl, T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22:5–53.
- [Holland et al. 2003] Holland, S., Ester, M., e Kießling, W. (2003). Preference Mining: A Novel Approach on Mining User Preferences for Personalized Applications. In Lavrac, N., Gamberger, D., Blockeel, H., e Todorovski, L. (editores), *PKDD*, volume 2838 de *Lecture Notes in Computer Science*, pp. 204–216. Springer.
- [Huang et al. 2004] Huang, Z., Chen, H., e Zeng, D. (2004). Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering. *ACM Trans. Inf. Syst.*, 22(1):116–142.
- [Iaquinta et al. 2008] Iaquinta, L., de Gemmis, M., Lops, P., Semeraro, G., Filannino, M., e Molino, P. (2008). Introducing Serendipity in a Content-Based Recommender System. In Xhafa, F., Herrera, F., Abraham, A., Köppen, M., e Benítez, J. M. (editores), *HIS*, pp. 168–173. IEEE Computer Society.
- [Jain 1991] Jain, R. K. (1991). *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley, 1 edition.

- [Jannach 2008] Jannach, D. (2008). Finding Preferred Query Relaxations in Content-Based Recommenders. In *Intelligent Techniques and Tools for Novel System Architectures*, volume 109 de *Studies in Computational Intelligence*, pp. 81–97. Springer, Berlin / Heidelberg.
- [Jiang et al. 2008] Jiang, B., Pei, J., Lin, X., Cheung, D. W., e Han, J. (2008). Mining Preferences from Superior and Inferior Examples. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pp. 390–398, New York, NY, USA. ACM.
- [Ko e Lee 2002] Ko, S.-J. e Lee, J.-H. (2002). User Preference Mining Through Collaborative Filtering and Content Based Filtering in Recommender System. In *Proceedings of the Third International Conference on E-Commerce and Web Technologies*, EC-WEB '02, pp. 244–253, London, UK, UK. Springer-Verlag.
- [Krulwich 1997] Krulwich, B. (1997). Lifestyle Finder: Intelligent User Profiling Using Large-Scale Demographic Data. *Artificial Intelligence Magazine*, 18(2):37–45.
- [Lieberman 1995] Lieberman, H. (1995). Letizia: An Agent That Assists Web Browsing. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pp. 924–929, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- [Linden et al. 2003] Linden, G., Smith, B., e York, J. (2003). Amazon.com recommendations. Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80.
- [Llamazares et al. 2013] Llamazares, B., Pérez-Asurmendi, P., e GarcíA-Lapresta, J. L. (2013). Collective Transitivity in Majorities Based on Difference in Support. *Fuzzy Sets Syst.*, 216:3–15.
- [Lops et al. 2011] Lops, P., de Gemmis, M., e Semeraro, G. (2011). Content-based Recommender Systems: State of the Art and Trends. In *Recommender Systems Handbook*, pp. 73–105.
- [Luo et al. 2008] Luo, H., Niu, C., Shen, R., e Ullrich, C. (2008). A Collaborative Filtering Framework Based on Both Local User Similarity and Global User Similarity. *Mach. Learn.*, 72(3):231–245.
- [Ma et al. 2007] Ma, H., King, I., e Lyu, M. R. (2007). Effective Missing Data Prediction for Collaborative Filtering. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pp. 39–46, New York, NY, USA. ACM.
- [Mahmood e Ricci 2007] Mahmood, T. e Ricci, F. (2007). Towards Learning User-Adaptive State Models in a Conversational Recommender System. In Hinneburg, A. (editor), *LWA*, pp. 373–378. Martin-Luther-University Halle-Wittenberg.
- [Melville et al. 2002] Melville, P., Mooney, R. J., e Nagarajan, R. (2002). Content-boosted Collaborative Filtering for Improved Recommendations. In *Eighteenth National Conference on Artificial Intelligence*, pp. 187–192, Menlo Park, CA, USA. American Association for Artificial Intelligence.

- [Miller et al. 2003] Miller, B. N., Albert, I., Lam, S. K., Konstan, J. A., e Riedl, J. (2003). MovieLens Unplugged: Experiences with an Occasionally Connected Recommender System. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, IUI '03, pp. 263–266, New York, NY, USA. ACM.
- [Mladenic 1999] Mladenic, D. (1999). Machine learning used by Personal WebWatcher. In *Proceedings of ACAI-99 Workshop on Machine Learning and Intelligent Agents*, Chania, Crete, Greece.
- [Mobasher et al. 2002] Mobasher, B., Dai, H., Luo, T., e Nakagawa, M. (2002). Discovery and Evaluation of Aggregate Usage Profiles for Web Personalization. *Data Min. Knowl. Discov.*, 6(1):61–82.
- [Montgomery 2006] Montgomery, D. C. (2006). *Design and Analysis of Experiments*. John Wiley & Sons.
- [Mooney e Roy 1999] Mooney, R. J. e Roy, L. (1999). Content-Based Book Recommending Using Learning for Text Categorization. *CoRR*, cs.DL/9902011.
- [NCBI 2009] NCBI (2009). dbEST: database of “Expressed Sequence Tags”:summary by organisms. <http://www.netflix.com>. January 16, 2009.
- [Nurmi 1981] Nurmi, H. (1981). Approaches to collective decision making with fuzzy preference relations. *Fuzzy Sets and Systems*, 6(3):249 – 259.
- [Pazzani 1999] Pazzani, M. J. (1999). A Framework for Collaborative, Content-Based and Demographic Filtering. *Artif. Intell. Rev.*, 13(5-6):393–408.
- [Pronk et al. 2007] Pronk, V., Verhaegh, W., Proidl, A., e Tiemann, M. (2007). Incorporating User Control into Recommender Systems Based on Naive Bayesian Classification. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, RecSys '07, pp. 73–80, New York, NY, USA. ACM.
- [Resnick et al. 1994] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., e Riedl, J. (1994). GroupLens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, CSCW '94, pp. 175–186, New York, NY, USA. ACM.
- [Ricci et al. 2011] Ricci, F., Rokach, L., e Shapira, B. (2011). Introduction to Recommender Systems Handbook. In Ricci, F., Rokach, L., Shapira, B., e Kantor, P. B. (editores), *Recommender Systems Handbook*, pp. 1–35. Springer.
- [Sarwar et al. 2001] Sarwar, B., Karypis, G., Konstan, J., e Riedl, J. (2001). Item-based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pp. 285–295, New York, NY, USA. ACM.
- [Sen 1970] Sen, A. (1970). *Collective choice and social welfare*. Holden-Day.
- [Shani e Gunawardana 2011] Shani, G. e Gunawardana, A. (2011). Evaluating Recommendation Systems. In *Recommender Systems Handbook*, pp. 257–297.

- [Shardanand e Maes 1995] Shardanand, U. e Maes, P. (1995). Social Information Filtering: Algorithms for Automating “Word of Mouth”. In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, volume 1, pp. 210–217.
- [Stefanidis et al. 2011] Stefanidis, K., Koutrika, G., e Pitoura, E. (2011). A Survey on Representation, Composition and Application of Preferences in Database Systems. *ACM Trans. Database Syst.*, 36(3):19:1–19:45.
- [Su e Khoshgoftaar 2009] Su, X. e Khoshgoftaar, T. (2009). A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009:4.
- [Tan et al. 2005] Tan, P.-N., Steinbach, M., e Kumar, V. (2005). *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- [Wilson 2004] Wilson, N. (2004). Extending CP-Nets with Stronger Conditional Preference Statements. In McGuinness, D. L. e Ferguson, G. (editores), *AAAI*, pp. 735–741. AAAI Press / The MIT Press.
- [Xue et al. 2005] Xue, G.-R., Lin, C., Yang, Q., Xi, W., Zeng, H.-J., Yu, Y., e Chen, Z. (2005). Scalable Collaborative Filtering Using Cluster-based Smoothing. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pp. 114–121, New York, NY, USA. ACM.
- [Ziegler et al. 2005] Ziegler, C.-N., McNee, S. M., Konstan, J. A., e Lausen, G. (2005). Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, WWW '05, pp. 22–32, New York, NY, USA. ACM.