

Sistemas de Recomendação com Filtros Colaborativos: um Estudo Comparativo

Matheus França Pinheiro¹, Nádia Félix Felipe da Silva²

¹Instituto de Informática (INF) – Universidade Federal de Goiás (UFG)
Alameda Palmeiras, Quadra D, Campus Samambaia CEP 74690-900 - Goiânia - GO

mathfrance@hotmail.com , nadia@inf.ufg.br

Abstract. *Recommendation Systems can be found every day in different ways and for different purposes, they have been highly relevant in dealing with the information overload on the Internet. Collaborative Filtering is one of the most widely used and successful techniques in Recommendation Systems, however, authors who discuss the topic do not generally approach the content from a comparative point of view, validating their results in particular data sets. Thus, a study is needed, with a representative data set in order to map the advantages and disadvantages of each method. In this context, this paper seeks to compare the main methods of recommendation systems with collaborative filters, in order to fill the current gap regarding these direct comparisons.*

Keywords: Recommendation Systems, Collaborative Filtering, Comparative Study, K-Nearest Neighbors, Matrix Factorization, Non-Negative Matrix Factorization, Singular Value Decomposition.

Resumo. *Sistemas de Recomendação podem ser encontrados todos os dias de diferentes formas e com diferentes finalidades, eles têm sido altamente relevantes para lidar com a sobrecarga de informações que há na Internet. A Filtragem Colaborativa é uma das técnicas mais usadas e bem sucedidas em Sistemas de Recomendação, porém os autores que discorrem sobre o tema, geralmente não abordam o conteúdo do ponto de vista comparativo, validando seus resultados em conjuntos de dados particulares. Com isso, faz-se necessário um estudo, com um conjunto de dados representativos a fim de mapear as vantagens e desvantagens de cada método. Neste contexto, este trabalho busca comparar os principais métodos de sistemas de recomendação com filtros colaborativos, com o objetivo de preencher a atual lacuna existente a respeito dessas comparações diretas.*

Palavras-chave: Sistemas de Recomendação, Filtragem Colaborativa, estudo comparativo, K-Nearest Neighbors, Matrix Factorization, Non-Negative Matrix Factorization, Singular Value Decomposition.

1. Introdução

Sistemas de Recomendação podem ser encontrados todos os dias de diferentes formas e com diferentes finalidades, eles auxiliam e ampliam um processo natural de recomendar itens, sejam eles filmes, artistas, músicas, produtos, serviços, entre outros [Resnick and Hal R. Varian 1997].

Os Sistemas de Recomendação têm sido altamente relevantes para lidar com a sobrecarga de informações que há na Internet, onde é sugerido automaticamente ao usuário coisas que ele provavelmente se interesse baseado em uma análise de padrões dos usuários com perfis semelhantes, pois normalmente, são as próprias pessoas que provêm os insumos para os Sistemas de Recomendação [Resnick and Hal R. Varian 1997, Camacho and Alves-Souza 2018].

Muitas empresas voltaram sua atenção aos Sistemas de Recomendação, afim de prover pacotes de recomendações personalizadas para cada tipo de perfil de seus usuários. Por conta dessas recomendações personalizadas, foi adicionado um novo patamar à experiência do usuário em como se recomendar serviços e/ou produtos como é o caso de empresas líderes de mercado como Amazon¹, YouTube² e Netflix³ que se tornaram referência em recomendações, com listas de produtos, vídeos e filmes/séries, respectivamente, personalizadas para cada perfil de usuário. [Bell et al. 2009].

Os Sistemas de Recomendação coletam informações das preferências dos usuários de maneira explícita ou implícita [Camacho and Alves-Souza 2018]. De maneira explícita, é comumente solicitado ao usuário que forneça avaliações, por exemplo, nota para um filme, uma série, um livro, ou estrelas por serviços prestados, etc. Porém, o mesmo valor de uma avaliação pode ter diferentes pesos para usuários diferentes, devido às noções de satisfação serem individuais. Por exemplo, uma pessoa pode dar cinco estrelas a um atendimento recebido, pelo fato de somente ter concluído ele sem maiores problemas, enquanto outra só as dará quando esse atendimento tiver sido excepcional, com isso há uma dificuldade nas comparações entre as avaliações dos usuários.

Outra maneira de coleta de informações sobre a satisfação do usuário é a implícita, normalmente monitorando os comportamentos dos usuários (*clicks* em *links*, tempo em que ficou assistindo um vídeo, filme ou série, sites visitados, produtos acessados, etc...). Essa coleta implícita fornece muito mais dados para se trabalhar e são mais difíceis de serem fraudados. Além disso, também podem ser usadas informações demográficas dos usuários (Gênero, idade, nacionalidade, localização atual) e informações das redes sociais (número de seguidores, curtidas, comentários, *hashtags*), para fornecer recomendações, com base nas informações coletadas, para diferentes nichos demográficos, combinando as classificações dos usuários nesses nichos [Bobadilla et al. 2013].

Na literatura, Sistemas de Recomendação são divididos em três diferentes categorias: (i) Filtragem Colaborativa (*Collaborative Filtering*), (ii) Baseada em Conteúdo (*Content-based*) e (iii) Sistemas de Recomendação Híbridos (*Hybrid Recommender Systems*) [Bobadilla et al. 2013]. Recomendações em Filtragem Colaborativa são feitas com base em avaliações realizadas previamente por um grupo de usuários que compartilham as preferências mais similares com o usuário que receberá a recomendação. A abordagem Baseada em Conteúdo analisa o conteúdo e as características dos itens que o usuário avaliou no passado, cria um perfil de interesses para esse usuário e realiza recomendações baseadas unicamente nesse perfil criado. Por fim, a abordagem híbrida combina os métodos de Filtragem Colaborativa com a Baseada em Conteúdo ou com diferentes variações de outros métodos colaborativos [Lü et al. 2012].

¹<http://amazon.com.br/>

²<http://youtube.com/>

³<https://www.netflix.com>

Os autores que propõem Sistemas de Recomendação com filtros colaborativos geralmente não abordam o conteúdo do ponto de vista comparativo, validando seus resultados em conjuntos de dados particulares. Faz-se necessário um estudo aprofundado com um conjunto de dados representativo, a fim de mapear as vantagens e desvantagens de cada método. Neste contexto, este trabalho busca comparar os principais métodos de sistemas de recomendação com Filtragem Colaborativa baseados em memória, com o uso de variações de similaridade da abordagem KNN (*K-Nearest Neighbors*) e baseados em modelo com o uso da abordagem SVD (*Singular Value Decomposition*) e NMF (*Non Negative Matrix Factorization*), de acordo com as métricas de avaliação precisão, MAE (*Mean Absolute Error*) e RMSE (*Root Mean Squared Error*), mantendo um grupo controle com recomendações geradas aleatoriamente, para se verificar o real ganho, ou não, das técnicas escolhidas para comparação. Com a análise comparativa foi possível ter um visão geral do desempenho das técnicas e as vantagens de umas sobre outras, trazendo assim uma base comparativa das técnicas entre si.

O restante deste artigo está organizado da seguinte maneira. Na seção 2 é dada uma visão geral dos trabalhos relacionados existentes. Na seção 3 é explicado o funcionamento dos Sistemas de Recomendação com Filtros Colaborativos e suas abordagens baseadas em memória e em modelo, além das métricas que serão utilizadas posteriormente para avaliar as recomendações. Em seguida, na seção 4 é apresentada a Avaliação Experimental descrevendo o conjunto de dados, as ferramentas utilizadas para viabilizar os experimentos, a configuração experimental e as técnicas que foram escolhidas para avaliação. Na seção 5 são apresentados os resultados. Por fim é apresentada a conclusão, seguido pelas recomendações para trabalhos futuros.

2. Trabalhos Relacionados

Tendo em vista que este trabalho busca comparar os principais métodos de sistemas de recomendação com Filtragem Colaborativa, esta seção tem como objetivo apresentar os principais trabalhos relacionados a este contexto.

Em [Chen et al. 2018] é feita uma revisão dos trabalhos existentes sobre Filtragem Colaborativa e abordagens híbridas de Sistemas de Recomendação, contribuindo com uma sumarização de variados métodos e técnicas e discorrendo sobre o funcionamento das principais, além de apresentar algumas métricas para avaliação e validação das mesmas, o que serviu como uma das bases de pesquisas desde artigo. Porém o *paper* careceu de uma comparação direta entre os métodos apontados.

[Bell et al. 2009] contribui com uma análise e explicação detalhada da abordagem de Sistemas de Recomendação baseados em Modelo, explicando as técnicas de *Matrix Factorization*, o que se relaciona com algumas das técnicas que serão usadas mais a frente. Além disso [Bell et al. 2009] traz como caso de uso a "*Netflix Prize*", que foi uma competição aberta afim de descobrir qual era o melhor algoritmo de Filtragem Colaborativa para prever avaliações (*rating*) de usuários para filmes. Porém não há uma análise comparativa com outras abordagens, por exemplo baseada em Memória, expondo as vantagens e/ou desvantagens entre ambas.

[Bobadilla et al. 2013] realizam uma pesquisa robusta sobre Sistemas de Recomendação em que os autores descrevem os principais métodos, algoritmos e modelos usados para prover recomendações, além de apresentar as formas de avaliação da

qualidade das predições e recomendações de Sistemas de Recomendação, do qual serviu como uma das bases de pesquisas desde artigo.

O trabalho de [Yang et al. 2016] foca em Sistemas de Recomendação baseados em Filtros Colaborativos para aplicativos de *Internet* móvel, classificando os algoritmos e expondo os principais procedimentos da Filtragem Colaborativa, além de apresentar dois casos de estudos analisando e comparado o algoritmos com dados reais. O que serviu como exemplificação de comparação de Sistemas de Recomendação.

[Badrul Sarwar 2001] aplica diferentes abordagens de algoritmos baseados em item existentes em Sistemas de Recomendação com Filtros Colaborativos afim de elencar algumas vantagens se comparados com as abordagens baseadas em usuário. O *paper* se limita a realizar esta análise direcionada a uma parte dos Sistemas de Recomendação com Filtros Colaborativos e realiza uma comparação detalhada, ao qual serviu como fonte para esse artigo ao abordar algoritmos baseados em item na seção 3.1.1.

No trabalho de [Hernando et al. 2016] é apresentada a *Non-Negative Matrix Factorization* (NMF), uma técnica de Sistemas de Recomendação com Filtragem Colaborativa baseados em Modelo, explicando como funciona e suas vantagens sobre outras técnicas conhecidas e serviu como fonte para esse artigo ao abordar *Non-Negative Matrix Factorization* (NMF) na seção 3.2.2.

3. Sistemas de Recomendação com Filtros Colaborativos

De modo geral, os Sistemas de Recomendação seguem um fluxo de atividades, conforme Figura 1, inicialmente é realizada a coleta dos dados dos usuários por dispositivos que tenham acesso a Internet e armazenados nos servidores. Na sequência a representação dos dados pode ser expressa, por exemplo, conforme a avaliação feita do usuário sobre o item, ou o comportamento de avaliar ou não um item. Posteriormente a estratégia de recomendação escolhida é treinada e modelada à partir dos dados, podendo ser feita com o uso de Filtros Colaborativos Baseados em vizinhos ou Filtros Colaborativos Baseados em Modelo. Na abordagem baseada em vizinhos, estes são procurados através do cálculo das similaridades entre os usuários ou entre os itens, já na abordagem baseada em modelo pode ser realizada a redução de dimensionalidade ou matriz de decomposição, chegando a modelos de agrupamento, SVD (*Singular Value Decomposition*), MF (*Matrix Factorization*) ou NMF (*Non-Negative Matrix Factorization*). Em seguida, o algoritmo escolhido prediz os *scores* sem avaliação do usuário, avalia os itens e ordena uma lista de *Top-N* itens, para, por fim, ser recomendados aos usuário.

Os Sistemas de Recomendação com Filtragem Colaborativa podem ser divididos em dois tipos: Baseado em Memória e Baseado em Modelo [Chen et al. 2018]. Tais métodos são descritos nas subseções a seguir.

3.1. Sistemas de Recomendação com Filtragem Colaborativa baseados em Memória

Em Sistemas de Recomendação com Filtragem Colaborativa baseados em Memória, o algoritmo obtém os relacionamentos semelhantes de usuários ou itens de acordo com a matriz de avaliação (*rating*), posteriormente seleciona os itens mais bem avaliados pelos usuários que possuem características similares, e por fim realiza a recomendação desses itens para o usuário alvo [Yang et al. 2016]. As avaliações de itens feitas por usuários são usadas diretamente para prever avaliações desconhecidas para novos itens.

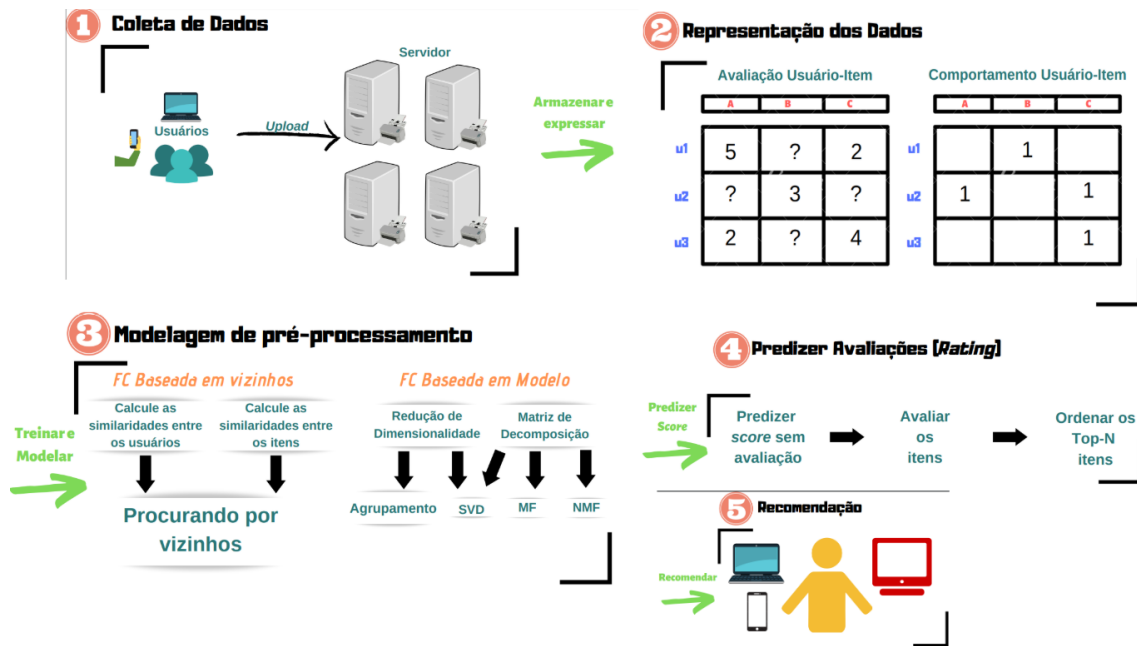


Figura 1. **Framework de Sistemas de Recomendação baseados em Filtragem Colaborativa - Adaptada de [Chen et al. 2018]**

As recomendações baseadas em memória podem ser subdivididas em duas formas: Filtragem Colaborativa baseada em usuário e baseada em item [Ricci F. 2010], a diferença entre ambas é demonstrada na Figura 2, onde caso o processo de Filtragem Colaborativa Baseada em memória seja baseada em item, a primeira etapa do processo irá calcular as similaridades existentes entre os itens vizinhos do item ativo (escolhido para comparação), e caso seja baseada em usuário esta primeira etapa do processo irá calcular as similaridades existentes entre os usuários vizinhos do usuário ativo. No caso da Figura 2, a abordagem escolhida foi a baseada em usuário, que após o cálculo de similaridade, encontra seus vizinhos mais próximos, ou seja, os mais similares, prediz as avaliações, ordena os itens candidatos a serem recomendados, baseado nas avaliações e por fim recomenda os itens para o usuário.

3.1.1. Algoritmo Baseado em Usuário

Essa abordagem leva em consideração que usuários com histórico de avaliação (*rating*) semelhantes provavelmente possuem interesses similares, então pode-se prever as avaliações que não foram feitas de certos itens por um determinado usuário, tendo como base as avaliações feitas por usuários similares. Esse processo pode ser dividido em três partes, inicialmente são calculadas as similaridades entre o usuário alvo e os demais, então os mais semelhantes (chamados de vizinhos) são selecionados e por fim são preditas as avaliações do usuário alvo e os resultados da recomendação são gerados [Alexander Tuzhilin 2005].

- **Calculando as similaridades entre usuários:** As avaliações (*rating*) de um usuário u são representadas por um vetor de avaliações $a_u = [a_{u1}, a_{u2}, \dots, a_{un}]$, as similaridades entre dois usuários é obtida comparando-se seus vetores de avaliações, as duas formas clássicas de se realizar essa comparação são por meio

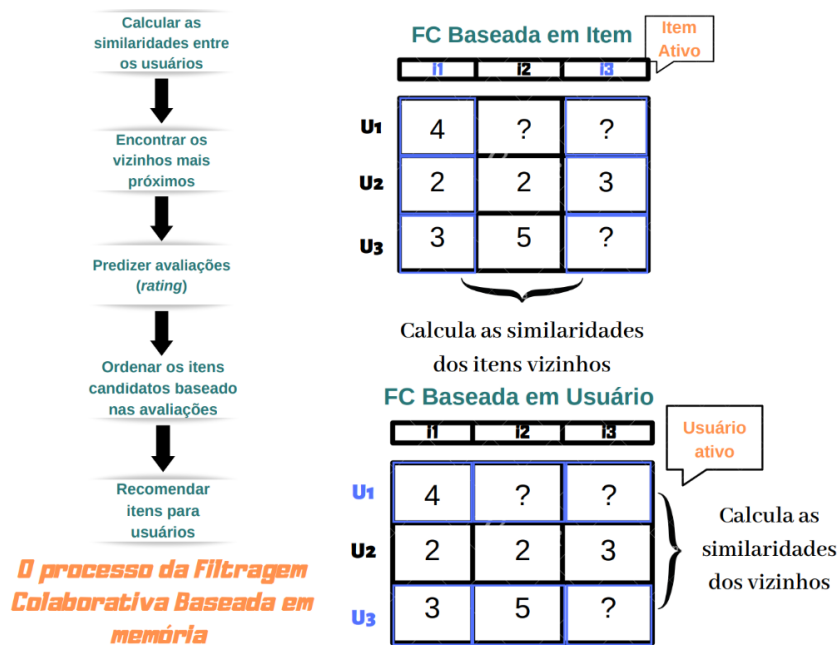


Figura 2. A lógica da Filtragem Colaborativa baseada em usuário e baseada em item - Adaptada de [Chen et al. 2018]

da Similaridade Cosseno (*Cosine similarity*) e a Correlação de Pearson (*Pearson correlation coefficient*) [Koohi and Kiani 2017], ambas detalhadas a seguir.

1. *Similaridade Cosseno*: As avaliações podem ser indicadas por um vetor n -dimensional e as similaridades entre usuários são obtidas através do ângulo do vetor de avaliação do usuário. Em geral, quanto menor for o ângulo maior é a similaridade [Ricci F. 2010].

$$sim_{uv} = \cos(\vec{a}_u, \vec{a}_v) = \frac{\vec{a}_u \cdot \vec{a}_v}{\|\vec{a}_u\|_2 \times \|\vec{a}_v\|_2} = \frac{\sum_{i \in I_{uv}} a_{ui} \cdot a_{vi}}{\sqrt{\sum_{i \in I_u} a_{ui}^2} \sqrt{\sum_{i \in I_v} a_{vi}^2}} \quad (1)$$

Onde, sim_{uv} representa as similaridades entre os usuários u e v , \vec{a}_u, \vec{a}_v representam os vetores de avaliações de u e v , respectivamente, $\|\vec{a}_u\|_2$ e $\|\vec{a}_v\|_2$ representam a norma 2 de u e v , respectivamente e a_{ui} e a_{vi} representa as avaliações do u e v sobre o item i , respectivamente. I_u e I_v representam o conjunto de itens avaliados pelos usuários u e v , respectivamente e I_{uv} representa o conjunto de itens em comum que foram avaliados pelos usuários u e v .

2. *Correlação de Pearson*:

$$sim_{uv} = \frac{\sum_{i \in I_{uv}} (a_{ui} - \bar{a}_u)(a_{vi} - \bar{a}_v)}{\sqrt{\sum_{i \in I_{uv}} (a_{ui} - \bar{a}_u)^2} \sqrt{\sum_{i \in I_{uv}} (a_{vi} - \bar{a}_v)^2}} \quad (2)$$

Onde, \bar{a}_u, \bar{a}_v representa a média de avaliações dos usuários u e v , respectivamente.

- **Encontrando os vizinhos mais próximos**: Os dois métodos mais usados são o KNN (*K-Nearest Neighbors*) e *Threshold*. O KNN seleciona os K vizinhos mais próximos (com a maior similaridade) do usuário ativo u . Para o método *Threshold*

é necessário definir inicialmente um limite δ mínimo. Quando as similaridades entre o usuário v e o usuário ativo u são maiores que δ , o usuário v é selecionado como um vizinho próximo.

- **Predizendo as avaliações:** Existem duas maneiras principais de realizar recomendações para um usuário: Predizendo sua avaliação (*rating*) ou através de uma lista de *Top-N* recomendações. Ambos precisam das avaliações sobre um item i de usuários mais semelhantes para prever a avaliação de um usuário ativo u . A avaliação predita é calculada a seguir [Moradi and Ahmadian 2015]:

$$\hat{a}_{ui} = \bar{a}_u + \frac{\sum_{v \in N_u} sim_{uv}(a_{vi} - \bar{a}_v)}{\sum_{v \in N_u} |sim_{uv}|} \quad (3)$$

Onde, N_u representa o conjunto de vizinhos similares ao usuário u .

3.1.2. Algoritmo Baseado em Item

O Algoritmo Baseado em Item também é dividido em três partes, semelhantes ao Algoritmo Baseado em Usuário, onde inicialmente são calculadas as similaridades entre os itens, depois selecionados os itens vizinhos mais semelhantes e por fim são preditas as avaliações de um item alvo de acordo com os itens vizinhos e a lista recomendação é gerada [Chen et al. 2018].

- **Calculando as similaridades entre Itens:** As duas formas clássicas de realizar o cálculo são ajustadas.

1. *Similaridade Cosseno Ajustada* [Koohi and Kiani 2017]:

$$sim_{ij} = \frac{\sum_{u \in U_{ij}} (a_{ui} - \bar{a}_u)(a_{uj} - \bar{a}_u)}{\sqrt{\sum_{u \in U_i} (a_{ui} - \bar{a}_u)^2} \sqrt{\sum_{u \in U_j} (a_{uj} - \bar{a}_u)^2}} \quad (4)$$

Onde, sim_{ij} representa as similaridades entre os itens i e j . U_i e U_j o conjunto de usuários que avaliaram os itens i e j , respectivamente e U_{ij} representa os usuários que avaliaram os dois itens i e j .

2. *Pearson correlation coefficient* ajustado [Salter and Antonopoulos 2006]:

$$sim_{ij} = \frac{\sum_{u \in U_{ij}} (a_{ui} - \bar{a}_i)(a_{uj} - \bar{a}_j)}{\sqrt{\sum_{u \in U_{ij}} (a_{ui} - \bar{a}_i)^2} \sqrt{\sum_{u \in U_{ij}} (a_{uj} - \bar{a}_j)^2}} \quad (5)$$

Onde, \bar{a}_i, \bar{a}_j representa a média de avaliações dos itens i e j , respectivamente.

- **Encontrando os vizinhos mais próximos:** Similar ao Algoritmo Baseado em Usuário, utilizando os dois métodos: KNN (*K-Nearest Neighbors*) e *Threshold*.
- **Predizendo as avaliações** [Badrul Sarwar 2001]:

$$\hat{a}_{ui} = \bar{a}_i + \frac{\sum_{j \in N_i} sim_{ij} \times (a_{uj} - \bar{a}_j)}{\sum_{j \in N_i} |sim_{ij}|} \quad (6)$$

Onde, N_i é o conjunto de vizinhos similares ao item i .

3.2. Sistemas de Recomendação com Filtragem Colaborativa baseados em Modelo

Na prática o Algoritmo de Filtragem Colaborativa Baseada em Memória não é muito adequado quando se lida com um volume muito grande de usuários e itens, por isso o Baseado em Modelo surge como uma alternativa [Hernando et al. 2016]. Este algoritmo requer uma fase de aprendizado prévio para descobrir os parâmetros do modelo ideal antes de fazer uma recomendação. Uma vez essa fase tenha sido concluída, o algoritmo poderá realizar predições de avaliações dos usuários. O LFM (*Latent Factor Model*) é bastante competitivo e amplamente adotado para a implantação de Sistemas de Recomendação, uma vez que tal técnica fatoriza a matriz de avaliações (*rating*) usuário-item em duas sub matrizes: De características de usuários e a de itens. SVD (*Singular Value Decomposition*) [Zhou et al. 2015], MF (*Matrix Factorization*) [Sherif and Zhang 2017] e NMF (*Non-Negative Matrix Factorization*) [HUANG Bo 2016] são os métodos de recomendação mais utilizados, dos quais todos se aproveitam do LFM.

3.2.1. MF (*Matrix Factorization*):

O processo de recomendação desse modelo é mostrado na Figura 3, onde inicialmente a partir da matriz de classificação de usuário-item constrói-se o *Latent Feature Model*. Em seguida se obtém a função objetivo que será otimizada. Posteriormente a função objetivo é resolvida atualizando os valores das matrizes de características (*Feature*) do usuário e do item, obtendo assim as matrizes de *Latent Feature*. Na sequência são preditas as avaliações, ordenado os itens de acordo com essas avaliações preditas e recomendado os itens para o usuário.

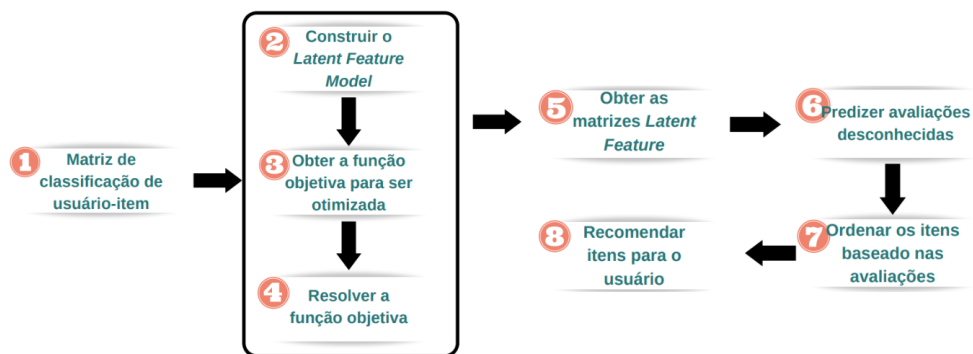


Figura 3. O processo de recomendação de Sistemas de Recomendação baseado em *Matrix Factorization* - Adaptada de [Chen et al. 2018]

- **Construir o *Latent Feature Model*:** O modelo da matriz de fatoração é descrito como em [Lü et al. 2012]:

$$A \approx PQ^T \quad (7)$$

Para a matriz de avaliação (*rating*) usuário-item A de $m \times n$, o modelo MF representa aproximadamente A como uma forma do produto da matriz de características dos usuários P de $m \times k$ e da matriz de características dos itens Q de

$n \times k$ de acordo com a técnica MF. Onde, m e n são o número de usuários e itens, respectivamente, e k é o número de características latentes. Este modelo é exemplificado na Figura 4, com o uso de números reais, nota-se que o produto da matriz de características do usuário P com a matriz de características do item Q resulta na matriz de avaliações preditas \hat{A} , que por sua vez podem prever avaliações desconhecidas na matriz de avaliação A , como por exemplo, A_{13} e \hat{A}_{13} ou A_{22} e \hat{A}_{22} :

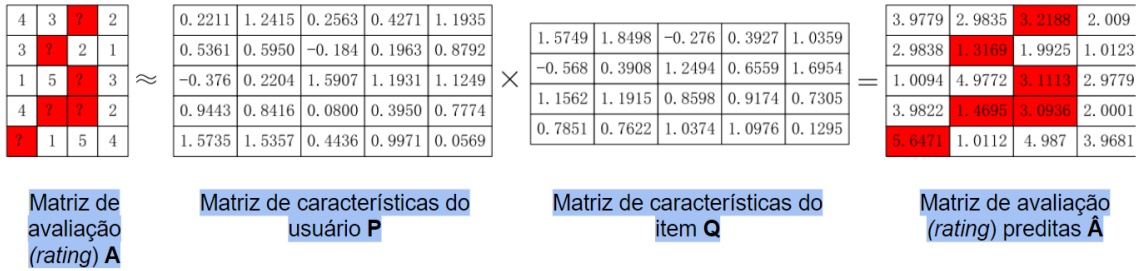


Figura 4. Um exemplo de *Matrix Factorization* em recomendação de filmes - Adaptada de [Chen et al. 2018]

- **Obter a função objetivo:** O modelo MF geralmente busca minimizar o desvio entre a decomposição da matriz aproximada e da matriz de avaliação (*rating*) original de usuário-item. Portanto, o modelo é treinado utilizando o método *Gradient Descent* [Bottou 2010] para alcançar a melhor solução. O objetivo da função é descrito abaixo [Lü et al. 2012]:

$$L = \min \|A - \hat{a}\| = \min \left(\sum (a_{ui} - p_{uk}q_{ki})^2 + \lambda_p \|p_{uk}\|^2 + \lambda_q \|q_{ki}\|^2 \right) \quad (8)$$

Onde, a_{ui} denota as avaliações do usuário u sobre o item i na matriz original, p_{uk} e q_{ki} denota a k -ésima característica do usuário u e do item i em P e Q^T , respectivamente. λ_p e λ_q são termos de parâmetro regularizadores para evitar sobre-ajuste (*Overfitting*) [Hawkins 2004].

- **Atualizar os valores das matrizes de características P e Q :** SGD (*Stochastic Gradient Descent*) [Bottou 2010], [Sherif and Zhang 2017] e ALS (*Alternating Least Squares*) são frequentemente usados para resolver os parâmetros da função objetivo. SGD atualiza continuamente os parâmetros desconhecidos de p_{uk} e q_{ki} até a convergência de acordo com a direção do *Gradient Descent* e a função objetivo. A fim de resolver a Equação 8, p_u e q_i são inicializadas aleatoriamente e, em seguida, o erro de previsão entre a avaliação real e a avaliação prevista é calculado da seguinte maneira [Ricci F. 2010]:

$$e_{ui} = a_{ui} - p_{uk}q_{ki} \quad (9)$$

Os valores de p_u e q_i são atualizados para obter os valores aproximados usando o método SGD, da seguinte forma:

$$p_{uk} \leftarrow p_{uk} + \eta (q_{ki} \cdot e_{ui} - \lambda_p p_{uk}) \quad (10)$$

$$q_{ki} \leftarrow q_{ki} + \eta (p_{uk} \cdot e_{ui} - \lambda_q q_{ki}) \quad (11)$$

- **Prever as avaliações desconhecidas de acordo com as Matrizes P e Q :** As avaliações desconhecidas podem ser preditas da seguinte maneira [Lü et al. 2012]:

$$\hat{a}_{ui} = \sum_{k=1}^k p_{uk} q_{ki} \quad (12)$$

Para dados binários, é possível fazer uma previsão usando este método, assumindo que $A = 1$ para todos os pares de usuário-item observados em cenários de *feedback* implícito.

3.2.2. NMF (*Non-Negative Matrix Factorization*):

Também fatoriza a matriz A (usuário-item original) em duas matrizes P e Q com avaliação a , onde P é igual a $|U| \times f$, Q é igual a $f \times |I|$ e $f \ll \min(|U|, |I|)$. O processo de decomposição é realizado sobre a restrição de ser não negativo (*Non-Negative*). Para garantir que P e Q não são negativos, as taxas de aprendizagem são manipuladas da seguinte forma [Luo et al. 2014]:

$$\alpha_{uk} = \frac{p_{uk}}{(PQQ^T)_{uk}}, \alpha_{ki} = \frac{q_{ki}}{(P^T PQ)_{ki}} \quad (13)$$

O processo de atualização é descrito da seguinte forma:

$$p_{uk} \leftarrow p_{uk} \frac{(AQ^T)_{uk}}{(PQQ^T)_{uk}}, q_{ki} \leftarrow q_{ki} \frac{(P^T A)_{uk}}{(P^T PQ)_{uk}} \quad (14)$$

3.2.3. SVD (*Singular Value Decomposition*):

Esparsidade nos dados (*Data sparsity*) e alta dimensionalidade (*high dimensionality*) [Dan Yang 2011] são problemas recorrentes em Sistemas de Recomendação. SVD é uma realização particular dos algoritmos de MF e é uma poderosa técnica para redução de dimensionalidade [Ricci F. 2010]. Uma matriz de avaliação (*rating*) original $A_{m \times n}$ pode ser decomposta em U , S e V de acordo com a expressão SVD a seguir:

$$A_{m \times n} = U_{m \times m} S_{m \times n} V_{n \times n}^T \quad (15)$$

Onde, $U^T U = I_{m \times m}$ e $V^T V = I_{n \times n}$, cada coluna de U é chamada vetor singular esquerdo, S é uma matriz diagonal e os valores diagonais são organizados do maior ao menor, que são chamados valores singulares e cada linha de V^T é chamada vetor singular direito. O valor da diagonal na matriz S é na verdade a raiz quadrada de AA^T ou $A^T A$.

3.3. Métricas de Avaliação

Existem várias métricas para os variados aspectos do desempenho das recomendações realizadas, para a métrica de precisão da avaliação (*rating*), duas métricas são notáveis:

- MAE (*Mean Absolute Error*):

$$\frac{\sum_{(u,i) \in T} |a_{ui} - \hat{a}_{ui}|}{|T|} \quad (16)$$

- RMSE (*Root Mean Squared Error*):

$$\sqrt{\frac{\sum_{(u,i) \in T} (a_{ui} - \hat{a}_{ui})^2}{|T|}} \quad (17)$$

Onde, T denota um conjunto de itens. A diferença entre as métricas é que RMSE leva em consideração o quão distante a predição foi do seu real valor, penalizando mais quando foi longe e penalizando menos quando foi consideravelmente perto. Por ser tratar de métricas de erros os menores valores para MAE ou RMSE correspondem a predições com precisão mais altas [Lü et al. 2012].

Existem outras métricas de precisão, e nem sempre precisão é o mais importante ao realizar previsões e recomendações, por esse motivo existem ainda métricas como cobertura, diversidade e novidade [Chen et al. 2018], porém nos nossos experimentos vamos nos limitar a avaliar com as métricas de MAE e RMSE.

4. Avaliação Experimental

4.1. Conjuntos de dados

O conjunto de dados escolhido para os testes foi a versão reduzida e mais recente (09/2018) do *MovieLens*⁴, este que trata-se da pesquisa do *GroupLens*⁵, no qual coleta e produz conjuntos de dados do *web site MovieLens*⁶, que é um serviço de recomendação de filmes. O conjunto de dados possui cerca de 100.000 (cem mil) avaliações (*rating*), 3.600 (três mil e seiscentas) *tags* aplicadas para 9.000 (nove mil) filmes, feitas por 600 (seiscentos usuários).

O conjunto de dados possui quatro arquivos, porém só dois serão necessários para os testes:

- *Rating*: Com todas as avaliações feitas pelos usuários, possui o identificador do usuário, do filme, a nota dada, e quando a avaliação foi realizada.
- *Movies*: Com todos os filmes do catálogo, possui o identificador do filme, o título seguido com o ano entre parênteses e o gênero, caso o filme possua mais de um estes são separados pelo símbolo *pipe* (`|`).

Os gêneros possuem dezoito classificações e as avaliações (*rating*) são realizadas em uma escala de 5 estrelas, com incremento de meia estrela. Para informações mais detalhadas sobre o conjunto de dados, a documentação está disponível em <http://files.grouplens.org/datasets/movielens/ml-latest-small-README.html>.

4.2. Ferramentais e materiais

Para os os testes foi utilizado a *Anaconda Distribution*⁷, uma plataforma baseada na linguagem *Python/R* para ciência de dados (*Data science*). Em seguida instalada a *Surprise*⁸, uma biblioteca em *Python* para facilitar a construção e análise de Sistemas de Recomendação, pois já possui vários algoritmos e métodos prontos. Por fim foi-se utilizado o *Spyder*⁹, um ambiente de desenvolvimento para edição do código.

⁴<https://grouplens.org/datasets/movielens/>

⁵<https://grouplens.org/>

⁶<http://movielens.org>

⁷<https://docs.anaconda.com/anaconda/>

⁸<http://surpriselib.com/>

⁹<https://docs.spyder-ide.org/index.html>

4.3. Configuração Experimental

Levando em consideração as abordagens descritas acima, foi executada uma bateria de testes simulando todo o processo de uma recomendação de Top 10 filmes para um mesmo usuário x , passando pelo treinamento, para abordagens baseadas em modelo o conjunto de dados foi dividido na proporção 75/25, ou seja 75% dos dados foram usados para treinar o modelo e os outros 25% foram usados para o teste e a validação, por fim foi realizada a recomendação em si. Os algoritmos selecionados para os experimentos foram:

- Aleatoriamente (Controle): Trata-se de um algoritmo que produz recomendações aleatórias, sem a aplicação estratégica, serve como base para identificar a melhora ou não da aplicação de outros algoritmos como estratégia;
- KNN (*K-Nearest Neighbors*) baseado em usuário utilizando *cosine similarity*;
- KNN (*K-Nearest Neighbors*) baseado em usuário utilizando (*Pearson correlation coefficient*);
- KNN (*K-Nearest Neighbors*) baseado em item utilizando *cosine similarity* ajustado;
- KNN (*K-Nearest Neighbors*) baseado em item ajustado utilizando *Pearson correlation coefficient* ajustado;
- SVD (*Singular Value Decomposition*);
- NMF (*Non Negative Matrix Factorization*).

Além das próprias recomendações em si, as métricas de precisão MAE (*Mean Absolute Error*) e RMSE (*Root Mean Squared Error*) foram coletadas para análise. Por falta de recurso computacional não foi possível realizar a coleta do tempo de processamento.

5. Resultados

Tabela 1. Comparação entre algoritmos de recomendação

ALGORITMO	SIMILARIDADE	RMSE	MAE
<i>Random</i>	<i>Random</i>	1,4385	1,1478
User KNN	cosine	0,9961	0,7711
User KNN	pearson	1,0016	0,7762
Item KNN	cosine	0,9995	0,7798
Item KNN	pearson	0,9928	0,7727
SVD	N/A	0,9039	0,6984
NMF	N/A	0,9523	0,7351

De modo geral, como pode ser visto na Tabela 1, qualquer abordagem se sai melhor se comparada com as recomendações aleatórias, o que já demonstra um ganho considerável para qualquer estratégia escolhida. Todas as variações de similaridade de recomendações em KNN, seja *User* ou *Item*, que são baseadas em memória tiveram índices bem parecidos entre si. Por fim as duas abordagens baseadas em modelo SVD e NMF, tiveram melhores índices, com a SVD se saindo melhor entre todas, nos experimentos.

Não foi possível coletar dados de tempo para cada experimento, por falta de recurso computacional, pois por diversas vezes a máquina usada travava durante o processamento, o que impossibilitou uma coleta precisa do tempo de execução. Mas como

observação geral esses travamentos ocorreram apenas nos métodos baseados em memória, ressaltando o custo computacional necessário para as variações dessa abordagem, o que influencia diretamente em outra vantagem das abordagens baseadas em modelo, pois os algoritmos em SVD e NMF conseguiram realizar as recomendações em pouco tempo e sem travar a máquina.

Por fim, em geral as recomendações baseadas em memória embora tenham uma diferença relativamente pequena comparado as recomendações baseadas em modelo, trouxeram algumas recomendações bastante obscuras e desconhecidas em relação ao algoritmo de SVD e NMF, que possivelmente na prática não seriam bem recebidas pelos usuários, onde talvez precisasse passar por um refinamento utilizando *Threshold* para uma quantidade mínima aceitável de avaliações *rating* para ser uma recomendação considerada em uma lista Top N.

6. Conclusão

Levando em conta a avaliação feita, deve-se considerar dentre as opções, as abordagens baseadas em modelo SVD e NMF como melhores opções, tendo fatores como indicador de precisão, tempo e custo computacional mais vantajosos entre todos.

Vale ressaltar que no mundo real, precisão não é tudo em um sistema de recomendação, pois isso depende muito do modelo de negócio em que será usado, métricas como diversidade, novidade, cobertura, entre tantas outras podem ser tão importantes quanto a própria precisão nas recomendações.

Como trabalhos futuros, pretende-se realizar um teste *online* das recomendações feitas pelas abordagens, pois mesmo as recomendações realizadas pelas variações de KNN serem mais obscuras e desconhecidas que as abordagens em SVD e NMF, tais considerações não passam apenas de especulação e só será conclusiva, colocando em prática e testando diretamente cada uma das recomendações com os usuários finais.

Referências

- Alexander Tuzhilin, G. A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge Data Engineering*, 17(03):734–749.
- Badrul Sarwar, George Karypis, J. K. J. R. (2001). Item-based collaborative filtering recommendation algorithms. *Proc. Int. Conf. World Wide Web*, 4:285–295.
- Bell, R., Koren, Y., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(08):30–37.
- Bobadilla, J., Ortega, F., Hernando, A., and Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, 46:109 – 132.
- Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. pages 177–186.
- Camacho, L. A. G. and Alves-Souza, S. N. (2018). Social network data to alleviate cold-start in recommender system: A systematic review. *Information Processing Management*, 54(4):529 – 544.

- Chen, R., Hua, Q., Chang, Y., Wang, B., Zhang, L., and Kong, X. (2018). A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks. *IEEE Access*, 6:64301–64320.
- Dan Yang, Zongming Ma, A. B. (2011). A sparse svd method for high-dimensional data.
- Hawkins, D. M. (2004). The problem of overfitting. *Journal of Chemical Information and Computer Sciences*, 44(1):1–12.
- Hernando, A., Bobadilla, J., and Ortega, F. (2016). A non negative matrix factorization for collaborative filtering recommender systems based on a bayesian probabilistic model. *Knowledge-Based Systems*, 97:188 – 202.
- HUANG Bo, YAN Xuanhui, L. J. (2016). Collaborative filtering recommendation algorithm based on joint nonnegative matrix factorization. 29(6):725–734.
- Koohi, H. and Kiani, K. (2017). A new method to find neighbor users that improves the performance of collaborative filtering. *Expert Systems with Applications*, 83:30 – 39.
- Luo, X., Zhou, M., Xia, Y., and Zhu, Q. (2014). An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Transactions on Industrial Informatics*, 10(2):1273–1284.
- Lü, L., Medo, M., Yeung, C. H., Zhang, Y.-C., Zhang, Z.-K., and Zhou, T. (2012). Recommender systems. *Physics Reports*, 519(1):1 – 49. Recommender Systems.
- Moradi, P. and Ahmadian, S. (2015). A reliability-based recommendation method to improve trust-aware recommender systems. *Expert Systems with Applications*, 42(21):7386 – 7398.
- Resnick, P. and Hal R. Varian, G. E. (1997). Recommender systems. pages 56+.
- Ricci F., Rokach L., S. B. K. P. (2010). Context-aware recommender systems. *Recommender Systems Handbook*, page 217–253.
- Salter, J. and Antonopoulos, N. (2006). Cinemascreen recommender agent: combining collaborative and content-based filtering. *IEEE Intelligent Systems*, 21(1):35–41.
- Sherif, N. and Zhang, G. (2017). Collaborative filtering using probabilistic matrix factorization and a bayesian nonparametric model. pages 391–396.
- Yang, Z., Wu, B., Zheng, K., Wang, X., and Lei, L. (2016). A survey of collaborative filtering-based recommender systems for mobile internet applications. *IEEE Access*, 4:3273–3287.
- Zhou, X., He, J., Huang, G., and Zhang, Y. (2015). Svd-based incremental approaches for recommender systems. *Journal of Computer and System Sciences*, 81(4):717 – 733.