Antônio Guilherme Ferreira Viggiano
Fernando Fochi Silveira Araújo

Desenvolvimento de um Sistema de Recomendação para E-Commerces

São Paulo, Brasil

3 de novembro de 2014

Antônio Guilherme Ferreira Viggiano
Fernando Fochi Silveira Araújo

Desenvolvimento de um Sistema de Recomendação para E-Commerces

Trabalho de Conclusão de Curso apresentado ao Departamento de Engenharia Mecatrônica da Escola Politécnica da Universidade de São Paulo com requisito parcial para obtenção do Grau de Engenheiro Mecatrônico.

Universidade de São Paulo

Escola Politécnica

Trabalho de Conclusão de Curso

Orientador: Prof. Dr. Fábio Gagliardi Cozman

São Paulo, Brasil

3 de novembro de 2014

Antônio Guilherme Ferreira Viggiano
Fernando Fochi Silveira Araújo
Desenvolvimento de um Sistema de Recomendação para E-Commerces/ Antônio Guilherme Ferreira Viggiano
Fernando Fochi Silveira Araújo. – São Paulo, Brasil, 3 de novembro de 2014-
59 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Fábio Gagliardi Cozman

Trabalho de Conclusão de Curso – Universidade de São Paulo
Escola Politécnica
Trabalho de Conclusão de Curso, 3 de novembro de 2014.

1. Sistema de recomendação. 2. E-Commerce. I. Prof. Dr. Fábio Gagliardi Cozman. II. Universidade de São Paulo. III. Escola Politécnica. IV. Desenvolcimento de um Sistema de Recomendação para E-Commerce

CDU xx.xxx.xxx.x

Antônio Guilherme Ferreira Viggiano
Fernando Fochi Silveira Araújo

Desenvolvimento de um Sistema de Recomendação para E-Commerces

Trabalho de Conclusão de Curso apresentado ao Departamento de Engenharia Mecatrônica da Escola Politécnica da Universidade de São Paulo com requisito parcial para obtenção do Grau de Engenheiro Mecatrônico.

Prof. Dr. Fábio Gagliardi Cozman
Orientador

Prof. Dr. Arturo Forner Cordero
Convidado 1

Profa. Dra. Larissa Driemeier
Convidado 2

Prof. Dr. Lucas Antonio Moscato
Convidado 3

Prof. Dr. Thiago de Castro Martins
Convidado 4

São Paulo, Brasil
3 de novembro de 2014

Dedicamos este trabalho ao Professor Fábio Cozman, pela orientação e apoio

Agradecimentos

Agradecemos ao professor Thiago Martins e aos demais orientadores das disciplinas PMR2500 e PMR2550 – Projeto de Conclusão do Curso I e II – por terem nos guiado na elaboração da monografia e por terem sempre exigido trabalhos de alta qualidade. Esse papel é fundamental na valorização do diploma de Engenharia Mecatrônica da Escola Politécnica.

Make things as simple as possible, but not simpler (Albert Einstein)

Resumo

Resumo em português

Palavras-chaves: este é o resumo em português

Abstract

This is the english abstract.

Key-words: latex. abntex. text editoration.

Lista de tabelas

Tabela 1 – Atributos a_{if} . . . 27
Tabela 2 – Avaliações r_{ui} . . . 27
Tabela 3 – Avaliações r_{ui} . . . 29
Tabela 4 – Atributos a_{if} . . . 29
Tabela 5 – Avaliações r_{ui} . . . 37
Tabela 6 – Atributos a_{if} . . . 37
Tabela 7 – d_{ij}^{f} . . . 40
Tabela 8 – Medidas de distância entre alguns atributos . . . 40
Tabela 9 – e_{ij} . . . 40
Tabela 10 – w_f . . . 40
Tabela 11 – s_{ij} . . . 42
Tabela 12 – $\hat{\imath}_u$ (FW) . . . 42
Tabela 13 – TF_{uf} . . . 42
Tabela 14 – IDF_f . . . 42
Tabela 15 – w_{uf} . . . 42
Tabela 16 – s_{uv} . . . 42
Tabela 17 – f_{uf} . . . 44
Tabela 18 – ω_{ui} (UP) . . . 44
Tabela 19 – $\hat{\imath}_u$ (UP) . . . 44
Tabela 20 – ω_{ui} (UI) . . . 44
Tabela 21 – $\hat{\imath}_u$ (UI) . . . 44
Tabela 22 – Avaliação de sistemas de predição . . . 45
Tabela 23 – Parâmetros de influência no desempenho dos algoritmos de recomendação 49

Lista de símbolos

k	Número de vizinhos mais próximos
N	Tamanho da lista de recomendação
$\mathcal{U}$	Conjunto de todos os usuários
$\mathcal{I}$	Conjunto de todos os itens
$\mathcal{F}$	Conjunto de todos os atributos dos itens
$\mathcal{C}$	Conjunto de todas as características dos usuários
u, v	Usuários
i, j	Itens
f	Atributos dos itens
c	Características dos usuários
$\mathbf{X}_{M\times N},\ \mathbf{X}$	Matriz de elementos x_{mn}
$\mathbf{x}_N,\ \mathbf{x}$	Vetor de elementos x_n
$\tilde{x}$	Valor ótimo de x
$\hat{x}$	Valor estimado de x
$\lvert\mathcal{X}\rvert$	Número de elementos do conjunto $\mathcal{X}$
$\mathbf{R}, r_{ui}$	Avaliação feita pelo usuário u do item i
$\mathbf{A}, a_{if}$	Atributo f presente no item i
$\mathbf{B}, b_{uc}$	Característica c do usuário u
$\mathbf{S}, s_{ij}, s_{uv}$	Similaridade entre itens i e j ou entre usuários u e v
$\mathbf{W}, w_{uf}$	Correlação ponderada entre usuário u e atributo f
$\mathbf{w}, w_f$	Peso do atributo f

Sumário

1 INTRODUÇÃO 21

2 OBJETIVOS 23

3 ESTADO DA ARTE 25
3.1 Estado da arte dos problemas 25
3.2 Estado da arte das soluções 28
3.3 Desafios científicos e tecnológicos 28
3.4 Soluções propostas 30

4 REQUISITOS 33

5 SÍNTESE DE SOLUÇÕES 37
5.1 Algoritmo baseado na ponderação de atributos (FW) 37
5.2 Algoritmo baseado no perfil de usuários (UP) 39
5.3 Algoritmo baseado na correlação usuário-item (UI) 43

6 AVALIAÇÃO DE DESEMPENHO 45

7 METODOLOGIA 47
7.1 Definição da Necessidade 47
7.2 Definição dos Parâmetros de Sucesso 47
7.3 Síntese de Soluções 47
7.4 Detalhamento da Solução 48
7.5 Modelamento e Simulação 48
7.6 Validação Cruzada 48

8 RESULTADOS 49
8.1 Tamanho da lista de recomendações N 49
8.2 Percentual da base de aprendizado T 52

9 OLD 53
9.1 Primeira etapa do Trabalho de Conclusão de Curso 53
9.2 Segunda etapa do Trabalho de Conclusão de Curso 54

Referências 57

1 Introdução

O comércio on-line se torna cada vez mais importante na vida das pessoas, de forma que a adoção deste método de compra é cada vez mais comum. Estima-se que em 2013 um bilhão de pessoas compraram online (1), gerando uma receita anual de 1,25 trilhão de dólares com expectativas de crescer 17% ao ano até 2017. (2). Para exemplificar, a gigante chinesa Alibaba está se preparando para abrir o seu capital na bolsa de valores americana. Esta oferta pública inicial poderá arrecadar até 25 bilhões de dólares, significando a maior oferta pública inicial do mercado acionário americano de todos os tempos. Isso transformaria a Alibaba o maior varejista online do mundo, com um valor avaliado em 158 bilhões de dólares (3).

Ao analisarmos o mercado brasileiro, percebemos que se trata de um comércio jovem e com bom potencial. Entretanto, percebe-se que o brasileiro não desenvolveu ainda o hábito de se fazer compras pela internet. Apenas 19% dos compradores usam esse serviço semanalmente, enquanto em países mais desenvolvidos estes números chegam a 35% na Alemanha e a 39% no Reino Unido. Outro indicador de que o mercado brasileiro ainda é jovem é que 61% dos consumidores de varejo online utilizaram o serviço pela primeira vez nos últimos 4 anos (4).

Como o mercado de varejo online é novo como um todo, este ainda passará por algumas mudanças drásticas em um curto espaço de tempo. Um dos itens-chave destas mudança é a capacidade de se analisar os dados gerados pelos consumidores. Com estes dados será possível segmentar os clientes mais facilmente e as empresas poderão direcionar suas investidas de forma mais eficiente, chegando ao ponto em que campanhas de marketing e precificação serão totalmente personalizadas (5). Uma das maneiras de se usar estes dados são os sistemas de recomendação.

"Sistemas de recomendação são ferramentas e técnicas de software destinadas a prover sugestões de itens para usuários" (6). O sistema tem o propósito de automatizar o processo de recomendação e auxiliar na tomada de decisão, podendo ser aplicado em diversas áreas da indústria, tais como na indicação de notícias, músicas, relações de amizade ou artigos científicos.

Estes sistemas são utilizados por diversos serviços online e geram um grande impacto quando utilizados corretamente. Em 2012, cerca de 75% dos vídeos assistidos através do site NetFlix foram acessados por meio de recomendações (7). Em 2006, as recomendações representaram 35% dos livros vendidos pela Amazon (8), enquanto em 2007 cerca de 38% das notícias lidas no Google News foram sugeridas por um sistema de recomendação (9).

De modo geral, um sistema de recomendação possui três etapas: a aquisição dos dados de entrada, a determinação das recomendações e finalmente a apresentação dos resultados ao usuário. A aquisição dos dados de entrada pode ser feita tanto de forma automática quanto manual, e em geral utiliza-se um banco de dados para armazenar essas informações. As sugestões são feitas segundo uma estratégia de recomendação determinada a priori, que pode ser fundamentada nas preferências do usuário, nas características dos itens ou em alguma formulação mista. Finalmente, os resultados são apresentados na interface sob variadas formas, como por exemplo em uma lista dos N itens mais relevantes para o usuário.

Conforme o tipo específico de itens recomendados, o design do sistema, a interface homem-máquina e o tipo de técnica de recomendação são construídos a fim de prover sugestões mais adequadas.

Os sistemas de recomendação são destinados primeiramente aos indivíduos que não possuem competência ou experiência suficiente para avaliar o grande número de opções do conjunto total de itens. Dessa forma, a interface homem-máquina é adaptada a cada um dos usuários, de maneira que eles recebam recomendações adequadas ao seu perfil. Essa ideia, amplamente divulgada por um antigo diretor executivo do e-commerce *Amazon.com*, se resume à sua fala de que "se você possui 2 milhões de clientes na web, você precisa ter 2 milhões de lojas na web" (10).

Motivados pela importância econômica crescente de lojas de varejo online, bem como pela possibilidade de criar um conjunto de ferramentas *open source* que possam ser utilizadas abertamente pela comunidade, propomos como Trabalho de Conclusão de Curso o desenvolvimento de um sistema de recomendação de produtos de e-commerces.

A contribuição científica e tecnológica do trabalho para a Engenharia Mecatrônica estão sobretudo nos campos de sistemas de informação, de automação de processos e de inteligência artificial. As competências acadêmicas necessárias para a sua execução envolvem algoritmos e estruturas de dados, aprendizado de máquina e modelagem de bancos de dados. As competências técnicas abrangem programação estatística e orientada a objetos (R ou Java, por exemplo) e em linguagem de consulta estruturada (SQL).

2 Objetivos

O objetivo do presente Trabalho de Conclusão de Curso é o desenvolvimento de um Sistema de Recomendação de produtos para e-commerces, e respectiva análise de desempenho das recomendações propostas.

Serão propostos três diferentes algoritmos de recomendação, e será feita uma avaliação comparativa entre eles. A explicação detalhada dos métodos se encontra no Capítulo 5.

Essa ferramenta tem como finalidade a automatização do processo de sugestão de itens, e pode ser aplicada em diversas áreas da indústria, tais como na indicação de notícias, músicas, relações de amizade ou artigos científicos. No nosso trabalho, o sistema terá como foco a sugestão de produtos de lojas de comércio online que disponham de um histórico de compras dos usuários e das características dos produtos.

A qualidade das recomendações será avaliada quanto a precisão e abrangência. Para os métodos em que se possui uma medida de similaridade entre produtos, será avaliada a distância entre os itens efetivamente comprados pelo cliente e aqueles previstos pelo sistema. Uma descrição detalhada da avaliação do sistema de recomendação está descrita no Capítulo 6.

Por meio de uma validação cruzada, analisaremos a influência dos principais parâmetros do problema na qualidade das recomendações, como o tamanho do banco de dados ou a quantidade de informações de itens e clientes utilizadas na recomendação.

Será discutido o impacto dos principais desafios tecnológicos e científicos dos sistemas de recomendação na nossa proposta de solução, tais como a escalabilidade, a adaptação a novos usuários e a esparsidade dos dados (11).

Ao final, será possível extrair uma validação experimental das diretrizes fundamentais a serem seguidas por e-commerces que desejem desenvolver um sistema de recomendação próprio ou que queiram utilizar o sistema desenvolvido neste trabalho.

3 Estado da Arte

As terminologias *cliente* e *usuário* neste texto serão intercambiáveis e sem distinção semântica, mesmo que na prática essas duas entidades possam ser diferentes. Da mesma forma, *item* e *produto* terão o mesmo significado neste trabalho.

A fim de tornar a formulação mais genérica, também não faremos distinção entre *avaliação positiva* de um item e *compra* de um item. Avaliação positiva é toda avaliação r_{ui} do item i feital pelo usuário u tal que $r_{ui} > M$, e avaliação negativa tal que $r_{ui} \leq M$, sendo M um valor mínimo escolhido a priori, indicador de que o usuário u "gostou" do item i. No caso de um banco de dados sem avaliações dos produtos, será levada em conta a compra dos itens e será admitida avaliação unitária e valor mínimo nulo. Desta forma, os bancos de dados que contenham informações do tipo "usuário u avaliou o item i em $r_{ui} = 3.54 > M$" e aqueles que contenham "usuário u comprou o item i, logo $r_{ui} = 1 > 0$" serão tratados equivalentemente. Vale observar que essa definição difere da Referência (12), em que avaliação positiva é aquela tal que $r_{ui} \geq M$.

3.1 Estado da arte dos problemas

O problema de recomendação pode ser formulado como se segue, adaptado da referência (13), com notação inspirada em (12):

"Seja $\mathcal{U}$ o conjunto de todos os usuários e seja $\mathcal{I}$ o conjunto de todos os itens que podem ser recomendados, tais como livros, filmes ou artigos científicos. Seja ℓ uma função de utilidade, que mede a relevância do produto i para usuário u. Em notação matemática, $\ell : \mathcal{U} \times \mathcal{I} \to \mathcal{R}$, onde $\mathcal{R}$ é um conjunto totalmente ordenado – por exemplo, números inteiros ou números reais dentro de um determinado intervalo, em geral $\{-1, 0, +1\}$ ou $[1, 5]$. O objetivo do sistema de recomendação é determinar o item $\tilde{\imath}_u$ que maximize a utilidade ℓ_{ui} do usuário u."

$$\forall u \in \mathcal{U},\ \tilde{\imath}_u = \arg\max_{i \in \mathcal{I}} \ell_{ui} \tag{3.1}$$

O problema central da recomendação é que "em geral a função ℓ é desconhecida ou não é definida para todo o espaço $\mathcal{U} \times \mathcal{I}$", e portanto determinar $\tilde{\imath}$ através da equação 3.1 é inviável.

Em algumas formulações, "a utilidade é descrita pela avaliação r_{ui} do item i feita pelo usuário u". Neste caso, o sistema de recomendação busca determinar $\hat{r}_{ui}$ que melhor se aproxime de r_{ui}, e a qualidade da recomendação é normalmente descrita pela distância

entre esses dois valores. Em outros sistemas, todavia, a utilidade é descrita diferentemente, de forma que o item com maior valor de $\hat{r}_{ui}$ não é necessariamente recomendado.

Para lidar com o problema da recomendação, existem três grandes grupos de estratégias de sugestão de itens, segundo as referências (13, 14):

- Recomendações baseadas em conteúdo: o usuário recebe sugestões de itens similares àqueles pelos quais ele se interessou no passado;
- Recomendações colaborativas: o usuário recebe sugestões de itens que pessoas com preferências semelhantes gostaram no passado;
- Recomendações híbridas: esses métodos combinam caracteríticas de sistemas colaborativos e baseados em conteúdo. O usuário recebe sugestões de itens compatíveis com seu perfil e de itens do interesse de usuários com perfil similar.

As estratégias de recomendação baseadas em conteúdo exploram os dados dos itens para calcular a sua relevância conforme o perfil do usuário. Suas técnicas de recomendação podem ser classificadas em dois grupos: aquelas baseadas em heurísticas ou memória – fazem a previsão com base em toda a coleção de itens anteriormente classificados pelos usuários – e aquelas baseadas em modelos – utilizam o conjunto de avaliações com o objetivo de descrever a interação entre usuários e itens, tal como em uma regressão linear ou em uma rede Bayesiana.

Na abordagem de sistemas baseados em conteúdo, a recomendação pode ser vista como um problema de aprendizado de máquina, em que o sistema adquire conhecimento sobre o usuário. Muitas vezes é recomendado que o aprendizado seja feito com base no perfil do usuário em uso continuo, ao invés de forçá-lo a responder diversas perguntas demográficas (15) – idade, gênero, classe social, etc. O objetivo é categorizar novas informações baseadas em informações previamente adquiridas e rotuladas como interessantes ou não pelo usuário. Com estas informações em mão, é possível gerar modelos preditivos que evoluem conforme aparecem novas informações.

Em sistemas baseados em conteúdo, os itens a serem recomendados podem possuir diversos atributos e formas de classificação. Em documentos como e-mails, *websites* ou comentários de usuários, os textos não tem estrutura definida e a abordagem mais comum para escolher o melhor item é a mineração de informações. O usuário procura por uma lista de termos desejados e o sistema retorna os textos de maior relevância, tal como é feito em um motor de busca (16). Nesses casos, calcula-se a similaridade entre documentos a partir da importância das palavras ou termos similares, como a TF-IDF ou o classificador Bayesiano (17).

Em bancos de dados relacionais, os itens possuem uma categorização pré-definida, e sua relevância depende das suas características, descritas pela matriz de atributos $\mathbf{A}$. Cada

feature pertence a um conjunto distinto, podendo ser booleano (possui ou não possui), inteiro ou real (preço, data, etc.), ou um coleção finita de valores (marca, modelo, gênero, etc.), como exemplifica a Tabela 1.

Tabela 1 – Atributos a_{if}

	f_1	f_2	f_3	f_4
i_1	1	50	0.8	P
i_2	0	75	0.3	M
i_3	1	30	0.4	G

As recomendações colaborativas, por sua vez, tentam prever a utilidade dos itens para cada cliente com base em itens previamente avaliados por outros usuários. Elas podem ser baseadas em usuários, isto é, na escolha de clientes que possuam avaliações similares de produtos, quanto baseadas em itens, na escolha de produtos avaliados similarmente (18).

Mais formalmente, quando a filtragem colaborativa é baseada em usuários, a utilidade ℓ_{ui} de um item i para um usuário u é estimada com base nas utilidades $\ell_{v_k^u i}$ dos usuários $v_k^u \in \mathcal{U}$ que são "similares" ao usuário u. De maneira análoga, quando baseada em itens, a utilidade ℓ_{ui} é prevista com base nas utilidades $\ell_{uj_k^u}$, dado itens $j_k^u \in \mathcal{I}$ que são "similares" aos itens i.

Na prática, o cálculo das recomendações para sistemas colaborativos é feito a partir da matriz de avaliações $\mathbf{R}$. Isso pode ser exemplificado pela Tabela 2, que possui avaliações de 1 a 5, sendo $M = 2$. Em um sistema usuário-usuário, o cliente u_1 receberia recomendação do item i_4, pois para os itens i_2 e i_3 suas avaliações foram similares às do cliente u_2. Já para um sistema item-item, o usuário u_3 receberia recomendação do item i_3, pois este tem avaliações similares às do item i_2, avaliado positivamente pelo usuário u_3.

Tabela 2 – Avaliações r_{ui}

	i_1	i_2	i_3	i_4
u_1	-	4	3	-
u_2	-	4	3	5
u_3	2	5	-	1

Por fim, as recomendações híbridas combinam aspectos tanto da filtragem colaborativa (baseada em usuários ou em itens) quanto da filtragem baseada em conteúdo, com o objetivo de atingir uma melhor recomendação ou de superar problemas recorrentes nas técnicas individuais, como a esparsidade (*sparsity*) dos dados ou o *cold start* (19).

3.2 Estado da arte das soluções

Do ponto de vista do estado da arte das soluções, as variáveis de interesse estão ligadas do número de usuários no sistema, ao número de itens, à medida de qualidade da recomendação e ao custo computacional (20).

No que se refere à dependência do número de usuários, a filtragem colaborativa baseada em usuários é extremamente efetiva para um baixo número de usuários. A filtragem colaborativa a base de itens é consideravelmente pior para um baixo número de usuários, mas supera todos os outros métodos baseados em memória conforme o número de clientes aumenta.

A dependência do número de itens é, de certa forma, oposta à de usuários: a filtragem colaborativa baseada em itens é extremamente efetiva para poucos itens, enquanto aquela baseada em usuários supera todos os outros métodos baseados em memória para grandes quantidades de itens.

Com relação à medida de qualidade, avaliada a partir da acurácia dos dados, a filtragem baseada em usuários e a baseada em itens mostram uma dependência semelhante. Na análise de menor erro quadrático médio entre o item sugerido e o item efetivamente comprado, todos os métodos de recomendação variam não-linearmente com o número de usuários, itens e acurácia, e de modo geral há um compromisso (*trade-off*) entre a esparsidade (*sparsity*) dos dados e o tempo de processamento.

3.3 Desafios científicos e tecnológicos

Um dos maiores desafios tecnológicos dos sistemas de recomendação é, atualmente, o da escalabilidade (15). O sistema de recomendação deve ser flexível no sentido de poder operar igualmente bem tanto em pequenas quanto em grandes bases de dados, que podem chegar até centenas de milhões de clientes (21) e de produtos (22). Isso significa que as recomendações devem ser suficientemente rápidas e ainda assim prover sugestões valiosas aos consumidores.

Um problema muito comum nos sistemas de recomendação é o do *cold start*, que atinge principalmente os sistemas de filtragem colaborativa, grandemente dependentes da matriz de avaliações $\mathbf{R}$. Quando itens ou usuários são inicialmente introduzidos no sistema, existe pouca ou nenhuma informação sobre eles. O sistema é incapaz de realizar inferências sobre quais itens recomendar ao novo usuário ou sobre quais produtos são similares ao novo item. Na Tabela 3, por exemplo, o item i_{100} não possui nenhuma avaliação, e nunca seria recomendado em sistemas puramente baseados em itens. Analogamente, o usuário u_3 também não teria nenhuma sugestão de itens para sistemas puramente baseados em usuários.

Tabela 3 – Avaliações r_{ui}

	i_1	i_2	$\cdots$	i_{100}
u_1	5	4	$\cdots$	-
u_2	-	2	$\cdots$	-
u_3	-	-	$\cdots$	-

Também é uma grande dificuldade dos algoritmos baseados em filtragem colaborativa a esparsidade dos dados. Como a maioria dos clientes interage com uma pequena quantidade de itens, a matriz de avaliações tem em geral menos de 1% dos valores preenchidos, e o sistema deve prever os outros valores (23).

Outro desafio científico é referente à diversidade das recomendações realizadas, também chamado de excesso de especialização (*over-specialization*) (13). Ao mesmo tempo que o sistema deve apresentar itens similares ao que o usuário está procurando, ele também deve sugerir itens que o usuário desconheça ou que nem saiba que poderiam interessá-lo. Esse problema afeta principalmente os sistemas baseados em conteúdo, pois itens com características similares tendem a ser sempre recomendados. Na Tabela 4, o item i_2 seria sugerido para um usuário que tenha avaliado i_1, apesar de esse não apresentar nenhuma característica diferente do item previamente comprado. Para contornar essa dificuldade, costuma-se introduzir elementos de aleatoriedade na recomendação, por exemplo a partir de algoritmos genéticos (14).

Tabela 4 – Atributos a_{if}

	f_1	f_2	f_3
i_1	1	50	0.8
i_2	1	50	0.8
i_3	0	75	0.3

Além desse desafio, existe também o da análise rasa do conteúdo (*shallow content analysis*). O sistema, ao avaliar a característica dos itens, não consegue extrair importantes aspectos para o usuário caso eles não estejam explicitamente descritos na categorização do banco de dados. Isso pode ocorrer, por exemplo, com fatores externos ao produto que influenciem na compra do usuário, como em datas comemorativas, em compras induzidas por propaganda, em compras "impulsivas", etc. Se um usuário comprou um arranjo de flores no dia das mães, não é necessariamente verdade que ele se interessa por flores. Da mesma maneira, sistemas que ignorem a sazonalidade de certos produtos não recomendariam arranjos de flores para clientes que não tenham comprado itens parecidos, mesmo no dia das mães.

Por fim, um desafio científico que este trabalho enfrentará é a execução de um sistema híbrido do ponto de vista de efemeridade e persistência, ao construir um modelo

de recomendação que integre as preferências de curto e longo termo dos usuários (10). A análise dos dados de compras anteriores, bem como de dados demográficos, deverá portanto ser incorporada à análise de característica dos produtos, a fim de enriquecer a acurácia do sistema (15).

Esse tópico de pesquisa inclui ainda diversos desafios científicos e tecnológicos que não serão tratados no nosso projeto, tais como a preservação da privacidade dos usuários, a criação de modelos de recomendação inter-domínios, o desenvolvimento de sistemas descentralizados operando em redes computacionais distribuídas, a otimização de sistemas para sequências de recomendações, a otimização de sistemas para dispositivos móveis e outros. Um sistema de recomendação inteligente também deveria prever quando enviar uma determinada recomendação, e não agir apenas mediante requisição dos clientes (24).

3.4 Soluções propostas

Este Trabalho de Conclusão de Curso aborda três propostas de solução para o problema da recomendação, sendo duas delas retiradas de referências bibliográficas (12, 25), e uma outra apresentada pela dupla. O objetivo é realizar uma análise comparativa entre cada um dos métodos e estabelecer diretrizes para sua aplicação em e-commerces. Os algoritmos propostos estão descritos com maior detalhe no Capítulo 5.

Todas as soluções são algoritmos híbridos, por utilizarem na recomendação tanto a matriz de avaliações $\mathbf{R}$ quanto a matriz de atributos $\mathbf{A}$. Optou-se por dar importância aos algoritmos híbridos em razão de os e-commerces estruturarem seus bancos de dados em torno da descrição dos itens à venda. De modo geral, as tabelas de itens possuem dezenas de atributos, dependendo do ramo de negócios da loja, e pouco detalhe é dado à interação entre o grupo de usuários e itens. A tabela de histórico de compras se limita a informações como data e método de pagamento, e detém pouca informação adicional que possa ser utilizada na recomendação de produtos. Dessa forma supusemos que métodos puramente colaborativos, fundamentados na avaliação dos itens por parte dos usuários, teriam pior desempenho que métodos baseados em conteúdo, que exploram as características dos itens na recomendação.

A solução *FW* determina a similaridade de dois itens a partir de medidas de distância para cada um dos atributos dos itens, ponderadas por pesos determinados na regressão linear de uma equação descrita pelo interesse dos usuários em cada *feature*.

O método *UP* parte do princípio que os usuários estão interessados nos atributos dos itens, e traça correlações entre esses dois elementos para obter pesos que servirão de base para o cálculo da similaridade inter-usuários, utilizada na recomendação pelo método da vizinhança (*nearest neighbors*).

A variante *UI*, elaborada pela dupla, recomenda o melhor item a partir das matrizes de correlação usuário-atributo e atributo-item, a fim de obter a matriz de correlação usuário-item. Espera-se que essa solução tenha desempenho similar ao método de base *UP*, pois ambos buscam explorar as características dos itens para determinar a preferência do usuário.

4 Requisitos

A partir dos objetivos deste Trabalho de Conclusão de Curso, é possível extrair os requisitos funcionais do sistema de recomendação. Esses requisitos ditam principalmente sobre a escalabilidade e o desempenho das recomendações do sistema.

Como as sugestões serão calculadas com antecedência, não há necessidade para uma elevada taxa de recomendações por período de tempo (*throughput*). Deseja-se contudo que o sistema possa gerar todas as recomendações para um banco de dados de cem mil clientes em uma hora, isto é, que tenha *throughput* mínimo de 28 recomendação por segundo. Os sistemas de recomendação tradicionais possuem *throughput* de cerca de 500 recomendações por segundo, mas operam em servidores dedicados de maior potência computacional (26).

A fim de poder estabelecer uma base comparativa entre o sistema proposto *UI* e os sistemas de referência *FW* e *UP*, serão utilizados os mesmos indicadores de desempenho dos artigos-base: precisão, abrangência e medida F_1 (12, 25). Precisão é a porcentagem de casos corretamente preditos em relação ao tamanho da lista de recomendações. Abrangência é a razão entre o número de itens corretamente preditos e daqueles que foram efetivamente avaliados pelo usuário. A medida F_1, por sua vez, é a média harmônica entre precisão e abrangência.

Todas essas métricas são dependentes dos diversos parâmetros do problema, como do tamanho da lista de recomendações N, da quantidade de vizinhos mais próximos k, e principalmente do banco de dados de teste. Como os artigos de referência não os disponibilizaram integralmente, serão estimados os valores de precisão, abrangência e medida F_1 para o banco de dados da dupla.

Espera-se que a precisão, abrangência e consequentemente a medida F_1 sejam maiores que 20%. Esses valores foram escolhidos por serem superiores aos de algoritmos puramente baseados em conteúdo ou em filtragem colaborativa (12, 25). Na prática, o resultado mais importante é a comparação entre os três métodos para um banco de dados de referência.

Neste trabalho o *benchmark* é feito por meio de dois bancos amplamente utilizados na comunidade científica de Sistemas de Recomendação. O primeiro, denominado MovieLens 100k, é composto de 100 000 avaliações (valores inteiros de 1 a 5) de 943 usuários para 1682 filmes (27). Além disso, cada usuário (idade, sexo, profissão, logradouro) avaliou pelo menos 20 filmes (categoria, ano de publicação). O segundo banco de dados é extraído do Internet Movie Database (IMDB), e possui 28 819 filmes. Esse banco está presente na biblioteca `ggplot2` da linguagem de programação R (28).

Os requisitos funcionais são suportados por requisitos não-funcionais, e estes são determinados pelas restrições sobre o projeto ou execução, tais como desenvolvimento e confiabilidade.

O sistema de recomendação deverá poder ser utilizado por qualquer e-commerce que disponha de um banco de dados de clientes, produtos e histórico de compras, desde que o formato de entrada, a ser especificado no Capítulo 8, seja seguido.

Além disso o sistema deverá ser desenvolvido em tecnologias abertas (*open source*) que tenham um alto número de colaboradores, como o sistema de gestão de banco de dados MySQL ou a linguagem de programação estatística R, a fim de torná-lo reutilizável por alunos ou e-commerces interessados.

Por fim, o sistema de recomendação deverá ser escalável e flexível no sentido de poder operar igualmente bem tanto em pequenas quanto em grandes bases de dados.

Apesar serem importantes parâmetros de um sistema de recomendação, a taxa de recomendações por período de tempo e a escalabilidade estão intimamente relacionados ao orçamento do projeto. Pode-se obter virtualmente qualquer *throughput* desejado, contanto que haja investimento equivalente em infra-estrutura computacional. O mesmo não é válido para os parâmetros de qualidade da recomendação, que dependem tão somente dos algoritmos de sugestão. Neste trabalho, assumimos que o sistema de operará em microcomputadores pessoais, e por isso o requisito funcional *throughput* se faz necessário.

Com os requisitos do sistema de recomendação definidos, devemos estruturar o seu relacionamento com o administrador do sistema. Para isto determinamos as ações que o administrador pode realizar com este sistema. Cada uma destas ações é um caso de uso, e ao total foram criados nove casos de uso para este sistema.

O primeiro caso de uso é o caso *Selecionar Método*, onde o administrador do sistema pode selecionar entre qual método de recomendação será utilizado para gerar as recomendações do sistema.

O segundo caso é o *Selecionar Banco de Dados*, pelo qual é permitido ao administrador selecionar o banco de dados para o qual serão geradas as recomendações.

O terceiro caso *Gerar Recomendações* é o caso onde o dá ínicio aos cálculos das recomendações com o método e o banco de dados préviamente escolhidos, o que caracteriza dependência indicada na figura.

O quarto caso de uso, *Escolher Cliente*, permite ao administrador selecionar o cliente para o qual quer ter acesso às recomendações. Isto que nos leva ao quinto caso de uso, *Ranquear Itens Para Cliente*,o qual ordenas os itens do banco de dados de forma decrescente de acordo com o valor da recomendação para o cliente escolhido.

O sexto caso, *Avaliar Recomendações*, permite ao usuário avaliar a precisão de a

acurácia do sistema. Este, em conjunto com o *Fazer Validação Cruzada*, servirá para avaliar se o método utilizado para gerar as recomendações é indicado para ser implementado no sistema de vendas do e-commerce.

O caso de uso *Devolver Banco de Recomendações*, é o principal output do sistema, onde se devolve um banco de dados com todas as precisões de avaliações de cada um dos itens por cada um dos usuários.

O último caso de uso, *Devolver Pesos dos Atributos*, retornará o peso que cada atributo dos itens tem. Assim definindo quais atributos são mais importantes para cada usuário.

5 Síntese de Soluções

A fim de facilitar a compreensão dos métodos propostos neste trabalho, serão utilizadas as matrizes de avaliações **R** e de atributos **A** abaixo, adaptadas da Referência 25. Em todos os exemplos, considera-se valor mínimo $M = 2$. Os logaritmos são expressos em base 10 e todos os pesos w_f (descritos a seguir) são utilizados.

Tabela 5 – Avaliações r_{ui}

	i_1	i_2	i_3	i_4	i_5	i_6
u_1	-	4	-	-	5	-
u_2	-	3	-	4	-	-
u_3	-	-	-	-	-	4
u_4	5	-	3	-	-	-

Tabela 6 – Atributos a_{if}

	f_1	f_2	f_3	f_4
i_1	0	1	0	0
i_2	1	1	0	0
i_3	0	1	1	0
i_4	0	1	0	0
i_5	1	1	1	0
i_6	0	0	0	1

5.1 Algoritmo baseado na ponderação de atributos (FW)

O primeiro algoritmo que utilizaremos no sistema de recomendação, adaptado da Referência 12 e denominado ponderação de atributos, *feature weighting* ou *FW*, trata-se de um híbrido entre filtragem colaborativa e filtragem baseada em conteúdo. A partir da regressão linear de dados de uma rede social (*Internet Movie Database, IMDB*), extraem-se os pesos que determinam a importância de cada atributo dos itens, e é onde ocorre a filtragem colaborativa dos usuários. Após obtenção dos pesos, realiza-se a filtragem baseada em conteúdo para determinar os itens com maior similaridade, que são finalmente recomendados.

Na filtragem baseada em conteúdo, "cada item é representado por um vetor de atributos ou *features*". A similaridade s_{ij} entre dois itens i e j é dada pela média ponderada

das distâncias entre as *features* dos itens:

$$s_{ij} = \sum_f w_f \left(1 - d_{fij}\right) \tag{5.1}$$

As distâncias entre os atributos d_f são determinadas conforme o tipo de dado avaliado e seu domínio, normalizadas no intervalo $[0, 1]$.

Para atributos literais, como categoria, marca, cor, etc., uma possível medida de distância é o delta de Kronecker descrito em 5.2. A similaridade entre as cores "azul" e "vermelho" é, nesse caso, 0, e sua distância é 1. O valor da distância é nulo se e somente se os atributos são idênticos.

Para atributos pertencentes a uma coleção finita de itens, tais como os atores participantes de um filme, é possível estabelecer a similaridade entre dois conjuntos a partir do índice Jaccard, descrito em 5.3. Neste caso, a similaridade entre os conjuntos {Al Pacino, Tom Hanks} e {Tom Hanks, Marlon Brando} é 1/3, e a sua distância é 2/3.

$$\delta_{mn} = \begin{cases} 1, & \text{se } m = n \\ 0, & \text{se } m \neq n \end{cases} \tag{5.2}$$

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{5.3}$$

Vale considerar a correlação entre atributos no cálculo das distâncias: a similaridade de duas marcas de calçado, por exemplo, é maior que a de duas marcas de produtos de categorias diferentes, mesmo que as marcas sejam distintas nos dois casos. Em uma primeira análise, todavia, utilizaremos para a maior parte das *features* as medidas de distância do delta de Kronecker 5.4 (Tabela 7) e do índice Jaccard 5.5. Isso significa que se os atributos de dois itens são idênticos, a distância é nula e portanto a similaridade é máxima. O sumário de algumas medidas de distância que podem ser utilizadas estão na Tabela 8.

$$\begin{aligned} d_{fij} &= 1 - \delta_{ij}^f \\ &= 1 - \delta_{a_{if} a_{jf}} \end{aligned} \tag{5.4}$$

$$\begin{aligned} d_{fij} &= 1 - J^f(i, j) \\ &= 1 - J(a_{if}, a_{jf}) \end{aligned} \tag{5.5}$$

Os pesos w_f são a priori desconhecidos. A Referência 12 os determina a partir de uma regressão linear do tipo 5.6, onde e_{ij} é o número de usuários que se interessam

tanto por i quanto por j. Esses valores permitem determinar "o julgamento humano de similaridade entre itens", e pode ser calculado a partir da matriz de avaliações, conforme a equação 5.7 (Tabela 9). O operador booleano b_M, descrito pela Equação 5.8, nada mais é que uma ferramenta matemática para se poder extrair o número de usuários que avaliaram *positivamente* tanto i quanto j a partir de $\mathbf{R}$.

$$e_{ij} = w_0 + \sum_f w_f \left(1 - d_{fij}\right) \tag{5.6}$$

$$e_{ij} = \sum_u \mathrm{b_M}\left(r_{ui}\ r_{uj}\right) \tag{5.7}$$

$$\mathrm{b}_M\left(x\right) = \begin{cases} 1, & \text{se } x > M \\ 0, & \text{se } x \leq M \end{cases} \tag{5.8}$$

Desta forma, os pesos w_f são determinados a partir resolução do sistema de equações lineares 5.9 (Tabela 10). Apenas os pesos positivos e com valor absoluto expressivo (maior que um piso arbitrariamente escolhido a posteriori) são utilizados na recomendação.

$$w_0 + \sum_f w_f \left(1 - d_{fij}\right) = \sum_u \mathrm{b}_0\left(r_{ui}\ r_{uj}\right),\ \forall i \neq j \tag{5.9}$$

Calcula-se a matriz de similaridade $\mathbf{S}$ pela Equação 5.1 (Tabela 11) e recomendam-se os itens similares àqueles já comprados, segundo 5.10 (Tabela 12).

$$\hat{\imath}_u = \underset{i \in \{i \mid r_{ui} > 0\}, j}{\arg\max}\ s_{ij} \tag{5.10}$$

5.2 Algoritmo baseado no perfil de usuários (UP)

O segundo algoritmo, adaptado da Referência 25, é um hibrido entre filtragem colaborativa e filtragem baseada em conteúdo. Os atributos dos itens são ponderados no cálculo de similaridade, com pesos extraídos de um modelo de perfil de usuários, denominado *user profile* ou *UP*. Esse perfil leva em consideração o interesse dos usuários por *features*, indiretamente calculado a partir de seu interesse pelos itens.

Para se determinar a relevância de f para u, deve-se levar em conta não somente a frequência com a qual uma característica aparece, mas também o fato de algumas características estarem contidas na maioria dos itens. Determina-se, então, os pesos w_{uf}, que mostram a relevância de f para u, a partir da medida estatística TF-IDF (*term frequency–inverse document frequency*), presente em formulações de recuperação de informação e mineração de dados (Equação 5.13).

Tabela 7 – d_{ij}^{f}

f_1	i_1	i_2	i_3	i_4	i_5	i_6
i_1	-	0	1	1	0	1
i_2	0	-	0	0	1	0
i_3	1	0	-	1	0	1
i_4	1	0	1	-	0	1
i_5	0	1	0	0	-	0
i_6	1	0	1	1	0	-

f_2	i_1	i_2	i_3	i_4	i_5	i_6
i_1	-	1	1	1	1	0
i_2	1	-	1	1	1	0
i_3	1	1	-	1	1	0
i_4	1	1	1	-	1	0
i_5	1	1	1	1	-	0
i_6	0	0	0	0	0	-

f_3	i_1	i_2	i_3	i_4	i_5	i_6
i_1	-	1	0	1	0	1
i_2	1	-	0	1	0	1
i_3	0	0	-	0	1	0
i_4	1	1	0	-	0	1
i_5	0	0	1	0	-	0
i_6	1	1	0	1	0	-

f_4	i_1	i_2	i_3	i_4	i_5	i_6
i_1	-	1	1	1	1	0
i_2	1	-	1	1	1	0
i_3	1	1	-	1	1	0
i_4	1	1	1	-	1	0
i_5	1	1	1	1	-	0
i_6	0	0	0	0	0	-

Tabela 8 – Medidas de distância entre alguns atributos

Atributo f	**Domínio** F	**Distância** d_f
Marca	Literal	$1 - \delta_{ij}^{f}$
Esporte	Literal	$1 - \delta_{ij}^{f}$
Gênero	Literal	$1 - \delta_{ij}^{f}$
Categoria	Conjunto Literal	$1 - J^{f}(i,j)$
Preço	$\mathbb{R}$	$\frac{\lvert a_{if} - a_{jf} \rvert}{\max_{i,j} \lvert a_{if} - a_{jf} \rvert}$
Data	$\mathbb{R}$ milissegundos a partir de *epoch* (29)	$\frac{\lvert a_{if} - a_{jf} \rvert}{\max_{i,j} \lvert a_{if} - a_{jf} \rvert}$

Tabela 9 – e_{ij}

	i_1	i_2	i_3	i_4	i_5	i_6
i_1	-	0	1	0	0	0
i_2	0	-	0	1	1	0
i_3	1	0	-	0	0	0
i_4	0	1	0	-	0	0
i_5	0	1	0	0	-	0
i_6	0	0	0	0	0	-

Tabela 10 – w_f

w_0	w_1	w_2	w_3	w_4
0.41	-0.22	-0.34	-0.03	-

Em nosso caso, TF ou *feature frequency* é a "similaridade intra-usuários", igual ao número de vezes em que a *feature* f aparece no perfil do usuário u (Equação 5.11, Tabela 13). Se o usuário avaliou *positivamente* algum item r_{ui}, tal que r_{ui} é superior a um valor mínimo M, considera-se que u tem interesse TF_{uf} nos atributos f dos itens i, representados por a_{if}.

$$\mathrm{TF}_{uf} = \sum_{i} \mathrm{b}_M \left(r_{ui}\ a_{if}\right) \tag{5.11}$$

O termo IDF ou *inverse user frequency* é a "dissimilaridade inter-usuários", relacionada com o inverso da frequência de um atributo f dentro de todos os usuários (Equação 5.12, Tabela 14).

$$\mathrm{IDF}_f = \log\left(\frac{\mid \mathcal{U} \mid}{\sum_u \mathrm{b}_0 \left(\mathrm{TF}_{uf}\right)}\right) \tag{5.12}$$

Os pesos w_{uf}, obtidos na TF-IDF 5.13 (Tabela 15), são utilizados para calcular a similaridade s_{uv} entre dois usuários u e v, conforme as Equações 5.14 e 5.15 (Tabela 16).

$$w_{uf} = \mathrm{TF}_{uf}\ \mathrm{IDF}_f \tag{5.13}$$

$$s_{uv} = \frac{\sum\limits_{f \in \mathcal{F}_{uv}} w_{uf}\ w_{vf}}{\sqrt{\sum\limits_{f \in \mathcal{F}_{uv}} w_{uf}^2}\sqrt{\sum\limits_{f \in \mathcal{F}_{uv}} w_{vf}^2}} \tag{5.14}$$

$$\begin{aligned} \mathcal{F}_{uv} &= \mathcal{F}_u \cap \mathcal{F}_v \\ \mathcal{F}_u &= \{f \in \mathcal{F} \mid t_{uf} > 0\} \end{aligned} \tag{5.15}$$

Dispondo-se de $\mathbf{S}$, selecionam-se os k vizinhos mais próximos v_k^u com maior similaridade s_{uv}, dentre todos $v \neq u$. Posteriormente, determina-se o conjunto $\mathcal{I}_{v_k^u} = \left\{i \mid r_{v_k^u i} > M\right\}$ de itens i avaliados positivamente por v_k^u. Em 5.16 avalia-se a frequência total f_{uf} dos atributos f para os itens de $\mathcal{I}_{v_k^u}$ (Tabela 17).

$$\mathrm{f}_{uf} = \sum_{i \in \mathcal{I}_{v_k^u}} \mathrm{b}_0 \left(a_{if}\right) \tag{5.16}$$

Por fim, a partir da Equação 5.17 calcula-se o peso ω_{ui} (Tabela 18) de cada item e gera-se a lista dos *top-N* produtos a serem recomendados para o usuário u, conforme 5.18 (Tabela 19).

$$\omega_{ui} = \sum_{f} a_{if}\ \mathrm{f}_{uf} \tag{5.17}$$

Tabela 11 – s_{ij}

	i_1	i_2	i_3	i_4	i_5	i_6
i_1	-	0.44	1.00	0.93	0.51	0.17
i_2	0.44	-	0.51	0.44	1	-0.32
i_3	1.00	0.51	-	1.00	0.44	0.24
i_4	0.93	0.44	1.00	-	0.51	0.17
i_5	0.51	1.00	0.44	0.51	-	-0.25
i_6	0.17	-0.33	0.24	0.17	-0.25	-

Tabela 12 – $\hat{\imath}_u$ (FW)

u_1	u_2	u_3	u_4
3	5	3	4

Tabela 13 – TF_{uf}

	f_1	f_2	f_3	f_4
u_1	2	2	1	0
u_2	1	2	0	0
u_3	0	0	0	1
u_4	0	2	1	0

Tabela 14 – IDF_f

f_1	f_2	f_3	f_4
0.30	0.12	0.30	0.60

Tabela 15 – w_{uf}

	f_1	f_2	f_3	f_4
u_1	0.60	0.25	0.30	0
u_2	0.30	0.25	0	0
u_3	0	0	0	0.60
u_4	0	0.25	0.30	0

Tabela 16 – s_{uv}

	u_1	u_2	u_3	u_4
u_1	-	0.96	0	1
u_2	0.96	-	0	1
u_3	0	0	-	0
u_4	1	1	0	-

$$\hat{\imath}_u = \underset{i \in \{i \mid r_{ui} = 0\}}{\arg\max} \ \omega_{ui} \tag{5.18}$$

5.3 Algoritmo baseado na correlação usuário-item (UI)

Este método se trata de uma variante da solução *UP*, e também está embasado no cálculo da preferência do usuário por *features*, medida através do seu interesse pelos itens. O algoritmo *UI* utiliza as matrizes de correlação ponderada entre usuários e atributos **W** e a matriz de atributos dos itens **A** no cálculo da correlação usuário-item.

A lista dos N produtos a serem recomendados decorre portanto do cálculo de ω_{ui} (Equação 5.19, Tabela 20) e da escolha dos itens que maximizem essa variável para cada usuário (Equação 5.18, Tabela 21).

$$\omega_{ui} = \sum_f w_{uf} \ a_{if} \tag{5.19}$$

Ao passo que o método *UP* recomenda itens a partir dos k vizinhos mais próximos, o algoritmo *UI* busca os itens com *features* mais similares aos atributos pelos quais u se interessa, diretamente através da matriz de atributos.

Espera-se que esse tipo de recomendação forneça sugestões de qualidade similar ao algoritmo original, pois os dois tem a mesma fundamentação inicial. Pode-se observar que, para o exemplo-base, ambos algoritmos forneceram a mesma recomendação para três de quatro usuários.

Tabela 17 – f_{uf}

	f_1	f_2	f_3	f_4
u_1	0	2	1	0
u_2	1	3	2	0
u_3	1	2	0	0
u_4	2	3	1	0

Tabela 18 – ω_{ui} (UP)

	i_1	i_2	i_3	i_4	i_5	i_6
u_1	2	0	3	0	0	0
u_2	3	0	5	0	6	0
u_3	0	3	0	2	0	0
u_4	0	5	0	3	6	0

Tabela 19 – $\hat{\imath}_u$ (UP)

u_1	u_2	u_3	u_4
3	5	2	5

Tabela 20 – ω_{ui} (UI)

	i_1	i_2	i_3	i_4	i_5	i_6
u_1	0.25	0.85	0.55	0.25	1.15	0
u_2	0.25	0.55	0.25	0.25	0.55	0
u_3	0	0	0	0	0	0.60
u_4	0.25	0.25	0.55	0.25	0.55	0

Tabela 21 – $\hat{\imath}_u$ (UI)

u_1	u_2	u_3	u_4
3	5	-	5

6 Avaliação de Desempenho

De modo geral os sistemas de recomendação tem o objetivo de apresentar ao usuário itens pelos quais ele possa se interessar. O desempenho de um sistema de recomendação se mede, portanto, na qualidade com a qual ele executa essa tarefa.

Essa qualidade pode ser medida de diferentes maneiras, tal como pela medida de distância entre os produtos recomendados $\hat{\mathbf{i}}$ e aqueles que seriam efetivamente comprados $\mathbf{i}$ pelo cliente em uma validação cruzada (*cross validation*). Outras medidas de predição também podem ser utilizadas, a exemplo de trabalhos de recuperação de informação, tais como acurácia (*accuracy*), especificidade (*specificity*), precisão (*precision*), abrangência (*recall*), medida F_1 (F_1-*score*), e outras.

No nosso Trabalho de Conclusão de Curso, serão utilizados precisão, abrangência, e medida F_1. Essas medidas foram escolhidas a fim de se poder estabelecer uma base comparativa com os textos de referência, que também as utilizam. Elas estão sumarizadas na Tabela 22. As quantidades VP, FP, VN e FN significam o número de verdadeiro e falso positivos e o número de verdadeiro e falso negativos.

Tabela 22 – Avaliação de sistemas de predição

Medida	**Fórmula**	**Significado**
Precisão	$\frac{VP}{VP+FP}$	Porcentagem de casos positivos corretamente preditos.
Abrangência	$\frac{VP}{VP+FN}$	Porcentagem de casos positivos sobre aqueles que foram marcados como positivos.
F_1	$2 \cdot \frac{\text{Precisão} \cdot \text{Abrangência}}{\text{Precisão} + \text{Abrangência}}$	Média harmônica entre precisão e abrangência.

Por fim, avaliaremos o desempenho do sistema mediante a mudança nas variáveis de importância do problema, como por exemplo na quantidade de atributos utilizados na recomendação. O tempo de execução também será avaliado em função do algoritmo utilizado e do tamanho do banco de dados.

7 Metodologia

A metodologia de projeto deste Trabalho de Conclusão de Curso foi fundamentada principalmente na Referência 30. Por se tratar de um projeto de Engenharia de Software, foi necessário dar ênfase às etapas iterativas de desenvolvimento dos algoritmos. Esse processo cíclico, com fases de especificação, desenvolvimento e validação, permitiu obter resultados preliminares e os modificar os algoritmos ao longo da disciplina, ajustando detalhes e melhorando o sistema gradativamente (31).

A metodologia de execução do projeto, assim como a de avaliação dos resultados, pode ser consolidada da seguinte maneira:

7.1 Definição da Necessidade

Com o crescente número de lojas de comércio online, tornou-se necessário a criação de sistemas que pudessem entender e prever o comportamento de consumidores, a fim de oferecer produtos específicos para cada um deles, aumentando o número de vendas e a satisfação do cliente. Observa-se atualmente que o número de sistemas de recomendação gratuitos, de fácil integração e de código aberto (*open source*) são limitados e não correspondem às necessidades do mercado. Existe, pois, a necessidade da criação de um sistema que possa ser utilizado por e-commerces que desejem estabelecer seu próprio sistema de recomendação ou mesmo por indivíduos interessados na temática da recomendação de itens.

7.2 Definição dos Parâmetros de Sucesso

O sucesso do projeto pode ser medido em duas frentes: a primeira, quantitativa, mede a precisão e a abrangência das recomendações; a segunda, qualitativa, avalia se o sistema responde bem aos problemas recorrentes desse tópico de pesquisa, tais como a escalabilidade, o excesso de especialização e outros.

7.3 Síntese de Soluções

Nesta fase do projeto, foram propostas possíveis soluções para o desafio da recomendação. Decidiu-se avaliar dois métodos híbridos do meio acadêmico e um outro elaborado pela dupla.

7.4 Detalhamento da Solução

Após a escolha dos métodos de recomendação, as soluções foram detalhadas matematicamente segundo uma mesma notação, e a estrutura dos algoritmos foi descrita e exemplificada. Neste ponto, escolheu-se também a linguagem de programação (R) e a forma de entrada e saída de dados (arquivos `.csv`).

7.5 Modelamento e Simulação

Os métodos escolhidos foram codificados em R e testados com inicialmente com o banco de dados 100k. Posteriormente, testamos os algoritmos no banco IMDB, a fim de avaliar a qualidade das recomendações mediante a mudanças na base de dados.

7.6 Validação Cruzada

A fim de realizar um estudo comparativo (*benchmarking*) com os artigos de referência, mantivemos a mesma metodologia de avaliação de qualidade do artigo 12.

Em particular, implementamos uma validação cruzada considerando-se $T = 75\%$ do banco de dados como base de treinamento ou aprendizado e os 25% restantes como usuários-teste. Em seguida, foram mascarados $H = 75\%$ das avaliações dos usuários-teste, de modo a medir a qualidade do sistema de recomendação em prever os itens que haviam sido positivamente avaliados.

reescrever

A realização de testes será feita com os bancos de dados de centenas de milhares de itens ou de avaliações. Visto que será feita uma validação cruzada, será necessário descartar os dados e reformular a solução caso as recomendações não atinjam os requisitos funcionais. Isso evita que o sistema seja moldado para operar somente com aquele banco de dados específico.

NAME="Amazon Linux AMI"VERSION="2014.09"ID="amzn"ID$_LIKE$ = ”$rhelfedora$”$VERSIO$
”2014.09”$PRETTY_NAME$ = ”$AmazonLinuxAMI$2014.09”$ANSI_COLOR$ = ”0; 33”CPE_NAME =
”$cpe : /o : amazon : linux : 2014.09 : ga$”$HOME_URL$ = ”$http : //aws.amazon.com/amazon-linux - ami/$”$AmazonLinuxAMIrelease$2014.09

Linux 3.14.20-20.44.amzn1.x86$_6$4$x86_6$4

8 Resultados

Os resultados deste trabalho são a análise de desempenho dos algoritmos propostos, em termos de precisão, abrangência e tempo computacional, mediante a mudanças em suas variáveis de importância (Tabela 23).

Além disso, as metodologias de solução de cada um dos sistemas serão debatidas, de modo a explorar casos de uso particulares e a propor melhorias nos métodos computacionais. Serão respondidas perguntas como "O que acontece com itens ou usuários sem nenhuma avaliação?" e "Qual o desempenho dos métodos para outros bancos de dados?".

Tabela 23 – Parâmetros de influência no desempenho dos algoritmos de recomendação

Variável	**Descrição**	**Valor padrão**
N	Tamanho da lista de recomendação	20
T	Percentual da base de aprendizado na validação cruzada	75%
H	Percentual de avaliações "escondidas" dos usuários-teste na validação cruzada	75%
M	Valor mínimo para avaliações positivas	2
k	Número de vizinhos mais próximos	10
$\mathcal{F}$	Conjunto de atributos dos itens	Escolhido *a priori* para cada método
d^f	Medida de distância entre atributos	$1 - \delta^f$
w_f	Pesos dos atributos	$w_f > 0$

8.1 Tamanho da lista de recomendações N

Assim como mostra a literatura, a medida que o tamanho da lista de recomendações aumenta, a precisão cai e a abrangência cresce (Figuras 8.1).

O método UP supera os dois outros algoritmos para todos os valores de N, tanto em precisão quanto em abrangência, como se observa pelo gráfico das medidas F_1.

Contrariamente ao esperado, a qualidade de recomendação do algoritmo UI é sensivelmente inferior à do algoritmo UP. Isso se deve ao fato de a correlação usuário-item daquele método colocar ênfase no valor do atributo a_{if}, mesmo que esses atributos não sejam diretamente proporcionais à preferência do usuário. Esse cálculo é incoerente, por exemplo, para atributos $f =$ data: mesmo que o usuário tenha um elevado interesse w_{uf} por filmes antigos, o valor de a_{if} não leva em conta se sua preferência é por filmes da década de 1970 ou 1990. Nesse caso, o algoritmo indicaria incorretamente que filmes mais recentes são mais adequados para aquele usuário, porque possuem maior a_{if}.

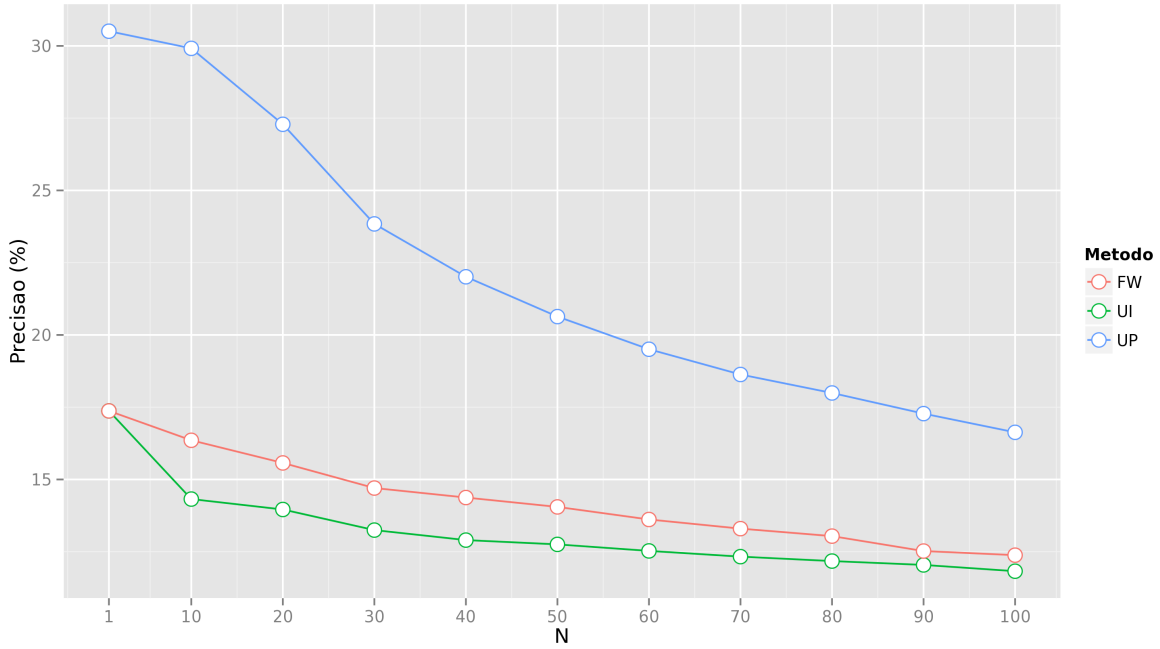

Figura 1 – Precisão em função do tamanho da lista de recomendações N

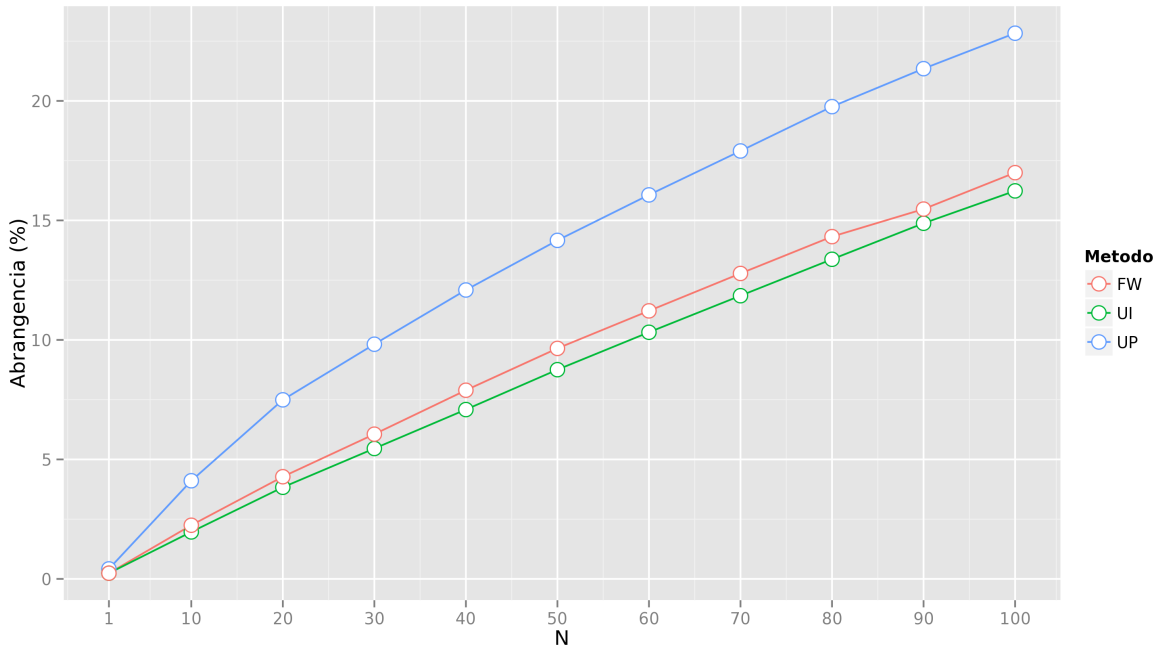

Figura 2 – Abrangência em função do tamanho da lista de recomendações N

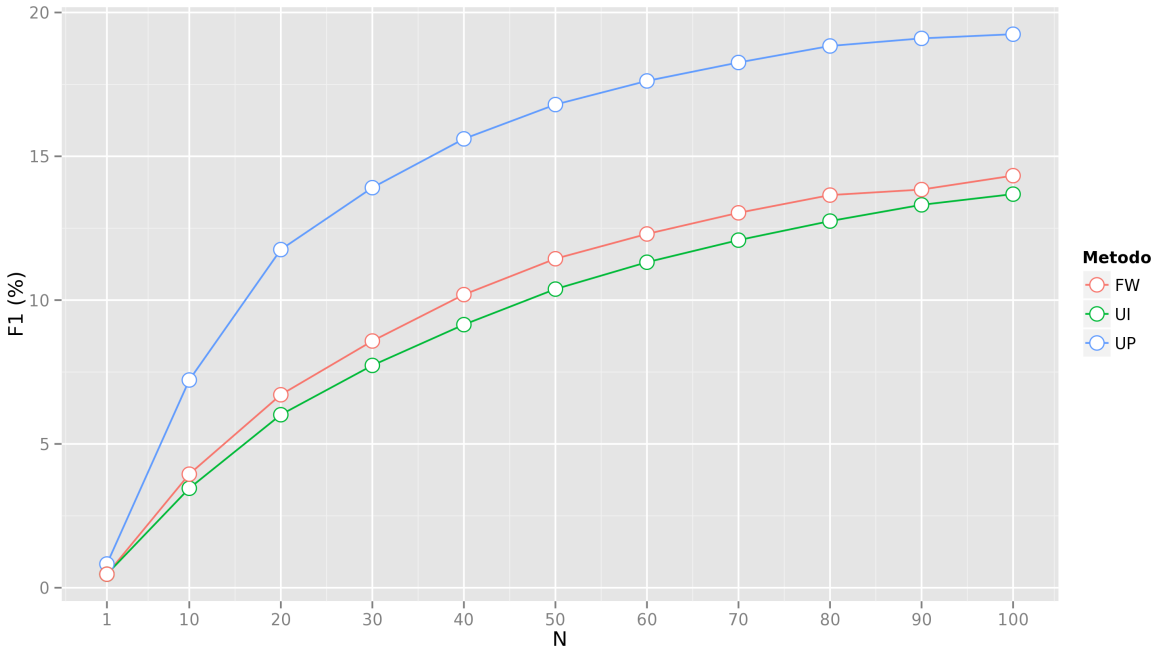

Figura 3 – Medida F_1 em função do tamanho da lista de recomendações N

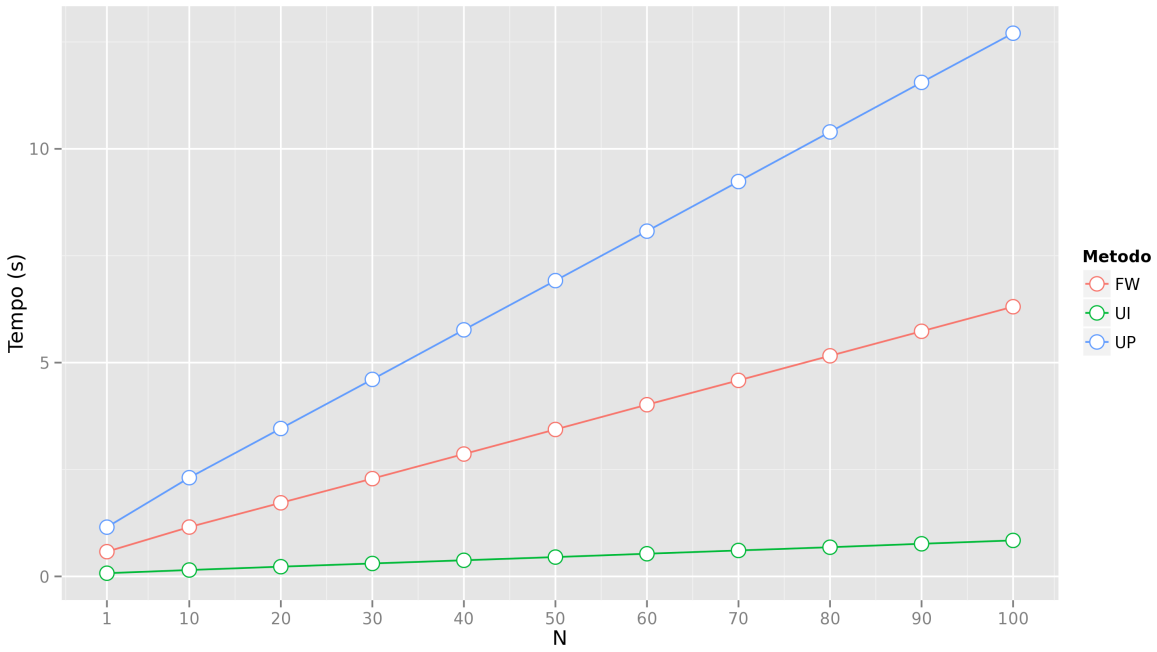

Figura 4 – Tempo de execução em função do tamanho da lista de recomendações N

A fim de corrigir essa falha no algoritmo UI, seria necessário, por exemplo, aplicar nos atributos a_{if} uma função g_f que crescesse no mesmo sentido do interesse do usuário por aquela *feature*. Dessa forma, o cálculo $\sum_f w_{uf}\ g_f\left(a_{if}\right)$ significaria de fato a similaridade entre o usuário u e o item i medida através de seu interesse $g\left(a_{if}\right)$ pelas *features* f.

Apesar de alta qualidade das recomendações do método UP, este possui também a maior complexidade computacional. Seu tempo de execução é 1.9 vezes maior que o do método FW e 13.9 vezes maior que o do método UI. Todavia, nenhum desses tempos de execução é crítico, tendo em vista que o sistema não seria colocado diretamente à disposição dos clientes, mas que as recomendações seriam enviadas via email, por exemplo.

Apenas o método UI, para valores de N inferiores ou iguais a 10, atendem ao requisito de *throughput* mínimo de 28 recomendações para cada usuário por segundo. Dado que a base de testes possui 25% do total de usuários, correspondente a 236 clientes para o banco 100k, o tempo de execução máximo dos métodos deveria ser de 0.15 min. A fim de melhorar a velocidade das recomendações, a solução mais eficiente é a mudança da linguagem de programação. O uso de linguagens C, C++ ou Python pode melhorar o desempenho computacional em até 500 vezes (32).

8.2 Percentual da base de aprendizado T

9 old

9.1 Primeira etapa do Trabalho de Conclusão de Curso

Os resultados da primeira etapa deste Trabalho de Conclusão de Curso, realizadas na disciplina PMR2500, foram principalmente a definição de necessidades, de parâmetros de sucesso e a elaboração de possíveis soluções.

Definimos que a aquisição de dados seria feita a partir de uma base qualquer, que deveria alimentar o sistema por meio de arquivos de texto com valores separados por vírgulas (`.csv`).

A fim de facilitar o pré-processamento dos dados, estabelecemos que seriam necessários dois arquivos. Um deles deve conter a matriz de atributos $\mathbf{A}$ e o outro, a matriz de avaliações $\mathbf{R}$.

$$\mathbf{A} = \begin{bmatrix} a_{i_1 f_1} & a_{i_1 f_2} & a_{i_1 f_3} & \dots \\ a_{i_2 f_1} & a_{i_2 f_2} & a_{i_2 f_3} & \dots \\ a_{i_3 f_1} & a_{i_3 f_2} & a_{i_3 f_3} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{9.1}$$

$$\mathbf{R} = \begin{bmatrix} r_{u_1 i_1} & r_{u_1 i_2} & r_{u_1 i_3} & \dots \\ r_{u_2 i_1} & r_{u_2 i_2} & r_{u_2 i_3} & \dots \\ r_{u_3 i_1} & r_{u_3 i_2} & r_{u_3 i_3} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{9.2}$$

Em alguns bancos de dados relacionais, a tabela de avaliações também contém informações adicionais θ, tais como método de pagamento, data da compra, data de entrega, etc., e é denominada matriz histórico de avaliações $\mathbf{H}$. Optamos, no nosso projeto, por não aceitar esse tipo de informação, e nem tampouco dados ligados a características de clientes (matriz $\mathbf{B}$). Para o emprego do sistema de recomendação em um e-commerce real, deve-se portanto efetuar ajustes no algoritmo a fim de tratar de particularidades envolvendo informações adicionais e arquivos suplementares.

$$\mathbf{H} = \begin{bmatrix} r_{u_1 i_1} & \theta_{h_1 1} & \theta_{h_1 2} & \dots \\ r_{u_1 i_2} & \theta_{h_2 1} & \theta_{h_2 2} & \dots \\ r_{ui} & \theta_{h1} & \theta_{h2} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{9.3}$$

$$\mathbf{B} = \begin{bmatrix} b_{u_1c_1} & b_{u_1c_2} & b_{u_1c_3} & \dots \\ b_{u_2c_1} & b_{u_2c_2} & b_{u_2c_3} & \dots \\ b_{u_3c_1} & b_{u_3c_2} & b_{u_3c_3} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{9.4}$$

Uma vez determinada a forma de entrada de informações, definiram-se os conjuntos de dados que serão utilizados. O primeiro conjunto de dados abertos é proveniente do sistema de recomendações de filmes MovieLens (<http://movielens.umn.edu>). Nessa base de dados, o catálogo de filme faz o papel de catálogo de produtos, e o histórico de compras se refere à avaliação dos filmes feita por cada usuário. Outro conjunto de dados abertos é do website Internet Movie Database (IMDB). Na nossa análise, esses dois bancos poderão ser utilizado complementar ou independentemente.

Na primeira etapa do projeto, buscamos parcerias com e-commerces que estivessem dispostos a doar anonimamente seu banco de dados. Há ainda a possibilidade de utilizarmos uma terceira base, mas visto que as negociações ainda não foram concluídas, daremos prioridades aos conjuntos *open source*.

9.2 Segunda etapa do Trabalho de Conclusão de Curso

A partir da síntese de soluções estabelecida na primeira etapa do projeto, implementamos os três algoritmos de recomendação e as medidas de recomendação na linguagem de programação estatística R. O código já está disponível para consulta através do endereço <https://github.com/aviggiano/tcc>.

Ainda na etapa de implementação, confirmamos a validade de cada um dos métodos aplicando-os nas matrizes-referência (Tabelas 5 e 6).

Em seguida, fizemos o tratamento das bases de dados, adequando-as ao formato de entrada especificado, e iniciamos o processo de desenvolvimento do *cross-validation*.

Para os trabalhos futuros, iremos realizar a validação cruzada e avaliar se os requisitos funcionais foram estabelecidos. Em seguida, procuraremos melhorar o sistema de recomendação a fim de torná-lo mais genérico. Buscaremos eliminar restrições quanto a entrada e saída de dados, de forma que elas sejam completamente arbitrárias. O objetivo é que o usuário possa informar ao sistema como é formado sua base, e que todo o tratamento preliminar seja feito automaticamente.

Caso haja tempo, trabalharemos também na construção de um *driver* que possibilite a conexão entre o sistema de recomendação e um banco de dados SQL, sem que seja necessária a etapa intermediária de arquivos `csv` para aquisição de dados. Planejamos elaborar um *website* para o sistema de recomendação e exportar toda a lógica para um

servidor dedicado. Outra melhoria desejada é a reconstrução dos métodos na linguagem de programação C, a fim de melhorar a performance computacional. Dessa forma, o serviço de "sistema de recomendação nas nuvens" estaria completo e poderia ser utilizado por e-commerces reais.

Referências

1 EMARKETER. *B2C Ecommerce Climbs Worldwide, as Emerging Markets Drive Sales Higher*. 2013. Disponível em: <http://www.emarketer.com/Article/B2C-Ecommerce-Climbs-Worldwide-Emerging-Markets-Drive-Sales-Higher/1010004>. Citado na página 21.

2 EMARKETER. *Global B2C Ecommerce Sales to Hit $1.5 Trillion This Year Driven by Growth in Emerging Markets*. 2014. Disponível em: <http://www.emarketer.com/Article/Global-B2C-Ecommerce-Sales-Hit-15-Trillion-This-Year-Driven-by-Growth-Emerging-Markets/1010575>. Citado na página 21.

3 MAC, R.; SOLOMON, B. *Alibaba Boosts IPO Price Range, Could Raise Up To $25 Billion*. 2014. Disponível em: <http://www.forbes.com/sites/ryanmac/2014/09/15/alibaba-raises-ipo-price-range-could-raise-up-to-25-billion/>. Citado na página 21.

4 COOPERS, P. W. *Total Retail Global Survey of Online Shoppers*. 2014. Disponível em: <http://www.pwc.com/gx/en/retail-consumer/retail-consumer-publications/global-multi-channel-consumer-survey/index.jhtml>. Citado na página 21.

5 COOPERS, P. W. *The Go-to-Market Revolution - Igniting Growth with Marketing, Sales, and Pricing*. 2014. Disponível em: <https://www.bcgperspectives.com/content/articles/go_to_market_strategy_growth_go_to_market_revolution_igniting_growth_marketing_sales_pricing>. Citado na página 21.

6 RICCI, L. R. F.; SHAPIRA, B. Introduction to recommender systems handbook. In: *Recommender Systems Handbook*. [S.l.]: Springer, 2011. p. 1–35. Citado na página 21.

7 AMATRIAIN, X. *Netflix Recommendations: Beyond the 5 stars*. 2012. Disponível em: <http://techblog.netflix.com/2012/04/netflix-recommendations-beyond-5-stars.html>. Citado na página 21.

8 MARSHALL, M. *Aggregate Knowledge raises $5M from Kleiner, on a roll*. 2006. Disponível em: <http://venturebeat.com/2006/12/10/aggregate-knowledge-raises-5m-from-kleiner-on-a-roll/>. Citado na página 21.

9 DAS, A. S. et al. Google news personalization: scalable online collaborative filtering. In: ACM. *Proceedings of the 16th international conference on World Wide Web*. [S.l.], 2007. p. 271–280. Citado na página 21.

10 SCHAFER, J. B.; KONSTAN, J.; RIEDL, J. Recommender systems in e-commerce. In: ACM. *Proceedings of the 1st ACM conference on Electronic commerce*. [S.l.], 1999. p. 158–166. Citado 2 vezes nas páginas 22 e 30.

11 SARWAR, B. et al. Analysis of recommendation algorithms for e-commerce. In: ACM. *Proceedings of the 2nd ACM conference on Electronic commerce*. [S.l.], 2000. p. 158–167. Citado na página 23.

12 SYMEONIDIS, P.; NANOPOULOS, A.; MANOLOPOULOS, Y. Feature-weighted user model for recommender systems. In: *User Modeling 2007*. [S.l.]: Springer, 2007. p. 97–106. Citado 6 vezes nas páginas 25, 30, 33, 37, 38 e 48.

13 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*, IEEE, v. 17, n. 6, p. 734–749, 2005. Citado 3 vezes nas páginas 25, 26 e 29.

14 BALABANOVIC, M.; SHOHAM, Y. Fab: Content-based, collaborative recommendation. *Communications of the ACM*, v. 40, p. 66–72, 1997. Citado 2 vezes nas páginas 26 e 29.

15 WEI, K.; HUANG, J.; FU, S. A survey of e-commerce recommender systems. In: IEEE. *Service Systems and Service Management, 2007 International Conference on*. [S.l.], 2007. p. 1–5. Citado 3 vezes nas páginas 26, 28 e 30.

16 SCHAFER, J. B.; KONSTAN, J. A.; RIEDL, J. E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, v. 5, p. 115–153, 2001. Citado na página 26.

17 LOPS, P.; GEMMIS, M. de; SEMERARO, G. Content-based recommender systems: State of the art and trends. In: *Recommender Systems Handbook*. [S.l.]: Springer, 2011. p. 73–105. Citado na página 26.

18 LINDEN, G.; SMITH, B.; YORK, J. Amazon.com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, IEEE, v. 7, n. 1, p. 76–80, 2003. Citado na página 27.

19 BURKE, R. Hybrid web recommender systems. In: *The adaptive web*. [S.l.]: Springer, 2007. p. 377–408. Citado na página 27.

20 LEE, J.; SUN, M.; LEBANON, G. A comparative study of collaborative filtering algorithms. *arXiv preprint arXiv:1205.3193*, 2012. Citado na página 28.

21 TUTOL, L. *Amazon Launches 'Login and Pay with Amazon' for a Seamless Buying Experience*. 2013. Disponível em: <http://services.amazon.com/post/Tx2A98P3EKP62O2/Amazon-Launches-Login-and-Pay-with-Amazon-for-a-Seamless-Buying-Experience>. Citado na página 28.

22 PALLADINO, V. *Amazon sold 426 items per second in run-up to Christmas*. 2013. Disponível em: <http://www.theverge.com/2013/12/26/5245008/amazon-sees-prime-spike-in-2013-holiday-season>. Citado na página 28.

23 FENNELL, J. Collaborative filtering on sparse rating data for yelp. com. 2009. Citado na página 29.

24 LOPS, P.; GEMMIS, M. de; SEMERARO, G. In: *Recommender Systems Handbook*. [S.l.]: Springer, 2011. Citado na página 30.

25 DEBNATH, S.; GANGULY, N.; MITRA, P. Feature weighting in content based recommendation system using social network analysis. In: ACM. *Proceedings of the 17th international conference on World Wide Web*. [S.l.], 2008. p. 1041–1042. Citado 4 vezes nas páginas 30, 33, 37 e 39.

26 SARWAR, B. et al. Item-based collaborative filtering recommendation algorithms. In: ACM. *Proceedings of the 10th international conference on World Wide Web*. [S.l.], 2001. p. 285–295. Citado na página 33.

27 MOVIELENS. *MovieLens 100k Dataset*. 1998. Disponível em: <http://files.grouplens.org/datasets/movielens/ml-100k-README.txt>. Citado na página 33.

28 WICKHAM, H. *Movies dataset*. 2006. Disponível em: <http://docs.ggplot2.org/0.9.3.1/movies.html>. Citado na página 33.

29 HOPE, C. *Epoch*. 2014. Disponível em: <http://www.computerhope.com/jargon/e/epoch.htm>. Citado na página 40.

30 A Guide To The Project Management Body Of Knowledge (PMBOK Guides). [S.l.]: Project Management Institute, 2004. ISBN 193069945X, 9781933890517. Citado na página 47.

31 LARMAN, C.; BASILI, V. R. Iterative and incremental development: A brief history. *Computer*, IEEE Computer Society, Los Alamitos, CA, USA, v. 36, n. 6, p. 47–56, 2003. ISSN 0018-9162. Citado na página 47.

32 COOK, J. D. *Benchmarking C++, Python, R, etc.* 2014. Disponível em: <http://www.johndcook.com/blog/2014/06/20/benchmarking-c-python-r-etc/>. Citado na página 52.