

Uma Investigação Sobre o Efeito da Combinação de Algoritmos em Sistemas de Recomendação

An Investigation on the Effect of Combining Algorithms on Recommendation Systems

Marcos Antônio Almeida Souto Júnior¹  orcid.org/0000-0001-7538-0958

Byron Leite Dantas Bezerra¹  orcid.org/0000-0002-8327-9734

¹ Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife, Brasil.

E-mail do autor principal: Marco Antônio Almeida Souto Júnior maasj@ecom.poli.br

Resumo

Em função do crescente investimento da indústria e do desenvolvimento científico na área, novos sistemas de recomendação surgem permanentemente, buscando incrementar a precisão da indicação de itens aos consumidores e abranger um maior número de contextos de aplicação. Entretanto, a escolha do algoritmo ideal para determinada aplicação nem sempre é uma tarefa trivial. Artigos recentes estudaram formas de realizar esta escolha de forma automática, através de estratégias de meta-aprendizado. Este trabalho busca investigar os efeitos da extensão desse processo de meta-aprendizado do nível de contexto de aplicação para o de usuário. Nesse sentido, foram realizados experimentos a partir de quatro modelos de recomendação, selecionando, a partir de dois critérios distintos, o algoritmo que apresenta melhor desempenho no mapeamento das preferências de cada usuário, verificando o efeito da aplicação personalizada dos algoritmos na performance global do sistema. Resultados positivos foram obtidos quando a seleção de algoritmos era baseada em abordagens de complexidades semelhantes.

Palavras-Chave: Sistemas de Recomendação; Combinação de Algoritmos, Recomendação Personalizada;

Abstract

Due to the increasing investment of industry and the scientific development, new recommendation systems are constantly emerging, seeking to increase the precision of item`s suggestions to consumers and to cover a greater number of application contexts. However, choosing the optimal algorithm for a given application is not always a trivial task. Recent papers have studied ways to accomplish this choice automatically, through meta-learning strategies. This work investigates the effects of the extension of this meta-learning process from the application context level to the user context. Some experiments were performed from four recommendation models, selecting, from two different criteria, the algorithm that presents better performance in the task of mapping the preferences of each user, verifying the effect of the customized application of the algorithms on the systems overall performance. Positive results were achieved when the algorithm selection was based on approaches with similar complexities.

Key-words: Recommendation System, Combination of Algorithms, Personalized Recommendation.

1 INTRODUÇÃO

Os sistemas de recomendação são ferramentas e técnicas de software que provêm sugestões de itens úteis aos usuários. É uma subclasse dos sistemas de filtragem de informação, que procura prever a preferência ou a avaliação que um usuário daria a um determinado item [1].

A fonte de informação dos sistemas de recomendação são as avaliações dos usuários (ratings) que podem ser implícitas ou explícitas. As avaliações explícitas são aquelas realizadas diretamente pelo usuário (numérica, número de estrelas, positivo ou negativo, etc.) enquanto as implícitas são inferidas a partir de sua atividade (cliques, compras, etc.).

PORIYA *et al.* [2] classificam os sistemas de recomendação em quatro tipos: não personalizados, baseados em conteúdo, algoritmos de filtragem colaborativa e os híbridos (combinação entre os tipos anteriores). Os não personalizados são baseados na popularidade ou preferência de grupo, os baseados em conteúdo buscam descobrir os atributos dos itens preferidos pelo usuário enquanto os algoritmos de filtragem colaborativa utilizam as classificações de outros usuários para prever a avaliação de um indivíduo alvo a um item específico [2].

Existem diversos algoritmos distintos para cada tipo de sistema de recomendação. Desta forma, na maioria das vezes não é trivial identificar o modelo mais eficaz para um determinado cenário de aplicação. Experimentalmente, pesquisadores e profissionais lançam mão de métricas como precisão, recall e RMSE para avaliar uma gama de algoritmos elencados e determinar a melhor escolha para um contexto específico [3], o que pode ser custoso em termos de consumo de recursos humanos e computacionais. Uma alternativa é a utilização de meta-aprendizado para selecionar automaticamente o melhor algoritmo para um dado contexto [4].

Meta-aprendizado utiliza técnicas de aprendizado de máquina para obter modelos preditivos capazes de identificar, a partir de características extraídas da base de dados, o algoritmo mais promissor para determinada tarefa [4].

Os algoritmos de recomendação também podem ser combinados, visando obter ganhos de qualidade das recomendações em relação a um modelo único.

O objetivo deste trabalho é investigar os efeitos da utilização de técnicas de combinação automática de algoritmos de recomendação que buscam identificar e aplicar modelos específicos para cada usuário do

sistema. As propostas foram inspiradas no trabalho de Collins *et al.* [3] que será descrito na seção 2.

Os experimentos foram realizados a partir de quatro modelos de filtragem colaborativa, sendo dois deles baseados em vizinhança (User KNN e Item KNN) e outros dois baseados em modelos mais complexos (SVD++ e BMF). A escolha de algoritmos com níveis de complexidade distintos teve a motivação de entender o comportamento da interação entre eles.

O artigo é estruturado da seguinte forma: a seção 2 aborda trabalhos relacionados ao tema da combinação de modelos de recomendação; a seção 3 apresenta uma visão geral dos algoritmos de filtragem colaborativa utilizados neste trabalho; a seção 4 detalha a arquitetura e as etapas da abordagem proposta; a seção 5 apresenta os resultados encontrados; e a seção 6 conclui o trabalho com as suas contribuições e proposições de trabalhos futuros.

2 TRABALHOS RELACIONADOS

Collins *et al.* [3] apresentam os conceitos de seleção de sistemas de recomendação em nível macro e micro [3]. No nível macro é selecionado o melhor algoritmo de recomendação para um determinado contexto, ou para uma determinada base de dados, enquanto no nível micro a seleção é realizada buscando o algoritmo de melhor performance para cada requisição individualmente, ou para cada par usuário-item. Os autores estudaram o comportamento de nove algoritmos de recomendação no nível micro e macro, e comparam o desempenho de cada um, identificando aqueles que apresentam melhor performance no nível macro a partir da métrica RMSE - *root mean square error* - e no nível micro a partir do percentual de pares usuário-item avaliados com menor erro.

O trabalho de Collins *et al.* [3] prossegue até a definição de uma arquitetura para seleção automática do melhor algoritmo em cada requisição no nível micro de recomendação baseada em um modelo linear de predição do erro de avaliação de cada algoritmo, selecionando para cada par aquele com menor erro previsto. Ao final do experimento, o RMSE da solução não conseguiu superar o obtido pelo melhor algoritmo que compunha o conjunto de algoritmos. A arquitetura proposta pelos autores é apresentada na Figura 1.

Ekstrand & Riedl [5] estudaram os motivos que podem levar algoritmos de recomendação a falhar em determinados contextos, quando comparado a outras abordagens. Embora os resultados tenham sido apenas preliminares, os autores constataram que

existem elementos ou características dos usuários e dos itens que podem facilitar o processo de seleção automática da melhor estratégia. Particularmente, o trabalho de Ekstrand & Riedl [5] correlacionou o número de avaliações do usuário à probabilidade de uma abordagem de filtragem colaborativa item-item ser o melhor preditor em um cenário genérico (dentre as abordagens avaliadas no trabalho).

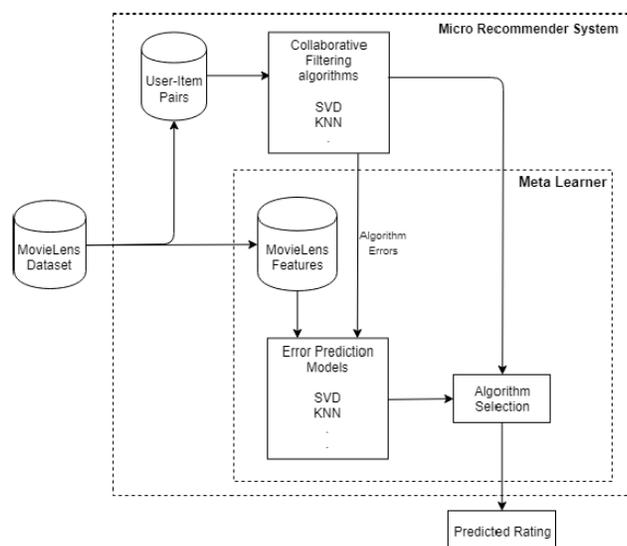


Figura 1 – Arquitetura de predição baseada na seleção em nível micro do melhor algoritmo por requisição. Fonte: Collins et al. [3].

Em um trabalho de revisão sistemática, Cunha et al. [4] realiza uma comparação entre diversas abordagens propostas em estudos primários relacionados à seleção de algoritmos de recomendação através da aplicação de técnicas de aprendizado de máquina. O trabalho se restringe à seleção de algoritmos que obtenha melhores taxas para uma base inteira e expande o critério de comparação entre os algoritmos de meta-aprendizado para além da acurácia, utilizando também as métricas de NDCG (*Normalized Discounted Cumulative Gain*), RMSE (*Root Mean Squared Error*) e NMAE (*Normalized Mean Absolute Error*).

3 ALGORITMOS DE RECOMENDAÇÃO

A arquitetura de seleção e combinação de sistemas de recomendação proposta neste artigo, que será detalhada na seção de metodologia, considera a utilização de quatro algoritmos de recomendação: KNN baseado em usuário (*User KNN*), KNN baseado em item (*Item KNN*), SVD++ e BMF. De maneira geral, todas as abordagens escolhidas são baseadas em filtragem colaborativa e em feedback explícito do

usuário, embora o SVD++ considere adicionalmente a contribuição de feedback implícito. Os algoritmos são apresentados nos subtópicos seguintes.

3.1 User KNN

O *K-Nearest Neighbors* (KNN) é um algoritmo consolidado em diversas aplicações de aprendizagem de máquina, incluindo a área de sistemas de recomendação.

O *User KNN* é um algoritmo que prediz a avaliação de um usuário u a um item i utilizando as avaliações dadas a i pelos usuários similares a u , os chamados vizinhos mais próximos [1]. É classificado como uma técnica de filtragem colaborativa.

Dada a classificação feita por um grupo de usuários a um conjunto de itens, a tarefa do algoritmo é encontrar os vizinhos mais próximos a partir de algum critério de semelhança, ou em outras palavras, encontrar os usuários mais correlatos com o usuário alvo, e em seguida prever a sua avaliação a partir daquelas de seus semelhantes.

Vários critérios de semelhança podem ser utilizados. Este trabalho utiliza um dos critérios mais populares, a correlação de Pearson [6], cujo cálculo é apresentado na equação 1, onde \vec{u} e \vec{v} representam os vetores de avaliação dos usuários u e v respectivamente, I é o conjunto de itens avaliados por ambos os usuários e ε é um número muito pequeno que possui a função de evitar a divisão por zero.

$$\Psi(u, v) = \frac{\sum_{i \in I} (\vec{u}(i) - \bar{u}) * (\vec{v}(i) - \bar{v})}{\sqrt{\sum_{i \in I} (\vec{u}(i) - \bar{u})^2 * \sum_{i \in I} (\vec{v}(i) - \bar{v})^2} + \varepsilon} \quad (1)$$

A partir da semelhança encontrada entre os usuários, a predição é feita conforme função utilidade, de acordo com a equação 2, onde v e l são o usuário e o item alvo respectivamente, h representa o número de vizinhos a serem considerados, V_k o k -ésimo vizinho mais próximo (mais semelhante) ao usuário alvo, \bar{v} é a média das avaliações do usuário v , \bar{V}_k é a média das avaliações do usuário V_k e $\bar{V}_k(l)$ representa a avaliação do usuário V_k ao item l .

$$\rho(v, l) = \bar{v} + \frac{\sum_{k=1}^h (\bar{V}_k(l) - \bar{V}_k) * \Psi(u, v_k)}{\sum_{k=1}^h \Psi(u, v_k)} \quad (2)$$

3.2 Item KNN

O KNN também pode ser baseado nos itens. A ideia geral é a mesma descrita no subtópico anterior,

substituindo a busca por usuários mais próximos pela identificação dos itens semelhantes ao item alvo. A partir dos itens correlatos, a predição é realizada de forma análoga àquela realizada pelo User KNN.

3.3 BMF

O BMF, ou *Biased Matrix Factorization*, é uma abordagem baseada em fatoração de matrizes, uma técnica que caracteriza itens e usuários por vetores de fatores inferidos a partir dos padrões de avaliação dos itens [7].

A fatoração da matriz de avaliações (ou matriz de *rankings*) de dimensionalidade f consiste em sua aproximação pela função $R' = PQ^t$, onde P representa os fatores de usuário e Q os fatores dos itens, em que cada item i é associado a um vetor $q_i \in R^f$ e cada usuário u é associado a um vetor $p_u \in R^f$ [1].

No algoritmo BMF, o modelo básico de fatoração de matrizes é modificado para considerar as médias de avaliações dadas pelos usuários e recebidas pelos itens. Desta forma, a função de predição da avaliação de um item i por um usuário u é definida por $r' = b_{ui} + p_u^t q_i$, onde b_{ui} , definido como $b_{ui} = \mu + b_u + b_i$, indica a diferença de estimativa do item i e usuário u em relação à média geral das avaliações μ [7].

Para aprender os vetores de fatores (p_u e q_i), o sistema minimiza o erro quadrático do conjunto de avaliações conhecidas [7], conforme expressão (3).

$$\begin{aligned} \min_{q^t p^t} \sum_{(u,i) \in k} & \left((r_{ui} - q_i^t p_u)^2 + \lambda (\|q_i\|^2 + \|p_u\|^2) \right) \\ \min_{q^t p^t} \sum_{(u,i) \in k} & \left((r_{ui} - \mu - b_u - b_i - p_u^t q_i)^2 \right. \\ & \left. + \lambda (b_u^2 + b_i^2 + \|q_i\|^2 + \|p_u\|^2) \right) \end{aligned} \quad (3)$$

Em seguida, a técnica de otimização de gradiente descendente é aplicada em todas as avaliações da base de treinamento. Para cada avaliação r_{ui} , uma predição \hat{r}_{ui} é realizada, o erro de predição associado $e_{ui} \stackrel{\text{def}}{=} r_{ui} - \hat{r}_{ui}$ é computado e os parâmetros são modificados movendo-se no sentido contrário do gradiente [1], conforme expressão (4) a (7).

$$b_u \leftarrow b_u + \gamma(e_{ui} - \lambda * b_u) \quad (4)$$

$$b_i \leftarrow b_i + \gamma(e_{ui} - \lambda * b_i) \quad (5)$$

$$q_i \leftarrow q_i + \gamma(e_{ui} * p_u - \lambda * q_i) \quad (6)$$

$$p_u \leftarrow p_u + \gamma(e_{ui} * q_i - \lambda * p_u) \quad (7)$$

3.4 SVD++

O algoritmo SVD++, *Singular Value Decomposition*, foi proposto por Koren *et al.* [7] e, assim como o BMF, é baseado na fatoração de matrizes.

Para lidar com o problema de *cold start*, que ocorre na entrada de novos usuários desconhecidos para o sistema [1], Koren buscou incorporar ao modelo fontes de informação adicionais sobre os usuários. Matematicamente, os autores utilizaram um vetor de fatores adicionais $y_i \in R^f$ e definiram $N(u)$, que contém todos os itens que o usuário u manifestou interesse implícito [8]. O modelo resultante é apresentado na equação (4).

$$\hat{r}_{ui} = b_{ui} + q_i^t \left(p_u + |N(u)|^{-1/2} \sum_{j \in N(u)} y_j \right) \quad (4)$$

O processo de aprendizado dos parâmetros mantém a ideia da minimização do erro quadrático através da utilização da técnica do gradiente [8], acrescentando para cada avaliação da base de treinamento, a modificação dos fatores adicionais, calculado para cada $j \in N(u)$, conforme equação 8.

$$y_j \leftarrow y_j + \gamma(e_{ui} * |N(u)|^{-1/2} * q_i - \lambda * y_j) \quad (8)$$

4 METODOLOGIA

Inspirado em Collins *et al.* [3] e buscando avaliar o efeito da combinação de algoritmos de recomendação, este trabalho utiliza uma arquitetura de meta-aprendizado conforme figura 2. Diferente do artigo mencionado, a proposta deste trabalho busca identificar a possibilidade de seleção de um modelo de recomendação ideal por usuário e não por requisição (par usuário-item). Além disso, todas as combinações possíveis entre quatro algoritmos de recomendação foram testadas como entrada para o modelo e dois critérios distintos de seleção foram empregados.

Os quatro algoritmos de base utilizados no experimento foram do tipo filtragem colaborativa e predizem a avaliação de determinado usuário a um item alvo. Consequentemente, a arquitetura completa preserva estas características.

A base de dados utilizada neste trabalho foi a MovieLens 100k [9]. Esta base foi coletada através do site MovieLens durante o período de sete meses, entre setembro de 1997 e abril de 1998 e contém

100.000 avaliações feitas por 943 usuários a 1682 filmes. Usuários com menos de 20 avaliações não foram incluídos nesta base. A base também apresenta informações demográficas sobre os usuários e características dos filmes avaliados.

A base de dados foi dividida em três conjuntos distintos na proporção 60% : 20% : 20%, respectivamente para realizar o treinamento dos

Como medida de comparação entre os modelos utilizados, os tempos de treinamento foram aferidos, demonstrando claramente a maior complexidade dos algoritmos baseados em fatoração de matrizes. Os dados são apresentados na tabela 2. Os tempos foram obtidos em uma máquina com processador i5 de terceira geração e memória RAM de 6 Gb.

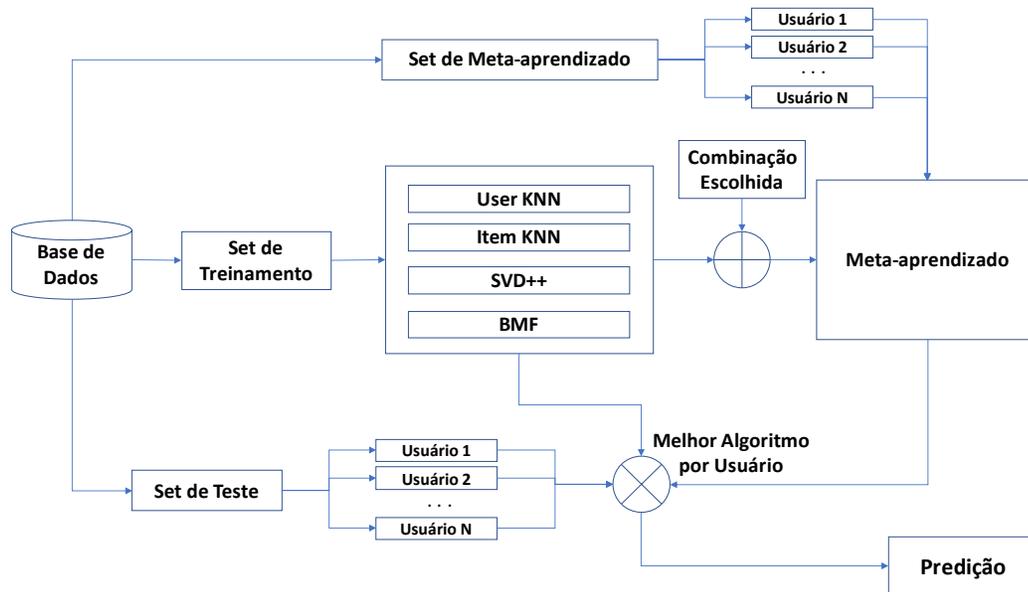


Figura 2 – Arquitetura de predição proposta.

algoritmos individualmente, para identificar o melhor algoritmo por usuário (meta-aprendizado) e para o teste da solução.

Os experimentos foram realizados a partir da biblioteca de algoritmos de sistemas de recomendação *MyMediaLite* [10], versão 3.11, disponível em <http://ismll.de/mymedialite>. A configuração dos principais parâmetros escolhidos para cada um dos algoritmos é apresentada na tabela 1 e foram determinados através de sugestões de aplicação disponíveis na página da biblioteca.

4.1 Treinamento dos Algoritmos

Cada um dos quatro algoritmos foi treinado e testado com os mesmos dados, utilizando a estratégia *5-fold cross validation* conforme apresentado na Figura 3. O treino foi realizado com 60% da base e o teste com 20%. Uma parte da base (20%), reservada para a aplicação do modelo de meta-aprendizado, não foi utilizada para permitir a comparação com os resultados obtidos nas fases posteriores do experimento.

Tabela 1 – Principais parâmetros de cada algoritmo.

Algoritmo	Parâmetros
User KNN	Correlação: Pearson; K = 60;
Item KNN	Correlação: Pearson; K = 40;
SVD++	Número de fatores = 50; Taxa de aprendizado = 0,01; Número de iterações = 50;
BMF	Número de fatores = 40; Taxa de aprendizado = 0,07; Número de iterações = 100;

Tabela 2 – Tempo de treinamento de cada modelo utilizando 20% da base Movielens-100k com os mesmos recursos computacionais.

Algoritmo	Tempo de Treinamento (ms)
User KNN	774

Item KNN	1.081
SVD++	288.421
BMF	20.001

4.2 Meta-Aprendizado: Melhor Algoritmo de Recomendação por Usuário pelo Critério NRMSE

Nesta etapa, 20% da base foi utilizada para encontrar o melhor algoritmo para cada usuário, através da aplicação do modelo de meta-aprendizado.

Os modelos individuais, treinados na etapa anterior, foram aplicados ao conjunto de meta-aprendizado. A métrica NRMSE foi então calculada por usuário, utilizando as predições geradas por cada um dos quatro algoritmos de base. Aquele que apresentou o menor erro foi selecionado para cada usuário. O NRMSE é a versão normalizada do RMSE e é calculado a partir da Equação 9:

$$NRMSE = \frac{1}{Y_{max} - Y_{min}} \sqrt{\frac{\sum_{i=1}^N (Y_i - \hat{Y}_i)^2}{N}} \quad (9)$$

A avaliação do modelo de meta-aprendizado utilizou o mesmo conjunto de teste da etapa de aplicação dos algoritmos individuais. A predição de cada observação (par usuário-item) do conjunto foi realizada pelo algoritmo selecionado para o respectivo usuário. O ciclo completo foi repetido utilizando a estratégia *5-fold cross validation* conforme figura 4.

Foram avaliadas todas as combinações de algoritmos de entrada: 1 combinação utilizando os quatro algoritmos, 4 combinações utilizando três algoritmos e 6 combinações utilizando 2 algoritmos.

4.3 Meta-Aprendizado: Melhor Algoritmo de Recomendação por Usuário pelo Critério Utilidade de Meia Vida

A métrica NRMSE, utilizada na etapa anterior, atribui o mesmo peso ao erro de predição de todos os itens, independentemente da importância destes para o usuário.

Com o objetivo de considerar o efeito da combinação de algoritmos de recomendação, levando em conta a importância dos itens melhor avaliados pelo usuário, todo o processo da etapa anterior foi repetido utilizando a métrica utilidade de meia vida

(*half-life utility*) em substituição ao NRMSE, tanto na fase de meta-aprendizado quanto de teste.

A métrica utilidade de meia vida foi criada por Breese [11] levando em consideração a probabilidade de o usuário avaliar positivamente determinado item.

A utilidade é calculada a partir das equações 10 e 11, onde d representa o voto neutro (para o nosso trabalho definido com a metade da escala), j é a posição do item em ordem decrescente dos itens com melhores avaliações de acordo com o sistema de recomendação, $r_{a,j}$ é a avaliação feita pelo usuário ao item que ocupa a posição j , e ε é a constante de meia vida. Para ε definimos o valor 5, o mesmo originalmente utilizado por Breese [11].

$$R_a = \sum_j \frac{\max(R_{a,j} - d, 0)}{2^{(j-1)/(\varepsilon-1)}} \quad (10)$$

$$R = 100 \frac{\sum_a R_a}{\sum_a R_a^{max}} \quad (11)$$

5 RESULTADOS

Os resultados da primeira etapa do experimento serviram de base comparativa para entender os efeitos da aplicação personalizada dos algoritmos de recomendação. Os NRMSE e a utilidade de meia vida foram aferidos de acordo com a arquitetura da figura 4 para os quatro algoritmos de base e são apresentados na Tabela 3.

Tabela 3 – Resultados de NRMSE e Utilidade de meia vida para os quatro algoritmos de base aplicados conforme arquitetura de treinamento e teste da Figura 3.

Algoritmo	NRMSE	Utilidade MV
User KNN	0,2374	0,7700
Item KNN	0,2362	0,7690
SVD++	0,2304	0,7878
BMF	0,2310	0,7843

Já na segunda etapa, tanto o critério de escolha do algoritmo por usuário como a métrica de avaliação foram o erro médio quadrático normalizado. A tabela 4 apresenta o resultado, indicando a combinação de algoritmos de entrada utilizado. Ou seja, o melhor algoritmo para cada usuário foi escolhido dentre as opções elencadas em cada simulação.

Comparando os resultados com aqueles da

combinações compostas por algoritmos de complexidade semelhantes.

A terceira etapa utilizou a mesma arquitetura de teste da etapa anterior (Figura 4), porém neste caso, o critério de escolha do algoritmo e a métrica de avaliação utilizada foi a Utilidade de Meia Vida. Mais uma vez, as combinações que obtiveram melhores resultados que seus componentes individuais foram a

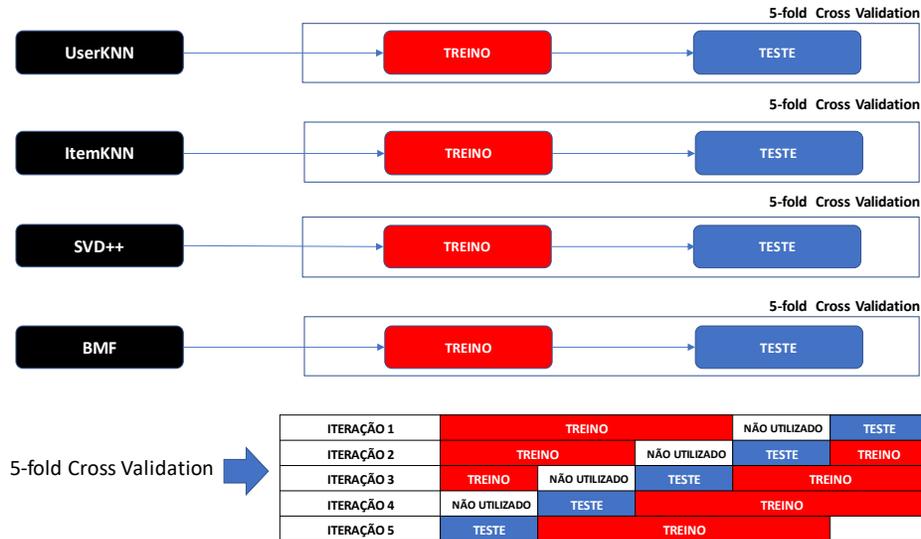


Figura 3 – Arquitetura de treinamento e teste dos algoritmos individualmente.

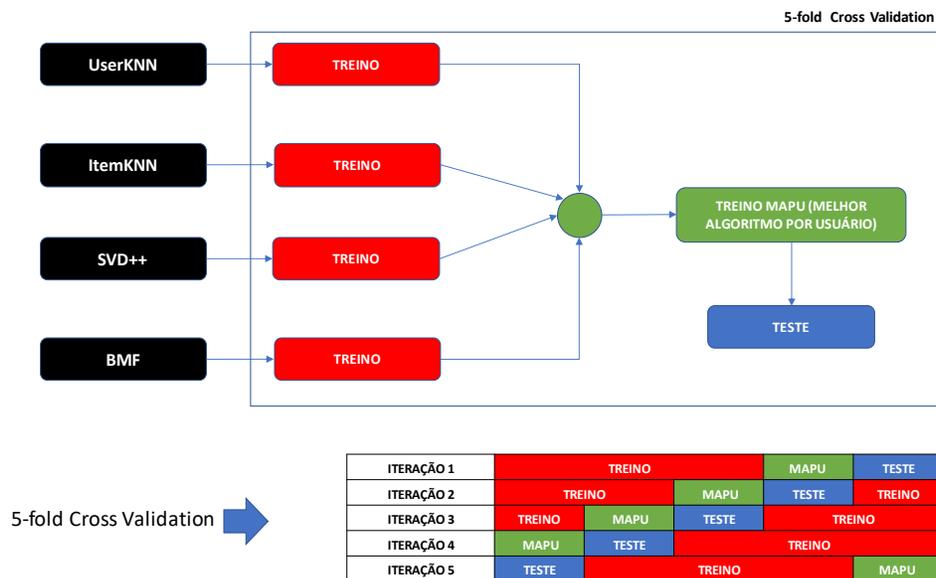


Figura 4 – Arquitetura de treinamento e teste do modelo com melhor algoritmo por usuário.

primeira etapa, apenas as combinações 6 (User KNN e Item KNN) e 11 (SVD++ e BMF) obtiveram ganho em relação ao NRMSE medido quando da aplicação individual dos algoritmos que compõem cada combinação. Observa-se que estas são as

6 e a 11, conforme Tabela 5.

Tabela 4 – Resultados de NRMSE para as combinações de algoritmos aplicados conforme arquitetura de treinamento e teste da Figura 4.

#	Combinação	NRMSE
1	User KNN / Item KNN / SVD++ / BMF	0,2311
2	User KNN / Item KNN / SVD++	0,2320
3	User KNN / Item KNN / BMF	0,2327
4	User KNN / SVD++ / BMF	0,2308
5	Item KNN / SVD++ / BMF	0,2306
6	User KNN / Item KNN	0,2356
7	User KNN / SVD++	0,2316
8	User KNN / BMF	0,2323
9	Item KNN / SVD++	0,2314
10	Item KNN / BMF	0,2320
11	SVD++ / BMF	0,2299

Tabela 5 – Resultados de Utilidade de meia vida para as combinações de algoritmos aplicados conforme arquitetura de treinamento e teste da Figura 4.

#	Combinação	NRMSE
1	User KNN / Item KNN / SVD++ / BMF	0,7817
2	User KNN / Item KNN / SVD++	0,7821
3	User KNN / Item KNN / BMF	0,7774
4	User KNN / SVD++ / BMF	0,7836
5	Item KNN / SVD++ / BMF	0,7842
6	User KNN / Item KNN	0,7711
7	User KNN / SVD++	0,7829
8	User KNN / BMF	0,7799
9	Item KNN / SVD++	0,7826
10	Item KNN / BMF	0,7791
11	SVD++ / BMF	0,7884

6 CONCLUSÕES

Este trabalho investigou a aplicação de uma arquitetura de combinação de modelos de recomendação para seleção de algoritmos personalizados por usuário.

Os testes foram realizados a partir de quatro algoritmos de recomendação, sendo dois baseados em vizinhança (modelos mais simples) e dois baseados em fatoração de matrizes (modelos mais complexos), e investigaram o efeito da recomendação personalizada considerando como entrada todas as combinações possíveis entre estes quatro modelos.

Foram avaliados os efeitos da combinação de algoritmos no desempenho do sistema a partir de duas métricas: o NRMSE baseado no erro médio

quadrático das predições do sistema, e a Utilidade de meia vida que considera a importância dos itens melhor avaliados pelo usuário. Em ambos os pontos de vista, as combinações que apresentaram taxas superiores aos algoritmos de entrada foram aquelas formadas por modelos de complexidade semelhante. Aparentemente, a participação dos algoritmos de menor complexidade combinados com os mais complexos prejudica a performance destes últimos, não se mostrando uma alternativa favorável à melhoria do desempenho do sistema de recomendação.

Os resultados do trabalho são limitados em função da utilização de apenas uma base de dados. Desta forma, sugerimos como ideia para trabalhos futuros, investigações mais detalhadas em outras bases e com algoritmos de entrada distintos, permitindo um maior conhecimento a respeito dos efeitos da aplicação personalizada de modelos de recomendação.

Recomendamos também estudos sobre o desempenho de combinações de algoritmos em situações com poucos dados de treinamento, que podem ser avaliadas através da aplicação da estratégia de validação cruzada invertida, onde a diferença de performance entre os algoritmos tende a ser mais contundente.

Além disso, novas pesquisas podem ser desenvolvidas para avaliar a qualidade deste tipo de abordagem para detecção de usuários classificados como outliers e para recomendação de itens para este tipo de usuário.

REFERÊNCIAS

- [1] RICCI, F.; ROKACH, L.; SHAPIRA, B.; KANTOR, P. B. **Recommender Systems Handbook**, 1st edition. Springer-Verlag, Inc., New York, NY, USA, 2010 URL: <http://dl.acm.org/citation.cfm?id=1941884>.
- [2] PORIYA, A.; PATEL, N.; BHAGAT, T.; SHARMA, R. **Non-Personalized Recommender Systems and User-based Collaborative Recommender Systems**. International Journal of Applied Information Systems (IJ AIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA. Volume 6– No. 9, March 2014.
- [3] COLLINS, A.C; BEEL, J.; TKACZYK, D. **One-at-a-time: A Meta-Learning Recommender-System for Recommendation-Algorithm Selection on Micro Level**. In Information Retrieval, 2018.

[4] CUNHA, T.; SOARES, C.; CARVALHO, A.C.P.L.F de; **Metalearning and Recommender Systems: A literature review and empirical study on the algorithm selection problem for Collaborative Filtering**. Information Sciences 423 (2018) 128–144.

[5] EKSTRAND, M.; RIEDL, J. **When recommenders fail: predicting recommender failure for algorithm selection and combination**, ACM Conf. Recommend. Syst. (2012) 233–236.

[6] PEARSON, K. (1948). **Early Statistical Papers**. Cambridge, England: University Press.

[7] KOREN, Y.; BELL, R.; VOLINSKY, C. **Matrix factorization techniques for recommender systems**, Computer 42(8) (2009) 30–37. URL: <http://dx.doi.org/10.1109/MC.2009.263>.

[8] DA COSTA, A. F.; MANZATO, M. G. **Exploiting multimodal interactions in Recommender Systems with Ensemble Algorithms**. Information Systems 56 (2016) 120–132.

[9] HARPER, F. M.; KONSTAN, J. A. **The MovieLens Datasets: History and Context**. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19. 2015. DOI=<http://dx.doi.org/10.1145/2827872>.

[10] GANTNER, Z.; *et al.* **MyMediaLite: A free recommender system library**. Proceedings of the fifth ACM conference on Recommender systems. ACM, 2011.

[11] BREESE, J.; HECKERMAN, D. & KADIE, C. (1998). **Empirical analysis of predictive algorithms for collaborative filtering**. In Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98), 43–52.