



UNIVERSIDADE FEDERAL DE PERNAMBUCO
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
CENTRO DE INFORMÁTICA

Estudo sobre Sistemas de Recomendação Colaborativos

Aluno: Igor Rafael Guimarães Medeiros (irgm@cin.ufpe.br)

Recife, Abril de 2013

Igor Rafael Guimarães Medeiros

Estudo sobre Sistemas de Recomendação Colaborativos

Monografia apresentada ao curso de Ciências da Computação da UFPE, como requisito parcial para obtenção do título de Bacharel em Ciências da Computação, orientado pelo Prof. Dr.: Ricardo Bastos Cavalcante Prudêncio (rbcp@cin.ufpe.br)

RECIFE, ABRIL DE 2013

Dedico este trabalho a todos aqueles que contribuíram de forma direta ou indireta para conclusão do mesmo.

Agradeço primeiramente a minha família, pais, Janilson e Edlene, e irmãs, Janielle e Camilla, e a Andreza pela paciência e apoio.

RESUMO

Sistemas de recomendação representam as preferências dos usuários com o propósito de sugerir itens de seu interesse. Estes sistemas se tornaram fundamental em aplicações de comércio eletrônico e busca de informações por sua capacidade de filtrar informações que melhor se aplicam aos interesses do usuário de forma efetiva e automática. Partindo do desafio de se propor sistemas de recomendação de qualidade, capazes de escalar e confrontar problemas de dispersão dos dados, diversas técnicas foram propostas nos últimos anos. Atualmente sistemas de recomendação colaborativos são uns dos mais usados para resolver tais problemas por sua eficiência e simplicidade. O presente estudo analisa estes sistemas com o objetivo de implantar e avaliar tais sistemas em uma aplicação real na WEB para aproximar pessoas com preferencias literárias em comuns e recomendar leituras de forma automática bem como apresentar uma visão geral do problema e técnicas usadas para resolvê-lo. Nesse estudo, diferentes formas de calcular a similaridades entre itens e de se obter recomendações serão avaliadas usando-se métricas de desempenho descritas em outros trabalhos de pesquisa.

Palavras-chave: Sistemas de Recomendação; Filtragem colaborativa;

ABSTRACT

Recommendation systems represent the preferences of users to suggest items of their interest. These systems have become critical in e-commerce applications and information search because its ability to filter information that best apply to the user's interests effectively and automatically. On the challenge to produce high quality recommendation systems, able to scale and in the face of data sparsity issues, several techniques have been proposed in recent years. Collaborative recommendation systems are currently one of the most used to solve such problems by its efficiency and simplicity. This study examines these systems in order to deploy and evaluate such systems in a real application on the WEB to bring people with common literary preferences and recommend readings automatically as well as present an overview of the problem and techniques used to solve it. In this study, different ways to calculate the similarities between items and to obtain recommendations will be evaluated using performance metrics described in others research works.

Keywords: Recommender Systems; Collaborative Filtering.

SUMÁRIO

1. Introdução.....	1
2. Sistemas de Recomendação	4
2.1 Definição de Sistemas de Recomendação	5
2.2 Classificação dos Sistemas de Recomendação	6
2.2.1 Filtragem por Conteúdo	7
2.2.2 Filtragem Colaborativa.....	8
2.2.3 Filtragem Híbrida	8
3. Filtragem Colaborativa	9
3.1 Baseado em Memória.....	10
3.1.1 Cálculo da Similaridade entre Usuários.....	11
3.2 Baseado em Modelo.....	13
3.2.1 Similaridade entre Itens	14
3.2.2 Predição	15
3.2.3 Tamanho do Modelo.....	16
4. Metodologia e Resultados.....	18
4.1 Dados	18
4.2 Métricas de avaliação.....	18
4.2.1 Procedimento Experimental.....	19
4.3 Resultados da Experimentação.....	20
4.3.1 Efeito da similaridade	20
4.3.2 Experimentos com o tamanho da vizinhança	20
4.3.3 Experimentos de Qualidade	21

4.1	Sensibilidade do Tamanho do Modelo.....	22
4.2	Discussão	23
5.	Conclusões e Trabalhos Futuros	25
6.	Referências Bibliográficas.....	27

LISTA DE FIGURAS

Figura 1: Matriz de Avaliação dos Usuários	pág. 6
Figura 2: Processo de Filtragem Colaborativa	pág. 10
Figura 3: Processo do Cálculo da Similaridade entre Itens	pág. 14
Figura 4: Desempenho Relativo entre Diferentes Algoritmos de Similaridade	pág.20
Figura 5: Sensibilidade da Vizinhança	pág. 21
Figura 6: Comparativo entre Sistemas Colaborativos	pág. 22
Figura 7: Sensibilidade do Tamanho do Modelo	pág. 23

1. Introdução

A tecnologia reduziu sensivelmente as barreiras para se publicar e disponibilizar informação. Em consequência, a quantidade de informação disponível cresce muito além de nossas capacidades de processá-la. Achar o que desejamos em meio a esse mar de informações pode fácil se tornar uma tarefa bastante árdua. A busca de uma solução frente a essa dificuldade impulsionou o desenvolvimento de diversas áreas de pesquisa. A área de Recuperação da Informação (RI) promove técnicas que possibilitam a busca e recuperação dessas informações de forma eficiente. O usuário realiza uma consulta e o sistema de RI, retorna o maior número de documentos relevantes, de forma rápida, que satisfaça os critérios de busca [3, 27].

Entretanto, os Sistemas de Recuperação da Informação (SRI) podem recuperar documentos que não sejam do interesse do usuário ou um número elevado de resultados [20, 23]. Outro problema é a dificuldade que o usuário por vezes tem de expressar suas necessidades através de uma consulta por palavras chave ou até mesmo saber com precisão o que desejam buscar [23].

Em alternativa e como forma superar as limitações dos SRI clássicos, surgiu a área de pesquisa de Sistemas de Filtragem de Informação (SFI). Enquanto os SRI recuperam informações baseadas na pesquisa do usuário os SFI atuam separando do fluxo, itens irrelevantes dos relevantes para um determinado usuário.

Sistemas de Recomendação (SR) surgiram como uma subárea de pesquisa da área de SFI e evoluiu para uma se tornar uma área independente. A proposta dessa área de pesquisa é construir sistemas capazes de identificar as preferências do usuário e a partir delas recuperar itens que sejam do interesse deste. Estes sistemas

têm sido intensamente pesquisados nos últimos anos e difundidos largamente em sites de comércio eletrônico, jogos, entretenimento, fórum de discussão e redes sociais [30].

Um dos maiores desafios da área de SR é o de produzir recomendações de qualidade, manipulando, muitas vezes, uma quantidade enorme de dados independente das condições adversas que estes dados se encontram. Estes dados podem apresentar problemas como a falta de quantidade de dados suficientes ou a falta de qualidade dos dados. Além disso, os dados podem apresentar grande variância ou dispersão [18]. Em todos os casos, a qualidade da recomendação destes sistemas pode ser bastante comprometida.

O desenvolvimento da área de SR promoveu o surgimento de diversas técnicas para resolver tais problemas. Dentre elas, uma das mais promissoras é Filtragem Colaborativa (FC). A FC baseia-se na ideia simples de que é possível inferir a preferência de uma pessoa por um determinado item ou informação comparando-se o seu registro de interesses com o registro de interesses de um grupo de pessoas que possuem gostos e preferências em comum a ele.

Este estudo analisa estes sistemas de recomendação com o objetivo de implantar e avaliar tais sistemas em uma aplicação real na WEB para aproximar pessoas com preferências literárias em comuns e recomendar leituras e compras de livros de forma automática. Além disso, se faz necessário apresentar uma visão geral da área de Sistemas de Recomendação mostrando sua evolução e sua importância no contexto atual.

O trabalho está dividido em cinco capítulos, descritos resumidamente, a partir do segundo capítulo, a seguir:

No capítulo 2, será apresentado os Sistemas de Recomendação e as principais contribuições dessa área de pesquisa. As principais técnicas serão apresentadas juntamente com as três classificações para SR: Filtragem por conteúdo; Filtragem Colaborativa e Abordagem Híbrida. O capítulo 3 é dedicado a um estudo mais aprofundado da técnica de Filtragem Colaborativa mostrando a variedade de algoritmos propostos de acordo com seus tipos e a diferença entre eles. No capítulo

4, a metodologia de avaliação e os resultados experimentais são mostrados. O capítulo 5 mostra a conclusão do presente estudo e aponta direcionamento para trabalhos futuros finalizando, assim, este estudo.

2. Sistemas de Recomendação

Sistemas de Recomendação se tornaram uma importante área de pesquisa em resposta ao problema de sobrecarga de informação. A área teve início com a publicação do artigo de seu primeiro artigo em meados da década de noventa [25].

Inicialmente os Sistemas de Recomendação foram chamados de Sistemas de Filtragem Colaborativa devido à utilização desse termo no primeiro sistema de recomendação Tapestry [14], sistema onde os usuários indicavam documentos entre si, desenvolvido pela Xerox. Shardanand & Maesusaram usaram o termo “filtragem social” para designar estes sistemas considerando que eles automatizam o processo social de indicação “boca-a-boca” [33]. O termo mais genérico, Sistema de Recomendação, foi proposto por [26]. Eles consideram o termo mais apropriado porque a recomendação não precisa ser necessariamente produto da colaboração entre usuários e recomendações sugerem da indicação de itens relevantes mais do que o processo de filtrar itens não relevantes. Na literatura, é comum aparecer os termos filtragem colaborativa e sistemas de recomendação nos trabalhos da área até hoje.

Desde o surgimento, o interesse de pesquisas na área vem aumentando nos últimos anos. Exemplos de sistemas desenvolvidos academicamente bem sucedidos como os do grupo de pesquisa GroupLens, da University of Minnesota [16] e a vasta aplicação destes sistemas como parte crucial de sistemas de e-commerce e sites de empresas como Amazon, Google, Netflix e eBay, colaboram para o aquecimento desta área de pesquisa [1].

Existem dois grandes desafios fundamentais na área que ainda são objetos de estudos dos pesquisadores [28]. O primeiro desafio é o de aprimorar a escalabilidade de sistemas de recomendação. A demanda dos dias atuais é de manipular dezenas de milhões de dados em tempo real. A Amazon, por exemplo, possui uma plataforma de comércio eletrônico com um catálogo de dezenas de milhões de produtos e algumas centenas de milhões de usuários. A segunda é o melhorar a qualidade das recomendações geradas. De certa forma esses dois

desafios são opostos uma vez que a aplicação de técnicas para reduzir a complexidade computacional implica em sistemas mais escaláveis, por outro lado, podem produzir recomendações de menor qualidade [15, 28].

2.1 Definição de Sistemas de Recomendação

Informalmente, a mais comum formulação do problema de recomendação é a solução para o problema de estimar a avaliação para um determinado item que ainda não foi avaliado por um usuário. Assim que tivermos uma estimativa da avaliação dos itens ainda não vistos, podemos recomendar ao usuário os itens que obtiveram as melhores avaliações estimadas. Intuitivamente, essa estimativa é obtida baseando-se nas avaliações dadas pelo usuário para outros itens [1].

A tarefa, então, de *recomendação* é gerar uma lista com os itens melhores avaliados enquanto que a tarefa individual de estimar a avaliação para um único item é conhecida como *predição* [28].

Uma definição formal para o problema foi proposta por Adomavicius et al. [1] é apresentada a seguir:

Definição: Seja U o conjunto de todos os usuários de um determinado sistema, e seja I o conjunto de todos os possíveis itens que podem ser recomendados como livros, filmes, restaurantes etc. Seja \hat{u} a função utilidade que mede o quão útil é um determinado item i para um determinado usuário u , i.e., $\hat{u}: U \times I \rightarrow A$, onde A é um conjunto totalmente ordenado. Então, para cada usuário $u \in U$, procura-se um item $i' \in I$ que maximiza a utilidade do usuário. Mais formalmente, isto pode ser expresso pela equação abaixo (1):

$$\forall u \in U, i' \in I = \operatorname{argmax}_{i \in I} \hat{u}(u, i)$$

Em Sistemas de Recomendação, a utilidade de um item normalmente é representada por uma nota. No site Netflix, os usuários podem classificar os filmes com notas de um (ruim) a cinco (bom) estrelas.

De forma geral, o problema de recomendação é estruturado como mostrado na figura 1. As preferencias são representadas como uma matriz A de n usuário e m itens onde cada célula $A_{u,i}$ corresponde à avaliação, nota, do usuário u ao item i .

Tipicamente essa matriz é esparsa, pois os usuários geralmente avaliam um subconjunto muito menor que o conjunto I . Por tanto o problema central dos sistemas de recomendação é extrapolar a função de utilidade \hat{u} para todo o conjunto $U \times I$. O usuário em questão no processo de recomendação é referenciado como o *usuário alvo*.

		<i>Itens</i>					
		1	2	...	i	...	m
<i>Usuários</i>	1	5	3		5		
	2		2		4	1	3
	⋮			5			
	a	4	2		?	1	
	⋮			3			4
	n				2		5

Figura 1 – Matriz A de avaliação dos usuários. Cada célula corresponde à avaliação do usuário u para o item i . A tarefa é estimar o valor da avaliação $A_{a,i}$ para o usuário a .

Uma grande variedade de técnicas e formas diferentes de se calcular as estimativas para calcular a avaliação dos usuários para itens ainda não vistos foram propostas na literatura. Métodos de aprendizagem de máquinas, teoria da aproximação e uma quantidade variada de heurísticas. Os sistemas de recomendação são comumente classificados de acordo com forma como eles obtêm essas estimativas.

2.2 Classificação dos Sistemas de Recomendação

Na literatura, a classificação de sistemas de recomendação é dividida em três categorias, propostas por Shoham & Balabanovic [4] e desde então vem sendo complementada por trabalhos subsequentes. São elas:

2.2.1 Filtragem por Conteúdo

A ideia central dessa abordagem, também conhecida como recomendação baseada em conteúdo, baseia-se na premissa de que os usuários gostariam de obter recomendações de itens semelhantes a itens preferidos do usuário no passado [4]. A técnica tem suas origens nas áreas de RI [3, 27] e FI [5].

A melhoria com relação às técnicas clássicas de RI deve-se ao uso de perfis de usuários. Este perfil guarda as preferências, gosto e necessidade do usuário. A construção desse perfil pode ser realizada de forma explícita, e.g. com uso de questionários, ou de implícita, aprendida com o comportamento do usuário com o tempo, e.g. monitoramento de clicks em páginas de e-commerce, tempo de visita entre outros.

A construção do perfil de cada item envolve a extração de características do item em questão. Sistemas de Recomendação baseado em conteúdo tem sido usado com sucesso em aplicações onde os itens contêm informações textuais, como documentos, páginas da web, livros, filmes e notícias [1]. Técnicas de RI são usadas para extrair palavras-chaves deste conteúdo. O perfil do item é representado então como um vetor TF-IDF.

O processo de recomendação baseado em conteúdo seleciona os itens mais similares aos itens identificados no perfil de interesse do usuário. Para isso ele compara o perfil do usuário com o perfil de cada item. O problema então se torna uma tarefa de recuperação de informação, onde o conteúdo associado ao perfil de usuário é tratado como uma consulta (query) e os documentos (itens) não avaliados são classificados e ranqueados de acordo com a relevância/similaridade em relação a essa consulta [4].

Por exemplo, dado que o gênero de um livro seja uma das características extraídas pelo sistema, e sabendo que um usuário gostou dos livros “1984” e “Eu Robô” o sistema poderia inferir uma predileção pelo gênero ficção científica e recomendar o livro “2001, uma Odisseia no Espaço”.

Contudo, um problema com esse tipo de abordagem é que o sistema tende a ser muito especializado recomendando apenas itens similares aos itens já avaliados

pelo usuário. Um usuário que gostou do livro “1984” pode receber apenas recomendações de livros de ficção científica mesmo que o usuário goste de outros gêneros, ou pior, pode receber apenas recomendações de livros escritos por George Orwell. Novidade é uma característica desejável em sistemas de recomendação e estudos foram conduzidos para resolver esse problema [13, 35, 37].

Além do problema de especialização mencionado anteriormente, a filtragem por conteúdo ainda encontra dificuldades com a precisão das recomendações para usuários novos. Um usuário precisa avaliar um número suficiente de itens antes que possa obter recomendações satisfatórias. Do mesmo modo, um novo perfil deve ser criado para cada item novo. Esta tarefa pode ser concretizada usando técnicas de RI caso o item possua conteúdo textual como citado antes, mas nem sempre este é o caso. Existem aplicações como o Pandora (www.pandora.com), sistema de recomendação de músicas, que conseguem extrair características de itens não textuais, no entanto a classificação dos itens é feita de forma manual [8] e conseqüentemente de forma não escalável.

2.2.2 Filtragem Colaborativa

Ao contrário da filtragem por conteúdo, a técnica de filtragem colaborativa, ou sistemas de recomendação colaborativos, tenta prever a avaliação de um determinado usuário para um item baseado nas avaliações dos outros usuários (similares a ele) para este item [1, 28]. Em outras palavras, a ideia é usar a inteligência coletiva de um grupo de pessoas para fazer recomendações para outras [2, 32].

A Filtragem Colaborativa será mais bem estudada no próximo capítulo, pois se mostra como a técnica mais adequada, segundo o autor, para o contexto de construir um sistema de recomendação de livros.

2.2.3 Filtragem Híbrida

A combinação das técnicas de filtragem por conteúdo e por colaboração tem como objetivos superar as limitações individuais de cada uma das técnicas e potencializar os seus benefícios. Diferentes formas de se atingir esse objetivo foram

propostas. De forma geral existem duas formas diferentes, uma é a construção de dois sistemas separados e a combinação de seus resultados como foi proposto em Claypool et al. [9]. A outra forma é combinar os dois sistemas em um único com predominância de uma técnica ou outra. Melville et al. [22] sugeriram um sistema onde a filtragem por conteúdo foi aplicada para reduzir a dispersão dos dados na matriz de avaliação e posteriormente a filtragem colaborativa era aplicada para gerar as recomendações. Essa abordagem se mostrou mais eficiente que abordagem de filtragem puramente colaborativa e puramente de conteúdo.

3. Filtragem Colaborativa

Sistemas de recomendação por filtragem colaborativa, como já foi dito anteriormente, baseiam-se na ideia de que uma pessoa tende a aceitar a sugestão de um grupo de pessoas próximas, ou semelhantes a ela. No entanto para se construir tais sistemas precisamos de um entendimento mais aprofundado de como funciona o mecanismo de recomendação.

Em um típico cenário de sistemas de FC, existe uma lista de m usuários $U = \{u_1, u_2, \dots, u_m\}$ e uma lista de n itens $I = \{i_1, i_2, \dots, i_n\}$. Cada usuário u_i possui uma lista de itens I_{u_i} , em que o usuário tenha expressado suas preferências ou gostos. Vale salientar que $I_{u_i} \subseteq I$ e que o conjunto I_{u_i} pode ser vazio. Definimos também que existe um usuário $u_a \in U$ chamado *usuário alvo* para o qual a tarefa de recomendação é realizada [28].

A tarefa de recomendação pode ser de duas formas:

- **Predição** é um número, $P_{a,j}$, que expressa a nota ou avaliação do item $i_j \notin I_{u_a}$ para o usuário alvo u_a .
- **Recomendação** é uma lista de N itens, $I_r \subset I$, que o usuário ativo provavelmente irá gostar, observando que $I_r \cap I = \emptyset$. Essa forma de recomendação é conhecida como *recomendação Top-N*.

O conjunto de dados usuário-item é representado nos algoritmos de recomendação como uma matriz $m \times n$, A de avaliações. Cada entrada $A_{i,j}$, representa a avaliação, nota, do i -ésimo usuário i para o j -ésimo item j . O processo de Filtragem Colaborativa [28] é mostrado na figura 2.

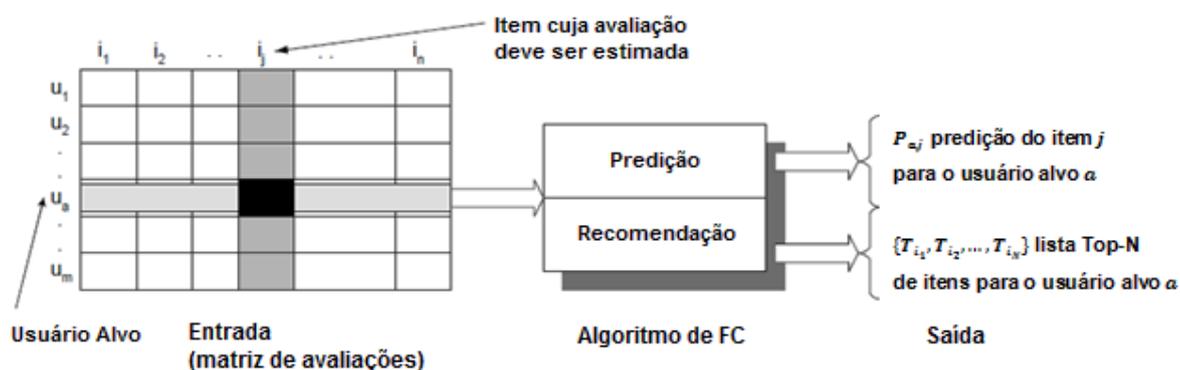


Figura 2 - Processo de Filtragem Colaborativa.

Sistemas de recomendação colaborativos podem ser classificados em duas categorias: sistemas baseados em memória e sistemas baseados em modelo [1].

3.1 Baseado em Memória

Algoritmos baseado em memória, ou baseado em usuários, utilizam toda a matriz de avaliação usuário-item para gerar predições [28, 29]. Estes sistemas usam heurísticas ou técnicas estatísticas para encontrar um grupo de usuário, chamados de *vizinhos* do usuário alvo das recomendações. Estes vizinhos possuem um histórico de concordância nas avaliações com o usuário alvo. Uma vez que esse conjunto de vizinhos é determinando, estes sistemas calculam uma predição ou

recomendação usando a combinação das notas dos vizinhos juntamente com o histórico de avaliações do usuário ativo.

A predição $P_{a,j}$, é dada por uma agregação das avaliações de k vizinhos mais similares ao usuário alvo, u_a . Onde o número k que determina o tamanho da vizinhança de u_a pode variar de 1 a m , número de todos os usuários da base. A função de agregação mais comumente [1] usada é dada por (2):

$$P_{a,j} = \rho \sum_{u' \in U} sim(u_a, u') \times A_{u',j}$$

Nesta equação, ρ é um fator de normalização dado pela equação $\rho = 1 / \sum_{u' \in U} |sim(u_a, u')|$ e u' representa um vizinho de u_a . Outro ponto é que ambos $P_{a,j}$ e $A_{u',j}$ são elementos da matriz de avaliação A . A diferença entre eles é que o primeiro é uma estimativa, predição, dentro da escala de avaliação, e.g., notas entre 1 e 5, enquanto que o segundo é de fato a avaliação dada por um usuário.

3.1.1 Cálculo da Similaridade entre Usuários

A similaridade $sim(x, y)$ é um artifício heurístico usado para definir níveis de semelhança entre os usuários. Isto permite que se encontre o grupo de usuários mais próximos dele, a vizinhança do usuário ativo. O cálculo nos dá uma visão do quão similar é as avaliações dos usuários x e y baseado nas avaliações de itens avaliações comumente entre eles, i.e., são definidas em $I_{xy} = \{i \in I \mid A_{x,i} \neq \emptyset \ \& \ A_{y,i} \neq \emptyset\}$. As duas medidas de similaridade mais usadas em sistemas de filtragem baseada em usuário são mostradas a seguir:

Cosseno

Essa métrica é bastante usada na área de RI para calcular a similaridade entre vetores TF-IDF. Em SR baseados em memória, os usuários são representados

por vetores de dimensão $|I_{xy}|$. A similaridade então é computada (3) como o cosseno do ângulo entre eles.

$$sim(x, y) = \cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\|_2 \times \|\vec{y}\|_2} = \frac{\sum_{i \in I_{xy}} A_{x,i} A_{y,i}}{\sqrt{\sum_{i \in I_{xy}} A_{x,i}^2} \sqrt{\sum_{i \in I_{xy}} A_{y,i}^2}}$$

Onde $\vec{x} \cdot \vec{y}$, representa o produto interno entre \vec{x} e \vec{y} .

Pearson

Um problema que ocorre com a similaridade baseada em cosseno é que ela não considera a diferença entre as notas dos usuários. Um usuário pode ser mais rigoroso em suas avaliações. Pensando nisso outras medidas foram sugeridas como o coeficiente de Pearson (4), que calcula a correlação entre os usuários.

$$sim(x, y) = \frac{\sum_{i \in I_{xy}} (A_{x,i} - \bar{A}_x)(A_{y,i} - \bar{A}_y)}{\sqrt{\sum_{i \in I_{xy}} (A_{x,i} - \bar{A}_x)^2} \sqrt{\sum_{i \in I_{xy}} (A_{y,i} - \bar{A}_y)^2}}$$

Onde \bar{A}_x e \bar{A}_y representam a média das avaliações de x e y respectivamente.

Sistemas baseados em memória foram bastante usados no passado por se mostrarem simples e precisos no processo de recomendação. No entanto, sua popularização revelaram fraquezas no algoritmo em lidar com problemas de dispersão dos dados e escalabilidade [1, 34].

O problema de escalabilidade deve-se ao fato de que o cálculo da vizinhança de um usuário calcula a similaridade entre ele e todos os outros usuários [1, 20, 28]. Essa tarefa é computacionalmente cara. Para contornar esse problema técnicas de redução de do conjunto dos dados como *clustering* ou descartar itens muito

populares ou impopulares foram sugeridas, no entanto essas medidas, no entanto, apesar de se mostrarem eficientes na tarefa de reduzir o conjunto de dados, essas mesmas medidas também reduzem a qualidade das recomendações [20].

A dispersão dos dados também representa um desafio para estes sistemas. Em um sistema de recomendação de livros, mesmo um usuário mais ativo que tenha avaliado 1000 livros em uma base de 1.000.000 terá avaliado apenas 0,1% dos livros. Nesse cenário, o cálculo de I_{xy} pode revelar pouco ou nenhum item em comum para a maioria dos pares de usuários. Implicando em uma precisão e qualidade ruim para o sistema.

3.2 Baseado em Modelo

Sistemas baseados em modelo, diferentemente dos baseados em memória, usam o conjunto de avaliações para aprender um modelo, que é usado então para fazer as previsões e recomendações. Modelos são entidades que sintetizam o comportamento dos dados. Sistemas baseado em modelos foram criados para resolver os problemas dos algoritmos baseado em memória [21, 28]. Estes sistemas analisam a estrutura da matriz A que relaciona usuários e itens para encontrar relações entre os itens. A ideia por trás dessa estratégia vem da intuição de que o usuário se interessaria por itens similares aos itens bem avaliados por ele e evitar os itens similares aos itens que ele não gostou no passado [21, 28]. Além disso, essa técnica não precisa identificar a vizinhança de usuários similares que apresenta o gargalo de desempenho dos algoritmos baseados em memória. Em consequência, tende a produzir recomendações muito mais rapidamente [21, 28].

Diferentes técnicas foram sugeridas para abordar o problema de recomendação por uma representação de modelo. Uma abordagem probabilística é vista em [6] enquanto que [20, 28] apresentam uma abordagem mais tradicional que explora a correlação item-a-item (baseado em itens).

Nas abordagens baseado em itens, analisa o conjunto de itens avaliados pelo usuário ativo, $I_a \in I$, e computa a similaridade de entre esses itens e um item i e depois seleciona uma lista com os k itens mais similares $\{i_1, i_2, \dots, i_k\}$. Ao mesmo

tempo guarda a lista de similaridades correspondente $\{s_{i_1}, s_{i_2}, \dots, s_{i_k}\}$. A predição é então obtida por uma média ponderada dos itens do usuário, I_α , sobre cada um dos itens i .

3.2.1 Similaridade entre Itens

A ideia básica para computar a similaridade entre dois itens i e j é isolar os usuário que tenham avaliados ambos i e j e depois aplicar uma técnica de similaridade para computar a similaridade $s_{i,j}$. O processo é mostrado na figura 3.

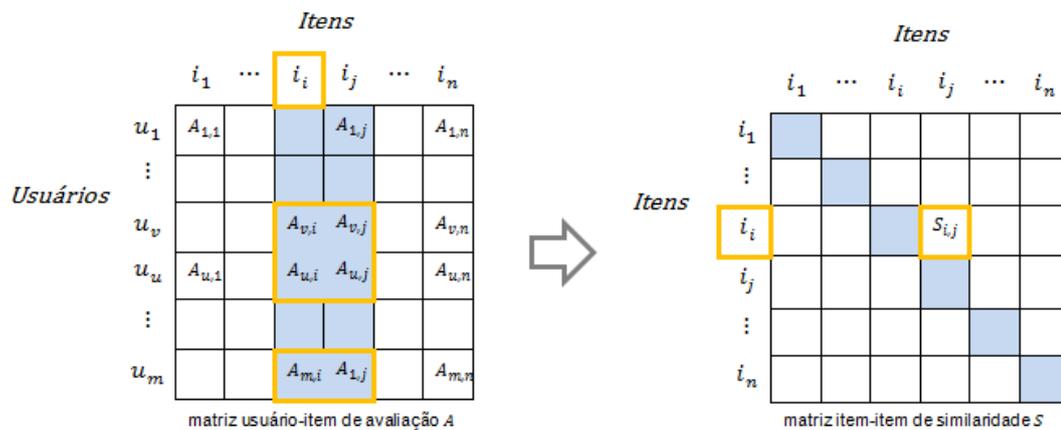


Figura 3 - A computação da similaridade entre i e j ($S_{i,j}$) é feita usando apenas os pares de avaliação de usuários em comuns. Nesse exemplo os pares vêm dos usuários v , u e m .

Assim que os usuários em comuns tenham sido identificados, podemos calcular a similaridade entre eles usando diferentes algoritmos de similaridade. Vemos abaixo alguns deles:

Cosseno

A similaridade entre dois itens, $sim(i, j)$, é dada por uma pequena modificação (5) na equação (3), pois é dada no conjunto, $U_{ij} = \{u \in U \mid A_{u,i} \neq \emptyset \ \& \ A_{u,j} \neq \emptyset\}$.

$$sim(i, j) = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\|_2 \times \|\vec{j}\|_2} = \frac{\sum_{u \in U_{ij}} A_{u,i} A_{u,j}}{\sqrt{\sum_{u \in U_{ij}} A_{u,i}^2} \sqrt{\sum_{u \in U_{ij}} A_{u,j}^2}}$$

Pearson

O coeficiente de Pearson também é usado para calcular similaridades entre itens sendo obtida pela equação (6):

$$sim(i, j) = \frac{\sum_{u \in U_{ij}} (A_{u,i} - \bar{A}_i)(A_{u,j} - \bar{A}_j)}{\sqrt{\sum_{u \in U_{ij}} (A_{u,i} - \bar{A}_i)^2} \sqrt{\sum_{u \in U_{ij}} (A_{u,j} - \bar{A}_j)^2}}$$

Onde \bar{A}_i e \bar{A}_j representam a média das avaliações de i e j respectivamente.

Cosseno Ajustado

Assim como o coeficiente de Pearson, o cosseno ajustado, leva em consideração a diferença entre as notas dos usuários. Esse algoritmo (7) de similaridade foi proposto no trabalho de Sarwar et al. [28].

$$sim(i, j) = \frac{\sum_{u \in U_{ij}} (A_{u,i} - \bar{A}_u)(A_{u,j} - \bar{A}_u)}{\sqrt{\sum_{u \in U_{ij}} (A_{u,i} - \bar{A}_u)^2} \sqrt{\sum_{u \in U_{ij}} (A_{u,j} - \bar{A}_u)^2}}$$

Onde \bar{A}_u é a média de todas as avaliações do usuário u .

3.2.2 Predição

Assim que o conjunto de itens mais similares seja determinado devemos aplicar uma técnica que permita calcular as predições usando o conjunto de itens avaliados. Com predições podemos enfim gerar uma lista Top-N [12] com os melhores itens avaliados como recomendação para o usuário.

Uma das formas de se calcular a predição de um item i para um usuário u é pela soma das notas dadas pelo usuário para itens similares ao item i ponderado pela similaridade $S_{i,j}$, onde j é um dos itens similares a i . Assim, podemos determinar a predição $P_{u,i}$ por (8):

$$P_{u,i} = \frac{\sum_{j \in I_a} (A_{u,j} \times S_{i,j})}{\sum_{j \in I_a} S_{i,j}}$$

3.2.3 Tamanho do Modelo

Como vimos na seção anterior, algoritmos baseado em memória apresentam problemas de escalabilidade. Os algoritmos baseados em modelos têm uma grande vantagem com relação aos primeiros, eles podem ser divididos em duas fases, uma off-line e outra online, i.e., a fase de calculo de similaridades entre itens pode ser computada e armazenada em uma estrutura. No momento da recomendação, um simples acesso a essa estrutura busca pelas similaridades necessárias tornando o processo de recomendação bastante rápido [12, 20, 28]. Outra vantagem é que a escalabilidade deste modelo independe do tamanho da base de usuários.

A estrutura que armazena de similaridade precisa apenas ser atualizada de tempos em tempos para refletir as mudanças nas relações entre os itens. Porém essas relações são estáveis e não mudam tão frequentemente quanto a relação entre usuários.

As similaridades podem ser vistas como uma matriz quadrada de tamanho $n = |I|$. Apesar de economizar tempo, esse modelo requer $O(n^2)$ de espaço. Na prática, precisamos de um número pequeno de itens similares para se realizar a predição. Em consequência, Sarwar et al. [28] sugeriram o conceito de *tamanho do modelo*. Para cada item j computamos os k itens mais similares de j onde $k \ll n$ e registramos esses itens e suas similaridades com j . O número k corresponde ao tamanho do modelo. Para gerar a predição para o item i para o usuário u nesse modelo, o algoritmo recupera a lista dos k itens similares ao item i e faz a interseção com a lista de itens avaliados por u . Em seguida o algoritmo procede com o cálculo padrão da filtragem colaborativa baseada em item descrita em (8).

A responsabilidade por esse problema é da fase de calculo da similaridade entre usuários. Para contornar esse problema, foi sugerido que as similaridades fossem previamente calculadas e armazenadas em uma estrutura de dados (modo

off-line). Assim, no momento em for realizado a recomendação basta acessar os dados de similaridade acelerando o processo de recomendação (modo online). Embora seja uma boa estratégia separar o processo em duas etapas, uma off-line e outra online, a quantidade de usuários do sistema sofre alterações frequentemente o que inviabiliza a aplicação dessa técnica de modelos à filtragem colaborativa baseada em usuário.

Adicionalmente, a técnica de sistemas baseados em itens é mais passível de se usar para explicar as recomendações. Isto porque usuários lembram o que já avaliaram, mas não pode racionalizar sobre a escolha dos vizinhos [19].

4. Metodologia e Resultados

Ao analisar um algoritmo de recomendação estamos interessados no seu desempenho futuro em novos dados, mais do que em seu desempenho nos dados passados [11]. Uma metodologia adequada é necessária para se atingir esse objetivo. É preciso realizar uma análise dos dados, das métricas e definir o procedimento experimental.

4.1 Dados

A base de dados usada para avaliação dos sistemas de recomendação colaborativos é *MovieLens* (ML). ML é um conjunto de dados público não binário extraído de um sistema de recomendação de pesquisa baseado na Web. A escala de avaliação adotada é [1... 5] e 0 representa itens não avaliados. Existem 943 usuários e itens de 1682. Cada usuário avaliou pelo menos 20 itens (filmes). O conjunto de dados tem 100,000 classificações. Outro fator levado em consideração é o nível de dispersão ou esparsidade dado por $1 - \frac{\text{entradas diferente de zero}}{\text{total de entradas}}$. O nível de dispersão de ML é então $1 - \frac{100000}{943 \times 1682} = 0,9369$. Por tanto o nível de dispersão de é de 93,69% e um número médio de classificações iguais a 106 por usuário.

4.2 Métricas de avaliação

A acurácia dos sistemas de recomendação pode ser medida de forma estatística ou por suporte à decisão [1]. Métricas estatísticas de acurácia comparam a estimativa das avaliações com as avaliações reais dos usuários [7, 31] e incluem Raiz do Erro Médio Quadrático (Root Mean Square Error) [RMSE], Erro Médio Absoluto (Mean Absolute Error) MAE, Erro Médio Absoluto Normalizado (Normalized Mean Absolute Error) NMAE. Métricas de suporte à decisão determinam o quão bem um sistema de recomendação pode fazer previsões de alta relevância, ou seja, itens que seriam bem classificados pelo usuário. Nessa categoria encontram-se as métricas clássicas de RI de precisão, i.e. a porcentagem de item verdadeiramente

bem avaliados em relação aos que foram estimados como bons pelo sistema de recomendação. Entre essas métricas podemos citar a medida F (F-Mesure), precisão (Precision) e cobertura (Recall) e curva ROC (Receiver Operating Characteristic) que mostra equilíbrio entre as taxas de verdadeiros positivos e falsos positivos em sistemas de recomendação [7].

O procedimento de avaliar desempenho de sistemas de recomendação usando métricas estatísticas é mais simples de se implementar e por essa razão essa é a forma mais usada [11]. Neste estudo vamos considerar métrica mais usada MAE, pois ela é menos sensível a exemplos díspares nos dados (outliers) do que MSE, por exemplo, e sua interpretação é direta [7, 11].

4.2.1 Procedimento Experimental

Para testar o desempenho futuro e estimar o erro de previsão, temos corretamente particionar o conjunto de dados original em subconjuntos de *treinamento* e *teste*. O conjunto de teste deve ser diferente e independente do conjuntotreinamento a fim de obter uma estimativa confiável do verdadeiro erro. Em [28], Sarwar et al. Experimentaram diferentes proporções de divisão dos dados variando de 20% para testes até 90% em incrementos de 10%. Nos testes realizados os autores escolheram a razão 80/20 (treinamento/teste) como sendo a ideal para os demais experimentos. Seguindo esses resultados o primeiro passo foi dividir 80% dos dados para treinamento e 20% para testes de forma aleatória em relação aos usuários. Para realizar a comparação do erro da predição para cada usuário, foi usada a técnica de deixar um de fora [24] (leave-one-out), i.e., a estimativa de um item do usuário obtida com a média ponderada dos demais itens avaliados pelo usuário.

Todos os experimentos foram implementados em Python 2.7 e realizados em um laptop com processador Intel i3 com velocidade de 2,26 GHz com 4GB de memória RAM.

4.3 Resultados da Experimentação

Os experimentos propostos investigam a qualidade e desempenho dos sistemas de recomendação propostos. Foram realizadas diversas comparações sobre as variáveis de estudo e os resultados são mostrados a seguir:

4.3.1 Efeito da similaridade

Foram implementados três algoritmos diferentes de similaridade: cosseno, correlação e cosseno ajustado. Os testes foram conduzidos em um sistema de recomendação baseado em usuário. A figura 4 mostra que o algoritmo de similaridade do cosseno ajustado proposto por Sarwar et al. [28] obteve o melhor desempenho. Já a similaridade por cosseno, obteve o pior desempenho. Por essa razão este foi escolhido como algoritmo de similaridade para os demais testes.

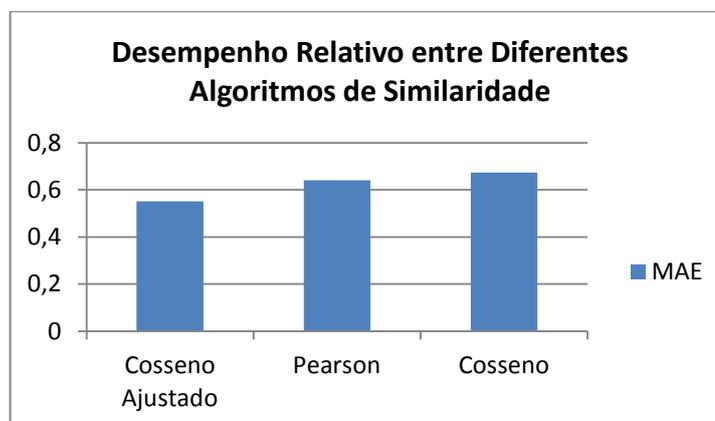


Figura 4 - Impacto de diferentes medidas de similaridades usando FC baseada em usuário.

4.3.2 Experimentos com o tamanho da vizinhança

O número de vizinhos tem impacto significativo no desempenho dos sistemas de recomendação [17]. Para determinar a importância desse parâmetro na qualidade das recomendações o número de vizinhos foi variado gradativamente usando um sistema de filtragem baseada em usuário e o valor do erro MAE foi computado. Os resultados são mostrados na figura 5. Pode se constatar que o valor do MAE cai

drasticamente assim que se considera uma vizinhança maior. No entanto, após atingir um valor mínimo o MAE aumenta e estabiliza. A explicação para isso é que a contribuição dos vizinhos na predição não tão similares tende a degradar a qualidade da predição considerando apenas os vizinhos mais similares. Além disso, à medida que a vizinhança aumenta o custo computacional da predição dada por (2) ou (8) também aumenta. O menor valor do MAE é obtido com vizinhança com tamanho próximo a 40, por esse motivo foi escolhido como o valor ótimo para esse parâmetro.

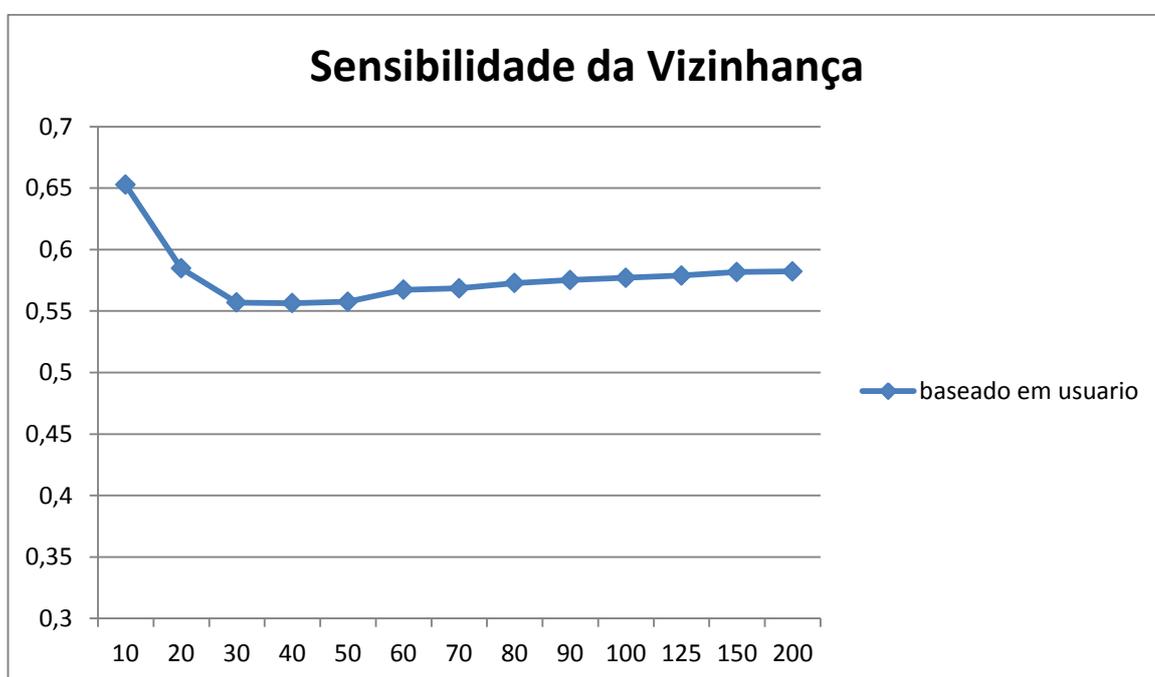


Figura 5 – Sensibilidade do parâmetro de vizinhança.

4.3.3 Experimentos de Qualidade

Com a finalidade de comparar a qualidade das recomendações entre sistemas baseados em usuário e sistemas baseados em itens, novos experimentos comparativos foram realizados. O objetivo foi medir o valor MAE para ambos os sistemas comparativamente para diferentes valores de tamanho de vizinhança. Um sistema usuário-a-usuário (user-user) e um sistema item-a-item (item-item) são sistemas baseados em memória que consideram cada outro usuário, ou item,

respectivamente como potencial vizinho na hora de determinar a vizinhança no processo de recomendação [28]. A experimentação relativa à qualidade das recomendações usa tais sistemas porque eles oferecem a melhor qualidade possível comparado com sistemas baseados em modelos já que usando todos os dados não há perda de informação. A figura 6 mostra o comparativo dos desempenhos dos dois sistemas. Os dados experimentais mostram que o sistema item-a-item tem desempenho superior ao usuário-a-usuário em todos os valores de vizinhança testados.

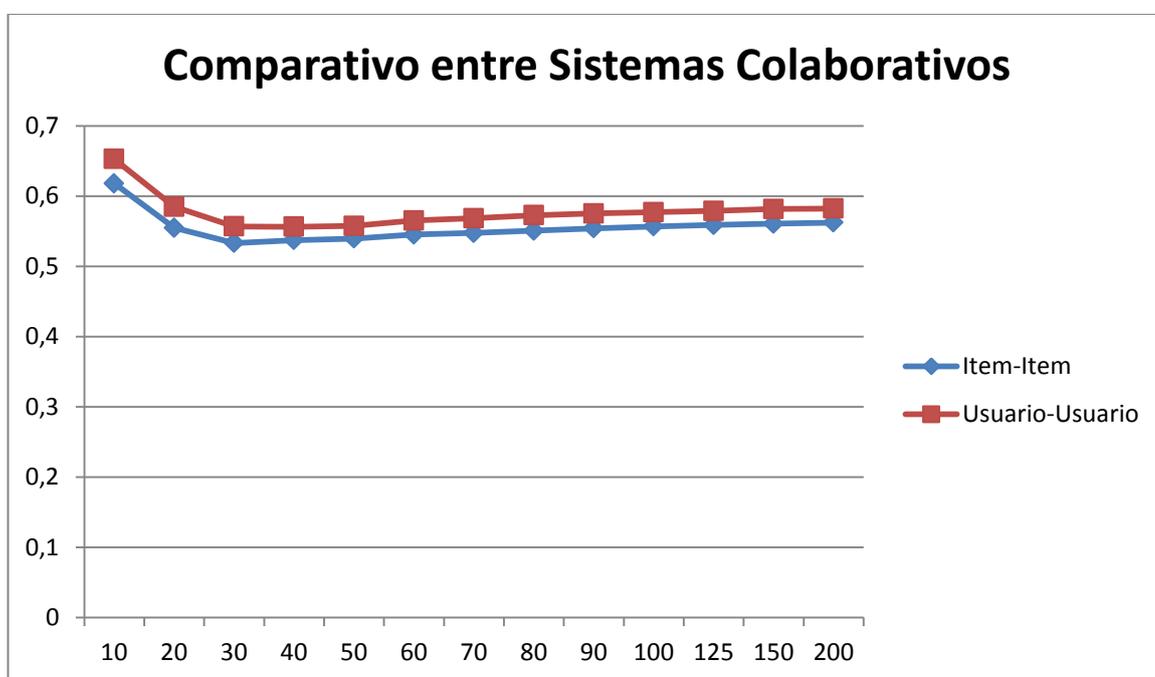


Figura 6 – Comparativo de desempenho entre sistemas baseados em usuários e itens.

4.1 Sensibilidade do Tamanho do Modelo

Sistemas de filtragem colaborativa por item baseado em modelo mantêm apenas as relações entre itens mais relevantes para resolver o problema de recomendação. Um modelo de tamanho k significa considera apenas os k itens mais similares para a construção do modelo depois usa os k itens para a geração de predição [28]. Usando um valor pequeno de k garante que a matriz de similaridade

entre itens é esparsa e pode armazenada na memória, promovendo uma rápida resposta no processo de recomendação [12].

Para determinar o impacto do tamanho do modelo na qualidade das recomendações foram realizados diferentes experimentos variando o valor de k de 25 a 200 com incrementos de 25. Os resultados apresentados na figura 7 mostram que a qualidade das predições aumenta com o tamanho do modelo, i.e., o valor do erro MAE diminui. No entanto a queda do MAE é mais acentuada no intervalo de k entre 25 a 75 e depois diminui de forma discreta. Esse resultado importante mostra que nesses modelos a qualidade da predição é alta mesmo usando uma fração dos itens e com custo computacional menor.

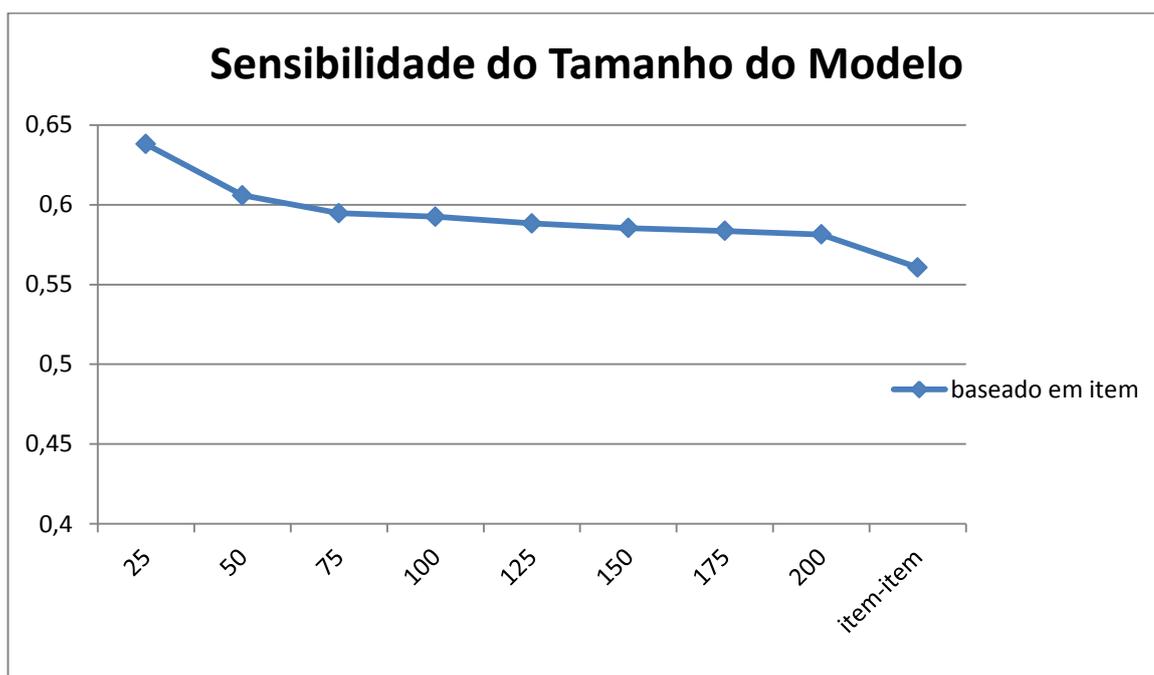


Figura 7 – Sensibilidade do tamanho do modelo na abordagem por item.

4.2 Discussão

Os resultados experimentais obtidos nesse estudo são consistentes com os resultados de outras pesquisas. Pode se observar, primeiramente, que a similaridade tem grande impacto no grau de exatidão das estimativas bem como o tamanho do

parâmetro de vizinhança escolhido. As diferenças entre os dois tipos de sistemas de filtragem colaborativa à cerca da qualidade das recomendações e detalhes de implementação mostram que os sistemas baseados em item se mostram mais adequados a grande demanda dados dos dias atuais. Pois estes são capazes atingir desempenho semelhante ou melhor que os primeiros a um custo computacional menor. Isso se deve em grande parte a observação de que a relação entre os itens é tipicamente estática e permite que ela seja pré-calculada tornando possível se obter alta performance na recomendação online. No tocante a sistemas baseado em item por modelo, proposto por Sarwar et al. [28], constata-se que é possível guardar apenas um subconjunto dos itens e ainda assim obter uma qualidade satisfatória nas predições. Portanto, os sistemas baseado em item são capazes de resolver os dois maiores desafios da área de SR: produzir recomendações de qualidade com alto desempenho e escalabilidade.

5. Conclusões e Trabalhos Futuros

Sistema de Recomendação é uma tecnologia poderosa cada vez mais indispensável às empresas que desejam potencializar seus negócios promovendo experiências, de compra e marketing, por exemplo, diferenciadas para seus clientes. Ao mesmo tempo, ela ajuda a solucionar o problema de excesso de informação filtrando pro ativamente conteúdo que seja relevante para as pessoas que interagem com essa tecnologia. Vimos que a grande variedade de aplicações possíveis, que varia de recomendação de produtos em sites de e-commerce à de recomendação de piadas e rotas de ciclismo, além da riqueza intelectual promovida pelas pesquisas, aquecem a área e a torna instigante.

Os sistemas de recomendação colaborativos, vistos no presente estudo, diferentemente dos sistemas de filtragem por conteúdo que se baseiam nas características dos itens que se deseja recomendar, usam a percepção da inteligência coletiva [2] como estratégia eficaz para obter recomendação. Observamos que os sistemas colaborativos baseados em item, em especial o baseado em modelo, são mais preparados para suportar a crescente demanda de dados garantindo ao mesmo tempo alta qualidade das recomendações e alto desempenho.

Embora as métricas de desempenho estudadas sejam largamente utilizadas para avaliar a precisão e qualidade dos sistemas de recomendação, é preciso investigar outras outros métodos e critérios [1, 10]. Sugerir a um cliente de supermercado que ele compre pão só porque ele faz isso todas as vezes que vai ao supermercado não é de grande valia embora essa abordagem certamente demonstraria uma alta acurácia [1, 37]. É esperado que as recomendações tenham certo nível de novidade, surpresa, descoberta e métricas específicas para avaliar tais critérios [13, 35, 37]. Diversas pesquisas se aprofundam nessas questões propondo métricas alternativas para avaliar a qualidade dos sistemas, e.g., métricas orientadas à economia estipulam os sistemas com relação ao retorno do investimento (ROI) e valor agregado ao cliente [1]. Estes ainda são ativos na área de pesquisa assim como

pesquisas para propor novas formas e algoritmos para computar similaridade de forma mais eficiente [36]. Por se tratar de uma área ampla e multidisciplinar, as pesquisas também exploram outros aspectos importantes como explicabilidade, confiabilidade e privacidade [1].

Os sistemas estudados recomendam uma lista ordenada com os itens melhores estimados (Top-N). Não obstante esta forma ser útil em diversas aplicações em outras pode não ser o caso. Um sistema de recomendação de música online em *stream* (como o jango.com ou grooveshark.com) não pode tocar as músicas na ordem da mais interessante para a menos, pois corre o risco de perder o interesse dos usuários com o tempo. Uma estratégia diferente se faz necessária como intercalar as melhores músicas em meio a outras.

Diante do exposto, considera-se que os sistemas apresentados são factíveis e adequados para implementação do sistema de recomendação de livros em particular o sistema de filtragem colaborativa por modelo proposto por Sarwar et al. [28]. A razão é a sua simplicidade, escalabilidade e qualidade nas predições mesmo em base de dados esparças.

Em trabalhos futuros uma melhor investigação com relação à melhoria na lista final de recomendação levando em consideração outros critérios além de métricas de precisão estatísticas além de estudos com relação à melhoria da performance geral dos sistemas será estudada.

6. Referências Bibliográficas

- [1] 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.
- [2] ALAG, S.; MACMANUS, R. **Collective intelligence in action**. Manning New York, 2009. ISBN 1933988312.
- [3] BAEZA-YATES, R.; RIBEIRO-NETO, B. **Modern information retrieval**. ACM press New York, 1999.
- [4] BALABANOVIĆ, M.; SHOHAM, Y. Fab: content-based, collaborative recommendation. **Communications of the ACM**, v. 40, n. 3, p. 66-72, 1997.
- [5] BELKIN, N. J.; CROFT, W. B. Information filtering and information retrieval: two sides of the same coin? **Communications of the ACM**, v. 35, n. 12, p. 29-38, 1992.
- [6] BREESE, J. S.; HECKERMAN, D.; KADIE, C. Empirical analysis of predictive algorithms for collaborative filtering. Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, 1998. Morgan Kaufmann Publishers Inc. p.43-52.
- [7] CARENINI, G. User-specific decision-theoretic accuracy metrics for collaborative filtering. Proceedings of the Workshop Beyond Personalization 2005, in conjunction with the International Conference on Intelligent User Interfaces IUI'05, 2005. p.26-30.
- [8] CELMA, O.; LAMERE, P. Music recommendation tutorial. **ISMIR. Vienna, Austria**, 2007.
- [9] CLAYPOOL, M. et al. Combining content-based and collaborative filters in an online newspaper. Proceedings of ACM SIGIR workshop on recommender systems, 1999. Citeseer.
- [10] CREMONESI, P.; KOREN, Y.; TURRIN, R. Performance of recommender algorithms on top-n recommendation tasks. Proceedings of the fourth ACM conference on Recommender systems, 2010. ACM. p.39-46.
- [11] CREMONESI, P. et al. An evaluation methodology for collaborative recommender systems. Automated solutions for Cross Media Content and

- Multi-channel Distribution, 2008. AXMEDIS'08. International Conference on, 2008. IEEE. p. 224-231.
- [12] DESHPANDE, M.; KARYPIS, G. Item-based top-n recommendation algorithms. **ACM Transactions on Information Systems (TOIS)**, v. 22, n. 1, p. 143-177, 2004.
- [13] FLEDER, D.; K. HOSANAGAR (2009). "Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity." *Management science* 55(5): 697-712.
- [14] GOLDBERG, D. et al. Using collaborative filtering to weave an information tapestry. **Communications of the ACM**, v. 35, n. 12, p. 61-70, 1992.
- [15] GOLDBERG, K. et al. Eigentaste: A constant time collaborative filtering algorithm. **Information Retrieval**, v. 4, n. 2, p. 133-151, 2001.
- [16] GROUPLENS, 2002, GroupLens homepage, <http://www.grouplens.org>, acessado em 26/04/2013.
- [17] HERLOCKER, J. L. et al. An algorithmic framework for performing collaborative filtering. Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, 1999. ACM. p.230-237.
- [18] HERLOCKER, J. L.; KONSTAN, J. A.; RIEDL, J. Explaining collaborative filtering recommendations. Proceedings of the 2000 ACM conference on Computer supported cooperative work, 2000. ACM. p.241-250.
- [19] KOREN, Y. Factor in the neighbors: Scalable and accurate collaborative filtering. **ACM Transactions on Knowledge Discovery from Data (TKDD)**, v. 4, n. 1, p. 1, 2010.
- [20] LAWRENCE, S. Context in web search. **IEEE Data engineering bulletin**, v. 23, n. 3, p. 25-32, 2000.
- [21] 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.
- [22] MELVILLE, P.; MOONEY, R. J.; NAGARAJAN, R. Content-boosted collaborative filtering for improved recommendations. Proceedings of the National Conference on Artificial Intelligence, 2002. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999. p.187-192.

- [23] MIZZARO, S. Relevance: The whole history. **Journal of the American society for information science**, v. 48, n. 9, p. 810-832, 1997.
- [24] REFAEILZADEH, P.; TANG, L.; LIU, H. Cross-validation. **Encyclopedia of Database Systems**, v. 3, p. 532-38, 2009.
- [25] RESNICK, P. et al. GroupLens: an open architecture for collaborative filtering of netnews. Proceedings of the 1994 ACM conference on Computer supported cooperative work, 1994. ACM. p.175-186.
- [26] RESNICK, P.; VARIAN, H. R. Recommender systems. **Communications of the ACM**, v. 40, n. 3, p. 56-58, 1997.
- [27] SALTON, G. **Automatic Text Processing: The Transformation, Analysis, and Retrieval of**. Addison-Wesley, 1989. ISBN 0201122278.
- [28] SARWAR, B. et al. Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th international conference on World Wide Web, 2001. ACM. p.285-295.
- [29] SCHAFER, J. B. et al. Collaborative filtering recommender systems. In: (Ed.). **The adaptive web**: Springer, 2007. p.291-324. ISBN 3540720782.
- [30] SCHAFER, J. B.; KONSTAN, J. A.; RIEDL, J. E-commerce recommendation applications. In: (Ed.). **Applications of Data Mining to Electronic Commerce**: Springer, 2001. p.115-153. ISBN 1461356482.
- [31] SCHEIN, A. I. et al. Methods and metrics for cold-start recommendations. Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, 2002. ACM. p.253-260.
- [32] SEGARAN, T. **Programming collective intelligence: building smart web 2.0 applications**. O'Reilly Media, 2008. ISBN 0596550685.
- [33] SHARDANAND, U.; MAES, P. Social information filtering: algorithms for automating "word of mouth". Proceedings of the SIGCHI conference on Human factors in computing systems, 1995. ACM Press/Addison-Wesley Publishing Co. p.210-217.
- [34] TAKÁCS, G. et al. Scalable collaborative filtering approaches for large recommender systems. **The Journal of Machine Learning Research**, v. 10, p. 623-656, 2009.
- [35] VARGAS, S. New approaches to diversity and novelty in recommender systems. Fourth BCS-IRSG symposium on future directions in information

access (FDIA 2011), Koblenz, 2011.

- [36] WANG, J.; DE VRIES, A. P.; REINDERS, M. J. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, 2006. ACM. p.501-508.
- [37] ZIEGLER, C.-N. et al. Improving recommendation lists through topic diversification. Proceedings of the 14th international conference on World Wide Web, 2005. ACM. p.22-32.