Explorando Relações entre Usuários em um Sistema de Recomendação Híbrido Baseado em Filmes

Por

Lássion Laíque Bomfim de Souza Santana

Trabalho de Graduação

Universidade Federal da Bahia
wiki.dcc.ufba.br/DCC/

SALVADOR, Julho/2018

Universidade Federal da Bahia
Departamento de Ciência da Computação

Lássion Laíque Bomfim de Souza Santana

Explorando Relações entre Usuários em um Sistema de Recomendação Híbrido Baseado em Filmes

Trabalho apresentado ao Departamento de Ciência da Computação da Universidade Federal da Bahia como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: *Frederico Araújo Durão*

SALVADOR, Julho/2018

Agradeço a Deus, por ter me proporcionado forças e coragem na realização deste trabalho. Dedico aos meus pais, gostaria de expressar minha gratidão a minha companheira por me dar suporte e aos meus amigos que estiveram ao meu lado me apoiando e incentivando. Ao professor Fred, por seus ensinamentos, paciência, confiança, disponibilidade e pelo incrível suporte, sempre impulsionando o melhor.

Resumo

Sistemas de Recomendação tornaram-se populares e amplamente aplicados em todo o mundo nas mais diversas linhas de atuação na indústria e na academia. Sites e serviços vem implementando esses conceitos para auxiliar usuários a filtrar informações que de fato são relevantes, tornando assim sua experiência mais personalizada. Há diversas formas de se construir um Sistema de Recomendação, como a filtragem baseada em conteúdo e filtragem colaborativa. Sistemas de Recomendação híbrido se propuseram a combinar os benefícios de ambas as abordagens, prometendo ser uma solução mais robusta para atender às necessidades e desafios desta área. Esse trabalho tem como objetivo analisar as descrições dos filmes como modelo de usuário, com o objetivo de melhorar as predições do algoritmo, e o algoritmo proposto é avaliado usando a métrica *Root Mean Square Error* e teste estatísticos. Os resultados indicam que a abordagem híbrida proposta apresenta uma melhora em comparação com o algoritmo clássico KNN baseado em filtragem colaborativa.

Palavras-chave: Filtragem Colaborativa; Filtragem por conteúdo; Filtragem Híbrida; Filmes

Abstract

Recommendation Systems have become popular and widely applied across the world in the most diverse lines of industry and academy. Websites and services have been implementing these concepts to help users filter relevant information which are in fact relevant, thus making their experience more personalized. There are several ways to build a Recommendation Systems, such as content-based filtering and collaborative filtering. Hybrid Recommender Systems have set out to combine the benefits of both approaches by making it a more robust solution to address the needs and challenges of this area. This work proposes a strategy such as analyzing the descriptions of films as a user model in order to improve the predictions of the algorithms, the proposed algorithm is evaluated using the *Root Mean Square Error* metric and statistical tests. The results show that the proposed hybrid approach presents an improvement compared to the classic algorithm based on collaborative filtering.

Keywords: Collaborative Filtering; Content-Based Filtering, Hybrid Filtering; Movies

Conteúdo

Lista de Figuras **15**

Lista de Tabelas **17**

Lista de Acrônimos **19**

1 Introdução **1**
- 1.1 Motivação . . . 3
- 1.2 Descrição do Problema . . . 4
- 1.3 Objetivo . . . 5
- 1.4 Contribuições . . . 5
- 1.5 Estrutura do Trabalho . . . 5
- 1.6 Sumário . . . 6

2 Sistemas de Recomendação **7**
- 2.1 Histórico . . . 7
  - 2.1.1 Conceitos . . . 8
- 2.2 Problemas . . . 9
  - 2.2.1 Partida a Frio . . . 9
  - 2.2.2 Ovelha Negra . . . 10
  - 2.2.3 Ramp-up . . . 10
  - 2.2.4 Esparsidade . . . 11
- 2.3 Tarefas de um Sistema de Recomendação . . . 12
- 2.4 Aplicações de Sistemas de Recomendação . . . 13
  - 2.4.1 Samba Tv . . . 14
  - 2.4.2 Amazon . . . 14
  - 2.4.3 Netflix . . . 15
- 2.5 Sumário . . . 16

3 Estratégias de Recomendação **17**
- 3.1 Filtragem Colaborativa . . . 17
  - 3.1.1 Baseado em Usuário . . . 19
  - 3.1.2 Baseado em Itens . . . 20
  - 3.1.3 Baseado em Modelo . . . 20
- 3.2 Gerando Recomendações . . . 20

3.2.1 K Vizinhos Mais Próximos 20
3.3 Desafios dos Sistemas de Recomendação Baseado em Filtragem Colaborativa 22
3.3.1 Escalabilidade 22
3.3.2 Desempenho 22
3.3.3 Memória 23
3.3.4 Esparsidade 23
3.4 Vantagens da Filtragem Colaborativa 24
3.5 Desvantagens da Filtragem Colaborativa 25
3.6 Filtragem por Conteúdo 25
3.6.1 Geração de Recomendações 28
3.6.2 Vantagens da Filtragem por Conteúdo 28
3.6.3 Desvantagens da Filtragem por Contéudo 29
3.7 Sistemas de Recomendação Híbrido 30
3.7.1 Métodos e Estratégias de Sistemas Híbridos 31
3.8 Sumário 33

4 Explorando Relações entre Usuários em Um Sistema de Recomendação Híbrido Baseado em Filmes. 35
4.1 Requisitos 35
4.1.1 Requisitos Funcionais e Não Funcionais 36
4.2 Arquitetura 36
4.3 Modelo de Recomendação Híbrido 38
4.4 Modelagem 39
4.5 Modelo de Usuário 39
4.6 Modelo de Dados Baseado em conteúdo 40
4.7 Modelo dos Itens de Recomendação 40
4.7.1 Modelo de Predição 41
4.8 Tecnologias 41
4.8.1 Java 41
4.8.2 Librec 41
4.9 Sumário 42

5 Avaliação 43
5.1 Metodologia 43
5.2 Conjunto de Dados 44

5.3 Métodos de Avaliação 45
5.3.1 *Root Mean Square Error (RMSE)* 46
5.3.2 Teste U de Mann-Whitney 46
5.4 Resultados 47
5.4.1 *Root Mean Square Error (RMSE)* 47
5.4.2 Resultados do Teste U de Mann - Whitney 49
5.5 Discussão 50
5.6 Pontos de Melhoria 50
5.7 Sumário 51

6 Conclusão **53**
6.1 Contribuições do trabalho 53
6.2 Trabalhos Futuros 54
6.3 Sumário 55

Referências Bibliográficas **56**

Lista de Figuras

2.1 Funcionamento básico de um sistema de recomendação. Imagem retirada de 8
2.2 Matriz de Usuários x Itens. 11
2.3 Interface Samba TV. 14
2.4 Recomendação de Livros no site da Amazon. 15
2.5 Exemplo de Recomendação na Plataforma Netflix. 16

3.1 Técnicas de recomendação e suas fontes de conhecimento (BURKE, 2007). 17
3.2 Exemplo do Funcionamento do KNN com o a similaridade cosseno . . 21
3.3 Distribuição das avaliações dos filmes no dataset 100k do MovieLens. . 24
3.4 Exemplo de características dos itens. Imagem de autoria do autor. . . . 27
3.5 Processo simplificado das etapas de recomendação pela abordagem por conteúdo. Imagem de autoria do autor. 27

4.1 Visão Geral da Arquitetura do Projeto 37
4.2 Modelagem do banco de dados. 39

5.1 Histograma de Notas. 45
5.2 RMSE KNN X RMSE Híbrido. 48
5.3 Teste de Mann-Whitney. 49

Lista de Tabelas

4.1 Requisitos funcionais. 36
4.2 Requisitos não funcionais. 36

5.1 Informações do Data Set. 45
5.2 Resultados do RMSE para cada algoritmo. 48
5.3 Estatísticas Descritivas do RMSE. 48

Lista de Acrônimos

SR	Sistemas de Recomendação
FC	Filtragem Colaborativa
FB	Filtragem por Conteúdo
FH	Filtragem Híbrida
DVS	Decomposição em valores singulares
RMSE	Root Mean Square Error

1

Introdução

A imaginação é mais importante que o conhecimento.

—ALBERT EINSTEIN

Entre os anos de 2015 e 2016 o aumento de dados circulando na Internet foi de 330 exabytes. Em 2016, a cada segundo circulou pela rede um volume de dados equivalente a 833 dias, mais de 2 anos de imagens por segundo (CISCO 2016)[1]. A IDC afirma que o universo digital está ampliando-se a cada dois anos. Em 2013, eram 4,4 trilhões de gigabytes no planeta. Esse número deve crescer para 44 trilhões de gigabytes até 2020, ou seja, vai se multiplicar por dez [2], entretanto a quantidade de usuários não chega nem a metade da população do planeta. Com toda essa quantidade de informação circulando na rede é humanamente impossível processar, discernir e filtrar o que chega até cada um, sendo uma grande porcentagem dessa parcela de dados para cada pessoa em especifico, considerada praticamente lixo, não agregando dessa forma, nenhum valor semântico para os usuários, fazendo-se necessário a filtragem dessas informações diante dessa sobrecarga de dados.

Em 2017, estudos [3] realizados informaram que 73% das pessoas pesquisam na Internet antes de comprar algum item, 64% dos entrevistados dessa pesquisa, que acessaram a internet nos últimos sete dias acreditavam que consultar as opiniões de outras pessoas na rede ajudava a tomar decisões sobre compras importantes. Antes de comprar algo, por exemplo, 73% dos internautas afirmaram que pesquisavam na Internet para se informar

[1] https://www.cisco.com/c/dam/en_us/about/annual-report/2016-annual-report-full.pdf

[2] https://exame.abril.com.br/tecnologia/conteudo-digital-dobra-a-cada-dois-anos-no-mundo/

[3] http://www.avellareduarte.com.br/fases-projetos/conceituacao/demandas-do-publico/pesquisas-de-usuarios-atividades-2/dados-sobre-o-publico-alvo/internet-no-brasil-2017-estatisticas/

e 71% concordaram que a rede oferece informações sobre produtos ou marcas não disponíveis em outros lugares.

Os entrevistados compreenderam dessa forma, que a conveniência da Internet era mais importante que o preço. A pesquisa também demostrou que 49% dos usuários de internet entrevistados haviam adquirido algum aplicativo nos últimos trinta dias. Entre os mais utilizados, estavam os apps de mensagens instantâneas, mídias sociais e internet banking. E em relação os usuários que acessaram as mídias sociais no último mês, 43% checaram seus perfis cinco vezes por dia ou mais. Quase metade dos entrevistados afirmaram que se sentiam "perdidos"sem seus smartphones e celulares (49%). Nesse contexto, a internet foi utilizada por 71% das pessoas nas principais regiões metropolitanas, sendo importante para além das compras (65%), mas também para o entretenimento (71%) e a comunicação (54%) dos internautas.

Diante do cenário apresentado, fica evidente a necessidade crescente na área, para o desenvolvimento de ferramentas que auxiliam-nos diariamente a lidar com essa enxurrada de dados. Conhecida como *Information Overload*, essa expressão é utilizada para descrever a dificuldade que usualmente temos de entender e tomar decisões efetivas, quando há muita informação a respeito de um tema (Yang *et al.*, 2003). Essas ferramentas, normalmente seguem determinadas informações, para assim recomendar itens para os usuários, a partir de uma string de busca ou recomendação com base no perfil traçado a partir das suas escolhas.

A medida que o sistema aprende mais sobre o usuário, essas recomendações podem ser geradas a partir de três paradigmas de recomendação: Filtragem Colaborativa (FC), Filtragem por Conteúdo (FB) e Filtragem Híbrida (FH) (Adomavicius and Tuzhilin, 2005). Baseando-se em conteúdo, o sistema personaliza a recomendação dos usuários a partir de itens acessados anteriormente por eles, e como resposta, calcula essa similaridade e traz informações alinhadas a essas escolhas, o objetivo é gerar como saída de forma automática descrições dos conteúdos dos itens e verificar estas descrições com os gostos dos usuários, buscando validar se o item é ou não relevante para cada um (Balabanović and Shoham, 1997). O paradigma da filtragem colaborativa, foi então desenvolvido para solucionar as limitações na filtragem baseada em conteúdo (Herlocker, 2000)(Ansari, 2000).

A Filtragem Colaborativa (FC) se diferencia da Filtragem por Conteúdo (FB) exatamente por não exigir o entendimento ou reconhecimento do conteúdo dos itens. Sistemas por filtragens colaborativas, configuram-se como um método que faz predições sobre interesses dos usuários a partir de preferências de muitos outros usuários, assim, se uma

pessoa A possui o mesmo interesse que uma pessoa B, possivelmente a pessoa A se interessará por conteúdos similares a pessoa B. Em um patamar mais amplo, filtragem colaborativa filtra informações ou padrões de muitos agentes, pontos de vista e fontes de dados. Unindo essas duas técnicas o sistema se torna híbrido, visando melhorar os pontos fracos dos modelos supracitados (Herlocker, 2000) (Ansari, 2000) constituída de vantagens da FC, o Sistemas de Recomendação (SR) Híbrido se beneficia do melhor de cada abordagem.

1.1 Motivação

Sistemas de Recomendação (SR) são empregados para diversos domínios e sua aplicação depende da natureza dos dados presentes. Os sistemas por filtragem colaborativa possuem um problema-chave, que é como mapear e combinar as preferências dos usuários relacionados (vizinhos), as vezes os usuários avaliam rapidamente um determinado conjunto de itens de forma massiva, assim fica mais fácil o cálculo de similaridade. Como resposta, o sistema ganha grande representação e aumenta sua precisão. A Filtragem colaborativa pode ser também, baseada em observações implícitas do comportamento normal: como a playlist do Spotify [4] que ouviram, o filme do Netflix [5] que assistiram, os itens que compraram e assim sucessivamente.

Um quesito de suma importância em Filtragem Colaborativa (FC) retratada é à obtenção de informações dos usuários, que podem originar alguns problemas, como:

- **Problema do primeiro avaliador:** quando um novo elemento aparece na base de dados não tem como recomendar este item visto que não se sabe nada a respeito do mesmo, problema do cold start, ficando assim em standby até mais informações do mesmo sejam recolhidas a partir de outros usuários.

- **Problema de pontuações esparsas:** caso a quantidade usuários do sistema não alcance um nível relativamente alto em detrimento da quantidade de informação que circula dentro do sistema, há uma chance das pontuações se tornarem esparsas, ou seja a matriz de itens x usuários apresentam muito pontos nulos o que torna difícil as predições imagine um sistema com 1 milhão de livros no catálogo e apenas 5 usuários.

- **Problema da Similaridade:** usuários com gostos "estranhos"terão dificuldades em encontrar recomendações de pessoas com o mesmo perfil, o que torna o desempenho

[4] https://www.spotify.com/br/
[5] https://www.netflix.com/br/

insuficiente para esse tipo de situação.

Contudo, existem outros métodos para lidar com a sobrecarga de informação, objetivando resolver esses problemas. É o caso da Filtragem Híbrida (FH) que é objeto principal deste estudo. A abordagem híbrida vem sendo de extrema importância neste contexto atual a medida que se impõem como uma abordagem que consegue em um bom tempo de resposta trazer resultados que superam as falhas descritas acimas. Essa abordagem, surge como uma solução que mescla as outras abordagens, conseguindo assim, modelar o perfil do usuário de forma mais fidedigna.

Possibilitando que possamos assim, resolver alguns problemas como o de similaridade, já que dispõem de diversas fontes de informações e cálculos para predizer itens nas mais diversas situações.

1.2 Descrição do Problema

O problema que este trabalho aborda diz respeito a incapacidade de SR tratarem dados heterogêneos quando são dependentes de apenas uma abordagem: FC ou FB. Esta dependência acarreta diversos impactos negativos como esparsidade, escalabilidade, ovelha negra, partida a frio, exemplos como o cold start (partida a frio), justamente devido a alguns itens não terem avaliações suficientes para se ter uma boa recomendação. Por exemplo, novos filmes que ainda não contém avaliações, pode ser muito impreciso dizer algo sobre eles, por requerer um grande conhecimento do usuário e da sua vizinhança.

A Filtragem Híbrida (FH) segundo Herlocker (Herlocker, 2000), por muito tempo os pesquisadores têm sido incentivada em suas tentativas para lidar com o problema desencadeado com a sobrecarga de informações, por meio de projetos que permeiam as tecnologias, em que automaticamente reconhecem e categorizam as informações, no caso de sistemas baseados em conteúdo, limitações como o fato do conteúdo dos dados ser pouco estruturado e assim trazer dificuldade a análise dos dados, o entendimento pode ser alterado por sinônimos por exemplo, risco de ocorrer super especialização, pois o sistema leva em considerações os ratings recolhidos das avaliações positivas e negativas dos usuários.

De acordo com (Herlocker, 2000), no inicio os sistemas de filtragem colaborativa esperavam que os usuários especificassem a relação da predição e suas opiniões. Um usuário de um sistema baseado em filtragem colaborativa deve fornecer sua opinião sobre os itens que eles tiveram a oportunidade de avaliar, assim favorecendo todo o ecossistema de usuários por meio de suas experiências, no fim o sistema por Filtragem colaborativa,

apresenta a média com o potencial interesse que aquele item deve trazer. Um ótimo exemplo de um sistema baseado em filtragem colaborativa é o de filmes MovieLens (Good *et al.*, 1999), onde o usuário insere pontuações de filmes já visto e o sistema combina esse dados para encontrar potenciais usuários com interesses similares.

1.3 Objetivo

Partindo da descrição do problema citado na seção anterior, o objetivo deste estudo é desenvolver um Sistemas de Recomendação (SR) híbrido em um domínio de filmes combinando as vantagens de um sistema baseado em conteúdo e colaborativo.

Os itens que seguem são considerados objetivos específicos deste estudo:

- Descoberta de novas relações entre os usuários, por exemplo c.
- Recomendação de itens diretamente ligado a histórico disponível sobre o usuário.
- Trazer boas recomendação para usuários que fogem ao padrão dos interesses em comum dos usuários.
- Uma alta taxa de precisão independente do número de usuários.

1.4 Contribuições

As contribuições geradas por este estudo são:

- Uma revisão comparativa entre sistemas baseados em conteúdo e filtragem colaborativa permeando a filtragem híbrida.
- Avaliação da performance do algoritmo proposto com referências em métricas e trabalho correlacionados.

1.5 Estrutura do Trabalho

Esse trabalho está estruturado da seguinte forma: Capítulo 2 a seguir apresentará o referencial teórico no qual nos aprofundaremos no estado da arte e nos trabalhos pioneiros e relevantes sobre o tema buscando entender, formalizar e consolidar nosso compreendimento sobre Sistemas de Recomendação (SR). No Capítulo 3 abordaremos as estratégias

de recomendação: Filtragem Colaborativa (FC), Filtragem por Conteúdo (FB), Filtragem Híbrida (FH). No Capitulo 4 será apresentado a proposta de um algoritmo híbrido de recomendação e no Capitulo 5 exporemos a avaliação do nosso modelo em comparação com outra abordagem utilizando métricas e data sets, por fim no Capítulo 6 concluiremos nossa abordagem.

1.6 Sumário

Este capítulo apresenta a motivação por trás deste estudo, assim como os sistemas de recomendação baseados em conteúdos colaborativos, estão inseridos no nosso dia-a-dia, e como isso tem sido cada vez mais forte dentro do nosso cotidiano. Observamos também, que as abordagens possuem seus pontos fortes e fracos, e a solução proposta busca aperfeiçoar as técnicas já existentes de forma a trazer benefícios, como desempenho através do uso de ratings e outras técnicas. E por fim, apresentamos a estrutura do trabalho.

2

Sistemas de Recomendação

Este capítulo é uma introdução sobre Sistemas de Recomendação (SR), e conceitos subjacentes. Veremos os principais termos relacionados a área e os trabalhos relacionados. Ele também apresenta os princípios, métodos, histórico, problemas e deveres de um SR.

2.1 Histórico

Em virtude da grande quantidade de informações e a disponibilidade facilitada dos indivíduos ao acesso a Internet, os usuários se deparam com uma diversidade muito grande de escolha, dentre a infinita quantidade de opções que existe na rede. Na maioria das vezes os indivíduos possuem pouca ou quase nenhuma experiência pessoal para realizar escolhas dentre as várias alternativas que lhe são apresentadas, e por essa razão lançam mão de SR, dos mais simples aos mais complexos.

Os Sistemas de Recomendação (SR) auxiliam no aumento da capacidade e eficácia deste processo de indicação já bastante utilizado na relação social entre seres humanos (Resnick and Varian, 1997). Em um sistema convencional, as pessoas fornecem recomendações como entradas de informações, e o sistema dessa forma agrega e direciona para outras pessoas consideradas potenciais interessadas neste modelo de recomendação, de conteúdos relevantes, como observamos na Figura 2.1.

Os formuladores do primeiro Sistema de Recomendação chamado: Tapestry (Goldberg *et al.*, 1992), (Resnick and Varian, 1997), descreveram a expressão "filtragem colaborativa", objetivando designar um modelo de sistema específico no qual a filtragem de informação era intermediada a partir do auxilio humano, ou seja, pela colaboração dos próprios grupos de interessados. Os autores apresentam que sistemas de Filtragem Colaborativa (FC) e sistemas de Filtragem por Conteúdo (FB) são tipos de Sistemas de Recomendação que possuem suas peculiaridades e exclusividades mas possuem como

Figura 2.1 Funcionamento básico de um sistema de recomendação. Imagem retirada de [2] acessada em 25/10/2017.

finalidade exclusiva a recomendação.

Os Sistemas de Recomendação (SR) são importantes aliados para personalizar sistemas de forma individual, principalmente na web e também se mostram aptos para identificar preferências e recomendar itens relevantes para cada usuário. Sistemas de Recomendação (SR), se configuram com uma sub área da aprendizagem de máquina (*Machine Learning*) [1], amplamente utilizada nos comércios eletrônicos de forma geral como estrategia de marketing, afim de recomendar os produtos que estejam alinhados ao perfil do consumidor com base na análise de seu comportamento de navegação, consulta e/ou compra, preferências, entre outros aspectos. Além de todos os benefícios citados, os SR também auxiliam os produtores de conteúdos e serviços na divulgação de seus conteúdos multimédia.

2.1.1 Conceitos

O conjunto de dados que os Sistemas de Recomendação (SR) utilizam, referenciam três tipos de objetos: itens, usuários e relações (Ricci *et al.*, 2011).

[1] https://www.ibm.com/developerworks/br/local/data/sistemas_recomendacao/index.html

- **Itens**: Itens são os objetos recomendados. Os itens podem ser produtos, músicas, vídeos etc, dependendo apenas do domínio no qual o sistema está sendo utilizado, por sua vez, estes itens possuem um valor dentro do SR que o caracteriza de forma positiva ou negativa, esse valor é o que informa para o sistema se aquele item é apropriado ou não para ser recomendado para o usuário final.

- **Usuários**: Os usuários dos Sistemas de Recomendação (SR) podem ter diversos objetivos e características. Para que o sistema possa gerar uma recomendação de forma mais personalizada possível é explorado um conjunto de informações sobre o usuário, quanto menos informações o sistema obtiver mais genérico será a recomendação, inversamente, quanto maior o número de informações mais personalizada são os itens recomendados. Essas informações podem e devem ser estruturadas de diversas maneiras, lembrando que dados não significam valor para o sistema, mas informações sim, a partir do momento que os dados obtidos do usuário sejam ele oriundos de histórico de navegação ou comportamentos dentro do sistema quando "mineramos"esses dados é obtida informações que passam a agregar valor para o usuário.

- **Transações**: Transações são registros de interação entre o usuário e os Sistemas de Recomendação (SR), e funcionam como dados de um backup sobre tudo que é feito pelo usuário no sistema, cada clique, compra, visualização de um item, cada interação entre o homem e a máquina, ao fim esse registro é útil para o algoritmo para criar o perfil do usuário e modelar o seu comportamento.

2.2 Problemas

Um dos grandes desafios deste tipo de sistema é realizar a combinação adequada entre as expectativas dos usuários e os produtos, serviços e pessoas a serem recomendados aos mesmos, ou seja, definir e descobrir este relacionamento de interesses é o grande problema.

2.2.1 Partida a Frio

Em um dado cenário onde SR não detém as propriedades dos objetos, seja isto por diversos motivos, como um novo filme ou livro que acabaram de chegar no catálogo e não possuem avaliações suficientes para gerar recomendação de forma satisfatória, eles não podem ser utilizados de forma total pelo sistema e geralmente são negligenciados.

Este problema é chamado de *Partida a Frio* e pode ser visto de três formas: novas comunidades, novos itens e novos usuários (Bobadilla *et al.*, 2013).

- *Novas comunidades*, faz referência a SR que acabaram de entrar em funcionamento e que muito provavelmente possui uma quantidade baixa de usuários ativos, que necessitam ser instigados a utilizarem e avaliarem os itens, de forma que o sistema possa começar a gerar boas recomendações (Bobadilla *et al.*, 2013).

- *Novos itens*, refere-se aos itens que acabaram de ser incorporados á um ambiente de SR em utilização e não possuem qualquer tipo de avaliação ou não foram avaliados de forma suficiente ainda(Adomavicius and Tuzhilin, 2005), portanto, por falta de informações sobre esses itens, eles ficam propensos a serem deixados de lado, o que por consequência leva menos pessoas a utilizarem, gerando um loop (Bobadilla *et al.*, 2013). Este problema está associado a cenários ou plataformas que vivem em constante adição de itens e apenas alguns são classificados (Burke, 2002).

- *Novos usuários*, refere-se a usuários que acabaram de ser registrados no SR e não possuem nenhum tipo de avaliação. Um novo usuário deve possuir uma certa quantidade de itens classificados de modo que o SR possa obter informações e aprender sobre seus interesses e preferências, assim tornando-se apto a propor recomendações para ele (Adomavicius and Tuzhilin, 2005).

2.2.2 Ovelha Negra

Existem usuários com determinados perfis, que ficam em parte dificultando o trabalho dos sistemas de recomendação, por exemplo: um certo usuário A pode não ter nada semelhante ao usuário B, C, D, ou seja, em termo de itens estes usuários não possuem semelhanças, logo nenhuma recomendação útil é oferecida (Su and Khoshgoftaar, 2009). Em outras palavras, usuários com interesses muito diferentes da maioria, encontram problemas para receber boas recomendações.

2.2.3 Ramp-up

O *Ramp-up* (Aceleração) é um termo utilizado para descrever um problema que geralmente ocorre em sistemas baseados em conteúdo e em Sistemas de Recomendação (SR) de forma geral. Este termo também é utilizado em economia e negócios para descrever um aumento na produção firme antes do aumento antecipado da demanda do produto (Terwiesch and Bohn, 2001).

Por exemplo, novos itens podem ficar em Stand-by muito tempo sem serem recomendados para nenhum usuário, até obterem algum tipo de avaliação. As recomendações para os itens que são novos dentro do sistema são essencialmente mais fracas do que os produtos que são amplamente divulgados e classificados por milhares de usuários, e este mesmo caso ocorre quanto temos usuários novos no sistema também. Em outras palavras até que o sistema não esteja consolidado com uma grande quantidade de usuários de forma que seu perfil de comportamento esteja bem definido, o sistema pode não ser útil para a maioria dos usuários, até que um número suficiente de itens não sejam avaliado, o sistema pode não ser útil para determinados usuários.

2.2.4 Esparsidade

Exemplo de uma matriz esparsa de usuários x itens onde o valor 0 corresponde a itens cujos ratings precisam serem previstos.

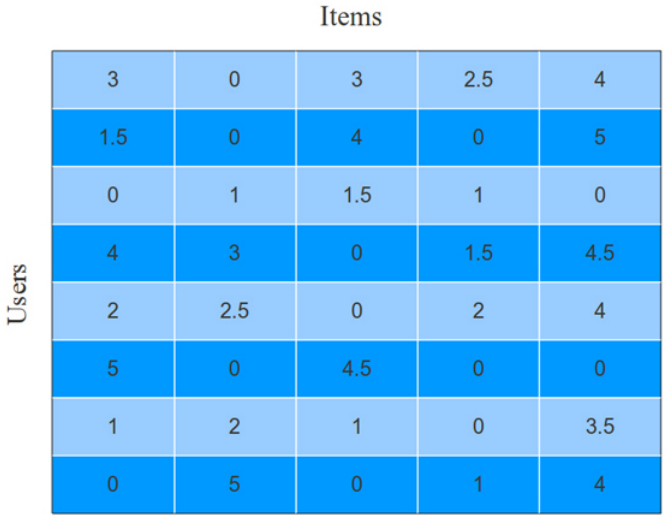


Figura 2.2 Matriz de Usuários x Itens.

Em geral o número de avaliações já obtidas é muito pequeno em relação ao número de avaliações que necessitam de predições, o que vai ser traduzido em muitas combinações utilizador/item não relevantes(células com zero ou valor indefinido). Por exemplo, num SR de filmes, os filmes que forem avaliados por poucos utilizadores são raramente recomendados, mesmo que estes atribuam pontuações elevadas.

A Esparsidade da relação *usuário x item*, nesse sentido, são alguns dos fatores que devem ser analisados para decidir qual tipo de abordagem será utilizada na solução de recomendação adotada no sistema.

2.3 Tarefas de um Sistema de Recomendação

Sistemas de Recomendação (SR) tem como base alguns objetivo (Ricci *et al.*, 2011) representou em uma lista de itens que os "deveres "que um SR devem ser capazes de realizarem:

- **Aumentar o número de itens vendidos**: Esta é um das funções mais essenciais dos SR, muito utilizado no mundo comercial. Pois, entende-se que a utilização do SR deve ser capaz de alavancar as vendas dos produtos através de análise do perfil dos usuários.

- **Vender itens diferentes**: Outro ponto importante é mostrar para o usuário muitas da vezes, coisas que ele precisa, mas que nem mesmo ele sabe, até que o sistema indique para eles, e desse modo ser capazes de mostrar itens diferente do conteúdo que ele está acostumado a consumir.

- **Aumentar a satisfação do usuário**: Uma série de combinações de recomendações precisa estar alinhada a uma boa interface de forma amigável, de forma que o usuário compre itens e/ou descubra novos relacionamentos. Este conjunto de fatores vai aumentar as chances de uma avaliação positiva do usuário e uma possível satisfação com os serviços.

- **Aumentar a fidelidade do usuário**: A partir do momento que o seu sistema consegue oferece para o usuário uma boa quantidade de recomendações e com uma boa precisão no que é recomendado, é provável que o cliente volte para fazer novas buscas, comprar novos produtos, ouvir novas músicas, etc, independente do contexto o usuário vai sentir-se único e como se o sistema soubesse exatamente do que ele precisa.

- **Melhor entendimento do que o usuário quer**: Coletar as informações do usuário seja de informa implícita ou explicita, para aprimorar com o tempo o seu perfil e recomendar assim, serviços mais próximo do seu perfil.

- **Anotação em Contexto**: Os Sistemas de Recomendação (SR) precisam ser capaz de recomendar e avaliar a necessidade do usuários nos mais diferentes contextos, não é adequado o sistema recomendar sempre a mesma coisa por exemplo, em um grande intervalo de tempo, ou até mesmo, não levar em consideração que as pessoas são complexas e seus interesses mudam a todo momento influenciados por

diversos fatores externos. Através desse sistema, pode-se recomendar a um usuário que está em um país diferente, por exemplo, o sistema tem a localização dele e assim recomenda lojas físicas de locais próximos.

- **Encontrar Bons Itens**: Os SR precisam dar ao usuário a possibilidade dele escolher entre todos os determinados itens que estão inseridos no contexto no qual ele deseja, mas o sistema precisa informar para ele por exemplo, a probabilidade daquela lista de itens lhe agradar, podendo atribuir uma nota a cada um daqueles produtos/serviços.

- **Encontrar Todos os Bons Itens**: Encontrar bons itens é um fator importante, compreendendo que é muito mais provável que o usuário não vá querer itens com avaliações mais baixas ou que a maioria das pessoas não desejam, porém é preciso ser transparente e mostrar todas as possibilidades.

- **Recomendar uma Sequência**: Recomendar itens que muitas das vezes não estão por exemplo, na mesma classe de produtos, mas de alguma forma está relacionados, por exemplo, muita das vezes quem procura por um video game novo deseja jogos novos e não necessariamente outros video games.

- **Recomenda um Grupo**: Recomendar um grupo de itens relacionados que possam interessar o usuário.

- **Apenas Navegar**: Mesmo que o usuário não esteja a procura de nenhum produto, ou que o sistema não tenha nenhum perfil formado sobre ele, é possível através de Filtragem colaborativa (tema aprofundado no Capítulo 3), recomendar quais itens naquela determinada seção especifica, as pessoas estão mais interessadas.

2.4 Aplicações de Sistemas de Recomendação

O primeiro sistema de recomendação foi denominado Tapestry, desenvolvido no inicio dos anos 90 (Goldberg *et al.*, 1992), desenvolvido por pesquisadores da Xerox Palo Alto Research Center (Resnick and Varian, 1997).

O objetivo do Tapestry era filtrar e arquivar os e-mails diariamente, desse modo criaram a expressão “Filtragem Colaborativa”, visando designar um tipo de sistema específico no qual a filtragem da informação era realizada com o auxílio humano.

2.4.1 Samba Tv

Com o avanço da tecnologia e a chegada das smart TVs em nosso cotidiano, rapidamente os aparelhos começaram a dispor de conexões com internet, integração com youtube, spotify, facebook entre outras.

Também não é novidade que empresas especializadas, como a Samba Tv [3] começassem a coletar informações dos programas e dos interesses do usuários para fornecer recomendações de outros programas, filmes e também anunciar produtos de seus patrocinadores.

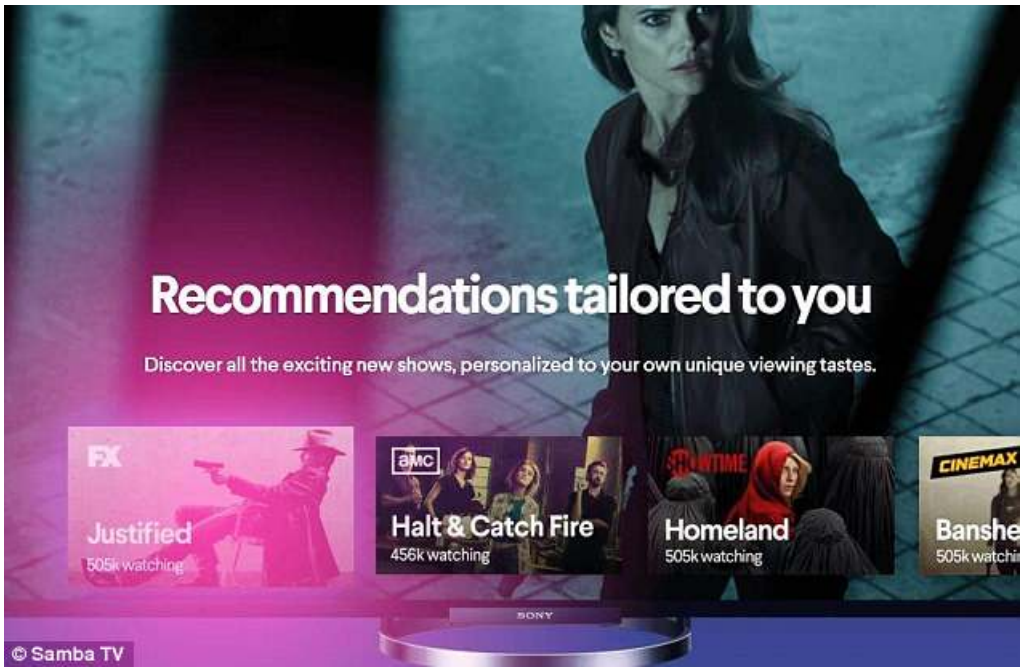


Figura 2.3 Interface Samba TV.

2.4.2 Amazon

A Amazon [4] é um dos principais *e-commerces* atualmente que utilizam Sistemas de Recomendação (SR) a seu beneficio, na venda de seus produtos.

O Sistemas de Recomendação (SR) da Amazon é baseado em simples elementos. O que o usuário comprou, quais itens estão guardados em sua "lista de desejos", itens que qualificou e gostou e quais outros clientes que viram e compraram aqueles produtos.

[3] https://samba.tv
[4] http://www.amazon.com

Uma espécie de filtro colaborativo item à item, bastante utilizado para proporcionar melhor engajamento a clientes que frequentam constantemente a loja. Desde o inicio da implantação deste SR o faturamento das vendas aumentou em 29% [5].

Figura 2.4 Recomendação de Livros no site da Amazon.

2.4.3 Netflix

A Netflix [6] é mundialmente reconhecida pelo seu excelente SR, desde 2007 quando ofereceu o prêmio de 1 Milhão de Dólares por uma melhora de 10% no seu algoritmo, a empresa vem crescendo e melhorando a qualidade das recomendações, e tornando-se cada vez mais personalizadas (Bennett *et al.*, 2007).

A empresa em seu catálogo dispõem de filmes, séries, documentários entre outras opções. Quando um usuário se cadastra na plataforma é solicitado que ele escolha algum filmes e/ou série de seu interesse, para formar uma "primeira impressão"e a partir dessas informações, recomendar diversas listas, exemplificando porque cada uma delas aparece, detalhes que podem ser visto na Figura 2.5.

[5] http://www.smarthint.co/sistema-de-recomendacao-da-amazon-e-seus-segredos/
[6] http://www.netflix.com

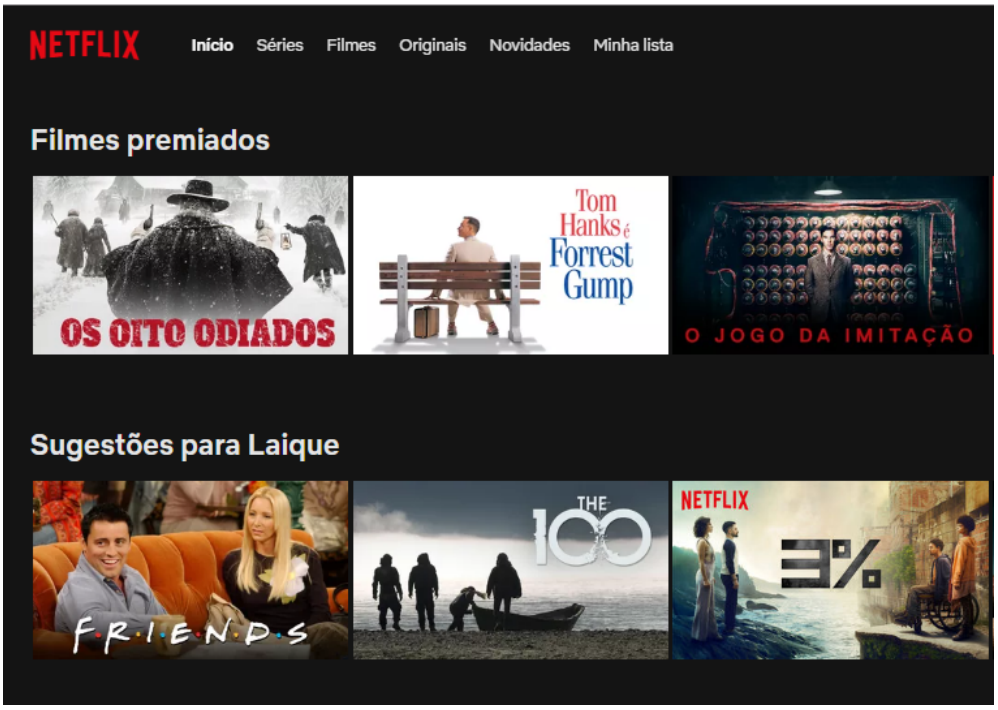


Figura 2.5 Exemplo de Recomendação na Plataforma Netflix.

2.5 Sumário

Este capítulo apresenta uma introdução breve sobre SR sua importância, histórico, problemas e diversos "deveres"de um SR, no Capítulo 3, serão abordados as estratégias de recomendação.

3

Estratégias de Recomendação

Este capítulo aborda as diversas Estratégias de recomendação: Filtragem Colaborativa (FC), Filtragem por Conteúdo (FB) e Filtragem Híbrida (FH) na seções seguintes serão abordados individualmente de forma mais aprofundada nos seus conceitos, desafios e metodologia por trás destas estratégias.

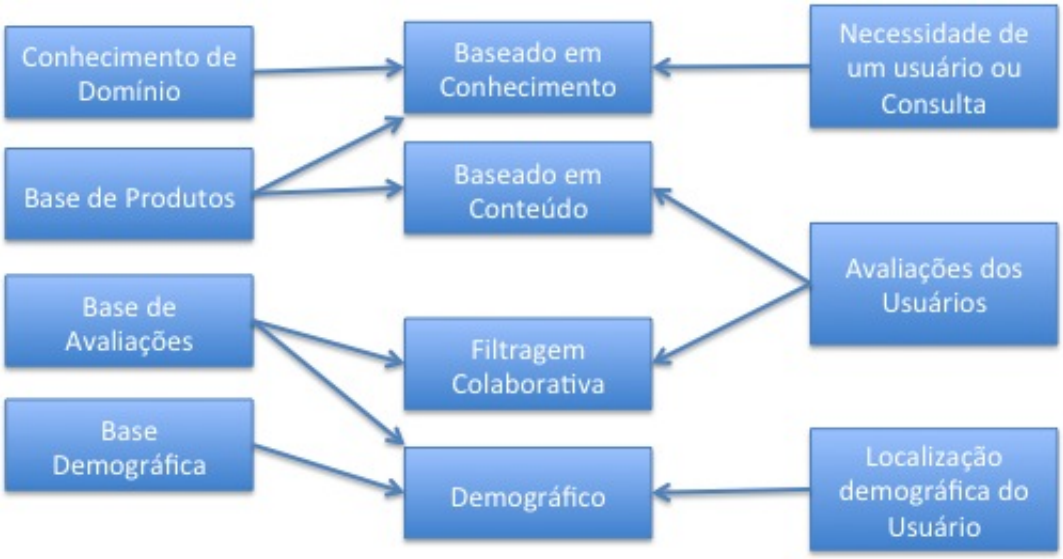


Figura 3.1 Técnicas de recomendação e suas fontes de conhecimento (BURKE, 2007).

3.1 Filtragem Colaborativa

Sistemas de recomendação empregam diversas técnicas de análise de dados ao problema, facilitando a vida dos usuários no quesito de encontrarem o que desejam. A FC é o método de selecionar informações ou padrões, ele trabalha construindo uma base de

dados de preferência de itens relacionado a determinado usuário, um novo usuário no sistema é comparado aos vizinhos desta mesma base, de modo que reconheçam interesses e características similares. Os itens que estão associados a este vizinho é recomendado ao usuário inicial (Adomavicius and Tuzhilin, 2005).

O método se assemelha à forma como as pessoas geralmente lidam com decisões: contando com experiências e conhecimentos de pessoas do seu círculo de amizade(Bobadilla *et al.*, 2013). O processo diferenciativo que abrange as técnicas colaborativas são: primeiro, a similaridade entre usuários que é calculada, então os usuários são classificados ou agrupados em bairros, finalmente a serventia de um item i para um usuário u é estimado de acordo com a utilidade de i para os vizinhos de u (Bozo *et al.*, 2016).

De modo formal, a importância $f(u,i)$ do item i para o usuário u é prevista baseando-se na utilidade $f(u_m,i)$ associado ao item i pelos usuários $u_m \in U$ que são semelhantes ao mesmo u, (Adomavicius and Tuzhilin, 2005). Por exemplo, em um cenário de recomendação de livros, o SR tenta encontrar usuários com interesses similares u, e apenas os livros que geram maior identificação com o usuário serão recomendados (Adomavicius and Tuzhilin, 2005). Em vista disso, as pessoas que classificaram os itens de forma semelhante receberão recomendações de acordo com essa semelhança (Bobadilla *et al.*, 2013).

Um dos notórios benefícios de usar as estratégias colaborativas é que elas não dependem de conteúdo compreendido por máquinas, e podem ser usados para objetos de maiores complexidades, como vídeos e músicas (Burke, 2002). As técnicas colaborativas podem ser divididas em duas categorias (Bobadilla *et al.*, 2013; Adomavicius and Tuzhilin, 2005):

Baseado em memória ou vizinhança: Bastante difundida em sistemas comerciais é eficaz e de fácil implementação, porém existem várias desvantagens dessa abordagem. O desempenho não é bom quando os dados são escassos, o que ocorre com frequência se levar em consideração os itens relacionados a web, o que dificulta a escalabilidade dessa abordagem e desenvolve problemas com grande conjunto de dados. Os métodos são aqueles baseados em uma matriz que relaciona os usuários e classificações de itens (Bobadilla *et al.*, 2013), essencialmente, são heurísticas que fazem classificação e previsões de acordo com todos os itens anteriormente classificados(Adomavicius and Tuzhilin, 2005). De modo geral o valor da classificação desconhecida $r_{u,i}$ para o usuário u e o item i é um agregado de informações de outros usuários para o mesmo item i (Adomavicius and Tuzhilin, 2005):

$$r_{u,i} = \text{aggr}_{u' \in \hat{C}} r_{u',i} \tag{3.1}$$

Onde $\hat{C}$ indica o N usuários que possuem interesses similares ao usuário u. A função de agregação pode ser *Média*, ou *Soma Ponderada* dos ratings com similaridade de usuários usando pesos entre eles (Adomavicius and Tuzhilin, 2005). Neste contexto o SR armazena os dados em memória, para em seguida calcular medidas de similaridade entre os usuários e entre itens, cada vez que uma nova recomendação é gerada.

3.1.1 Baseado em Usuário

Baseado em usuário: sabendo que o algoritmo é baseado em usuário, o método de predição da apreciação de um item por um usuário é iniciado, primeiro, é necessário saber quais os vizinhos que participam ativamente. Os métodos mais utilizados para calcular a similaridade entre dois usuários e, assim possibilitar descobrir seus vizinhos, são o *Coeficiente de Correlação de Pearson* (Lee Rodgers and Nicewander, 1988) e *Similaridade Cosseno* (Berry *et al.*, 1999).

O primeiro método normalmente é utilizado quando os ratings dos usuários estão disponíveis em um escala numérica entre 1 a 5 (Su and Khoshgoftaar, 2009) por exemplo, em sites como Netflix, MovieLens, Amazon, entre outros, o usuário é instigado a da sua avaliação para determinado filme ou produto utilizando um notação de estrelas (5 estrelas) que representam valores entre péssimo e excelente. Já o método de similaridade de cosseno, é mais utilizado para cálculos entre dois usuários quando é uma avaliação 0 ou 1 (binária) que corresponde a gostei ou não gostei (Su and Khoshgoftaar, 2009).

A ***Similaridade Cosseno*** é calculada pela equação abaixo:

$$sim = \cos(\theta) = \frac{A.B}{||A_2||.||B_2||} = \frac{\sum_{i=1}^{N} A_i B_i}{\sqrt{\sum_{i=1}^{N} A_i^2}\sqrt{\sum_{i=1}^{N} B_i^2}} \tag{3.2}$$

onde A_i e B_i são componentes do vetor **A** e **B** respectivamente.

A ***Correlação de Pearson*** é calculada da seguinte forma:

$$p = \frac{\sum_{i=1}^{N}(x_i - x^-)(y_i - y^-)}{\sqrt{\sum_{i=1}^{N}(x_i - x^-)^2}.\sqrt{\sum_{i=1}^{N}(x_i - x^-)^2}} \tag{3.3}$$

onde $(x_1, x_2, \ldots, x_n)$ $(y_1, y_2, \ldots, y_n)$ são os valores medidos de ambas as variáveis. além disso

$x^{-} = \frac{1}{n} \cdot \sum_{i=1}^{N} x_i$ e $y^{-} = \frac{1}{n} \cdot \sum_{i=1}^{N} y_i$

3.1.2 Baseado em Itens

Baseados em Itens: Esse algoritmo foi proposto por (Sarwar *et al.*, 2001) e ganhou grande popularidade nas lojas virtuais principalmente depois da sua adoção pela Amazon. Em seu princípio é selecionado um item e então é identificado quais dos seus vizinhos possuem comum avaliação dadas pelo usuário do sistema. Esse cálculo pode ser realizado pela equação Cosseno 3.2 essa abordagem por exemplo, é bastante conhecida como "quem comprou esse item também comprou este".

3.1.3 Baseado em Modelo

Baseado em modelo: Nessa abordagem os modelos são desenvolvidos usando diferentes métodos de mineração de dados, machine learning para predizer as classificações dos usuários de itens ainda não classificados. Existe muitos algoritmos baseados em FC desenvolvido baseado em modelos. Bayesian networks, clustering models, latent semantic models such as singular value decomposition, probabilistic latent semantic analysis, multiple multiplicative factor, latent Dirichlet allocation e Markov decision process. Uma vantagem de usar esas abordagem é que trabalharemos com uma matriz muito menor e dimensões mais baixas.

Uma maneira melhor de lidar com esses problemas e superar essas limitações é utilizar aplicações que combinam algoritmos Filtragem Colaborativa (FC) baseados em memória e baseados em modelos. O objetivo é melhorar o desempenho das predições e lidar com a perda de informações e a esparsidade dos dados isto é, a matriz associada contem uma grande proporção de elementos nulos. Porém teremos um aumento de complexidade no modelo.

3.2 Gerando Recomendações

Em sequências será apresentado o KNN algoritmo este de extrema relevância no estado da arte de Sistemas de Recomendação (SR).

3.2.1 K Vizinhos Mais Próximos

O KNN pode ser usado para problemas preditivos de classificação e regressão (Duda *et al.*, 2012). No entanto, é mais amplamente utilizado em problemas de classificação na

indústria. Para avaliar qualquer técnica, geralmente olhamos para 3 aspectos importantes:

- Facilidade para interpretar a saída.
- Tempo de cálculo.
- Potência Preditiva.

Para identificar os vizinhos (itens ou usuários), é medida a similaridade de um usuário-alvo com os outros usuários. A função mais conhecida para medir a similaridade é a distância euclidiana. A distância euclidiana é uma medida de distância entre dois pontos ou vetores em um espaço bidimensional ou multidimensional (euclidiano) baseado no teorema de Pitágoras. A distância é calculada tomando a raiz quadrada da soma das distâncias quadradas de cada dimensão.

A distância entre dois vetores que representam os perfis do usuário. Quanto menor a distância calculada de um vetor a outro vetor, mais semelhantes são os usuários. O valor da distância varia de 0, maior similaridade, a 1, menor similaridade. A equação 3.4 define a distância euclidiana.

$$W_{a,u} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{3.4}$$

Na equação 3.4, $W_{a,u}$ representa a distância entre o usuário ativo a com um determinado usuário u , x_i , é a avaliação que o usuário ativo deu para o item i , y_i , é a avaliação de um outro usuário para o mesmo item.

Existem diversas funções que podem ser utilizadas para medir essa similaridade como a similaridade *cosseno* e de similaridade de *pearson* apresentadas nas seções 3.2 e 3.3 respectivamente.

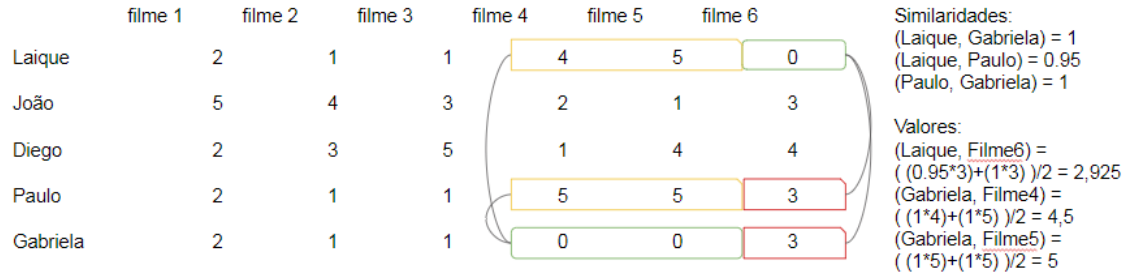


Figura 3.2 Exemplo do Funcionamento do KNN com o a similaridade cosseno

Na Figura 3.2 temos uma matriz NxM (5X6) de n usuários e m avaliações, onde cada elemento da matriz (i, j) representa a avaliação do usuário (1,5) i sobre o filme j e 0 caso o usuário não tenha avaliado o filme.

Para cada usuário, queremos recomendar filmes que o usuário não tenha assistido, para cada filme j em que o usuário i não tenha assistido, encontramos um conjunto de usuários U, similares ao usuário i que já tenham assistido j.

Para cada usuário u, pegamos a avaliação de u sobre o filme j e multiplicamos pelo cosseno da similaridade dos usuários i e u. Somamos esses valores e dividimos pelo número de usuários em U. Organizamos os filmes baseado nos valores encontrados. Esses valores representam a avaliação estimada que o usuário i dará àquele filme.

3.3 Desafios dos Sistemas de Recomendação Baseado em Filtragem Colaborativa

As técnicas de recomendação apresentada acima têm sido o foco de vários estudos, e seus desafios mais comuns são muito bem conhecidos (Burke, 2007). Alguns desses desafios são descritos a seguir.

3.3.1 Escalabilidade

À medida que o número de usuários e itens crescem, aumentam também os questionamentos com escalabilidade, desempenho e uso de memória. A escalabilidade está diretamente ligada a quantidade de itens a serem analisados, quanto a quantidade de usuários que utilizam o sistema em busca de recomendações. Existem algumas técnicas como redução da dimensionalidade (Sarwar *et al.*, 2000) e paralelismo (Olsson, 2003) que podem ser utilizadas para reduzir este empecilho. Muitos sistemas precisam reagir imediatamente aos requisitos on-line e fazer recomendações para todos os usuários, independentemente de suas compras e histórico de classificações, o que exige uma maior escalabilidade de um sistema de FC. As grandes empresas da web, como o Twitter, utilizam clusters de máquinas para avaliar as recomendações de seus milhões de usuários, com a maioria dos cálculos acontecendo em máquinas de memória muito grandes (Gupta *et al.*, 2013).

3.3.2 Desempenho

Desempenho está atrelado ao tempo que o sistema leva para dar como saída a lista de recomendações (sugestões) ao usuário ou predizer a nota de alguns itens da matriz de

itens x usuários. Comumente, esse tempo de resposta é diretamente proporcional a quantidade de itens avaliados pelo usuário. Complementarmente, o desempenho também é persuadido pela quantidade de usuários ativos no sistema, pelos seguintes motivos: o sistema lida com a realidade onde há acesso múltiplo de usuários, e se tratando de Filtragem Colaborativa (FC), o sistema solicitará mais tempo para calcular as preferências do usuário alvo com base nos demais.

3.3.3 Memória

Por fim, podemos observar que a medida que cresce o número de itens avaliados pelo usuário, isso solicitará uma quantidade maior de memória para processar e armazenar as suas características.

3.3.4 Esparsidade

De modo geral em SR, a quantidade de categorizações já identificadas é muito pequena em relação a quantidade que ainda necessita ser predita (Adomavicius and Tuzhilin, 2005). Os itens que têm uma quantia inferior de classificações podem não ser recomendados mesmo que as avaliações sejam bastante positivas. A esparsidade dos dados pode ser observada pela grande quantidade de avaliações faltantes. Esse problema ocorre com frequência nos sistemas reais de recomendação, pois existe uma enorme quantidade de itens (Sobrecarga de Informações) para a capacidade dos usuários em avaliá-los, podendo acontecer de itens não terem nenhuma avaliação.

Em vista disso, os domínios em que o número de objetos evoluem a um ritmo rápido, ou aqueles com um número inferior de usuários, podem oferecer algumas dificuldades para fatores que dependem de alta consistência de classificações, como abordagens colaborativas (Burke, 2002). Igualmente, algumas técnicas de FC só operam de forma satisfatória para usuários que estão dentro de um padrão ou nichos com muitos vizinhos que têm interesses similares (Burke, 2002). O problema de esparsidade pode ser contornado de forma parcial pelo emprego de abordagens baseadas em modelos, como a *Decomposição em valores singulares (DVS)* , que é uma técnica de redução de dimensionalidade (Adomavicius and Tuzhilin, 2005).

A esparsidade é calculada da seguinte maneira:

$$Sparsity = 1 - |R|/|I|.|U| \tag{3.5}$$

Onde R = Ratings, I = Itens, U = Usuários.

Na Figura 3.3 temos um exemplo real de esparsidade em um data set de filmes do MovieLens [1], podemos observar a quantidade de ratings em função do número de filmes.

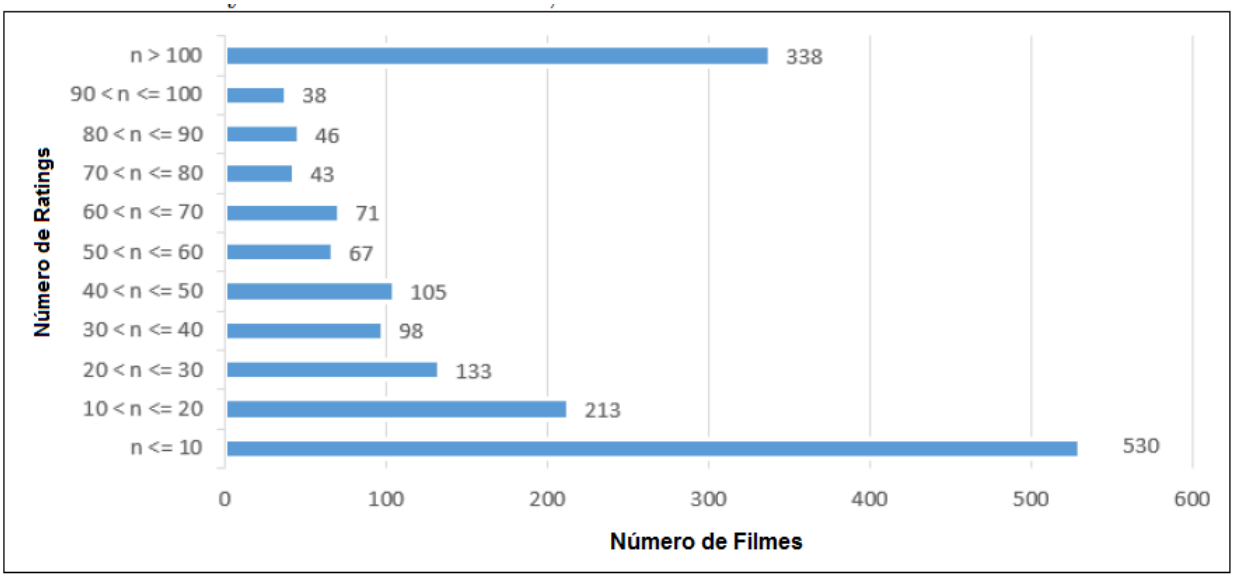


Figura 3.3 Distribuição das avaliações dos filmes no dataset 100k do MovieLens.

3.4 Vantagens da Filtragem Colaborativa

A Filtragem Colaborativa (FC) possui 3 principais vantagens:

- **Independência de conteúdo** : A técnica de filtragem colaborativa parte das avaliações de outros usuários para fornecer recomendações independentes de conteúdo, podendo assim gerar recomendações de produtos, filmes, livros, CDs tudo isso dentro de um sistema.
- **Geração de recomendações baseadas em preferências dos usuários**: A capacidade de um SR produzir boas recomendações está relacionada a qualidade com que o sistema consegue abstrair as preferências do usuário, é bastante difícil um SR fazer recomendações baseados apenas em um gênero ou algo do tipo, mas quando um usuário avalia um filme por exemplo, fica mais simples produzir recomendações com qualidade.
- **Possibilidade de produzir recomendações inesperadas e de alta qualidade**: Com a FC o sistema consegue oferecer produtos diferenciados e inesperados

[1] https://grouplens.org/datasets/MovieLens/100k/

mas com qualidade porque se baseia nas boas avaliações dos vizinhos, ou seja, uma recomendação que impacte positivamente no usuário, essa ação é chamada de *serendipity*.

3.5 Desvantagens da Filtragem Colaborativa

Existem três desvantagens intrínsecas à Filtragem Colaborativa (FC):

- **O problema do avaliador**: A base da FC são a avaliações de itens por usuários, então, um item que não tenha sido avaliado nunca será recomendado, pois este produto não vai aparecer como opção para o SR.

- **A dispersão da base de dados**: Exemplo, um usuário de e-commerce, sites que possuem uma base de dados muito grande constituída de diversos produtos, se esse site tiver por exemplo 2 milhões de itens em seu catálogo, para um usuário formar um perfil com 0,1% dos produtos, seria necessário que ele comprasse ou avaliasse no mínimo 2000 itens.

  Esse problema é chamado de dispersão dos dados, devido a isso é muito difícil por exemplo, encontrar bons vizinhos pelo baixo número de produtos em comum consumidos ou avaliados. Acarretando no problema do falso vizinho ou ovelha negra, que acontece quando usuários são considerados semelhantes para o sistema, mas na verdade esses usuários não possuem preferências parecidas para outros itens. Pode ocorrer que usuários, coincidentemente, avaliem alguns itens iguais e nos perfis dos mesmos possuam poucos itens.

- **Custo de Processamento**: Outro problema ligado a FC é o cálculo para realizar a formação dos grupos de vizinhança que é computacionalmente caro, pois para se obter as informações de pessoas com interesses semelhantes é necessário calcular a similaridade do usuário-alvo com todos os outros usuários do sistema.

3.6 Filtragem por Conteúdo

Abordagems baseadas em conteúdo precisam do conhecimento das características dos itens para gerar recomendações (Bozo *et al.*, 2016). Os usuários recebem recomendações de itens similares aos que eles já avaliaram positivamente em um momento anterior (Adomavicius and Tuzhilin, 2005) Tomemos como exemplo se um usuário se interessou

por vários celulares da marca X, o SR irá provavelmente recomendar itens da mesma marca, desta maneira o perfil de usuário é uma representação de forma estruturada de seus interesses.

As características dos itens são extraídas do conteúdo textual usando técnicas de *Recuperação de informação* (Burke, 2002; Bozo *et al.*, 2016). Estes atributos também pode ser extraidos de metadados como pode ser visto na Figura 4.2, principalmente quando a informação textual não se encontra de forma trivial para extração, por exemplo em videos, imagens e áudios (Burke, 2002).

Metadados podem ser atributos que caracterizam os itens que geralmente são atribuídos por especialistas na área ou por usuários em um ambiente web formando o que é chamado de folksonomia que nada mais é do que uma maneira de indexar informações (Bobadilla *et al.*, 2013). Independente se são extraídos de metadados ou conteúdo textual os atributos acabam correspondendo a um conjunto de palavras chaves (Adomavicius and Tuzhilin, 2005).

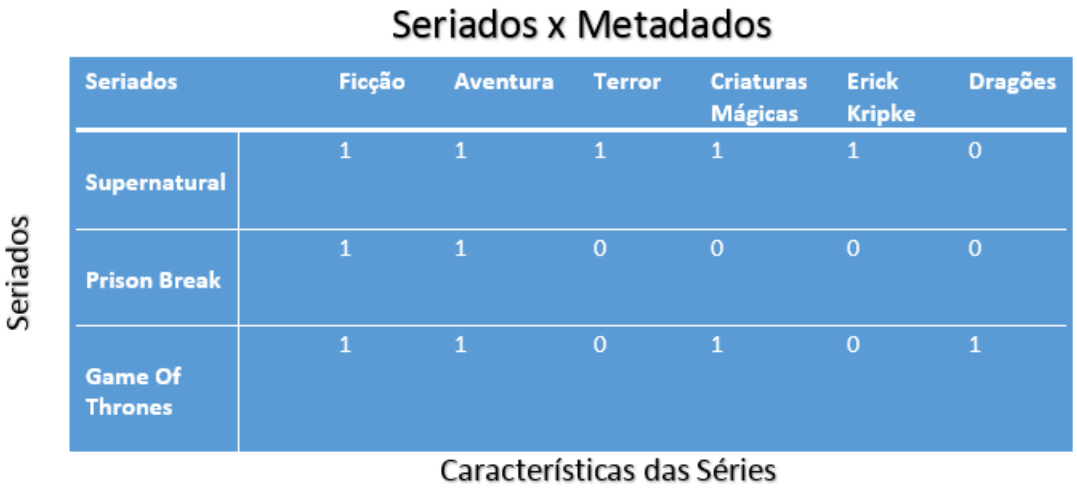
Pode-se calcular a utilidade de cada item para um usuário e filtrar os itens que serão retornados a partir de um patamar pre-estabelecido. A utilidade $u(c, s_i)$, atribuídas pelo usuário c aos itens $s_i \in$ S que são similares ao item s (Adomavicius and Tuzhilin, 2005). Em outras palavras a avaliação do usuário seja ela de forma implícita ou explicita sobre um item espelha a finalidade para qual ela se aplica. Então, esses itens similares avaliados anteriormente pelo usuário provavelmente possuirão finalidades semelhantes.

Na Figura 3.4 podemos ver um exemplo de seriados x metadados, no eixo com valores binários 0 e 1 para representar a existência ou a inexistência de determinada característica nos seriados.

A recomendação por filtragem baseada em conteúdo pode se dar em três etapas (Lops *et al.*, 2011):

- **Analisador de conteúdo**: este item é responsável pela extração das características do item seja ela vídeos, musicas, etc de forma estruturada que permita o acesso de forma organizada para as outras etapas. Técnicas de extração de características (Manzato, 2010) são aplicadas de forma a gerar por exemplo um array de palavras chaves.

- **Construtor do perfil**: essa parte tem a finalidade de a partir das extrações das características dos itens do qual o usuário possui interesse, estruturar seu perfil para que suas preferências sejam levadas em consideração na hora da recomendação para produzir resultados eficientes.



Seriados	Ficção	Aventura	Terror	Criaturas Mágicas	Erick Kripke	Dragões
Supernatural	1	1	1	1	1	0
Prison Break	1	1	0	0	0	0
Game Of Thrones	1	1	0	1	0	1

Figura 3.4 Exemplo de características dos itens. Imagem de autoria do autor.

- **Filtragem**: analisa o conjunto de itens disponíveis e o perfil traçado do usuário, confrontando sua similaridade para geração de um rank de itens relevantes para serem recomendados.

Figura 3.5 Processo simplificado das etapas de recomendação pela abordagem por conteúdo. Imagem de autoria do autor.

A principio o analisador de conteúdo extrai as características dos itens, palavras chaves, conceitos, tags e processa esse conjunto, e vai armazenando de forma estruturada para ser utilizada posteriormente. O comportamento do usuário dentro do SR é coletado para construir seu perfil, informações estas são cruzadas dentro do sistema para obter

quais itens possuem a similaridade ideal para ser rankeada como um possível interesse do usuário.

O usuário pode inicialmente informar alguns interesses para o SR, como é feito por exemplo quando acessamos o Netflix, o que facilita inicialmente um perfil básico a cerca dos interesses. O *feedback* obtido pelo SR em resumo configura sentimento de aceitação ou rejeição pelo conteúdo presente, além do *feedback* explicito quando o usuário avalia diretamente o item, pode-se obter o *feedback* implícito quando poderíamos analisar por exemplo quanto tempo ele demora assistindo um filme de um determinado gênero, o horário que ele costuma acessar a plataforma, quantas vezes o usuário pausou o conteúdo etc.

3.6.1 Geração de Recomendações

Em contraste com a técnica colaborativa, na FB não há necessidade de formação de vizinhança entre os usuários, já que o essencial aqui é a comparação do perfil do usuário identificando produtos similares.

As descrições dos produtos são textos que expressam o conteúdo do produto. A técnica *TF-IDF (Term-frequency Inverse-Document-Frequency)* (Salton and Buckley, 1988) é uma das técnicas mais utilizadas hoje em dia para a FB.

$$similaridade(Q,D) = \frac{\sum_{K=1}^{t}(W_{qk}) * (W_{dk})}{\sqrt{\sum_{k=1}^{t}(W_{qk})^2 * \sum_{k=1}^{t}(W_{dk})^2}} \tag{3.6}$$

Os pesos W_{qk}, W_{dk} variam de 0 a 1, onde 1 é utilizado para termos com a maior importância e 0 para termos de menor importância. Em algumas circunstâncias esses valores são normalizados. Em resumo a formula 3.6 mensura o número de termos que estão associados entre a Query Q e o texto e/ou documento D.

3.6.2 Vantagens da Filtragem por Conteúdo

Segundo (Lops *et al.*, 2011), a Filtragem por Conteúdo (FB) possui algumas vantagens em relação a Filtragem Colaborativa (FC):

- **Independência do *feedback* de outros usuários**: Diferentemente dos sistemas com abordagem colaborativa que necessitam de aprendizado coletivo sobre os itens, a abordagem por conteúdo dependente unicamente das características dos item presente dentro daquele domínio que será aplicado o SR e da formação do perfil do usuário e buscar conteúdo similar.

- **Maior transparência**: o sistema pode oferecer de forma explicita ao usuário a explicação pelo qual aquele conteúdo foi recomendado, também fornecer explicações sobre os itens que foram recomendados e as suas características. Enquanto um sistema colaborativo o que se sabe de ante-mão é que vários usuários avaliaram de forma positiva aquele item e provavelmente você vai se interessar também.

- **Novos itens**: na abordagem colaborativa um item novo só precisa que as suas informações façam parte do conjunto de interesse do perfil do usuário para ser recomendado. Exemplo, um novo filme só precisa que as características dele confrontem os interesses do usuário, Enquanto na Filtragem Colaborativa (FC) um novo item só sera recomendado quando tiver uma certa quantidade de avaliações positivas, o famoso problema do *Cold Start*.

3.6.3 Desvantagens da Filtragem por Contéudo

Porém, também a Filtragem por Conteúdo (FB) possui alguns problemas que merecem ser destacados Adomavicius and Tuzhilin (2005):

- **Limitação da análise dos itens**: Processar todas as informações referentes aos itens tem um alto custo operacional pela dependência de possuir itens bem descritos suficientemente para poder categorizá-los, independente de ser realizada de maneira automaticamente ou não. O que pode fazer com que apenas alguns aspectos dos itens sejam explorados, que podem ocasionar uma recomendação errônea, por exemplo, um seriado bem avaliado do gênero terror cuja única característica explorada foi o gênero do filme pode inferir que o usuário goste de todo e qualquer filme de terror o que não pode ser verdade, quando na verdade o que o chamou atenção no filme foi outro aspecto.

- **Sobre-Especialização**: Este fenômeno ocorre se o sistema utilizar apenas abordagem baseada em conteúdo que necessita que o item possua uma alta similaridade para ser recomendada o que limita o leque de opções para o usuário. Se ele só avaliar filmes de aventura o sistema só vai recomendar itens neste sentido e nunca vai recomendar um filme de comédia por exemplo.

- **Problema de um novo usuário**: O sistema só vai conseguir recomendar itens de forma eficiente e confiável após o usuário avaliar uma certa quantidade de itens, para que o SR possa entender seu perfil e seus interesses.

3.7 Sistemas de Recomendação Híbrido

Sistemas de Filtragem Híbrida (FH) surgem a partir de limitações dos sistemas de FC e FB, destacando-se os problemas de:

- **Primeiro avaliador**: um item novo fica em Stand-by até que possua uma certa quantidade de avaliações.
- **Pontuações esparsas**: quando possuímos uma enorme quantidade de itens mas um pequeno conjunto de usuários para avalia-los.
- **Usuários com interesses muito diferentes**: estes vão possuir certa dificuldade de usufruir de boas recomendações por falta de similaridade com outros usuários.

Para lidar com estes empecilhos, existe a abordagem híbrida que une as vantagens da Filtragem Colaborativa (FC) e Filtragem por Conteúdo (FB) (Cazella *et al.*, 2010). Sistemas de Recomendação Híbridos surgem da combinação de duas ou mais técnicas de recomendação (Burke, 2002). Diferentes maneiras podem ser usadas para combinar técnicas de FC e FB, que segundo (Adomavicius and Tuzhilin, 2005) se classificam das seguintes formas:

- Implementando métodos colaborativos e baseados em conteúdo separadamente e combinando suas predições.
- Incorporando algumas características baseadas em conteúdo em uma abordagem colaborativa.
- Incorporando algumas características colaborativas em uma abordagem baseada em conteúdo.
- Construção de um modelo unificador que incorpora características baseadas em conteúdo e ferramentas colaborativas.

A primeira forma nos orienta que o caminho para a implementação de um bem sucedido sistema hibrido é implementando separadamente as técnicas de FC e FB. A segunda abordagem incentiva o modelo ter como técnica principal a FC e utilizar por exemplo características da FB que complementem as falhas da FC, mantendo um perfil baseado em conteúdo do usuário e utilizar dessas informações para cálculos de similaridades o que contorna a escassez de dados.

A terceira maneira é utilizar uma técnica popular, que envolve utilizar alguma técnica de redução de dimensionalidade em um grupo de perfis baseados em conteúdo. Por exemplo, (Soboroff and Nicholas, 1999) usa indexação semântica latente para criar uma visão colaborativa de uma coleção de perfis de usuários, onde os perfis de usuários são representados pelo termo vetores resultando em uma melhoria de desempenho em comparação com a abordagem pura baseada em conteúdo.

A quarta abordagem vem sendo utilizada por muitos pesquisadores nos últimos anos. E de acordo com (Basu *et al.*, 1998), propõe o uso de conteúdo e colaborativo características (por exemplo, a idade ou o gênero dos usuários ou o gênero de filmes) em uma base única. (Popescul *et al.*, 2001) apresenta um método probabilístico unificado para combinar colaborativo e recomendações baseadas em conteúdo, que se baseiam na análise semântica probabilística latente (Hofmann, 1999).

3.7.1 Métodos e Estratégias de Sistemas Híbridos

Existem diferentes estratégias pelas quais a sistemas híbridos podem ser alcançados e são amplamente classificados em sete categorias:

- **Weighted**: Um recomendador híbrido deste tipo é aquele em que a pontuação de um item recomendado é calculada a partir dos resultados de todos das técnicas de recomendação disponíveis no sistema. O benefício de um híbrido que pondera é que todas as capacidades do sistema são levadas em conta na recomendação.

- **Switching**: Um determinado critério de comutação é usado pelo sistema para trocar entre dois sistemas de recomendação que operam no mesmo objeto, ou seja, dependendo do contexto determinada estratégia é utilizada.

- **Feature Combination**: Os recursos das diferentes fontes de dados dos sistemas de recomendação são colocados em um único algoritmo de recomendação.

- **Cascading**: Para esta categoria, um sistema de recomendação refina os resultados fornecidos por outro.

- **Meta Level**: Neste caso, um recurso como um modelo aprendido por uma recomendação é usado como entrada para outro. Ele difere do Sistema de aumento de recursos, na medida em que o modelo inteiro é usado como entrada.

- **Feature Augmentation**: A saída de um sistema é usada como recurso de entrada para outro; por exemplo, usando o modelo gerado por um para gerar recursos que

são usados por outro.

- **Mixed**: Incorpora duas ou mais técnicas ao mesmo tempo; por exemplo: Filtragem baseada em conteúdo e colaborativo.

Em seu estudo (Burke, 2002), concorda que, assim como já foi mencionado nesse trabalho, a característica dos sistemas de recomendação híbridos de combinarem duas ou mais técnicas de recomendação para obter um melhor desempenho com menos desvantagens do que individualmente. Frequentemente, a filtragem colaborativa é combinada com alguma outra técnica na tentativa de evitar o problema de *Ramp-up*.

Os sistemas de recomendação híbridos também podem ser ampliados por técnicas baseadas no conhecimento (Burke, 2000), como o raciocínio baseado em casos, para melhorar a precisão da recomendação e abordar algumas das limitações (por exemplo, novos usuários, novos problemas de itens) dos sistemas de recomendação tradicionais. Por exemplo, o sistema de recomendação baseado no conhecimento Entree utiliza algum conhecimento de domínio sobre restaurantes, cozinhas e alimentos (o tipo de alimentação, vegetariana ou não, por exemplo) para recomendar restaurantes aos seus usuários (Adomavicius and Tuzhilin, 2005).

O Entree é um sistema de recomendação de restaurantes que usa técnicas baseadas em casos (Burke, 2000) para selecionar restaurantes de acordo com a classificação. Nesse sistema o usuário interage com o sistema enviando um ponto de entrada, um restaurante conhecido ou um conjunto de critérios, e é exibido restaurantes similares. O usuário então interage com o sistema em uma caixa de diálogo, criticando as sugestões do sistema e refinando de forma interativa a busca até que uma opção aceitável seja alcançada.

A principal desvantagem dos sistemas baseados no conhecimento é a necessidade de aquisição de conhecimento - *bottleneck* que é um engarrafamento do sistema, podendo reduzir o tráfego da informação, bem conhecido para muitos aplicativos de inteligência artificial. Contudo, sistemas de recomendação baseados no conhecimento foram desenvolvidos para domínios de aplicação onde o conhecimento de domínio está prontamente disponível em alguma forma estruturada legível por máquina, por exemplo, como uma ontologia, uma ontologia é um modelo de dados que representa um conjunto de conceitos dentro de um domínio e os relacionamentos entre estes. Uma ontologia é utilizada para realizar inferência sobre os objetos do domínio (Adomavicius and Tuzhilin, 2005).

Sistemas de recomendação híbridos podem desfrutar de diferentes níveis de ganhos em precisão de predição, pelo fato de utilizarem de muitas fontes de informação, diversificando de simples benefícios a melhorias significativas. Todavia, esta adição de informação nem sempre nos leva a melhores resultados. A constatação de melhorias

significativas na qualidades dos resultados requer um estudo muito mais detalhado a respeito disto.

3.8 Sumário

Este capítulo demonstra como é realizado a abordagem baseada em conteúdo FB suas vantagens em relação a Filtragem Colaborativa (FC) e as desvantagens que esta abordagem possui. Demostra seu potencial e a necessidade de um bom perfil do usuário e uma grande quantidade de informações acerca do domínio, no qual o SR vai está inserido, também apresenta uma revisão a cerca do que é FC sua importância, vantagens, desvantagens e contexto no qual é utilizado e como ocorre o processo de recomendação utilizando essa técnica.

4

Explorando Relações entre Usuários em Um Sistema de Recomendação Híbrido Baseado em Filmes.

Os Sistemas de Recomendação (SR) como citados no Capítulo 2, demonstraram ser ferramentas indispensáveis para superar os desafios da sobrecarga de informação. Algumas estratégias como as descritas no Capítulo 3, se baseiam na similaridade dos itens focando apenas no que já é recomendado para o usuário, por um lado obtemos uma recomendação precisa e por outro lado há uma carência de uma grande quantidade de informações para que as decisões sejam tomadas com um certo grau de acurácia. Foi apresentada também uma estratégia colaborativa na qual ao contrário de enfatizar a similaridade dos itens, é trabalhado a similaridade entre os perfis dos usuários.

Entre as vantagens e desvantagens das abordagem colaborativa e por conteúdo, apresenta-se a Filtragem Híbrida (FH) que é o foco da nossa proposta. Este capítulo apresenta a nossa abordagem que combina uma medida de similaridade por conteúdo FB e colaborativa FC. Através da nossa personalização do perfil do usuário espera-se, considerando diversos fatores como descrições dos filmes que é uma informação obtida a parte do data set original, que obtenha-se uma melhor recomendação.

4.1 Requisitos

O objetivo desta etapa é reunir os requisitos funcionais e não funcionais que refletem as funcionalidades que devem ser implementadas para atender as necessidades do sistema.

4.1.1 Requisitos Funcionais e Não Funcionais

Na identificação dos requisitos, foram utilizadas as seguintes nomenclaturas: [RFXX] para requisitos funcionais e [NFRXX] para os requisitos não funcionais. São dadas prioridades aos requisitos, que servem para indicar a relevância do requisito para o sistema proposto. Essas são classificadas em:

- **Básica**: são requisitos que devem ser implementados. Caso não sejam realizados, o sistema não pode funcionar ou não atende o objetivo da proposta.
- **Significativo**: são requisitos que o sistema pode funcionar, porém de forma parcial.
- **Relevante**: estes requisitos não comprometem o funcionamento básico do sistema; podem ser deixados para versões posteriores deste trabalho.

Tabela 4.1 Requisitos funcionais.

Código	*Nome*	*Descrição*	*Prioridade*
FR01	Realizar recomendações de Filmes	Mostrar qual o provável interesse do usuário em um determinado filme	Básica
FR02	Construir o perfil do usuário com base nos filmes previamente avaliados por ele	Aprender dinamicamente a modelar o perfil do usuário	Básica
FR03	Realizar recomendações com base em múltiplas metadados	Utilizar outras características para compor o perfil do usuários	Relevante
FR04	Realizar recomendações de outros itens fora do domínio de filmes	Realizar recomendações para múltiplos domínios através de informações gerais do usuário	Relevante

Tabela 4.2 Requisitos não funcionais.

Código	*Nome*	*Característica*	*Prioridade*
NFR01	O sistema deve realizar predições com alta precisão.	Corretude	Básica
NFR02	O sistema deve fazer recomendações de maneira rápida	Eficiência	Significativo
NFR03	O sistema deve ser transparente na avaliação dos resultados	Usabilidade	Relevante

4.2 Arquitetura

Abaixo segue o modelo de Arquitetura do projeto para a predição e recomendação dos filmes.

A Figura 4.1 mostra a arquitetura para este trabalho. Primeiramente é obtida e pré processada as informações dos filmes para obtenção do perfil do usuário. Após essa etapa é feito os cálculos de similaridade necessários, separação dos conjuntos de treino e teste, então é repassado esse conjunto de atributos para o SR que nos retorna as predições.

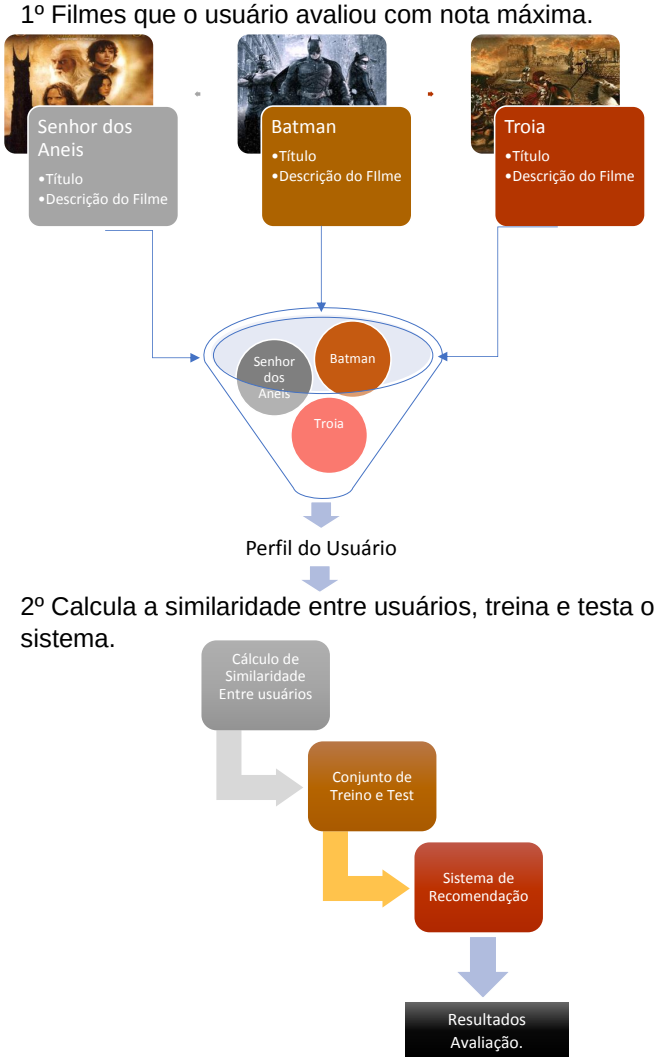


Figura 4.1 Visão Geral da Arquitetura do Projeto

4.3 Modelo de Recomendação Híbrido

A seguir será apresentado um pseudocódigo de como é o funcionamento do nosso modelo híbrido de recomendação.

Algoritmo 1: HÍBRIDO

Entrada: $TesteSet$,$Training$,$ListaUsuarios$,$Perfil$,$ListaUsuariosAvaliados$
Saída: Predição Das Notas Para os Filmes do Conjunto de Teste

```
1  início
2      para cada usuarioU ∈ ListaUsuarios faça
3          ConstruirPerfil(U,Training)))
4      fim
5      para cada usuarioU ∈ ListaUsuarios faça
6          para cada usuarioJ ∈ ListaUsuarios faça
7              SimilaridadeEntreUsuarios(S) ←
                SimilaridadeConseno(Perfil(U),Perfil(J))
8          fim
9      fim
10     para cada usuarioK ∈ ListaUsuariosAvaliados faça
11         para cada FilmeU ∈ TesteSet faça
12             Predicao(S) ← getSimilaridade(UsuarioK,FilmeU)
13         fim
14     fim
15 fim
16 retorna Predicao
```

O algoritmo de forma sucinta é realizado em três passos bases :

1. Construir o perfil do usuário, através das descrições dos filmes que ele mais se interessou que significa um filme avaliado com rating 5, conforme descrito na Seção 4.5. As descrições dos filmes são obtidas através da API do IMDB[1].

2. Fazer um pré processamento da similaridade entre usuários. Essa similaridade é calculada através de uma técnica de FB que é a similaridade *cosseno* descrita na Seção 3.2 aplicada entre os perfis dos usuários.

3. Calcular a predição da nota para os filmes pertencentes ao conjunto de teste. Essa nota é calculada aplicando uma técnica de Filtragem Colaborativa (FC), no algoritmo 4.3 é identificada pelo método ***getSimilaridade***, onde esse método

[1] https://www.imdb.com

procura os usuários que são mais similares ao usuário em questão, e que avaliaram o filme que estamos tentando predizer a nota. A similaridade entre usuários foi calculada utilizando FB na string de perfil do usuário, o método retorna as notas que os vizinhos deste usuário deram para o filme e calcula uma média para ser nossa predição.

4.4 Modelagem

Na Figura 4.2 podemos observar o modelo o banco de dados para manipulação do datam set MovieLens e das recomendações.

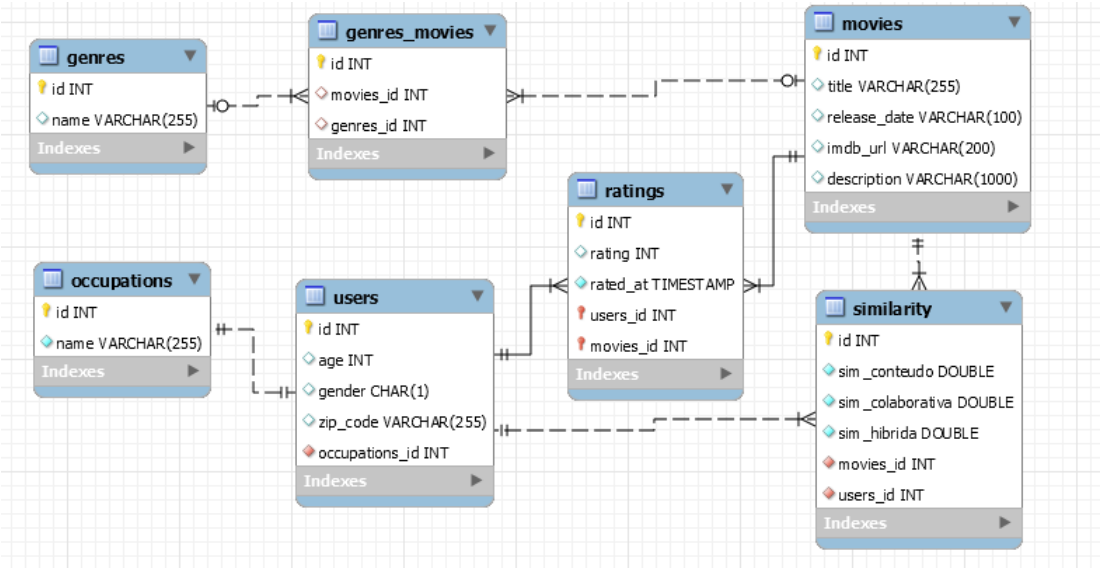


Figura 4.2 Modelagem do banco de dados.

4.5 Modelo de Usuário

Para a construção da modelagem do usuário, é preciso abstrair o que é relevante das informações que estão disponíveis e que serão utilizadas para compor o modelo de recomendações.

Neste trabalho o perfil do usuário é constituído de um "merge"de todas as descrições dos filmes que ele avaliou com nota 5, no data set as notas variam de 1 a 5 que é a nota

máxima atribuída dentro do data set MovieLens. Foi realizado um tratamento nessa conjunto de descrições retirando caracteres especias, formando assim um complexo conjunto de palavras chaves.

4.6 Modelo de Dados Baseado em conteúdo

Na composição no modelo de dados baseado em conteúdo, é indispensável calcular a similaridade entre os usuários e/ou itens baseado no perfil de cada um deles, é utilizada a similaridade *cosseno* apresentada na Seção 3.2, a equação vai receber como parâmetros duas strings que são referentes aos perfis dos usuários e nos retornará a similaridade entre eles. As strings que serão calculadas a similaridade se refere as descrições dos filmes, ou seja sera utilizado um vetor de usuários onde cada posição do vetor corresponde ao perfil do usuário que nada mais é que a coleção de palavras chaves do perfil deste indivíduo que como foi mencionado é retirado das descrições do filme, para fins de avaliação serão consideradas similaridades maior igual a 25% como o mínimo necessário para serem consideradas similares .

$$similaridade(StringA, StringB) = \cos(\theta) = \frac{A.B}{||A_2||.||B_2||} = \frac{\sum_{i=1}^{N} A_i B_i}{\sqrt{\sum_{i=1}^{N} A_i^2}\sqrt{\sum_{i=1}^{N} B_i^2}} \quad (4.1)$$

4.7 Modelo dos Itens de Recomendação

Os itens que compõem a recomendação, e que no nosso domínio são filmes são usados em 3 etapas dentro do algoritmo:

- Primeiro para cada filme no data set tem atrelado a ele um conjunto de características que os definem, uma dessas informações é a nota, que é avaliação do usuário para este item, informação esta importante para inferir que estes itens, com nota máxima estarão no conjunto do perfil do usuário.
- Segundo passo é a definição do conjunto de treinamento e de teste, este é um processo totalmente automático implementado pelo LibRec. Nessa segunda parte o algoritmo irá poder aprender sobre o usuário e estará apto a fazer recomendações.
- Na última parte o algoritmo vai inferir a nota para os filmes que fazem parte do conjunto de teste.

4.7.1 Modelo de Predição

Assim como o KNN nosso algoritmo de predição retorna um valor entre 1 e 5 para o filme do conjunto de teste baseado na nota que é calculada conforme e média das avaliações.

$$pred_{a,u} = \frac{\sum_{i=1}^{n}(v_i)}{n} \tag{4.2}$$

A predição da nota de filme u para um usuário a é calculada pelo noso modelo da seguinte forma 0: o vetor v_i é formado pelo conjunto de usuários que tem maior similaridade com o usuário a e que avaliaram o filme u, esse conjunto v_i é calculado pela equação 4.1 que foi vista na Seção 4.6, apos isto é então é calculado a média das notas que esses vizinhos deram conforme a equação 4.2. É feito então o somatório das notas e dividido pela quantidade n de usuários vizinhos.

4.8 Tecnologias

Para o desenvolvimento deste trabalho foram utilizadas diversas tecnologias: linguagens de programação, FrameWorks entre outras. Em sequência apresentaremos as mesmas.

4.8.1 Java

Java é uma linguagem de programação orientada a objetos, desenvolvida na década de 90 por uma equipe de programadores chefiada por James Gosling, na empresa Sun Microsystems. Diferente das linguagens de programação convencionais, que são compiladas para código nativo, a linguagem Java é compilada para um bytecode que é interpretado por uma máquina virtual (Java Virtual Machine, mais conhecida pela sua abreviação JVM) [2].

4.8.2 Librec

O LibRec é uma biblioteca Java open source de Sistemas de Recomendação (SR) com cerca de 70 algoritmos de recomendação múltipla, que podem efetivamente resolver os problemas de ratings e ranking. Implementa uma série de algoritmos de recomendação presentes no estado da arte e atualizados.Consiste em três componentes principais: Interfaces genéricas, Estruturas de dados e Algoritmos de recomendações (Guo *et al.*, 2015).

[2]https://www.oracle.com/br/java/index.html

O LibRec[3] possui algumas características principais:

- **Rich Algorithms**: Mais de 70 algoritmos de recomendação .
- **Alta Modularidade**: Seis componentes principais, incluindo divisão de dados, conversão de dados, similaridade, algoritmos, avaliadores e filtros.
- **Ótimo desempenho**: Implementações mais eficientes do que outras contrapartes, ao mesmo tempo que produzem precisão comparável.
- **Configuração flexível**: Acoplamento baixo, flexível e configuração externa de API textual ou interna.
- **Uso simples**: Pode ser executado em algumas linhas de códigos, e uma série de demonstrações são fornecidas para um fácil início. Expansão fácil: um conjunto de interfaces de recomendação para fácil expansão para implementar novos recomendadores.

4.9 Sumário

Neste capítulo, apresentamos uma visão geral sobre os aspectos do desenvolvimento do sistema, onde foi apresentado a arquitetura do projeto, o algoritmo proposto, modelagem do banco de dados e tecnologias. No Capítulo 5 faremos uma avaliação do trabalho realizado. Serão discutidas metodologia, métricas de avaliação, resultados obtidos e pontos de melhorias.

[3] https://www.librec.net/

5 Avaliação

Neste capítulo será apresentado o processo de avaliação utilizado para verificar se os objetivos previstos foram alcançados. Espera-se que com a criação e modelagem do problema baseado em um algoritmo híbrido onde foi explorada novas tentativas de relações entre usuários e itens, haja uma melhora na qualidade das recomendações. Para isso, os experimentos aqui relatados utilizam o dataset MovieLens 100k [1] como fonte de dados completa com informações dos usuários e filmes. A avaliação consiste em comparar a abordagem proposta no Capítulo 4 com a abordagem clássica do KNN no intuito de verificar os resultados almejados.

Este capítulo apresenta os detalhes da metodologia utilizada para desenvolver e avaliar este trabalho, bem como o conjunto de dados utilizados durante os experimentos. Em seguida são será apresentado a metodologia, conjunto de dados, métodos de avaliação, as métricas utilizadas na avaliação e os resultados obtidos. Por fim, é feita uma discussão e apresentados pontos de melhoria.

5.1 Metodologia

Os testes realizados para a avaliação têm por objetivo mostrar que ao utilizar uma abordagem híbrida podemos obter melhores recomendações conforme a teoria. Além disso os testes exibem as comprovações estatísticas e gráficas através da métrica escolhida para análise do modelo recomendação.

Para avaliação do modelo serão realizadas comparações das predições realizadas pelo algoritmo proposto contra o algoritmo clássico KNN. No caso do KNN para seu calculo de predição interno a similaridade aplicada foi o *cosseno* e na híbrida para calcular a

[1] https://grouplens.org/datasets/movielens/100k/

similaridade dos perfis de usuários também utilizamos o *cosseno*.

Em relação ao valor k, não existe um valor único para a constante, a mesma varia de acordo com a base de dados, todavia, você pode deixar o desempenho geral do modelo bem lento na etapa de seleção de k. Então realizou-se testes para encontrar esse valor de forma empírica. Para testes, utilizou-se de *Cross-Validation*, criando conjuntos de treino e test, o conjunto de treinamento corresponde a 80% e o conjunto de test a 20% esse conjunto é escolhido aleatoriamente pelo FrameWork Librec utilizado durante o desenvolvimento da solução. Para o KNN foi utilizado uma quantidade k = 50 que foi o valor considerado bom durante os testes empíricos.

Utilizando o conjunto de treino, o modelo é treinado e testa-se com os exemplos do conjunto de teste. Depois, diferentes conjuntos de treino e de teste são selecionados para iniciar o processo de treino e teste novamente, sendo repetido k vezes (Ricci *et al.*, 2011). Com os arquivos e os modelos mencionados anteriormente, todos os testes foram realizados com a integração do FrameWork Librec [2] juntamente com o FrameWork LodWeb [3]. Para medir a qualidade das predições foi utilizada a métrica *Root Mean Square Error (RMSE)* e aplicado um teste estatístico *U de Mann-Whitney* para verificar se as diferenças encontradas são estatisticamente significantes.

5.2 Conjunto de Dados

Todos os testes foram executados com dados disponíveis pelo data set público de filmes 100k do MovieLens (Harper and Konstan, 2016). Desse conjunto de dados, foram coletados informações de usuários, filmes, ratings, idade, gênero, ocupação. Foi gerado um script e persistido tudo em banco de dados Mysql. Para a nossa proposta híbrida foi necessário um item a mais, que foi a busca das descrições dos filmes através de uma consulta na *API* do IMDB [4] informação está que foi utilizada para cálculos de similaridade, o data set consiste em:

- 100,000 ratings (1-5) de 943 usuários para 1682 filmes.
- Cada usuário classificou pelo menos 20 filmes.
- Informações demográficas simples para os usuários (idade, sexo, ocupação, zip).

Os dados foram coletados através do site da MovieLens [5] durante o período de sete

[2] https://www.librec.net
[3] https://recsysufba.wordpress.com/lodweb/
[4] https://www.imdb.com
[5] movielens.umn.edu

meses a partir de 19 de setembro, 1997 até 22 de abril de 1998. Esses dados foram limpos e usuários que tinha menos de 20 classificações ou não tinha demografia completa foram removidas deste conjunto de dados. Essas informações podem ser averiguadas em [6].

Abaixo através do histograma de notas podemos visualizar como está a distribuição dos ratings entre os filmes, recordando que as notas variam entre 1 (mínimo) e 5 (máximo).

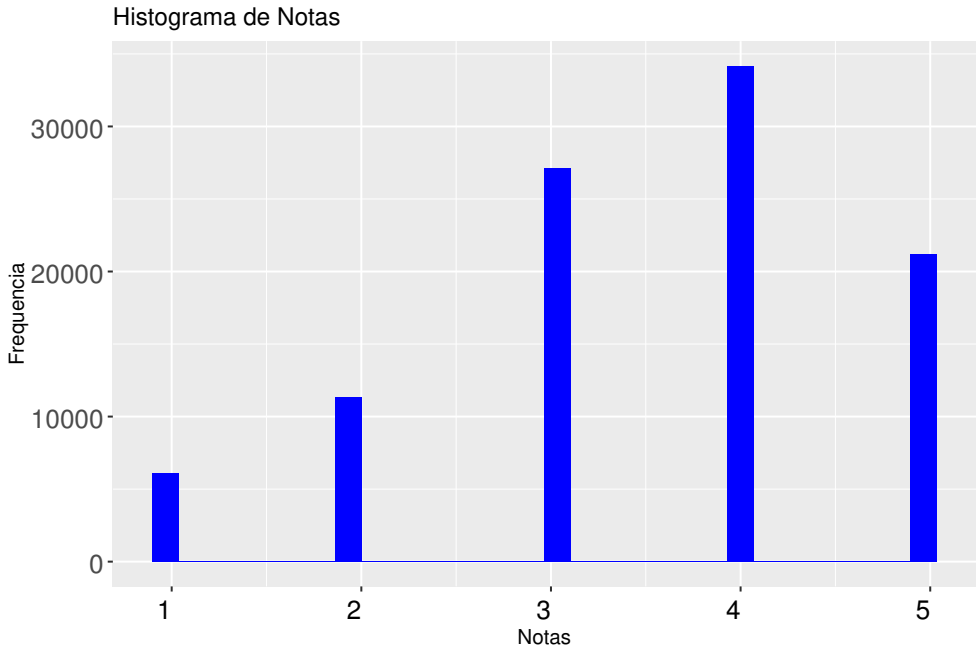


Figura 5.1 Histograma de Notas.

Tabela 5.1 Informações do Data Set.

Conjunto de Dados	
Usuários	947
Filmes	1682
Ratings	100000
Gêneros	19

5.3 Métodos de Avaliação

Em um cenário onde é fornecida uma lista de predições a partir de um conjunto de testes é possível analisar a corretude dessa predição com base em uma análise de erro, e posteriormente analisar a diferença das médias de erro entre os algoritmos.

[6] http://files.grouplens.org/datasets/movielens/ml-100k-README.txt

5.3.1 *Root Mean Square Error (RMSE)*

Geralmente no cenário acadêmico e industrial como no famoso prêmio *Netflix Prize* onde foi exigido que os **RMSE** fosse o menor possível, é realizado buscas constantes por algoritmos que apresentem uma baixa quantidade erros nas suas inferências porque isto agrega valor para o negócio gerando maior quantidades de clicks, curtidas, engajamento.

Para os recomendadores que trabalham estimando a classificação de um usuário para novos itens como é nosso caso é possível comparar a classificação estimada com a classificação real nos dados de teste. Um recomendador é então julgado com base no **RMSE**.

A Equação 5.1 abaixo demonstra como o RMSE é calculado:

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(p_i - r_i)^2} \tag{5.1}$$

RMSE é uma regra de pontuação quadrática que também mede a magnitude média do erro. É a raiz quadrada da média das diferenças quadradas entre previsão e observação real. Os valores podem variar de 0 a ∞ e são indiferentes à direção dos erros. **RMSE** é negativamente orientado, o que significa que valores mais baixos são melhores.

Tirar a raiz quadrada dos erros quadrados médios tem algumas implicações interessantes para o RMSE. Como os erros são elevados antes da média, o RMSE atribui um peso relativamente alto a erros grandes. Isso significa que o RMSE deve ser mais útil quando erros grandes são particularmente indesejáveis.

O RMSE não aumenta necessariamente com a variação dos erros. O RMSE aumenta com a variância da distribuição de frequência das magnitudes de erro.

5.3.2 Teste U de Mann-Whitney

Quando se dispõe de uma amostra pequena e a variável numérica não apresenta sabidamente uma variação normal (ou não dá para ser verificada satisfatoriamente), ou ainda, quando não há homogeneidade das variâncias (embora exista uma correção no teste t que considera as variâncias desiguais), o teste t não é apropriado.

Nessa situação, pode-se utilizar o teste não paramétrico de Mann-Whitney. O teste de Mann-Whitney foi desenvolvido primeiramente por F. Wilcoxon em 1945, para comparar tendências centrais de duas amostras independentes de tamanhos iguais (Wilcoxon, 1945). Em 1947, H.B. Mann e D.R. Whitney generalizaram a técnica para amostras de tamanhos diferentes (Mann and Whitney, 1947). O teste de Mann-Whitney (*Wilcoxon rank-sum test*)

é indicado para comparação de dois grupos não pareados para se verificar se pertencem ou não à mesma população e cujos requisitos para aplicação do teste t de Student não foram cumpridos. Na verdade, verifica-se se há evidências para acreditar que valores de um grupo A são superiores aos valores do grupo B.

O teste U de *Mann-Whitney* pode ser considerado a versão não paramétrica do *Teste t*, para amostras independentes. Ao contrário do *Teste t*, que testa a igualdade das médias, o teste de *Mann-Whitney (U)* testa a igualdade das medianas. Os valores de U calculados pelo teste avaliam o grau de entrelaçamento dos dados dos dois grupos após a ordenação. A maior separação dos dados em conjunto indica que as amostras são distintas, rejeitando-se a hipótese de igualdade das medianas.

5.4 Resultados

Para avaliar os resultados foram utilizadas os métodos de avaliação:

- RMSE apresentado na seção 5.3.1.
- Teste estatístico de Mann-Whitney também comentado na seção 5.3.2.

Foram realizados 10 casos de testes começando com 50 usuários e cada novo teste foi incrementado mais 50 usuários até um total de 500. Os resultados e a distribuição serão detalhados nas subseções subsequentes.

5.4.1 *Root Mean Square Error (RMSE)*

Na Tabela 5.2 pode ser visualizado o nosso cenário de testes e os valores obtidos do RMSE para cada algoritmo. Com Figura 5.2 é possível analisar graficamente os resultados das comparações entre cada algoritmo agrupado pela quantidade de usuários.

O menor valor alcançado com RMSE foi obtido ao utilizar 100 usuários com um erro de 0,97 onde o valor do RMSE do KNN foi 1,1 vezes maior, o pior valor do RMSE obtido pela nossa proposta foi na primeira interação com 50 usuários, mesmo assim obtivemos um desempenho de 9,24% melhor nas predições.

Como a Tabela 5.3 demonstra a média de desempenho do RMSE do nosso algoritmo foi de 1,0625 contra 1,0166 do KNN, o valor mínimo obtido pelo KNN foi 1,02 com ponto máximo em 1,19. O desvio padrão foi de 0,02682 do algoritmo híbrido enquanto o KNN obteve 0,04900 o que caracteriza uma menor dispersão nos resultados.

Tabela 5.2 Resultados do RMSE para cada algoritmo.

Quantidade de Usuários	KNN	Algoritmo Proposto
50	1,19	1,07
100	1,06	0,97
150	1,07	1
200	1,06	1,01
250	1,08	1,02
300	1,05	1,02
350	1,03	1,01
400	1,02	1
450	1,04	1,02
500	1,04	1,03

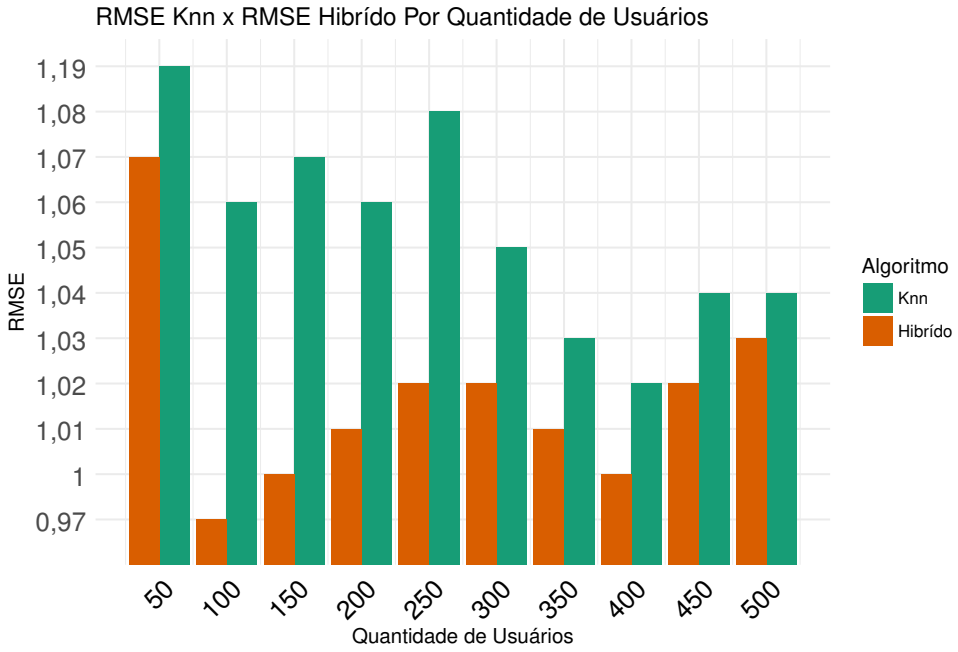


Figura 5.2 RMSE KNN X RMSE Híbrido.

Tabela 5.3 Estatísticas Descritivas do RMSE.

Estatística Descritivas	KNN	Algoritmo Proposto
Máximo RMSE	1,19	1,07
Mínimo RMSE	1,02	0,97
Desvio Padrão	0,04900	0,02682
Variância	0,002	0,001
Média	1,06	1,01

5.4.2 Resultados do Teste U de Mann - Whitney

Na subseção anterior 5.4.1 foi possível visualizar as diferenças de RMSE obtidas pelo nosso algoritmo contra a proposta clássica KNN. É interessante que possamos validar se os resultados que obtivemos na realidade trazem diferenças significativas que passem maior confiança sobre nosso algoritmo.

Foi apresentado na seção 5.3.2 o que o teste de *Mann - Whitney* representa e o que ele avalia. Em nosso caso temos um total de 10 amostras, número relativamente pequeno. A opção principal seria utilizar o teste *T Student* porém o ideal seria um número superior a 30 amostras pelo fato da sua média aproximar-se cada vez mais de uma distribuição normal.

Por os testes de normalidade não se confirmarem nas duas amostras então o indicado é o teste de Mann Whiteney. O teste de hipóteses é formulada da seguinte maneira:

- **H0** : Hipótese nula onde afirma que a distribuição do RMSE é a mesma entre os dois algoritmos, ou seja não existe diferença estatística significativa entre a mediana dos dois algoritmos.

- **H1** : Hipótese alternativa que nega a primeira hipótese, então sim existe diferença.

Os valores da mediana são 1,0185 para o algoritmo híbrido proposto e 1.0518 para o KNN. Os resultado desta análise pode ser observado na Figura 5.3.

Resumo de Teste de Hipótese

	Hipótese nula	Teste	Sig.	Decisão
1	A distribuição de RMSE é a mesma entre as categorias de Algoritmos.	Teste U de Mann-Whitney de amostras independentes	,004[1]	Rejeitar a hipótese nula.

São exibidas significâncias assintóticas. O nível de significância é ,05.

[1] A exata significância é exibida para este teste.

Figura 5.3 Teste de Mann-Whitney.

Então como é mostrado no teste, há evidências estatísticas, para afirmar que com 95% de confiança, a um nível de significação de 0,05%, como p valor é menor 0,004 < 0,05 a mediana do nosso algoritmo é maior que a mediana do KNN, os valores do algoritmo híbrido tendem a ser menor do que o KNN lembrando que para o RMSE quanto menor melhor e essa diferença é significativa estatisticamente.

5.5 Discussão

Neste trabalho foi demostrado a proposta de um algoritmo híbrido, com o objetivo de conseguir explorar relações entre os usuários, e modelar o perfil do usuário de um jeito simples e direto e que conseguisse trazer benefícios em termos de qualidade de predição com poucos metadados do usuário. Este conhecimento adicional nos permitiu explorar uma forma diferente de relacionar os usuários usando informações diretamente ligadas ao histórico disponível sobre ele, e com uma boa taxa de precisão.

Após analisar os resultados obtidos com as métricas utilizadas, podemos concluir que combinar as duas técnicas clássicas de recomendação FC e FB aumenta a performance da recomendação consideravelmente. Combinar atributos dos filmes com o perfil do usuário produziu melhores resultados. o que indica que analisar os filmes de maior interesse do usuário levando em considerações critérios podem extrair boas relações e inferir de forma bastante precisa novas predições com as vantagens de uma FH.

Todos os testes realizados apresentaram melhorias significativas, como podem ser visto nas Figuras 5.3 e 5.2, ao utilizar a descrição do filme coma uma string mestre que representa todos os interesses do usuário. Assim, podemos perceber o quanto a utilização de dados sobre o domínio pode ajudar na precisão da recomendação.

Este trabalho serve como um começo para explorar profundamente relações entre usuários e como podemos cada vez mais fornecer recomendações de qualidades com atributos dos filmes que é um domínio grande e que cada dia mais faz parte do cotidiano das pessoas. Atualmente, é crescente a demanda por produtos personalizados e que entendam a preferência e a necessidade de cada um. Na seção a seguir iremos apresentar pontos de melhorias.

5.6 Pontos de Melhoria

Os principais problemas na criação de um algoritmo com abordagem híbrida diz respeito ao tempo de performance que é uma das principais desvantagens como foi explicada no

Capítulo 2. Utilização de um *data set* maior com mais filmes com mais avaliações pare que se possa fazer uma quantidade maior de amostras e aplicar mais testes

Outra possibilidade é a utilização de outras fontes de conhecimento sobre o usuário, como redes sociais que possam fornecer mais conhecimento, como Twitter[7], Google Plus[8], e outras.

O modelo do usuário ainda tem possibilidade de ser alimentado com muito mais informações que podem ser obtidas criando novas relações entre o domínio. Então existe muito espaço para explorar e gerar novos dados seguindo a abordagem apresentada neste trabalho.

5.7 Sumário

Este capítulo apresentou os principais resultados obtidos durante o desenvolvimento dos experimentos para avaliar a implementação de um algoritmo híbrido em um domínio de filmes, utilizando extração automática de informações dos usuários. Inicialmente foram apresentadas as metodologias de avaliação utilizadas no estudo, data set, e métricas utilizadas. O estudo apresentado neste capítulo mostrou que a proposta tem potencial de crescimento e futuramente poderemos aplicar em um contexto real com feedback reais dos usuários.

[7] Twitter, http://www.twitter.com
[8] Google+, https://plus.google.com/

6 Conclusão

Neste trabalho foi apresentado um sistema de recomendação baseado em Filtragem Híbrida (FH). Inicialmente foi apresentado a motivação para a criação do sistema de recomendação, relatou-se as principais estratégias aplicadas atualmente os seus problemas, vantagens e desvantagens. Foi proposta uma solução para ajudar os usuários a encontrarem filmes de seu maior interesse e com uma maior personalização.

No Capítulo 2 descreve sobre os sistemas de recomendação, apresentando um histórico, discussão de conceitos quanto aos dados que são utilizados em um sistema além das tarefas desempenhadas pelos sistemas de recomendação e as suas principais técnicas.

No Capítulo 3 apresentou-se mais detalhadamente sobre as estratégias de recomendação, os modelos aplicados exemplificando os desafios de cada uma das abordagem e falando mais afundo sobre a abordagem híbrida.

O Capítulo 4 explana a proposta de solução híbrida para a recomendação de filmes. Foi apresentado os requisitos funcionais e não funcionais do software desenvolvido, a sua arquitetura e as tecnologias utilizadas. Esclarecendo todo o seu funcionamento para o usuário além de expor sobre os modelos do usuário e do item.

Para finalizar, o sistema de recomendação foi apresentado em detalhes e realizou-se uma avaliação experimental. Os métodos de avaliação e os resultados foram apresentados, mostrando a viabilidade da proposta, ou seja, a recomendação de filmes sendo o perfil do usuário modelado com metadados que podemos extrair de *Apis* disponíveis na Web, o que reflete na qualidade das recomendações. A seguir, mencionamos as contribuições principais deste trabalho e os trabalhos futuros.

6.1 Contribuições do trabalho

As principais contribuições deste trabalho são explicados a seguir:

- **Revisão:** Uma revisão comparativa entre sistemas baseados em conteúdo e filtragem colaborativa permeando a filtragem híbrida.

- **Relações Entre Usuários:** Estudo das relações entre usuários através de análise léxica dos perfil entre usuários.

- **Aplicação de código Aberto:** Uma aplicação de código aberto que implementa as técnicas propostas neste trabalho a ferramenta LibRec e LodWeb é um open source público completo e implementa tudo necessário para replicação dos testes.

- **Avaliações do Experimento:** Com o sistema de recomendação pronto, foram feitos testes para avaliar os dois modelos de recomendação. Foram realizados diversos testes, diferenciando os modelos de perfis do usuário e como os métodos de similaridade eram aplicados e os atributos do banco de dados dos itens. Fazendo uma comparação, é possível definir as condições de utilizar os modelos de recomendação e suas limitações.

6.2 Trabalhos Futuros

Mesmo com os bons resultados, é possível melhorar ainda mais o sistema de recomendação e proporcionar uma melhor experiência das seguintes formas:

1. **Aumento na base de dados de itens e avaliações:** Enriquecer essa base de dados com um dat set mais atual e com uma quantidade de metadados sobre os filmes maior, será possível melhorar e muito a modelagem do perfil do usuário.

2. **Criação de um sistema Web:** O desenvolvimento de um sistema web onde podemos aplicar o nosso modelo em tempo real e com usuários reais avaliando e dando um feedback mais interativo, vai propiciar um ambiente com maior impacto e utilidade na vida das pessoas, que por muitas estão em busca do produto certo.

3. **Utilização de Web Semântica**: A análise léxica entre as palavras chaves dos filmes é uma técnica que pode ser combinada e melhorada. Por exemplo, poderíamos calcular a similaridade entre as descrições atráves das ligações semânticas existentes entre os filmes, através de algoritmos como LDSD (*Linked Data Semantic Distance*) a plataforma Dbpedia [1] poderia ser utilizada para essa finalidade.

[1] https://wiki.dbpedia.org

4. **Cross-Domain:** Porque depender apenas de dados sobre filmes, enquanto atualmente os usuários utilizam diversas plataformas online e podemos oferecer e recomendar produtos de todas as variedades possíveis. Outra abordagem muito interessante seria avaliar todas essas relações e a potencia do modelo de recomendação em múltiplos domínios.

5. **Criação de um Ontologia:** Uma ontologia é um modelo de dados que representa um conjunto de conceitos dentro de um domínio e os relacionamentos entre estes. Uma ontologia é utilizada para realizar inferência sobre os objetos do domínio. Poderíamos modelar nosso conceito com novas informações de indivíduos, atributos e classes.

6.3 Sumário

Este capítulo apresentou um resumo de tudo que foi feito e discutido neste projeto. Mostrou-se as principais contribuições da nossa proposta de sistema de recomendação, os objetivos alcançados e os pontos de melhorias do nosso sistema de recomendação para os trabalhos futuros.

Bibliografia

Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, **17**(6), 734–749.

Ansari, Asim; Essegaier, S. K. R. (2000). Internet recommendation systems. *Journal of Marketing Research*, **37**.

Balabanović, M. and Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Communications of the ACM*, **40**(3), 66–72.

Basu, C., Hirsh, H., Cohen, W., *et al.* (1998). Recommendation as classification: Using social and content-based information in recommendation. In *Aaai/iaai*, pages 714–720.

Bennett, J., Lanning, S., *et al.* (2007). The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. New York, NY, USA.

Berry, M. W., Drmac, Z., and Jessup, E. R. (1999). Matrices, vector spaces, and information retrieval. *SIAM review*, **41**(2), 335–362.

Bobadilla, J., Ortega, F., Hernando, A., and GutiéRrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, **46**, 109–132.

Bozo, J., Alarcón, R., Peralta, M., Mery, T., and Cabezas, V. (2016). Metadata for recommending primary and secondary level learning resources. *Journal of Universal Computer Science*, **22**(2), 197–227.

Burke, R. (2000). Knowledge-based recommender systems. encyclopedia of library and information systems, a. kent, ed., vol. 69, supplement 32.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, **12**(4), 331–370.

Burke, R. (2007). Hybrid web recommender systems. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web*, chapter Hybrid Web Recommender Systems, pages 377–408. Springer-Verlag, Berlin, Heidelberg.

Cazella, S. C., Nunes, M., and Reategui, E. (2010). A ciência da opinião: Estado da arte em sistemas de recomendação. *André Ponce de Leon F. de Carvalho; Tomasz Kowaltowski..(Org.). Jornada de Atualização de Informática-JAI*, pages 161–216.

Duda, R. O., Hart, P. E., and Stork, D. G. (2012). *Pattern classification*, volume 24. John Wiley & Sons.

Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, **35**(12), 61–70.

Good, N., Schafer, J. B., Konstan, J. A., Borchers, A., Sarwar, B., Herlocker, J., Riedl, J., *et al.* (1999). Combining collaborative filtering with personal agents for better recommendations. In *AAAI/IAAI*, pages 439–446.

Guo, G., Zhang, J., Sun, Z., and Yorke-Smith, N. (2015). Librec: A java library for recommender systems. In *UMAP Workshops*.

Gupta, P., Goel, A., Lin, J., Sharma, A., Wang, D., and Zadeh, R. (2013). Wtf: The who to follow service at twitter. In *Proceedings of the 22nd international conference on World Wide Web*, pages 505–514. ACM.

Harper, F. M. and Konstan, J. A. (2016). The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, **5**(4), 19.

Herlocker, J. L. (2000). Understanding and improving automated collaborative filtering systems. *University of Minnesota*, page 144.

Hofmann, T. (1999). Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc.

Lee Rodgers, J. and Nicewander, W. A. (1988). Thirteen ways to look at the correlation coefficient. *The American Statistician*, **42**(1), 59–66.

Lops, P., De Gemmis, M., and Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, pages 73–105. Springer.

Mann, H. B. and Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60.

Manzato, M. G. (2010). *Uma arquitetura de personalização de conteúdo baseada em anotações do usuário*. Ph.D. thesis, Universidade de São Paulo.

Olsson, T. (2003). *Bootstrapping and decentralizing recommender systems*. Ph.D. thesis, Uppsala universitet.

Popescul, A., Pennock, D. M., and Lawrence, S. (2001). Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 437–444. Morgan Kaufmann Publishers Inc.

Resnick, P. and Varian, H. R. (1997). Recommender Systems. *Communications of the ACM*, **40**(3), 56–58.

Ricci, F., Rokach, L., and Shapira, B. (2011). Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, **24**(5), 513–523.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2000). Application of dimensionality reduction in recommender system-a case study. Technical report, Minnesota Univ Minneapolis Dept of Computer Science.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM.

Soboroff, I. and Nicholas, C. (1999). Combining content and collaboration in text filtering. In *Proceedings of the IJCAI*, volume 99, pages 86–91. sn.

Su, X. and Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in artificial intelligence*, **2009**, 4.

Terwiesch, C. and Bohn, R. E. (2001). Learning and process improvement during production ramp-up. *International journal of production economics*, **70**(1), 1–19.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics bulletin*, **1**(6), 80–83.

Yang, C. C., Chen, H., and Hong, K. (2003). Visualization of large category map for internet browsing. *Decision support systems*, **35**(1), 89–102.