



**Pós-Graduação em Ciência da Computação**

**Investigação da Combinação de Filtragem Colaborativa e  
Recomendação Baseada em Confiança Através de Medidas de  
Esparsidade**

Por

***GABRIEL DE ALBUQUERQUE VELOSO AZUIRSON***

**Dissertação de Mestrado Profissional**



Universidade Federal de Pernambuco

[posgraduacao@cin.ufpe.br](mailto:posgraduacao@cin.ufpe.br)

[www.cin.ufpe.br/~posgraduacao](http://www.cin.ufpe.br/~posgraduacao)

**Recife, PE**

**2015**

**GABRIEL DE ALBUQUERQUE VELOSO AZUIRSON**

INVESTIGAÇÃO DA COMBINAÇÃO DE FILTRAGEM COLABORATIVA E  
RECOMENDAÇÃO BASEADA EM CONFIANÇA ATRAVÉS DE MEDIDAS DE  
ESPARSIDADE

**Dissertação de Mestrado** apresentada  
ao Centro de Informática da Universidade  
Federal de Pernambuco sob a orientação  
do **Prof. Dr. Ricardo Bastos C.  
Prudêncio**, como parte dos requisitos  
para obtenção do título de Mestre em  
Ciência da Computação.

**Recife, PE**

**2015**

Catalogação na fonte  
Bibliotecário Jefferson Luiz Alves Nazareno CRB4-1758

A997i Azuirson, Gabriel de Albuquerque Veloso.  
Investigação da combinação de filtragem colaborativa e recomendação baseada em confiança através de medidas de esparsidade / Gabriel de Albuquerque Veloso Azuirson. – Recife: O Autor, 2015.  
86 f.: fig., tab.

Orientador: Ricardo Bastos Cavalcante Prudêncio.  
Dissertação (Mestrado profissional) – Universidade Federal de Pernambuco. CIN, Ciência da Computação, 2015.  
Inclui referências.

1. Ciência da computação. 2. Inteligência artificial. 3. Algoritmos computacionais. I. Prudêncio, Ricardo Bastos Cavalcante (Orientador). II. Título.

004                    CDD (22. ed.)                    UFPE-MEI 2015-121

Dissertação de Mestrado Profissional apresentada por **Gabriel de Albuquerque Veloso Azuirson** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título, **“Investigação da Combinação de Filtragem Colaborativa e Recomendação Baseada em Confiança Através de Medidas de Esparsidade”**, orientada pelo Professor Ricardo Bastos Cavalcante Prudêncio e aprovada pela Banca Examinadora formada pelos professores:

---

Prof. Cleber Zanchettin

Centro de Informática / UFPE

---

Prof. Renato Fernandes Corrêa

Centro de Artes e Comunicação / UFPE

---

Prof. Ricardo Bastos Cavalcante Prudêncio

Centro de Informática / UFPE

Visto e permitida a impressão.

Recife, 06 de Agosto de 2015.

---

**Prof<sup>a</sup> Edna Natividade da Silva Barros**

Coordenadora de Pós-Graduação em Ciência da Computação  
da Universidade Federal de Pernambuco

Dedico este trabalho a minha família e a todos que direta ou indiretamente me ajudaram a concluir este trabalho.

## AGRADECIMENTOS

Gostaria de agradecer primeiramente a Deus, por ter me guiado até este momento com saúde e por ter colocado pessoas maravilhosas na minha vida: Minha mãe Martha, minha irmã Iris, meu padrasto Pedro e minhas sobrinhas pequenas Iasmin e Iara que têm trazido muito amor e sorte para a família. Gostaria de agradecer a minha namorada Bruna por me amar do jeito que eu sou, e por ter me dado carinho e apoio em todos os momentos.

Gostaria de agradecer ao meu orientador Ricardo Prudêncio por ter me aceitado como aluno e por ter me ajudado com seus ensinamentos que possibilitaram o desenvolvimento deste trabalho.

Aos meus amigos de trabalho do NTI da UFPE pelo convívio e troca de ideias que são essenciais para meu crescimento profissional e pessoal. A todos os meus amigos e parentes que me incentivam e que torcem pelo meu sucesso.

## RESUMO

Sistemas de recomendação têm desempenhado um papel importante em diferentes contextos de aplicação (e.g recomendação de produtos, filmes, músicas, livros, dentre outros). Eles automaticamente sugerem a cada usuário itens que podem ser relevantes, evitando que o usuário tenha que analisar uma quantidade gigantesca de itens para realizar sua escolha. Filtragem colaborativa (FC) é a abordagem mais popular para a construção de sistemas de recomendação, embora sofra com problemas relacionados à esparsidade dos dados (e.g., usuários ou itens com poucas avaliações). Neste trabalho, investigamos a combinação de técnicas de FC, representada pela técnica de Fatoração de Matrizes, e técnicas de recomendação baseada em confiança (RBC) em redes sociais para aliviar o problema da esparsidade dos dados. Sistemas de RBC têm se mostrado de fato efetivos para aumentar a qualidade das recomendações, em especial para usuários com poucas avaliações realizadas (e.g., usuários novos). Entretanto, o desempenho relativo entre técnicas de FC e de RBC pode depender da quantidade de informação útil presente nas bases de dados. Na arquitetura proposta nesse trabalho, as previsões geradas por técnicas de FC e de RBC são combinadas de forma ponderada através de medidas de esparsidade calculadas para usuários e itens. Para isso, definimos inicialmente um conjunto de medidas de esparsidade que serão calculadas sobre a matriz de avaliações usuários-itens e matriz de confiança usuários-usuários. Através de experimentos realizados utilizando a base de dados Epinions, observamos que a proposta de combinação trouxe uma melhoria nas taxas de erro e na cobertura em comparação com as técnicas isoladamente.

**Palavras-Chave:** Sistemas de Recomendação. Filtragem Colaborativa. Fatoração de Matriz. Sistemas de Recomendação Baseados em Confiança. Medidas de Esparsidade.

## ABSTRACT

Recommender systems have played an important role in different application contexts (e.g recommendation of products, movies, music, books, among others). They automatically suggest each user items that may be relevant, preventing the user having to analyze a huge amount of items to make your choice. Collaborative filtering (CF) is the most popular approach for building recommendation systems, although suffering with sparsity of the data-related issues (eg, users or items with few evaluations). In this study, we investigated the combination of CF techniques represented by matrix factorization technique, and trust-based recommendation techniques (TBR) on social networks to alleviate the problem of data sparseness. TBR systems have in fact proven to be effective to increase the quality of the recommendations, especially for users with few assessments already carried out (e.g., cold start users). However, the relative performance between CF and TBR techniques may depend on the amount of useful information contained in the databases. In the proposed architecture in this work, the predictions generated by CF and TBR techniques are weighted combined through sparsity measures calculated to users and items. To do this, first we define a set of sparsity measures that will be calculated on the matrix of ratings users-items and matrix of trust users-users. Through experiments using Epinions database, we note that the proposed combination brought an improvement in error rates and coverage compared to combined techniques.

Keywords: Recommender Systems. Collaborative Filtering. Matrix Factorization. Trust-Based Recommendation Systems. Sparsity Measures.

## LISTA DE FIGURAS

Figura 2.1 - Distribuição de cauda longa no conjunto de dados MovieLens e Netflix. O ponto (x,y) significa que x itens tiveram y avaliações. Fonte: Shi et al., 2014. ....	23
Figura 2.2 - FC baseada em memória.....	26
Figura 2.3 - Fatoração da matriz R. Fonte: Adaptada de Ma et al. (2008). ....	29
Figura 2.4 - Gráfico do erro de treinamento e teste.....	33
Figura 3.1 - Rede de confiança formada por 6 usuários e valores de confiança entre eles : Fonte: Ma et al. (2008).....	37
Figura 3.2 - Transitividade da confiança .....	39
Figura 3.3 - Rede de confiança. Fonte: Adaptada de Massa e Avesani (2007b) .....	41
Figura 3.4 - Passo 1 de MoleTrust. Fonte: Adaptada de Massa e Avesani (2007b) .	41
Figura 3.5 - Passo 2 de MoleTrust. Fonte: Adaptada de Massa e Avesani (2007b) .	42
Figura 4.1 - Arquitetura do trabalho proposto.....	50
Figura 5.1 - Quantidades exemplos de acordo com a nota .....	57
Figura 5.2 - Divisão dos dados.....	61
Figura 6.1 - MAEU dos algoritmos em diferentes visões (MT com profundidade 1)..	71
Figura 6.2 - MAEU dos algoritmos em diferentes visões (MT com profundidade 2)..	72
Figura 6.3 – Usuários cobertos pelos algoritmos (MT com profundidade 1) .....	72
Figura 6.4 – Usuários cobertos pelos algoritmos (MT com profundidade 2) .....	73

## LISTA DE TABELAS

Tabela 2.1 - Exemplo de matriz de usuários x itens.....	21
Tabela 3.1 - Matriz de confiança .....	38
Tabela 5.1 - Quantidades do conjunto de avaliações de usuários sobre itens.....	55
Tabela 5.2 - Quantidades do conjunto de avaliações de confiança .....	56
Tabela 5.3 - Estatísticas sobre dados de avaliações de usuários sobre itens .....	57
Tabela 6.1 - MAE (MT com profundidade 1) .....	67
Tabela 6.2 – MAE por usuário (MT com profundidade 1).....	67
Tabela 6.3 – Cobertura (MT com profundidade 1) .....	68
Tabela 6.4 – Cobertura por usuário (MT com profundidade 1).....	68
Tabela 6.5 – MAE (MT com profundidade 2).....	69
Tabela 6.6 - MAE por usuário (MT com Profundidade 2) .....	69
Tabela 6.7 – Cobertura (MT com profundidade 2) .....	70
Tabela 6.8 – Cobertura por usuário (MT com profundidade 2).....	70
Tabela 6.9 - Variáveis aleatórias consideradas no testes de hipótese para a tabela 6.6 .....	74
Tabela 6.10 - Desvios padrões dos MAEs e MAEUs dos algoritmos na visão “todos”. .....	75
Tabela 6.11 - Visões sobre os dados no trabalho de Massa e Avesani (2007a).....	75
Tabela 6.12 – Resultados de MT no trabalho de Massa e Avesani (2007a) e nesta dissertação segundo a MAE e visões de usuário.....	76
Tabela 6.13 - Resultados de MT no trabalho de Massa e Avesani (2007a) e nesta dissertação segundo a MAE e visão todos.....	77

## LISTA DAS ABREVIATURAS/ACRÔNIMOS

FBC	Filtragem Baseada em Conteúdo
FC	Filtragem Colaborativa
RBC	Recomendação Baseada em Confiança
CCP	Coeficiente de Correlação de Pearson
FCBM	Filtragem Colaborativa Baseada em Modelo
FM	Fatoração de Matriz
SVD	<i>Singular Value Decomposition</i>
MQA	Mínimos Quadrados Alternados
MT	MoleTrust
MAE	Média Absoluta dos Erros
MAEU	Média Absoluta dos Erros por Usuário
USS	User Specific Sparsity

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
1.1.	CONTEXTO DA DISSERTAÇÃO .....	13
1.2.	OBJETIVOS .....	14
1.3.	ESTRUTURA DO TRABALHO .....	15
<b>2</b>	<b>SISTEMAS DE RECOMENDAÇÃO</b>	<b>17</b>
2.1.	INTRODUÇÃO AOS SISTEMAS DE RECOMENDAÇÃO .....	17
2.2.	FILTRAGEM COLABORATIVA .....	20
2.3.	DESAFIOS EM FILTRAGEM COLABORATIVA .....	21
2.4.	FILTRAGEM COLABORATIVA BASEADA EM MEMÓRIA .....	25
2.5.	FILTRAGEM COLABORATIVA BASEADA EM MODELO .....	27
2.6.	FATORAÇÃO DE MATRIZ .....	28
2.7.	CONSIDERAÇÕES FINAIS .....	35
<b>3</b>	<b>SISTEMAS DE RECOMENDAÇÃO BASEADOS EM CONFIANÇA</b>	<b>36</b>
3.1.	CONFIANÇA EM RECOMENDAÇÃO .....	36
3.2.	CONFIANÇA EM FILTRAGEM COLABORATIVA .....	38
3.2.1.	<i>Medidas de Confiança Globais e Locais.</i>	39
3.2.2.	<i>MoleTrust</i>	40
3.2.3.	<i>Confiança em FC baseada em memória</i>	43
3.2.4.	<i>Confiança em FC baseada em modelos</i>	46
3.3.	CONSIDERAÇÕES FINAIS .....	47
<b>4</b>	<b>COMBINANDO FILTRAGEM COLABORATIVA E RECOMENDAÇÃO BASEADA EM CONFIANÇA ATRAVÉS DE MEDIDAS DE ESPARSIDADE</b>	<b>48</b>
4.1.	MOTIVAÇÃO .....	48
4.2.	ARQUITETURA PROPOSTA .....	50
4.3.	MEDIDAS DE ESPARSIDADE .....	52
<b>5</b>	<b>EXPERIMENTOS</b>	<b>55</b>
5.1.	BASE DE DADOS .....	55
5.2.	MÉTRICAS DE AVALIAÇÃO .....	57
5.3.	MÉTODOS COMPARADOS .....	59
5.4.	METODOLOGIA DOS EXPERIMENTOS .....	61
5.5.	FERRAMENTAS UTILIZADAS .....	64
<b>6</b>	<b>RESULTADOS</b>	<b>65</b>
6.1.	MOLETRUST COM PROFUNDIDADE 1 (APENAS AMIGOS) .....	65
6.2.	MOLETRUST COM PROFUNDIDADE 2 (AMIGOS E AMIGOS DOS AMIGOS) .....	68
6.3.	COMPARAÇÃO DOS MÉTODOS BASE COM A PROPOSTA HÍBRIDA DE MEDIDA 1 .....	70
6.4.	DISCUSSÃO DOS RESULTADOS .....	73
6.5.	COMPARAÇÃO COM OS RESULTADOS DO TRABALHO DE MASSA E AVESANI .....	75
<b>7</b>	<b>CONCLUSÕES</b>	<b>78</b>
7.1.	LIMITAÇÕES E TRABALHOS FUTUROS .....	79
	<b>REFERÊNCIAS</b>	<b>81</b>

## 1 INTRODUÇÃO

Atualmente, com a grande quantidade de produtos, serviços, livros, revistas, trabalhos científicos, músicas, amigos em redes sociais, e outras infinidades de itens que os usuários se deparam no uso da internet, sistemas de recomendação têm se tornado uma ferramenta importante para fornecer recomendações mais personalizadas a esses usuários.

Dentro deste contexto, a Filtragem Colaborativa (FC) se apresenta como a abordagem mais popular em sistemas de recomendação e grande parte dos sistemas comerciais online utilizam essa abordagem para recomendar produtos para seus clientes (SU & KHOSHGOFTAAR, 2009; LIU & LEE, 2010). FC trabalha sobre avaliações prévias do usuário sobre itens, para recomendar itens que o usuário ainda não avaliou. É assumido, em FC, que se dois usuários avaliaram um conjunto de itens de maneira similar, eles também possuem gostos similares.

Duas abordagens têm tido destaque em FC: (1) técnicas baseadas em vizinhança (ou baseadas em memória) (SARWAR et al., 2001; DESHPANDE & KARYPIS, 2004); e (2) técnicas baseadas em modelos utilizando fatoração de matriz (MNIH & SALAKHUTDINOV, 2007; KOREN et al., 2009) . A FC baseada em vizinhança realiza a recomendação após ter calculado um conjunto de  $k$  vizinhos mais similares ao usuário alvo. Uma medida é utilizada para calcular o grau de similaridade entre dois usuários a partir do conjunto de itens em comum que eles avaliaram. A Fatoração de Matrizes (FM), por sua vez, usa técnicas de decomposição de matrizes aplicadas a matriz de avaliações de usuários a itens. A decomposição é realizada para encontrar fatores latentes que são utilizados então para comparar usuários e itens. Técnicas de FM em geral superam as técnicas baseadas em memória em termos de desempenho preditivo (KOREN et al., 2009; SU & KHOSHGOFTAAR, 2009).

Apesar de sua grande popularidade, FC sofre de algumas fraquezas intrínsecas. Essas fraquezas estão relacionadas principalmente a problemas de esparsidade nas bases de dados de sistemas reais (i.e. Usuários e Itens com poucas avaliações). Sistemas de recomendação baseados em confiança (RBC) têm surgido mais recentemente, como uma das formas de lidar com o problema de

bases de dados esparsas, fazendo uso de informações potencialmente úteis contidas em redes sociais. Tipicamente a FC tradicional ignora as informações das redes sociais (e.g., relações de amizade ou confiança), pois ela se baseia apenas nas informações de avaliação dos usuários em itens.

Por exemplo, um usuário novo pode não ter avaliado ainda uma quantidade suficiente de itens de forma a receber boas recomendações por parte de um sistema tradicional de FC. No entanto, esse usuário pode estar conectado em uma rede a um bom número de amigos que já avaliaram itens. Desta forma, a recomendação para o usuário alvo poderia ser feita considerando os gostos e preferências dos seus amigos na rede social.

### **1.1. CONTEXTO DA DISSERTAÇÃO**

Para amenizar problemas com a esparsidade dos dados, alguns trabalhos vêm incorporando informações de confiança nos modelos de FC. Segundo Yang et al. (2014), a grande maioria desses trabalhos se dividem em incorporar confiança em dois casos: (1) dentro dos modelos de FC baseada em vizinhança ou (2) dentro do modelo de FM.

Por exemplo, Massa e Avesani (2007a) propuseram uma medida de confiança, chamada MoleTrust, que calcula o nível de confiança entre dois usuários na rede. Em seu trabalho, eles avaliaram a troca de uma medida de similaridade tradicional pela medida MoleTrust, em uma arquitetura de FC baseada em vizinhança. Os resultados mostraram que para usuários com poucas avaliações, os vizinhos formados pela medida de confiança forneciam melhores recomendações que os vizinhos formados pela medida de similaridade tradicional. Deste modo, a medida de confiança conseguiu amenizar a fraqueza dos usuários com poucas avaliações em FC.

Mais recentemente, alguns trabalhos (MA et al., 2009; JAMALI & ESTER; 2010; BAO & ZHANG; 2014) incorporam informações de confiança aos modelos de FM. Esses trabalhos melhoraram a qualidade da recomendação incorporando uma matriz de confiança usuários-usuários dentro do modelo tradicional de FM, que utiliza apenas a matriz de avaliação usuários-itens. Os resultados desses trabalhos

mostram que o uso de confiança nos modelos de FM consegue amenizar os problemas de usuários com poucas avaliações e diminuir o erro da recomendação.

Nesta dissertação, a técnica de FC (Fatoração de Matriz) e de sistemas de RBC (MoleTrust) serão combinadas através do uso de medidas de esparsidade de dados com o intuito de melhorar a predição das avaliações de usuários a itens. Nos experimentos realizados por Massa e Avesani (2007a), técnicas de RBC se destacam para usuários com poucas avaliações, enquanto que técnicas tradicionais de FM (KOREN et al., 2009) se mostraram superiores para usuários com muitas avaliações sobre itens, ou seja, as duas abordagens possuem bons resultados em porções diferentes dos dados, dependendo da presença de informação relevante. Motivado por essas observações, a arquitetura proposta neste trabalho pondera as predições realizadas por cada abordagem através de um peso definido a partir de medidas de esparsidade dos dados. O intuito de usar essas medidas é dar um peso maior para as predições fornecidas pelo método baseado em confiança quando o usuário tiver poucas avaliações (esparsidade alta da matriz de avaliações usuários-itens) ou muitos amigos (esparsidade baixa da matriz de confiança usuários-usuários). Para usuários com muitos amigos, esperamos que exista uma quantidade de informação útil maior para que RBC forneça boas recomendações. No entanto, esta última condição não foi analisada na literatura até o momento.

## 1.2. OBJETIVOS

Diante do que foi exposto na seção anterior, esse trabalho tem como objetivo principal:

- Investigar a combinação de FC com RBC utilizando medidas de esparsidade.

Para alcançar o objetivo principal deste trabalho, pretendemos também:

- Fazer uma revisão dos assuntos envolvidos na dissertação e descrever alguns trabalhos relacionados.
- Propor medidas de esparsidade para combinação de FC e RBC.

- Realizar experimentos que verifiquem a eficácia da proposta que combina as duas abordagens através das medidas de esparsidade em relação aos algoritmos isoladamente.
- Analisar os resultados diante de métricas que avaliem a qualidade da predição das avaliações.
- Investigar a eficácia da arquitetura proposta para aliviar os problemas relacionados a usuários com poucas avaliações.

### 1.3. ESTRUTURA DO TRABALHO

Esta dissertação está organizada da seguinte forma:

**Capítulo 2** – Este capítulo fará uma breve introdução sobre sistemas de recomendação e as principais abordagens utilizadas, dentre elas está a FC que é o foco deste capítulo. Serão apresentados conceitos fundamentais de FC, os principais desafios de FC e as duas principais abordagens: FC baseada em memória e FC baseada modelos. Será descrito com mais detalhes o modelo de FM em recomendação, ele será o algoritmo de FC utilizado neste trabalho.

**Capítulo 3** – Este capítulo abordará os sistemas de RBC. Serão apresentados alguns conceitos iniciais sobre este tema, uma explicação sobre medidas de confiança globais e locais e como confiança tem sido incorporada na abordagem de FC. Serão apresentados trabalhos que envolvem FC baseada em memória utilizando confiança e FC baseada em modelos utilizando confiança

**Capítulo 4** – Este capítulo apresentará a arquitetura proposta que combina FC e RBC através de medidas de esparsidade. Será apresentada a motivação para o desenvolvimento da arquitetura, os componentes da arquitetura e as medidas de esparsidade que foram propostas.

**Capítulo 5** – Este capítulo apresentará a metodologia utilizada para os experimentos e os algoritmos que serão comparados. Será descrita a base de dados, métricas de avaliações e como foi feita a divisão dos dados.

**Capítulo 6** – Este capítulo fará a apresentação e análise dos resultados dos experimentos auxiliados por tabelas e gráficos. Será avaliada se a proposta

apresentada neste trabalho trouxe alguma melhoria em relação aos algoritmos que foram comparados.

**Capítulo 7** – Este capítulo apresentará as conclusões da dissertação, os pontos mais relevantes e descrição de possíveis trabalhos futuros.

## 2 SISTEMAS DE RECOMENDAÇÃO

Este capítulo está organizado do seguinte modo. A seção 2.1 fará uma breve introdução sobre sistemas de recomendação e as principais abordagens utilizadas. Na seção 2.2, serão apresentados conceitos fundamentais de FC. Na seção 2.3, os principais desafios de FC e trabalhos relacionados a esses desafios. Nas seções 2.4 e 2.5, serão apresentadas as abordagens de FC baseada em memória e FC baseada modelos, respectivamente. Na seção 2.6, será descrito com mais detalhes o modelo de FM em recomendação.

### 2.1. INTRODUÇÃO AOS SISTEMAS DE RECOMENDAÇÃO

Atualmente, com a grande quantidade de produtos, serviços, livros, revistas, trabalhos científicos, músicas, amigos em redes sociais, e outras infinidades de itens que os usuários se deparam no uso da internet, torna-se de grande importância um sistema que ajude o usuário a escolher produtos ou informações que mais lhe agradam. Seja para entretenimento ou trabalho, os sistemas de recomendação surgiram como uma alternativa útil para ajudar ao usuário a encontrar itens que ele gostaria de obter, diminuindo a sobrecarga de informações.

A grande maioria das lojas virtuais já utiliza algum mecanismo de recomendação, possibilitando assim um aumento das vendas e aumento da satisfação do usuário no uso do sistema. Essas recomendações podem variar desde recomendações não personalizadas, aquelas que são apresentadas para todos os usuários que entram na loja virtual (e.g., Lista dos mais vendidos, lista dos mais procurados), até recomendações mais personalizadas, coletando do indivíduo suas preferências. Por exemplo, um sistema de recomendação pode utilizar o caminho percorrido pelo usuário no site para coletar informações sobre o interesse deste. Geralmente as empresas possuem em seus websites a abordagem personalizada e não personalizada de recomendação. Segundo Kantor (2011), a não personalizada é geralmente mais simples de ser implementada e pode ser muito útil em certas situações (e.g., quando não há um registro único para cada usuário). Neste trabalho estamos interessados apenas em sistemas de recomendação que fornecem recomendações personalizadas, em geral esse tipo de sistemas utiliza ferramentas de inteligência computacional.

Basicamente, podemos dividir os sistemas de recomendação em três abordagens: a Filtragem Baseada em Conteúdo (*Content-Based Approach*), Filtragem Colaborativa (*Collaborative Filtering Approach*) e Abordagem Híbrida (*Hybrid Approach*). (ADOMAVICIUS & TUZHILIN, 2005; KANTOR et al., 2011; FORSATI et al., 2014)

Na Filtragem Baseada em Conteúdo (FBC), o sistema cria um perfil de cada usuário e cada produto para caracterizar sua natureza. Por exemplo, um livro pode ser caracterizado pelo assunto, autor, editora, ano de publicação e outros atributos. Nesse caso, o sistema requer um conjunto de informações de conteúdo que podem ser difícil de extrair, dado a grande quantidade de diferentes itens em um sistema de recomendação. A abordagem baseada em conteúdo tem suas raízes nas áreas de Recuperação de Informação e Filtragem de Informação (ADOMAVICIUS e TUZHILIN, 2005). Muitos dos sistemas que utilizam essa abordagem trabalham com itens baseados em textos e conteúdos que possam ser descritos através de palavras-chaves.

O sistema de FBC analisa a descrição e o conjunto de características dos itens previamente avaliados pelo usuário. Em seguida, o sistema cria um perfil do usuário baseado nas características dos itens que o usuário avaliou (KANTOR et al., 2011). Algumas das vantagens na utilização dessa abordagem estão na independência entre os usuários, transparência na recomendação e recomendação para novos itens na base (RICCI et al., 2011).

- **Independência** - A independência entre os usuários ocorre, pois na construção do perfil dos usuários apenas a características dos itens avaliados pelo próprio usuário são levadas em consideração.
- **Transparência** – Pelo fato da descrição dos itens e o perfil dos usuários ser formado por um conjunto explícito de características, torna-se possível conhecer a causa de uma lista de itens ter sido apresentada para um usuário na recomendação.
- **Novos itens** – Novos itens podem ser avaliados assim que entram no sistema e após terem suas características descritivas fornecidas.

Por outro lado, a recomendação baseada em conteúdo tem algumas limitações (RICCI et al., 2011).

- **Limitada análise de conteúdo** – O número de características coletadas de um objeto/item tem um limite natural, seja de forma automática ou manual. Muitos sistemas estão associados a domínio específico (e.g., filmes, livros, carros). É necessário algum conhecimento do domínio para a representação dos itens (e.g., um filme pode ser representado pelos atores, diretor, categoria, ano). No entanto, nenhum sistema poderá diferenciar dois itens caso eles não tenham informações suficientes para diferenciá-los.
- **Superespecialização (*Over-Specialization*)** – A FBC não é boa estratégia para recomendar itens não esperados. As recomendações tendem a ser lógicas e associadas com a representação do perfil de cada usuário. Por exemplo, caso um usuário tenha avaliado apenas livros de um autor o sistema só será capaz de recomendar livros desse autor.
- **Novos usuários** – Quando o usuário avalia poucos itens, torna-se difícil para o sistema de recomendação conseguir gerar um perfil rico desse usuário para poder fornecer boas recomendações. Esse problema pode ser amenizado quando o sistema de FCB utiliza informações demográficas (e.g., gênero, idade, endereço, escolaridade) ou questionários apresentados ao usuário (KOREN et al., 2009).

A segunda grande abordagem em sistema de recomendação é chamada de Filtragem Colaborativa (FC). A FC surgiu após a FBC e atualmente tem atraído mais atenção nas pesquisas de sistemas de recomendação. No geral, FC supera os resultados obtidos com a FBC em relação à acurácia das recomendações (KOREN et al., 2009). Em FC, é bastante comum a recomendação de itens não esperados (grau de inovação), o que não ocorre com a FBC que fornece recomendação associada com a representação do perfil do usuário. Ao contrário da FBC, a FC não precisa guardar informações descritivas dos itens.

Em FC, a recomendação é realizada utilizando apenas as informações de avaliações dos usuários em itens e na similaridade dos gostos entre usuários baseada nessas avaliações. Por exemplo, o usuário A avaliou os mesmos filmes de

um usuário *B* fornecendo notas muito semelhantes. Possivelmente o usuário *B* pode ter assistido a um filme que o usuário *A* ainda não assistiu e esse filme tem grandes chances de ser recomendado através de FC. O grande problema da FC é com usuários que avaliaram poucos itens e itens pouco avaliados (MELVILLE et al., 2002; KOREN et al., 2009). Neste aspecto, a FBC é superior (KOREN et al., 2009). Sistemas de FBC podem utilizar outros meio além das avaliações de usuários a itens para construir o perfil dos usuários (e.g., dados fornecidos pelos usuários) e assim amenizar esse problema.

Com o objetivo de superar as limitações de ambas as abordagens surgiram algoritmos híbridos unindo FC e FBC. Um exemplo dessa abordagem é o trabalho de Melville et al. (2002), no qual foi utilizado conteúdo externo de informações textuais para superar problemas da FC com os novos usuários e novos itens.

Nesta dissertação, não trabalharemos com técnicas de filtragem baseada em conteúdo. O foco desta dissertação será em FC e como o uso de informações de confiança entre usuários pode ajudar a superar os problemas dessa abordagem.

## 2.2. FILTRAGEM COLABORATIVA

A Filtragem colaborativa é uma abordagem que assume que se dois usuários *A* e *B* avaliaram um conjunto de *N* itens de maneira similar, eles possuem também gostos similares. A FC utiliza as informações explícitas (e.g., avaliações de usuários sobre produtos ou informações) ou avaliações implícitas (e.g., visita do usuário ao produto, compra do produto, assistir um filme, ouvir uma música) para poder realizar a recomendação (SU & KHOSHGOFTAAR, 2009). Em FC, basta apenas guardar um identificador para cada usuário e item dos dados, e uma lista de itens que o usuário avaliou. Diferentemente da FBC, a FC é livre de domínio e não necessita guardar informações sobre as características dos itens (KOREN et al., 2009). Isto significa que toda a informação utilizada em FC deriva exclusivamente do julgamento humano, permitindo assim um conjunto de dados repleto de conhecimento tácito difícil de representar (SAMPAIO, 2006).

Digamos que em um sistema de recomendação que utiliza FC, os usuários podem avaliar um conjunto de itens de qualquer natureza (e.g., produtos eletrônicos, filmes ou serviços de hotéis) em uma escala que vai de 1 a 5, com 1 indicando que o

usuário não gostou do item ou gostou muito pouco e 5 quando o usuário gostou muito do item. Em um cenário típico, os dados deste sistema são representados através de uma matriz, onde usuários estão dispostos em linhas e itens dispostos em colunas. No caso de um sistema de recomendação de filmes, uma matriz de usuários x itens utilizando a escala mencionada seria semelhante a da tabela 3.1.

Tabela 2.1 - Exemplo de matriz de usuários x itens

	Os Vingadores	Shrek	O Poderoso Chefão
Marcos	4	2	
Silvia		3	1
Pedro	?	4	2

Na tabela 2.1, Pedro seria o usuário ativo para o qual nós gostaríamos de predizer a nota para o filme “Os Vingadores”. Possivelmente, podem existir alguns ou vários valores faltantes (matriz muito esparsa) em uma matriz deste tipo.

### 2.3. DESAFIOS EM FILTRAGEM COLABORATIVA

Naturalmente, a FC também apresenta alguns problemas e desafios. Muitos desses problemas têm sido estudados e têm sido fonte de diversos trabalhos em FC. Grandes companhias como Amazon<sup>1</sup> e eBay<sup>2</sup> utilizam FC em suas lojas virtuais (LINDEN et al., 2003). Essas companhias operam com um conjunto muito grande de produtos e usuários. Uma melhoria na acurácia da recomendação traria um grande retorno para essas companhias.

Alguns dos principais desafios em filtragem são os seguintes:

---

<sup>1</sup> [www.amazon.com](http://www.amazon.com)

<sup>2</sup> [www.ebay.com](http://www.ebay.com)

- **Esparsidade dos dados**

A esparsidade dos dados pode ser observada pela grande quantidade de avaliações faltantes na matriz de usuários x itens (Ver tabela 1). Esse problema ocorre com frequência nos sistemas reais de recomendação, pois existe uma enorme quantidade de itens (Sobrecarga de Informações) para a capacidade dos usuários em avaliá-los, podendo acontecer de itens não terem nenhuma avaliação.

O problema da esparsidade dos dados pode causar uma baixa da cobertura da recomendação (SU & KHOSHGOFTAAR, 2009). Itens que não tiveram nenhuma avaliação não conseguem ser preditos utilizando apenas FC. Cobertura pode ser definida como o percentual de itens relevantes que o algoritmo consegue recomendar.

Para lidar com problemas de grande esparsidade dos dados os autores (MNIH & SALAKHUTDINOV, 2007; KOREN et al., 2009) utilizaram a técnica de Decomposição em Valores Singulares (SVD, do inglês: Singular Value Decomposition) para reduzir a dimensionalidade da matriz de usuários-itens (matriz de avaliações).

A esparsidade da base está diretamente relacionada à problemas como usuários novos (usuários com poucas avaliações) e itens novos, embora seja possível encontrar usuários novos em bases densas e usuários com muitas avaliações em bases esparsas.

- **Problema do Usuário Novo ou Usuário com Poucas Avaliações (Cold Start User)**

O problema de usuários novatos ocorre com usuários que acabaram de entrar no sistema ou usuários que não possuem o hábito de realizar avaliações. Isto representa um grande problema para os sistemas de recomendação, pois eles não conseguem fornecer recomendações personalizadas para estes usuários utilizando FC (Bobadilla et al., 2013). Desde as primeiras avaliações de um usuário, o sistema começa a ser habilitado a recomendar, embora seja

necessário um número suficiente de avaliações para que a recomendação tenha uma confiabilidade razoável.

- **Problema do Item Novo ou Item com Poucas Avaliações**

O mesmo problema que acontece com usuários também acontece com os itens novos que entram no sistema. Itens novos que não foram avaliados por nenhum usuário não podem ser recomendados ficando de fora da recomendação por FC até que algum usuário os avalie. Esse problema também é denominado de Recomendação de Cauda Longa (Long Tail Recommendation) por alguns autores como Shi et al. (2014) e Anderson (2006). A cauda longa é um fenômeno que ocorre no gráfico mostrado na figura 2.1. Nos sistemas reais é comum que se tenha alguns itens muito populares com mais de 1000 avaliações enquanto a grande maioria dos itens tem poucas avaliações. Em um sistema que recomenda filmes, por exemplo, a recomendação de um filme menos popular, que seria mais difícil de o usuário descobrir informações, resulta em uma grande satisfação dos usuários (SHI et al., 2014). O problema do item novo tem menos impacto sobre os sistemas de recomendação que o problema do usuário novo, pois esses sistemas podem criar políticas de propaganda para esses itens (Bobadilla et al., 2013).

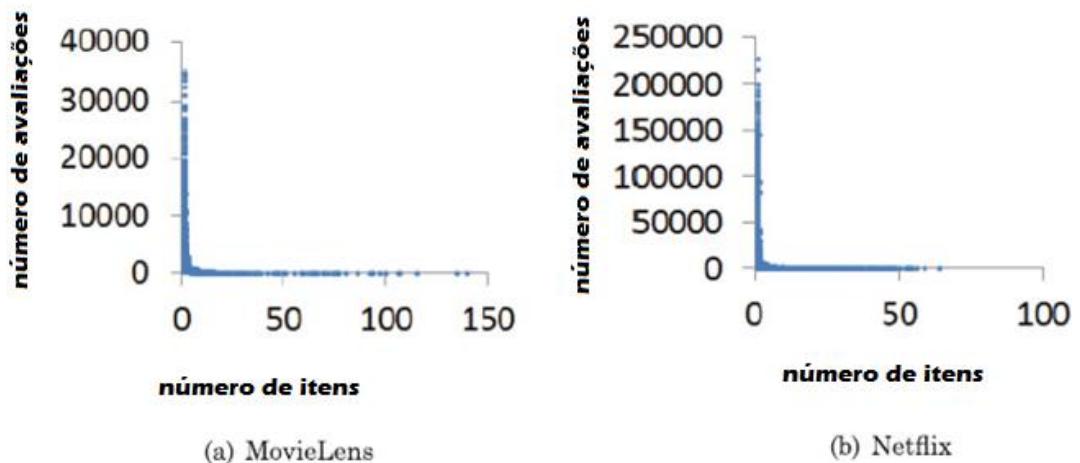


Figura 2.1 - Distribuição de cauda longa no conjunto de dados MovieLens e Netflix. O ponto (x,y) significa que x itens tiveram y avaliações. Fonte: Shi et al., 2014.

- **Usuário Ovelha Negra.**

Ocorre geralmente com aqueles usuários que têm avaliado um conjunto de itens não usuais, ou seja, poucos ou nenhum usuário têm gostos similares ao dele. O sistema de recomendação não consegue descobrir bons vizinhos para esse tipo de usuário, apresentando geralmente uma recomendação com pouca confiabilidade. Existem também os casos descritos como usuário ovelhas cinza (SU & KHOSHGOFTAAR, 2009). Neste caso, os usuários têm avaliado itens que já foram avaliados por outros usuários, mas as avaliações deles não concordam com as avaliações de outros usuários.

A estratégia mais comum para amenizar os problemas de usuários e itens novos é buscar informações adicionais ao conjunto de avaliações de usuários sobre itens. Geralmente são utilizados algoritmos híbridos que combinam FC com outros tipos de recomendação (e.g., filtragem baseada em conteúdo, recomendação baseada em dados demográficos e recomendação baseada em redes sociais) (SU & KHOSHGOFTAAR, 2009). Um dos primeiros trabalhos (BALABANOVIĆ & SHOHAM, 1997), desenvolvido na Universidade de Stanford em 1997, aplicou FC em conjunto com a recomendação baseada em conteúdo em um sistema de recomendação de páginas web chamado FAB. A proposta híbrida permitiu superar algumas fraquezas das duas abordagens, inclusive recomendação para usuários novos e itens novos. Já Loh et al. (2009) utilizaram uma representação do perfil do usuário obtida a partir das informações das suas publicações científicas para aliviar o problema com usuários novatos. Enquanto Martinez et al. (2009) apresentaram um sistema híbrido que combinava FC com técnicas de aprendizagem baseado em conhecimento.

Do mesmo modo, a recomendação para usuários ovelhas negra ou ovelhas cinza pode ser realizada através da combinação de FC com a FBC (Bobadilla et al., 2013). Massa e Avesani (2007a) mostraram que a utilização de informação de confiança entre usuários pode melhorar a qualidade da recomendação para os casos de usuário novo e usuário ovelha negra.

## 2.4. FILTRAGEM COLABORATIVA BASEADA EM MEMÓRIA

As duas principais abordagens em FC são: baseada em memória (memory-based) e a baseada em modelo (model-based). Nesta seção, é descrita a primeira delas.

As abordagens baseadas em memória são as mais populares e mais utilizadas pelos sistemas comerciais que adotam FC (MA et al., 2008). Elas correspondem às primeiras gerações de técnicas em FC. Seu funcionamento é baseado no cálculo de similaridades entre usuários (baseado em usuário) ou itens (baseado em itens). Na abordagem baseada em usuário (BREESE et al., 1998; MASSA & AVESANI, 2007a), o usuário ativo tem sua nota predita para um item (com nota desconhecida) baseado nas notas fornecidas por um conjunto de  $N$  usuários similares a ele e que já tenham avaliado o item. Enquanto que na abordagem baseada em itens (SARWAR et al., 2001; DESHPANDE & KARYPIS, 2004), é avaliado quem são os itens mais similares, dentre os previamente avaliados pelo usuário ativo, ao item cuja nota se deseja predizer, e a partir daí é calculada a nota predita para aquele item.

As duas abordagens de FC baseada em memória realizam a predição de uma avaliação através de 2 passos bastante semelhantes. A seguir será descrita a abordagem por usuário, contudo, para maiores detalhes da abordagem baseada em itens, o trabalho de Sarwar et al. (2001) poderá ser consultado.

Na abordagem baseada em usuário (Figura 2.2), o primeiro passo realiza o cálculo da similaridade entre o usuário ativo e os outros usuários da base, resultando em um peso que reflete grau de similaridade entre cada usuário da base e o usuário ativo. Em geral, são procurados os  $N$  vizinhos mais similares. O cálculo de similaridade entre dois usuários  $x$  e  $y$  é feito baseado nos itens em comuns que eles avaliaram.

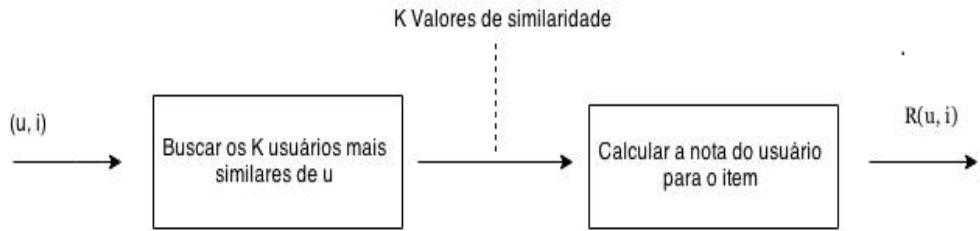


Figura 2.2 - FC baseada em memória

Existem diferentes medidas para calcular a similaridade entre usuários. A principal e mais utilizadas delas é o coeficiente de correlação de Pearson (CCP), que assume que dois usuários avaliaram pelos menos alguns itens em comum para poder calcular a similaridade. Quando a base de dados cresce e aumenta o número de usuários e itens, o cálculo da similaridade entre o usuário ativo e cada usuário da base pode ser tornar custoso, sendo a escalabilidade um dos problemas relacionados a esse tipo de abordagem (SU & KHOSHGOFTAAR, 2009).

Para realizar a busca pelos  $N$  vizinhos mais similares, diversos trabalhos vêm utilizando CCP (RESNICK et al., 1994; BREESE et al., 1998; MASSA & AVESANI, 2007a; SARWAR et al., 2001). O cálculo do CCP é dado pela seguinte fórmula, onde a saída é um valor de similaridade entre o usuário  $a$  e o usuário  $u$  ( $S_{a,u}$ ).

$$S_{a,u} = \frac{\sum_{i=1}^m (r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^m (r_{a,i} - \bar{r}_a)^2 \sum_{i=1}^m (r_{u,i} - \bar{r}_u)^2}} \quad (2.1)$$

Na equação acima, o  $r_{a,i}$  e  $r_{u,i}$  são as avaliações do usuário  $a$  e usuário  $u$  para o item  $i$ , respectivamente;  $\bar{r}_a$  e  $\bar{r}_u$  são as médias das avaliações dos usuários  $a$  e  $u$ , respectivamente. Nesta equação  $m$  representa o número de itens que foram avaliados por  $a$  e também por  $u$ , ou seja, o cálculo apenas leva em consideração os itens avaliados pelos dois usuários.

Em alguns casos, existem poucos itens avaliados em comum entre dois usuários. Consequentemente, o valor de similaridade fornecido para esses casos pode não ser muito confiável. Por exemplo, dois usuários em sistema de recomendação de filmes que possuem poucos filmes avaliados em comum podem ter valor de similaridade alta pela equação 2.1. Herlock et al. (1999) propôs a

redução da medida de similaridade quando a interseção do conjunto de itens avaliados por dois usuários tiver tamanho menor que 50.

Após o cálculo de similaridade entre o usuário ativo e todos os outros usuários base, é criada uma lista dos tops  $k$  usuários mais similares ao usuário ativo. No segundo passo da FC, é realizada a predição da nota para um item que o usuário ativo ainda não avaliou. A equação mais utilizada para realizar esse cálculo, que foi descrita por Resnick et al. (1994), é dada por:

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^n [(r_{u,i} - \bar{r}_u) w_{a,u}]}{\sum_{u=1}^n w_{a,u}} \quad (2.2)$$

Onde  $p_{a,i}$  é a nota de usuário  $a$  para o item  $i$ ,  $\bar{r}_a$  é a média das avaliações do usuário  $a$  (usuário ativo),  $r_{u,i}$  e  $\bar{r}_u$  são a avaliação de um usuário  $u$  para o item  $i$  e a média das avaliações do usuário  $u$ , respectivamente. O número de usuários similares é dado por  $n$ . Finalmente, o  $w_{a,u}$  é o peso calculado através de alguma medida de similaridade.

## 2.5. FILTRAGEM COLABORATIVA BASEADA EM MODELO

A FCBM consiste em gerar modelos (e.g., algoritmos de aprendizagem de máquina e mineração de dados) a partir dos dados, guardando os parâmetros aprendidos durante o treinamento. Os modelos gerados a partir da interação dos usuários com itens guardam representações de características como, preferências de usuários e categorias dos itens. As abordagens baseadas em modelos são muito rápidas após os parâmetros serem aprendidos. Pode-se dizer que enquanto o gargalo na FCBM é a fase de treinamento, o gargalo da recomendação baseada em memória seria na fase de predição (JAMALI & ESTER, 2010).

Geralmente, algoritmos de classificação são utilizados como modelos de FC quando as avaliações dos usuários são categóricas e modelos de regressão são utilizados quando avaliações são numéricas. Alguns algoritmos já investigados para criação dos modelos são: Redes Neurais (ROH et al., 2003), Classificador Bayesiano (PARK et al., 2007), Sistemas Fuzzy (YAGER, 2003), Algoritmo Genéticos (GAO & LI, 2008), Fatoração de Matriz (JAMALI & ESTER, 2010).

Essa abordagem é geralmente mais difícil de ser implementada que a abordagem baseada em memória. Por outro lado, de modo geral, a FCBM possui melhor desempenho de predição e melhor cobertura (GUO et al., 2014). Pode ocorrer uma dificuldade de interpretação dos resultados da recomendação, pois alguns algoritmos de aprendizagem de máquina funcionam como caixas-pretas.

Apesar de tanto a abordagem baseada em memória quanto a abordagem baseada em modelos sofrer com todos os problemas descritos na seção 2.3, a abordagem baseada em memória sofre ainda mais com os problemas de usuários novos ou com poucas avaliações. Isso é causado pela dificuldade de encontrar os usuários mais similares em bases muito esparsas através das medidas de similaridade. A escalabilidade dos sistemas de recomendação é outro problema que afeta mais aqueles que utilizam recomendação baseada em memória. Quando o número de usuários e itens de um sistema cresce muito, o custo computacional de calcular os  $k$  vizinhos mais similares torna-se bastante oneroso. Em resumo, escalabilidade e esparsidade dos dados são os dois principais problemas existentes em recomendação baseada em memória, e que a recomendação baseada em modelos tenta superar (FORSATI et al., 2014).

Devido ao sucesso alcançado em sistemas de recomendação com grande quantidade de dados, a FM tem se tornado um dos mais populares métodos dentro da abordagem baseada em modelos (GUO et al., 2014). Na próxima seção, é descrito um pouco a sobre Fatoração de Matrizes, pois ela será umas das técnicas utilizadas no trabalho desenvolvido nesta dissertação.

## **2.6. FATORAÇÃO DE MATRIZ**

A FM é uma técnica baseada em modelos de fatores latentes. Um modelo de fatores latentes tenta caracterizar os usuários e itens a partir de um conjunto de  $k$  fatores. Em um sistema de recomendação de filmes, por exemplo, esses fatores podem representar uma tendência no estilo do filme. Um filme pode durante toda sua duração pertencer a uma serie de categorias, e muitas vezes podem ser difíceis categoriza-lo em apenas uma, podendo o filme pertencer a diversas categorias com diferentes graus de pertinência. Fatores do modelo de FM podem representar implicitamente categorias de filmes ou estilo de gostos de pessoas, que estão

escondidos nos dados e que são difíceis de interpretar. Uma alta correspondência entre os fatores dos itens e dos usuários levam a uma recomendação.

Uma definição simples pode ser encontrada no site do Mahout<sup>3</sup> sobre a FM. “Na disciplina de álgebra linear, a decomposição ou fatoração de matriz é uma técnica de redução de dimensionalidade que fatora a matriz em um produto de matrizes, usualmente duas”. A figura 3.3 abaixo representa a ideia da decomposição de matrizes.

$$\begin{array}{c}
 \begin{array}{c} R \\ \hline \end{array} \\
 \begin{array}{|c|c|c|c|c|c|c|c|} \hline & i_1 & i_2 & i_3 & i_4 & i_5 & i_6 & i_7 & i_8 \\ \hline u_1 & 5 & 2 & & 3 & & 4 & & \\ \hline u_2 & 4 & 3 & & & 5 & & & \\ \hline u_3 & 4 & & 2 & & & & 2 & 4 \\ \hline u_4 & & & & & & & & \\ \hline u_5 & 5 & 1 & 2 & & 4 & 3 & & \\ \hline u_6 & 4 & 3 & & 2 & 4 & & 3 & 5 \\ \hline \end{array} \\
 \approx \\
 \begin{array}{c} P \\ \hline \end{array} \\
 \begin{bmatrix} 1.55 & 1.22 & 0.37 & 0.81 & 0.62 & -0.01 \\ 0.36 & 0.91 & 1.21 & 0.39 & 1.10 & 0.25 \\ 0.59 & 0.20 & 0.14 & 0.83 & 0.27 & 1.51 \\ 0.39 & 1.33 & -0.43 & 0.70 & -0.90 & 0.68 \\ 1.05 & 0.11 & 0.17 & 1.18 & 1.81 & 0.40 \end{bmatrix} \\
 \begin{array}{c} Q \\ \hline \end{array} \\
 \begin{bmatrix} 1.00 & -0.05 & -0.24 & 0.26 & 1.28 & 0.54 & -0.31 & 0.52 \\ 0.19 & -0.86 & -0.72 & 0.05 & 0.68 & 0.02 & -0.61 & 0.70 \\ 0.49 & 0.09 & -0.05 & -0.62 & 0.12 & 0.08 & 0.02 & 1.60 \\ -0.40 & 0.70 & 0.27 & -0.27 & 0.99 & 0.44 & 0.39 & 0.74 \\ 1.49 & -1.00 & 0.06 & 0.05 & 0.23 & 0.01 & -0.36 & 0.80 \end{bmatrix} \\
 \begin{array}{c} PQ^T = \\ \hline \end{array} \\
 \begin{array}{|c|c|c|c|c|c|c|c|} \hline & i_1 & i_2 & i_3 & i_4 & i_5 & i_6 & i_7 & i_8 \\ \hline u_1 & 5 & 2 & 2.5 & 3 & 4.8 & 4 & 2.2 & 4.8 \\ \hline u_2 & 4 & 3 & 2.4 & 2.9 & 5 & 4.1 & 2.6 & 4.7 \\ \hline u_3 & 4 & 1.7 & 2 & 3.2 & 3.9 & 3.0 & 2 & 4 \\ \hline u_4 & 4.8 & 2.1 & 2.7 & 2.6 & 4.7 & 3.8 & 2.4 & 4.9 \\ \hline u_5 & 5 & 1 & 2 & 3.4 & 4 & 3 & 1.5 & 4.6 \\ \hline u_6 & 4 & 3 & 2.9 & 2 & 4 & 3.4 & 3 & 5 \\ \hline \end{array}
 \end{array}$$

Figura 2.3 - Fatoração da matriz R. Fonte: Adaptada de Ma et al. (2008).

Tendo por base a figura 2.3 e utilizando um pouco de notação matemática, definiremos alguns conceitos sobre FM. Chamaremos de  $R$  a matriz de usuários e itens de dimensões  $m \times n$ , onde  $m$  é o número de usuários e  $n$  o número de itens. Chamaremos de  $P$  e  $Q$  duas matrizes de dimensões  $m \times k$  e  $n \times k$  respectivamente, onde  $k$  é o número de fatores latentes, e o produto de  $P$  com a transposta de  $Q$  será uma matriz  $\hat{R}$  que aproxima  $R$ .

$$R \approx P \times Q^T = \hat{R} \quad (3.3)$$

<sup>3</sup> <https://mahout.apache.org/users/recommender/matrix-factorization.html>

Deste modo, cada linha de  $P$  representa a força da associação de cada um dos  $m$  usuários com cada um dos  $k$  fatores, e cada linha de  $Q$  representa a força da associação de cada item com cada um dos  $k$  fatores. A ideia apresentada está muito relacionada com a técnica de SVD que identifica fatores latentes semânticos em Recuperação de Informação (KOREN et al., 2009). SVD pode ser considerada a abordagem mais básica de FM (YANG et al, 2014). Alguns modelos (FUNK, 2006; MNIH & SALAKHUTDINOV, 2007; KOREN et al., 2009;) vêm alterando a ideia original de SVD e realizando a decomposição através de duas matrizes apenas (ver equação 3.3), sendo chamados apenas por modelos de fatoração de matriz.

Após a fatoração da matriz  $R$  em duas matrizes, podemos calcular a nota predita de um usuário  $u$  para um item  $i$ , a partir de equação 3.4. Ela mostra o produto cartesiano de um vetor  $p_u$  com um vetor  $q_i^T$ . Sendo  $p_u$  um vetor da matriz  $P$  e dimensão  $k$ , que representa quanto o usuário  $u$  está relacionado a cada um dos  $k$  fatores. Da mesma forma, o vetor  $q_i^T$  mostra a correlação entre um item  $i$  e um conjunto de  $k$  fatores. Finalmente,  $\hat{r}_{ui}$  representa o valor predito da nota de um usuário  $u$  para um item  $i$ .

$$\hat{r}_{ui} = q_i^T p_u \quad (3.4)$$

Em um sistema de recomendação de filmes, o filme Ghost pode ser representado por um vetor como  $(-1; 2; 2.5; 4)$ . Supondo que os 4 fatores do vetor estejam implicitamente relacionados com as seguintes categorias de gênero dos filmes: terror, ficção, comédia e romance. Maria é uma usuária do sistema que tem aversão por filmes de terror e cujo vetor de características é dado por  $(-1; 1.2; 4; 3)$ . Pela equação 3.4, pode-se calcular a nota predita da usuária para o filme. A nota é calculada somando o produto dos fatores de Maria com o do filme Ghost.

$$\hat{r}_{ui} = -1 \times (-1) + 2 \times 1.2 + 2.5 \times 4 + 4 \times 3 \quad (3.5)$$

Note que o fator  $-1$  do filme com o fator  $-1$  da usuária são multiplicados, e a troca de sinal reforça que o filme se adéqua ao perfil do usuário.

Uma grande dificuldade para se trabalhar com FM é o grande número de valores faltantes. A técnica de SVD convencional não está preparada com matrizes incompletas. Funk (2006) propôs uma evolução no SVD que trabalha ignorando os

valores faltantes, criando o modelo diretamente com os valores preenchidos. Como a quantidade de dados preenchidos na matriz  $R$  muitas vezes não passa de 10%, existe uma preocupação com o problema do Superajuste que pode ser evitado através da inclusão de um fator regularização no modelo. Evitar o Superajuste é essencial para que o modelo possa recomendar bem para dados que não estejam no conjunto original de dados.

Para encontrar as matrizes  $P$  e  $Q$  fatoradas, cujo produto dado pela equação 3.3 irá resultar em  $R$  aproximada ( $\hat{R}$ ), nós devemos minimizar a “equação de erro” ou “função de custo”, representado neste trabalho por  $J$ . A equação de erro é dada pelo somatório do quadrado diferença entre cada valor preenchido na matriz  $R$  ( $r_{ui}$ ) e o seu valor predito ( $\hat{r}_{ui}$ ).

$$J = \sum_{(u,i) \in R} (r_{ui} - \sum_k p_{uk} q_{ik}^t)^2 = \sum_{(u,i) \in R} e^2 \quad (3.6)$$

Sendo  $e$  a diferença de cada valor preenchido da matriz  $R$  para seu valor predito, e seu quadrado dado pela equação.

$$e^2 = (r_{ui} - \sum_k p_{uk} q_{ik}^t)^2 \quad (3.7)$$

Onde  $r_{ui}$  é o valor preenchido na matriz  $R$ , e  $\sum_k p_{uk} q_{ik}^t$  é o valor predito ( $\hat{r}_{ui}$ ).  $u$  e  $i$  são a linha que representa o usuário e a coluna que representa o item da matriz  $R$ .

Existem duas abordagens principais para minimizar a equação de erro (KOREN et al., 2009): Gradiente Descendente e Mínimos Quadrados Alternados (MQA). A ideia principal da abordagem de MQA é minimizar a complexidade da equação de erro, fixando ora a matriz  $P$  e ora a matriz  $Q$  para ajuste dos parâmetros do modelo. O método MQA pode ser utilizado quando se quer trabalhar com computação paralela tornando os sistemas mais escaláveis, como descrito no trabalho de Zhou et al. (2008). Explicaremos mais detalhadamente como funciona o método do Gradiente Descente, pois foi utilizado durante o trabalho dessa dissertação.

### 2.6.1. MODELO DE TREINAMENTO DO GRADIENTE DESCENDENTE

O método do Gradiente Descendente para treinamento da técnica de FM foi popularizado por Funk. Em Funk (2006), ele descreveu com detalhes como conseguiu ficar na terceira posição do Premio Netflix<sup>4</sup>.

O método do gradiente funciona de maneira iterativa e apenas utiliza os valores preenchidos da matriz  $R$  (matriz esparsa) para treinar duas matrizes  $P$  e  $Q$  (matrizes densas) que serão utilizadas para predizer os valores faltantes da matriz  $R$  (ver figura 3.3). As matrizes  $P$  e  $Q$  geralmente iniciam o treinamento com valores aleatórios ou fixos muito baixos e são ajustados a cada iteração do algoritmo até que o critério de parada seja atingido (e.g., número de iterações, ponto de mínimo).

Durante o treinamento das matrizes  $P$  e  $Q$ , para cada valor preenchido o algoritmo busca minimizar o quadrado do erro  $e$  da equação 3.7, atualizando os valores dos vetores  $p_{uk}$  e  $q_{ki}$  na direção oposta do gradiente. Para isso, será necessário derivar a equação 3.7 com relação aos dois vetores ( $p_u$  e  $q_i$ ), separadamente.

$$\frac{\partial}{\partial p_{uk}} e_{ui}^2 = -2(r_{ui} - \hat{r}_{ui})(q_{ki}) = -2e_{ui}q_{ki} \quad (3.8)$$

$$\frac{\partial}{\partial q_{ik}} e_{ui}^2 = -2(r_{ui} - \hat{r}_{ui})(p_{uk}) = -2e_{ui}p_{uk} \quad (3.9)$$

Obtido o gradiente, podemos formular a equação de atualização dos vetores da seguinte maneira.

$$p'_{uk} = p_{uk} + \alpha \frac{\partial}{\partial p_{uk}} e_{ui}^2 = p_{uk} + 2\alpha e_{ui}q_{ki} \quad (3.10)$$

$$q'_{ki} = q_{ki} + \alpha \frac{\partial}{\partial q_{ik}} e_{ui}^2 = q_{ki} + 2\alpha e_{ui}p_{uk} \quad (3.11)$$

Note que foi incluído um elemento  $\alpha$ , cujo valor determina a taxa de aprendizagem, ou seja, a velocidade de aproximação para o mínimo. A atualização

---

<sup>4</sup> Premio Netflix – É uma competição patrocinada pelo sistema de aluguel e recomendação de filmes online Netflix. O objetivo é diminuir o erro de predição do sistema em um percentual de 10% e o ganhador leva um premio em dinheiro.

dos vetores ocorre sobre cada valor preenchido da matriz  $R$ , como dito anteriormente. Após uma iteração do método de treinamento são somados os quadrados dos erros obtidos de cada valor preenchido em relação ao valor predito ( $e^2$ ) para saber se o valor de  $J$  está diminuindo.

### 2.6.2. REGULARIZAÇÃO

O treinamento do algoritmo utilizando a equação de erro dada pela equação 3.6 pode conduzir ao problema do Superajuste. Conforme pode ser observado na figura 2.4, o Superajuste pode ser representado graficamente por uma diminuição continua dos erros de treinamento (curva azul) ao longo das iterações, enquanto que o erro do conjunto de teste aumenta depois de certo tempo (curva vermelha). Esse comportamento nas curvas de erro mostra que o algoritmo está se adaptando exageradamente aos dados de treinamento e não está generalizando bem para dados novos.

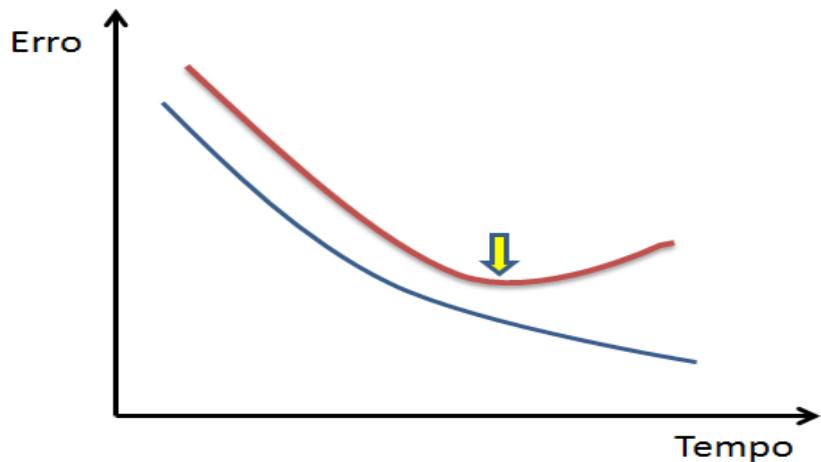


Figura 2.4 - Gráfico do erro de treinamento e teste

Em geral, é comum o acréscimo de um fator de regularização na equação de erro 3.7, resultando na equação 3.11. A adição desse fator de regularização é ajustada por um parâmetro  $\sigma$  que controla a magnitude das matrizes  $P$  (matriz de usuários) e  $Q$  (matriz de itens). Na verdade, ocorre uma penalização para valores muito altos. O valor do fator  $\sigma$  é geralmente calculado por validação cruzada

(KOREN et al., 2009), sendo bastante comum um valor na ordem de 0,02 (YEUNG, 2010).

$$e^2 = (r_{ui} - \sum_{k=1}^K p_{uk} q_{ik}^t)^2 + \sigma(\|q_{ki}\|^2 + \|p_{uk}\|^2) \quad (3.12)$$

onde  $\|\cdot\|$ , é a norma Frobenius (MEYER, 2000).

Para encontrar as novas regras de atualização dos vetores  $p_u$  e  $q_i$ , devemos derivar a equação 3.12, de modo semelhante ao que foi feito com a equação 3.7. As novas regras de atualização são:

$$p'_{uk} = p_{uk} + \alpha \frac{\partial}{\partial p_{uk}} e_{ui}^2 = p_{uk} + \alpha(2e_{ui}q_{ki} - \sigma p_{uk}) \quad (3.13)$$

$$q'_{ki} = q_{ki} + \alpha \frac{\partial}{\partial q_{ik}} e_{ui}^2 = q_{ki} + \alpha(2e_{ui}p_{uk} - \sigma q_{ki}) \quad (3.14)$$

### 2.6.3. BIAS

Na predição de uma nota de usuário  $u$  para um item  $i$ , até agora temos utilizado a equação 3.4, que multiplica os vetores de características correspondentes ao usuário com o vetor de características correspondentes ao item. Esta equação captura a interação dos usuários com os itens. No entanto, alguma parcela da nota predita pode ser atribuída ao usuário ou ao item em questão de modo independente da interação entre eles. Por exemplo, alguns usuários podem ser mais críticos que outros, possuindo assim uma nota média baixa. Alguns itens podem tender a serem avaliados sempre com notas altas pelos usuários (e.g., um filme ganhador do Oscar). Nestes casos pode não ser satisfatório explicar a nota apenas utilizando o produto dos vetores  $p_u$  e  $q_i$  (KOREN et al., 2009). De maneira simples, o problema pode ser resolvido através da seguinte equação:

$$\hat{r}_{ui} = \mu + b_i + b_u + \sum_k p_{uk} q_{ik}^t \quad (3.11)$$

Onde  $\mu$  corresponde à média das avaliações fornecidas na base inteira,  $b_i$  a média de notas sobre o item  $i$  e  $b_u$  a média de notas fornecidas pelo usuário  $u$ . Resumindo, a nota final predita dependerá de um conjunto de 4 fatores (KOREN et al., 2009).

Concluindo, a equação de erro final ( $J$ ) utilizando bias e regularização será dada por:

$$J = \sum_{(u,i) \in R} (r_{ui} - \mu - b_i - b_u - \sum_k p_{uk} q_{ik}^t)^2 + \\ + \sigma(\|q_{ki}\|^2 + \|p_{uk}\|^2) + \lambda(b_u^2 + b_i^2 + \mu^2) \quad (3.12)$$

## 2.7. CONSIDERAÇÕES FINAIS.

Neste capítulo, iniciamos apresentando as três abordagens mais populares em sistemas de recomendação: FBC, FC e Algoritmos Híbridos que tentam superar as fraquezas das duas primeiras abordagens. A filtragem colaborativa foi o foco deste capítulo. Foram apresentadas algumas limitações e desafios que FC tenta superar para realizar previsões de alta qualidade. Essas limitações atingem as duas abordagens de FC que foram descritas neste capítulo: FC baseada em memória e FC baseada em modelos, embora a abordagem baseada em memória sofra mais com os problemas relacionados a esparsidade dos dados (e.g. usuários novos) e escalabilidade (FORSATI et al., 2014). No final deste capítulo, foi descrito um dos modelos de maior sucesso em FC, que é baseado em técnicas de fatoração de matrizes. Este modelo será utilizado neste trabalho. A fatoração de matrizes vem obtendo destaque pelo seu bom desempenho de previsão e sua boa capacidade para trabalhar com grande volumes de dados.

Como mencionado neste capítulo, à utilização de informações complementares a matriz de avaliação de usuários-itens é uma estratégia comum para amenizar problemas de esparsidade dos dados. Informações demográficas, conteúdo dos itens, questionários aos usuários ou informações obtidas das redes sociais podem ajudar aos algoritmos de FC a melhorar a qualidade de suas previsões. No próximo capítulo, descreveremos como informações sobre confiança entre usuários vêm sendo utilizadas nas duas abordagens de FC (baseada em memória e baseada em modelos).

### 3 SISTEMAS DE RECOMENDAÇÃO BASEADOS EM CONFIANÇA

Neste capítulo, faremos a revisão de alguns trabalhos relacionados ao trabalho desenvolvido nesta dissertação. Iniciaremos esclarecendo a motivação para o uso de confiança em recomendação (seção 3.1), em seguida falaremos de recomendação baseada em confiança no contexto de FC (seção 3.2), abordagem baseada em memória (subseção 3.2.3) e abordagem baseada em modelos (subseção 3.2.4). Na subseção 3.2.1 falaremos a respeito de medidas de confiança locais e globais e na subseção 3.2.2 acerca da medida de confiança local Moletrust.

#### 3.1. CONFIANÇA EM RECOMENDAÇÃO

Tradicionalmente, a FC foca principalmente na matriz de usuários-itens ignorando informações sobre a conexão social entre usuários ou relações de confiança. No entanto, em diversas situações de nossas vidas, quando pedimos recomendações de filmes, músicas, restaurantes e livros aos nossos amigos, nós utilizamos conexões sociais para fazer isso. Impulsionado pelo crescimento das redes sociais e das informações provenientes dela, tem surgido uma nova abordagem em recomendação chamada de Recomendação Baseada em Confiança.

Segundo Shi et al. (2011), informações de confiança são um subconjunto de informações sociais, e as duas são informações complementares à matriz usuários-itens. Informações sociais são aquelas derivadas das atividades de uma rede social. Além das informações de confiança ou desconfiança entre usuários, as redes sociais são fonte de outras informações que são fornecidas pelos usuários (e.g., tag, postagens, comentários, informações de demográficas e localização atual) (BOBADILLA et al., 2013). Essas informações também são utilizadas como fontes adicionais para amenizar problemas de esparsidade em FC (BOBADILLA et al., 2013). Esse capítulo não tem como objetivo tratar da utilização destes outros tipos de informações derivadas das redes sociais em FC.

As chamadas “informações de confiança” são informações que estão tipicamente dispostas em forma de grafo e que relacionam dois indivíduos (nós do grafo) da rede através de um valor de confiança (peso da aresta). Tipicamente, o desenho de uma rede de confiança é algo como a figura 3.1. A avaliação de

confiança entre os usuários pode ser explícita ou implícita. Por exemplo, de maneira explícita um usuário  $x$  pode atribuir um valor de confiança para outro usuário  $y$ , mesmo que  $x$  e  $y$  não se conheçam na vida real, e que  $y$  não confie em  $x$ . A ligação de confiança indica simplesmente que  $x$  confia nas avaliações do usuário  $y$ , e acredita que o usuário  $y$  tenha gostos parecidos com os seus. Em contrapartida, nas redes sociais como facebook.com, Google+ e LinkedIn, as conexões são bidirecionais e refletem as ligações do mundo real. Dessa forma, as ligações e de mais algumas informações (e.g., informações demográficas, grau de parentesco entre usuários, interesses em comum) contidas na rede, também podem ser obtidas, de maneira implícita, valores de confiança entre usuários. O trabalho de Chen e Fong (2010) propôs uma mensuração de valores de confiança entre usuários na rede social Facebook.com a partir de suas ligações e informação de conteúdo e atividades em comum entre os usuários. Por definição, pode-se dizer que confiança explícita tende a fornecer melhor predição em relação à confiança implícita (GUO et al. 2014), no entanto, seria necessário um estudo comparativo para responder essa questão.

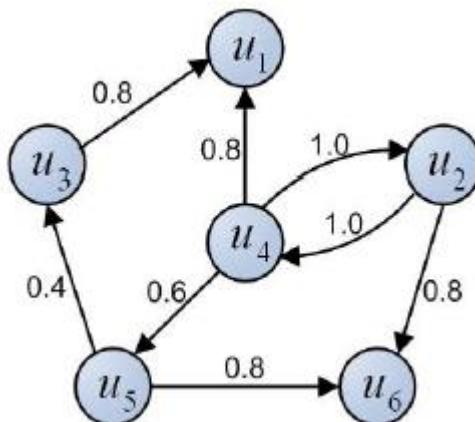


Figura 3.1 - Rede de confiança formada por 6 usuários e valores de confiança entre eles : Fonte: Ma et al. (2008)

Alguns autores, como Jamali e Ester (2010) e Forsati et al. (2014), tratam redes de confiança e redes sociais como sinônimos, por acreditar que as ligações existentes nas redes sociais são ligações de confiança implícita. Neste capítulo, os trabalhos que serão revisados apenas lidam com informações de confiança, sejam

elas obtidas de maneira implícita ou explicita (fornecida de um usuário para outro diretamente).

É comum a utilização de uma representação dos valores de confiança entre usuários em forma de uma matriz  $n \times n$ , onde  $n$  é o número de usuários. Podendo a matriz ser simétrica no caso em que a confiança seja bidirecional. Um exemplo de matriz de confiança é mostrado na tabela 4.1. Neste caso, a tabela representa um sistema onde os valores de confiança variam de 1 a 5, e o relacionamento de confiança é unidirecional.

Tabela 3.1 - Matriz de confiança

	Alice	Pedro	Ana	Mario
Alice	-	1	3	2
Pedro	5	-	3	1
Ana	4	5	-	2
Mario	1	3	4	-

### 3.2. CONFIANÇA EM FILTRAGEM COLABORATIVA

Muitos trabalhos vêm incorporando informações sobre confiança como informação adicional nas suas recomendações por FC, melhorando a acurácia da recomendação e minimizando os problemas relacionados com a esparsidade (e.g. usuário novo, usuário ovelha negra e ovelha cinza) (MASSA & AVESANI, 2007). De um modo geral, pode-se dizer que o problema dos usuários novos é o mais grave dentre os problemas relacionados à esparsidade (MARTINEZ et al., 2009), pois nas bases de dados reais existem muitos usuários deste tipo, aliviar esse problema é um dos objetivos da maioria dos trabalhos que utilizam confiança (GUO et al., 2014).

De maneira geral, a presente seção 3.2 está divida em trabalhos de FC baseada em memória com uso de confiança e trabalhos de FC baseada em modelos com uso de confiança. Nos trabalhos de FC baseada em memória tipicamente são utilizadas medidas de confiança locais, combinadas ou substituindo as medidas de

similaridade. Nos trabalhos de FC baseada em modelos, por sua vez, a grande maioria utiliza modelos de FM com incorporação de uma matriz de usuários-usuários. Na próxima subseção 3.2.1, será abordado o assunto de medidas de confiança locais e globais, e na subseção seguinte, a métrica de confiança local Moletrust 3.2.2. Estes conceitos serão importantes quando foram descritos os trabalhos de FC baseada em memória com uso de confiança.

### 3.2.1. MEDIDAS DE CONFIANÇA GLOBAIS E LOCAIS.

Algumas medidas de confiança foram propostas para inferir valores de confiança entre usuários não diretamente conectados. Por exemplo, um usuário Alice em sistema de recomendação onde usuários utilizam uma escala de confiança de 1 a 5, avaliou diretamente um usuário Pedro com valor de confiança 2. Pedro, por sua vez, avaliou Daniel com valor de confiança 5. Uma métrica de confiança poderia dizer qual o valor que Alice avaliaria Daniel (figura 3.2). O cálculo da propagação de confiança pode variar entre as medidas.

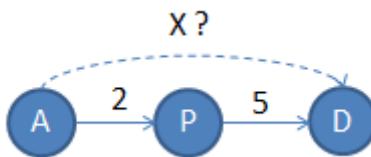


Figura 3.2 - Transitividade da confiança

Medidas de confiança podem ser classificadas em medidas globais e medidas locais (MASSA & AVESANI, 2007a; ZIEGLER & LAUSEN, 2004). Todas as duas tentam predizer a confiabilidade de um dado usuário. Medidas globais tentam determinar um valor de confiança para o usuário que é visto do mesmo modo por qualquer outro usuário do sistema. Sistemas como o [mercadolivre.com.br](http://www.mercadolivre.com.br) e [ebay.com](http://www.ebay.com) usuários podem avaliar outros usuários em relação a confiabilidade deles como vendedores. Esses sistemas agrupam as avaliações fornecidas para cada vendedor gerando uma nota que mede a confiabilidade global dele. PageRank (PAGE et al., 1999) , por exemplo, é uma medida de confiança global que é utilizado pelo motor de busca Google.com. O PageRank considera o link de um site para outro como uma expressão de confiança.

Por outro lado, utilizando uma métrica de confiança local, um usuário pode ter diferentes valores de confiança em relação a diferentes usuários. Massa e Avesani (2004) investigaram a diferença entre medidas globais e locais com relação a usuários controversos. Usuários controversos são usuários que são confiáveis para um conjunto de usuários e não confiáveis para outro conjunto de usuários. Utilizando a base do Epinions.com os autores provaram que utilizando medidas locais eles obtiveram melhor acurácia na recomendação. Medidas de confiança local personalizam melhor a recomendação, pois dependem do julgamento de cada usuário. No caso de usuários controversos isto é particularmente evidente, pois as avaliações deles serão úteis para uns usuários enquanto que para outros não. Uma medida de confiança global não seria adequada para avaliar a confiança de usuários controversos. Os mesmos autores em 2007, chegaram a mesma conclusão ao comparar o uso de uma métrica local Moletrust em relação a uma métrica global PageRank. Em resumo, eles concluíram que medidas globais de confiança não são adequadas para tarefa de encontrar bons vizinhos e fornecer recomendações personalizadas (MASSA & AVESANI, 2007a).

### **3.2.2. MOLETRUST**

Massa e Avesani (2007b) desenvolveram o MoleTrust, uma métrica de confiança local, que trabalha como uma busca em largura, para inferir valores de confiança entre um usuário fonte e outros usuários não diretamente conectados ao fonte. Em MoleTrust é possível informar o nível de propagação horizontal desejado. Quando o nível é 1 atingimos apenas os amigos diretos, para um nível 2 são atingidos até os amigos dos amigos , e assim por diante. Quanto maior o valor deste parâmetro, um número maior de pessoas é levado em consideração, embora o nível de confiança tenda a diminuir para as pessoas em um nível maior de profundidade. Como cada propagação de confiança pode começar por diferentes usuários fonte, o valor de confiança de um usuário em relação a outro pode variar dependendo do usuário fonte escolhido.

MoleTrust pode ser modelado em 2 passos. O primeiro passo é remover os ciclos de uma rede de confiança gerando um grafo direcionado acíclico que parte de um usuário fonte. Sem ciclos o tempo de processamento tende a diminuir, pois assim não existe o risco de um nó ser visitado mais de uma vez. O Segundo passo

consiste em caminhar pelo grafo a partir do usuário fonte calculando os valores de confiança dos nós visitados.

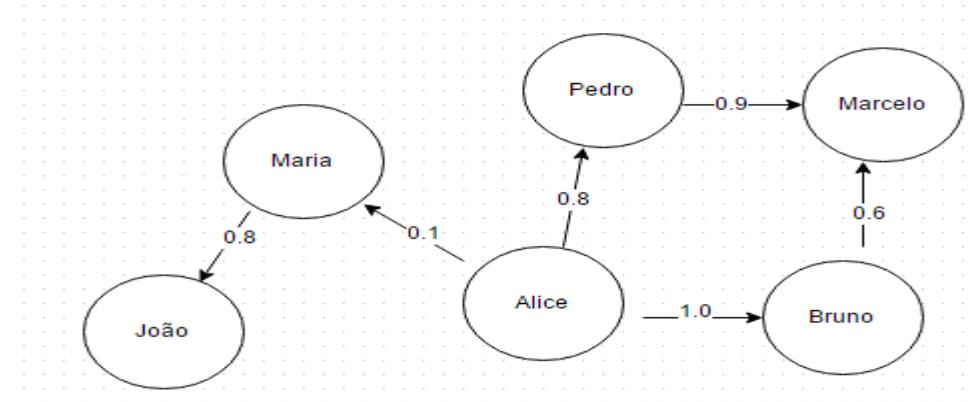


Figura 3.3 - Rede de confiança. Fonte: Adaptada de Massa e Avesani (2007b)

A figura 3.4 abaixo mostra como uma rede de confiança representada na figura 3.3 é modificada após o passo 1, tomando Alice como usuário fonte.

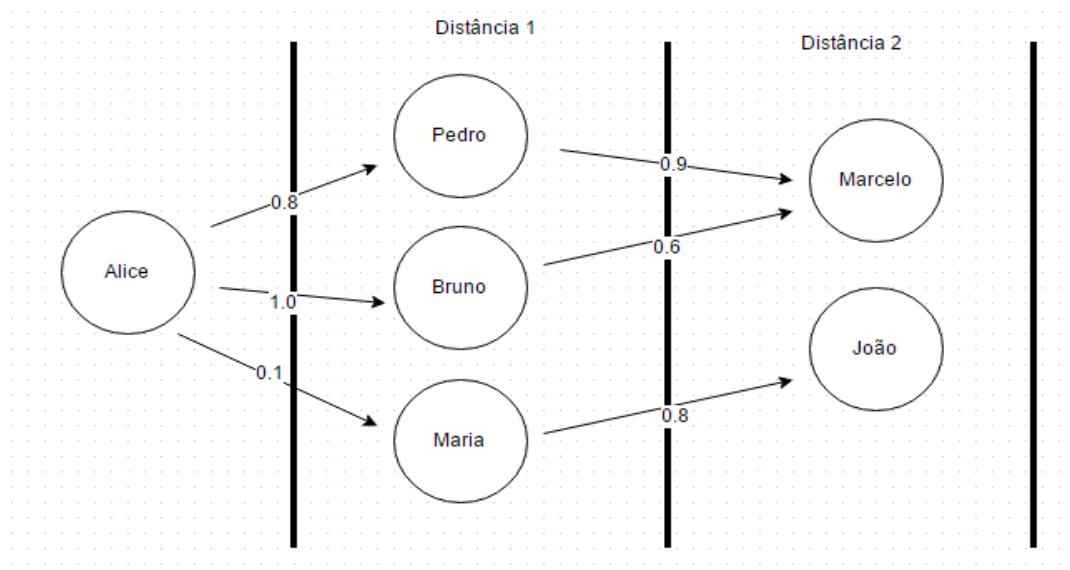


Figura 3.4 - Passo 1 de MoleTrust. Fonte: Adaptada de Massa e Avesani (2007b)

No passo 2, caminha-se a partir do usuário fonte (Alice) em direção aos usuários diretamente conectados (Pedro, Bruno e Maria) computando os seus valores de confiança. Para os usuários diretamente conectados os valores de confiança serão os mesmo já atribuídos pelo usuário fonte. Em seguida, MoleTrust

se move para camada dos usuário com distância 2 e assim por diante, computando os valores de confiança. Pode-se notar que apenas é necessário visitar cada nó uma vez para calcular o valor de confiança. Na camada 2, os usuários Marcelo e João terão seus valores de confiança determinados a partir dos valores dos usuários da camada anterior. O cálculo do valor de confiança para um usuário  $u$  é calculado pela seguinte equação, descrita por Massa e Avesani (2007b).

$$\text{Confiança}(a, u) = \frac{\sum_{i \in \text{predecessores}} (\text{Confiança}(a, i) \times \text{aresta}(i, u))}{\sum_{i \in \text{predecessores}} \text{Confiança}(a, i)} \quad (4.1)$$

Para o usuário Marcelo, por exemplo, o valor de confiança depende dos usuários Pedro e Bruno. O cálculo deste valor é dado por:

$$\text{Confiança}(Alice, Marcelo) = \frac{0.8 \times 0.9 + 1 \times 0.6}{0.8 + 1} = 0.73 \quad (4.2)$$

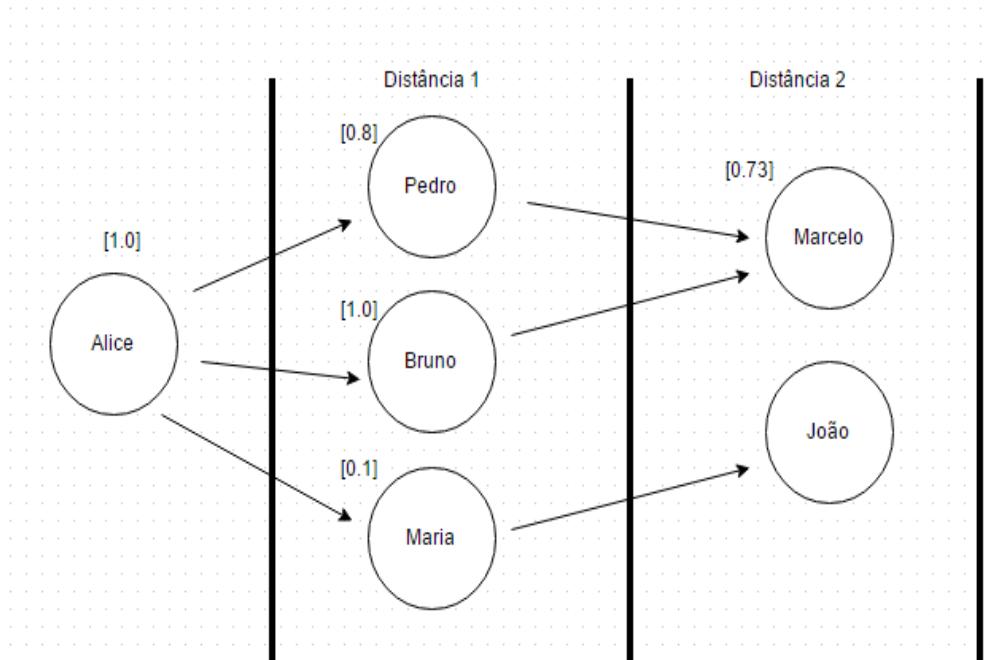


Figura 3.5 - Passo 2 de MoleTrust. Fonte: Adaptada de Massa e Avesani (2007b)

MoleTrust determina que seja adotado um valor mínimo de confiança durante a propagação. O limiar adotado no exemplo foi de 0,5. Usuários que recebem valores de confiança abaixo do limiar adotado são considerados não confiáveis, e suas opiniões não são consideradas no cálculo do valor de confiança de outros

usuários. No exemplo adotado, a usuária Maria recebeu avaliação de confiança 0.1, e como o usuário João apenas foi avaliado por Maria ele não será visitado por Moletrust.

TidalTrust (Golbeck, 2006) funciona de maneira muito similar a Moletrust. A principal diferença entre os dois é que MoleTrust recebe um parâmetro que determina a profundidade máxima da vizinhança (JAMALI & ESTER, 2009)

### **3.2.3. CONFIANÇA EM FC BASEADA EM MEMÓRIA**

Como já descrito no capítulo 2, o primeiro passo da abordagem baseada em memória consiste basicamente na busca dos  $k$  vizinhos a partir de uma medida de similaridade. Essa busca dos  $k$  vizinhos é feita na abordagem tradicional apenas a partir do conjunto de dados de avaliações dos itens (matriz de avaliações usuários-itens).

Atualmente, diversos sistemas têm permitido ao usuário não apenas avaliar os itens mais também a qualidade de outros usuários quanto avaliadores. Deste modo, o usuário para o qual se deseja recomendar, chamado de usuário ativo, tem a possibilidade de escolher diretamente os seus vizinhos, que neste contexto são chamados de amigos (BAO & ZHANG, 2014; HAYDAR et al., 2014). Alguns sistemas permitem que seja atribuído um peso da confiança em relação ao amigo escolhido. Medidas de confiança (MASSA & AVESANI, 2007a; GOLBECK, 2006) têm sido elaboradas para predizer valores de confiança entre dois usuários não diretamente conectados na rede. Alguns trabalhos (MASSA & AVESANI, 2007a; GOLBECK, 2006) têm utilizado essas medidas de confiança no lugar das já tradicionais medidas de similaridade entre usuários. O resultado da troca das medidas de similaridade por medidas de confiança tem mostrado que o uso de confiança tem trazido benefícios para usuários com poucas avaliações e também aumento de cobertura das avaliações preditas. O uso de medidas de similaridade tem mostrado ser inapropriado para descobrir bons vizinhos para usuários com poucas avaliações (JAMALI & ESTER, 2010). Usuários que não avaliaram itens ou avaliaram poucos itens (usuários novos), mas que contêm ligações de confiança podem receber recomendações a partir dos seus amigos confiáveis. Alguns trabalhos que envolvem

o uso de recomendação baseada em memória com recomendação baseada em confiança são descritos a seguir.

Ziegler e Lausen (2004) provaram uma correlação entre confiança e similaridade entre usuários através de um estudo empírico em uma comunidade online chamada All Consuming<sup>5</sup>. O trabalho concluiu que existia uma maior similaridade entre um usuário com seus amigos confiáveis do que outros usuários da comunidade.

Massa e Avesani (2004) mostraram que o uso de confiança pode amenizar muitas fraquezas da FC baseada em memória e em especial o problema de recomendação para usuários novos. Eles mostraram que o simples uso de informações de confiança binária pode melhorar a cobertura e o número de recomendação que podem ser realizadas.

Golbeck (2006) desenvolveu um sistema online denominado Filmtrust, no qual os usuários podem avaliar filmes e também outros usuários em relação a quanto eles confiam nas avaliações desses usuários. Golbeck desenvolveu também TidalTrust, uma métrica de confiança que trabalha como uma busca em largura semelhante a Moletrust, para inferir valores de confiança entre usuários não diretamente conectados. Mesmo com uma base de dados de apenas 500 usuários, Golbeck provou que o uso de recomendação baseada em confiança pode melhorar a acurácia em relação à FC baseada em CCP, principalmente nos casos que a opinião do usuário difere da média.

No trabalho denominado TrustWalker, os autores Jamali e Ester (2009) combinaram recomendação baseada em confiança com recomendação baseada em similaridade dos itens. A ideia principal é percorrer o grafo dos usuários confiáveis de um usuário ativo, procurando quais dos seus amigos confiáveis avaliaram o item ativo (item para qual se deseja predizer uma nota). Caso nenhum amigo do usuário ativo tenha avaliado o item ativo, o algoritmo terá duas opções de caminho a seguir. O primeiro é encontrar um item similar ao item ativo que tenha sido avaliado por algum amigo. O segundo é continuar percorrendo randomicamente o grafo, procurando amigos em uma profundidade maior. O objetivo do algoritmo é balancear

---

<sup>5</sup> <http://allconsuming.net/>

questões como acurácia de predição e cobertura. Muitas vezes os amigos mais próximos de um usuário ativo, não avaliaram o item ativo. Neste caso, o algoritmo procura itens similares, possibilitando assim uma recomendação baseada nas avaliações recebidas por estes itens que são similares. Avaliações de usuários mais próximos ou itens similares são considerados mais confiáveis. A abordagem proposta pelos autores superou a acurácia de predição e cobertura da FC utilizando similaridade via CCP e também as abordagens utilizando apenas medidas de confiança.

### **Proposta de Massa e Avesani**

Na arquitetura proposta por Massa e Avesani (2007a), os autores combinaram FC baseada em memória utilizando CCP, com uma medida de confiança criadas por eles, denominada MoleTrust (subseção 3.2.2). A arquitetura proposta é semelhante à arquitetura de FC tradicional, pois também é baseada em duas etapas, a primeira é o cálculo dos usuários mais similares (vizinhos) e a segunda o cálculo da nota predita utilizando uma estratégia de média ponderada pelo peso da similaridade de cada vizinho (ver seção 1.2 sobre FC baseada em memória). Os vizinhos e os pesos foram calculados de dois modos. O primeiro através do cálculo similaridade via CCP e o segundo através da estimativa de confiança com MT. Depois de encontrados os vizinhos e seus pesos, a nota predita é calculada utilizando a mesma equação 1. Eles concluíram que a combinação da FC com CCP e MoleTrust (FC+MT) não era melhor que a utilização apenas de MT. Utilizando apenas MoleTrust para geração dos vizinhos foi observado um aumento da cobertura (número de avaliações preditas) enquanto a acurácia foi mantida a mesma em relação a acurácia de FC utilizando CCP. Os autores também analisaram os resultados com relação a 6 diferentes visões além da visão geral, para ajudar a entender onde uma técnica era melhor que a outra. As 6 visões foram: usuários novos, usuários com muitas avaliações, itens controversos, itens de nicho, usuário ovelha negra e usuário com opinião forte. Na análise dos resultados, ocorreu um destaque da recomendação baseada em confiança para usuários com poucas avaliações, onde os resultados foram bem melhores que o obtido por FC utilizando CCP.

### 3.2.4. CONFIANÇA EM FC BASEADA EM MODELOS

A fatoração de matriz é uma técnica que modela através de um pequeno número de fatores as atitudes ou preferências dos usuários (KOREN et al., 2009). Ma et al. (2008) desenvolveram o SoRec, onde os autores propuseram a fusão da matriz de usuários-itens com informações da rede de confiança através da matriz de fatores do usuário, o seu resultado mostrou uma melhoria em relação a FM tradicional, que utiliza apenas matriz usuário-item. Um dos problemas é que a proposta tem baixa correspondência com processos de recomendação que acontecem no mundo real.

Logo em seguida os mesmos autores Ma et al. (2009) propuseram um framework chamado Social Trust Ensemble (STE). O objetivo do trabalho era modelar a informação de confiança de maneira mais realista. Baseado em processos observados do mundo real, foi proposto que a nota predita de um usuário a um item dependesse de dois componentes. O primeiro são os gostos pessoais do usuário e o segundo os gostos dos amigos dele. Os resultados obtidos provaram a melhoria em relação a FM probabilística (MNIH & SALAKHUTDINOV, 2007) e ao SoRec (MA et al., 2008). A equação de predição da nota de um usuário  $u$  para um item  $i$  é dada por:

$$\hat{R}_{ui} = g(\alpha P_u^T Q_i + (1 - \alpha) \sum_{v \in N_u} T_{ui} P_u^T Q_i) \quad (4.3)$$

onde  $g(x) = \frac{1}{(1 - e^{-x})}$ , é a função logística.

O parâmetro  $\alpha$  controla o efeito dos vizinhos sobre a nota predita.  $P$ ,  $Q$ ,  $T$  representa a matriz de fatores latentes dos usuários, dos itens e a matriz de confiança. Sendo  $P \in R^{k \times N}$ ,  $Q \in R^{k \times M}$  e  $T \in R^{N \times N}$ .  $N$  e  $M$  representa o número de usuários e itens, respectivamente. Para encontrar o melhor  $\alpha$  foi fixado alguns valores entre 0 e 1, sendo 0.4 o melhor valor de encontrado para o parâmetro

Jamali e Ester (2010) melhoraram o trabalho anterior incorporando a propagação de confiança no modelo proposto por Ma et al. (2008), chamando esse novo modelo de SocialMF.

Bao e Zhang (2014) utilizaram máquina de vetor de suporte (SVM, do inglês: *Support Vector Machine*) para decompor a confiança explícita fornecida em um conjunto de 4 gerais aspectos da confiança (i.e. benevolência, integridade, competência e previsibilidade). Então foi utilizado o valor de confiança predita por SVM dentro do tradicional modelo de fatoração de matriz probabilística (MNIH & SALAKHUTDINOV, 2007).

### **3.3. CONSIDERAÇÕES FINAIS**

A recomendação baseada em confiança é uma abordagem que utiliza informações potencialmente úteis das redes sociais como forma de lidar com o problema da esparsidade das bases de dados. O Trabalho de Massa e Avesani (2007a) mostrou que a utilização de medidas de confiança é mais indicada que a utilização de medidas de similaridade tradicionais nos casos de usuários com poucas avaliações. Para esses casos, as medidas tradicionais de similaridade não são capazes de encontrar bons vizinhos, no entanto, esse tipo de usuário pode se beneficiar das informações de redes sociais e conseguir boas recomendações. A respeito dos trabalhos deste capítulo, na abordagem FC baseada em memória, a grande maioria utiliza medidas de similaridade tradicionais combinadas com medidas de confiança calculadas na rede. Por outro lado, nos trabalhos de FC baseada em modelos, FM se destaca no uso de confiança por suas boas previsões, esses modelos incorporam informações de confiança a partir de matrizes de confiança usuários-usuários.

No próximo capítulo, descreveremos a proposta deste trabalho. A proposta que será apresentada difere dos trabalhos anteriores envolvendo FM ou trabalhos envolvendo medidas de confiança, visto que tenta combinar modelos de FM, geralmente melhores que os algoritmos de FC baseada em memória, com modelos de RCB utilizando Moletrust no lugar de medidas de similaridade, proposto por Massa e Avesani (2007a).

## 4 COMBINANDO FILTRAGEM COLABORATIVA E RECOMENDAÇÃO BASEADA EM CONFIANÇA ATRAVÉS DE MEDIDAS DE ESPARSIDADE

Neste capítulo será apresentada a motivação para o desenvolvimento deste trabalho e sua relação com trabalhos anteriores (seção 4.1), além da descrição do modelo arquitetural proposto (seção 4.2) e as medidas de esparsidade que o modelo utiliza (seção 4.3).

### 4.1. MOTIVAÇÃO

O objetivo deste trabalho é combinar os resultados de duas abordagens (FC e RBC) através de medidas de esparsidade sobre os dados (matriz de avaliações usuários-itens e matriz de confiança usuários-usuários) do sistema Epinions. Do lado da filtragem colaborativa pura será utilizada a técnica de fatoração de matriz e do lado da RBC será utilizada a técnica proposta por Massa e Avesani (2007a) que utilizada a medida de confiança MoleTrust (descrita na subseção 3.2.2).

Experimentos realizados por Massa e Avesani (2007a) mostraram que a RBC utilizando MoleTrust apresenta bons resultados para usuários com poucas avaliações, por outro lado, a Fatoração de Matriz apresenta bons resultados para usuários com muitas avaliações. Deste modo, as abordagens apresentam bons resultados para usuários com características diferentes. Como uma abordagem híbrida envolvendo Fatoração de Matriz e medidas de confiança não tem sido estudada até o momento, decidimos investigar uma estratégia de combinação entre elas.

A Fatoração de matrizes é uma técnica de FC baseada em modelo que utiliza modelos de fatores latentes com grande sucesso em recomendação e tem se tornado a técnica dominante em FC (KOREN et al., 2009). O modelo tradicional de fatoração de matriz apenas utiliza a matriz de usuários-itens para ajustar seus parâmetros (MNIH & SALAKHUTDINOV, 2007). Mais recentemente, alguns trabalhos vêm incorporando informações de confiança dentro do modelo de Fatoração de Matrizes (MA et al., 2009; JAMALI & ESTER; 2010; BAO & ZHANG; 2014). Nesses casos, a informação da rede de confiança é incorporada durante o treinamento do modelo de matrizes. Nesses trabalhos, o uso de informações de

confiança tem diminuído os problemas com usuários com poucas avaliações e melhorado a acurácia da recomendação.

MoleTrust, juntamente com TidalTrust, é uma das medidas de confiança locais mais conhecidas. Seu objetivo é calcular o valor de confiança entre dois usuários não diretamente conectados em uma rede onde os usuários explicitamente informam quem são seus amigos confiáveis. MoleTrust foi escolhido para compor o algoritmo híbrido proposto neste trabalho pela sua popularidade, facilidade de implementação e os bons resultados para usuários com poucas avaliações.

A ideia de combinar algoritmos de recomendação através medidas de esparsidade foi inspirada no trabalho de Anand e Bharadwaj (2011). O trabalho deles utilizou a equação 4.1 para realizar uma combinação linear de técnicas de FC baseada em memória, que utilizam similaridade local e global, respectivamente. O intuito dos autores é ponderar a importância de cada abordagem através de uma medida de esparsidade ( $\beta$ ). Foi observado que quando o conjunto de dados de avaliações é esparsa, os algoritmos que utilizam similaridade local não consegue achar muitos vizinhos e o uso de similaridade global é preferível. Por outro lado, quando a matriz de avaliações é densa existe geralmente um conjunto de vizinhança local rica e deste modo um peso maior é dado para a predição com similaridade local. Os autores consideraram diversas medidas de esparsidade, medidas de esparsidade global da base e também de esparsidade ao nível de usuários e itens.

$$predR = (1 - \beta) * predR_l + \beta * predR_G \quad (4.1)$$

Um estudo sobre a utilização de recomendação baseada em confiança na arquitetura proposta foi uma das futuras direções indicadas pelos autores. Deste modo, foi realizada uma adaptação da equação 4.1 para este trabalho e gerada uma arquitetura que será descrita na próxima seção. Algumas medidas de esparsidade utilizadas no presente trabalho foram baseadas em medidas de esparsidade descritas no trabalho de Anand e Bharadwaj (2011). Diferentemente do trabalho deles, o trabalho atual mede a esparsidade tanto da matriz de avaliações usuários-itens quanto da matriz de confiança usuários-usuários. Foram propostas algumas medidas de esparsidade por usuário e por usuário e item sobre esses dois conjuntos

de dados. Medidas de esparsidade global não foram utilizadas, pois seria necessário trabalhar com mais de uma base de dados.

#### 4.2. ARQUITETURA PROPOSTA



Figura 4.1 - Arquitetura do trabalho proposto

A figura 4.1 mostra a arquitetura deste trabalho, que combina 2 formas básicas de recomendação: (1) recomendação com FC tradicional, que é baseada apenas nas avaliações disponíveis dos usuários aos itens (matriz usuários-itens); e (2) recomendação baseada em confiança, que considera também uma rede social (matriz de usuários-usuários). Na arquitetura proposta, inicialmente as previsões de cada modelo são obtidas separadamente. Em seguida, as previsões são combinadas de forma ponderada:

$$p_{a,i} = \beta_{a,i} p_{a,i}^{FC} + (1 - \beta_{a,i}) p_{a,i}^{RBC} \quad (4.2)$$

onde  $p_{a,i}$  é a nota final predita para o usuário ativo  $a$  e item ativo  $i$ ,  $p_{a,i}^{FC}$  e  $p_{a,i}^{RBC}$  são respectivamente as notas preditas pela recomendação com FC tradicional e pela recomendação baseada em confiança. O parâmetro  $\beta_{a,i}$ , por sua vez, define a ponderação entre as duas técnicas e será definido usando medidas de esparsidade para as matrizes de usuários-itens e de usuários-usuários.

O parâmetro  $\beta_{a,i}$  neste trabalho é definido de forma a aumentar o peso da recomendação baseada em confiança em duas situações opostas:

- (1) Esparsidade alta na matriz de usuários-itens, i.e., quando o usuário ativo tiver poucos itens avaliados. Esse caso é motivado pelos resultados apresentados por Massa e Avesani (2007a), onde RBC se mostrou uma solução para usuários com poucas avaliações, enquanto FC obteve bons resultados para usuários com muitas avaliações.
- (2) Esparsidade baixa na matriz de usuários-usuários, i.e., quando o usuário tiver muitos amigos confiáveis. Neste caso, espera-se que a matriz de usuários-usuários contenha uma maior quantidade de informação potencialmente útil para aumentar o desempenho preditivo de RBC.

Esperamos com a arquitetura proposta, diminuir as taxas de erro observadas pelas técnicas sendo combinadas. Outro aspecto a ser considerado no trabalho é o aumento da cobertura das recomendações. Neste trabalho, quando alguma das técnicas não conseguir retornar uma predição para um dado par de usuário e item, ela não será utilizada para compor a nota predita final. Por exemplo, RBC isoladamente não consegue recomendar itens para usuários que não possuem amigos confiáveis e FC por sua vez quando o usuário não avaliou ainda nenhum item. Esses casos são refletidos diretamente nos valores do peso ( $\beta$ ) de combinação.

Para verificar a viabilidade da nossa proposta, foi implementado um sistema onde MT foi utilizado como componente RBC na arquitetura da figura 4.1 e FM baseado em gradiente foi utilizada como componente de FC, por sua vez. A escolha da técnica de FM se deve ao bom desempenho desse tipo de técnica em comparação com FC baseada em memória (KOREN et al., 2009).

Descreveremos algumas medidas de esparsidade que foram utilizadas na nossa arquitetura. Algumas delas são baseadas nas medidas propostas do trabalho de Anand e Bharadwaj (2011) e adaptadas ao nosso trabalho. O trabalho desses autores difere do nosso como mencionado na seção anterior, pois combinava técnicas de recomendação baseado em vizinhança que utilizavam medidas de similaridade global e local.

#### 4.3. MEDIDAS DE ESPARSIDADE

Neste trabalho, foi avaliado um conjunto de cinco medidas de esparsidade que são descritas abaixo. Todas as medidas geraram valores no intervalo fechado de  $[0, 1]$ . Os valores 0 e 1 também foram atribuídos caso não fosse possível a recomendação por FC ou RBC, respectivamente (ver equação 4.2). As medidas 1, 2 e 3 são medidas de esparsidade específicas do usuário, ou seja, não dependem do item que está sendo recomendado. Neste caso, por simplicidade, será utilizada apenas a notação  $\beta_a$  para o peso da combinação. Por sua vez, as medidas 4 e 5 são medidas específicas de usuários e itens. Neste caso, será mantida a notação  $\beta_{a,i}$  da seção anterior.

O trabalho de Anand e Bharadwaj (2011) definiu uma medida chamada de Medida específica por usuário (USS, do inglês: User Specific Sparsity), dada pela seguinte equação.

$$USS(a) = \frac{n_a}{\max_{u \in U} n_u} \quad (4.3)$$

onde  $n_a$  é o número de itens avaliados pelo usuário ativo  $a$ ;  $U$  é o conjunto de todos os usuários. Nossa trabalho, utiliza a medida USS para servir de base para o cálculo da medida 1.

I. Medida 1: Em bases como a do Epinions onde se tem grande número de usuários com poucas avaliações e um pequeno número deles com muitas avaliações, o cálculo através da medida USS tende a ser na grande maioria das vezes muito próximo a 0. Consequentemente, se fosse aplicada a medida USS a equação 4.2 diretamente, teríamos na maioria das vezes um peso muito maior para a nota predita por MT do que a nota calcula por FM. Para tentar diminuir a dispersão entre os valores da medida, foi aplicada uma transformada logarítmica à medida USS. Utilizando o logaritmo da USS, conseguimos encontrar valores, dentro do intervalo de 0 a 1, menos dispersos entre si. Por exemplo, utilizando a equação 4.3, dado que o número máximo de avaliações é de 1023, a taxa fornecida para um usuário que avaliou apenas 1 valerá 0,000978 e para o usuário que avaliou 907 itens é de 0,8866. Depois da aplicação da

transforma a taxa para esses dois casos será de 0,75247 para um usuário que avaliou 1 item e de 0,01307 para o usuário que avaliou 907 itens. No nosso caso, quanto maior o número de itens avaliados por um usuário maior deve ser o peso dado à técnica FM em relação a MT. Baseado nas considerações citadas, a medida 2 será definida por:

$$\beta_a = 1 - \log \text{USS}(a) \quad (4.5)$$

Para um usuário que avaliou uma grande quantidade de itens o logaritmo de USS resulta em um valor bem próximo a 0 e  $\beta_a$  resulta em um valor próximo a 1, deste modo quando o usuário tem muitas avaliações a FM terá peso maior em relação a MT.

- II. Medida 2 : Esta medida também é uma medida específica do usuário. Ela é dada pelo cálculo do número de avaliações do usuário ativo dividido por 2 vezes o valor da mediana do número de avaliações dos usuários sobre itens. Foi utilizado à mediana, pois, na base do Epinions, o histograma construído a partir dos dados do número de avaliações de cada usuário é um gráfico assimétrico positivo devido a grande maioria dos usuários com poucas avaliações. A mediana neste caso sintetiza melhor a massa de dados, pois a média sofre grande influência de dados nos extremos, os chamados *outliers*. Utilizando essa medida, caso um usuário ativo tenha avaliado exatamente a quantidade de itens igual ao valor da mediana, o resultado final da medida 2 valerá 0,5. Ou seja, usuários que avaliam quantidade de itens próximos à mediana tem o mesmo peso para MT e FM na equação 5.2. Já usuários que avaliam uma quantidade maior que a mediana tem maior peso de FM em sua nota final.

$$\beta_a = \frac{n_a}{2 \times \text{Mediana}_{u \in U} n_u} \quad (4.6)$$

- III. Medida 3: A medida 3 é bastante parecida com a medida 2 e é dada pelo cálculo do número de usuários confiáveis do usuário ativo ( $n_{a_a}$ ), dividido pelo dobro da mediana do número de usuários confiáveis de todos os usuários. A

mediana é calcula fazendo uma lista com o número de usuários confiáveis de cada usuário e em seguida obtendo o valor que divide o conjunto ao meio. A ideia é aumentar o peso de MT quando o número de amigos for maior que a mediana.

$$\beta_a = 1 - \frac{na_a}{2x\text{Mediana}_{u \in U}na_u} \quad (4.7)$$

- IV. Medida 4 : Esta medida é similar à medida anterior. No entanto, o cálculo da medida é restrito aos amigos de  $a$  que avaliaram o item  $i$ . Esta medida está baseada em informações de usuários e itens, e utiliza informações de avaliações da matriz de usuários-itens e informações de confiança da matriz usuários-usuários.

$$\beta_{a,i} = 1 - \frac{na_{a,i}}{2x\text{Mediana}_{u \in U}na_u} \quad (4.8)$$

A medida é calculada pelo número de amigos do usuário ativo  $a$  que avaliaram o item  $i$  ( $na_{a,i}$ ), dividido pelo dobro da mediana da quantidade de amigos de todos os usuários. Essa medida de esparsidade leva em consideração que, em geral, usuários que possuem mais amigos confiáveis têm erro de predição menor para MT do que usuários com poucos amigos confiáveis. No entanto, apenas ter amigos confiáveis não garante que todos eles avaliaram o item  $i$  para o qual se quer predizer uma nota. O que mais importa para MT é a quantidade de amigos que avaliaram o item  $i$ .

- V. Medida 5 : Esta medida de esparsidade é baseada no número de amigos confiáveis que avaliaram o item  $i$  ( $na_{a,i}$ ), dividido pelo número total de pessoas que avaliam o item  $i$  ( $Ni$ ). O objetivo é dar mais peso a MT quando mais amigos avaliaram o item  $i$  em relação ao número total de avaliações do item. A medida é representada pela equação 4.9.

$$\beta_{a,i} = 1 - \frac{na_{a,i}}{Ni} \quad (4.9)$$

## 5 EXPERIMENTOS

Neste capítulo, iremos responder as seguintes perguntas: O que foi avaliado? Que conjunto de dados foi utilizado? Como foram comparados os resultados? A primeira pergunta será respondida na seção 5.3, onde serão descritos os algoritmos comparados. O conjunto de dados será apresentado na seção 5.1 e as métricas utilizadas para comparação dos algoritmos serão apresentadas na seção 5.2.

### 5.1. BASE DE DADOS

Epinions é um sistema real e ativo desde 1999 cujo objetivo é compartilhar conhecimento e avaliações sobre itens de diferentes tipos. Funciona também como uma rede social, na medida em que cada usuário pode avaliar outros, dizendo se confia nas avaliações desses usuários. As categorias dos itens variam desde aparelhos eletrônicos como, DVD players, câmeras até filmes e músicas.

A base de dados Epinions utilizada foi coletada por Massa e Avesani (2006) e pode ser encontrada na página do ambiente cooperativo de pesquisa de métricas de confiança TrustLet<sup>6</sup>. Os dados dessa base são formados por dois arquivos. Um conjunto de avaliações de usuários sobre itens com notas que variam de 1 a 5, podendo essas avaliações ser representadas por uma matriz  $N \times M$  de usuários-itens. Um conjunto de avaliações de confiança de usuários sobre usuários, podendo ser representado em uma matriz  $N \times N$  de usuários-usuários. Os totais de cada conjunto de dados são mostrados nas tabelas 5.1 e 5.2, respectivamente. A relação de confiança entre usuários é unidirecional, ou seja, o usuário a pode confiar nos gostos de um usuário b, mas a recíproca pode não ser verdadeira.

Tabela 5.1 - Quantidades do conjunto de avaliações de usuários sobre itens

Número de usuários	40163
Número de itens	139738
Avaliações	664824

<sup>6</sup> [http://www.trustlet.org/wiki/Extended\\_Epinions\\_dataset](http://www.trustlet.org/wiki/Extended_Epinions_dataset)

Tabela 5.2 - Quantidades do conjunto de avaliações de confiança

Número de usuários	33960
Sentenças de confiança	487182

A densidade na matriz de usuários-itens é de aproximadamente 0,012%, ou seja, a matriz tem apenas esse percentual de células preenchidas. Outras bases bem conhecidas como MovieLens<sup>7</sup> e EachMovie<sup>8</sup> têm aproximadamente 4,25% e 2,29% de densidade (Ma et al., 2008). Na base MovieLens, em particular, tem-se a garantia de que todos os usuários têm pelo menos 20 avaliações, o que não acontece na base Epinions.

Na base Epinions, do total de 40163 usuários, um total de 20468 avaliaram menos de 7 itens. Por outro lado, o usuário que avaliou mais itens teve 1023 avaliações sobre itens. Outro dado interessante é que 301053 das avaliações foram de nota 5, o que representa 45% das notas. Do restante das avaliações, 29% avaliaram como 4, 11% como 3, 8% como 2 e 7% como 1 (ver figura 5.1). A média das notas dadas foi 3,99. Na tabela 5.3, é exibido um conjunto de informações a respeito das médias dos usuários e itens do conjunto de dados de avaliações.

Pelos dados apresentados, pode-se perceber que base de dados Epinions contém algumas características desafiadoras para os algoritmos de recomendação atuais, dentre eles: a grande esparsidade dos dados e grande quantidade de usuários com poucas avaliações (50% dos usuários tem menos de 7 avaliações).

<sup>7</sup> <http://www.cs.umn.edu/Research/GroupLens>.

<sup>8</sup> <http://www.research.digital.com/SRC/EachMovie>.

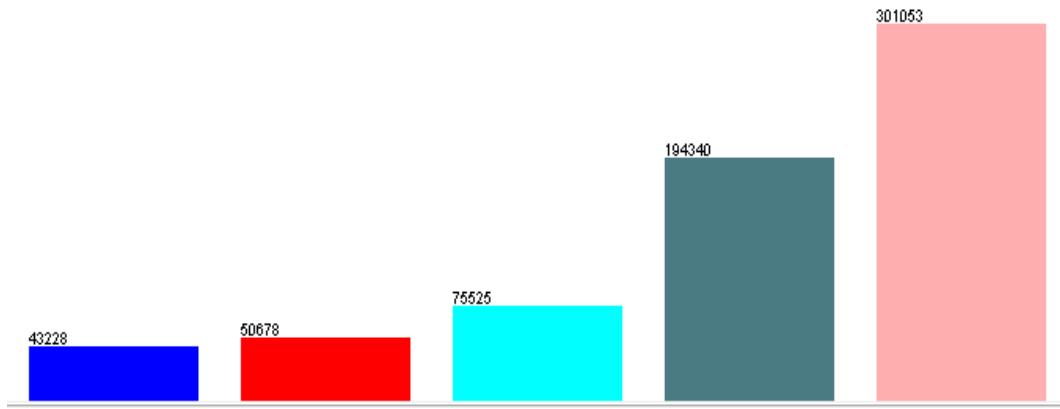


Figura 5.1 - Quantidades exemplos de acordo com a nota

Tabela 5.3 - Estatísticas sobre dados de avaliações de usuários sobre itens

	Por usuário	Por item
Número mínimo de avaliações	1	1
Número máximo de avaliações	1023	2026
Número médio de avaliações	16,55	4,76
Desvio padrão	37,17	20,02
Mediana	6	1

## 5.2. MÉTRICAS DE AVALIAÇÃO

Nesta seção serão descritas as métricas de avaliação dos algoritmos de recomendação comparados neste trabalho. Estes algoritmos serão avaliados quanto a sua acurácia da predição de uma avaliação (nota de um usuário para um item). Existe outra abordagem de avaliação de sistema de recomendação, mas que não é o foco deste trabalho, chamada de recomendação *top-N*, que recomenda os N itens mais relevantes para um usuário. Para obter mais detalhes sobre esse tipo de recomendação e métricas de avaliações para *top-N*, o trabalho de Sarwar (2000) poderá ser consultado.

### 1) MAE E MAE POR USUÁRIO (MAEU)

Media Absoluta dos Erros (MAE) é a média aritmética do módulo das diferenças entre cada valor real de uma avaliação em relação ao valor predito desta avaliação. É definida pela seguinte equação:

$$\text{MAE} = \frac{\sum_{i,j} |R_{i,j} - \hat{R}_{i,j}|}{N} \quad (4)$$

onde  $R_{i,j}$  representa a avaliação real dada por um usuário  $i$  a um item  $j$ , e o  $\hat{R}_{i,j}$  representa a nota predita, e  $N$  é o número de avaliações no conjunto de teste.

Um dos problemas com MAE é que a média dos erros é calculada sobre todas as previsões sem levar em consideração os usuários. Desta forma, previsões para usuários com maior número de avaliações tendem a influenciar mais no resultado. Além disso, sistemas de recomendação tendem a produzir melhores previsões para usuários com muitas avaliações (e.g., usuário antigo e ativo) comparadas com as previsões para usuários com poucas avaliações (e.g., usuário novo). Na base Epinions, temos alguns usuários com mais de 500 avaliações e vários usuários com apenas uma avaliação. Nestes casos, a medida MAE pode ser enviesada.

Para minimizar o problema acima, também utilizamos a Média absoluta dos erros por usuário, proposta por Massa e Avesani (2007a). O MAE por usuário tende a ver cada usuário com o mesmo peso no resultado final. O cálculo é realizado somando os erros médios (MAE) de cada usuário e em seguida dividindo pela quantidade de usuários existentes.

### 2) COBERTURA E COBERTURA POR USUÁRIO

Outra importante medida na avaliação de sistemas de recomendação é a Cobertura. Cobertura corresponde à fração de avaliações, ou exemplos do conjunto de teste, que conseguem ser preditos pelos algoritmos de recomendação. Dependendo da técnica de FC que está sendo utilizada a cobertura pode variar bastante. Em bases de dados como o Epinions, muito esparsa e com grande quantidade de usuários com poucas avaliações, isso se torna uma questão

importante pela dificuldade de predição de algumas avaliações. Por acreditar que a fração pode esconder alguma diferença de cobertura importante entre os algoritmos comparados, neste trabalho, não utilizaremos como medida de Cobertura e a Cobertura por usuário a fração das avaliações cobertas ou usuários cobertos, mas o inteiro que representa o total de avaliações preditas e o total de usuários cobertos (usuários que tiveram pelo menos uma avaliação predita pelo algoritmo de recomendação). No entanto, o percentual pode ser obtido, para isso, basta dividir o total de avaliações preditas pelo total de avaliações do conjunto de teste, ou dividir o total de usuários cobertos pelo total de usuários no conjunto de teste.

### 5.3. MÉTODOS COMPARADOS

Nesta seção serão apresentados os métodos que foram comparados nos experimentos. Os métodos foram comparados utilizando as métricas descritas na seção anterior.

- a) Fatoração de Matriz: Foi utilizada a versão descrita por (KOREN et al., 2009) com bias, fator de regularização e treinada com Gradiente Descendente. Essa técnica não leva em consideração informações sobre confiança.
- b) RBC utilizando MoleTrust: Proposto por Massa e Avesani (2007a), utiliza as informações de confiança para encontrar os melhores vizinhos. Esse algoritmo se mostrou efetivo para usuários com poucas avaliações. Está descrito com maiores detalhes na subseção 3.2.2.
- c) Proposta Híbrida: Método proposto neste trabalho e descrito na seção 4.2. Foram utilizadas cada uma das 5 medidas de esparsidade descritas na seção 4.3, e também o  $\beta$  valendo 0,5, que é igual a combinação de resultados através de média aritmética entre as duas primeiras técnicas. A partir dessas medidas, foram formadas 6 configurações para a arquitetura proposta. Na análise de resultados, as arquiteturas serão nomeadas com o prefixo Híbrida + número da medida de esparsidade (e.g., Híbrida1, Híbrida2)

### 5.3.1. CONFIGURAÇÃO DOS PARÂMETROS

Para a FM, treinado com gradiente descendente, foi utilizado para o número de características ( $k$ ), taxa de aprendizagem ( $\alpha$ ) e taxa de regularização ( $\sigma$ ) os valores 5, 0.002, 0.02, respectivamente. Para chegar a esta configuração, foram analisadas 3 possibilidades de valores para cada parâmetro. Número de características  $\in \{5, 10, 15\}$ , taxa de aprendizagem  $\in \{0.0002, 0.002, 0.02\}$  e taxa de regularização  $\in \{0.002, 0.02, 0.2\}$ . Os conjuntos de valores de cada parâmetro foram baseados em trabalhos anteriores (MA et al., 2009; JAMALI & ESTER; 2010; WEIDELE; 2014) que também envolvem Fatoração de Matriz com a base Epinions. Foram testadas todas as possíveis combinações entre os valores dos parâmetros gerando um total de 27 possibilidades de configuração para a técnica de FM onde a melhor foi selecionada (menor média absoluta dos erros de teste). Foi utilizado um número de 1000 iterações como critério de parada do algoritmo gradiente descendente, novamente esse valor foi baseado nos trabalhos anteriores (MA et al., 2009; JAMALI & ESTER; 2010; WEIDELE; 2014). Em todos os 3 trabalhos, o algoritmo de FM só começa a convergir após 300 iterações. Devido à existência de um fator de regularização ( $\sigma$ ) que controla o Superajuste, os autores mostraram pouca diferença na taxa de erro do conjunto de teste para diferentes números de iterações após o algoritmo ter convergido. Por conta deste fato, não foi analisado outros valores para números de iterações.

Para MoleTrust, apenas um parâmetro pode ser configurado, o nível de profundidade da busca de amigos confiáveis, ou seja, o nível de propagação da confiança na rede. Foi avaliada a recomendação utilizando MoleTrust com valor de profundidade 1 e 2, separadamente. A primeira leva em consideração apenas os gostos dos amigos e a segunda dos amigos e também os amigos dos amigos.

O Algoritmo Híbrido foi executado utilizando os mesmos parâmetros da FM padrão. Como foi utilizado o MoleTrust com profundidades 1 e 2, foram geradas 2 visões do algoritmo Híbrido. A primeira com profundidade 1 e a segunda com profundidade 2.

#### 5.4. METODOLOGIA DOS EXPERIMENTOS

Para a execução dos experimentos e determinação do desempenho de cada técnica foi utilizando o método *Holdout*, um método simples de avaliação de desempenho de algoritmos, que consiste em dividir a base de dados em dois conjuntos: conjunto de treinamento e conjunto de teste. Neste trabalho, foi dividido o conjunto de dados de avaliações de usuários sobre itens na proporção 90% de treinamento e 10% de testes. Para deixar o desempenho obtido menos dependente das partições, foi realizado o *Holdout* 5 vezes, gerando 5 subconjuntos de treinamento e teste (ver figura 5.2). No capítulo 6, serão apresentados os resultados obtidos por cada algoritmo segundo as métricas de avaliação (MAE, MAE por usuário, cobertura e cobertura por usuário), essas métricas foram obtidas das médias do desempenho de cada algoritmo para cada um dos 5 subconjuntos.

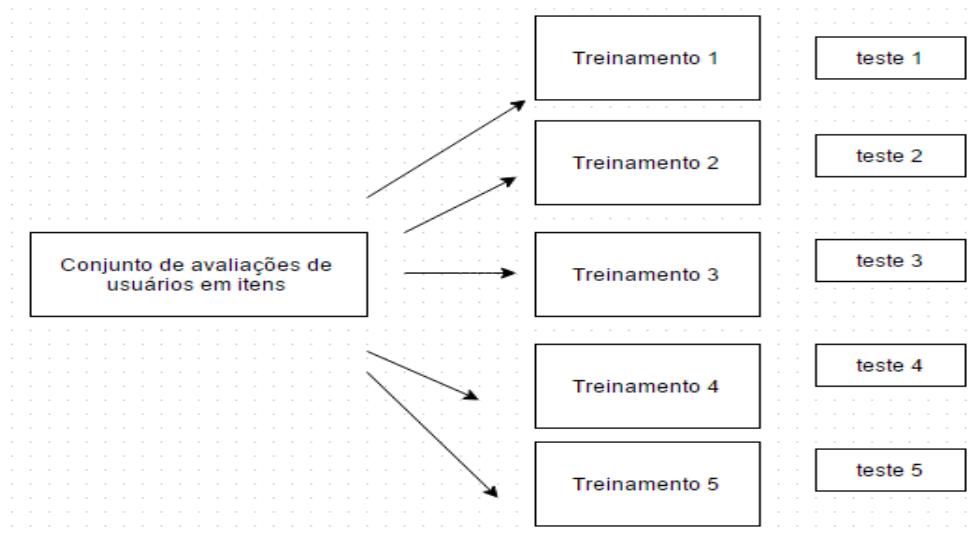


Figura 5.2 - Divisão dos dados

A seleção dos exemplos de teste foi uma seleção estratificada em relação aos usuários da base. A seleção estratificada pelos usuários tende a ser mais justa com usuários com poucas avaliações do que a divisão dos dados com seleção aleatória, pois tenta trazer para o conjunto de testes a mesma proporção de usuários existente na base toda. A seleção aleatória pode enviesar os resultados na medida em que as amostras de avaliações tendem a selecionar usuários com muitas avaliações, na grande maioria das vezes.

Das 664824 avaliações de usuário a itens existentes na base do Epinions, 611756 avaliações pertencem a usuários com mais de 7 avaliações, ou seja metade dos usuários possuem 92% das avaliações, enquanto que a outra metade dos usuários com menos de 7 avaliações possui apenas 8%. A preparação do conjunto de dados de treinamento e teste é uma questão importante, pois dependendo de como for feita poderá subestimar alguns problemas (e.g., usuários com poucas avaliações).

A fim de entender como cada uma das técnicas comparadas se comportava para usuários com diferente quantidade de avaliações foram analisados usuários dentro de 3 categorias: usuários com 0 até 10 avaliações, de 11 a 100 avaliações e mais que 100 avaliações. No capítulo de resultados (capítulo 6), chamaremos essas categorias de *visão 1*, *visão 2* e *visão 3*, respectivamente. As tabelas 5.4 e 5.5 apresentam os quantitativos de avaliações e usuários em cada uma das 3 visões, para o conjunto de dados inteiro e para os 5 conjuntos de teste. Apesar do quantitativo de avaliações variar entre os 5 conjuntos de teste nas visões 1, 2 e 3, todos os conjuntos de teste tiveram o mesmo número de total de avaliações (10% do total da base, segunda coluna da tabela 5.4). Os conjuntos de teste não tiveram o mesmo número total de usuários (ver tabela 5.5, segunda coluna), pois a seleção estratificada por usuário não garante que exista a mesma quantidade de usuários em cada conjunto de teste, apenas que a quantidade de usuários no conjunto de teste se aproxime da quantidade de usuários na base inteira.

Tabela 5.4 - Quantidade de avaliações na base inteira e no conjunto de teste em diferentes visões

	Total	Visão 1	Visão 2	Visão 3
Total de avaliações de usuários em itens na base	664.820	94.231	394.521	176.068
Avaliações no conjunto de teste 1	66.483	9.374	39.520	17.589
Avaliações no conjunto de teste 2	66.483	9.410	39.482	17.591
Avaliações no conjunto de teste 3	66.483	9.362	39.519	17.602
Avaliações no conjunto de teste 4	66.483	9.380	39.500	17.603
Avaliações no conjunto de teste 5	66.483	9.449	39.427	17.607
Média dos 5 conjuntos de teste	66.483	9.395	39.489,6	17.598,4

Tabela 5.5 - Quantidade de usuários na base inteira e no conjunto de teste em diferentes visões

	Total	Visão 1	Visão 2	Visão 3
Total de usuários na base	40.163	25.393	13.888	882
Usuários no conjunto de teste 1	24.144	9.374	13.888	882
Usuários no conjunto de teste 2	24.180	9.410	13.888	882
Usuários no conjunto de teste 3	24.132	9.362	13.888	882
Usuários no conjunto de teste 4	24.150	9.380	13.888	882
Usuários no conjunto de teste 5	24.218	9.448	13.888	882
Média dos 5 conjuntos de teste	24.164,8	9.394,8	13.888	882

## 5.5. FERRAMENTAS UTILIZADAS

Os métodos descritos no trabalho foram desenvolvidos utilizando a linguagem Java, versão 7.0, no ambiente integrado de desenvolvimento Eclipse, versão Luna. Foi empregada uma API Java do projeto Mahout, de Aprendizagem de Máquina, para gerar recomendação com Fatoração de Matriz. O Mahout<sup>9</sup> é um projeto de código fonte aberto mantido pela Apache Software Foundation (ASF). O objetivo principal do projeto é construir algoritmos de aprendizagem de máquina que sejam escaláveis e livres para serem utilizados sob a licença Apache. A versão da API utilizada foi a 0.9, ela implementa a fatoração de matriz descrita por Koren et al., (2009) com bias, regularização e treinada por gradiente descendente. RatingSGDFactorizer é a classe que implementa a recomendação com Fatoração de Matriz.

Foi utilizada a ferramenta WEKA para a preparação do conjunto de dados e seleção estratificada das instâncias do conjunto de teste e treinamento. Foi utilizada a classe `StratifiedRemoveFolds` do WEKA, ela pertence abordagem de filtragem supervisionada de instâncias. Essa função permite realizar uma amostragem estratificada passando como parâmetro o atributo alvo (os usuários, neste caso). Foram selecionados os 5 subconjuntos de treinamento e teste, na proporção de 90% de dados para treinamento e 10% de dados para teste, utilizando esta função do WEKA.

---

<sup>9</sup> <http://mahout.apache.org/>

## 6 RESULTADOS

Neste capítulo serão apresentados os resultados obtidos a partir da proposta de algoritmo Híbrido com as medidas de esparsidade propostas (seção 4.3) e os métodos base que compõem a arquitetura deste trabalho (seção 5.3).

Neste trabalho, foram avaliadas duas possibilidades para o parâmetro que determina a profundidade do algoritmo MT. Deste modo, os resultados serão avaliados para cada uma das profundidades testadas. Na seção 6.1, serão apresentados os resultados para MT utilizando profundidade 1, e na seção 6.2, os resultados para MT com profundidade 2. Na seção 6.3, foi selecionado o melhor resultado deste trabalho para realizar uma comparação com os resultados dos métodos base.

Os resultados foram obtidos através da metodologia descrita na seção 5.4, deste modo, as taxas de erro e coberturas apresentadas neste capítulo correspondem a média dos resultados obtidos para cada um dos 5 subconjuntos de treinamento e teste. Os valores de cobertura foram arredondados para os valores inteiros mais próximos, pois os valores originais são decimais obtidos da média dos 5 subconjuntos de treinamento e teste.

### 6.1. MOLETRUST COM PROFUNDIDADE 1 (APENAS AMIGOS).

Nas tabelas 6.1 e 6.2, podemos observar que todas as propostas de medidas associadas ao algoritmo Híbrido melhoraram as medidas de erro em relação aos métodos base quando todos os usuários são considerados (segunda coluna). A proposta híbrida com medida 1 (Híbrida 1) foi a que obteve a menor taxas de erro<sup>10</sup>. FM obteve um melhor resultado em relação a MT, mas não melhor que a combinação usando  $\beta = 0,5$  (tabelas 6.1 e 6.2, segunda coluna).

Na visão 1 (usuários com 0 até 10 avaliações), MT obteve a menor taxa de erro, no entanto, todas as cinco medidas e o  $\beta$  fixado em 0,5 melhoraram o MAE e o MAE por usuário em relação a FM, o que mostra que a estratégia híbrida conseguiu aliviar o problema de usuários com poucas avaliações em FM.

---

<sup>10</sup> As taxas de erro neste trabalho são calculadas pela MAE e MAE por usuário (ver seção 5.2).

Na visão 2 (usuários com 11 até 100 avaliações) e na visão 3 (usuários com mais de 100 avaliações), novamente todas as cinco medidas melhoraram ou igualaram os resultados (MAE e MAE por usuário) em relação a FM e em relação a MT. Entretanto, para o  $\beta$  fixado em 0,5 o resultado na visão 3 foi pior que o da FM simplesmente. Neste caso, enquanto nas outras medidas o  $\beta$  fornecia peso maior para o resultado de FM no algoritmo híbrido, a medida fixada em 0,5 atribuía o mesmo pesos para os 2 métodos base.

Dado que o algoritmo híbrido proposto combina resultados a partir dos exemplos cobertos por cada um dos 2 métodos base, esse algoritmo apresenta a mesma cobertura para todas as 5 medidas de esparsidade testadas (ver tabela 6.3 e 6.4). Em outras palavras, as diferentes medidas de esparsidade não afetam a cobertura da proposta híbrida deste trabalho.

Em relação à cobertura dos exemplos de teste preditos (avaliações de usuários a itens), o algoritmo Híbrido cobriu 109 exemplos a mais que FM — método base de maior cobertura de exemplos de teste (ver tabela 6.3, segunda coluna). O ganho de cobertura por usuário foi de 63 usuários em relação aos usuários cobertos pela FM (ver tabela 6.4, segunda coluna).

As tabelas 6.1 e 6.2 mostram a média absoluta dos erros (MAE e MAE por usuário) dos algoritmos. Como descrito na seção 5.2, a MAE é calculada somando o módulo de todos os erros obtidos entre cada valor predito e cada valor real das avaliações do conjunto de teste, e dividindo pelo total de avaliações do conjunto. Intuitivamente podemos considerar os resultados apresentados nas tabelas 6.1 e 6.2 como bons, visto que a escala de avaliação dos itens na base Epinions varia de 1 a 5, e na média o erro de predição foi sempre menor que 1 (número inteiro) para qualquer um dos algoritmos avaliados. O algoritmo MT foi o que apresentou maior erro geral (tabelas 6.1 e 6.2, segunda coluna) com o MAE de 0,8014 e MAE por usuário de 0,8041.

Observando as tabelas 6.1 e 6.2, pode-se notar que FM é o método base dominante sobre os resultados da proposta híbrida, dado que os algoritmos híbridos possuem taxas de erro próximas a FM. A causa desta grande influência dos resultados de FM sobre os algoritmos híbridos deve-se principalmente pela

superioridade da cobertura de FM em relação à cobertura de MT. A cobertura de exemplos de testes cobertos e de usuários cobertos de FM é 3 vezes maior que a cobertura de MT (ver tabelas 6.3 e 6.4, segunda coluna). É valido ressaltar que apesar do resultado de MT ser muito bom para usuários com até 10 avaliações (tabela 6.3 e 6.4, terceira coluna), ele possui baixa cobertura e por isso influencia menos no resultado final da proposta híbrida. Em resumo, as taxas de erro obtidas pelos algoritmos híbridos são influenciadas diretamente pelo poder de cobertura e taxas de erro dos métodos base, deste modo as tabelas de erro e cobertura devem ser analisados em conjunto.

Tabela 6.1 - MAE (MT com profundidade 1)

MAE (Média Absoluta dos Erros)				
Algoritmos	Visões			
	Todos	Visão 1	Visão 2	Visão 3
MoleTrust	0,8014	<b>0,6881</b>	0,8125	0,7995
Fatoração de Matriz	<b>0,7695</b>	0,8191	<b>0,7669</b>	<b>0,7481</b>
Híbrido 1	<b>0,7634</b>	0,8033	<b>0,7620</b>	0,7446
Híbrido 2	0,7673	0,8041	0,7667	0,7482
Híbrido 3	0,7645	0,8089	0,7626	0,7442
Híbrido 4	0,7660	0,8141	0,7642	<b>0,7434</b>
Híbrido 5	0,7666	0,8093	0,7649	0,7470
$\beta = 0,5$	0,7676	0,8118	0,7644	0,7509

Tabela 6.2 – MAE por usuário (MT com profundidade 1)

MAE por usuário				
Algoritmo	Visões			
	Todos	Visão 1	Visão 2	Visão3
MoleTrust	0,8041	<b>0,6880</b>	0,8211	0,8122
Fatoração de Matriz	<b>0,7873</b>	0,8191	0,7717	<b>0,7431</b>
Híbrido 1	<b>0,7786</b>	0,8033	<b>0,7666</b>	0,7395
Híbrido 2	0,7819	0,8041	0,7713	0,7434
Híbrido 3	0,7807	0,8089	0,7668	0,7400
Híbrido 4	0,7839	0,8140	0,7692	<b>0,7389</b>
Híbrido 5	0,7824	0,8092	0,7693	0,7420
$\beta = 0,5$	0,7828	0,8117	0,7682	0,7461

Tabela 6.3 – Cobertura (MT com profundidade 1)

Algoritmos	Cobertura			
	Visões			
	Todos	Visão 1	Visão 2	Visão 3
Total de avaliações	66.483	9.395	39.490	17.598
MoleTrust	17.498	800	9.414	7.284
Fatoração de Matriz	57.475	7.921	35.434	14.121
Híbrido	<b>57.584</b>	<b>7.966</b>	<b>35.456</b>	<b>14.163</b>

Tabela 6.4 – Cobertura por usuário (MT com profundidade 1)

Algoritmo	Cobertura por usuário			
	Visões			
	Todos	Visão 1	Visão 2	Visão 3
Total de usuários	24.165	9.395	13.888	882
MoleTrust	6.708	800	5.078	830
Fatoração de Matriz	22.363	7.921	13.560	882
Híbrido	<b>22.411</b>	<b>7.966</b>	<b>13.563</b>	<b>882</b>

## 6.2. MOLETRUST COM PROFUNDIDADE 2 (AMIGOS E AMIGOS DOS AMIGOS).

Analizando os resultados da proposta híbrida utilizando MT com profundidade 2, os ganhos relativos aos métodos base foram ainda mais destacados, tanto nas medidas de erro quanto na cobertura. Novamente a proposta híbrida com a medida 1 obteve os menores erros (MAE e MAE por usuário), dentre as medidas utilizadas (ver tabela 6.5 e 6.6, segunda coluna). A proposta híbrida com medida 1 melhorou seu desempenho com aumento da profundidade de 1 para 2 e também o desempenho em relação a FM. A proposta híbrida com medida 4 obteve destaque com profundidade 2, chegando a ter um resultado melhor que a proposta Híbrida 1, no entanto, a Híbrida 1 mostrou ser mais equilibrada nas diversas visões e apresentou MAE por usuário superior.

Na visão 1 (usuários com 0 até 10 avaliações), novamente todas as cinco medidas e o  $\beta$  fixado em 0,5 melhoraram o MAE e o MAE por usuário em relação a FM, no entanto os resultados foram melhores com profundidade 2 em relação a profundidade 1, ou seja, amenizou-se ainda mais o problema dos usuários com poucas avaliações em FM.

Nas tabelas 6.7 e 6.8 (segunda coluna), podemos ver que houve o aumento de cobertura dos exemplos de testes e usuários que conseguiram ser preditos. Isto era de se esperar, já que houve aumento da cobertura do método com MT por si. Em relação à FM, a proposta híbrida teve um aumento de 327 exemplos de teste e 147 usuários preditos.

Tabela 6.5 – MAE (MT com profundidade 2)

Algoritmos	MAE			
	Todos	Visão 1	Visão 2	Visão 3
MoleTrust	0,7797	<b>0,7608</b>	0,7910	0,7629
Fatoração de Matriz	0,7695	0,8191	0,7669	0,7481
Híbrido 1	0,7599	0,7884	0,7610	0,7408
Híbrido 2	0,7654	0,7908	0,7667	0,7479
Híbrido 3	0,7626	0,7884	0,7648	0,7423
Híbrido 4	<b>0,7598</b>	0,8034	<b>0,7601</b>	<b>0,7345</b>
Híbrido 5	0,7675	0,8088	0,7661	0,7477
$\beta = 0,5$	0,7675	0,8138	0,7655	0,7464

Tabela 6.6 - MAE por usuário (MT com Profundidade 2)

Algoritmos	MAE por Usuário			
	Todos	Visão 1	Visão 2	Visão 3
MoleTrust	0,7887	<b>0,7608</b>	0,7991	0,7635
Fatoração de Matriz	0,7873	0,8191	0,7717	0,7431
Híbrido 1	<b>0,7719</b>	0,7884	<b>0,7646</b>	0,7351
Híbrido 2	0,7772	0,7908	0,7714	0,7434
Híbrido 3	0,7731	0,7884	0,7663	0,7383
Híbrido 4	0,7772	0,8034	0,7647	<b>0,7295</b>
Híbrido 5	0,7834	0,8088	0,7709	0,7432
$\beta = 0,5$	0,7834	0,8138	0,7680	0,7425

Tabela 6.7 – Cobertura (MT com profundidade 2)

Algoritmos	Cobertura			
	Todos	Visão 1	Visão 2	Visão 3
Total de avaliações	66.483	9.395	39.490	17.598
MoleTrust	38.350	2.709	23.160	12.481
Fatoração de Matriz	57.475	7.921	35.434	14.121
Híbrido	57.802	8.062	35.507	14.233

Tabela 6.8 – Cobertura por usuário (MT com profundidade 2)

Algoritmos	Cobertura por Usuário			
	Todos	Visão 1	Visão 2	Visão 3
Total de usuários	24.165	9.395	13.888	882
MoleTrust	12.915	2.709	9.348	858
Fatoração de Matriz	22.363	7.921	13.560	882
Híbrido	22.510	8.062	13.567	882

### 6.3. COMPARAÇÃO DOS MÉTODOS BASE COM A PROPOSTA HÍBRIDA DE MEDIDA 1.

Nas figuras que seguem, apresentaremos uma comparação de resultados entre a proposta híbrida com medida 1 (melhor resultado entre as medidas) e os métodos base utilizados.

Na figura 6.1, é apresentado o resultado segundo a medida de erro MAEU utilizando MT com profundidade 1. Podemos ver que RBC com MT possui erro muito baixo para usuários com poucas avaliações, o que confirma que essa é uma boa abordagem para resolver esse tipo de problema em FC. Como já mencionado no trabalho Massa e Avesani (2007a), este resultado é bastante relevante dado que 50% dos usuários da base Epinions tem menos de 7 itens avaliados.

Utilizando MT com profundidade 2, ou seja, considerando também os amigos dos amigos além dos amigos diretos, foi notada uma diminuição da acurácia de MT para usuários com poucas avaliações (ver na figuras 6.1 e 6.2). Apesar da diminuição da acurácia de MT na visão 1, houve um aumento da acurácia na visão todos os usuários. Na figura 6.4, podemos observar que a cobertura dos usuários da visão 1 mais do que triplicou, passando de aproximadamente 800 para 2709

usuários. Resumidamente, cobertura e acurácia para MT na visão 1 são inversamente proporcionais, pois quanto maior a profundidade, maior quantidade de amigos e menos similares serão os amigos em relação ao usuário ativo.

Em relação à proposta híbrida com medida 1, houve um aumento da acurácia na visão todos os usuários após o aumento da profundidade. Esse aumento de acurácia deve-se ao aumento da acurácia obtido por MT na visão todos os usuários e também um aumento de cobertura significativo na visão 1 de MT, como mencionado no parágrafo anterior. FM não influenciou no aumento da acurácia da proposta híbrida, pois ele não é influenciado pelo aumento da profundidade.

Pode-se notar observando as figuras 6.3 e 6.4, que também houve aumento de cobertura da proposta híbrida com aumento da profundidade. Mais especificamente, um aumento de 100 usuários entre o algoritmo Híbrido com profundidade 2 em relação ao de profundidade 1. Por outro lado, não houve aumento de cobertura na FM, pois ela não é influenciada com a profundidade. Logo, pode-se concluir que o aumento de cobertura e acurácia ocorrido com a proposta híbrida após o aumento da profundidade, está relacionado exclusivamente com o aumento de cobertura e acurácia que ocorreu em MT.

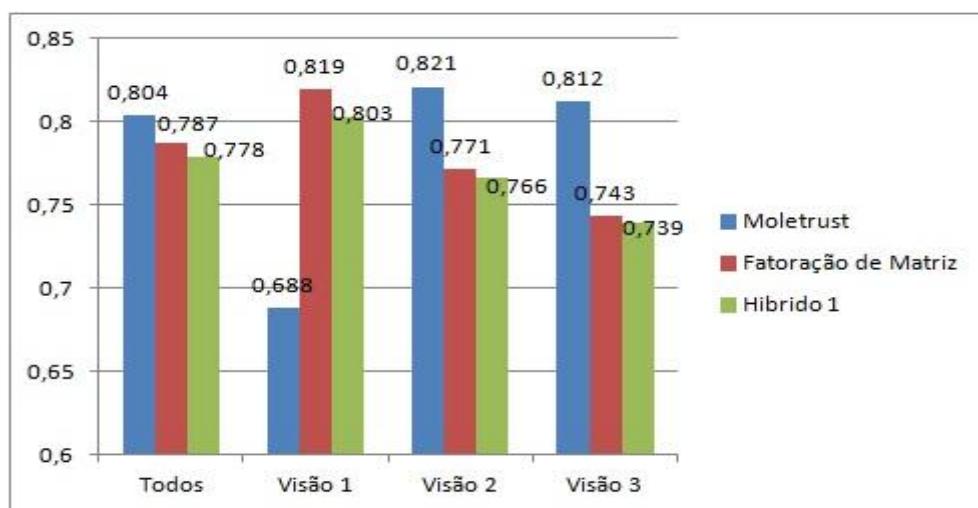


Figura 6.1 - MAEU dos algoritmos em diferentes visões (MT com profundidade 1)

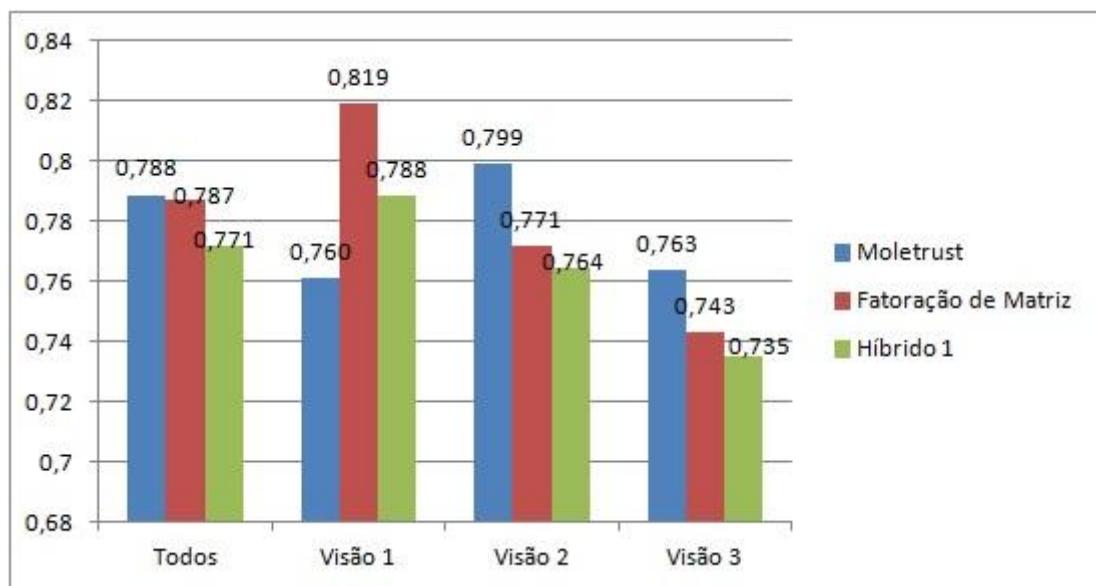


Figura 6.2 - MAEU dos algoritmos em diferentes visões (MT com profundidade 2)

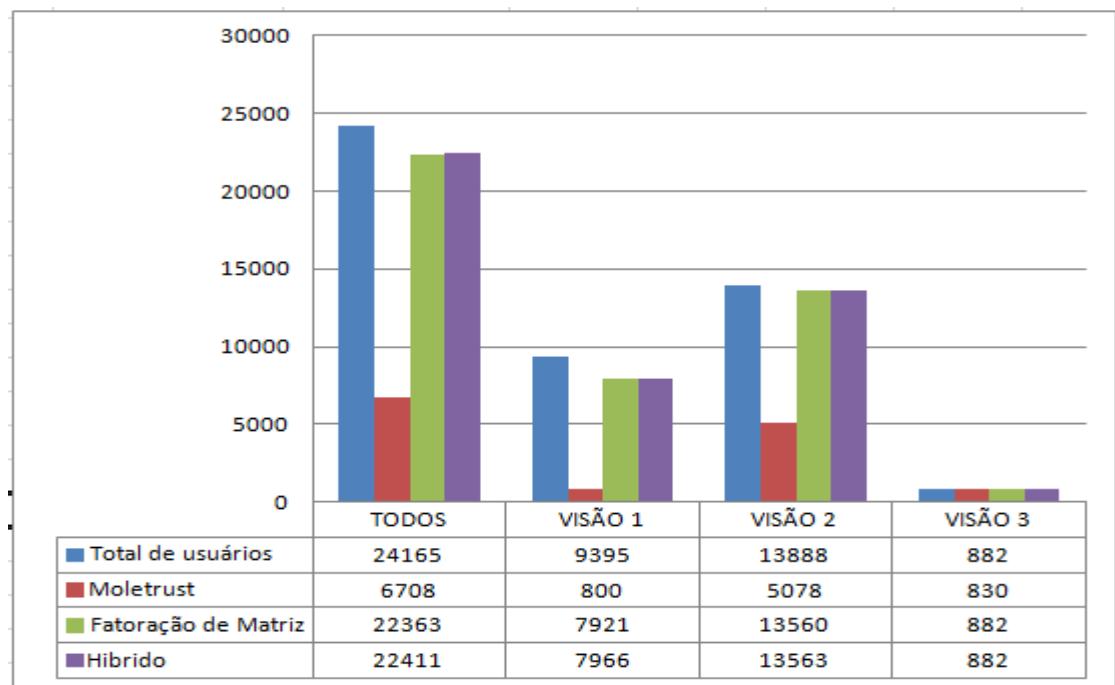


Figura 6.3 – Usuários cobertos pelos algoritmos (MT com profundidade 1)

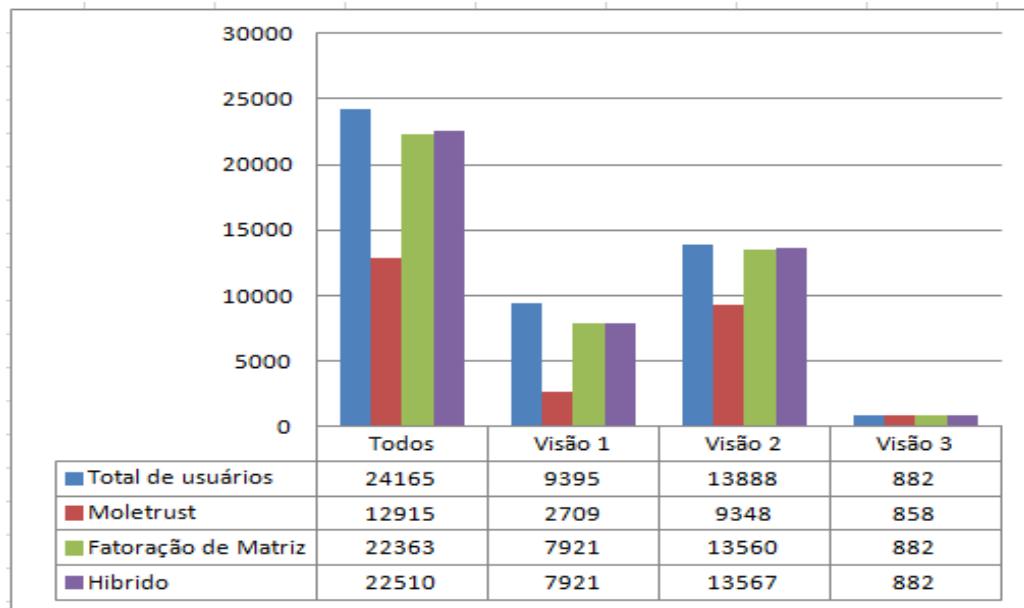


Figura 6.4 – Usuários cobertos pelos algoritmos (MT com profundidade 2)

#### 6.4. DISCUSSÃO DOS RESULTADOS

Nesta seção, sumarizamos os resultados mais importantes. O primeiro deles é o ganho com relação à acurácia que foram obtidos com a proposta híbrida. Na visão geral dos dados (utilizando todos os exemplos de testes ou todos os usuários), todas as medidas de esparsidade utilizadas geraram um ganho de acurácia em relação aos métodos base, nas duas profundidades de MT testadas. O melhor resultado para esse trabalho foi do algoritmo Híbrido com medida 1 utilizando profundidade 2 (ver Tabela 6.6).

O segundo é o ganho de cobertura da proposta híbrida em relação aos métodos base. Um aumento de cobertura de usuários é um resultado importante, pois esse aumento permite que o sistema de recomendação consiga atingir usuários que antes não recebiam nenhuma recomendação. Em relação a FM que é o método base de maior cobertura e acurácia, os resultados com todas as medidas se mostraram superiores nos dois critérios.

Foi realizado um teste-t pareado com nível de confiança de 99%, ou seja, nível de significância de 0,01, para saber se os resultados obtidos com os algoritmos híbridos na visão “todos” diferem estatisticamente dos resultados dos métodos base.

Foi utilizado o teste pareado devido aos algoritmos terem sido treinados e testados nas mesmas condições (mesma plataforma, conjunto de treinamento e testes, etc.). As variáveis aleatórias consideradas no teste-t foram os valores das MAEs e MAEUs de cada algoritmo (tabelas 6.1, 6.2, 6.5 e 6.6, segunda coluna). Por exemplo, para a tabela 6.6 (MAEU com MT utilizando profundidade 2) foram testadas as seguintes hipóteses nulas ( $H_0$ ):  $X_m = X_1, X_m = X_2, X_m = X_3, X_m = X_4, X_m = X_{0,5}, X_m = X_{0,5}$  e  $X_f = X_1, X_f = X_2, X_f = X_3, X_f = X_4, X_f = X_{0,5}$ . A tabela 6.9 contém a descrição dessas variáveis aleatórias. Para realizar os testes-t, foram obtidos também os desvios padrões dos 5 subconjuntos de testes. A tabela 6.11 contém os valores dos desvios padrões dos MAEs e MAEUs de todos os algoritmos na visão “todos”.

Os resultados dos testes determinaram a rejeição das hipóteses nulas, ou seja, é possível dizer com nível de confiança de 99% que todos os resultados das MAEs e MAUs dos algoritmos híbridos diferem estatisticamente dos resultados dos métodos base. Apenas um caso não passou neste teste, que foi o algoritmo híbrido com  $\beta$  fixado em 0,5 para MAE com MT utilizando profundidade 1, no entanto, é possível com uma confiança de no mínimo 80% dizer que existe uma diferença estatística entre os valores deste algoritmo híbrido e os valores dos métodos base.

Tabela 6.9 - Variáveis aleatórias consideradas no testes de hipótese para a tabela 6.6

$X_m$	MAE por usuário utilizando MT com profundidade 2 do algoritmo MT
$X_f$	MAE por usuário de FM
$X_1$	MAE por usuário utilizando MT com profundidade 2 do algoritmo Híbrido1
$X_2$	MAE por usuário utilizando MT profundidade 2 do algoritmo Híbrido2
$X_3$	MAE por usuário utilizando MT profundidade 2 do algoritmo Híbrido3
$X_4$	MAE por usuário utilizando MT profundidade 2 do algoritmo Híbrido4
$X_5$	MAE por usuário utilizando MT profundidade 2 do algoritmo Híbrido5
$X_{0,5}$	MAE por usuário utilizando MT profundidade 2 do algoritmo Híbrido com $\beta = 0,5$

Tabela 6.10 - Desvios padrões dos MAEs e MAEUs dos algoritmos na visão “todos”.

	MT PROFUNDIDADE 1		MT PROFUNDIDADE 2	
	DESVIOS PADRÕES (MAE)	DESVIOS PADRÕES (MAEU)	DESVIOS PADRÕES (MAE)	DESVIOS PADRÕES (MAEU)
MoleTrust	0,002044482	0,005485205	0,003761611	0,002621835
Fatoração de Matriz	0,003912950	0,003538393	0,004104397	0,004115884
Híbrido 1	0,004012828	0,004274203	0,003876974	0,004354386
Híbrido 2	0,004250812	0,004585561	0,004600540	0,005251395
Híbrido 3	0,004329904	0,004280043	0,003901514	0,004779701
Híbrido 4	0,003743417	0,003772397	0,003715969	0,004583297
Híbrido 5	0,004135524	0,004233478	0,004107120	0,004265942
beta=0,5	0,003864770	0,003584131	0,003252745	0,003237131

## 6.5. COMPARAÇÃO COM OS RESULTADOS DO TRABALHO DE MASSA E AVESANI.

O trabalho de Massa e Avesani (2007a) é o que possui a metodologia dos experimentos mais próxima ao trabalho desta dissertação. Os autores utilizaram a mesma base de dados Epinions e as métricas de erro MAE e MAEU. Os autores dividiram os resultados segundo algumas visões dos dados: usuários novos, usuários com muitas avaliações, itens controversos, itens de nicho, usuário ovelha negra e usuário com opiniões fortes. As categorias ou visões utilizadas pelos autores estão resumidas na seguinte tabela.

Tabela 6.11 - Visões sobre os dados no trabalho de Massa e Avesani (2007a)

Usuário novo	Usuário com até 5 avaliações
Usuário com muitas avaliações	Usuário com mais de 10 avaliações
Itens controversos	Itens que receberam avaliações cujo desvio padrão das notas é maior que 1.5
Itens de nicho	Itens que receberam menos de 5 avaliações
Usuário ovelha negra <sup>11</sup>	Usuário com mais de 4 avaliações cuja média das notas fornecidas pelo usuário difere das medias da nota dos itens de 1
Usuário com opiniões fortes	Usuário com mais de 4 avaliações cujo desvio padrão entre as notas é maior que 1.5

<sup>11</sup> Este caso é relatado como ovelha cinza no trabalho de Su e Khoshgoftaar (2009).

O trabalho deles também utilizaram apenas a profundidade 1 e profundidade 2 de MT. A tabela 6.12 abaixo mostra uma comparação com os resultados desta dissertação. O presente trabalho conseguiu resultados bem próximos ao trabalho original de Massa e Avesani para o algoritmo MT. No entanto, é importante ressaltar que existem algumas diferenças de metodologia dos experimentos entre os dois trabalhos o que explica uma diferença entre os resultados. O trabalho de Massa e Avesani utilizou uma estratégia de *Leave-on-out* para testar a predição realizada por MT, no presente trabalho foi utilizado um *Hold-out* na proporção 90/10. O leave-on-out seria muito custoso se aplicado no presente trabalho, pois devido ao treinamento da Fatoração de matriz essa abordagem torna-se inviável. Outra diferença importante é na quantidade de avaliações consideradas para determinação de usuários novos ou usuários antigos (com muitas avaliações). Massa e Avesani consideraram usuários com menos de 5 e mais que 10 avaliações, respectivamente. O trabalho desta dissertação considerou usuários com menos de 10 avaliações como usuários novos. Os resultados para os usuários antigos do trabalho de Massa e Avesani serão comparados com os resultados obtidos para os usuários da Visão 2 desta dissertação (ver tabela 6.11).

Tabela 6.12 – Resultados de MT no trabalho de Massa e Avesani (2007a) e nesta dissertação segundo a MAE e visões de usuário.

	MAE			
	MT com profundidade 1		MT com profundidade 2	
	Usuários novos	Usuários antigos/Visão 2	Usuários novos	Usuários antigos/Visão 2
Massa e Avesani	0,6740	0,8730	0,8330	0,8690
Dissertação	<b>0,6880</b>	0,8125	<b>0,7608</b>	<b>0,7910</b>

Em relação à visão “todos os exemplos de teste”, ocorreu uma pequena diferença entre o trabalho de Massa e Avesani e o trabalho desta dissertação. No trabalho original, houve um aumento do erro geral com o aumento da profundidade, já no trabalho desta dissertação houve uma diminuição do erro geral (ver tabela 6.12). Como mencionado, os trabalho utilizam diferentes metodologias para treinamento e divisão dos dados o que provavelmente podem ter influenciado nos

erros apresentados. Por exemplo, no *Leave-on-out* existe uma garantia que todas as avaliações de base e usuários da base participaram do conjunto de teste a cada rodada, já no *Hold-out* não existe essa garantia. Os resultados com a proposta híbrida 1 mostra que existem uma diminuição significativa do erro em relação ao trabalho original de MT e ao MT implementado nesta dissertação (ver tabela 6.12).

Tabela 6.13 - Resultados de MT no trabalho de Massa e Avesani (2007a) e nesta dissertação segundo a MAE e visão todos.

	MAE	
	MT profundidade 1	MT profundidade 2
	Visão Todos	Visão Todos
MT (Massa e Avesani)	0,8320	0,8460
MT (Dissertação)	0,8014	0,7797
Híbrida 1	<b>0,7634</b>	<b>0,7599</b>

No presente trabalho, não foi avaliado o desempenho da proposta híbrida em relação a outros problemas listados na tabela 6.10, como por exemplo, usuários ovelhas negras, itens de nicho (itens com poucas avaliações), itens controversos, etc. Segundo Massa e Avesani (2007a), MT consegue melhores resultados para esses problemas em relação à FC tradicional. Para entender como seria o desempenho proposta híbrida para esses outros problemas seria necessário visualizar os resultados também sob esses aspectos. No entanto, o escopo do trabalho desta dissertação está limitado a análise de apenas 3 visões (visão 1, 2 e 3) relativas a quantidade de avaliações do usuário. Em trabalhos futuros, poderão ser criadas outras medidas de esparsidade que favoreçam os resultados de MT em relação aos da FM nos casos usuários ovelhas negras, itens de nicho (itens com poucas avaliações), itens controversos, etc.

## 7 CONCLUSÕES

Os sistemas de recomendação têm sido uma ferramenta importante de sugestão de informações para usuários. Como descrito no capítulo 2, FC tem se tornado a abordagem mais popular para sistemas de recomendação, embora sofra de algumas fraquezas intrínsecas, como por exemplo, os usuários com poucas avaliações. Para amenizar esse e outros problemas em FC tradicional, surgiram estratégias que utilizam as informações obtidas das redes sociais sobre confiança entre os usuários.

No capítulo 3, foram descritas as duas principais estratégias de FC que utilizam informações de confiança em suas recomendações: A recomendação baseada em memória com confiança e recomendação baseada em modelos com confiança.

Neste trabalho, investigamos uma estratégia de arquitetura Híbrida (seção 4.2) envolvendo um algoritmo de FC baseada em modelos (Fatoração de Matriz) com um algoritmo de RBC utilizando a medida de confiança MoleTrust. Um conjunto de 5 medidas de esparsidade foi desenvolvido para realizar a combinação ponderada entre os dois algoritmos. Foi testada também a combinação onde cada técnica tinha o mesmo peso na combinação ( $\beta = 0,5$ ), independentemente da esparsidade dos dados.

Os resultados dos experimentos realizados mostraram que a arquitetura proposta melhorou a acurácia e cobertura em relação aos dois algoritmos combinados. A combinação utilizando medidas de esparsidade mostrou ser útil para superar as fraquezas dos dois algoritmos, que na fatoração de matriz são os usuários com poucas avaliações e em MoleTrust são usuários com muitas avaliações.

Diante do trabalho apresentado na dissertação, podemos concluir que as principais contribuições do mesmo foram:

- Revisão do estado da arte nos assuntos investigados
- Desenvolvimento de uma arquitetura Híbrida de FC e RBC através de medidas de esparsidade. Anand e Bharadwaj (2011) foram os primeiros a incorporar medidas de esparsidade dentro de estratégias híbridas de FC, no

entanto, no contexto de RBC essa estratégia não tinha sido investigada, até o momento.

- Foi realizada uma combinação de FM e RBC utilizando MoleTrust. Até o momento, a maioria das estratégias que combinaram FM com confiança o faziam incorporando confiança dentro do modelo de FM (MA et al., 2009; JAMALI & ESTER; 2010; BAO & ZHANG; 2014).
- Desenvolvimento de um conjunto de medidas de esparsidade que podem ser aplicadas ao conjunto de dados de avaliação e/ou ao conjunto de dados de confiança (seção 5.3).
- Melhoria na acurácia de predição em relação aos algoritmos combinados. A melhoria de acurácia obtida foi conseguida pela superação das fraquezas de cada algoritmo base, que em FM é a recomendação para usuários com poucas avaliações e em MT, a recomendação para usuários com muitas avaliações.
- Aumento de cobertura em relação aos algoritmos combinados com a proposta híbrida. Isto possibilitou que mais usuários conseguiram se beneficiar com alguma recomendação personalizada.

## 7.1. LIMITAÇÕES E TRABALHOS FUTUROS

O presente trabalho atingiu uma melhoria em termos de acurácia de predição e cobertura com a proposta Híbrida descrita no capítulo 5 em relação à acurácia e cobertura dos algoritmos de FC e RBC que foram combinados. No entanto, existem alguns pontos interessantes que podem se estender deste trabalho, alguns desses pontos são originados de algumas limitações do trabalho.

- Utilizar outros algoritmos de FC e outras medidas de confiança.
- Investigar a melhor configuração de parâmetros do algoritmo de FM testado para a versão da base Epinions utilizada.
- Não foram investigadas medidas de esparsidade que avaliasse o nível geral de esparsidade da base. Esse tipo de medida poderia ser utilizado para ponderar a importância de uma técnica de FC em relação à RBC a depender do nível de esparsidade da base inteira. Seria interessante trabalhar com mais de uma base, tendo essas bases diferentes níveis de

espadididade. No entanto, o presente trabalho utilizou apenas a base do Epinions.

- Investigar outras medidas de esparsidade por usuário ativo e item ativo, apenas por usuário ativo e apenas por item ativo. Neste trabalho não foi investigada nenhuma medida de esparsidade que dependesse apenas do item ativo. Juntamente com medidas de esparsidade por item ativo, seria interessante analisar como se comportam as diferentes abordagens que serão combinadas para itens com poucas avaliações (item novo) e itens com muitas avaliações.
- Investigar uma estratégia para unificar as medidas de esparsidade da base de avaliação usuário-item e da base de avaliação usuário-usuário. Uma das possibilidades é criar uma equação única a partir das medidas das duas bases através de uma equação ponderada, onde os pesos serão calculados por algum algoritmo de busca e otimização. Anand e Bharadwaj (2011) utilizaram algoritmo genético para criar uma medida de esparsidade unificada em seu trabalho.
- Utilizar bases de avaliações de usuário-usuário com informações de confiança e desconfiança entre usuários.
- Utilizar bases de dados com diferentes níveis de confiança explicitamente informados pelos usuários. Na versão da base Epinions utilizada, os valores de confiança eram binários, e a ocorrência de uma amizade era indicada apenas com o valor 1.
- Realizar uma comparação da proposta híbrida deste trabalho com outros trabalhos que incorporam confiança nos modelos de fatoração de matriz (MA et al., 2009; JAMALI & ESTER, 2010; BAO & ZHANG). Analisar os resultados para usuários com diferentes níveis de avaliação, afim de entender qual das abordagens consegue lidar melhor com os usuários novos.

Esperamos que as limitações apontadas tragam possibilidades para novos trabalhos futuros.

## REFERÊNCIAS

- ADOMAVICIUS, G.; TUZHILIN, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, v. 17, n. 6, p. 734-749, 2005.
- ANAND, D; BHARADWAJ, K. K. Utilizing various sparsity measures for enhancing accuracy of collaborative recommender systems based on local and global similarities. *Expert systems with applications*, v. 38, n. 5, p. 5101-5109, 2011.
- ANDERSON, C. **The long tail: Why the future of business is selling less of more**. Hyperion, 2006.
- BALABANOVIĆ, M.; SHOHAM, Y. Fab: content-based, collaborative recommendation. *Communications of the ACM*, v. 40, n. 3, p. 66-72, 1997.
- BAO, H. F. Y.; ZHANG, J.. Leveraging decomposed trust in probabilistic matrix factorization for effective recommendation. 2014.
- BOBADILLA, J. et al. Recommender systems survey. *Knowledge-Based Systems*, v. 46, p. 109-132, 2013.
- BREESE, J. S.; HECKERMAN, D.; KADIE, C. Empirical analysis of predictive algorithms for collaborative filtering. In: *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1998. p. 43-52.
- CHEN, Wei; FONG, Simon. Social Network Collaborative Filtering Framework and Online Trust Factors: a Case Study on Facebook. In: *ICDIM*. 2010. p. 266-273.
- DESHPANDE, M.; KARYPIS, G.. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, v. 22, n. 1, p. 143-177, 2004.
- FORSATI, R. et al. Matrix Factorization with Explicit Trust and Distrust Relationships. *arXiv preprint arXiv:1408.0325*, 2014.
- FUNK, S. Netflix Update: Try this at Home, Dec. 2006, Disponível em: <<http://sifter.org/~simon/journal/20061211.html>>, Acesso em 25 de junho de 2015

- GAO, L.; LI, C. Hybrid personalized recommended model based on genetic algorithm. In: Wireless Communications, Networking and Mobile Computing, 2008. WiCOM'08. 4th International Conference on. IEEE, 2008. p. 1-4.
- GOLBECK, Jennifer. **Generating predictive movie recommendations from trust in social networks**. Springer Berlin Heidelberg, 2006.
- GUO, G; ZHANG, J.; THALMANN, D.. Merging trust in collaborative filtering to alleviate data sparsity and cold start. *Knowledge-Based Systems*, v. 57, p. 57-68, 2014.
- JAMALI, M.; ESTER, M. Trustwalker: a random walk model for combining trust-based and item-based recommendation. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009. p. 397-406.
- JAMALI, M.; ESTER, M. A matrix factorization technique with trust propagation for recommendation in social networks. In: Proceedings of the fourth ACM conference on Recommender systems. ACM, 2010. p. 135-142.
- KOREN, Y.; BELL, R.; VOLINSKY, C. Matrix factorization techniques for recommender systems. *Computer*, n. 8, p. 30-37, 2009.
- LINDEN, G.; SMITH, B.; YORK, J. Amazon.com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, v. 7, n. 1, p. 76-80, 2003.
- LIU, F.; LEE, H. J.. Use of social network information to enhance collaborative filtering performance. *Expert systems with applications*, v. 37, n. 7, p. 4772-4778, 2010.
- LOH, S. et al. Identifying Similar Users by their Scientific Publications to Reduce Cold Start in Recommender Systems. In: WEBIST. 2009. p. 593-600.
- MA, H. et al. Sorec: social recommendation using probabilistic matrix factorization. In: Proceedings of the 17th ACM conference on Information and knowledge management. ACM, 2008. p. 931-940.

- MA, H.; KING, I.; LYU, M. R. Learning to recommend with social trust ensemble. In: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval. ACM, 2009. p. 203-210.
- MASSA, P.; AVESANI, P. Trust-aware collaborative filtering for recommender systems. In: On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE. Springer Berlin Heidelberg, 2004. p. 492-508.
- MASSA, P.; AVESANI, P. Trust-aware bootstrapping of recommender systems. In: ECAI Workshop on Recommender Systems. 2006. p. 29-33.
- MASSA, P.; AVESANI, P. Trust-aware recommender systems. In: Proceedings of the 2007 ACM conference on Recommender systems. ACM, 2007a. p. 17-24.
- MASSA, P.; AVESANI, P. Trust metrics on controversial users: Balancing between tyranny of the majority. International Journal on Semantic Web and Information Systems (IJSWIS), v. 3, n. 1, p. 39-64, 2007b.
- MARTINEZ, L.; PEREZ, L. G.; BARRANCO, M. J. Incomplete preference relations to smooth out the cold-start in collaborative recommender systems. In: Fuzzy Information Processing Society, 2009. NAFIPS 2009. Annual Meeting of the North American. IEEE, 2009. p. 1-6.
- MEYER, C. D. **Matrix analysis and applied linear algebra**. Siam, 2000.
- MNIH, A.; SALAKHUTDINOV, R. Probabilistic matrix factorization. In: Advances in neural information processing systems. 2007. p. 1257-1264.
- PAGE, L. et al. The PageRank Citation Ranking: Bringing Order to the Web. In: Stanford InfoLab. 1999.
- PARK, M.; HONG, J.; CHO, S. Location-based recommendation system using bayesian user's preference model in mobile devices. In: Ubiquitous Intelligence and Computing. Springer Berlin Heidelberg, 2007. p. 1130-1139.
- RESNICK, P. et al. GroupLens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM conference on Computer supported cooperative work. ACM, 1994. p. 175-186.

RICCI, F.; ROKACH, L.; SHAPIRA, B. **Introduction to recommender systems handbook**. Springer US, 2011.

ROH, T. H.; OH, K. J.; HAN, I.. The collaborative filtering recommendation based on SOM cluster-indexing CBR. *Expert Systems with Applications*, v. 25, n. 3, p. 413-423, 2003.

SAMPAIO, I. A. Aprendizagem Ativa em Sistemas de Filtragem Colaborativa. Dissertação. 2006. 86 p. Dissertação (Ciência da Computação). Universidade Federal de Pernambuco, Recife, PE.

SARWAR, B. et al. Analysis of recommendation algorithms for e-commerce. In: *Proceedings of the 2nd ACM conference on Electronic commerce*. ACM, 2000. p. 158-167.

SARWAR, B. et al. Item-based collaborative filtering recommendation algorithms. In: *Proceedings of the 10th international conference on World Wide Web*. ACM, 2001. p. 285-295.

SHI, Y.; LARSON, M.; HANJALIC, A. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys (CSUR)*, v. 47, n. 1, p. 3, 2014.

SU, X.; KHOSHGOFTAAR, T. M. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, v. 2009, p. 4, 2009.

WEIDELE, D. Social-aware Matrix Factorization for Recommender Systems. 2013. 143p. Dissertação (Mestrado em Engenharia da Informação). Universidade de Constança. Alemanha.

YANG, X. et al. A survey of collaborative filtering based social recommender systems. *Computer Communications*, v. 41, p. 1-10, 2014.

YAGER, R. R. Fuzzy logic methods in recommender systems. *Fuzzy Sets and Systems*, v. 136, n. 2, p. 133-149, 2003.

YEUNG, A. A. Matrix factorization: A simple tutorial and implementation in python. 2010. Disponível em: <<http://www.quuxlabs.com/blog/2010/09/matrix-factorization-a-simple-tutorial-and-implementation-in-python>>, Acesso em 25 de junho de 2015

ZIEGLER, C.; LAUSEN, G. Analyzing correlation between trust and user similarity in online communities. In: Trust management. Springer Berlin Heidelberg, 2004. p. 251-265.