



Universidade Federal de Pernambuco
Centro de Informática

Graduação em Ciência da Computação

**Rec-Vest: um sistema de recomendação de
vestuário em lojas de comércio eletrônico**

Artur de Carvalho Montenegro Henriques

Orientador: Prof. Ricardo Bastos Cavalcante
Prudêncio

Trabalho de Graduação

Recife
Dezembro de 2017

*Dedico este trabalho ao meu avô, Francisco Manoel
Henriques, que por muito pouco não esteve aqui comigo
neste momento.*

Agradecimentos

Quero agradecer aos meus pais, que lutaram desde o início, fazendo o possível e impossível, para que eu pudesse ter as melhores coisas na vida, principalmente quando o assunto era educação, mesmo que eu algumas vezes não tivesse maturidade para aproveitá-las completamente. Além disso por terem cuidado de mim com todo zelo e carinho.

Agradeço também a minha família, tios, tias, primos, avôs e avós e tia-avó, que sempre cuidaram de mim como se eu fosse o próprio filho e me proporcionaram momentos muito felizes e um acolhimento incrível.

De forma não diferente, gostaria de agradecer a minha namorada Tamíz Freitas, por ser um exemplo de mulher. Humana, forte, focada, inteligente. Me inspira todo dia para que eu seja uma pessoa melhor. Agradeço por todo o suporte dado e pela paciência nos momentos de estresse.

Agradeço aos amigos que fizeram que os pesares da vida fossem muito mais leves, não vou citar nomes um a um devido que a lista é muito grande e posso deixar alguém de fora.

Obrigado por último ao transporte público de Recife e Jaboatão e a mobilidade urbana, que é tão demorado, passei tanto tempo nele e me deslocando, que pude ouvir várias músicas de vários artistas (que hoje não sei o que seria de mim caso não as ouvisse), li vários livros (se tivesse um curso EAD, teria agora duas graduações), pude dormir bastante, e pude aprender muito da sabedoria popular.

Meus sinceros agradecimentos.

Resumo

No ambiente online, os comentários (do inglês, *reviews*) de produtos ou serviços direcionam as compras em sites de *e-commerce*. Logo, os Sistemas de Recomendação (SR) baseiam-se no comportamento dos consumidores para gerar recomendações dos itens nos sites de compras. Nesse contexto, este trabalho teve como objetivo o desenvolvimento de um Sistema de Recomendação, com posterior aplicação em um domínio na web vinculado a uma empresa de vestuário. Adotou-se aqui uma técnica híbrida de Recomendação baseada em avaliações do usuário e no conteúdo. O protótipo foi desenvolvido utilizando a linguagem Python e foi testado com um corpus de 6158 usuários e 3269 itens, tendo obtidos resultados satisfatórios.

Palavras-chave: Sistemas de Recomendação, Filtragem Colaborativa, Comércio Online, Venda Personalizada

Abstract

In the online environment, reviews of products or services direct the purchases in e-commerce sites. Therefore, the Recommender Systems (RS) are based on the customers' behavior to generate recommendations of the items on shopping sites. In this context, this study had the goal a RS development, with further application in a web domain linked to clothing business. The approach used was grounded in Colaborative Filtering and Content-based. The prototype had been developed using the Python language and tested with a corpus of 6158 users and 3269 items.

Keywords: Recommender Systems, Colaboratve Filtering, E-Commerce, Customized Sales

Sumário

1	Introdução	1
2	Sistemas de Recomendação	3
2.1	Definição de Sistemas de Recomendação	4
2.2	Classificação dos Sistemas de Recomendação	5
2.2.1	Recomendação Baseada em Conteúdo	5
2.2.2	Recomendação baseada em Filtragem Colaborativa	5
2.2.3	Recomendação baseada em filtragem híbrida	6
2.3	Considerações Finais	6
3	Recomendação Baseada no Conteúdo	7
3.1	Processo de Filtragem	7
3.2	Conhecendo o usuário	8
3.2.1	Cálculo de Similaridade	8
3.3	Benefícios e Limitações	8
3.4	Considerações Finais	9
4	Recomendação baseada em Filtragem Colaborativa	10
4.1	Filtragem baseada em Vizinhança (Memória)	11
4.1.1	Cálculo de similaridade	12
4.1.1.1	Correlação de Pearson	12
4.1.1.2	Distância Euclidiana	13
4.1.1.3	Coseno	13
4.2	Filtragem Baseada em Modelo	13
4.2.1	Decomposição em valores singulares	14
4.3	Considerações Finais	15
5	Metodologia e Resultados	16
5.1	Conjunto de Dados	16
5.2	Métricas de avaliação	16
5.3	Procedimento Experimental	18
5.4	Resultados da experimentação	19
5.4.1	Efeitos da similaridade	19
5.4.2	Comparativo da acurácia entre abordagens	19
5.5	Experimentos de Qualidade	23
5.5.1	Sensibilidade ao variar o tamanho do conjunto de treinamento	24

SUMÁRIO	7
5.5.2 Sensibilidade ao variar o tamanho do conjunto de itens recomendados	24
5.6 Considerações finais	25
6 Conclusão e trabalhos futuros	26
Referências Bibliográficas	27

Lista de Figuras

5.1	Parte da lista de predições obtidas pelo algoritmo de Decomposição em Valores Singulares	20
5.2	Parte da lista de predições obtidas pelo algoritmo baseado em vizinhança	20
5.3	As dez melhores predições do algoritmo baseado em vizinhança	20
5.4	As dez melhores predições do algoritmo baseado em modelo	21
5.5	As dez piores predições do algoritmo baseado em vizinhança	21
5.6	As dez piores predições do algoritmo baseado em modelo	22
5.7	Resultado da abordagem em modelo sobre as 10 piores predições do algoritmo baseado em vizinhança	22
5.8	Comportamento das estimativas dos algoritmos	23

Lista de Tabelas

5.1	Configurações iniciais do conjunto de dados	16
5.2	Configurações pós processamento do conjunto de dados	16
5.3	Possíveis resultados para os itens recomendados e escondidos	17
5.4	Cálculo do MAE Médio para cada similaridade e abordagem	19
5.5	Erro médio para o problema de partida a frio	23
5.6	Precisão e Cobertura para abordagem em vizinhança	24
5.7	Precisão e Cobertura para abordagem em modelo	24
5.8	Métrica de cobertura para abordagens em modelo e vizinhança variando-se o K (tamanho da lista de recomendação)	24

CAPÍTULO 1

Introdução

Com o aumento do acesso às tecnologias pela população, aliado à facilidade de produzir conteúdo em massa e disponibilizá-lo em rede para todo o mundo, chegou-se ao problema da sobrecarga de informação. Não raro, quando procuramos por um termo simples em sites de busca, a exemplo do Google, milhões e às vezes bilhões de resultados são ofertados. A necessidade de filtrar a pesquisa é inevitável, e quanto mais específica, mais provável de encontrar os melhores resultados.

A busca de uma solução frente a essa dificuldade impulsionou o desenvolvimento de diversas áreas de pesquisa. A Recuperação da Informação (RI), por exemplo, 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 [1].

Entretanto, os Sistemas de Recuperação da Informação (SRI) podem recuperar documentos que não são do interesse do usuário ou um número elevado de resultados [1]. Outro problema é a dificuldade que o usuário tem, por vezes, de expressar suas necessidades através de uma consulta por palavras chave, ou até mesmo de saber com precisão o que deseja buscar [2].

Sistemas de Recomendação (SR) surgiram como uma subárea de pesquisa dos Sistemas de Filtragem da Informação e evoluiu para se tornar uma área independente. A proposta desse campo de pesquisa é construir sistemas capazes de identificar as preferências do usuário e, a partir delas, recuperar itens que sejam do interesse deste. Esses sistemas têm sido intensamente pesquisados nos últimos anos e difundidos largamente em sites de comércio eletrônico, jogos, entretenimento, fóruns de discussão e redes sociais [2].

No contexto de lojas online, o objetivo é impulsionar as vendas, mostrando a cada usuário itens relevantes ao seu perfil e que ainda não foram consumidos por ele. Para sistemas de comércio modernos, os benefícios de um SR são indispensáveis. Eles permitem aumentar significativamente o número de compras, o tempo que o usuário despende no sistema e, por consequência, o número de acessos ao sistema.

Um exemplo bastante significativo é o da Amazon¹, que obtém uma grande quantidade de dados rastreando o usuário em seu website e, assim, fornece ofertas personalizadas em todos os seus processos de compras. Outro exemplo de igual relevância é o da Netflix², que não só garante que a página inicial do cliente tenha conteúdo personalizado relevante, mas que a partir de recomendações, produz séries com conteúdo baseado em nichos específicos.

Ou seja, qualquer que seja o negócio, integrá-lo a um SR é uma ação positiva para a empresa, com resultados tais como a adição de mais um item na venda, aumentando o lucro da

¹www.amazon.com

²www.netflix.com

organização; mas também benéfico para o cliente, simplificando o processo de escolha daquele consumidor.

O desenvolvimento da área de SR promoveu o surgimento de diversas técnicas para resolver tais problemas. Dentre elas, a Filtragem Colaborativa (FC) e a Recomendação Baseada no Conteúdo[2]. A FC baseia-se na ideia 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 de um grupo de pessoas que possuem gostos e preferências em comum. Já a Recomendação baseada no conteúdo faz uso dos metadados dos itens para recomendá-los de acordo com a preferência dos usuários.

Este estudo analisa os SRs, com o objetivo de implantar e avaliar tais sistemas em uma aplicação real na web na área do comércio online com o objetivo de aproximar pessoas com preferências em comum ou gostos mais individualizados, e recomendar novos produtos de forma automática. Além disso, fez-se necessário apresentar uma visão geral da área de Sistemas de Recomendação, mostrando sua evolução e sua importância no contexto atual. Como estudo de caso, utilizamos um corpus de dados oriundos de uma empresa que atua no comércio de vestimentas.

O trabalho está dividido em seis capítulos, descritos resumidamente, a seguir. No capítulo 2, serão apresentados os Sistemas de Recomendação e as principais contribuições dessa área de pesquisa. As principais técnicas serão levantadas juntamente com as três classificações para SR: Filtragem por Conteúdo; Filtragem Colaborativa e Abordagem Híbrida. O capítulo 3 explicará de forma objetiva a técnica de Recomendação Baseada em Conteúdo. O capítulo 4 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 5, a metodologia de avaliação e os resultados experimentais serão trazidos à tona. O capítulo 6 coloca a conclusão da realização do protótipo e aponta direcionamentos para trabalhos futuros finalizando, assim, este estudo.

Sistemas de Recomendação

Sistemas de recomendação (SR) são ferramentas de software baseados em um conjunto de técnicas que têm como objetivo prever a probabilidade de um usuário escolher um determinado item, a partir de suas preferências e restrições. Esse histórico de cada usuário pode ser criado com base em interações anteriores com um determinado site de vendas online (E-commerce). Na filtragem colaborativa, por exemplo, o sistema colhe dados de outros usuários com perfil semelhante ou até mesmo transações passadas feitas pelo próprio cliente e, assim, torna-se possível a recomendação de um determinado produto. A partir dessa ótica, permite-se a visualização das variadas aplicações e de como pode ser útil um Sistema de Recomendação.

O desenvolvimento de SRs surgiu da simples observação de que indivíduos cada vez mais tomam suas decisões a partir de opiniões fornecidas por outros que criam rotinas nas tomadas de decisões diárias. Esse comportamento é observado, por exemplo, na recomendação de livros, filmes e outros produtos culturais, entre amigos; o fornecimento de cartas de recomendação para potenciais empregados em seleções de emprego, ou até mesmo na escolha de um filme fundamentada em críticas anteriores daqueles que já o assistiram.

Além disso outra motivação é que, no mundo virtual, onde a quantidade de informação está crescendo exponencialmente, encontrar informação válida é um grande desafio. Outro desafio maior ainda é fazer com que o usuário sinta confiança nos resultados procurados.

Primeiramente, os Sistemas de Recomendação foram chamados de Sistemas de Filtragem Colaborativa devido à utilização dessa técnica no primeiro sistema de recomendação Tapestry, sistema onde os usuários indicavam documentos entre si, o qual foi desenvolvido pela Xerox. O termo mais genérico, Sistema de Recomendação, foi proposto por [3]. Os autores consideram o termo mais apropriado porque a recomendação não precisa ser necessariamente produto da colaboração entre usuários e recomendações sugeridas.

A explosão de implementações de SRs ocorreu por causa do grande sucesso obtido por gigantes da tecnologia. Alguns exemplos de casos reais incluem as sugestões de produtos feitas pela Amazon, sugestões de amigos em redes sociais pelo Facebook, Twitter e LinkedIn. Além disso, citamos a recomendação de vídeos no Youtube e notícias no Google News. Toda essa jornada de desenvolvimento e aperfeiçoamento de sistemas de recomendação abriu precedentes para outras áreas como saúde, viagens e bancos.

Dada essa complexidade, desafios e a quantidade de esforço tecnológico, financeiro, planejamento e de desenvolvimento para se construir um SR, será que vale a pena investir tanto? Vamos atentar para alguns fatos:

- $\frac{2}{3}$ dos filmes assistidos no Netflix são oriundos de recomendações.
- 35% das vendas da Amazon são geradas pelo seu sistema de recomendação.

De acordo com um paper elaborado por executivos da Netflix, sistemas de recomendação propiciam uma economia de 1 bilhão de dólares por ano, fazendo com que sobre mais dinheiro para a criação de novos conteúdos [3].

Observando a importância de ter o melhor engenho de recomendação, a Netflix colocou bastante esforço em otimizar seu algoritmo, e em 2009 ofereceu um prêmio de 1 milhão de reais em uma competição para qualquer grupo de pesquisa que pudesse aumentar a eficiência do seu algoritmo em 10% [3]. Portanto, a jornada da Netflix e Amazon para oferecer uma experiência de visualização do conteúdo e compra, respectivamente, mostram o quão poderoso um sistema desse tipo pode ser.

2.1 Definição de Sistemas de Recomendação

Sistemas de Recomendação, em termos coloquiais, são uma mistura de conjuntos de tecnologias e técnicas das áreas de Recuperação da Informação (RI) e Inteligência Artificial (IA), desenvolvidas para analisar um grande volume de dados, especialmente informações de produtos e informações sobre o usuário, e então prover sugestões relevantes para usuários, para que eles possam tomar melhores decisões dentre as alternativas disponíveis. Um SR é direcionado a prover recomendações personalizadas a partir de rastros do usuário, como informações demográficas, detalhes de uma transação, logs de interação e informação sobre o produto como por exemplo especificação e feedback dos usuários, ou até mesmo comparação entre os produtos.

Um SR tem, portanto, duas tarefas principais: recomendação e predição. O objetivo da recomendação é gerar uma lista com os melhores itens avaliados, enquanto que a tarefa individual de estimar a avaliação para um único item é conhecida como predição [4].

Em termos técnicos, o problema do sistema de recomendação é desenvolver um modelo matemático ou uma função objetivo que pode prever o quanto um usuário vai gostar de determinado item. Para ser mais preciso, precisaremos de definições com notação mais formal, apresentadas a seguir[5]:

Definição 1. *Um item pertence a uma classe de objetos do mesmo tipo, indistinguíveis pelo usuário, e com pelo menos uma instância.*

Definição 2. *O usuário é uma pessoa capaz de prover sua própria preferência avaliando um determinado item.*

Além disso, o conjunto de usuários será denotado por U e o conjunto de itens, por I . Temos também que identificar o conjunto R de avaliações dos usuários, que dependendo do sistema, pode armazenar valores numéricos, de um a cinco, ou valores categóricos como *like* (avaliação positiva) ou *dislike* (avaliação negativa). É importante também assumir que não mais do que uma avaliação de um usuário $u \in U$ pode ser feita para um item $i \in I$. Da mesma forma, I_u representa o subconjunto de itens que foram avaliados pelo usuário u .

Portanto, o problema consiste em encontrar, para um determinado usuário $u \in U$, um novo item $i \in I \setminus I_u$ pelo qual u mais se interesse. Quando avaliações numéricas são fornecidas, essa tarefa é frequentemente definida como uma classificação onde a função objetivo é $f : U \times I \rightarrow S$ que prediz a avaliação de um usuário u para um novo item i . Esta função é, então, usada para

recomendar ao usuário atual u_a um item i^* para o qual a avaliação estimada tem o maior valor (ou seja, que tem mais utilidade para o usuário):

$$i^* = \max_{j \in I \setminus I_u} f(u_a, j). \quad (2.1)$$

2.2 Classificação dos Sistemas de Recomendação

Aggarwal [6] provê uma classificação que se tornou bastante popular para se distinguir entre os tipos de SR e se tornou base para estudos que o sucederam. O autor diferencia seis classes de sistemas por suas abordagens de recomendação, a saber: Recomendação baseada em conteúdo, Filtragem Colaborativa, Filtragem Demográfica, Recomendação baseada em conhecimento, Recomendação baseada em comunidade, Recomendação Híbrida. As mais encontradas na literatura são Recomendação Baseada em conteúdo, Filtragem Colaborativa e Recomendação Híbrida.

2.2.1 Recomendação Baseada em Conteúdo

Como o nome sugere, a Recomendação Baseada em Conteúdo (ou Recomendação Cognitiva) utiliza descrições (em forma de metadados) sobre os itens para construir um modelo de recomendação. Por exemplo, se você procura sobre League of Legends no Youtube, um SR baseado em conteúdo iria aprender sua preferência e recomendar outros vídeos relacionados a League of Legends, e além disso, poderia recomendar outros vídeos relacionados a jogos online. Em palavras simples, o sistema de recomendação indicaria itens similares àqueles que o usuário tenha gostado no passado baseado nas informações sobre o item comparado com outros itens presentes no histórico de preferência do usuário ativo.

Para itens que têm conteúdos na forma de um documento de texto, essa abordagem é bastante utilizada [7]. O perfil do item é representado por um vetor TF-IDF (Term Frequency-Inverse Document Frequency) [8], técnica oriunda das áreas de Processamento de Linguagem Natural e Recuperação Informação, que consiste em procurar e quantificar frases de uma, duas ou três palavras que aparecem múltiplas vezes na descrição de um item e dividir pelo número de vezes que essas mesmas palavras aparecem em todas as descrições de todos os itens existentes. Então, os termos que são mais distintos para um determinado item recebem um *score* maior, e termos que aparecem com alta frequência em um documento, mas também aparecem em outros documentos recebem um *score* menor.

Uma vez que o vetor foi calculado para cada item, usamos uma medida de similaridade que calcular o cosseno dos ângulos entre os vetores para identificar quais produtos são mais parecidos entre si. Detalharemos mais sobre essa abordagem no capítulo 3.

2.2.2 Recomendação baseada em Filtragem Colaborativa

Sistemas Baseados em Filtragem Colaborativa são os mais recorrentes em engenhos de recomendação. Nessa abordagem, itens são filtrados de um grande conjunto de alternativas a partir de comparações entre as preferências dos usuários. Observando de outra maneira, se dois usuários tiveram os mesmos interesses no passado, eles terão os mesmos interesses no futuro.

Exemplificando, se um usuário A e um usuário B têm a mesma preferência de música, e o usuário A recentemente escutou Lazy de Deep Purple, o qual o usuário B ainda não escutou, então a ideia é recomendar essa música ainda não escutada para o usuário B. Entraremos em detalhes mais profundos sobre essa abordagem, explicando os prós e contras, no capítulo 4.

2.2.3 Recomendação baseada em filtragem híbrida

Sistemas de Recomendação Híbridos são construídos combinando várias abordagens para conceber um sistema mais robusto. Assim, podemos sanar as desvantagens de uma única abordagem com as vantagens de outras. Por exemplo, combinar Filtragem Colaborativa, cujo modelo falha quando um novo item não tem avaliações suficientes, com um SR baseado em conteúdo, onde as informações dos itens estão disponíveis, pode gerar recomendações mais acuradas e eficientes. Por exemplo, se você é um leitor frequente do Google News, o sistema recomenda novos artigos combinando as notícias mais populares lidas por usuários iguais a você e usando suas preferências calculadas pelo o que você clica. Como podemos ver, com esse tipo de recomendação híbrida, Filtragem Colaborativa é combinada com a baseada em conteúdo, ao invés de utilizar apenas uma das abordagens.

2.3 Considerações Finais

Como foi exposto neste capítulo, vimos a motivação da elaboração de Sistemas de Recomendação e quais benefícios eles agregam ao comércio eletrônico, uma vez implementados. Além disso, também foram abordados o problema de recomendação, suas atividades e notações, definidos formalmente e guiados por fontes científicas válidas.

Finalmente, foram apresentadas as classificações dos SRs, a saber:

- **Recomendação baseada em Conteúdo:** Utilizam metadados dos itens disponíveis e verificam se é do gosto do usuário, recomendando-os, em caso positivo.
- **Recomendação baseada em Filtragem Colaborativa:** Usam a percepção da inteligência coletiva como estratégia eficaz para obter as recomendações.
- **Recomendação baseada em Filtragem Híbrida:** Combinam diferentes abordagens para fornecer recomendações de maneira mais robusta.

Recomendação Baseada no Conteúdo

Particularmente neste trabalho, cujo objetivo foi desenvolver um SR baseado em conteúdo, cabe a nós debruçarmos em profundidade nessa tipologia. Entre os vários paradigmas existentes, destacamos dois, conforme adiantamos, que são: 1) Filtragem colaborativa e 2) Recomendação baseada no conteúdo. Nesta, os itens sugeridos levam em conta a atividade do usuário no passado e não a de usuários com preferências similares.

Nessa abordagem, a análise dos itens previamente visualizados pelo usuário juntamente com a realizada nos itens disponíveis no sistema guiam o processo de recomendação, conforme elaborou Pazzani em sua pesquisa (1999, apud Silva, 2014, p.10). Ainda de acordo com Silva (2014, p.10), a origem dos Métodos de Recomendação Baseados no Conteúdo remete às áreas de Recuperação da Informação e de Aprendizado de Máquina (AM).

3.1 Processo de Filtragem

Baseado nas leituras de Pasquale Lops et al (2011, p.75, tradução nossa) e Silva (2014, p.10-11), estabelece-se o processo de recomendação compilado em três etapas, que foram baseadas nas classificações de [9], [2] e [10], e podem ser estendidas para qualquer SR deste tipo:

- **Estruturação da Informação:** O conteúdo é extraído das fontes de informações e representado de modo a facilitar os passos posteriores. Segundo o autor, o analisador de conteúdo é o responsável pela representação do conteúdo de um determinado item. Essa representação estruturada fica armazenada no repositório de Itens Representados por metadados. Conforme mostra Silva (2014, p.10, tradução nossa), há diversas formas para representar o conteúdo, como: frases, categorias conceituais e outras, mas a mais comum é a BOW (Bag of Words, em tradução livre, saco de palavras). Esta abordagem pode ser feita por n-gram, quando há similaridade no radical dos termos adotados (em diversos graus) ou de acordo com a localização da palavra.
- **Geração do perfil do usuário:** Através do componente “Aprendiz do perfil”, os dados representativos de um determinado usuário, suas preferências e recusas, são coletados e analisados para a construção do seu perfil. O feedback, seja ele negativo ou positivo, fica armazenado no repositório, que posteriormente será revisitado na hora de sugerir um item interessante ao usuário.
- **Seleção de itens:** A partir do cruzamento de dados das duas etapas anteriores (Conteúdo dos Itens e Perfil do Usuário), chega-se ao julgamento de relevância binário ou contínuo.

Por sua vez, esse resultado gerará uma lista ranqueada de itens potencialmente interessantes para a recomendação.

3.2 Conhecendo o usuário

Tentar decifrar os gostos dos usuários – clientes em potencial – é a chave do processo de recomendação. Mencionada na seção anterior como uma das etapas, faz-se necessário dedicar mais algumas linhas a esse aspecto. O feedback mencionado como um dos resultados da segunda etapa pode ser extraído de forma explícita ou implícita, conforme delineou Pasquale Lops (et al. 2011, p.76, tradução nossa). Para obter o feedback explícito do usuário, há duas principais técnicas, segundo os autores:

- Gostar / Não gostar: através da escala binária, os itens passam a ser classificados como relevantes ou irrelevantes;
- Comentários textuais: podem ser úteis, mas podem sobrecarregar o usuário que deve lê-los e interpretá-los e, então, absorvê-los como negativos ou positivos

3.2.1 Cálculo de Similaridade

A combinação entre as duas variantes no processo de recomendação (Conteúdo dos Itens e Perfil do Usuário) é desenvolvida através de um cálculo de similaridade, no qual, quanto mais próximos forem os itens das preferências dos usuários, mais eficaz será a recomendação dos mesmos. Na literatura sobre o assunto, encontram-se as mais variadas formas de determinar essa similaridade, sendo as duas mais utilizadas, segundo Silva (2014, p. 12): a Similaridade Jaccard e a Similaridade Cosseno. Utilizamos esta última na concepção do protótipo desenvolvido neste trabalho, que pode ser definida por:

“Essa medida é mais utilizada, em uma representação BOW, com pesos nos termos calculados por sua importância no documento. O método de cálculo dos pesos é normalmente dado pela multiplicação $TF \times IDF$ ”, instrui o mesmo autor (2014, p. 13). Sendo TF = Frequência do Termo e IDF = Frequência Inversa do Documento.”

3.3 Benefícios e Limitações

Toda e qualquer estratégia de Sistema de Recomendação sempre terá seus prós e contras no mapeamento de usuários, influenciando no resultado final, que é a recomendação propriamente dita. Ainda assim, a abordagem baseada no conteúdo foi utilizada no protótipo que apresentamos. Gorakala traz a conhecimento, por exemplo, que 38% dos cliques no Google Notícias são de links recomendados (2016, p.10, tradução nossa).

Contudo, ainda que alguns dados corroborem sua eficácia diante de grandes *cases* no mercado, o modelo também possui suas limitações e pode ser aprimorado a cada dia. Elencamos abaixo duas das principais vantagens percebidas na revisão bibliográfica realizada para este estudo:

- **Transparência:** Faz-se necessário conhecê-lo individualmente e não em comunidade, como na abordagem de filtragem colaborativa;
- **Novo item:** Os sistemas de recomendação podem indicar um item que não foi avaliado ainda, desde que suas especificidades atendam às preferências do usuário previamente computadas. Dessa forma, o novo item no catálogo não precisa esperar uma avaliação de outro usuário, como acontece na abordagem colaborativa, para entrar no escopo de indicações.

Entre os pontos negativos da metodologia, cita-se a ampla variedade na representação de um item, devido à variedades de formas para discriminar um determinado objeto, dificultando na padronização da busca dos itens semelhantes. Silva [10] aponta ainda para outras limitações, como o fator de não surpreender o usuário, pelas buscas sempre baseadas nos mesmos termos e também para as altas dimensionalidades geradas na descrição dos produtos. Essa última problemática pode inviabilizar a aplicação do Sistema de Recomendação.

Para o futuro, já são pesquisadas maneiras de como eliminar as limitações dentro do campo de estudo dos Sistemas de Recomendação. Como sugere Gorakala [9], quando chegarmos à fase futurista desse campo de estudo, os sistemas de recomendação irão vigiar os usuários, levando em conta fatores como localização, horário, estado de espírito, ciclo de sono e outros na hora de recomendar coisas em tempo real. Enquanto essa realidade não se torna efetiva, faz-se necessário trabalhar com as metodologias já desenvolvidas, no caso deste trabalho, especificamente a que se baseia no conteúdo para recomendar itens aos usuários.

3.4 Considerações Finais

A recomendação baseada no Conteúdo se mostrou muito eficaz quando se trata de sugerir itens de acordo com o gosto individual do usuário, uma vez que ela utiliza metadados para pará-los com o perfil do mesmo.

A capacidade dessa abordagem de representar produtos a partir das suas descrições, utilizando o método TF-IDF, para gerar um *score*, assim, mapear com as descrições das preferências do usuário mostrou-se bastante simples, fácil de implementar e eficaz.

Um dos problemas nessa abordagem é que ela não gera indicações surpreendentes, uma vez que ela tem o objetivo de fornecer itens semelhantes ao que o usuário já aprecia.

Recomendação baseada em Filtragem Colaborativa

O termo “filtragem colaborativa” (FC) foi introduzido por Goldberg et al. em 1992, no qual elaborou que “a filtragem da informação pode ser mais efetiva quando humanos são envolvidos no processo de filtragem. [11]. O conceito atual de filtragem colaborativa que está estabelecido na literatura foi introduzido dois anos depois por Resnick et al [12]. Sua teoria era que usuários gostam do que usuários com a mesma opinião gostam, onde dois usuários são considerados com a mesma opinião quando eles atribuem notas semelhantes a itens avaliados por ambos.

Na prática, o algoritmo de FC funciona encontrando um conjunto de usuários com preferências e gostos similares a um usuário alvo. Utilizando esse conjunto menor de usuários similares, ele constrói uma lista ordenada de recomendações. Há várias maneiras de medir a similaridade entre dois usuários, que explicaremos detalhadamente mais à frente. É importante ressaltar que filtragem colaborativa não usa atributos ou descrições dos itens para recomendá-los, apenas são utilizadas preferências sobre aquele item.

Em um típico cenário de sistemas de FC, existe uma lista de m usuários $U = (u_1, u_2, u_3, \dots, u_m)$ e uma lista de n itens $I = (i_1, i_2, i_3, \dots, i_n)$. Cada usuário u_i possui uma lista de itens I_{u_i} , para os quais 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 chamado usuário alvo para o qual a tarefa de recomendação é realizada [4].

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 para o j -ésimo item. É importante salientar que todas as avaliações disponíveis devem ser representadas em valores numéricos.

Métodos de filtragem colaborativa podem ser agrupados basicamente em duas classes, baseadas em vizinhança (também conhecidas como baseadas em memória) e baseados em modelo. Nos métodos baseados em vizinhança, toda a matriz de avaliações é utilizada para prever notas para novos itens. Isto pode ser feito de duas maneiras, que são conhecidas como recomendação baseada no usuário e recomendação baseada no item. Esses sistemas usam heurísticas ou técnicas estatísticas para encontrar um grupo de usuário, chamados de vizinhos do usuário alvo das recomendações. Esse processo será melhor explicado nas seções posteriores.

Em contraste, os métodos baseados em modelos utilizam as avaliações para gerar um modelo preditivo (estatístico). Pesquisas recentes em sistemas de recomendação mostram que FC baseada em modelo prediz melhor do que baseada em vizinhança [13]. É sabido também que a qualidade das predições não implica, por si só, na satisfação do usuário. [14]

Um ponto a ser considerado é que a recomendação baseada em modelo pode levar à super

especialização, que é facilmente contornada e em muitos sistemas requerida. Uma grande vantagem da abordagem baseada em memória é a sua fácil implementação, possibilitando testar várias heurística. Além disso, simplicidade leva a robustez.

4.1 Filtragem baseada em Vizinhaça (Memória)

Filtragem Colaborativa Baseada em Memória pode ser dividida em duas etapas (métodos) principais: recomendação baseada no usuário e recomendação baseada no item. A primeira seleciona um usuário em particular, encontra usuários que são similares a aquele usuário baseado na semelhança das avaliações e recomenda itens similares aos quais esses usuários gostaram. A recomendação baseada no item escolhe um item, encontra usuários que também gostaram daquele item, e recomenda itens similares aos quais esses usuários preferem. Em resumo:

- Filtragem baseada no usuário: “Usuários que são similares a você gostam de ...”
- Filtragem baseada no item: “Usuários que gostam deste item também gostam de ...”

De acordo com [15], há 5 principais fatores que devem ser levados em conta ao escolher uma dessas duas abordagens:

- Acurácia: Que depende do ratio entre usuários e itens. Se temos vários usuários e não muitos itens, a abordagem baseada no item é preferível (como a Amazon.com faz, por exemplo).
- Eficiência: A complexidade depende também do ratio entre usuários e itens. Seja $a = R/U$ e $b = R/I$ onde R, U, I são o número de avaliações, usuários e itens. Então a complexidade de abordagens baseadas no usuário é proporcional a a^2/I , enquanto que baseadas no item a b^2/U .
- Estabilidade: Temos que levar em conta o que cresce mais rápido, seja o número de usuários ou o número de itens. Se, por exemplo, o número de usuários é mais estático, então as abordagens baseadas no usuário aprendem mais rápido, vice-versa.
- Acaso: É uma métrica de quão surpreendente uma recomendação é. Em outras palavras, essa medida informa o montante informação relevante que é nova para o usuário em uma recomendação. Portanto, otimizar esse fator em algoritmos baseados no item é menos preferível, porque as avaliações de um item são previstas pelo uso de notas de itens similares, o que leva a recomendações seguras porém menos surpreendentes ou interessantes.

Em ambos os casos é criada uma matriz usuário-item a partir de todo o *dataset* disponível. Depois de construída, uma outra matriz é gerada, chamada de matriz de similaridade. Os valores gerados na abordagem baseada no item são medidas realizadas observando todos os usuários que tenham gostado de itens semelhantes. Para a recomendação baseada no usuário, os valores de similaridade são gerados observando todos os usuários avaliam de maneira semelhante.

4.1.1 Cálculo de similaridade

Na essência de algoritmos de filtragem colaborativa há uma forte suposição de que usuários similares gostam de itens similares. Dessa maneira, surge a questão: como definimos e identificamos usuários similares para encontrar possíveis itens que possamos recomendar? Para resolver essa questão, nós temos que considerar os dados disponíveis na matriz esparsa de usuários-itens, já que muitos usuários apenas avaliaram uma pequena quantidade dos itens disponíveis.

No sistema de recomendação do GroupLens, o primeiro sistema automático, a Correlação Pearson foi sugerida como uma boa medida de similaridade [12]. Várias medidas de similaridade foram sugeridas ao longo do tempo, como Distância Euclidiana, Coseno, entre outras. Muito se tem discutido sobre qual é a melhor medida para um sistema de recomendação. Há autores que dizem que a Correlação Pearson é melhor para algoritmos que usam métodos baseados no usuário [16], enquanto que para baseadas no item a Coseno [17].

4.1.1.1 Correlação de Pearson

A função de similaridade Pearson é uma das mais conhecidas para algoritmos de filtragem colaborativa baseada no usuário. Essa medida representa a similaridade em uma escala de -1 a 1, onde um valor alto e positivo sugere uma alta correlação, um alto valor negativo sugere o inverso, e por último um valor 0 indica nenhuma correlação. A equação pode ser definida como:

$$s(u, v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \mu_u) \cdot (r_{vi} - \mu_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \mu_u)^2} \cdot \sqrt{\sum_{i \in I_{uv}} (r_{vi} - \mu_v)^2}}$$

Onde $s(u, v)$ configura similaridade entre os usuários u e v , I_{uv} retrata o conjunto de itens avaliados por ambos os usuários u e v , r_{ui} e r_{vi} representam as notas do usuário u e v para o item i , e μ_u e μ_v representam, respectivamente, a média do usuário u e v , entre todos os itens avaliados.

Para a recomendação baseada no usuário, a fórmula é levemente modificada como a seguir:

$$s(i, j) = \frac{\sum_{u \in U_{ij}} (r_{ui} - \mu_i) \cdot (r_{uj} - \mu_j)}{\sqrt{\sum_{u \in U_{ij}} (r_{ui} - \mu_i)^2} \cdot \sqrt{\sum_{u \in U_{ij}} (r_{uj} - \mu_j)^2}}$$

Onde $s(i, j)$ representa a similaridade entre os itens i e j , U_{ij} representa o conjunto de usuários que avaliaram os itens i e j , e μ_i e μ_j representam, respectivamente, a média das avaliações dos itens i e j , entre todos os usuários que avaliaram os mesmos.

4.1.1.2 Distância Euclidiana

Esta função de similaridade é definida como:

$$s(u, v) = \sqrt{\frac{\sum_{i \in I_{uv}} (r_{ui} - r_{vi})^2}{|I_{uv}|}}$$

Onde $s(u,v)$ representa a similaridade entre os usuários u e v , considerando todos os itens comumente avaliados.

No caso de algoritmos de recomendação baseados no item, a fórmula se torna:

$$s(i, j) = \sqrt{\frac{\sum_{u \in U_{ij}} (r_{iu} - r_{ju})^2}{|U_{ij}|}}$$

Onde $s(i,j)$ representa a similaridade entre os itens i e j , considerando todos usuários que avaliaram tais itens.

4.1.1.3 Coseno

Essa função quantifica a similaridade entre dois itens com base no cosseno do ângulo formado pelos vetores que representam cada um dos itens. A escala usada nestes caso é igual à usada na Pearson, onde +1 e -1 representam alta (direta e inversa) correlação, e zero representa nenhuma correlação. Trazendo para o problema de recomendação, a equação se torna:

$$s(u, v) = \frac{\sum_{i \in I_{uv}} r_{ui} \cdot r_{vi}}{\sqrt{\sum_{i \in I_{uv}} r_{ui}^2} \cdot \sqrt{\sum_{i \in I_{uv}} r_{vi}^2}}$$

Onde $s(u,v)$ representa a similaridade entre os usuários u e v , considerando todos os itens em comum avaliados. No caso de algoritmos baseados no item, temos a seguinte equação:

$$s(i, j) = \frac{\sum_{u \in U_{ij}} r_{ui} \cdot r_{uj}}{\sqrt{\sum_{u \in U_{ij}} r_{ui}^2} \cdot \sqrt{\sum_{u \in U_{ij}} r_{uj}^2}}$$

4.2 Filtragem Baseada em Modelo

Para sanar as limitações da abordagem anterior, pesquisadores começaram a aplicar vários métodos para aumentar a performance da recomendação, como: modelos probabilísticos, de aprendizagem de máquina (modelos supervisionados e não-supervisionados), abordagens utilizando matrizes como Fatoração de Matrizes e Decomposição em Valores Singulares [2]. E em abordagens baseada em modelos, utilizando os dados disponíveis, um modelo é construído

para que os pesos sejam aprendidos automaticamente. Logo, novas predições são geradas utilizando esses pesos e os resultados finais ordenados em uma ordem específica, gerando assim recomendações relevantes.

Modelos que utilizam fatoração de matrizes emergiram como estado da arte nessa classe de técnicas. Basicamente, fatoração de matriz caracteriza ambos usuários e itens por vetores de fatores inferidos dos padrões de avaliações. Alta correspondência entre item e usuário leva a uma recomendação. Esses métodos se tornaram famosos atualmente porque combinam boa escalabilidade com acurácia. Em adição, eles oferecem mais possibilidades para modelar várias situações do mundo real [18]. Outros métodos incluem baseados em cluster [19], classificadores Bayesianos [20] e baseados em regressão [21]. Por último e não menos importante, existe um método chamado SlopeOne [22] que consegue conciliar rápida computação com acurácia. Veremos a seguir o método mais utilizado dentro dessa abordagem: a Decomposição em valores singulares.

4.2.1 Decomposição em valores singulares

Decomposição em valores singulares (SVD, acrônimo em inglês) no contexto de sistemas de recomendação é usado como um algoritmo de filtragem colaborativa. SVD é uma técnica de fatoração de matriz que é usualmente utilizada para reduzir o número de features (propriedades) de um conjunto de dados de N para K , onde $K < N$. Para o propósito de recomendação, no entanto, estamos apenas interessados na fatoração da matriz mantendo a mesma dimensionalidade.

A fatoração de matriz é feita utilizando a matriz usuário-item de avaliações. Uma descrição básica e de alto nível é que a fatoração da matriz pode ser pensada como encontrar duas matrizes cujo o produto é a matriz original.

Cada item é representado por um vetor q_i . Similarmente, cada usuário é representado por um vetor p_u , e o produto escalar desses dois vetores é a avaliação esperada.

$$\text{nota esperada} = \check{r} = q_i^T p_u$$

Esses vetores possuem os fatores latentes que expõem suas características mais importantes. q_i e p_u podem ser encontrados de uma maneira que a diferença quadrática de erro entre o produto vetorial de ambos e a nota obtida da matriz original é mínima.

$$\min(p, q) \sum_{(u,i) \in K} (r_{ui} - q_i^T p_u)^2$$

Para o modelo ser capaz de generalizar, e não ficar super especializado a partir do conjunto de treino, é introduzido um termo de penalidade para a equação de minimização, que é representado por um fator de regularização lambda multiplicado pela soma dos quadrados das magnitudes dos vetores de usuários e itens.

$$\min(p, q) \sum_{(u,i) \in K} (r_{ui} - q_i^T p_u)^2 + \lambda (\|q_i\|^2 + \|p_u\|^2)$$

Para ilustrar o quão útil é esse fator, devemos imaginar um caso extremo onde uma avaliação baixa é dada por um usuário para um item, sendo essa a única avaliação realizada por

esse usuário até o momento. O algoritmo irá minimizar o erro dando a q_i um valor alto. Em consequência, todas as outras notas desse usuário para outros itens serão baixas, o que é intuitivamente errado. Adicionando a magnitude dos vetores para a equação, dar valores altos irá minimizar a equação e tal situação relatada será evitada.

Para reduzir o erro entre a nota prevista e o valor real, o algoritmo faz uso de algumas características do *dataset*. Particularmente, para cada par de usuário item (u,i) são extraídos 3 parâmetros: μ , que é a média das notas para todos os itens, b_i , que é a média de notas para item i menos μ e b_u , que é a média das notas dadas pelo usuário u menos μ . A nota esperada será dada pela equação:

$$\tilde{r}_{ui} = q_i^T p_u + \mu + b_i + b_u$$

A equação completa de minimização de erros é:

$$\min(p, q, b_i, b_u) \sum_{(u,i) \in K} (r_{ui} - q_i^T p_u - \mu - b_i - b_u) + \lambda (\|q_i\|^2 + \|p_u\|^2 + b_i^2 + b_u^2)$$

4.3 Considerações Finais

Focamos em abordagens baseadas em vizinhança que envolvem vários cálculos de similaridade entre usuários e produtos. Todo esse esforço se deu no sentido de representar usuários e itens em vetores e encontrar as medidas de similaridades e identificar itens semelhantes às preferências do usuário. O principal objetivo de algoritmos que usam similaridade é calcular o peso das preferências dos usuários em relação aos itens disponíveis e, então, usá-lo para a atividade de recomendação.

Essas abordagens foram bem sucedidas durante anos e continuam sendo utilizadas até hoje, mas elas têm suas limitações. Como os dados têm de ser carregados no sistema para o cálculo de similaridades, abordagens em vizinhança também são conhecidas como filtragem baseadas em memória. Essas abordagens são muito lentas em cenários da vida real, quando o montante de dados é muito grande e todos os dados têm de ser carregados. Outra limitação é que os pesos calculados não são aprendidos automaticamente, como em aplicações de aprendizagem de máquina.

Por outro lado, abordagens baseadas em modelo tentam simular o preenchimento da matriz de similaridades e adivinhar, por meios de técnicas de aprendizagem de máquina e além de outras, o quanto usuário irá gostar daquele item. Portanto, tornam-se mais suscetíveis à dinamicidade de sistemas de comércio online reais em que a volatilidade de surgimento e desaparecimento de itens é constante.

Metodologia e Resultados

5.1 Conjunto de Dados

Os testes foram conduzidos sobre um conjunto de dados real, gentilmente cedidos por uma empresa recifense. Devido a um acordo de confidencialidade não podemos revelar seu nome. O conjunto de dados foi obtido a partir de um E-commerce que realiza vendas de roupas femininas. A tabela 5.1 mostra as configurações originais do conjunto de dados.

Nome	#Usuários	#Items	Escala das Notas	Esparsidade
reviews.csv	6158	3269	1..5	0.99925

Tabela 5.1 Configurações iniciais do conjunto de dados

Um pré-processamento foi feito na base de dados, pois foi notado que pouco usuários avaliavam mais de uma vez, o que prejudicava a acurácia dos algoritmos, e inviabilizando a comparação com a literatura atual. Portanto foi considerado apenas usuários que realizaram mais que sete avaliações. A tabela 5.2 mostra a configuração do conjunto pós-processamento.

Nome	#Usuários	#Items	Escala das Notas	Esparsidade
reviews7.csv	70	661	1..5	0.95893

Tabela 5.2 Configurações pós processamento do conjunto de dados

5.2 Métricas de avaliação

Há várias métricas listadas na literatura como: 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, que são métricas estatísticas de acurácia. Além delas, há métricas de classificação, que medem a frequência com que o sistema de recomendação fez recomendações corretas ou incorretas. Dentre essas métricas podemos citar a medida F (F-Mesure), precisão (Precision) e cobertura (Recall) e curva ROC (Receiver Operating Characteristic). Cada métrica utilizada depende do propósito do algoritmo e do objetivo da medição [23].

O procedimento de avaliar desempenho de sistemas de recomendação usando métricas estatísticas apresenta algumas vantagens como: fácil computação e propriedades estatísticas diretas. Portanto, neste estudo uma das métricas que vamos considerar, é a métrica estatística mais usada, MAE, pois ela é menos sensível a exemplos díspares nos dados (outliers) do que RMSE por exemplo, e sua interpretação é direta [24].

Erro médio absoluto é uma métrica usada para computar a média de todos os valores absolutos das diferenças entre a nota real da avaliação e a prevista. Quanto mais baixo o MAE melhor a acurácia. Computar o MAE é obtido pela aplicação direta da equação a seguir:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\check{r}_i - r_i|$$

Como métricas de classificação, foram utilizadas nesse estudo precisão e cobertura (Precision and Recall) que são medidas de avaliação clássicas originadas na área de RI, porém adaptadas para a área de recomendação.

Para utilizar essa métrica, a primeira coisa a ser feita é converter as avaliações numéricas em binárias (recomendar ou não recomendar, por exemplo). Itens cuja predição for recomendar serão mostrados ao usuário e os restantes escondidos. Cada item desses pode ser relevante ou irrelevante para o usuário. Nós então temos quatro possíveis resultados para os itens recomendados e escondidos, como mostrado na tabela 5.3

	Recomendado	Não Recomendado	Total
Relevante	RR	RN	R = RR + RN
Não Relevante	FP	NN	IR = FP + NN
Total	REC = RR + FP	NREC = RN + NN	N = R + IR = REC + NREC

Tabela 5.3 Possíveis resultados para os itens recomendados e escondidos

Portanto, precisão é a fração de todos os itens recomendados que são relevantes, no caso:

$$\text{precisao} = \frac{RR}{RR + FP} = \frac{RR}{REC}$$

E cobertura é a fração de todos os itens relevantes que são recomendados, ou seja:

$$\text{cobertura} = \frac{RR}{RR + RN} = \frac{RR}{R}$$

Em sua derivação mais útil dessa métrica é precisão e cobertura em K, que é a proporção de itens recomendados no conjunto de top-K itens que são relevantes. A interpretação dessas métricas é a seguinte: suponha que a precisão em 10 numa recomendação de top-10 itens é 80%. Isso significa que 80% das recomendações do sistema são relevantes ao usuário. Portanto matematicamente:

$$\text{precisao}@K = \frac{\# \text{ de itens recomendados em K que são relevantes}}{\# \text{ de itens recomendados em K}}$$

Suponha agora que a cobertura em 10 encontrada foi de 40% na recomendação top-10. Isso significa que 40% do número total de itens relevantes aparecem nos resultados da recomendação top-10. Portanto, de maneira formal:

$$cobertura@K = \frac{\# \text{ de itens recomendados em } K \text{ que são relevantes}}{\# \text{ total de itens relevantes}}$$

5.3 Procedimento Experimental

Para a realização dos experimentos foi utilizada a biblioteca Surprise Lib, que é uma framework de avaliação e construção de sistemas de recomendação. Informações mais detalhadas sobre o propósito e utilização pode ser encontrada em <http://surpriselib.com/>. As razões para a escolha da Surprise Lib foram as seguintes:

- Algoritmos relevantes já implementados e baseados em fontes científicas válidas
- Parâmetros dos algoritmos são fáceis de modificar para alterar a performance dos mesmos
- Documentação intuitiva
- Fácil de manipular conjunto de dados

Além disso, para testar o desempenho futuro e estimar o erro de previsão, temos que corretamente particionar o conjunto de dados original em subconjuntos de treinamento e teste. O conjunto de teste deve ser diferente e independente do conjunto de treinamento a fim de obter uma estimativa confiável do verdadeiro erro. Em [4], Sarwar et al, experimentaram, diferentes proporções de divisão dos dados variando de 20% para testes até 90% em incrementos de 10%. Seguindo esses resultados o primeiro passo foi dividir 80% dos dados para treinamento e 20% para testes de forma aleatória em relação aos usuários (pois no estudo citado essa foi a divisão ideal encontrada).

Os testes foram realizados de maneira offline [25] que é quando os experimentos são feitos utilizando dados pré-coletados. Dessa maneira podemos simular o comportamento dos usuários que interagem com o sistema de recomendação. Experimentos offline são atrativos pois não requerem interação com um usuário real e nos permite, testar uma gama de algoritmos disponíveis a um custo baixo.

Portanto, o principal objetivo de testes offline é filtrar as abordagens inapropriadas [25], deixando assim um pequeno conjunto de candidatos que seguem para uma próxima fase, por exemplo, refinamento de parâmetros.

O teste da abordagem baseada em conteúdo foi feita de maneira manual, e os perfis do usuário para serem pareados com os metadados dos itens foram gerados artificialmente. Os clusters formados a partir da representação dos produtos pelo score do TF-IDF eram corretos, uma vez que não há métodos simples e diretos de validar a formação de clusters. Portanto, os resultados não serão discutidos nem mostrados na seção 5.4

5.4 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:

5.4.1 Efeitos da similaridade

Foram implementados três algoritmos diferentes de similaridade: cosseno, correlação e MSD. Os testes foram conduzidos em um sistema de recomendação baseado em memória com as duas abordagens citadas no seção 4.1. A tabela 5.4 mostra que o algoritmo baseado no item que utiliza a similaridade msd apresentou a melhor acurácia em relação a outras técnicas, enquanto que o cosseno obteve o pior desempenho em ambas as abordagens. Por essa razão a abordagem baseada em memória e no item utilizando a similaridade msd foi escolhida para a próxima fase, à saber, comparar com uma abordagem baseada em modelo.

	COSSENO	PEARSON	MSD
Baseado no usuário	0.7044	0.6744	0.6788
Baseado no item	0.7136	0.6720	0.6648

Tabela 5.4 Cálculo do MAE Médio para cada similaridade e abordagem

5.4.2 Comparativo da acurácia entre abordagens

Um experimento comparativo foi realizado para determinar como as abordagens se comportavam em relação a acurácia dado as características do conjunto de dados. Para tal, utilizamos a abordagem baseada em memória com a função de similaridade msd e o algoritmo baseado em modelo, Decomposição em Valores Singulares, citada na seção 4.2.1.

Nas figuras 5.1 e 5.2 temos uma porção do conjunto de predições realizadas por ambas as abordagens já citadas, onde:

- uid: É o índice do usuário
- iid: É o índice do item
- rui: É a nota real dada pelo usuário
- est: É a nota estimada pela abordagem
- err: É o erro absoluto entre a nota estimada e a nota real

Perceba que ambos os algoritmos foram testados com o mesmo conjunto de dados, portanto apresentam os mesmos índices de usuário e item.

	uid		liid	ruil	est	err
0	c81eb60468aff456cd070dff07e1859d	f501d2693c06f905f4c210b495748a79	2.0	4.138470	2.138470	
1	851451fef163d0865a5d4e7c92d6d3b2	2dabf68638e6540704cb179d61594faf	4.0	4.593409	0.593409	
2	5b9baae07db9d6edfad1896c693c3908	45a9c220aa008c10a0e581f44c0627e9	5.0	4.454811	0.545189	
3	5b85936fbbbf8413870d86a72731698f	5990653da56b4075bff360efcc52383f	4.0	4.463161	0.463161	
4	6395d30218015a535e973c2a974ef185	c12437a105d67a1e5fc09827346e2b49	5.0	4.406408	0.593592	

Figura 5.1 Parte da lista de predições obtidas pelo algoritmo de Decomposição em Valores Singulares

	uid		liid	ruil	est	err
0	c81eb60468aff456cd070dff07e1859d	f501d2693c06f905f4c210b495748a79	2.0	4.304668	2.304668	
1	851451fef163d0865a5d4e7c92d6d3b2	2dabf68638e6540704cb179d61594faf	4.0	4.304668	0.304668	
2	5b9baae07db9d6edfad1896c693c3908	45a9c220aa008c10a0e581f44c0627e9	5.0	4.304668	0.695332	
3	5b85936fbbbf8413870d86a72731698f	5990653da56b4075bff360efcc52383f	4.0	4.304668	0.304668	
4	6395d30218015a535e973c2a974ef185	c12437a105d67a1e5fc09827346e2b49	5.0	4.083333	0.916667	

Figura 5.2 Parte da lista de predições obtidas pelo algoritmo baseado em vizinhança

As estimativas mostraram-se bastante acuradas como podemos ver nas figuras 5.3 e 5.4

	uid		liid	ruil	est	err
16	bb875e248ad05309da38e6ee7c23e460	47af7fca91dd663217384fadcd58c8ab	5.0	5.000000	0.000000	
159	e129baef45c7c11ffc683e70e46e0567	751d51528afe5e6f7fe95dece4ed32ba	5.0	5.000000	0.000000	
34	c81eb60468aff456cd070dff07e1859d	d8030afe156a4de429b42187a7bb28eb	4.0	3.997715	0.002285	
19	5f4f537738a86f28d1b17d293fe55780	c7c3e78e3c9d26cc1158a8735d548eaa	4.0	4.012874	0.012874	
97	710474d4f5b8c047af09ffa1f4cfd352	b8536aeea46484dd753df6bab1641dc7	4.0	4.025631	0.025631	
133	345ff7ce8635a08866e7f4ad65aa5492	6d9014db7e6634821274df396bc3e372	4.0	4.030364	0.030364	
55	4702ca7bca0a66d54774138d70e27ebc	757334847f188c6304b84ec6b018a965	5.0	4.945249	0.054751	
18	4702ca7bca0a66d54774138d70e27ebc	757334847f188c6304b84ec6b018a965	5.0	4.945249	0.054751	
186	4702ca7bca0a66d54774138d70e27ebc	757334847f188c6304b84ec6b018a965	5.0	4.945249	0.054751	
10	719df4c75427132d0edc43ababa56c3c	0c836be97564457619349887cf51b3ba	4.0	3.930902	0.069098	

Figura 5.3 As dez melhores predições do algoritmo baseado em vizinhança

Porém algumas vezes são bem díspares da realidade como pode ser vistos nas figuras 5.5 e 5.6, que mostram as dez piores estimativas de cada abordagem.

	uid	uid	liid	rui	est	err
48	d3201df577e2b485907cf9f3f72be83d	64af19031c975bb2383a39153ff053f6	4.0	3.364008	0.635992	
18	4702ca7bca0a66d54774138d70e27ebc	757334847f188c6304b84ec6b018a965	5.0	4.945249	0.054751	
159	e129baef45c7c11ffc683e70e46e0567	751d51528afe5e6f7fe95dece4ed32ba	5.0	5.000000	0.000000	
93	45d29ee06862810f6f1c53d7f1c01ac5	2456b9cd2668fa69e3c7ecd6f51866bf	5.0	4.362399	0.637601	
96	da5dbodd1779202d57655c13775b1325	bb6961095e9a6dc1aac2af24ea88fc52	5.0	4.849187	0.150813	
83	a7ee9587fdf8c059019cfdef125db7fc	01b6397888c09d84f3dc89d807aa1004	5.0	4.834555	0.165445	
186	4702ca7bca0a66d54774138d70e27ebc	757334847f188c6304b84ec6b018a965	5.0	4.945249	0.054751	
19	5f4f537738a86f28d1b17d293fe55780	c7c3e78e3c9d26cc1158a8735d548eaa	4.0	4.012874	0.012874	
102	e129baef45c7c11ffc683e70e46e0567	d976f9a7a2ee232ce4143426a181bd0a	5.0	4.729254	0.270746	
72	71fef66c390657fb2d3e7f601b0ea770	4d42d2f5010c1c13f23492a35645d6a7	4.0	4.346885	0.346885	

Figura 5.4 As dez melhores predições do algoritmo baseado em modelo

	uid	uid	liid	rui	est	err
0	c81eb60468aff456cd070dff07e1859d	f501d2693c06f905f4c210b495748a79	2.0	4.304668	2.304668	
85	5b9baae07db9d6edfad1896c693c3908	6b6769819569f3c8ef0785f500832831	2.0	4.304668	2.304668	
16	f6314f0b9c0ad5235541636f7c3a69e5	23d79697ad3c613f5d6a710258591024	2.0	4.304668	2.304668	
173	7b5b88b40872d3f46481ca0f222340b9	d976f9a7a2ee232ce4143426a181bd0a	5.0	2.000000	3.000000	
194	20839fc8b6c28387a060e4096e77b9f4	62306523b3c77c077b2938f0d6ab91f5	1.0	4.304668	3.304668	
27	741b8e45fe6e0cd5e2f46d9df24179a4	58155fc7f528a19b9fc3599b65047290	1.0	4.304668	3.304668	
169	d3201df577e2b485907cf9f3f72be83d	2c60e40b399dc55d8b755ec6b5d09f8a	1.0	4.304668	3.304668	
165	a8e2d80a289e137ac8e1a40a9abd6d97	c647f2a6f34278b30c28af729766bdd4	1.0	4.304668	3.304668	
113	a8e2d80a289e137ac8e1a40a9abd6d97	96e76cc974cf7bdcc09b86bfad3eee35	1.0	4.304668	3.304668	
78	a7ee9587fdf8c059019cfdef125db7fc	ab38e9193c5e03752556a0a238ed5080	1.0	4.304668	3.304668	

Figura 5.5 As dez piores predições do algoritmo baseado em vizinhança

	uid	lid	ru	est	err
133	7238f581cc164fecc15877e0863f5d49	a625f892fe81b6b562c3f08c163307d5	1.0	3.078598	2.078598
0	c81eb60468aff456cd070dff07e1859d	f501d2693c06f905f4c210b495748a79	2.0	4.138470	2.138470
145	741b8e45fe6e0cd5e2f46d9df24179a4	ff12592a08c792faa97fd3d2dc40f1cd	2.0	4.155450	2.155450
85	5b9baae07db9d6edfad1896c693c3908	6b6769819569f3c8ef0785f500832831	2.0	4.418309	2.418309
165	a8e2d80a289e137ac8e1a40a9abd6d97	c647f2a6f34278b30c28af729766bdd4	1.0	3.516420	2.516420
113	a8e2d80a289e137ac8e1a40a9abd6d97	96e76cc974cf7bdcc09b86bfad3eee35	1.0	3.516420	2.516420
169	d3201df577e2b485907cf9f3f72be83d	2c60e40b399dc55d8b755ec6b5d09f8a	1.0	3.817673	2.817673
27	741b8e45fe6e0cd5e2f46d9df24179a4	58155fc7f528a19b9fc3599b65047290	1.0	4.155450	3.155450
194	20839fc8b6c28387a060e4096e77b9f4	62306523b3c77c077b2938f0d6ab91f5	1.0	4.645029	3.645029
78	a7ee9587df8c059019cfdef125db7fc	ab38e9193c5e03752556a0a238ed5080	1.0	4.692867	3.692867

Figura 5.6 As dez piores predições do algoritmo baseado em modelo

Além disso, podemos perceber que os algoritmos tem diferentes comportamentos em suas estimativas como pode ser observado na figura 5.8, por exemplo, um dos comportamentos esperados é que os algoritmos baseados em vizinhança tenha suas estimativas frequentes concentradas ao redor da média das avaliações e que o SVD acerte valores mais presentes nas extremidades. Além disso, a figura 5.7 mostra as estimativas do SVD para os dez piores resultados da abordagem em comparação.

	uid	lid	ru	est	err
0	c81eb60468aff456cd070dff07e1859d	f501d2693c06f905f4c210b495748a79	2.0	4.138470	2.138470
85	5b9baae07db9d6edfad1896c693c3908	6b6769819569f3c8ef0785f500832831	2.0	4.418309	2.418309
16	f6314f0b9c0ad5235541636f7c3a69e5	23d79697ad3c613f5d6a710258591024	2.0	3.354191	1.354191
173	7b5b88b40872d3f46481ca0f222340b9	d976f9a7a2ee232ce4143426a181bd0a	5.0	4.085931	0.914069
194	20839fc8b6c28387a060e4096e77b9f4	62306523b3c77c077b2938f0d6ab91f5	1.0	4.645029	3.645029
27	741b8e45fe6e0cd5e2f46d9df24179a4	58155fc7f528a19b9fc3599b65047290	1.0	4.155450	3.155450
169	d3201df577e2b485907cf9f3f72be83d	2c60e40b399dc55d8b755ec6b5d09f8a	1.0	3.817673	2.817673
165	a8e2d80a289e137ac8e1a40a9abd6d97	c647f2a6f34278b30c28af729766bdd4	1.0	3.516420	2.516420
113	a8e2d80a289e137ac8e1a40a9abd6d97	96e76cc974cf7bdcc09b86bfad3eee35	1.0	3.516420	2.516420
78	a7ee9587df8c059019cfdef125db7fc	ab38e9193c5e03752556a0a238ed5080	1.0	4.692867	3.692867

Figura 5.7 Resultado da abordagem em modelo sobre as 10 piores predições do algoritmo baseado em vizinhança

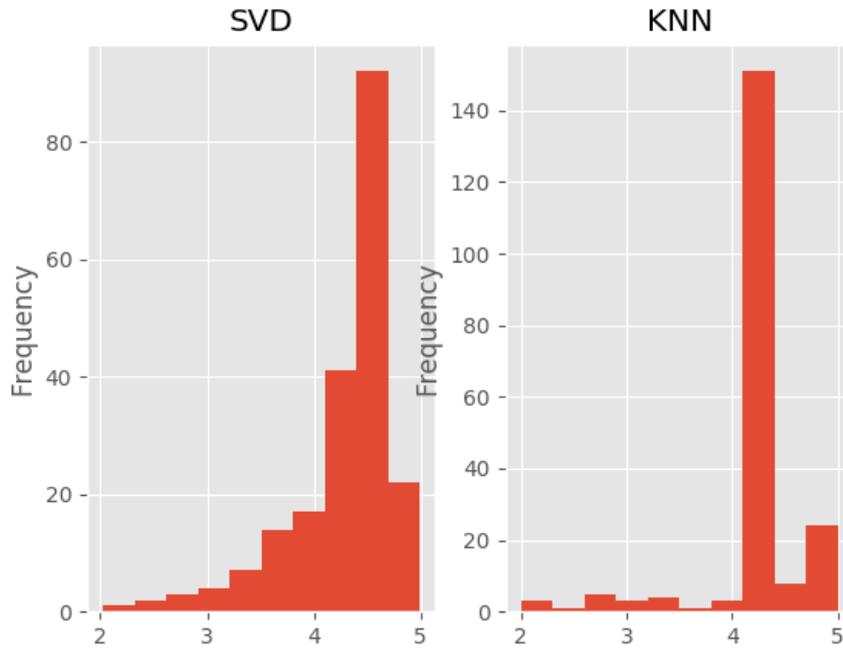


Figura 5.8 Comportamento das estimativas dos algoritmos

Outro questionamento que também foi respondido nesse estudo comparativo foi: qual dessas abordagens apresenta melhor acurácia quando poucos itens são avaliados? (Problema de partida a frio)

	Erro Médio
SVD	0.72295216075562396
KNN	0.81478792842429204

Tabela 5.5 Erro médio para o problema de partida a frio

5.5 Experimentos de Qualidade

Com a finalidade de comparar a qualidade das recomendações entre sistemas baseados em modelo e memória, novos experimentos comparativos foram realizados. O objetivo foi medir os valores de precisão e cobertura em K para cada uma das abordagens.

O primeiro passo foi adaptar essas métricas para área de sistemas de recomendação. Portanto, assumimos que qualquer valor das avaliações reais do usuário acima de 3.5 corresponde a um item relevante e qualquer uma avaliação abaixo disso é irrelevante. Um item relevante corresponde a um bom item para se recomendar.

O limiar de 3.5 foi escolhido devido a distribuição das notas, porém há técnicas mais avançadas de determinar esse limiar.

5.5.1 Sensibilidade ao variar o tamanho do conjunto de treinamento

A variação no tamanho do conjunto de treinamento foi alcançada variando o k em uma validação cruzada k -fold. O quanto o k cresce o tamanho do conjunto de treinamento também aumenta. O resultado é mostrado nas tabelas 5.6 e 5.7.

k Fold	Precisão@10	Cobertura@10
2	0.83229	0.96979
4	0.84944	0.97748
6	0.84885	0.98009
8	0.85274	0.98093
10	0.85950	0.98416

Tabela 5.6 Precisão e Cobertura para abordagem em vizinhança

k Fold	Precisão@10	Cobertura@10
2	0.89504	0.85257
4	0.91559	0.86170
6	0.92265	0.88029
8	0.92580	0.88716
10	0.93841	0.89297

Tabela 5.7 Precisão e Cobertura para abordagem em modelo

5.5.2 Sensibilidade ao variar o tamanho do conjunto de itens recomendados

Por último, foi investigado o efeito de se mudar o tamanho da lista de recomendação (ou seja, o K da precisão e cobertura). Intuitivamente, quanto maior o K maior a cobertura. Como mostrado na tabela 5.8, o valor da precisão@1 é aproximadamente 0.70 e a precisão@3 é aproximadamente 0.98, o que significa que para a maioria dos casos, recomendações corretas aparecem cedo nas listas de recomendações.

K	Abordagem em Modelo	Abordagem em Vizinhança
10	0.98447	0.98262
7	0.98440	0.98235
5	0.98435	0.98239
3	0.98430	0.98023
1	0.70111	0.70806

Tabela 5.8 Métrica de cobertura para abordagens em modelo e vizinhança variando-se o K (tamanho da lista de recomendação)

5.6 Considerações finais

Para abordagens baseadas em conteúdo percebemos que ela é bastante útil quando aliada a uma outra abordagem de recomendação, devido a sua difícil testabilidade e por causa que essa abordagem não gera recomendações que surpreendam o usuário, apenas recomendações do mesmo conteúdo do produto em questão ou do mesmo gosto do usuário.

Para algoritmos de filtragem colaborativa 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 para algoritmos baseados em memória. Além disso deve-se escolher sabiamente entre abordagem baseada em usuário ou no item a depender da proporção de usuários e itens na base do sistema.

Do outro lado, as abordagens baseadas em modelos tentam preencher essa matriz de similaridade de outra maneira. Eles atacam a tarefa de "adivinhar" o quanto o usuário irá gostar daquele item, o que é útil para casos de adversidade como o problema da partida a frio. Para isso eles utilizam diversas técnicas como a decomposição em fatores singulares ou aprendizagem de máquina.

As diferenças entre os dois tipos de abordagens à cerca da qualidade das recomendações, acurácia e detalhes de implementação mostram que os sistemas baseados modelo se mostram mais interessantes uma vez que itens emergem muito rapidamente, e também desaparecem rapidamente, portanto faz sentido que o sistema desenvolva uma maneira esperta de detectar quando um novo produto é relevante para o usuário ao invés de ter uma medida fixa como uma similaridade.

Portanto um sistema de recomendação que engloba a abordagem baseada em conteúdo somada a um algoritmo de filtragem colaborativa baseada em modelo são capazes de resolver os dois maiores desafios da área de SR: produzir recomendações de qualidade com alto desempenho e escalabilidade.

Conclusão 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 proativamente 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 interessante.

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 como estratégia eficaz para obter recomendação. Observamos que os sistemas colaborativos baseados em modelo, em 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 outros métodos e critérios. 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 [26]. É esperado que as recomendações tenham certo nível de novidade, surpresa, descoberta e métricas específicas para avaliar tais critérios [26].

Diante do exposto, considera-se que os sistemas apresentados são factíveis e adequados para implementação do sistema de recomendação de vestuário. A razão é a sua simplicidade, escalabilidade e qualidade nas predições mesmo em base de dados esparças.

Porém, um dos maiores desafios de sistemas de recomendação no mundo comercial é que usuários raramente avaliam produtos e poucas vezes são encorajados para isso. Portanto, em trabalhos futuros pretende-se investigar como informações indiretas[27], técnica conhecida como feedback implícito (as quais não são fornecidas diretamente pelo usuário) dos usuários como quanto tempo ele passou em determinada página do site, quais links ele clicou, quais tags ele pesquisou, quais produtos ele colocou na sua lista de compras, podem ajudar a melhorar a qualidade da recomendação.

Referências Bibliográficas

- [1] R. A. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999.
- [2] F. Ricci, L. Rokach, B. Saphira, and P. Kantor, *Recommender Systems Handbook*. Springer, 2011.
- [3] C. A. Gomez-Uribe and N. Hunt, “The netflix recommender system: Algorithms, business value, and innovation,” *ACM Trans. Manage. Inf. Syst.*, vol. 6, pp. 13:1–13:19, Dec. 2015.
- [4] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in *Proceedings of the 10th International Conference on World Wide Web, WWW '01*, (New York, NY, USA), pp. 285–295, ACM, 2001.
- [5] L. Kidziński, “Statistical foundations of recommender systems,” 2011.
- [6] C. C. Aggarwal, *An Introduction to Recommender Systems*. Cham: Springer International Publishing, 2016.
- [7] M. Balabanović and Y. Shoham, “Fab: Content-based, collaborative recommendation,” *Commun. ACM*, vol. 40, pp. 66–72, Mar. 1997.
- [8] R. J. Mooney and L. Roy, “Content-based book recommending using learning for text categorization,” in *Proceedings of the Fifth ACM Conference on Digital Libraries, DL '00*, (New York, NY, USA), pp. 195–204, ACM, 2000.
- [9] S. K. Gorakala and M. Usulli, *Building a Recommendation System with R*. Packt Publishing, 2015.
- [10] R. G. N. e. Silva, “Sistema de recomendação baseado em conteúdo textual: avaliação e comparação,” pp. 1–14, Oct. 2014.
- [11] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, “Using collaborative filtering to weave an information tapestry,” *Commun. ACM*, vol. 35, pp. 61–70, Dec. 1992.
- [12] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, “GroupLens: An open architecture for collaborative filtering of netnews,” in *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, CSCW '94*, (New York, NY, USA), pp. 175–186, ACM, 1994.

- [13] Y. Koren, “Factorization meets the neighborhood: A multifaceted collaborative filtering model,” in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, (New York, NY, USA), pp. 426–434, ACM, 2008.
- [14] N. Good, J. B. Schafer, J. A. Konstan, A. Borchers, B. Sarwar, J. Herlocker, and J. Riedl, “Combining collaborative filtering with personal agents for better recommendations,” in *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, AAAI '99/IAAI '99, (Menlo Park, CA, USA), pp. 439–446, American Association for Artificial Intelligence, 1999.
- [15] C. Desrosiers and G. Karypis, *A Comprehensive Survey of Neighborhood-based Recommendation Methods*, pp. 107–144. Boston, MA: Springer US, 2011.
- [16] J. Herlocker, J. A. Konstan, and J. Riedl, “An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms,” *Inf. Retr.*, vol. 5, pp. 287–310, Oct. 2002.
- [17] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender Systems: An Introduction*. New York, NY, USA: Cambridge University Press, 1st ed., 2010.
- [18] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” *Computer*, vol. 42, pp. 30–37, Aug. 2009.
- [19] B. M. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering,” 2002.
- [20] K. Miyahara and M. J. Pazzani, “Collaborative filtering with the simple bayesian classifier,” in *Proceedings of the 6th Pacific Rim International Conference on Artificial Intelligence*, PRICAI'00, (Berlin, Heidelberg), pp. 679–689, Springer-Verlag, 2000.
- [21] S. Vucetic and Z. Obradovic, “Collaborative filtering using a regression-based approach,” *Knowl. Inf. Syst.*, vol. 7, pp. 1–22, Jan. 2005.
- [22] J. Li, L. Sun, and J. Wang, “A slope one collaborative filtering recommendation algorithm using uncertain neighbors optimizing,” in *Proceedings of the 2011 International Conference on Web-Age Information Management*, WAIM'11, (Berlin, Heidelberg), pp. 160–166, Springer-Verlag, 2012.
- [23] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, “Evaluating collaborative filtering recommender systems,” *ACM Trans. Inf. Syst.*, vol. 22, pp. 5–53, Jan. 2004.
- [24] P. Cremonesi, R. Turrin, E. Lentini, and M. Matteucci, “An evaluation methodology for collaborative recommender systems,” in *Proceedings of the 2008 International Conference on Automated Solutions for Cross Media Content and Multi-channel Distribution*, AXMEDIS '08, (Washington, DC, USA), pp. 224–231, IEEE Computer Society, 2008.

- [25] G. Shani and A. Gunawardana, “Evaluating recommender systems,” tech. rep., November 2009.
- [26] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, “Improving recommendation lists through topic diversification,” in *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, (New York, NY, USA), pp. 22–32, ACM, 2005.
- [27] Y. Hu, Y. Koren, and C. Volinsky, “Collaborative filtering for implicit feedback datasets,” in *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, (Washington, DC, USA), pp. 263–272, IEEE Computer Society, 2008.
- [28] “Why business needs ai-based personalized product recommendations - disponível em <http://goo.gl/utpx5e>.”
- [29] “An introductory recommender systems tutorial - disponível em <https://goo.gl/kh5caz>.”
- [30] “Calendário acadêmico 2017 - disponível em <https://goo.gl/zrnvc3>.”
- [31] J. Beel, B. Gipp, S. Langer, and C. Breitinger, “Research-paper recommender systems: a literature survey,” *International Journal on Digital Libraries*, vol. 17, pp. 305–338, Nov 2016.