Uma Abordagem Híbrida para Sistemas de Recomendação Baseados em Filtragem Colaborativa.

Por

Alesson Bruno Santos Souza

Trabalho de Graduação

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

SALVADOR, Julho/2018

Universidade Federal da Bahia
Departamento de Ciência da Computação

Alesson Bruno Santos Souza

Uma Abordagem Híbrida para Sistemas de Recomendação Baseados em Filtragem Colaborativa.

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

Eu gostaria de agradecer e dedicar esse trabalho aos meus pais, que me apoiaram incondicionalmente e sem eles nada disso seria possível. Aos meus amigos Aléxia Victória e Logan Carvalho por suas revisões desta monografia. Ao meu amigo Lassion Laique por ter me ajudado em todo o processo, desde criação de gráficos a compartilhamento de ideias. Ao meu orientador Frederico Durão, por sempre ser muito prestativo, paciente e ter me passado conhecimento essencial para conclusão deste trabalho.

Filler text

—AUTHOR

Resumo

O crescimento constante do uso da Internet aumenta a quantidade de informações que trafegam na Web. Em meio a essa quantidade de dados, nem sempre é fácil encontrar conteúdo relevante, o que torna necessário um mecanismo de filtragem para os serviços disponíveis. Os Sistemas de Recomendação são uma das soluções para esse problema. Estes sistemas utilizam os dados dos usuários e itens para personalizar o conteúdo que será disponibilizado para um determinado usuário de acordo com sua preferência. Com isso, este trabalho visa minimizar alguns problemas encontrados em algoritmos que utilizam técnicas de Filtragem Colaborativa, propondo a extensão deles para uma abordagem híbrida. Nesta abordagem proposta, utilizamos resultados da predição de alguns algoritmos colaborativos junto com o resultado de nossa proposta, que utiliza a similaridade entre as descrições dos filmes preferidos dos usuários e gera uma nova predição.

Palavras-chave: Sistemas de Recomendação, Filtragem Colaborativa, Filtragem Híbrida.

Abstract

The constant growth in Internet use increases the amount of information that travels the Web. Amidst this amount of data, it is not always easy to find relevant content, which requires a filtering mechanism for the services available. Recommendation Systems are one of the solutions to this problem. These systems use user data and items to customize the content that will be made available to a particular user according to their preference. Therefore, this work aims to minimize some problems found in algorithms that use Collaborative Filtering techniques, proposing the extension of them to a hybrid approach. In this proposed approach, we use results from the prediction of some collaborative algorithms along with the result of our proposal, which uses the similarity between the descriptions of the users' favorite movies and generates a new prediction.

Keywords: Recommender System, Collaborative Filtering, Hybrid Filtering

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Acrônimos

SR	Sistema de Recomendação
FC	Filtragem Colaborativa
FB	Filtragem por Conteúdo
FH	Filtragem Híbrida
BPoissMF	Bayesian Poisson Factorization
HCF	Algoritmo Filtragem Híbrida

Lista de Códigos Fonte

1 Introdução

Com a popularização da Internet nos anos 90, uma grande quantidade de informação passou a ser compartilhada e consumida pelo mundo inteiro (Gehrke, 2002), gerando com o tempo uma sobrecarga de informação e uma variedade de conteúdo para que usuários tivessem acesso. Entretanto, em meio a tanto conteúdo e serviços disponíveis, as pessoas querem apenas ter acesso àqueles que lhe são relevantes. Encontrar o conteúdo desejado em meio a essa sobrecarga trata-se de uma tarefa cansativa e não tão objetiva, onde tornou-se necessária a filtragem desses dados em meio a essa sobrecarga de informação.

No contexto atual, onde mais da metade da população do mundo tem acesso à Internet, as pessoas a utilizam para buscar informações que as ajudem em tarefas diárias, sejam elas pessoais ou profissionais. Os Sistemas de Recomendação são técnicas computacionais que analisam o perfil do usuário e os compararam com produtos candidatos disponíveis para recomendar produtos de interesse do usuário. Estes sistemas têm auxiliado os usuários personalizando suas informações, tornando o processo de busca por conteúdo relevante mais eficaz no menor tempo possível. No Brasil, aproximadamente 70% da população tem acesso à Internet. Deste total, 73% utiliza a internet para pesquisar sobre um produto antes de realizar a compra. Isso mostra que a opinião e experiência de outros usuários é importante para que se tome uma decisão em relação a compra de um produto. Acredita-se que 80% dessas vendas tenham a influência de Sistemas de recomendação.[1]

A recomendação pode ser gerada baseada em algumas técnicas existentes: filtragem colaborativa, filtragem por conteúdo, baseado em conhecimento, entre outras. Filtragem colaborativa é uma das técnicas mais bem-sucedidas na área da recomendação. Segundo Herlocker (2000) essa técnica foi feita para solucionar limitações existentes na filtragem baseada em conteúdo. Ela baseia-se em informações passadas pelo próprio usuário e associa isso ao histórico de outros usuários que tenham perfis similares, assim conse-

[1] http://www.convergenciadigital.com.br

guindo gerar a recomendação, descartando totalmente a necessidade de conhecimento sobre o item como é feita na filtragem por conteúdo. Por exemplo, se um indivíduo A tem interesse pelo mesmo filme de um individuo B, através da similaridade entre os filmes é provável que eles tenham interesses em comum em outros itens também. O Tapestry (Goldberg *et al.*, 1992) é um dos primeiros Sistemas de Recomendação baseados nessa técnica, que foi projetado para recomendar documentos de notícias para grupos de usuários.

O objetivo do Sistema de Recomendação é que a partir de grandes quantidades de informação, seja possível oferecer ao usuário aquilo que seja do seu interesse (Ricci *et al.*, 2011a). Seja essa informação um filme, música, produtos a serem comprados na Internet como celulares, computadores, entre outros. Evitando assim uma grande exposição de informação não relevante ao utilizador, o que tornaria todo o processo de busca uma tarefa demorada e pouco produtiva.

1.1 Motivação

O processo da recomendação baseado em usuário direciona conteúdo relevante de outros usuários que tenham perfis similares para o usuário atual, ou seja, dos seus vizinhos mais próximos. Ele depende de como os dados são coletados e como são repassados do usuário para o sistema no processo de avaliação da informação. Porém nem sempre os usuários deixam seu feedback ou deixam uma avaliação no conteúdo. Essa coleta de dados pode ser feita implicitamente, onde o sistema coleta padrões de navegação do usuário como, por exemplo, quando se visualiza um video no *Youtube* sobre certo conteúdo mesmo que você não o classifique, porém vale ressaltar que a recomendação na maioria das vezes depende das avaliações dos usuários. Só o *Youtube* em Setembro de 2016 tinha 1.3 bilhões usuários e tinha uma média de aproximadamente 5 bilhões vídeos vistos todos os dias[2], o que torna extremamente eficiente esse tipo de coleta de dados para conhecer os usuários. Pela grande quantidade de visualizações de vídeos mesmo que esses usuários não façam nenhum tipo de avaliação direta aos mesmos.

Outra maneira é de forma explícita, onde as informações cedidas pelo usuário irão traçar seu perfil. Por exemplo, quando se faz uma conta no *Netflix* [3] e o sistema te pede informações diretas como qual seu gênero de filme favorito, ou até mesmo redes sociais que são presentes para a maioria das pessoas no dia a dia como o *Twitter* que tem em

[2]https://www.statisticbrain.com/youtube-statistics/
[3]http://www.netflix.com

média 58 Milhões de tweets todos os dias[4] e o *Facebook* que tem em média 1 milhão de compartilhamentos de links a cada 20 minutos[5].

Essas informações passadas pelos usuários dessas plataformas, concorda com informações: sejam implícitas ou explícitas, são de grande importância para que seja feita uma recomendação eficiente. Pois para que um sistema tenha um bom desempenho nas suas recomendações é essencial que ele se adéque ao usuário com base nas informações passadas pelo mesmo. Por isso, esse trabalho tem o objetivo de minimizar os problemas encontrados quando um usuário não faz nenhum tipo de avaliação, mostrando que uma boa estrategia de modelagem do perfil do usuário torna o Sistema de Recomendação mais eficiente, pois ele salva e atualiza as informações coletadas para fazer a análise de similaridade entre usuários.

1.2 Descrição do Problema

O problema tratado neste trabalho baseia-se na dificuldade que os Sistemas de Recomendação por filtragem colaborativa têm em predizer sobre o que os usuários têm preferência quando não há informações suficientes sobre eles. Os usuários devem fornecer a um Sistema colaborativo, de forma implícita ou explicita (Ricci *et al.*, 2011b), informações de suas preferencias e interesses para que o sistema possa realizar sua recomendação com mais precisão e confiabilidade.

Os sistemas de filtragem colaborativa esperam que os usuários especifiquem a relação da predição e suas opiniões (Herlocker, 2000). Porém, geralmente os usuários avaliam poucos itens, o que acaba gerando uma esparsidade nos dados. Ou seja, uma matriz de similaridade entre itens e usuários com vários campos sem avaliação, o que dificulta a predição (Schafer *et al.*, 2007). Para que um item seja recomendado para um usuário é necessário que ele tenha sido avaliado por usuários similares a ele. Com isso, um item que nunca recebeu avaliação não será recomendado.

Outro cenário seria quando um usuário acaba de entrar em um Sistema de Recomendação. Por ser novo, o usuário nunca fez algum tipo de avaliação nos itens disponíveis, assim o Sistema de Recomendação não consegue construir um perfil para o usuário atual. Como já visto, uma das formas das recomendações serem realizadas é encontrando os vizinhos mais próximos do usuário, e como neste cenário ele ainda não realizou avaliações, não existirão vizinhos.

[4]https://www.statisticbrain.com/twitter-statistics/
[5]https://www.statisticbrain.com/facebook-statistics/

Considere um novo cenário onde trata de um novo item no sistema. Este item não poderá ser recomendado pois ao entrar no sistema ele nunca foi classificado por nenhum usuário, fazendo com que o item fique em *stand by* até que tenha avaliações suficientes para que seja recomendado para algum usuário que o sistema entenda como relevante para o mesmo. Diante desses cenários é difícil para o sistema fazer qualquer tipo de recomendação que seja confiável.

1.3 Objetivo

Partindo dos problemas citados, o objetivo deste estudo é propor uma solução híbrida que minimize as desvantagens de um sistema colaborativo baseado no usuário e nas informações que descrevem o conteúdo do item, onde ele não disponibiliza suas informações para o sistema. Os itens abaixo são objetivos específicos deste trabalho:

- Recomendar itens diretamente ligados ao histórico disponível sobre o usuário.
- Gerar boas recomendações para usuários atípicos.
- Gerar 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. Sistema de Recomendaçãos (SRs);
- A extensão de um algoritmo de recomendação colaborativa a partir da biblioteca LibRec[6]
- 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:

[6]https://www.librec.net/dokuwiki/doku.php

No Capítulo 2 a seguir apresentará o referencial teórico no qual nos aprofundaremos na abordagem de Modelagem de Perfil de Usuário, onde discutimos sua importância para os Sistemas de Recomendações e problemas encontrados.

No Capítulo 3 será abordado os conceitos de Sistema de Recomendação, tipos de técnicas onde focaremos na filtragem colaborativa baseado em usuário, técnicas e fórmulas de similaridades que são aplicadas nessa área e exemplo de algumas métricas de avaliação.

No Capítulo 4 é abordada a proposta deste trabalho, onde será mostrado os requisitos, o modelo de usuário proposto, modelo de itens, o referencial teórico dos algoritmos colaborativos utilizados, as equações do algoritmo proposto e as tecnologias utilizadas.

No Capítulo 5 abordaremos a metodologia deste trabalho, o conjunto de dados utilizado, as métricas utilizadas na avaliação, os resultados obtidos com os testes e uma breve discussão sobre eles.

No Capítulo 6 mostraremos as contribuições desse trabalho e os trabalhos futuros que podem ser abordados.

1.6 Sumário

Este capítulo fez uma breve introdução sobre o que são sistemas de recomendação, como os sistemas colaborativos são relevantes e estão inseridos nas tarefas do dia a dia a motivação por trás deste estudo. Também podemos notar alguns problemas que buscamos resolver aperfeiçoando as técnicas existentes através da extensão do algoritmo a partir da biblioteca LibRec, visando uma melhoria nos resultados de predição nos cenários abordados na problemática descrita nesse capítulo.

2

Modelagem de Perfil de Usuário

Este capítulo discute a modelagem do perfil do usuário, e sua importância para uma recomendação eficiente. Serão mostrados os métodos usados pelo sistema para a coleta de informações passadas pelo usuário, como os perfis podem sofrer mudanças ao longo do tempo e os principais desafios para que seja construído um perfil de usuário em certos cenários que abordaremos.

2.1 Modelagem do Usuário

Para desenvolver um sistema que gere uma grande satisfação aos usuários, é necessário que ele tenha um padrão visto como modelo geral, onde os usuários consigam realizar uma tarefa estabelecida chegando ao objetivo final, como fazer uma busca por um livro em um sistema de uma biblioteca. Porém, a maior desvantagem dessa abordagem é justamente a homogeneidade pressuposta no modelo geral, partindo da premissa que todos os usuários envolvidos no sistema seguem um certo padrão. Um sistema que se adéque ao usuário atual seria mais produtivo tanto do ponto de vista do usuário quanto do ponto de vista da máquina. Por isso, a modelagem de perfil do usuário é uma etapa importante para um sistema personalizado (Carmagnola *et al.*, 2011). Pois, é nela que são armazenadas e atualizadas todas as informações passadas pelo usuário, além de seus interesses e objetivos.

2.1.1 Dados de Usuário Implícitos contra Explícitos

A precisão dos dados do modelo do usuário afeta diretamente o desempenho das personalizações em sistemas adaptáveis aos usuários. Assim, é necessário uma estratégia eficiente para obter os dados fornecidos pelo usuário, seja de forma implícita ou explícita.

(Ricci *et al.*, 2011b).

Os dados explícitos como fonte de informação são mais confiáveis para demonstrar a preferência do usuário do que dados implícitos. Um usuário expõem seu interesse para um sistema quando há algum mecanismo disponibilizado para isso. Como por exemplo, deixar um comentário em um certo conteúdo, ou por classificação de estrelas uma certa escala. A *Amazon*[1] oferece um sistema de classificação de estrelas de 1-5 em seus produtos onde os clientes podem expressar suas opiniões e ter suas preferências analisadas. Enquanto o *Youtube* define as preferências do usuário utilizando um sistema binário, indicando preferência (curtir) ou não preferência (não curtir). Outras plataformas, como a *Netflix*[2], usam um abordagem mais direta: após o usuário se cadastrar no serviço, uma das etapas é fazer um pequeno questionário com o objetivo de conseguir informações sobre os principais interesses do usuário como podemos ver na Figura 2.1. Com isso, ela constrói um perfil baseado nas informações passadas e já consegue gerar recomendações desde o início do uso do sistema.

Figura 2.1 O usuário informa explicitamente ao sistema quais os gêneros de filmes mais relevantes para ele.

Entretanto, nem todos os sistemas oferecem mecanismos explícitos de coleta de dados, portanto métodos de controle de dados implícitos que rastreiam a atividade do usuário são utilizados (Hu *et al.*, 2008). Uma lista de tipos de dados implícitos é apresentada por Hu *et al.* (2008) e inclui navegação e histórico do site, log de compras do aplicativo de comércio eletrônico, etiquetagem, eventos de interação da interface e padrões de pesquisa.

[1] www.amazon.com
[2] www.netflix.com

Por exemplo, um usuário que costuma comprar celulares da mesma marca provavelmente gosta do design, da funcionalidade e estilo desse celular. Um usuário que constantemente atribui muitas tags em um determinado tópico provavelmente tem interesse nesse tópico.

Esse tipo de abordagem implícita é a mais frequente em nosso dia a dia, o Google[3] é uma ferramenta que faz muito bem o uso desse recurso. Com apenas alguns cliques e umas pesquisas feitas na plataforma, ele já consegue inferir várias recomendações em diversos outros sites que utilizam de seus recursos para obter essas informações disponibilizadas implicitamente pelo usuário. Isto é muito comum e fácil de perceber quando queremos comprar algum produto pela internet. Devido as pesquisas que serão realizadas sobre o mesmo, começará a aparecer produtos similares em outros sites que o usuário geralmente utiliza frequentemente, como Facebook[4] e Youtube[5].

Nosso estudo se concentra-se em modelos adequados para dados implícito, pois não há dados explícitos já que o usuário não oferece informações sobre seu perfil ao entrar no sistema.

2.1.2 Atualização de Preferência com o Tempo

Com o passar do tempo, as preferências dos usuários e seus objetivos podem mudar, suas habilidades, seu conhecimento, e acompanhar essas mudanças é um problema típico para os Sistemas de Recomendação Picault *et al.* (2011). Por exemplo, um usuário adolescente tem preferência por filmes do gênero animação, porém com o passar do tempo esse adolescente pode mudar seu gênero favorito em filmes para ação. Então recomendar filmes do gênero de animação passaria não mais a ser o ideal para esse usuário. Apesar de algum momento da sua vida ele gostar de filmes de animação, não significa necessariamente que no futuro este gênero continuará a ser o seu favorito.

Uma vez que as preferências do usuário podem mudar ao longo do tempo, o desempenho da maioria dos sistemas recomendadores depende das atualizações de informações. No entanto, um sistema de recomendação deve executar com precisão a recomendação desde a sua primeira execução para que os usuários possam confiar no serviço e continuar a usá-lo Ricci *et al.* (2011b); Picault *et al.* (2011). Por isso um Sistema de Recomendação deve continuar coletando informações dos seus usuários sempre, seja essa coleta feita de forma explícita, fazendo questionários periodicamente para identificar quais preferências mudaram ou não, ou mesmo de forma implícita observando o comportamento e uso do

[3] www.google.com
[4] www.facebook.com
[5] www.youtube.com

sistema pelo usuário.

Este problema é resolvido adicionando uma função de decadência do tempo (decayRec), definida na Equação 6.

$$decayRec(i) = r_{u,i}.\alpha^{(\theta - d_{u,i})} \quad (2.1)$$

A definição de decayRec envolve a tupla: (i, r, d), onde i representa um item que foi recomendado com uma determinada pontuação r em um determinado momento ou data d. Assim, a lista de recomendações ao item levam em consideração duas datas: a data atual e a data passada da recomendação. Quanto mais velha for a sugestão, menos relevante será.

Para cada item, precisamos diminuir sua relevância de acordo com a diferença da data atual θ e da data $d_{u,i}$ que o item foi avaliado pelo usuário. O $r_{u,i}$ (que corresponde a avaliação do usuário u ao item i) é multiplicado pela função de decaimento $\alpha^{(\theta - d_{u,i})}$, onde α [0, 1]. A variável α representa o coeficiente de decaimento. Quanto maior o número de dias, menor será a relevância da classificação dada pelo usuário ao item e consequentemente. O α define quão rápido o decaimento será, quanto mais próximo de 1, mais lento será, caso seja 1, o decaimento será constante.

A mudança de preferência com o tempo foi abordado neste estudo, aplicando um simples fator de decomposição para informações de modelos de usuários excluindo qualquer dado com mais de dois meses. Esse cuidado deve ser diretamente proporcional à frequência da atividade do usuário, uma vez que mais atividades levam à mudanças dinâmicas de interesses (Ricci *et al.*, 2011b; Picault *et al.*, 2011).

2.2 Modelo de Usuário

Seria o modo de classificar os usuários em grupos considerando a seu comportamento para o sistema, ou seja, qual a ação e comportamento do usuário ao usar o sistema. Por exemplo, o comportamento do usuário ao acessar a categoria de filmes de ação com maior frequência que as outras, a busca no sistema por um filme de determinado ator ou diretor, entre outros metadados, a classificação positiva feita pelo usuário a determinados filmes. Todos esses comportamentos tornam provável que esse usuário tenha maior interesse nestes determinados filmes pela frequência de acesso ou pela avaliação que ele dá aos mesmos. O modelo de usuário que será usado neste trabalho é focado na classificação dada pelo usuário aos filmes do sistema, uma vez que a filtragem colaborativa é a principal técnica usada nesse trabalho e ela depende de que façam avaliações dos filmes.

	Lagoa Azul	X-men	Matrix	Jogos Mortais	Click	Procurando Nemo
Pedro	?	5	5	?	3	3
Danielle	3	1	2	1	?	4
Renan	5	3	?	2	4	5
Ana	3	?	3	2	3	?
Thiago	1	5	5	4	3	?
Julia	?	3	3	2	5	4

Tabela 2.1 Exemplo de uma matriz Usuário X Filme, onde a primeira coluna corresponde aos usuários e a primeira linha aos filmes. Os campos onde então os números (1-5) corresponde a nota que o usuário deu ao respectivo filme da coluna. A interrogação (?) corresponde a falta da nota do usuário em relação ao filme. É de uma matriz similar a essa que serão feitos os cálculos de similaridade entre os usuários e previsão de notas.

Para que a previsão seja gerada segue-se alguns passos como a construção da vizinhança que consiste em agrupar os usuários que tenham preferências semelhantes, sendo que os k-vizinhos mais próximos de determinado usuário u, serão os considerados por avaliarem determinados itens similarmente. Por fim, as classificações desses vizinhos pelo determinado item i são agrupadas no valor da previsão final de $\hat{r}_{u,i}$, sendo esse agrupamento a classificação média sobre a classificação dos vizinhos ou a soma Ponderada das classificações com similaridade dos usuários usando pesos entre eles (Adomavicius and Tuzhilin, 2005).

$$\hat{r}_{u,i} = \text{aggr}_{u' \in \hat{C}} r_{u',i} \tag{2.2}$$

Onde $\hat{C}$ indica os N usuários que possuem interesses similares ao usuário u, o $\hat{r}_{u,i}$ corresponde a classificação predita do usuário u para o item i.

Para que seja determinado se um filme é ou não relevante para um usuário, o sistema depende da nota que o usuário atribuiu para esse filme. Usando uma escala de 1-5, onde 1 significa muito ruim e 5 significa ótimo, representaremos esse interesse da seguinte forma:

$$r_{u,i} > 4 \tag{2.3}$$

Onde $r_{u,i}$ como já vimos na seção anterior, corresponde a avaliação do usuário u ao item i, que no caso é um filme. Para que um filme seja considerado bom para o sistema, a nota/avaliação do filme deve ser maior que 4 (nota > 4). Em nosso sistema, será recomendado os filmes entre os usuários que tenham similaridades entre si, que atribuíram uma nota nesta condição. Como já dito anteriormente, esse modelo depende

totalmente da avaliação dada pelo usuário ao filme. Porém existem cenários em que o usuário não avalia determinado filme, o que traria um problema para o sistema inferir se o usuário atual acha o filme bom ou ruim. Veremos mais a diante, alguns métodos usados para que seja prevista a avaliação do usuário para determinados filmes.

2.3 Obstáculos da Modelagem

Para que seja construído um modelo de perfil do usuário é preciso que o usuário forneça informações relevantes, seja ela de maneira implícita ou explicita. Porém, nem sempre as expectativas são concretizadas. Serão discutidos a seguir cenários onde é difícil construir um perfil de usuário adequado.

2.3.1 Partido a frio

O problema conhecido como *Partida a frio* (*cold start*), deixa o usuário sem sugestões de produtos até que ele indique algum tipo de informação ao sistema por meio de mecanismos de classificações ou até mesmo acesso de navegação, impossibilitando que um perfil seja modelado para tal usuário. Esse problema pode ser dividido em três cenários: quando há novos itens no sistema, novos usuários e novas comunidades (Bobadilla *et al.*, 2013).

- **Novos itens:** Quando um item ou conteúdo novo é cadastrado no Sistema de Recomendação e ainda não possui nenhum tipo de avaliação ou histórico de acesso (Adomavicius and Tuzhilin, 2005), este item fica sujeito a ser esquecido e por consequência não será recomendado a nenhum usuário, assim a sua condição não mudará e acabará estagnado no sistema.

  Esse tipo de problema não só ocorre quando um novo item é cadastrado em um Sistema de Recomendação, ele persiste quando determinado item não é avaliado suficientemente, não sendo recomendado como consequência. Esses dois cenários citados, por exemplo, são bem frequentes em sites de notícias em geral, pois como a todo instante ocorre atualização de conteúdo, muitos podem não ser vistos ou avaliados e assim passarão despercebidos pelos usuários e não serão recomendados.

- **Novas comunidades:** Ocorre quando um novo Sistema de Recomendação é disponibilizado para o uso. Por ser novo é provável que ainda tenha poucos usuários o utilizando, fazendo com que os seus produtos não tenham avaliações suficientes para que se gere recomendações eficientes. Existem duas abordagens para minimizar este problema que são: incentivar os usuários a avaliarem os produtos de

algumas maneiras, seja ela implicitamente ou explicitamente ou usar filtragem colaborativa quando já houver usuários e avaliações suficientes (Bobadilla *et al.*, 2013).

- **Novos Usuários (Rashid *et al.*, 2008):** Quando um usuário se cadastra ou simplesmente começa a usar um Sistema de Recomendação, ele ainda não fez nenhum tipo de avaliação ao conteúdo. Por esse motivo, o Sistema de Recomendação não consegue fazer uma recomendação personalizada utilizando filtragem colaborativa, pois com a falta de dados do usuário o sistema não consegue construir um perfil para o mesmo. Mesmo quando o usuário faz suas primeiras avaliações a um conteúdo e espera que o sistema ofereça recomendações personalizadas, elas não ocorrem. Pelo fato do número de avaliações não ser suficiente para fazer recomendações confiáveis baseadas em filtragem colaborativa. Portanto, novos usuários podem sentir que o sistema não oferece o serviço que eles esperavam e podem parar de usá-lo (Rashid *et al.*, 2008).

  Um bom exemplo desse problema é quando um usuário entra no *Youtube* pela primeira vez sem ser cadastrado, como mostra a Figura 2.2. Como a plataforma não contém nenhum tipo de informação do usuário, ela não consegue recomendar um conteúdo personalizado, então ela oferece conteúdos que estejam mais relevantes para os usuários que já são cadastrados, porém isso não necessariamente agradará um novo usuário. Desse modo, a plataforma tentará obter informações do usuário por meio da coleta de dados implicitamente, esperando com que o usuário comece a clicar em vídeos de seu interesse para que comece a traçar um perfil para o mesmo.

  Um outro cenário que ocorre o problema de novos usuários na mesma plataforma é quando o usuário é recém-cadastrado. Nesse caso o *Youtube* tem poucas informações do usuário, como a idade e sua localização que foram passadas pelo usuário de forma explícita no momento do cadastro. Porém as informações ainda não são suficientes para que seja construído um perfil concreto para esse usuário e consequentemente ainda não serão feitas recomendações confiáveis ao mesmo.

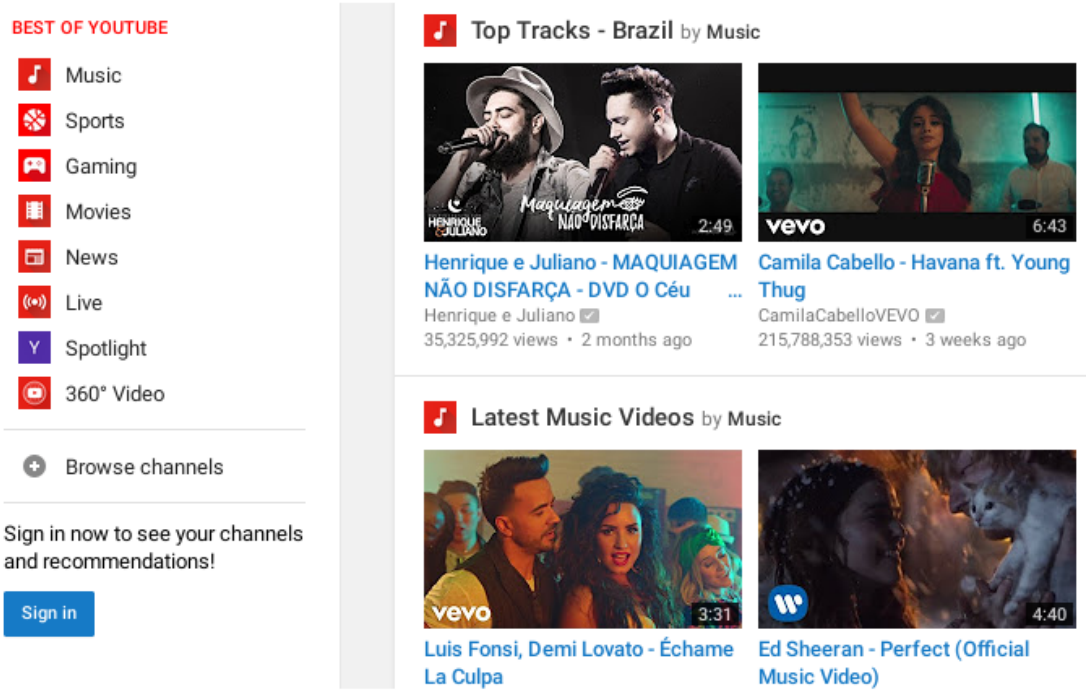

Figura 2.2 Um usuário sem cadastro no Youtube não recebe recomendações personalizadas pois ainda não fez nenhum tipo de avaliação nos vídeos. O recomendado pela plataforma é que o usuário faça um cadastro para receber recomendações.

Esse trabalho foca em minimizar esse tipo de problema, onde um filme que tenha poucas avaliações ou até mesmo nenhuma por ser um item novo, consiga ser recomendado. Evitando assim que o item fique em *stand by* e que se ocorra o problema de *Cold Start* (Partido a Frio).

2.3.2 Ovelha Negra

Usuários possuem gostos e interesses particulares e nem sempre se encaixam entre a maioria, ou seja, possuem gosto incomum e não encontrarão usuários com interesses semelhantes aos seus, com isso, as recomendações podem se tornar pobres (Reategui and Cazella, 2005). Os usuários podem ser classificados de 3 formas que representam a co-relação entre os usuários (McCrae *et al.*, 2004):

- **Ovelha Branca:** São aqueles usuários que possuem relação com vários outros usuários, ou seja, são os usuários ideais, pois o Sistema de Recomendação consegue fazer recomendações eficientes.

- **Ovelha Negra:** São basicamente os usuários opostos aos Ovelhas Branca. Um usuário que não tem interesses semelhante aos demais usuários do sistema, com isso dificultando que o sistema consiga fazer algum tipo de recomendação relevante para o mesmo.

- **Ovelha cinza:** Esses usuários são os mais problemáticos para um Sistema de Recomendação. Eles concordam e discordam com outros usuários do sistema, com isso eles podem gerar recomendações estranhas para outros usuários e pode ser igualmente difícil de prever suas avaliações assim como um usuário classificado como Ovelha Negra (McCrae *et al.*, 2004).

2.3.3 Ramp-up

Ramp-up (aceleração) é um problema que geralmente ocorre em sistemas de filtragem colaborativa e também em filtragem por conteúdo. Este problema se baseia totalmente nas classificações dadas pelos usuários a itens do sistema. Como já vimos, a recomendação baseada em filtragem colaborativa depende de comparação entre usuários, então um usuário com pouca classificação no sistema ou sistemas que tenham uma quantidade baixa de usuários, dificultam a categorização deles (Burke, 2002).

O mesmo acorre com a Filtragem por Conteúdo, onde itens novos no sistema não possuem classificações ou itens que já estão no sistema e não foram classificados suficientemente. Esses itens dificilmente serão recomendados e podem ficar em *stand-by* por muito tempo ou serem esquecidos pelo sistema. Pois se nenhum usuário o classificar ele nunca aparecerá para outros usuários fazerem o mesmo. Um exemplo onde ocorre muito esse problema são em artigos ou sites de notícias, pois há um fluxo constante de novos conteúdos e os usuários avaliam apenas alguns Burke (2002).

2.3.4 Esparsidade de Matriz

Tendo como exemplo uma biblioteca de filmes, sabemos que nem todos os usuários que os assistem deixam sua classificação no mesmo. Quando um usuário ou filme é cadastrado, o sistema vai construindo uma matriz na qual relaciona o usuário com os filmes, essa matriz é chamada de interação usuário-item (Huang *et al.*, 2004). É nela onde fica guardada a classificação ou nota que cada usuário dá aos filmes.

Como já visto, a maioria dos usuários não avaliam os itens de um Sistema de Recomendação, isso gera mais um problema conhecido como esparcidade de matriz. Esse problema consiste na falta de avaliação de itens como podemos ver na Figura 2.3, onde é

representada por uma '?'. Podemos ver a grande esparsidade desta matriz, o que dificulta que seja encontrado um conjunto de usuários com avaliações similares (Melville *et al.*, 2002).

MATRIZ USUÁRIO-FILMES

USUÁRIO	FILMES						
	I	II	III	IV	V	VI	VII
A	4	?	?	3	?	?	?
B	?	3	?	?	?	1	?
C	?	?	?	?	3	?	2
D	?	?	5	?	?	4	?

Figura 2.3 Essa tabela representa uma matriz de classificação na escala de 1-5 entre usuários e filmes. Nos espaços onde se encontra o símbolo '?' significa que o usuário não avaliou tal filme em questão.

Esse tipo de problema não só ocorre porque os usuários não querem ou esquecem de avaliar os itens, mas também ocorre quando uma grande quantidade de usuários novos são inseridos no sistema ou quando a proporção de item por usuário é muito alta, assim vários itens ficam sem avaliações, o que faz persistir o problema de esparsidade (Melville *et al.*, 2002).

2.4 Sumário

Este capítulo apresentou como é feita a modelagem do usuário, como são obtidos os dados dele e quais obstáculos são encontrados. No capítulo Capítulo 3 serão abordados os tipos de Sistemas de Recomendação com o foco na filtragem colaborativa e suas estratégias.

3

Sistema de Recomendação

Recomendações são muitos úteis não só para a área computacional mas também para outras atividades do dia a dia. Nós compartilhamos informações e experiências o tempo inteiro com outras pessoas, sejam essas informações sugestões de um amigo ou experiências vividas pelos nossos parentes mais velhos. Muitas vezes, nós tomamos decisões sem ter alguma experiência prévia, portanto, a recomendação de fontes confiáveis nos ajuda a tomar decisões quando deparamos com várias opções de escolha. Os Sistemas de Recomendação são ferramentas de software automatizadas que auxiliam a tomada de decisões entre várias alternativas, sugerindo itens que poderiam ser de interesse (Mahmood and Ricci, 2009).

Sistemas de Recomendação tornaram-se indispensáveis para o *E-commerce* nos dias de hoje. Além de ajudarem usuários a encontrarem produtos de interesse entre várias ofertas disponíveis (Ricci *et al.*, 2011b), também promovem um grande número de vendas para as empresas. Com os aplicativos de redes sociais, os Sistemas de Recomendação podem obter e analisar informações dos usuários através de grupos e comunidades dos quais eles participam e aprender quais são as suas preferências individuais e/ou coletivas. A qualidade da recomendação depende da capacidade do sistema de fazer uma boa escolha para o usuário quando ele busca algo, como fazer a compra de um produto ou em realizar uma pesquisa sobre algum assunto, entre outros.

Os sistemas de recomendação são softwares que analisam informações disponíveis sobre itens, usuários e/ou contexto para sugerir um subconjunto de itens disponíveis que são interessantes para um usuário específico (Mahmood and Ricci, 2009), conforme resumido pela Figura 3.1. Os sistemas de recomendação podem ser categorizados considerando o algoritmo usado para prever quais itens devem ser recomendados (Silva, 2016). De acordo com Burke (2007), os sistemas de recomendação podem ser classificados nos seguintes grupos:

- **Baseado em Conteúdo:** Técnica que utiliza as informações de avaliação de um usuário a outros itens.
- **Filtragem Colaborativa (CF):** Técnica que recomenda itens aos usuários apenas baseando-se em classificação de outros usuários que tenham similaridade com o mesmo, sem utilizar informações sobre os itens.
- **Demográfica:** Utilizam informações demográficas sobre o usuário para recomendar itens.
- **Baseado em Conhecimento:** Técnicas que utilizam itens com base em inferências sobre os interesses e necessidades de um usuário e como os itens podem combinar esses interesses.
- **Baseado em Comunidade:** Técnica que recomenda itens com base nas informações encontradas em um perfil do usuário de uma rede social, como interesses de amigos, postagens, tags, entre outros.
- **Híbrido:** Técnica que utiliza duas ou mais das técnicas citadas acima com objetivo de minimizar as desvantagens que cada uma delas apresenta.

Geralmente, as recomendações são personalizadas para cada usuário com base em suas preferências e interesses ou por similaridade entre usuários (Mahmood and Ricci, 2009). Entretanto, também existem recomendações não personalizadas, que não levam em consideração o interesse e nem o perfil do usuário e sim na similaridade de itens. Geralmente a recomendação não personalizada é aplicada a comércios não eletrônicos, como revistas e jornais. Embora esse tipo de abordagem seja útil para algumas aplicações, nesse trabalho não será utilizada. Será utilizado a recomendação personalizada baseada em similaridade entre usuários.

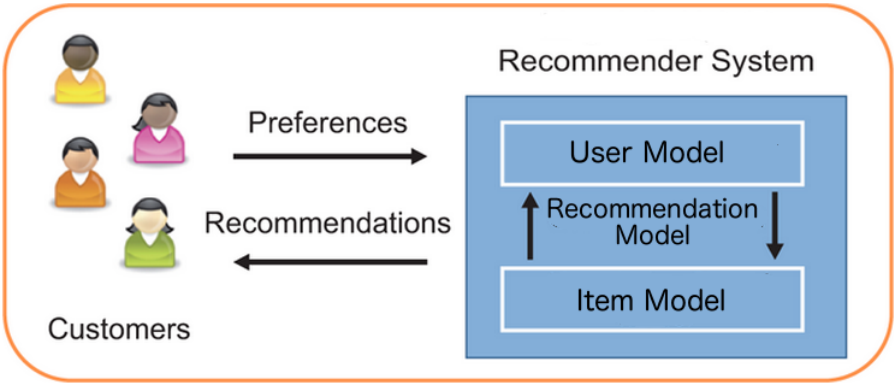

Figura 3.1 O sistema coleta os interesses dos usuários para criar um modelo de usuário, e em seguida, analisa a similaridade entre os modelos de itens para fazer a recomendação de itens relevantes.

3.1 Filtragem Colaborativa

A técnica de filtragem colaborativa é umas das mais eficientes em Sistemas de Recomendação, pressupondo que a preferência do usuário não irá variar ao longo do tempo sobre um item de seu interesse. Por exemplo, um Sistema de Recomendação baseado em filtragem colaborativa pode prever quais filmes um usuário provavelmente gostaria de assistir baseado no seu histórico de filmes já assistidos. Métodos como a filtragem baseada em conteúdo não podem criar recomendações personalizadas por não terem nenhum tipo de informação dos usuários, ao contrário da filtragem colaborativa. Para que seja feita a recomendação, os sistemas de filtragem colaborativa precisam analisar a similaridade entre os modelos de usuário e do item, onde normalmente essa informação é fornecida de forma explicita. Por exemplo, quando um usuário classifica um filme na Netflix com 5 estrelas em uma escala de 1-5, ele está sinalizando que gostou muito do filme e provavelmente o recomendaria. Mas caso o classificasse com 1 estrela, significaria que ele não gostou do filme.

Para quem esse filme deve ser recomendado baseado na classificação feita por este usuário? É justamente a resposta para essa questão que mostra a principal característica de um Sistema Colaborativo, pois o filme deve ser recomendado para outros usuários que tenham o perfil semelhante ao usuário atual. A abordagem colaborativa pode ser usada quando podemos assumir que um usuário tem um comportamento semelhante a outros usuários (Linden *et al.*, 2003), como mostrado na Figura 3.2. Essa semelhança entre os usuários chamamos de similaridade entre vizinhos.

A grande vantagem dessa técnica é que a precisão aumenta ao longo do tempo, já que

a tendência é o número de avaliadores (usuários) dos itens aumente. Entretanto, quando um usuário novo entra em um determinado sistema, caso ele não passe uma quantidade de informações significativas, o sistema não irá conseguir fornecer boas recomendações. Baseado em item, o mesmo terá que ser avaliado por uma quantidade relevante de usuários para que ele seja recomendado corretamente. Como já foi abordado no capítulo anterior, esse problema é conhecido como *Cold Star Problem*.

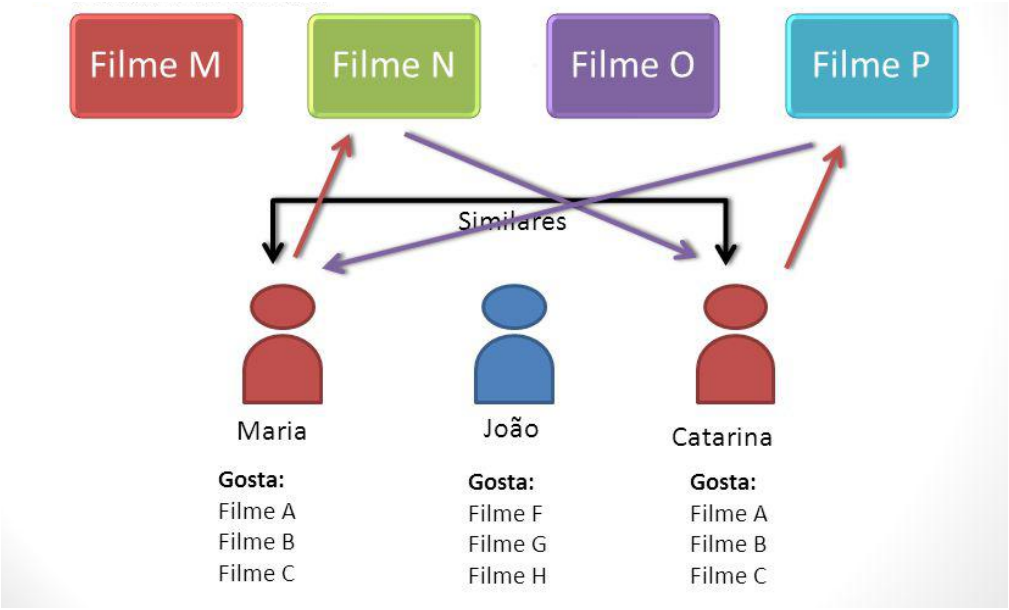

Figura 3.2 As usuárias Maria e Catarina tem gostos similares, por isso O filme N assistido pode Maria foi recomendado a Catarina e o Filme P assistido por Catarina foi recomendado a Maria. Essa imagem segue os conceitos de filtragem colaborativa. Imagem de Fausto J F B Gominho (2014)

Existem duas técnicas de filtragem colaborativa, são elas a *Baseada em Memória* e a *Baseada em Modelo.*

3.1.1 Técnica de Filtragem Colaborativa Baseada em Memória

Essa técnica consiste em usar os dados contidos na matriz usuário-item, que se encontra na memória do sistema, para diretamente gerar previsões. (Su and Khoshgoftaar, 2009) Basicamente essas previsões são geradas com base na similaridade computacional dos usuários encontrados na matriz para o usuário atual. Na filtragem colaborativa, o valor de previsão $\hat{r}_{u,i}$ é calculado explorando as preferências dos usuários que estão na matriz (Gedikli, 2013).

Existem dois métodos que são muito utilizados para que seja calculada a similaridade entre dois usuários, são eles *Correlação de Pearson* (Lee Rodgers and Nicewander, 1988) e *Similaridade Cosseno* (Berry *et al.*, 1999). Na computação, a correlação de Pearson mede a medida em que dois usuários ou itens relacionam-se linearmente entre si (Resnick *et al.*, 1994).

3.1.1.1 Correlação de Pearson

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

$$p = \frac{\sum_{i \in I}(r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I}(r_{u,i} - \bar{r}_u)^2} \cdot \sqrt{\sum_{i \in I}(r_{v,i} - \bar{r}_v)^2}} \tag{3.1}$$

onde $i \in I$ e corresponde ao item onde tanto o usuário u quanto o v avaliaram, $r_{u,i}$ é a classificação dada ao item i pelo usuário u e $r_{v,i}$ é a classificação dada ao item i pelo usuário v. $\bar{r}_u$ é a classificação média dos itens avaliados do usuário u e $\bar{r}_v$ é a classificação média dos itens avaliados do usuário v.

3.1.1.2 Similaridade do Cosseno

Na estatística, a similaridade cosseno é uma medida de semelhança entre dois vetores não nulos de um espaço de produto interno que mede o cosseno do ângulo entre eles. Na computação esses vetores são como os usuários, e da mesma forma a semelhança entre eles é calculada entre seus ângulos. Esse tipo de método é melhor empregado em sistemas colaborativos que possuam uma avaliação binária, por exemplo o Youtube, onde você avalia um vídeo com um "gostei" ou "não gostei". A Similaridade do Cosseno entre os usuários i e j, denominados sim(i,j) é calculada da seguinte forma:

$$sim(i,j) = \cos(\vec{i}, \vec{j}) = \frac{\vec{i}.\vec{j}}{||\vec{i}|| * ||\vec{j}||} \tag{3.2}$$

3.1.1.3 Calculo de Predição

Apos ter feito o calculo de similaridade entre os usuários, então pode ser feita a previsão da avaliação do usuário u para o item i com base na soma da classificação atribuída pelos usuários semelhantes u e v (Yao and Cai, 2017). Essa similaridade é denotada na Equação 3.2 como $s_{u,v}$. A soma ponderada final é dividida pela soma de similaridade para obter o valor de predição normalizado. A predição da avaliação do item i é calculada pela soma

ponderada de diferentes classificações dos usuários no item i. A predição $P_{u,v}$ é dada por onde $r_{v,i}$ é a classificação do usuário v no item i.

$$P_{u,i} = \frac{\sum_v (r_{v,i} * r_{u,v})}{\sum_v s_{u,v}} \tag{3.3}$$

Basicamente, essa abordagem mostra como o item é classificado pelos usuários similares. A soma ponderada é escalada pela soma dos termos de similaridade para se certificar de que a predição está dentro do intervalo predefinido. Esse método de previsão é chamado de Weighted Sum (Soma Ponderada).

3.1.1.4 Desafios

O principal problema com a abordagem de modelo baseado em memória é que diferentes usuários podem fornecer classificações em diferentes escalas. Um usuário pode classificar todos os itens positivamente, enquanto outro usuário pode avaliar todos os itens negativamente (Aggarwal, 2016). Pois sendo a interseção entre as avaliações feitas por dois usuários desta forma, o sistema pode julgar esses dois usuários como muito semelhantes ou muito diferentes. Também temos problema quando itens no sistema são avaliados igualmente pelos usuários, esses itens acabam dificultando a recomendação pois baseado nele, não dá para fazer uma análise precisa da preferência do usuário. Um bom exemplo disso são filmes clássicos que independente do seu gênero e outros metadados, os usuários que o assistem acabam o classificando como muito bons.

3.1.2 Técnica de Filtragem Colaborativa Baseada em Modelo

Nesta abordagem, os modelos são desenvolvidos usando algoritmos diferentes de mineração de dados, aprendizagem de máquinas para prever a classificação dos usuários de itens não classificados (Su and Khoshgoftaar, 2009). Essa técnica foi criada com o objetivo de aprimorar e resolver algumas limitações que temos na técnica baseada em memória. Alguns exemplos desses métodos baseados em modelos incluem modelos baseados em regras, métodos Bayesianos e modelos de Clusterização.

3.1.2.1 Modelos Baseados em Regras

Os modelos baseados em regras são visíveis em sistemas de lojas online. Ele analisa quais itens geralmente são comprados juntos pelos usuários, ou seja, quando alguém compra um item A geralmente também compro o item B, como podemos ver nas Figuras

3.3 e 3.4, que foram obtidas no site da loja *Saraiva* [1].

Figura 3.3 Imagem do produto que o usuário pretende comprar.

Figura 3.4 Logo abaixo do produto da Figura 3.3, o site recomenda os seguintes produtos. baseado em compras feitas por usuários anteriores que adquiriram o Box Harry Potter e consequentemente também comprando um ou mais desses produtos da imagem.

3.1.2.2 Modelos Bayesianos

As redes Bayesianos são modelos gráficos probabilísticos (um tipo de modelo estatístico) que representam um conjunto de variáveis e suas dependências condicionais através de um gráfico acíclico dirigido (grafo sem ciclo, ou seja para qualquer vértice v, não há nenhuma ligação dirigida começando e acabando em v).

O algoritmo Bayesiano de filtragem colaborativa simples usa uma estratégia Naive Bayes para fazer previsões para tarefas de filtragem colaborativa. Supondo que os recursos

[1] www.saraiva.com.br

sejam independentes, dada a classe, a probabilidade de uma determinada classe, com todas as características, pode ser calculada, e então a classe com maior probabilidade será classificada como a classe prevista (Miyahara and Pazzani, 2002). Por exemplo, a probabilidade de um item ser de certa classe para um usuário (como as classes gostei e não gostei) que não avaliou este item, é calculada através da avaliação de outros usuários e então a classe com maior probabilidade será dita como a classe prevista.

A Naive Bayes para filtragem colaborativa tem uma complexidade muito menor quando comparado com o método de que já vimos baseado em memória. Porem eles são bons para sistemas que possuem classes binárias (Miyahara and Pazzani, 2002). Caso seja aplicado em um sistema com multi-classe, como por exemplo o do nosso sistema que podemos avaliar um filme com notas de 1 a 5, essas classes serão convertidas em classes binárias o que pode ocasionar problemas de escalabilidade e perda de informações.

3.1.2.3 Modelos de Clusterização

Clusterização ou Agrupamento é uma técnica de agrupamento de conjunto de dados que sejam semelhantes entre si. Os modelos dividem a base de clientes em muitos segmentos e tratam a tarefa como um problema de classificação. O objetivo do algoritmo é atribuir o usuário ao segmento contendo os clientes mais similares (Linden *et al.*, 2003).

O armazenamento em cluster de grandes conjuntos de dados é impraticável, por isso a maioria dos aplicativos usam várias formas de geração de clusters gulosos. Esses algoritmos geralmente começam com um conjunto inicial de segmentos, que geralmente contêm um cliente selecionado aleatoriamente. Depois, eles repetidamente conectam os clientes aos segmentos existentes, geralmente com alguma provisão para criar novos ou mesclar os existentes (Bradley *et al.*, 1998).

3.2 Filtragem por Conteúdo

Os sistemas baseados em conteúdo são projetados para explorar cenários em que itens podem ser descritos com conjuntos descritivos de atributos. Nesses casos, as avaliações e ações de um usuário em outros filmes são suficientes para fazer recomendações significativas (Aggarwal, 2016). Diferentemente da filtragem colaborativa que é dependente de uma relação semelhante entre as preferências dos usuários do sistema com o usuário atual, o Sistema de Recomendação baseado em conteúdo depende da relação dos usuários com os itens no sistema. Portanto, as classificações feitas apenas pelo usuário atual a certos itens que são relevantes para o sistema, diferente do colaborativo que a classificação dos

demais usuários a certo item é importante para o sistema fazer uma recomendação. Esse tipo de abordagem tem uma eficácia melhor do que a colaborativa em itens que acabam de serem registrados no sistema, pois mesmo que ele não tenha sido avaliado ainda, o sistema baseado em conteúdo consegue recomendá-lo. Diferente do sistema colaborativo que gera o famoso problema de *Cold Start*.

Geralmente a recomendação nessa abordagem é feita analisando os atributos do determinado item, que pode ser uma característica ou um grupo a qual aquele item se encaixa. Por exemplo, utilizando o metadado gênero de um filme, um Usuário A pode avaliar positivamente um Filme B que seja do gênero comédia, em uma análise simples podemos dizer que esse usuário gosta de filmes do gênero de comédia, com isso outros filmes que tenham o gênero igual podem ser recomendados a esse usuário, portanto, a recomendação se baseia nas características encontradas no item de interesse do usuário como podemos ver na Figura 3.5. Esse tipo de atributo também pode ser extraído de outra forma como na textual, como em notícias na web em que se fala muito de futebol, assim o sistema irá reconhecer que o usuário se interessa por matérias que falam sobre tal tema. Independente se são extraídos de metadados ou conteúdo textual os atributos acabam correspondendo a um conjunto de palavras chaves (Adomavicius and Tuzhilin, 2005).

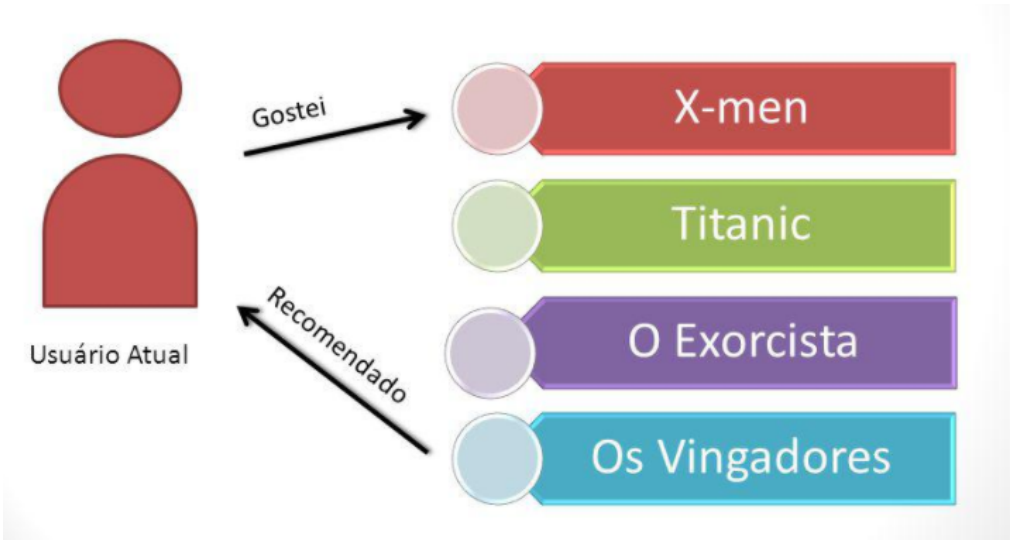

Figura 3.5 Os filmes desta imagem que tem características similares são X-men e Os Vingadores. O usuário demostrou interesse no filme X-men e por isso lhe foi recomendado o filme similar a ele. Imagem de Fausto J F B Gominho (2014).

3.2.1 Processo de Recomendação por Conteúdo

Para que seja feito o processo de recomendação na filtragem baseado em conteúdo é importante que sejam realizadas as seguintes etapas:

- **Analisador de Conteúdo:** Nesta etapa é feita uma análise de todo conteúdo pois a maioria das informações não são estruturadas, por isso é necessário um tipo de pré-processamento para que apenas as informações relevantes sejam extraídas (Ricci *et al.*, 2011a). Um bom exemplo são sinopses de filmes, onde algumas palavras como artigos (a, e, o) e preposição (de), não são relevantes para que seja feita uma recomendação baseada nelas.

- **Construção do Perfil:** Esta etapa coleta dados representativos do usuário, suas preferências e tenta generalizar esses dados, para construir o seu perfil. Geralmente, a estratégia de generalização é realizada através de técnicas aprendizagem por máquina (Degemmis *et al.*, 2007), que conseguem inferir se a avaliação do usuário foi feita de forma positiva ou negativa. Um exemplo seria quando um usuário faz um comentário em determinado filme, do qual o sistema faria um processamento deste comentário e inferir se o usuário gostou ou não do filme através dele.

- **Componente de Filtragem:** Esta etapa é a que faz a recomendação dos itens relevantes para o usuário com base no seu perfil montado na etapa anterior. Isso será feito usando métricas de similaridade (como as que vimos na seção anterior) entre os itens e o usuário.

Em resumo, o analisador coleta as características do item que já foram pré-definidas, logo após o sistema coleta o comportamento do usuário ao usar o sistema sendo essa coleta de dados implícitos ou explícitos. Com essas informações que será construído o perfil do usuário e então o sistema fará o cruzamento dos dados com os itens que tem maior similaridade com os mesmos.

3.2.2 Limitações de Filtragem por Conteúdo

Apesar de a filtragem por conteúdo ser muito usada e eficiente para várias aplicações, esta técnica apresenta uma série de limitações(Adomavicius and Tuzhilin, 2005), são elas:

- Os atributos dos itens geralmente precisam ser manualmente cadastrados, o que faz com que o sistema sempre tenha que passar por manutenção, o que demanda tempo

e mais custos. Esse fator também aponta problema quando se trata de mídias como som e vídeo, pois existe uma grande dificuldade na análise destes para extração automática de atributos.

- O sistema se limita a encontrar apenas itens que sejam similares ao que o usuário se interessou, ou seja, não é possível encontrar outros itens que podem vir a ser de interesse do usuário caso ele ainda não tenha demostrado esse interesse.

- Incapacidade na avaliação de um conteúdo quanto a dimensões subjetivas, como qualidade. Por exemplo, o sistema pode recomendar para um usuário uma loja não confiável, pois o sistema tem dificuldade de identificar isto, ele apenas vai recomendar a loja baseado na similaridade dos produtos encontrados na loja com de alguma outra que o usuário teve interesse.

3.3 Sistema de Recomendação Híbrido

O Sistemas de Recomendação Híbridos combinam duas ou mais técnicas de recomendação visando suprir as desvantagens que cada uma tem individualmente (Burke, 2002). Por exemplo, na técnica de filtragem colaborativa, como já vimos, temos o problema de *Cold Start*, já a tecnica de filtragem por conteúdo não ocorre tal problema como destacaremos a seguir.

3.3.1 Filtragem Colaborativa X Filtragem por Conteúdo

Podemos abordar as principais desvantagens de cada técnica em relação a outra. A principal desvantagem das abordagens baseadas em filtragem por conteúdo em comparação com as baseadas em filtragem colaborativa é que elas dependem muito dos metadados que descrevem os itens, assim essas informações devem ser obtidas e inseridas no sistema. Isso requer uma manutenção maior nesses sistemas, pois as informações sempre devem ser atualizadas.

Já na filtragem colaborativa temos a dependência de avaliação de outros usuários aos itens para que eles sejam recomendados, o que torna um item novo no sistema indisponível para recomendação até que os usuários o avaliem, causando um *stand by* no item até ele possuir classificação suficiente. Isso não ocorre na técnica de filtragem por conteúdo já que nela não é preciso que de classificação do item para que seja feita a recomendação do mesmo, apenas precisa de suas características para que ele seja associado com outros

itens que sejam de interesse do usuário. Como já vimos no Capítulo 2.3.1, esse problema é conhecido como *Cold Start*.

3.3.2 Combinação das Técnicas

Além do problema do *Cold Start*, também temos alguns outros citados no capítulo anterior, como o *Ovelha Negra* (Capítulo 2.3.2), *Ramp Up* (Capítulo 2.3.3) e alguns outros vistos no Capítulo 3.2.2. Para que seja tratado esses problemas, juntou as técnicas de filtragem colaborativa e filtragem por conteúdo. Existem combinações entre estas duas técnicas (Adomavicius and Tuzhilin, 2005), são elas:

- Implementação dos 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 colaborativas.

Estas combinações têm como objetivo suprir as limitações de cada técnica. Na primeira abordagem o sistema usa as duas técnicas, porém elas serão implementadas separadamente.

A segunda abordagem, tem como principal técnica a de filtragem colaborativa e usa características da filtragem por conteúdo. Desta forma, o problema de *Cold Star* seria resolvido usando uma das características da Filtragem por Conteúdo de conhecer os atributos dos novos itens que são inseridos no sistema e desta forma conseguindo recomendá-los mesmo sem classificação dos usuários ao mesmo.

A terceira abordagem é justamente o contrário da abordagem anterior, portanto tem como principal técnica a de filtragem por conteúdo e usa características da filtragem colaborativa. Desta forma, o problema de não recomendar itens que não são similares aos itens que o usuário demonstrou interesse fosse suprido, pois na filtragem colaborativa também é usado a similaridade entre os usuários, assim podendo recomendar itens que sejam de interesse dos outros usuários ao usuário atual.

Na quarta abordagem emprega um sistema que utilize as duas técnicas implementadas de forma conjunta, unificando as principais características de cada técnica.

Existem vários trabalhos que mostram em seus resultados que a filtragem híbrida tem um desempenho melhor do que as outras duas técnicas abordadas nesse capítulo de forma individual. Como os trabalhos de Balabanović and Shoham (1997) e Billsus and Pazzani (1999).

3.4 Métricas de Avaliação

As métricas são para medir a relevância ou a precisão de um método de avaliação, a frequência que um sistema de recomendação toma decisões corretas ou incorretas sobre determinado item recomendado. Até hoje, é um desafio selecionar qual a métrica ideal para um sistema, pois ainda não há uma padronização. Existem várias métricas na área de recomendação, como por exemplo: Precision, Recall, Mean Average Precision (MAP) e Mean Absolute Error (MAE).

3.4.1 Precision e Recall

As métricas mais populares são Recall e Precision na área de recuperação da informação, que foram propostas no trabalho Cleverdon *et al.* (1966) como métricas chaves. Para essas métricas, o sistema deve ter uma forma de avaliação binária, ou seja, o usuário apenas pode avaliar um item com gostei ou não gostei. Caso o sistema não seja dessa forma (como o sistema desse trabalho o LibRec, onde a avaliação é dada na escala de 1-5), então será realizada uma conversão para esse modelo, onde como já vimos no Capítulo 2.2, na escala de 4-5 será assumido como gostei e de 1-3 como não gostei. A Precision representa a fração de itens recuperados que são relevantes, calculada com a equação a seguir.

$$P = \frac{|RI \cap IR|}{|IR|} \tag{3.4}$$

onde P é a precisão, RI representa os itens relevantes e IR os itens recuperados. Já a métrica Recall representa a fração de itens relevantes que são recuperados.

$$R = \frac{|RI \cap IR|}{|RI|} \tag{3.5}$$

3.4.2 Mean Average Precision (MAP)

Mean Average Precision (Manning *et al.*, 2008), entre as métricas de avaliação ela tem boa descrição e estabilidade nos resultados. Para uma necessidade de informação, a MAP é calculada com média do valor de precisão obtido para o conjunto de k documentos superiores existentes após cada documento relevante ser recuperado, como mostra sua equação a seguir.

$$MAP(Q) = |Q|^{-1} + \sum_{j=1}^{|Q|} m_j^{-1} \sum_{k=1}^{m_j} Precision(R_{jk}) \tag{3.6}$$

Dado o conjunto $\{d_1,..., d_{mj}\}$ de documentos relevantes para uma necessidade de informação $q_j \in Q$, R_{jk} é o conjunto de resultados classificados como o melhor resultado até o documento d_k.

3.5 Sumário

Este capítulo apresenta os tipos de Sistemas de Recomendação, o sistema de filtragem colaborativa baseada em usuário, filtragem por conteúdo e híbrida, onde foi mostrado quais as principais técnicas, vantagens e desvantagens de cada um apresentado. Também vimos brevemente algumas métricas de avaliação usadas nesses sistemas. No capítulo 4 será mostrado a proposta desse trabalho para minimizar os problemas abordados.

4

Uma Abordagem Híbrida para Sistemas de Recomendação Baseados em Filtragem Colaborativa.

Este capítulo apresenta o modelo proposto: o Sistema de Recomendação híbrido proposto, Algoritmo Filtragem Híbrida (HCF), cujo os objetivos são minimizar o problema de *Cold Start* encontrado nos algoritmos de filtragem colaborativa baseado em usuário, recomendar itens diretamente ligados ao histórico disponível sobre o usuário e assim gerar predições melhores, ou seja, com uma taxa de erro em relação as predições feitas pelos algoritmos de Filtragem Colaborativa (FC) utilizados.

4.1 Requisitos

Nessa Seção serão mostrados todos os métodos e técnicas necessárias para a implementação do algoritmo proposto nesse trabalho.

No algoritmo é utilizada a técnica de um sistema de recomendação híbrida onde foi combinada uma estratégia colaborativa por usuário e outra de filtragem por conteúdo. A ideia consiste em utilizar um algoritmo colaborativo disponível da biblioteca do Librec que veremos na Seção 4.4.2, executá-lo e aplicar o seu resultado de predição em um novo cálculo que será gerado pelo algoritmo proposto. O algoritmo utilizado da biblioteca Librec sempre será executado de forma colaborativa com base no usuário, onde será aplicada a técnica de Similaridade do Cosseno 3.2 foi visto no capítulo 3, Seção 3.1.1.2, que realiza o cálculo de similaridade entre os usuários. Os resultados obtido de cada predição de classificação dos filmes são salvos para serem utilizados no cálculo do HCF.

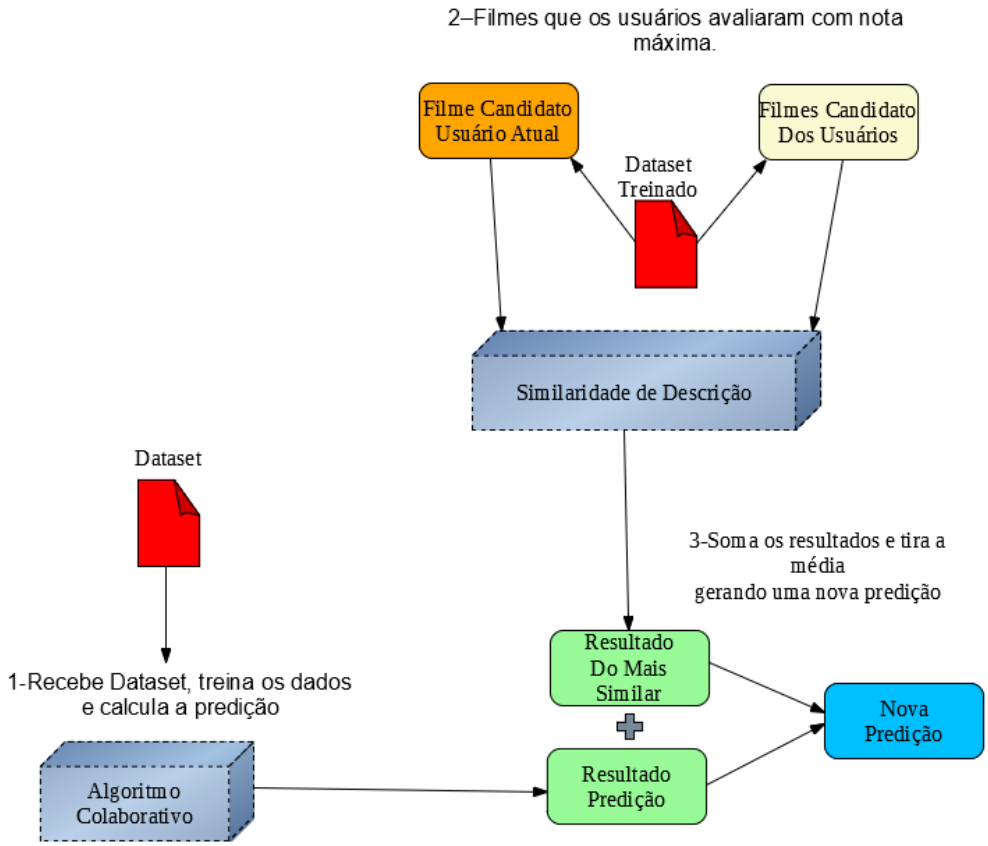

Figura 4.1 Diagrama de funcionamento do algoritmo

4.2 Modelo de Dados

Nesta Seção será descrito detalhadamente como são modelados os usuários e itens a serem recomendados.

4.2.1 Modelo de Usuário

Para construir o modelo de usuário é preciso analisar todas as avaliações feitas por ele aos filmes. É dessa análise que serão obtidos os filmes candidatos de cada usuário, esses filmes são os que foram avaliados com nota 5, ou seja, são os filmes escolhidos neste trabalho que representam a preferência do usuário.

Após o reconhecimento de todos os filmes candidatos do usuário, será obtida a descrição desses filmes e com elas serão feitas comparações com a descrição de outros filmes candidatos. Nesse contexto é esperado uma melhor definição do perfil desse

usuário mesmo não se tratando do mesmo filme em questão, pois será a similaridade entre as suas descrições que determinará se o filme também é de interesse do usuário usando informações do conteúdo do filme. Um exemplo na Tabela 4.1 abaixo.

	Matrix	Titanic	Matrix Reload
Daniel	5	3	?
Logan	3	1	2
Alan	2	1	5

Tabela 4.1 Exemplo de uma matriz Usuário X Filme, onde a primeira coluna corresponde aos usuários e a primeira linha aos filmes e a interseção entre eles correspondem as avaliações do usuário ao filme.

No exemplo descrito na Tabela 4.1, Matrix é o filme candidato de Daniel e Matrix Reload é o filme candidato de Alan. Além de levar em conta a similaridade entre Daniel e Alan, também será levado em consideração a descrição dos filmes candidatos entre eles.

4.2.2 Modelo dos Itens de Recomendação

Os itens usados nesse sistema são filmes. Eles foram avaliados por usuários que os classificaram com notas de 1 a 5, sendo 1 a nota miníma e considerado como "não gostei"pelo usuário e a nota máxima 5 considerada como "gostei".

A informação dos filmes utilizados nessa proposta foi a sua descrição. A descrição nada mais é que uma introdução explicativa do conteúdo do filme no geral. Essas descrições foram obtidas através do site IMDb[1], também conhecido como Internet Movie Database (Em Português: Base de Dados de Filmes na Internet) é uma base de dados online de informação sobre música, cinema, filmes, programas e comerciais para televisão e jogos de computador.

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

Figura 4.2 Imagem de um filme tirada do IMDb, onde se pode ver a storyline (descrição) obtida.

4.3 Modelo de Recomendação Híbrido

O algoritmo proposto (HCF) é uma extensão para algoritmos de filtragem colaborativa, o método consiste em pegar o resultado de predição de algum algoritmo obtido do Librec e somá-lo com a similaridade obtida pela Fórmula 4.6, dessa soma calcula-se a média e assim gerando a nova predição. Segue a equação:

$$NP = n^{-1}\left[\sum_{i=1}^{n}(RN_i + RVP_i)\right] \tag{4.1}$$

Onde NP corresponde à nova predição, n representa o número de algoritmos usados e RVP corresponde à predição feita pelo algoritmo escolhido e RN que veremos na Subseção 4.3.2 na Equação 4.7 a seguir, corresponde ao resultado da similaridade normalizado.

4.3.1 Modelo de Predição

Os algoritmos de filtragem colaborativa utilizados nesse trabalho foram: AspectModelRecommender, BPoissMFRecommender e LDARecommender. A seguir, serão descritas suas principais características.

4.3.1.1 Latent Dirichlet Allocation (LDA)

LDA é um modelo probabilístico generativo de uma coleção de dados discretos, como um conjunto de textos. A ideia básica é que os documentos são representados como misturas aleatórias sobre tópicos, onde cada tópico é caracterizado por uma distribuição sobre palavras (Blei *et al.*, 2003).

Os dados consistem em palavras $W = \{w_1, ..., w_n\}$, onde cada w_i pertence a algum documento d_i, como em uma matriz de palavra-documento. Para cada documento têm-se uma distribuição multinomial sobre os tópicos T, com os parâmetros $\theta^{(di)}$, então para uma palavra no documento d_i, P(z_i = j) = $\theta^{(di)}_j$. O tópico j é representado por uma distribuição multinomial sobre as palavras w no vocabulário, com parâmetros $\phi^{(j)}$, então P $(w_i|z_i = j) = \phi^{(j)}_{w_i}$. Para fazer previsões sobre novos documentos, precisamos assumir uma distribuição prévia nos parâmetros $\theta^{(di)}$. A distribuição de Dirichlet é conjugada ao multinomial, então pegamos um Dirichlet antes em $\theta^{(di)}$. A distribuição de modelo sobre palavra em qualquer documento é descrita assim:

$$P(w_i) = \sum_{j=1} P(w_i|z_i = j)P(z_i = j) \tag{4.2}$$

Este algoritmo pode ser utilizado para fazer recomendações como o desse trabalho, no modelo Usuário-item. A maneira de utilizá-lo dessa forma é tratar os itens como documentos e classificações como palavras (Porteous *et al.*, 2008).

4.3.1.2 Bayesian Poisson Factorization (BPoissMF)

A modelagem dos dados que esse algoritmo utiliza é feita com a com distribuições fatorada de Poisson (Canny, 2004). Onde cada item i é representado por um vetor de K atributos latentes β_i e cada usuário u por um vetor de K preferências latentes θ_u. As avaliações y_{ui} são modeladas com uma distribuição de Poisson, parametrizada pelo produto das preferências do usuário e atributos do item, $y_{ui} \sim$ Poisson($\theta\,\tau_u\beta_i$).

Além dos dados básicos que geram a distribuição, é utilizada Gamma (Canny, 2004), que prioriza os atributos latentes e preferências latentes, que encorajam o modelo para

representações esparsas dos usuários e itens. Além disso, há prévias adicionais no parâmetro de taxa específica do usuário e do item desses Gammas, que controla o tamanho médio da sua representação. Essa estrutura hierárquica permite obter a diversidade dos usuários, pois alguns adquirem mais itens do que outros, assim também com os itens, onde alguns são mais populares que outros.

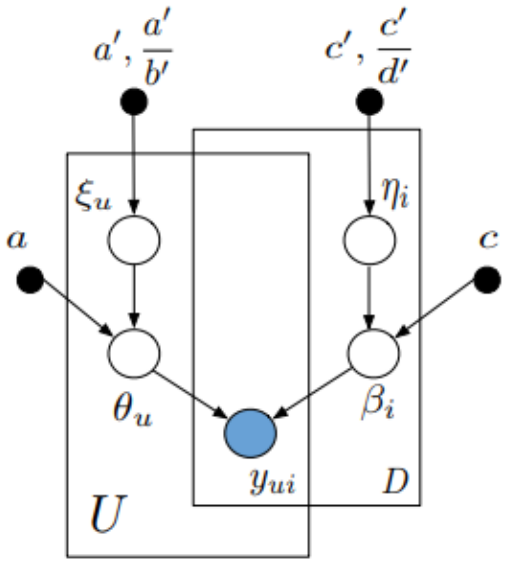

Figura 4.3 Modelo hierárquico fatoração de Poisson. Imagem de (Canny, 2004).

Utilizando o modelo da Figura 4.3 com a utilização da prioridade Gamma, temos os passos a seguir:

1. Para cada usuário u:
   - Amostra de atividade $\xi_u \sim \text{Gamma}(a', a' / b')$.
   - Para cada componente K, amostra de preferência $\theta_{uk} \sim \text{Gamma}(a, \xi_u)$.
2. Para cada item i:
   - Amostra de popularidade $\eta_i \sim \text{Gamma}(c', c' / d')$.
   - Para cada componente K, amostra de preferência $\beta_{ik} \sim \text{Gamma}(c, \eta_i)$.
3. Para cada usuário e item, a avaliação $y_{ui} \sim \text{Poisson}(\theta \tau_u \beta_i)$.

Todo esse processo foi denominado Bayesian Poisson Factorization (Canny, 2004).

4.3.1.3 Aspect Model

Neste modelo, uma classe latente variável $z \in Z = \{z_1, \ldots, z_k\}$ está associado a cada avaliação (x, y) (Hofmann and Puzicha, 1999). A principal suposição feita é que x e y são independentes, condicionados em z. O modelo de probabilidade pode assim ser simplesmente escrito como a Equação 4.3 abaixo:

$$P(x,y) = \sum_{z \in Z} P(z)P(x|z)P(y|z) \tag{4.3}$$

onde P(x | z) e P(y | z) são classe de distribuições multinomial e Pz são as probabilidades anteriores da classe. Observe que o modelo é perfeitamente simétrico com respeito às entidades x e y. No entanto, pode-se também re-parametrizar o modelo de maneira assimétrica, usando a identidade P(z)P(x | z) = P(x , z) = P(x)P(z | x) que origina a Equação 4.3:

$$P(x,y) = P(x)P(y|x)onde \tag{4.4}$$

$$P(y|x) = \sum_{z \in Z} P(z|x)P(y|z). \tag{4.5}$$

4.3.2 Modelo de Similaridade Sintática

O novo algoritmo usa a similaridade porém de uma forma diferente do citado anteriormente, em vez de comparar os usuários entre si, ele compara a descrição dos filmes candidatos de um usuário (ou seja, filmes que o usuário considera como bom (rating = 5), com todos os filmes candidatos dos outros usuários do banco (essa comparação não leva em conta filmes iguais) e salva apenas o resultado da maior similaridade entre eles. Esse resultado é obtido em porcentagem, então ele é normalizado para rating (de 1 a 5). Segue a equação:

$$P(w_i) = \sum(\vec{i}, \vec{j}) = \frac{U_{\vec{i}}.U_{\vec{j}}}{|U_{\vec{i}}|X|U_{\vec{j}}|} \tag{4.6}$$

Onde $U_{\vec{i}}$ e $U_{\vec{j}}$ representam respectivamente o vetor com as descrições dos filmes candidatos do usuário atual e o vetor de descrição dos filmes candidatos de todos os outros usuários do banco. O resultado desse cálculo é obtido em porcentagem (onde 0.0 corresponde a 0% e 1.0 a 100%), então é necessário fazer uma normalização para rating com o seguinte equação:

$$RN = simi(\vec{i}, \vec{j}).y \tag{4.7}$$

Onde *RN* corresponde ao resultado normalizado. A normalização se dá pelo produto da similaridade entre a constante 4 pois os ratings dados pelos usuários são entre 1 e 5, e então soma-se a constante 1 para que o rating seja coerente e válido entre 1 a 5.

	Lagoa Azul	X-men	Matrix
Pedro	?	5	5
Danielle	3	1	2
Renan	5	3	?

Tabela 4.2 Exemplo de uma matriz Usuário X Filme, onde a primeira coluna corresponde aos usuários e a primeira linha aos filmes.

Um exemplo de como esse código de similaridade funciona podemos ver analisando a Tabela 4.2 acima onde temos em destaque vermelho o usuário Renan e três notas 5 para alguns filmes. Renan avaliou o filme Lagoa Azul com nota 5, então esse filme é um filme candidato de Renan, assim como os filmes X-men e Matrix são filmes candidatos para Pedro. O algoritmo de similaridade proposto pegará o filme Lagoa Azul e comparará a sua descrição com os filmes X-men e Matrix já que eles são filmes candidatos de outro usuário, ou seja, o objetivo dessa abordagem é, de forma colaborativa, comparar os filmes que o usuário mais gostou com os dos demais usuários que também avaliaram com nota máxima, utilizando nessa comparação o metadado de descrição do filme que caracteriza uma abordagem de filtragem por conteúdo.

4.4 Tecnologias

Para o desenvolvimento do algoritmo foram utilizadas algumas tecnologias: linguagem, biblioteca, dataset, entre outros. A seguir serão apresentadas tais tecnologias.

4.4.1 Java

Java[2] é uma linguagem de programação orientada a objeto (baseada em classes) e plataforma computacional lançada pela primeira vez pela Sun Microsystems em 1995.

[2] https://www.java.com/pt_BR/

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). Como veremos nesse capítulo na Seção 4.4.2, a biblioteca Librec usada nesse trabalho utiliza Java e todo o código proposto foi desenvolvido nessa linguagem com a utilização da IDE (Ambiente de Desenvolvimento Integrado) Eclipse[3].

4.4.2 Librec

O LibRec[4] é uma biblioteca Java open source, licenciada pela GPL, com cerca de 70 algoritmos de recomendação múltipla com o objetivo de resolver duas tarefas clássicas em sistemas recomendadores, a previsão de classificação e classificação de itens, implementando um conjunto de algoritmos de recomendação de última geração de Sistemas de Recomendação[5].

Desses 70 algoritmos podemos identificar que existem três tipos deles que são: (1) linhas de base que fazem pouco uso de informações personalizadas; (2) algoritmos de núcleo que são abordagens de última geração com base em interações com elementos do usuário e informações contextuais; e (3) outros algoritmos[6]. Segundo Guo *et al.* (2015) o LibRec tem performance melhor do que outras bibliotecas, como o Personalized Recommendation Algorithms (PREA) e MyMediaLite[7], ao mesmo tempo que obtém desempenho avaliativo e competitivo.

[3] https://www.eclipse.org
[4] https://www.librec.net/index.html
[5] https://www.librec.net/dokuwiki/doku.php
[6] https://www.librec.net/dokuwiki/doku.php?id=AlgorithmList
[7] http://www.mymedialite.net

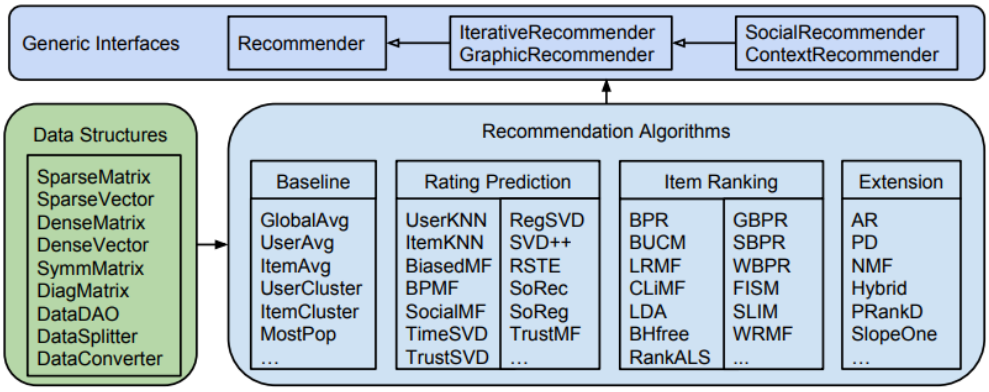

Figura 4.4 LibRec consiste em três componentes principais: interfaces genéricas, estruturas de dados e algoritmos de recomendação. Imagem de Guo *et al.* (2015).

O LibRec possui um bom encapsulamento e pode carregar configurações diretamente através da linha de comando para executar o código. Os usuários também podem adotar LibRec em outros projetos, fazendo instâncias de classes Java correspondentes. Por essa facilidade de manipulação dos algoritmos encontrados na biblioteca que se torna viável a realização desse trabalho.

4.5 Sumário

Este capítulo apresenta o algoritmo e a proposta do trabalho. Foram apresentados os modelos de dados de usuário, modelos dos itens, modelo de predição dos algoritmos usados na avaliação, o modelo híbrido proposto e o modelo de similaridade. Também foram apresentadas as tecnologias e ferramentas utilizadas. No capítulo 5 será realizada uma avaliação do trabalho realizado, discutida a metodologia, base de dados, métricas de avaliação, os resultados obtidos e uma discussão.

5 Avaliação

Neste capítulo será mostrado o processo de avaliação utilizado para verificar se os objetivos previstos foram alcançados. O esperado é que com os algoritmos de CF e a estratégia de comparação do metadado descrição entre os filmes candidatos dos usuários, que torna o algoritmo híbrido, haja uma melhora na predição dos usuários em relação aos filmes, ou seja, a média de erro seja menor. Também será descrito toda metodologia utilizada nesse trabalho, as métricas utilizadas na avaliação, o conjunto de dados de todo experimento, os algoritmos de CF testados e todos os resultados obtidos.

5.1 Metodologia

Os testes realizados têm por objetivo mostrar que com uma maior quantidade de informações e combinando abordagens colaborativas com abordagens por conteúdo é possível melhorar a predição de usuários a itens e assim consequentemente melhorar recomendações sugeridas ao mesmo.

Para a avaliação foram utilizado três algoritmos de CF, o AspectModelRecommender, BPoissMFRecommender e LDARecommender. Eles foram escolhidos pois a proposta teve um bom desempenho em algoritmos colaborativos baseados em estatística. Foram comparados os desempenhos de cada algoritmo com e sem a proposta deste trabalho. A base de dados usada foi dividida em alguns casos de teste, eles foram feitos com todos os usuários do banco e outros com redução dos mesmos a fim de observar se a estratégia proposta tem alguma diferença de resultado relevante para as situações dadas. Todos os testes foram realizados com os dados do *dataset* MovieLens 100K[1].

Foi aplicada uma técnica chamada de Cross-Validation (Validação Cruzada) que

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

avaliar a capacidade de generalização de um modelo, a partir de um conjunto de dados (Kohavi *et al.*, 1995). O método de validação cruzada utilizada neste projeto foi o Holdout pois essa abordagem é indicada quando há uma grande quantidade de dados. Ela consiste em dividir o conjunto total de dados em dois subconjuntos, um para treino e outro para teste (Kohavi *et al.*, 1995). A proporção para este projeto foi de 80% para o conjunto de treino e 20% para o conjunto de testes para criar o conjunto de validação. Foi utilizada esses valores pois essa técnica tem um melhor desempenho com valores próximos de 2/3 de treino.

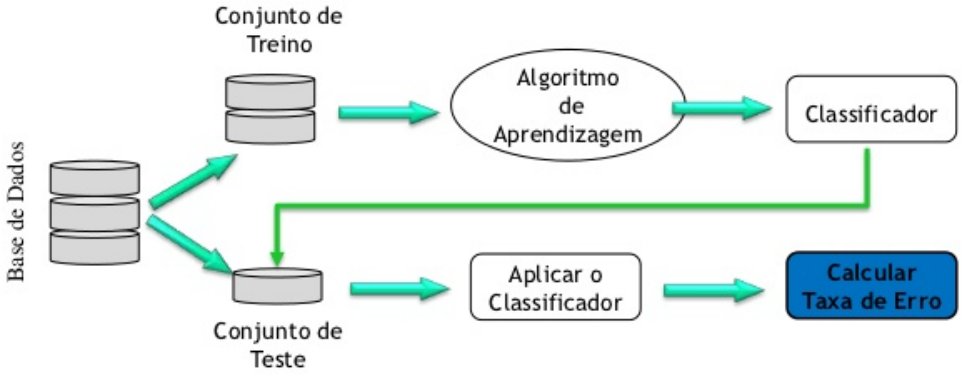

Figura 5.1 Exemplo de como é executada a técnica de Validação Cruzada HoldOut. Imagem de Gladys Castillo[3]

.

Para calcular a diferença entre as avaliações reais e as previstas, ou seja, a média de erros do conjunto de previsões foram utilizadas as métricas Root Mean Square Error (RMSE) 5.3.2 e a Mean Absolute Error (MAE) 5.3.1.

5.2 Conjunto de Dados

Todos os testes foram executados com dados do MovieLens que é um sistema de recomendação em Web e uma comunidade virtual que recomenda filmes para seus usuários assistirem, com base em suas preferências de filme, usando a filtragem colaborativa das avaliações de filmes e críticas de filmes dos membros. Ele contém cerca de 11 milhões de avaliações para cerca de 8500 filmes [4]. Um grupo de pesquisa do Departamento

[3]https://pt.slideshare.net/gladysCJ/evaluation-and-comparison-of-supervised-learning-algorithms
[4]http://license.umn.edu/technologies/z05173$_{m}ovielens-database$

de Ciência da Computação e Engenharia na Universidade de Minnesota chamado de GroupLens[5], realiza coleta de dados do MovieLens para estudo.

O conjunto de dados utilizado nesse trabalho é o MovieLens 100K Dataset[6] que é composto por 942 usuários, 1682 filmes e 100.000 avaliações (1-5), como mostra a Tabela 5.1 abaixo.

Conjunto de Dados	
Usuários	942
Filmes	1682
Avaliações	100.000

Tabela 5.1 Conjunto de dados do MovieLens 100K Dataset.

Cada usuário classificou pelo menos 20 filmes e informações demográficas simples foram coletadas dos usuários, como idade, sexo e ocupação. Os dados foram coletados durante o período de sete meses a partir de 19 de setembro de 1997 até 22 de abril de 1998. Esses dados foram salvos de forma íntegra, ou seja, usuários que tinham menos de 20 classificações ou não tinham passados informações demográficas completas foram removidas deste conjunto de dados (Herlocker *et al.*, 1999).

5.3 Métricas da Avaliação

As métricas utilizadas na avaliação foram Mean Absolute Error (MAE) e Root Mean Squared Error (RMSE). Essas métricas foram escolhidas pois o objetivo desse trabalho é melhorar as predições dos usuários aos filmes e elas calculam a magnitude de erros entre a avaliação real e a prevista.

5.3.1 Mean Absolute Error (MAE)

Está métrica mede a magnitude média dos erros em um conjunto de previsões, sem considerar sua direção. É a média sobre a amostra de teste das diferenças absolutas entre a previsão e a observação real, onde todas as diferenças individuais têm o mesmo peso. O resultado do MAE varia entre 0 a infinito e quanto menor for, melhor foi a predição gerada pelo algoritmo. Ela é calculada pela seguinte equação:

$$MAE = \left[n^{-1} \sum_{i=1}^{n} |y_i - \hat{y}_i| \right] \tag{5.1}$$

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

Onde y_i e $\hat{y}_i$ correspondem ao valor original e o valor da predição respectivamente. O cálculo do MAE é relativamente simples, se resume a somar as magnitudes (valores absolutos) de erros para obter o "erro total"e depois dividir o erro total por n, sempre assumindo que os pesos são sempre iguais (Willmott and Matsuura, 2005).

5.3.2 Root Mean Squared Error (RMSE)

Está métrica é parecida com a Mean Absolute Error (MAE) pois assim como ela, mede a magnitude média dos erros em um conjunto de previsões. A diferença entre elas é que a RMSE não atribui os mesmos pesos as diferenças individuais, ou seja, quanto maior a diferença da nota original para a nota prevista, maior será o RMSE Willmott and Matsuura (2005). Ela é calculada pela seguinte equação:

$$RMSE = \sqrt{n^{-1} \sum_{i=1}^{n} |y_i - \hat{y}_i|^2} \tag{5.2}$$

Onde assim como no MAE, y_i e $\hat{y}_i$ correspondem ao valor original e o valor da predição respectivamente. Assim como o MAE, resultado do RMSE varia entre 0 a infinito e quanto menor for, melhor foi a predição gerada pelo algoritmo, sendo que sempre será MAE <= RMSE, só acontecendo de MAE = RMSE quando a magnitude de todos os erros forem iguais (Willmott and Matsuura, 2005).

5.4 Resultados

Para avaliar os resultados foram utilizadas as seguintes casos de testes:

- Utilizando as classificações de 30 usuários do banco.
- Utilizando as classificações de 100 usuários do banco.
- Utilizando as classificações de todos os usuários (942) do banco.

Os resultados obtidos no teste com 30 usuários estão na Tabela 5.2, onde esses 30 correspondem a 3943 avaliações, sendo 3142 (80%) de treino e 799 (20%) de teste. Os resultados dos testes com 100 e todos os usuários correspondem as Tabelas 5.3 e 5.4 respectivamente. Onde 100 usuários correspondem a 11.019 de avaliações, sendo 8836 (80%) de treino e 2183 (20%) e todos os usuários correspondem a 100.000 avaliações, sendo 80.000 (80%) para treino e 20.000 (20%) para teste.

Algoritmos	Original MAE	Proposta MAE	Original RMSE	Proposta RMSE
BPoissMF	1.480	1.070	1.925	1.260
AspectModel	1.348	1.409	1.760	1.686
LDARecommender	2.559	2.364	2.848	2.631

Tabela 5.2 Resultados do RMSE e MAE dos três algoritmos no caso de teste com 30 usuários.

Algoritmos	Original MAE	Proposta MAE	Original RMSE	Proposta RMSE
BPoissMF	1.389	1.024	1.796	1.213
AspectModel	2.579	2.421	2.829	2.661
LDARecommender	2.614	2.447	2.839	2.667

Tabela 5.3 Resultados do RMSE e MAE dos três algoritmos no caso de teste com 100 usuários.

Algoritmos	Original MAE	Proposta MAE	Original RMSE	Proposta RMSE
BPoissMF	1.467	0.975	1.853	1.160
AspectModel	2.536	2.305	2.774	2.536
LDARecommender	2.525	2.220	2.767	2.452

Tabela 5.4 Resultados do RMSE e MAE dos três algoritmos no caso de teste com todos os usuários (942).

5.4.1 Análise Geral

No geral, analisando todos os casos de teste, conclui-se que essa estratégia diminui a taxa de erro da predição nos algoritmos testados. Pode-se notar que o algoritmo que se destacou nessa abordagem com a melhora mais considerável foi o BPoissMF que no seu pior resultado entre os casos de teste, teve um MAE aproximadamente 26% melhor e no seu melhor caso com 34% melhor que o algoritmo original. Já na métrica o resultado foi melhor com 32% no pior caso e 37% no melhor, o que mostra que houve uma redução de casos em que a diferença do rating predito com o rating real são grandes.

No algoritmo LDARecommender houve um desempenho melhor entre 8% a 10% na métrica RMSE em relação ao algoritmo original para todos os casos. Nesse caso, nota-se que os o ganho de desempenho em proporção do MAE não foi distante do RMSE, onde ele variou de 8% a 12%.

O algoritmo AspectModelRecommender os desempenhos nos casos de 100 e 942 usuários, as métricas tiveram um percentual relativamente iguais. O RMSE e MAE no

primeiro caso tiveram precisamente 5,93% e 6,11% respectivamente, e no segundo caso obteve 8,61% e 9,08% respectivamente. Já no caso de teste com 30 usuários pode-se visualizar que a abordagem não obteve um resultado satisfatório sendo 4% menos eficiente do que o algoritmo original. Devido a forma como testset é escolhido(aleatoriamente) combinado com a baixa quantidade de usuários (o que resulta em um número menor de avaliações e filmes analisados).

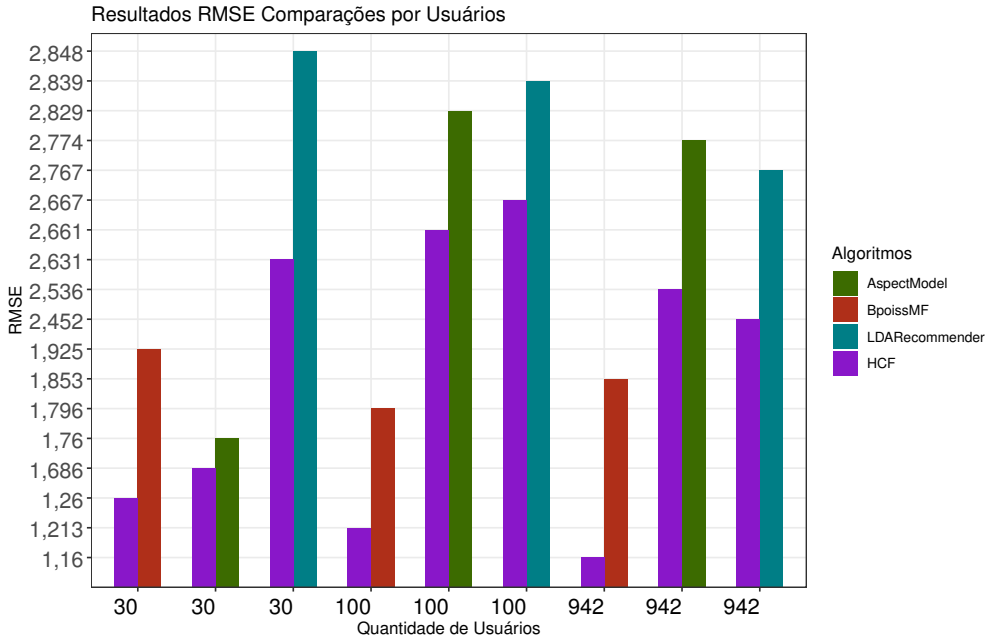

Figura 5.2 Cada par de barras (roxo e cor do algoritmo) apresentado no gráfico, representam o resultado do RMSE, para o HCF e para o algoritmo testado respectivamente, correspondentes ao número de usuários utilizados em cada caso de teste.

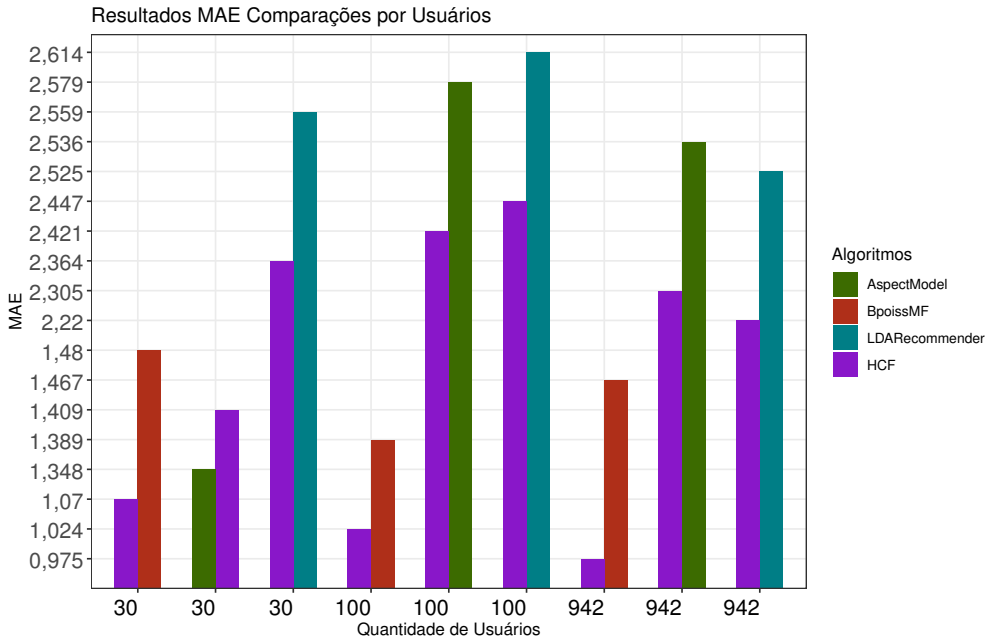

Figura 5.3 Cada par de barras (roxo e cor do algoritmo) apresentado no gráfico, representam o resultado do MAE, para o HCF e para o algoritmo testado respectivamente, correspondentes ao número de usuários utilizados em cada caso de teste.

5.5 Discussão

Neste trabalho foi desenvolvido um algoritmo híbrido com o objetivo de melhorar a recomendação de alguns algoritmos colaborativos. Esse algoritmo permite que a predição da avaliação dos usuários sejam mais precisas em relação a avaliação original, utilizando informação de um metadado do item de forma colaborativa.

Após analisar os resultados obtidos com as métricas utilizadas, conclui-se que é possível melhorar as recomendações utilizando em conjunto, a similaridade entre os usuários e entre os itens de sua preferência. Com base nos atributos dos itens de seu interesse, é possível prever uma avaliação mais próxima da real utilizando a semelhança das características dos itens com o dos demais usuários.

5.6 Ponto de Melhoria

O principal problema na criação do modelo do item foi é na identificação do filme. O dataset usado (movielens 100k) disponibiliza apenas o id (identificação) do filme, os nomes dos filmes vem em um dataset separado.

Outro problema é na obtenção das descrições dos filmes. Alguns filmes do dataset são muito antigos e apesar do site IMBd[7] ter uma grande quantidade de filmes, alguns deles presentes no Movielens não tinham disponíveis no IMBd. Algumas descrições tinham informações pobres, ou seja, descrições simplórias ou com informações que não descreve o filme em si, dificultado na similaridade.

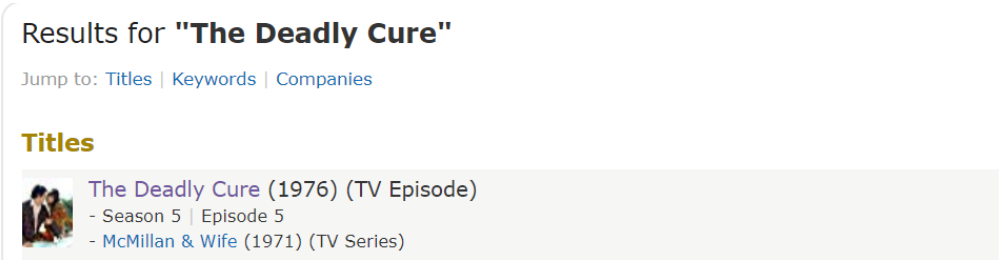

Figura 5.4 Exemplo de um filme do Movielens que não tem no IMDB. A busca feita apenas retorna uma série de TV.

5.7 Trabalhos Relacionados

Esta seção discute alguns trabalhos recentes, relacionados a este trabalho que aborda filtragem híbrida, predição de valores e similaridade entre usuários e entre conteúdos.

O trabalho de Li *et al.* (2017) utiliza uma abordagem híbrida com o objetivo de prever a influência social dos usuários em redes sociais baseadas em eventos. Os problemas abordado por eles são devido ao fato de que a matriz de influência social construída é muito esparsa e os valores de sobreposição na matriz são poucos, isto torna desafiador encontrar vizinhos similares confiáveis utilizando a técnica de Similaridade do Cosseno 3.1.1.2. Para enfrentar esse desafio, eles propõem um método adicional de descoberta de vizinhança baseada em informações, considerando os recursos específicos do evento e específicos do usuário.

No trabalho de Chu and Tsai (2017) é utilizado uma abordagem híbrida para prever as preferências dos usuários a restaurantes. Além de usar a similaridade entre os usuários e atributos referentes ao restaurante, eles levam em consideração que muitos clientes reveem restaurantes em artigos de blog, nos quais os comentários e várias fotos podem estar disponíveis. Neste artigo, é investigado especialmente a influência da informação visual, ou seja, fotos tiradas por clientes e postadas em blogs, para previsão de restaurantes para qualquer usuário.

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

5.8 Sumário

Este capítulo apresenta os principais resultados obtidos de um algoritmo híbrido que visa melhorar alguns tipos de algoritmos de CF. Foram apresentadas as metodologias de avaliação, a base de dados a as métricas utilizadas. O estudo apresentado neste capítulo mostrou resultados positivos os três algoritmos utilizados nos testes.

6
Conclusão

Neste trabalho foi apresentado um sistema de recomendação híbrido. Inicialmente foi apresentado a motivação para a criação do sistema de recomendação, relatou-se problemas que alguns algoritmos de filtragem colaborativos apresentam. Foi proposta uma solução para ajudar os usuários a receberem recomendações de sua preferência mesmo quando não se adquiriu muita informação do mesmo.

No Capítulo 2 descreve modelagens de perfil de usuário, apresentando algumas técnicas e como é modelado, os meios como os dados são obtidos e as dificuldades encontradas na construção do perfil.

No Capítulo 3 apresentou-se sobre os Sistemas de Recomendação , mostrando os principais conceitos e técnicas, abordando a vantagens e as limitações de cada técnica. Também mostrou técnicas de similaridades e algumas métricas de precisão.

O Capítulo 4 explica a proposta híbrida de solução para melhoria de algoritmos colaborativos que geram predições de usuários a filmes. Também foi descrito o funcionamento do sistema, os modelos do usuário e do item, as equações dos algoritmos utilizados e do algoritmo proposto. Além de expor todas as tecnologias usadas no processo de desenvolvimento.

Para finalizar, foi apresentado em detalhes a metodologia e realizou-se uma avaliação experimental. Os métodos de avaliação e os resultados foram apresentados, obtendo um resultado satisfatório para proposta, mostrando que a extensão dos algoritmos testados conseguiu melhorar as predições feitas pelos mesmos.

6.1 Contribuições do Trabalho

As principais contribuições deste trabalho são explicados a seguir:

- Uma revisão comparativa entre sistemas baseados em conteúdo e filtragem colaborativa. SRs;
- **A extensão de algoritmo:** Nesse projeto, proporcionou um estudo dos algoritmos disponíveis na biblioteca Librec, e mostrou a possibilidade de criação de métodos para melhoria de algoritmos colaborativos presentes nesta biblioteca.
- Avaliação da performance do algoritmo proposto com referências em métricas e trabalho correlacionados.

6.2 Trabalhos Futuros

Mesmo com os resultados satisfatórios, é possível melhorar as predições e consequentemente as recomendações das seguintes formas:

- **Inserir mais metadados nos filmes:** Adicionando mais metadados no cálculo de similaridade como por exemplo o gênero do filme. Será possível aumentar a similaridade entre os filmes combinando sua descrição e gênero.
- **Mudança na técnica de Validação Cruzada:** Pode-se aplicar uma técnica diferente da usada neste trabalho. Por exemplo, a técnica K-Fold (Kohavi *et al.*, 1995) e analisar o comportamento aplicando-a nessa proposta.
- **Utilizar outros algoritmos:** A técnica utilizada neste trabalho pode ser aplicada para outros algoritmos de recomendação como extensão dos mesmos.
- **Alterar modelagem do usuário:** Pode-se aplicar algum método diferente a essa modelagem o que pode mudar o jeito de calcular a predição proposta.
- **Utilizar outras métricas de avaliação:** Existem varias métricas que podem avaliar o desempenho da estratégia aplicada nesse trabalho com relação ao proposito delas. Algumas delas foram inseridas no Capítulo **??**, na Seção 3.4.

6.3 Sumário

Este capítulo apresentou um resumo de tudo que foi feito e discutido neste trabalho. Mostrou-se as principais contribuições da nossa proposta de sistema de recomendação e pontos de melhorias da nossa proposta híbrida para sistemas de recomendação colaborativos a serem feitas em 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.

Aggarwal, C. C. (2016). *Recommender systems*. Springer.

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

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

Billsus, D. and Pazzani, M. J. (1999). A hybrid user model for news story classification. In *UM99 User Modeling*, pages 99–108. Springer.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, **3**(Jan), 993–1022.

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

Bradley, P. S., Fayyad, U. M., Reina, C., *et al.* (1998). Scaling clustering algorithms to large databases. In *KDD*, volume 98, pages 9–15.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, **12**(4), 331–370.

Burke, R. (2007). The adaptive web. chapter Hybrid web recommender systems, pages 377–408. Springer-Verlag, Berlin, Heidelberg.

Canny, J. (2004). Gap: a factor model for discrete data. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 122–129. ACM.

Carmagnola, F., Cena, F., and Gena, C. (2011). User model interoperability: a survey. *User Modeling and User-Adapted Interaction*, pages 1–47.

Chu, W.-T. and Tsai, Y.-L. (2017). A hybrid recommendation system considering visual information for predicting favorite restaurants. *World Wide Web*, **20**(6), 1313–1331.

Cleverdon, C. W., Mills, J., and Keen, M. (1966). Factors determining the performance of indexing systems.

Degemmis, M., Lops, P., and Semeraro, G. (2007). A content-collaborative recommender that exploits wordnet-based user profiles for neighborhood formation. *User Modeling and User-Adapted Interaction*, **17**(3), 217–255.

Fausto J F B Gominho, V. C. M. d. L. (2014). Sistema de recomendação. http://slideplayer.com.br/slide/1594308/.

Gedikli, F. (2013). *Recommender systems and the social web: Leveraging tagging data for recommender systems*. Springer Science & Business Media.

Gehrke, M. I. E. (2002). Rotinas digitais de comunicação pessoal: Internet e sociabilidade contemporânea. page 144.

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.

Guo, G., Zhang, J., Sun, Z., and Yorke-Smith, N. (2015). Librec: A java library for recommender systems. In *UMAP Workshops*, volume 4.

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

Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237. ACM.

Hofmann, T. and Puzicha, J. (1999). Latent class models for collaborative filtering. In *IJCAI*, volume 99.

Hu, Y., Koren, Y., and Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 263–272. IEEE.

Huang, Z., Chen, H., and Zeng, D. (2004). Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, **22**(1), 116–142.

Kohavi, R. *et al.* (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada.

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

Li, X., Cheng, X., Su, S., Li, S., and Yang, J. (2017). A hybrid collaborative filtering model for social influence prediction in event-based social networks. *Neurocomputing*, **230**, 197–209.

Linden, G., Smith, B., and York, J. (2003). Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, (1), 76–80.

Mahmood, T. and Ricci, F. (2009). Improving recommender systems with adaptive conversational strategies. In *Proceedings of the 20th ACM conference on Hypertext and hypermedia*, HT '09, pages 73–82, New York, NY, USA. ACM.

Manning, C. D., Raghavan, P., Schütze, H., *et al.* (2008). *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge.

McCrae, J., Piatek, A., and Langley, A. (2004). Collaborative filtering. *http:// www. imperialviolet. org*.

Melville, P., Mooney, R. J., and Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. In *Aaai/iaai*, pages 187–192.

Miyahara, K. and Pazzani, M. J. (2002). Improvement of collaborative filtering with the simple bayesian classifier. *Information Processing Society of Japan*, **43**(11).

Picault, J., RibiËre, M., Bonnefoy, D., and Mercer, K. (2011). How to get the recommender out of the lab? In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 333–365. Springer US.

Porteous, I., Bart, E., and Welling, M. (2008). Multi-hdp: A non parametric bayesian model for tensor factorization. In *Aaai*, volume 8, pages 1487–1490.

Rashid, A. M., Karypis, G., and Riedl, J. (2008). Learning preferences of new users in recommender systems: an information theoretic approach. *ACM SIGKDD Explorations Newsletter*, **10**(2), 90–100.

Reategui, E. B. and Cazella, S. C. (2005). Sistemas de recomendaçao. In *XXV Congresso da Sociedade Brasileira de Computação*, pages 306–348.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186. ACM.

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

Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors (2011b). *Recommender Systems Handbook*. Springer.

Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. (2007). Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer.

Silva, R. d. S. (2016). Recomendações de lojas para clientes de um shopping center.

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

Willmott, C. J. and Matsuura, K. (2005). Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, **30**(1), 79–82.

Yao, G. and Cai, L. (2017). User-based and item-based collaborative filtering recommendation algorithms design. *University of California, San Diego*.