3
FILTRAGEM COLABORATIVA

3.1
Introdução

Diversas empresas no setor de varejo investem significativamente na formação de bancos de dados de clientes e de compras. No caso de lojas virtuais, este investimento é ainda mais pronunciado devido ao fato de bancos de dados fazerem parte da própria operação. Juntamente com o investimento no armazenamento de dados, algoritmos vêm sendo formulados para aproveitar o volume de dados armazenados e gerar resultados para as empresas.

Sistemas de recomendação aplicam técnicas de análise de dados ao problema de auxiliar usuários a encontrar os itens que eles possam desejar. Filtros Colaborativos (FC) [16][17] constituem-se em uma técnica bem sucedida em diversas aplicações de recomendação, buscando similaridades em hábitos dos usuários para predizer suas decisões futuras.

Nesse capítulo será examinado o estado da técnica sobre algoritmos de filtragem colaborativa. Este é o algoritmo mais comumente utilizado em grandes lojas de varejo para recomendação. No capítulo 4 serão apresentados outros algoritmos de recomendação em estudo e uso.

3.2
Recomendações Baseados em Regras de Associação

Antes de iniciar a discussão dos filtros colaborativos em si, apresentamos o seu precursor, o algoritmo de recomendação baseado em regras de associação. Regras de associação foram inicialmente introduzidas por Agrawal [18] e pesquisas subseqüentes levaram à criação do conhecido algoritmo Apriori. Desde a sua criação, o algoritmo original sofreu diversas modificações com o objetivo de aumentar a sua eficiência.

Algoritmos baseados em regras de associação buscam, em bases de dados, regras que permitam fazer recomendações dos itens. Uma regra de associação é uma regra da forma X→Y, onde X e Y são conjuntos de itens, denominados, respectivamente, corpo e cabeça da regra. A razão dessa regras é que a presença de (todos os itens de) X em uma transação implica na presença de (todos os itens de) Y na mesma transação, com a mesma probabilidade. Cada regra de associação tem duas medidas relativas ao conjunto de transações: sua confiança e seu suporte. A confiança é a percentagem de transações que contêm Y entre as transações que contêm X; já o suporte é a percentagem de transações que contêm ambos X e Y entre todas as transações no banco de dados. [19].

Os algoritmos deste tipo geralmente criam muitas regras a um custo computacional enorme e com baixa acurácia.

3.3
Filtros Colaborativos

Filtragem colaborativa é o processo de filtrar informação ou padrões usando técnicas que envolvem colaboração de múltiplos agentes, pontos de vista, fontes de dados, etc. Filtros colaborativos funcionam construindo uma base de dados de preferências de itens para usuários. Um novo usuário é comparado a uma base de dados de forma a descobrir vizinhos, os quais são outros usuários que possuem características similares. Os itens de interesse para esses usuários vizinhos são então recomendados ao usuário inicial.

Existem diversos algoritmos para implementação de Filtragem Colaborativa, incluindo redes de crença bayesianas, *clusterização – k-nearest neighbour*, por exemplo – e algoritmos baseados em regressão. O Filtro Colaborativo é baseado na premissa de que, se dois usuários X e Y tiverem interesses similares, refletidos em votos similares com relações a n itens, então estes usuários irão demonstrar da mesma forma sua similaridade de interesses com relações a outros itens [20]. O Filtro Colaborativo (FC) pode obter opiniões dos usuários com relações a itens de forma explícita, como votações feitas pelos usuários, ou de forma implícita, a partir de históricos de compra [21].

Filtros colaborativos têm sido bem sucedidos em pesquisas e na prática, em aplicações em E-commerce e filtragem de informações. Porém, ainda existem importantes desafios a serem pesquisados no seu uso em sistemas de recomendação.

O primeiro desafio é aumentar a invariância à escala da base de dados nestes algoritmos de filtragem colaborativa. Esses algoritmos são capazes de buscar dezenas de milhares de vizinhos potenciais em tempo real, porém a demanda de sistemas modernos está em busca de dezenas de milhões de potenciais vizinhos. Além disso, os algoritmos apresentam problemas de desempenho em situações especiais, tal como o caso da existência de um usuário novo para o qual não existem informações sobre histórico de compras.

O segundo desafio é aumentar a qualidade das recomendações para usuários. Estes necessitam de recomendações que sejam confiáveis para ajudá-los a encontrar os itens que desejam.

Os dois desafios existem em conflito, uma vez que, quanto menos tempo o algoritmo leva buscando vizinhos, mais escalável ele será, e pior sua qualidade. Por essa razão, é importante tratar ambos os desafios simultaneamente para que as soluções sejam ao mesmo tempo utilizáveis e práticas. Em sistemas de recomendação, algoritmos "*anytime*" podem oferecer a capacidade de se substituir o esforço de se reduzir o tempo de execução pela qualidade dos resultados [22]. Mais a frente neste capítulo, serão apresentados, com maiores detalhes, os principais problemas dos filtros colaborativos (FC).

As técnicas de filtragem colaborativa podem ser divididas em três categorias:

Baseadas em Memória: Os algoritmos baseados em memória foram os primeiros a serem desenvolvidos e consideravam as avaliações de cada usuário sobre os itens, calculando uma medida de similaridade entre os usuários ou itens para poder efetuar predições ou recomendações. Apesar de serem fáceis de implementar e apresentarem bons resultados [23][24], tais algoritmos apresentavam problemas quando a quantidade de dados era esparsa e havia poucos itens semelhantes.

Baseadas em Modelo: Com o objetivo de tentar aprimorar o desempenho, desenvolveram-se os algoritmos baseados em modelo. Estes incorporam conceitos de aprendizado de máquina e de mineração de dados e atuam nas avaliações do

usuário sobre os itens, de forma a treinar um modelo capaz de efetuar predições [25]. Um exemplo desta abordagem é o uso de redes neurais bayesianas para recomendação [25][26][27].

Híbridas: Existem técnicas que aproveitam tanto os algoritmos de filtragem colaborativa baseados em modelo quanto os baseados em memória. A ligação pode ser feita de diversas formas, incluindo o uso em cascata ou usando ambos os algoritmos simultaneamente e tendo os resultados somados com aplicação de pesos para cada algoritmo. Também existem hibridizações de filtros colaborativos com outras técnicas de recomendação, como algoritmos baseados em conteúdo [28]. Em ambos os casos, busca-se gerar, via hibridização das técnicas, um algoritmo final que contorne as limitações das técnicas individuais.

O objetivo de um algoritmo para filtragem colaborativa é sugerir novos itens ou prever a utilidade de um determinado item para um usuário em particular com base nas preferências anteriores do usuário ou nas de usuários semelhantes. Em um cenário típico de uso de um FC, considere-se uma lista de *m* usuários U= $\{u_1, u_2,..., u_m\}$ e uma lista de *n* itens I= $\{i_1, i_2,..., i_n\}$. Cada usuário u_i possui uma lista de itens I_{ui} para os quais expressou seu interesse.

Opiniões podem ser oferecidas explicitamente por *notas,* geralmente dentro de uma escala numérica, ou podem ser explicitamente derivadas de históricos de compra, analisando-se dados temporais ou minerando-se hiperlinks de web, entre outras formas. É importante notar que $I_{ui} \subset I$ e é possível que I_{ui} seja um conjunto nulo. Existe um usuário distinguível $u_a \in U$, denominado usuário ativo, para o qual é tarefa do algoritmo de filtro colaborativo encontrar um item de interesse, que pode ocorrer de duas formas:

- **Predição:** valor numérico, P_{aj}, expressando a aproximação prevista do item $i_j \notin I_{u_a}$ para o usuário ativo u_a. Esse valor previsto está dentro da mesma escala dos valores de opinião oferecidos por u_a.
- **Recomendação:** lista de N itens, $i_r \subset I$, pelos quais o usuário ativo pode mais se interessar. Note-se que a lista recomendada deve ser de itens ainda não comprados pelo usuário ativo, i.e., $I_r \cap I_{u_a} = \phi$. Essa interface dos algoritmos FC é também conhecida como recomendação Top-N.

3.3.1
Algoritmos Baseados em Memória

Algoritmos baseados em memória utilizam toda a base de dados de usuários e itens para gerar a predição. Fazem uso de técnicas estatísticas para encontrar o conjunto de usuários, conhecidos como *vizinhos,* que possuem uma história de concordar com o usuário-alvo (i.e., ou avaliam item diferentes de forma similar ou tendem a comprar conjuntos de itens semelhantes). Uma vez formada uma vizinhança de usuários, esses sistemas usam diferentes algoritmos para combinar as preferências da vizinhança e produzir uma predição ou uma lista de recomendações para o usuário ativo.

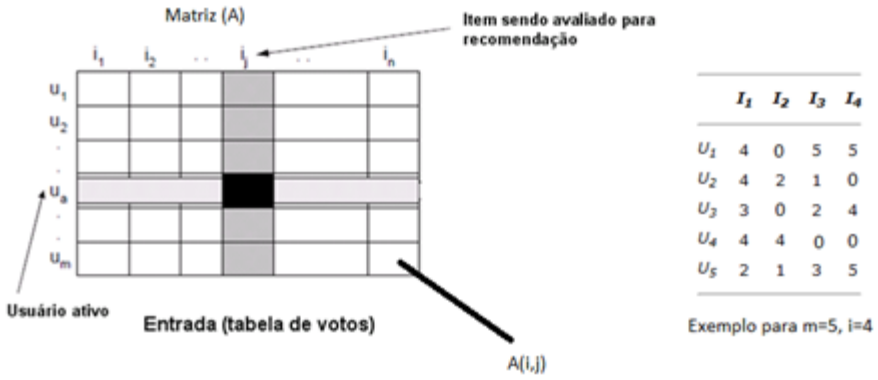
Essas técnicas, também conhecidas como vizinhança mais próxima ou filtragem colaborativa baseada em memória, são mais populares e vastamente utilizadas na prática, inclusive em aplicações comerciais (Amazon e Barnes & Noble, por exemplo), devido a sua facilidade de implementação e alta efetividade [23][24]. A customização do sistema de FC para cada usuário faz decrescer o esforço de busca por usuários. Essa técnica também promete grande lealdade de consumidores, altas vendas, mais receitas e benefícios em promoções customizadas [3].

3.3.1.1
FC Baseados em Memória – Usuários

No algoritmo de filtragem colaborativa baseada em usuários, o usuário-alvo (para o qual se quer fazer a recomendação) é comparado a outros k usuários que possuem maior similaridade (de vizinhança mais próxima) por meio de um modelo de vetor-espaço [25][29]. Nesse modelo, cada usuário é tratado como um vetor no espaço m-dimensional de itens e as similaridades entre o usuário ativo e os outros usuários são computadas entre os vetores.

A Figura 5 apresenta o modelo vetor-espaço para FC baseados em memória e usuários. Algoritmos de FC representam todos os dados $m \times n$ usuários-itens como uma matriz de pontuação (A). Cada entrada a_{ij} em A representa uma nota de preferência do i-ésimo usuário para o j-ésimo item. Cada pontuação individual

está dentro de uma escala numérica, podendo, inclusive, assumir o valor 0, demonstrando que o usuário não pontuou o item.



	I_1	I_2	I_3	I_4
U_1	4	0	5	5
U_2	4	2	1	0
U_3	3	0	2	4
U_4	4	4	0	0
U_5	2	1	3	5

Figura 5 – Matriz de Pontuação (A) com exemplo do modelo vetor-espaço em FC baseados em usuários.

Para cada dois usuários, o FC deve calcular a similaridade, ou peso entre eles. Essa medida reflete a distância ou correlação entre os itens ou usuários medidos. Após os *k* usuários mais similares terem sido descobertos, suas linhas correspondentes na matriz usuário-item (*A*) são agregadas para identificar o conjunto de itens, *C*, comprado pelo grupo, em conjunto com sua freqüência. Com o conjunto *C*, o algoritmo pode recomendar top-N itens mais freqüentes que o usuário ativo ainda não adquiriu.

Filtros colaborativos baseados em usuário foram muito bem sucedidos no passado, mas o seu uso bem disseminado permitiu que desafios reais emergissem, tais como:

- Esparsidade: na prática, recomendadores comerciais são usados para avaliar grandes conjuntos de itens (ex: Amazon.com recomenda livros e CDnow.com recomenda música). Nesses sistemas, mesmo usuários ativos compram menos de 1% do total de itens (1% de 2 milhões de livros são 20 mil livros). Um sistema de recomendação baseado na vizinhança mais próxima apresenta dificuldades de recomendar itens para um usuário em particular. A acurácia dos algoritmos é baixa.
- Escalabilidade: algoritmos de vizinhança mais próxima crescem computacionalmente com o número de usuários e itens. Com

milhões de itens e usuários, um sistema web típico de recomendação usando algoritmos baseados em usuários terá sérios problemas de escala.

3.3.1.2
FC Baseados Memória - Itens

Ao contrário do algoritmo de filtro colaborativo baseado em usuário, discutido na seção 3.2.1.1., o método baseado em item considera o conjunto de itens que o usuário-alvo avaliou previamente e computam quão similares eles são com relação a um item-alvo *i*. O algoritmo primeiro computa os *k* itens mais similares para cada item de acordo com suas similaridades, então identifica o conjunto *C* de itens candidatos a recomendação fazendo a união entre os k itens mais similares e removendo cada item no conjunto *U* que o usuário já comprou; então calcula as similaridades entre cada item do conjunto *C* e do conjunto *U*. O conjunto resultante de itens em C, é ordenado em ordem decrescente de similaridade, sendo recomendado como uma lista Top-N.

Um ponto crítico no algoritmo de FC baseada em itens é o cálculo da similaridade entre itens e a seleção dos mais semelhantes. A idéia básica no cálculo de similaridade entre dois itens *i* e *j* é, primeiramente, separar usuários que tenham avaliados esses itens e então aplicar a técnica de cálculo de similaridade para determinar a similaridade s_{ij} entre eles. A Figura 6 apresenta o processo, onde as linhas da matriz representam usuários e as colunas itens.

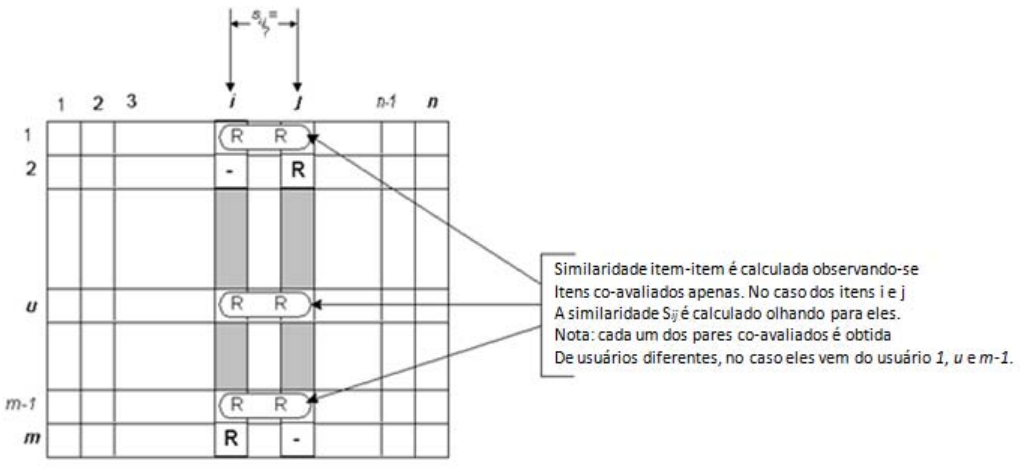


Figura 6 – Isolando itens co-avaliados e cálculo de similaridade.

Isolado o conjunto de itens similares baseados em medidas de similaridade, o próximo passo é analisar as avaliações do usuário e escolher uma técnica para gerar previsões de itens de seu interesse.

Uma das técnicas é a **soma ponderada,** que envolve computar as predições do item *i* para o usuário *u* pela soma das avaliações dadas pelo usuário a um item similar a *i*. Cada avaliação $R_{u,j}$ é ponderada pela similaridade s_{ij} entre os itens *i* e *j*. Formalmente, usando a técnica apresentada na Figura 7, pode-se denotar a predição de $P_{u,i}$ como:

$$P_{u,i} = \frac{\sum_{todos\ itens\ similares,N}(s_{i,N} * R_{u,N})}{\sum_{todos\ itens\ similares,N}(|s_{i,N}|)} \qquad 1$$

Essa técnica tenta capturar como o usuário ativo avalia itens similares. A média aritmética é ponderada com a soma dos termos de similaridades para garantir que as predições estejam em limites pré-definidos.

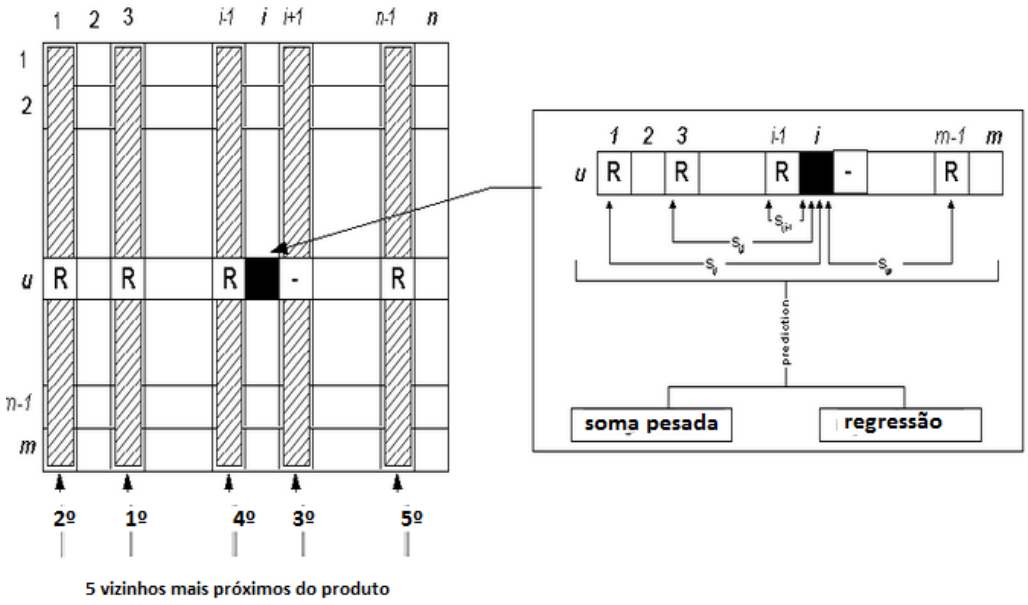


Figura 7 – Algoritmo de filtragem colaborativa baseada em item. Processo de geração de predição ilustrado por 5 vizinhos.

A técnica de **regressã**o, similar à da **soma ponderada**, usa uma aproximação das avaliações baseada em um modelo de regressão. Na prática, as similaridades são computadas usando o cosseno ou medidas de correlação que podem causar resultados ruins quando usado o método anterior, na medida em que

dois vetores de avaliação podem estar distante no sentido euclidiano, mas ainda assim possuírem alta similaridade, sobretudo devido aos casos de ovelha cinza (quando o usuário de um determinado nicho tem comportamento de compra diferente de outros do mesmo nicho) ou de um novo usuário ou item na base de dados. Nesse caso, usar as avaliações básicas do item "similar" pode resultar em uma predição ruim. A idéia básica é usar a mesma expressão da técnica da soma ponderada, mas, ao invés de utilizar as avaliações puras dos N itens, R_{uN}, esse modelo utiliza valores aproximados R_{uN} baseados em um modelo de regressão linear. Denotando-se os vetores respectivos do item alvo i e o item similar N por R_i e R_N, o modelo de regressão linear pode ser expresso como:

$$\bar{R}_N = \alpha \bar{R}_i + \beta + \epsilon \qquad 2$$

Os parâmetros de regressão do modelo α e β são determinados sobre ambos os vetores de avaliação. ϵ é o erro do modelo de regressão.

FC baseados em itens foram desenvolvidos para resolver problemas de escalabilidade existentes nos recomendadores baseados em usuários. Um dos problemas que podem incorrer do uso desta metodologia é que quando a distribuição conjunta de itens é diferente da distribuição de itens individuais, o resultado do uso deste algoritmo pode gerar recomendações sub-ótimas. Para resolver esse problema, Deshpandle e Karypis [30] desenvolveram um sistema de recomendação usando FCs baseados em itens de maior ordem que empregam todas as combinações de itens até um tamanho em particular para determinar os conjuntos de itens a serem recomendados ao usuário.

3.3.1.3
Cálculo de Similaridade

Os algoritmos de FC apresentados até o momento necessitam de uma forma de comparar usuários ou itens de modo a descobrir aqueles mais próximos entre si. Existem diversas formas de se calcular a similaridade entre usuários e itens. As mais comumente utilizadas são apresentadas nas seções a seguir.

- **Similaridade por Correlação Pearson**

A similaridade por correlação utiliza o cálculo da *Correlação Pearson* como base para comparar itens e usuários. Essa medida de similaridade considera o quanto duas variáveis são relacionadas linearmente [16]:

$$w_{u,v} = \frac{\sum_{i \in I}(r_{u,i} - \overline{r_u})(r_{v,i} - \overline{r_v})}{\sqrt{\sum_{i \in I}(r_{u,i} - \overline{r_u})^2}\sqrt{\sum_{i \in I}(r_{v,i} - \overline{r_v})^2}} \quad 3$$

Onde, os somatórios são sobre todos os itens que ambos os usuários *u* e *v* avaliaram e $\overline{r_u}$ é a média das avaliações dos itens co-avaliados pelo *u*-ésimo usuário. No exemplo da Tabela 2 tem-se $w_{1,5}$=0.8. A Figura 8 apresenta uma representação genérica desta matriz de avaliação.

Tabela 2 – Exemplo de matriz de avaliação

	I_1	I_2	I_3	I_4
U_1	4	0	5	5
U_2	4	2	1	0
U_3	3	0	2	4
U_4	4	4	0	0
U_5	2	1	3	5

$$w_{1,5} = \frac{\sum_{i \in I}(r_{1,i} - \overline{r_1})(r_{5,i} - \overline{r_5})}{\sqrt{\sum_{i \in I}(r_{1,i} - \overline{r_1})^2}\sqrt{\sum_{i \in I}(r_{5,i} - \overline{r_5})^2}} =$$

$$\frac{(4-3.5)*(2-2.75)+(0-3.5)*(1-2.75)+(5-3.5)*(3-2.75)+(5-3.5)*(5-2.75)}{\sqrt{(4-3.5)^2+(0-3.5)^2+(5-3.5)^2+(5-3.5)^2}\sqrt{(2-2.75)^2+(1-2.75)^2+(3-2.75)^2+(5-2.75)^2}} =$$

$$\frac{(0.5)*(-0.75)+(-3.5)*(-1.75)+(1.5)*(0.25)+(1.5)*(2.25)}{\sqrt{(0.5)^2+(-3.5)^2+(1.5)^2+(1.5)^2}\sqrt{(-0.75)^2+(-1.75)^2+(0.25)^2+(2.25)^2}} =$$

$$\frac{-0.375+6.125+0.375+3,375}{\sqrt{17}\sqrt{8,1875}} = \frac{-0.375+6.125+0.375+3,375}{\sqrt{17}\sqrt{8,1875}} =$$

$$\frac{-0.375+6.125+0.375+3,375}{4,123*2,86} = \frac{9,5}{11,79} = 0,8$$

Esta mesma expressão pode ser aplicada com ligeiras alterações no caso do FC baseado em itens. Considerando-se o conjunto de usuários $u \in U$ que avaliaram ambos os itens *i* e *j*, a *Correlação Pearson* será:

$$w_{i,j} = \frac{\sum_{u \in U}(r_{u,i} - \overline{r_i})(r_{u,j} - \overline{r_j})}{\sqrt{\sum_{u \in U}(r_{u,i} - \overline{r_i})^2}\sqrt{\sum_{u \in U}(r_{u,j} - \overline{r_j})^2}} \qquad 4$$

Onde $r_{u,i}$ é a avaliação do usuário *u* ao item *i* e $\overline{r_i}$ é a avaliação média do *i*-ésimo item pelos usuários [31].

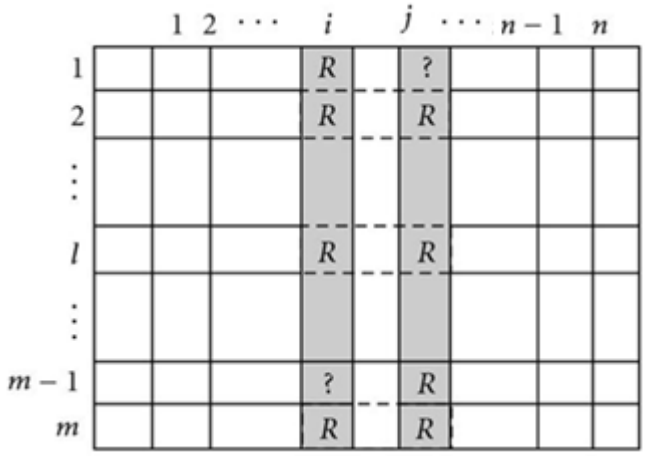


Figura 8 – Similaridade para FC baseados em itens ($w_{i,j}$) calculo baseado nos itens co-avaliados i e j pelos usuários 2, l e n.

- **Similaridade por cosseno de vetor**

A similaridade por cosseno de vetor considera os vetores de avaliação entre dois usuários ou itens com os quais se quer avaliar a similaridade. Seja *A* a matriz *m x n* usuário-item; então, a similaridade entre dois itens *i* e *j* é definida pelo cosseno entre os vetores de dimensão *n* correspondentes às *i*-ésima e *j*-ésima colunas da matriz *A*. A similaridade por cosseno entre itens $i=\{i_1,i_2,...,i_n\}$ e $j=\{j_1,j_2,...,j_n\}$ é dada por:

$$w_{i,j} = \cos(\vec{i},\vec{j}) = \frac{\vec{i}\cdot\vec{j}}{\|\vec{i}\|\|\vec{j}\|} = \frac{i_1 j_1 + i_2 j_2 + \cdots + i_n j_n}{\sqrt{{i_1}^2 + {i_2}^2 + \cdots + {i_n}^2}\sqrt{{j_1}^2 + {j_2}^2 + \cdots + {j_n}^2}} \quad (5)$$

Onde "." denota o produto escalar entre ambos os vetores [29].

- **Outras Similaridades**

Outras medidas de similaridade baseadas em correlação incluem: *Correlação Pearson* Limitada, Correlação de Ranking Spearman, Correlação Kendall, entre outras que são pouco utilizadas na literatura, uma vez que a grande maioria de testes com filtros colaborativos ou utilizam correlação Pearson simples ou cosseno. Mais informações sobre estas e outras correlações podem ser encontradas em [20][32][33].

Outras medidas de similaridade como a de probabilidade condicional [30] [34] não são usualmente empregadas e não serão discutidas aqui.

3.3.1.4 Extensões a Algoritmos Baseados em Memória

- **Votação Padrão**

Um dos problemas de filtros colaborativos baseados em memória ocorre quando alguns itens não possuem muitas avaliações. Isso acontece sobretudo com itens e usuários novos, uma vez que um usuário novo não terá avaliado item algum e um item novo não terá sido avaliado por usuário algum. Este problema afeta o desempenho do algoritmo e, para contornar esta situação, foi proposto atribuir valores padrões quando há poucas avaliações. Uma forma de se fazer isto é usar a média de avaliação dos itens como valor padrão para complementar o histórico de avaliação de cada usuário [35].

- **Frequência de Usuário Inversa**

Quando, em uma base existem itens que são avaliados por todos os usuários igualmente, estes itens contribuem pouco para a geração de recomendações. Considere-se, por exemplo, o caso de Best-sellers que são bem avaliados por uma grande quantidade de pessoas: estas recomendações dão poucas informações sobre quão similares esses usuários realmente são.

Buscando dar maior destaque aos itens que não são tão avaliados, adota-se a freqüência de usuário inversa [36] definida como $f_j=log(n/n_j)$, onde n_j é o número de usuários que avaliaram o item j e n o número total de usuários. Se todos tiverem avaliado o item j, então f_j será zero. Para aplicar este método com algoritmos de similaridade baseados em vetores, é necessário utilizar uma avaliação transformada, na qual as avaliações são multiplicadas pelo fator f_j [25].

- **Amplificação de Pesos**

A amplificação de pesos atua sobre os valores calculados de similaridades e busca reduzir o ruído, favorecendo similaridades mais fortes e praticamente zerando as similaridades menores [25]. Tal método faz uso da expressão:

$$w_{i,j} = w_{i,j} * \left|w_{i,j}\right|^{\rho-1} \qquad 6$$

Onde ρ é a potência de amplificação, $\rho \geq 1$, sendo ρ=2.5 um valor típico [37].

- **Algoritmos de Adição Impulsionada**

Como mencionado anteriormente, um dos problemas comuns de filtros colaborativos ocorre na presença de bancos de dados esparsos. Isto ocorre com muita freqüência, pois as lojas virtuais tendem a ter uma quantidade de itens imensamente grande e os usuários são incapazes de avaliá-los todos.

O algoritmo de adição impulsionada proposto por Su [38] [39] inicialmente adiciona os dados faltantes, por meio de um classificador baseado em aprendizado de máquina, e depois usa a *correlação Pearson* sobre os dados completos para

efetuar a avaliação específica do usuário em um item. Essa adição pode ser feita de diversas formas: por média, regressão linear e preditiva por média [40].

3.3.2
Algoritmos Baseados em Modelo

Algoritmos de filtragem colaborativa baseados em modelo oferecem recomendações de itens a partir de um modelo de votos do usuário. Algoritmos dessa categoria utilizam uma abordagem probabilística e transformam o processo de recomendação no processamento do valor esperado de um item para um usuário, dadas as avaliações de outros itens. O processo de modelagem é realizado por outros algoritmos de aprendizado de máquina, tais como redes bayesianas, clusterização e extração de regras.

As redes bayesianas [25] formulam um modelo probabilístico para o problema de filtragem colaborativa. Modelos de clusterização [25],[41],[42] agrupam usuários similares na mesma classe. O modelo de clusterização estima a probabilidade de um usuário particular estar em uma classe particular C e infere para todos os membros dessa classe as avaliações dos itens.

Os modelos baseados em extração de regras fazem a análise de itens que foram comprados juntos e a partir destes dados criam regras que permitam fazer a recomendação [29].

Filtros Colaborativos baseados em modelo, como modelos Bayesianos, modelos de clusterização, e redes de dependência, têm sido investigados para resolver os problemas de FC baseados em memória [25] [41]. Usualmente, algoritmos de classificação podem ser usados como modelos de FCs se as avaliações dos usuários forem categóricas; modelos de regressão e métodos de Decomposição de Valores Singulares (SVD – Singular Value Decomposition) podem ser usados para avaliações numéricas [41].

3.3.2.1
Algoritmos FC com redes de crença Bayesiana

Uma rede Bayesiana (BN) é um modelo gráfico que representa um conjunto de variáveis aleatórias e suas interdependências condicionais via um grafo acíclico

direcionado (DAG – Directed Acyclic Graphic). Os nodos de tais grafos representam variáveis randômicas em um sentido bayesiano. As variáveis podem ser quantidades observáveis, latentes, parâmetros desconhecidos ou hipóteses. As extremidades do grafo representam dependências condicionais; nós que não são conectados representam variáveis que são condicionalmente independentes entre si. Cada nó é associado a uma função de probabilidade que tem como entrada um conjunto particular de valores para as variáveis do nó pai e associa uma probabilidade à variável representada pelo nó.

Por exemplo, se os pais são *m* variáveis Booleanas, então a função de probabilidade pode ser representada como uma tabela de 2^m entradas, sendo uma entrada para cada uma das 2^m possíveis combinações de seus pais. Cada entrada pode ser verdadeira ou falsa.

Redes Bayesianas são comumente utilizadas em problemas de classificação. Quando utilizado para a tarefa de filtragem colaborativa, o algoritmo bayesiano simples considera o problema de predição como um problema de classificação. Utiliza-se bayes ingênuo (Naive Bayes – NB) em tarefas de classificação, assumindo-se que a presença (ou ausência) de uma característica em particular de uma classe não é relacionada à presença (ou ausência) de qualquer outra característica. Aplicado à FC, as características são as avaliações dos usuários e cada item possui seu classificador. A probabilidade calculada de certo item pertencer a uma classe para o usuário (por exemplo, classe comprar ou não comprar) é calculada dadas todas as características (avaliações de outros usuários), e então a classe com maior probabilidade será classificada como a classe prevista [43].

O classificador é treinado com todos os usuários que votaram no item. Para dados incompletos, o cálculo de probabilidade e classificação é computado sobre dados observados:

$$c = \arg max_{j \in C} \; P(c_j) \prod_o P(X_o = x_o | c_j) \qquad 7$$

Onde *c* é uma classe, *C* é um conjunto de classes, c_j é a *j*-ésima classe do conjunto C, *o* representam dados observados e X_o representa uma característica observada.

Em Miyahara e Pazzani [26], são utilizadas duas classes (*gostou* e *não gostou*) e as avaliações do usuário sobre os itens são distribuídas em uma matriz de avaliação booleana. O algoritmo passa a predizer a avaliação (classe) de itens não avaliados pelo usuário. A implementação de NB para aplicações de FC reduz a complexidade computacional quando comparado aos baseados em memória.Por outro lado, em [26] os dados multi-classe são inicialmente convertidos em dados binários, e então convertidos em uma vetor de avaliações com características booleanas. Essas conversões causam problemas com escalabilidade e perda de informação multi-classe caso os dados sejam multi-classe.

Um FC simples bayesiano tem funcionamento similar ao FC baseado em correlação, uma vez que ambos fazem predições baseadas em avaliações observadas, e o processo de predição é menos demorado [27].

3.3.2.2
Algoritmos de FC com Regressão Logística Estendida

A Regressão Logística Estendida (ELR) tem sido igualmente utilizada para aplicações em FC. Em estatística, regressão logística é usada para prever a probabilidade de ocorrência de um evento ao encaixar dados em uma curva logística. É um modelo generalizado linear usado para regressão binomial. ELR é um algoritmo de gradiente decrescente de aprendizado de parâmetros discriminativos que maximiza a probabilidade condicional logarítmica e estende a regressão logística para problemas de classificação, com performance melhor que algoritmos bayesianos simples [44][45].

Os resultados empíricos mostram que os algoritmos de filtragem colaborativa baseados em Bayes Ingênuo otimizado por ELR (NB-ELR), quando atuando em bancos de dado reais de FC multiclasse e usando MAE (Erro Médio Absoluto) como critério de avaliação, apresentam desempenho significativamente superior ao do algoritmo de FC simples bayesiano, e consistentemente melhor do que a *Correlação Pearson* em algoritmos de FC baseados em memória [27]. O NB-ELR, porém, precisa de um maior tempo de treinamento de modelos. Usualmente este problema é contornado aplicando o treinamento off-line, e o estágio de predição online para ter um menor tempo.

Outra abordagem para o uso de algoritmos bayesianos em filtragem colaborativa é utilizar árvores de decisão em cada nó das redes de crença bayesianas. Cada nó da árvore de decisão corresponde a um item e os estados de cada nó correspondem às possíveis avaliações [25] Este modelo tem desempenho de predição similar ao FC de *Correlação Pearson*, e com melhor desempenho que Clusterização Bayesiana e que algoritmos baseados em cosseno de vetor em FC baseados em memória.

3.3.2.3
Algoritmos FC de *Clusterização*

Clusterização ou Agrupamento são algoritmos que determinam conjuntos de dados que possuem semelhanças entre seus padrões constituintes. As técnicas de *clusterização* buscam separar dados em categorias ou grupos com padrões parecidos. As similaridades entre os dados dentro de um mesmo grupo podem ser mensuradas por métricas tais como a *Correlação Pearson*.

É possível classificar os métodos de *clusterização* em três categorias: métodos de particionamento, métodos baseados em densidade e métodos hierárquicos [46][47]. A aplicação de algoritmos de *clusterização* em FC não é feita de forma isolada, mas sim associada a outras técnicas de FC de forma a reduzir suas deficiências. Por exemplo, se utiliza um algoritmo de *clusterização* para dividir dados em clusters e depois filtros colaborativos baseado em *Correlação Pearson* dentro de cada cluster para fazer predições [48][49].

O método RecTree [35], por exemplo, utiliza o método *k-means* com k=2 para dividir a base de dados de avaliação em dois sub-clusters e constrói o *RecTree* a partir da raiz e folhas, gerando uma árvore binária não balanceada onde cada nó folha tem uma matriz de similaridade e nós internos possuem centróides de avaliações de sub-árvores. A predição é feita dentro das folhas nas quais cada usuário ativo pertence. O algoritmo possui acurácia melhor que FC baseados em *Correlação Pearson*, porém é restrito na medida em que necessita que o tamanho do banco de dados e das partições sejam constantes, o que é incomum no mundo real.

Modelos de *clusterização* possuem melhor escalabilidade do que os métodos de filtragem colaborativa típicos devido ao fato de fazerem predição

dentro de clusters de menor tamanho ao invés de toda a base de consumidores [29] [35][50][51]. A qualidade das recomendações é geralmente baixa e técnicas para tornar essa qualidade melhor geralmente tornam o problema tão pesado computacionalmente que seu desempenho passa a ser similar a filtros colaborativos baseados em memória [20]. Além disto, o uso desta técnica em bancos de dados grandes é impraticável, necessitando de técnicas de amostragem que reduzem a acurácia.

3.3.2.4
Algoritmos FC baseados em regressão

Métodos de regressão apresentam bons resultados em valores numéricos tais como as avaliações de usuários. A regressão é usada para aproximar avaliações e fazer as predições baseadas em regressão. Se $X=\{X_1,X_2,...,X_n\}$ for uma variável randômica que representa as preferências do usuário por itens, o modelo de regressão é expresso por:

$$Y = \Delta X + N \qquad 8$$

Onde Δ é uma matriz *n x k*, $N=\{N_1,N_2,...,N_n\}$ é uma variável aleatória representando o ruído das escolhas do usuário, Y é uma matriz *n x m* onde Y_{ij} representa a avaliação do usuário *i* sobre o item *j*, e X é a matriz *k x m* com cada coluna sendo uma estimativa do valor da variável randômica X (avaliações do usuário no espaço *k*-dimensional de avaliações) para um usuário. Tipicamente, a matriz Y é muito esparsa.

Vucetic e Obradovic [53] propuseram uma abordagem a FC baseada em regressão em dados de avaliação numérica que busca por similaridade entre itens, cria uma coleção de modelos lineares simples, e combina-os eficientemente para prover predições de avaliação para usuários ativos. Eles usaram médias quadráticas ordinárias para estimar parâmetros da função linear de regressão. Seus resultados sintética mostram que a abordagem possui boa performance com relação à esparsidade, latência de predição e problemas de predição numérica em tarefas de FC.

3.3.2.5
Outras técnicas de FC baseadas em modelo

Ao invés de abordarem o processo de recomendação como um problema de predição, Shani [54] abordaram o problema como otimização seqüencial e usaram modelos de Processos de Decisão de Markov (MDPs) [55] para sistemas de recomendação. Trabalhando em uma livraria online israelita, Mitos, o sistema de recomendação MDP utilizado produziu maior lucro que o sistema sem o recomendador. Também, o FC MDP atua muito melhor que uma cadeia simples de Markov (MC).

Existem abordagens ao problema de filtragem colaborativa utilizando semântica latente. O modelo de aspecto, proposto por Hofmann e Puzicha [51] cria um modelo de espaço-latente probabilístico, no qual modelos de avaliação individuais são associados a cada par observado de {usuário, item}, assumindo que usuários e itens são independentes. A performance do modelo de aspecto é muito melhor que o modelo de clusterização trabalhando no banco de dados de filmes *MovieLens*.

3.3.3
Técnicas de Filtragem Colaborativa Híbridas

As técnicas de FC híbridas contam com o uso de dois ou mais algoritmos de recomendação trabalhando em conjunto para gerar predições ou recomendações. Geralmente, essas técnicas contam com uso de recomendações baseadas em conteúdo para reduzir as limitações das técnicas puramente baseadas em filtragem colaborativa, porém existem casos de uso de diferentes técnicas de filtragem colaborativa em conjunto (por exemplo, baseadas em memória e modelo juntas).

Os sistemas de recomendação baseados em conteúdo serão explicados com mais detalhes no capítulo 4. Ao contrário de filtragem colaborativa, que se baseia em vizinhança dos usuários ou itens e em suas avaliações de produtos, os recomendadores baseados em conteúdo fazem recomendações analisando o conteúdo das informações textuais, tais como documentos, URLs, mensagens de notícias, logs web, descrições de itens, características sobre os gostos dos usuários, etc [5].

Técnicas baseadas em conteúdo necessitam de informações sobre os usuários e itens e sofrem de problemas de super-especialização [56][57].

Na esperança de evitar limitações de cada tipo de sistema de recomendação e melhorar o desempenho, recomendadores FC híbridos são combinados de diversas formas conforme será abordado a seguir.

3.3.3.1
Recomendadores Híbridos incorporando FC e Características baseadas em conteúdo.

Um dos grandes problemas que existem em filtros colaborativos é o de esparsidade dos dados e de inicialização. O primeiro ocorre devido ao fato de um usuário avaliar muito poucos itens da base, o segundo ocorre quando um novo item/usuário é adicionado à base e não foi avaliado/não avaliou.

Uma abordagem híbrida busca resolver este problema, preenchendo as avaliações faltantes em ambos os casos com predições. Por exemplo, utilizando-se de Bayes ingênuo como classificador de conteúdo, cria-se uma nova matriz de avaliação onde todos os itens são avaliados por todos os usuários, sendo que as avaliações já feitas são mantidas. A partir desta pseudo-matriz, utiliza-se um FC baseado em *Correlação Pearson* para fazer a recomendação ou predição final [58]. Esta técnica, chamada de Recomendador de FC com Conteúdo Impulsionado (Content-Boosted Collaborative Filter), melhorou a performance da predição com relação aos recomendadores baseados em conteúdo puros e sobre o FC baseado em memória puro, resolvendo o problema de inicialização e o problema de esparsividade nas tarefas de FC.

Outra forma de combinar recomendadores por conteúdo e Filtros Colaborativos está em combinar os pesos (similaridades nos casos de FC) de ambas as técnicas [59]. A combinação pode ser feita de forma linear, por pesos ajustáveis [60], por votos nos pesos maiores [61] ou votos dos pesos médios [62]. Também se pode usar a troca de técnicas de recomendação, na qual o Filtro Colaborativo é substituído por um recomendador baseado em conteúdo quando começa a ter problemas em sua performance [59].

Outros recomendadores híbridos nessa categoria incluem Recomendadores Híbridos misturados [63], recomendadores híbridos em cascata [59] entre outros.

Muitos artigos empiricamente comparam o desempenho de recomendadores híbridos com o FC puro e métodos baseados em conteúdo e descobrem que recomendadores híbridos podem levar a melhor acurácia nas recomendações, especialmente para situações de novo usuário ou novo item onde o FC regular não traz recomendações satisfatórias. Porém, recomendadores híbridos dependem de informação externa que usualmente não está disponível, e eles possuem geralmente maior complexidade de implementação [5] [59].

3.3.3.2
Recomendadores Híbridos combinando Algortitmos de FC

Existem diversas formas de mesclar os filtros colaborativos baseados em memória e modelo, e os desempenhos destes algoritmos híbridos são geralmente melhores que algoritmos baseados em memória ou modelo puros [64][65].

FC baseados em memória probabilística (PMFC) combina técnicas baseadas em memória e em modelo [64]. Tais métodos utilizam um modelo misturado com base em um conjunto de informações do usuário e fazem uso de uma distribuição a posteriori das avaliações dos usuários para fazer predições. Para atuar sobre o problema do novo usuário, uma extensão de aprendizado do PMFC pode ser usada para ativamente buscar no usuário informação adicional quando existirem informações insuficientes disponíveis. Para reduzir o tempo computacional, PMFC seleciona um pequeno subconjunto chamado espaço de informações do banco de dados inteiro de avaliações de usuários e faz predições deste pequeno subconjunto ao invés do banco inteiro. PMFC tem melhor acurácia que os FC baseados em *Correlação Pearson* e FC baseados em modelo usando Bayes ingênuo.

3.4
Desafios de Filtros Colaborativos

Os desafios encontrados em filtros colaborativos se baseiam tanto em características do banco de dados sobre os quais atuam quanto das necessidades de negócio que são demandadas deles. Quanto ao banco de dados, em aplicações reais de filtragem colaborativa, tais bancos terão uma quantidade extremamente

grande de usuários e itens diversificados, onde dificilmente um usuário terá avaliado uma quantidade substancial de itens. Pelo ponto de vista das necessidades de negócio, o recomendador é demandado a ter velocidade alta, acurácia e de trazer lucro para as empresas que o empregam, na medida em que aumentem as vendas.

3.4.1
Esparsidade dos Dados

Os bancos de dados comerciais nos quais os filtros colaborativos geralmente são utilizados, fora do ambiente de pesquisa, são constituídos de uma grande quantidade de usuários e itens, uma vez que se considera que empresas virtuais não possuem limitações de tamanho da prateleira onde por seus produtos. Assim, a matriz usuários-itens é extremamente esparsa e a performance de sistemas de FC para recomendação ou predição podem ser desafiadoras.

O problema de esparsidade dos dados pode ser divido em três:

- Inicialização: Ocorre quando um novo usuário é adicionado à base de dados, não tendo ainda avaliado nenhum item, ou quando um novo item é adicionado à base de dados, não tendo sido avaliado por nenhum outro usuário. Em ambas as situações os filtros colaborativos não terão bases sobre as quais fazerem recomendações [56].
- Cobertura: Bancos de dados de empresas varejistas, sobretudo as online, são considerados de cauda longa (long tail), significando que existem diversos itens que nunca foram avaliados. Isso é possível devido ao fato do custo de um item na base de dados ser quase nulo. É comum em bancos de dados de lojas virtuais que um usuário não recomende nem 10% do total de itens a disposição. Quando o número de itens é muito superior ao de recomendações de usuário, o FC se torna incapaz de recomendar bem os itens menos avaliados.
- Transitividade de Vizinhança: Ocorre quando usuários com mesmos tipos de gostos não tiverem, apesar disto, avaliado os mesmos itens. Por exemplo, digamos dois usuários que gostem da mesma marca de produtos. Porém um deles comprou uma televisão da marca e o outro comprou um DVD desta marca. Ambos têm essa similaridade de gostar

da marca, porém como a base de dados é esparsa e possui muitos itens, pode ser que essa similaridade não seja vista pelo FC, uma vez que itens diferentes foram comprados.

Diversos métodos foram apresentados no estado da arte para resolver o problema de esparsidade do banco de dados. Entre eles estão o de redução de dimensionalidade (por exemplo, com SVD – Singular Value Decomposition), Indexação por semântica latente [67] e Análise de Componentes Principais (PCA). Porém estas técnicas causam o descarte de usuários e itens menos relevantes e podem degradar a qualidade da recomendação [29].

Algoritmos de FC híbridos que mesclam o uso de filtragem colaborativa com recomendadores baseados em conteúdo demonstraram bons resultados quanto à esparsabilidade do banco de dados [58]. Por outro lado estes algoritmos dependem de informações externas quanto aos usuários e itens que podem não estar presentes.

Algoritmos de FC baseados em modelo tratam a esparsidade oferecendo predições acuradas para dados esparsos. Algumas novas técnicas de FC baseadas em modelos que atacam o problema de esparsidade incluem técnica de recuperação de associação [68].

3.4.2
Escalabilidade dos Dados

Em aplicações reais de uso de filtragem colaborativa, os bancos de dados podem ser extremamente extensos. Por exemplo, o banco de dados da Amazon.com possui 29 milhões de usuários e vários milhões de itens listados no catálogo. A maioria dos algoritmos tradicionais são testados sobre bases de dados muito menores, tal como a MovieLens que possui 35 mil usuários e 5 mil itens. Estes sistemas de recomendação ainda por cima devem rodar em tempo-real e fazer recomendações a todos os usuários independentes de seu histórico de avaliação.

Técnicas de redução de dimensionalidade como SVD resolvem problemas de escala e produzem boas recomendações, porém se faz necessário pesados passos de fatorização de matriz, o que pode causar perda de informação. Os algoritmos baseados em *clusterização* podem ter seu processamento mais pesado

rodado off-line, porém causam problemas significativos de desempenho. Algoritmos de FC baseados em memória, como os de *Correlação Pearson* baseados em itens podem alcançar escalabilidade satisfatória [23].

3.4.3
Sinônimos

O problema relacionado a sinônimos ocorre quando existem na base de dados itens iguais nomeados de forma diferente. Por exemplo, digamos que haja em uma base de dados de livros o mesmo livro adicionado em versões diferentes (edição 1 e edição 2 por exemplo). O livro possui o mesmo conteúdo e provavelmente atrairá os mesmos usuários independentes da versão, porém para o filtro colaborativo, será tratado como itens diferenciados, causando problemas de desempenho.

Algoritmos de FC hibridizados com recomendadores baseados em conteúdo podem ser bem sucedidos em resolver este problema, uma vez que o recomendador de conteúdo teria condições de reconhecer a semelhança dos itens baseado em suas características.

3.4.4
Ovelha Cinza

Ovelha cinza refere-se a usuários cujas opiniões não são consistentes em concordância ou discordância com nenhum grupo de pessoas e, portanto, não se beneficiam de filtros colaborativos. Um exemplo disto ocorre quando um leitor de romances entra em uma livraria de culinária. Como o usuário é muito diferente dos demais, o filtro colaborativo não encontrará vizinhos próximos e isso causará problemas de performance. Apesar de esta ser uma falha dos sistemas de recomendação, recomendadores não-eletrônicos podem resolver estes casos e geralmente são considerados um problema aceitável.

Claypool criou um sistema híbrido que combina FC e recomendadores baseados em conteúdo para predizer por uma média de pesos. Nessa abordagem, os pesos de ambos os algoritmos são determinados por usuário, permitindo que o sistema tenha resultados positivos para o problema da ovelha cinza [60].

3.4.5
Má Fé

É comum em ambientes colaborativos onde a pessoa tem direito a adicionar itens e ter acesso a votos nestes itens, que algumas pessoas dêem muitas recomendações positivas a seus próprios itens e recomendações negativas aos de seus concorrentes. Uma forma de se evitar este problema que pode ser feita para alguns sistemas é utilizar a própria compra do item como avaliação. É importante que os sistemas em que os filtros colaborativos sejam projetados de forma a impedir a má fé dos usuários para manipular as recomendações [69].

3.4.6
Outros Desafios

Um dos problemas importantes para empresas quando se lida com seus bancos de dados é a privacidade. Problemas de privacidade de informação dos usuários podem causar desde problemas judiciais até desfavorecimento da marca e redução nas vendas. Existem métodos de filtragem colaborativa que abordaram o assunto da privacidade, dentre eles: [21] e [70]. Outros desafios incluem o ruído quando a população é muito diversa e a ausência de explicação dos resultados, pois filtros colaborativos geralmente não trazem uma explicação para seus resultados que poderiam ser usados para vender os itens.

3.5
Métricas de Avaliação

Existem diversas métricas utilizadas na literatura para comparar sistemas de recomendação. De acordo com Herlocker [33], métricas de avaliação de sistemas de recomendação podem ser classificada segundo as seguintes categorias: métricas de acurácia preditiva, tal qual o *Erro Médio Absoluto* (MAE) e suas variações; métricas de acurácia de classificação, tal qual *precisão*, *medida-F1* e *sensitividade ROC*; métricas de acurácia de ranking, tal como *Correlação Produto-Momento de Pearson*, *Tau de Kendall e Precisao de Média* (MAP).

Diversas métricas de avaliação podem ser encontradas em Herlocker [33] e em [71].

3.5.1.1
Erro Médio Absoluto (MAE) e Erro Médio Absoluto Normalizado (NMAE)

Ao invés da acurácia de classificação ou erro de classificação, a mais usada métrica para pesquisa em FC na literatura é a média absoluta de erro (MAE) [20][33], a qual computa a média absoluta da diferença entre as predições e as avaliações reais:

$$MAE = \frac{\sum_{\{i,j\}} \left| p_{i,j} - r_{i.j} \right|}{n} \qquad 9$$

Onde n é o total de avaliações feitas por todos os usuários, $p_{i,j}$ é a avaliação prevista do usuário i no item j, e $r_{i,j}$ sua real avaliação. Quanto menor a MAE melhor a predição.

Diferentes sistemas de recomendação usam diferentes escalas de avaliação. A média normalizada da média absoluta de erro (NMAE) normaliza a MAE e expressa os erros em porcentagens de uma escala cheia [20]:

$$NMAE = \frac{MAE}{r_{max} - r_{min}} \qquad 10$$

onde r_{max} e r_{min} são os valores máximo e mínimo das avaliações, respectivamente.

3.5.1.2
Erro Médio Quadrático (RMSE)

Erro médio quadrático (RMSE) está se tornando popular devido ao prêmio NetFlix [72] para performance de recomendações de filmes:

$$RMSE = \sqrt{\frac{1}{n} \sum_{\{i,j\}} \left(p_{i,j} - r_{i.j} \right)^2} \qquad 11$$

onde n é o total número de avaliações sobre todos os usuários, p_{ij} é a avaliação prevista para o usuário i sobre o item j e r_{ij} é a avaliação real. A métrica RMSE amplifica as contribuições de erros absolutos entre predições e valores reais.

Apesar de métricas de acurácia tenham ajudado na área de sistemas de recomendação, os recomendadores que são mais acurados são muitas vezes os menos úteis para usuários, por exemplo, usuários podem preferir ser recomendados por itens que são desconhecidos por eles, ao invés de ter os antigos favoritos que eles provavelmente não vão querer novamente [73]. Portanto é necessário reavaliar as métricas.

3.5.1.3
Precisão/Revocação e F1

Precisão e Revocação são métricas populares de avaliação de sistemas de recuperação de informação, inicialmente propostas em 1968 por Cleverdon [74]. Elas vêm sido usados desde então inclusive em sistemas de recomendação [41][48].

O cálculo é realizado com o auxílio de uma tabela 2x2, como mostrado na Tabela 3. O conjunto de itens deve ser separado em duas classes: relevantes e não relevantes (comprados ou não comprados, por exemplo). É também necessário separar o conjunto de itens em recomendados e não-recomendados.

Tabela 3 – Categorização de itens para revocação e precisão

	Recomendado	Não-recomendado	Total
Relevante	N_{RS}	N_{RN}	N_R
Não Relevante	N_{IS}	N_{IN}	N_I
Total	N_S	N_N	N

A Precisão representa a probabilidade de um item selecionado ser relevante e é definida como a razão entre os itens relevantes selecionados e o número de itens selecionados, conforme a Eq. 12:

$$precisão(\Pr) = \frac{N_{RS}}{N_S} \qquad 12$$

A Revocação representa a probabilidade de um item relevante ser selecionado e é definida como a razão entre os itens relevantes selecionados e o total de itens disponíveis, conforme a Eq. 13:

$$revocação(\mathrm{Re}) = \frac{N_{RS}}{N_R} \qquad 13$$

Precisão e revocação dependem da separação de itens relevantes dos não-relevantes. Todavia, a noção de relevância em sistemas de recomendação é subjetiva, pois cada usuário avalia um item de acordo com sua própria visão e necessidades pontuais.

Outro problema da métrica é que o cálculo de revocação pode ser impossível de ser realizado, uma vez que é necessário saber se cada item é relevante para cada usuário. A avaliação, por cada pessoa, de todos os itens disponíveis para recomendação pode ser impossível em uma base de dados real. Para contornar este problema, aproximações foram criadas para avaliar sistemas de recomendação. Sawar [48] avaliou seus algoritmos dividindo o banco de dados de avaliações do usuário em um conjunto de treinamento e de teste. Treina-se o algoritmo de recomendação com o conjunto de dados de treinamento e se gera a lista top-N de itens recomendados ao usuário. A revocação é a porcentagem dos itens relevantes do conjunto de teste que estão presentes nos itens recomendados no top-N. Como o número de itens que cada usuário avalia é muito menor do que o total de itens da base de dados (por esta ser esparsa), o número de itens relevantes no conjunto de teste pode ser uma pequena fração do número de itens relevantes na base de dados. A precisão também é calculada dessa forma: os itens relevantes são selecionados de um pequeno conjunto de itens avaliados e os itens previstos são selecionados de um conjunto muito maior de itens.

Precisão e revocação devem ser empregadas em conjunto para avaliar o desempenho de um algoritmo. Os valores de precisão e de revocação são inversamente relacionados [74] e dependem do tamanho da lista recomendada ao usuário. Quanto mais itens são recomendados, maior a revocação e menor a precisão. Para contornar essa característica, foi criada a métrica F1 (Eq. 14), que combina precisão e revocação em um único número e tem sido utilizada para avaliar sistemas de recomendação [29]

$$F1 = \frac{2 * Pr * Re}{Pr + Re} \qquad 14$$

3.5.1.4
Sensitividade ROC

A métrica baseada em curva ROC (*relative operating characteristic*) oferece uma alternativa ao método de precisão e revocação, tendo sido introduzida na comunidade de Recuperação de Informação por Swets [75]. A curva ROC é utilizada para medir o quanto um valor produzido por um sistema é capaz de distinguir os elementos relevantes dos não relevantes.

O sistema medido tem como saída uma variável de relevância associada a cada elemento. Dessa saída são construídas duas curvas de distribuição, uma para os valores obtidos para os elementos relevantes e outra para os elementos não relevantes, mostradas na Figura 9.

No momento de selecionar os elementos relevantes, o sistema usa um limiar t. Os elementos que ultrapassem esse limiar serão considerados relevantes e, portanto, selecionados. Caso contrário, esses elementos serão considerados irrelevantes e então rejeitados. Para cada valor escolhido para o limiar t é possível calcular a cobertura (proporção dos elementos relevantes que são selecionados) e o ruído (proporção dos elementos não relevantes que são selecionados);

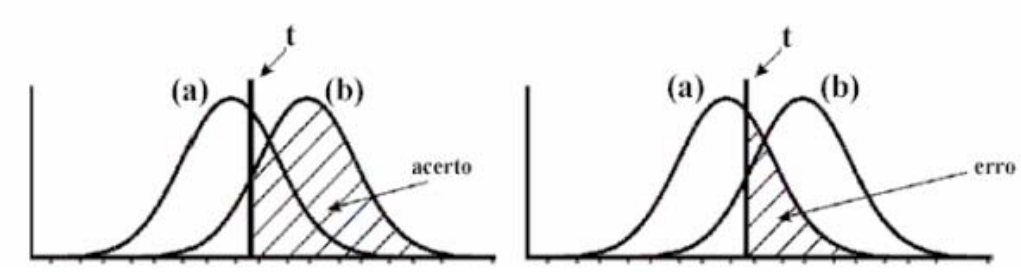


Figura 9 – Curvas de distribuição dos elementos relevantes (b) e não relevantes (a). O valor de limiar (t) determina a taxa de precisão (acerto) eo o ruído (erro) do sistema. [76]

A distribuição a esquerda representa a probabilidade de que o sistema vai prever um nível dado de relevância (o eixo x) para um item que em realidade não possui informação relevante. A distribuição na direita da indica a mesma

distribuição probabilidade para itens que são relevantes. Intuitivamente, pode-se ver que quanto mais separadas essas duas distribuições se encontram, melhor o sistema é para diferenciar itens relevantes dos não-relevantes.

A cobertura será a razão do número de acertos pelo número de elementos relevantes e o ruído a razão do número de erros pelo número de elementos não relevantes. A curva ROC é a curva obtida quando desenhados em um plano cartesiano os valores da cobertura (ordenada) versus o ruído (abscissa) para diferentes valores do limiar t. A Figura 10 mostra um gráfico com uma curva ROC e alguns pontos para diferentes valores de t.

Como a área sob a curva ROC é uma métrica de seleção, ela é adequada para medir a capacidade do sistema de selecionar os itens relevantes para o usuário. Descobrir quais são os itens relevantes ainda dependerá de um limiar que pode ser calculado para o sistema ou para cada usuário, mas o valor da área sob a curva ROC é capaz de diferenciar a capacidade de seleção de dois algoritmos de filtragem, independente do limiar.

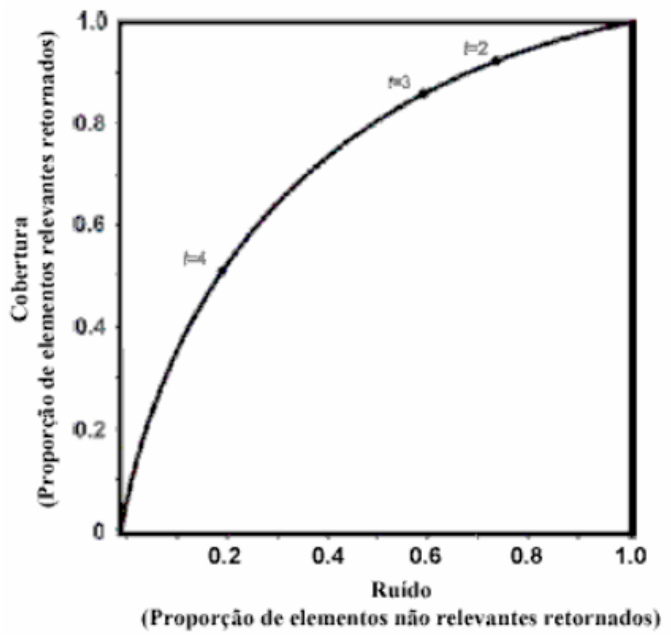


Figura 10 - Exemplo de uma curva ROC.

3.6
Bancos de Dados para Filtragem Colaborativa

A avaliação de algoritmos de recomendação requer testes em bases de dados, o que levanta a necessidade de uma análise mais aprofundada sobre as condições de criação destas bases de dados e suas propriedades.

Uma importante escolha de pesquisadores quanto a base de dados é se ela será sintética ou real. A base sintética é criada por algoritmos, de forma a seguir características específicas para testes de recomendadores em domínios específicos, enquanto a base real é criada pela interação real de usuários com itens reais a serem recomendados.

Bases de dados sintéticas possuem a vantagem de se adequar ao domínio criado e permitir testes de algoritmos de recomendação por falhas, mas por outro lado não podem modelar com precisão total a ação de usuários reais em bases reais. Em seu estudo sobre filtragem colaborativa, [77] usou uma base sintética e notou que os resultados do algoritmo de recomendação criado por ele "não eram justos para com os outros algoritmos" porque a base de dados se adequava a seu recomendador perfeitamente. O uso de bases sintéticas para tirar conclusões comparativas de performance entre recomendadores é, portanto, arriscado.

Por outro lado, existem hoje novas técnicas avançadas para modelar o interesse do usuário e gerar bases de dados sintéticas, capazes de ajustar algoritmos de recomendação. A pesquisa na criação de novas bases de dados sintéticas pode levar a geração de algoritmos de recomendação mais precisos. [78].

A base de dados mais utilizada para avaliação de recomendadores foi a EachMovie (HTTP://research.compaq.com/SRC/eachmovie/). Esta extensa base de dados possui 2.8 milhões de avaliações de 70 mil usuários, tendo informações como data e dados demográficos dos usuários. A base EachMovie foi fonte da base de dados MovieLens (HTTP://www.movielens.org) e foi usados em dezenas de projetos de pesquisa em aprendizado de máquina para estudar novas e melhores formas de predizer o comportamento de consumidores.

Mais recentemente, vários pesquisadores estão usando a base de dados "Jester" foi coletada do site de recomendação de piadas Jester [20].

A maioria das publicações relacionadas com filtros colaborativos em recomendação utiliza uma das três bases de dados acima descritas. As poucas outras bases de dados usadas não foram disponibilizadas para verificação. A falta de variedade de bases de dados para avaliação de filtros colaborativos (particularmente com números significativos de avaliações) ainda é hoje um dos maiores desafios dessa área de pesquisa. A maioria dos pesquisadores não possuem os recursos para coletar dados suficientes em bases de dados para comprovar suas hipóteses de pesquisa.

Em 2006 a empresa Netflix de aluguel de filmes criou o NetFlix Prize, uma competição para que os pesquisadores da área de recomendação usassem um banco de dados oferecido pela empresa para criar o melhor recomendador possível. O Netflix disponibilizou dados de treinamento com 100.480.507 avaliações feitas por 480.189 usuários sobre 17.770 filmes. Esta base de dados foi disponibilizada até 2009 quando a empresa Netflix recebeu um processo por divulgar informações confidenciais de seus usuários e hoje se encontra inacessível para uso.

3.7
Resumo Filtros Colaborativos

A tabela 4 apresenta uma visão geral dos filtros colaborativos.

Tabela 4 – Resumo de Técnicas de Filtragem Colaborativa com suas vantagens e desvantagens.

Categorias de FC	Técnicas Representativas	Vantagens Principais	Desvantagens Principais
Baseados em memória	FC baseados em vizinhança (baseados em item/baseado em usuário com correlação Paerson/cosseno de vetor) Baseados em Item/Baseado em Usuário top-N recomendações	Fáceis de usar Novas informações podem ser adicionadas facilmente e incrementalmente Desnecessário considerar o conteúdo dos itens recomendados Boa escalabilidade com itens co-avaliados	Dependem de votos de usuários Desempenho decresce quando dados são esparsos Não podem recomendar para novos usuários e itens Têm limitada escalabilidade para grandes bancos de dados
Baseados em Modelos	Redes de Crença Bayesiana Clusterização Baseados em MDP Semântica Latente Analise de Fatores Esparsos Técnicas de Redução de Dimensionalidade em FC (SVD,PCA)	Atuam melhor em problemas de esparsabilidade, escalabilidade entre outros Melhoram desempenho de predição Oferecem recomendações intuitivas e racionais	Produção de modelos pesada Trade-off entre performance e escala Perdem informação na redução da dimensionalidade
Híbridos	Baseados em Conteúdo (Fab) Impulsionados por Conteúdo Combinação de memória e modelo (Diagnostico de Personalidade)	Resolvem limitações entre FC e recomendadores baseados em conteúdo Melhoram performance de predição Resolvem problemas de esparsabilidade e Ovelha cinza.	Aumento de complexidade e dificuldades de implementação Necessitam de informação externa.