Hybrid Recommendation Algorithm Fusing Content and Matrix Factorization (Postprint)

Authors: Wang Yonggui, Chen Yuwei

Date: 2019-04-01T00:00:00+00:00

Abstract

Traditional content-based recommendation algorithms typically exhibit low accuracy, while collaborative filtering recommendation algorithms commonly suffer from data sparsity and item cold-start problems. To address these issues, we propose a hybrid recommendation algorithm that integrates content and collaborative matrix factorization techniques. This algorithm achieves factorization of content and collaborative matrices in a shared low-dimensional space while preserving the local structure of the data. For parameter optimization, it employs an iterative method based on multiplicative update rules to enhance learning capability. Experimental results demonstrate that the proposed algorithm outperforms other representative item cold-start recommendation algorithms, effectively alleviating data sparsity and improving recommendation accuracy.

Full Text

Preamble

Vol. 37 No. 5
Application Research of Computers
ChinaXiv Cooperative Journal

Hybrid Recommendation Algorithm Based on Content and Matrix Factorization

Wang Yonggui, Chen Yuwei†
(College of Software, Liaoning Technical University, Huludao, Liaoning 125105, China)

Abstract: Traditional content-based recommendation algorithms often suffer from low accuracy, while collaborative filtering recommendation algorithms commonly face data sparsity and item cold-start problems. To address these issues, this paper proposes a hybrid recommendation algorithm that integrates content and collaborative matrix factorization techniques. The algorithm simultaneously decomposes content and collaborative matrices in a shared low-dimensional space while preserving the local structure of the data. For parameter optimization, an iterative method based on multiplicative update rules is employed to enhance learning capability. Experimental results demonstrate that the proposed algorithm outperforms other representative item cold-start recommendation algorithms, effectively alleviating data sparsity and improving recommendation accuracy.

Keywords: hybrid recommendation; matrix factorization; cold start; parameter optimization; local structure

---

0 Introduction

With the advent of the big data era, how to effectively process and utilize user information to solve information overload has become a crucial research topic in the Internet domain. Consequently, personalized recommendation systems have emerged [**?**]. Recommendation algorithms are now widely applied in platforms such as Amazon, Taobao, and NetEase Cloud Music. These systems can filter and mine deep relationships between users and items from massive historical data, generating accurate personalized recommendations that satisfy user preferences.

Recommendation algorithms can be categorized into three main types: content-based [**?**], collaborative filtering [**?**], and hybrid approaches [**?**]. Content-based methods extract item attributes for feature learning and recommend items with high similarity scores. Collaborative filtering, the most extensively studied approach, includes user-based and item-based variants that compute similarities through user-item rating matrices. However, in practice, only a small fraction of users rate or review a limited number of items, leading to severe data sparsity and item cold-start problems [**?**]. Cold-start issues can be divided into user cold-start and item cold-start—recommending items to new users or recommending new items to existing users. While content-based methods can mitigate cold-start problems, their inherent low accuracy prevents them from being the sole solution.

To address these challenges, this paper combines content-based recommendation with matrix factorization to propose a hybrid recommendation algorithm. The main contributions are summarized as follows:

a) Content and collaborative information are integrated within a unified matrix factorization framework while leveraging data local structure. The local smoothness of low-dimensional representations is measured through a weight matrix formed by nearest neighbors.

b) A learning model based on multiplicative update rules is utilized for parameter optimization, with non-negativity constraints and regularization applied to optimize the objective function and reduce runtime.

c) Extensive experiments on the public nips12raw_str602 dataset demonstrate that the proposed algorithm outperforms several representative item cold-start recommendation algorithms, improving recommendation accuracy and effectively solving the item cold-start problem.

---

1 Related Work

Hybrid recommendation algorithms primarily aim to combine the advantages of content-based and collaborative filtering methods while avoiding their shortcomings, effectively alleviating cold-start problems and achieving diversity. However, they often suffer from high complexity, increased recommendation time, and difficulty in achieving balance. Common hybrid approaches include weighted [?], merged [?], and feature combination [?] types. Weighted hybrid recommendation combines results from multiple algorithms through weighting. Merged hybrid recommendation generates multiple recommendation results using various algorithms. Feature combination hybrid recommendation combines features from different recommendation data sources.

Several relevant algorithms are briefly introduced below. Soboroff et al. [?] proposed a latent semantic indexing-based technique, an early hybrid recommendation algorithm combining collaborative filtering and document content for text recommendation. This method discovers topics in collections by constructing content profiles for each user and recommends documents most similar to user profiles. Singh et al. [?] introduced collaborative matrix factorization, a general framework for multi-relational factorization models that addresses item recommendation and rating prediction problems. Gantner et al. [?] used KNN (K-nearest-neighbor) and BPR (Bayesian personalized ranking) to optimize the mapping between items/users and latent factors. Wang et al. [?] proposed a hybrid recommendation framework that classifies users based on context, operation history, and user features, then dynamically selects recommendation algorithms. Ahn [?] introduced a hybrid similarity calculation method incorporating personalized information to mitigate cold-start problems, though with high implementation complexity. Rosen-Zvi et al. [?] proposed a generative probabilistic model ATM (author-topic model) that associates each author with a multinomial distribution over topics, each having a multinomial distribution over words. In news recommendation, users are modeled as authors. Felfernig et al. [?] proposed mapping high-dimensional rating matrices to low-dimensional feature matrices using implicit user data and social information for recommendation, improving accuracy but with some information loss. Cai et al. [?] utilized data local geometric structure for better low-dimensional representation. Inspired by label propagation in nearest-neighbor graphs, they proposed a constrained clustering technique that achieved good results.

While these algorithms incorporate different factors and effectively alleviate cold-start problems, they suffer from high complexity or information loss issues. The proposed algorithm incorporates advantages from the above methods while avoiding information loss. It leverages both low-rank and local properties of matrices, applies non-negativity constraints to enhance effects, and uses multiplicative update rules for parameter optimization, effectively solving cold-start problems and improving recommendation accuracy.

2 Hybrid Recommendation Algorithm Fusing Content and Matrix Factorization

For initial item matrix X_I and user matrix X_U, the algorithm performs collaborative factorization using parameter matrices W, H_I, and H_U, processes and optimizes the factorized data by considering local characteristics, defines an optimization objective function controlled by parameters α, β, and λ, and employs a parameter learning algorithm to iteratively update matrices W, H_I, and H_U under the optimization objective until reaching a stable point to generate recommendations.

2.1 Problem Description

To address the item cold-start problem, new items should be recommended to potentially interested users—those who haven't expressed interest but may be interested. Based on user history and new item descriptions, the most likely interested users should be retrieved. This problem can be defined as follows: For a set of n items, matrix $X_I \in \mathbb{R}^{n\times m}$ stores m item attributes, where each row represents an item and each column represents an attribute (see Table 1). Matrix $X_U \in \mathbb{R}^{n\times u}$ stores u users, where cell (i, j) indicates whether user j is interested in item i (see Table 2).

Based on these definitions, each item and its description are associated with consuming users, such as news articles. Each news article is described by vocabulary and users who commented. This scenario can be represented by two matrices: an item matrix $X_I \in \mathbb{R}^{n\times v}$ where n is the number of documents and v is vocabulary size, and a user matrix $X_U \in \mathbb{R}^{n\times u}$ where n is the number of documents and u is the number of users. Matrix X_I can represent TF-IDF scores of words in documents, while X_U reflects whether relevant users commented on the current document.

Factorizing X_I into two low-dimensional matrices can discover document topics and the degree to which each document belongs to each topic. Similarly, factorizing X_U can discover user communities and the degree to which documents generate interest within communities. The problem with such separate factorization is that documents and users are not represented in a common latent space; each factorization represents a different latent space, making it impossible to connect topics with communities. To represent documents and users in the same space, where each item can be described by both a topic and a community (set of users), this paper performs low-dimensional representation of X_I and X_U in a common space. For matrices X_I and X_U, the following optimization is defined:

$$Z = \alpha\|X_I - WH_I\|_F^2 + (1-\alpha)\|X_U - WH_U\|_F^2 + \lambda(\|W\|_F^2 + \|H_I\|_F^2 + \|H_U\|_F^2)$$

where the first two terms achieve factorization of item matrix X_I and user matrix X_U in the same latent space using the same matrix W, with $\|\cdot\|_F$ denoting the Frobenius norm, enabling dimensionality reduction of X_I and X_U. Hyperparameter α controls the importance of matrix factorization within $[0,1]$. If $\alpha = 0.5$, both factorizations are equally important; if $\alpha < 0.5$, user matrix X_U factorization is more important; if $\alpha > 0.5$, item content matrix X_I factorization is more important. The last term is a regularization term controlled by hyperparameter $\lambda \geq 0$ on parameter matrices W, H_I, and H_U to prevent overfitting and measure function smoothness.

Table 1 Item-Attribute Table
Table 2 Item-User Table

2.2 Local Structure Characteristics and Measurement

When performing collaborative factorization, a common low-dimensional space should be found to process and optimize data of different dimensions, as shown in Equation (1). When considering local structure, the manifold assumption is introduced, which primarily considers the model's local properties: if two data points x_i and x_j are close in the original distribution geometry, they should also be close in low-dimensional space. This plays an important role in dimensionality reduction [**?**] and semi-supervised learning [**?**]. However, this theory cannot be directly applied when the distribution's geometric structure is unknown. According to manifold learning research [**?**], local geometric structure can be modeled through nearest-neighbor graphs.

Assume a graph with n nodes, where each node represents a data point. Find g nearest neighbors for each point and connect them. These connections can be represented using binary or weighted methods. In binary representation, nearest neighbors are marked as 1 and others as 0. In weighted representation, cosine similarity values can be used. This paper adopts weighted representation using cosine similarity values to form adjacency matrix A, measuring local closeness between two data points. If $x_i = (x_1, y_1)$ and $x_j = (x_2, y_2)$, the cosine similarity between x_i and x_j is calculated as:

$$sim(x_i, x_j) = \frac{x_i^T x_j}{\|x_i\|\|x_j\|}$$

After collaborative factorization using matrix W, each data point x_i is mapped to a low-dimensional representation w_i. The distance between two low-dimensional data points is measured by Euclidean distance, i.e., $\|w_i - w_j\|_2$. The adjacency weight matrix A can measure the local smoothness of low-dimensional representations. The loss function is given by:

$$S = \sum_{i=1}^{n} \sum_{j=1}^{n} \|w_i - w_j\|_2^2 A_{ij} = Tr(W^T DW) - Tr(W^T AW) = Tr(W^T LW)$$

where D is a diagonal matrix with diagonal values equal to the sum of each row in A, i.e., $D_{ii} = \sum_j A_{ij}$. Tr denotes the matrix trace, and L is the Laplacian matrix [**?**].

2.3 Parameter Learning Algorithm Based on Multiplicative Updates

To incorporate this local characteristic, Equation (1) is optimized by adding parameter β to control the degree of locality. The optimized objective function becomes:

$$Z = \alpha\|X_I - WH_I\|_F^2 + (1-\alpha)\|X_U - WH_U\|_F^2 + \beta Tr(W^T LW) + \lambda(\|W\|_F^2 + \|H_I\|_F^2 + \|H_U\|_F^2)$$

where α controls the importance of X_I and X_U factorization, λ controls regularization to prevent overfitting, and L is the Laplacian matrix from Equation (3).

The above objective function aims to find the minimum through iteration. However, since parameters W, H_I, and H_U are non-convex, finding the global minimum is extremely difficult. Therefore, a multiplicative update rule-based iterative algorithm is derived for parameter learning to achieve a stable point. First, partial derivatives with respect to W, H_I, and H_U are computed as shown in Equations (5)-(7).

Then, using KKT (Karush-Kuhn-Tucker) first-order optimality conditions [**?**], the following conclusions can be derived:

$$W \geq 0, \quad H_I \geq 0, \quad H_U \geq 0$$

$$W \odot \nabla_W Z = 0, \quad H_I \odot \nabla_{H_I} Z = 0, \quad H_U \odot \nabla_{H_U} Z = 0$$

where $\odot$ denotes element-wise matrix multiplication. Substituting Equations (5)-(7) into the three conclusions in Equation (10) yields the update rules for parameters W, H_I, and H_U:

$$W \leftarrow W \odot \frac{\alpha X_I H_I^T + (1-\alpha) X_U H_U^T}{\alpha W H_I H_I^T + (1-\alpha) W H_U H_U^T + \beta D W + \lambda W}$$

$$H_I \leftarrow H_I \odot \frac{W^T X_I}{W^T W H_I + \lambda H_I}$$

$$H_U \leftarrow H_U \odot \frac{W^T X_U}{W^T W H_U + \lambda H_U}$$

Through these iterations, W, H_I, and H_U achieve learning effects and ultimately reach a stable point. The objective function Z remains unchanged under the update rules when W, H_I, and H_U are at stable points. Considering the sample sizes of item matrix X_I and user matrix X_U, the influence of sample numbers is incorporated into the objective function to form the final optimization function:

$$Z = \alpha \|X_I - W H_I\|_F^2 + (1-\alpha) \|X_U - W H_U\|_F^2 + \beta Tr(W^T L W) + \lambda (\|W\|_F^2 + \|H_I\|_F^2 + \|H_U\|_F^2)$$

where v_1 represents the number of items and v_2 represents the number of users. This optimization objective function can reduce iteration count and effectively decrease runtime. Once the model learns W, H_I, and H_U, these parameters gain predictive capability. For example, given descriptions of a new news article, the most likely users to comment on it can be predicted. Using least squares [**?**], $q_i = w H_i$ is solved, then the document vector q_i is projected into the common latent space. The continuously updated w captures q_i in the latent space and computes $w H_U$, i.e., the likelihood q_u of users commenting on the new article, enabling user ranking based on these scores.

---

3 Experimental Results and Analysis

3.1 Experimental Environment

Hardware: AMD A8-5545M APU quad-core, 1.70 GHz, 4 GB RAM, 750 GB hard drive
Operating System: Windows 7
Programming Environment: MATLAB R2016a, Microsoft VC++ 2015

The experiment uses the public nips12raw_str602 dataset, available at https://cs.nyu.edu/~roweis/data.html. This is a sparse dataset with preprocessed article content (stop words, numbers, punctuation, and infrequent tokens removed), containing 2,037 articles with usernames, 13,649 descriptive words, and 1,740 titles.

3.3.1 Experimental Method The dataset is sorted chronologically, and training/test sets are generated using a sliding time window. The test set is restricted to users who have commented on at least one article. k-fold cross-validation [**?**] is employed, dividing the dataset into 10 equal groups. One group is selected as the test set while the remaining groups serve as the training set. Each experiment is repeated 10 times, with average results taken as final outcomes for comparative analysis with other algorithms.

3.3.2 Evaluation Metrics Common recommendation system performance metrics include precision, recall, mean absolute error (MAE), and root mean square error (RMSE) [**?**]. However, these are accuracy metrics with limitations in evaluating ranking performance. This experiment adopts mean average precision (MAP) and normalized discounted cumulative gain (NDCG) [**?**], both considering position factors in results. MAP balances precision and recall, with higher values for better rankings. NDCG evaluates users and recommendation lists across the entire test set with normalization. The definitions are as follows:

a) MAP is defined in Equations (15)-(16):

$$AP = \frac{1}{R}\sum_{r=1}^{R}\frac{r}{\text{position}(r)}$$

$$MAP = \frac{1}{Q}\sum_{q=1}^{Q} AP_q$$

where AP denotes average precision, R represents the number of documents, position(r) is the position of the r-th document in the list, and Q represents the total number of AP values computed.

b) NDCG measures the distance between DCG (discounted cumulative gain) and IDCG (ideal discounted cumulative gain):

$$DCG_p = \sum_{i=1}^{p}\frac{2^{\text{rel}_i}-1}{\log_2(i+1)}$$

$$NDCG_p = \frac{DCG_p}{IDCG_p}$$

where rel_i denotes the relevance grade of the i-th document, and $|REL|$ represents the relevance-ordered list of top p documents.

The experiment compares two models of the proposed algorithm with four other recommendation algorithms: the classic hybrid recommendation algorithm UP-LSI [**?**], the cold-start recommendation algorithm BPR-KNN [**?**], the generative probabilistic model ATM [?], and a recent hybrid recommendation algorithm SGHR (semi-genetic hybrid recommendation) [?] that combines genetic algorithms with weighting to improve performance.

The algorithm is evaluated through average performance, impact of latent topic number k, influence of content weight α on collaborative information, effect of smoothness parameter λ, impact of nearest neighbor count g, and runtime comparison.

1) Average Performance Comparison
Figure 1 [Figure 1: see original paper] shows the average performance of the two proposed models (CMF and CMF with $\beta = 0$) across 10 different training/test sets with optimal parameters. Both models significantly outperform the four comparison cold-start algorithms, with MAP improved by approximately 15% and NDCG improved by 5%-10%. The CMF model outperforms CMF($\beta = 0$), proving that the proposed algorithm effectively improves recommendation accuracy.

2) Impact of Latent Topic Number
The topic number k controls model complexity. Too small a k oversimplifies the model, while too large a k causes overfitting and performance degradation. Therefore, an appropriate k must be found to balance these issues. This algorithm controls k within 100-1000. Figure 2 [Figure 2: see original paper] shows NDCG at different k values. Performance is stable when k is between 400-700, reaching optimal performance at $k = 600$. Performance declines significantly when $k > 700$, with NDCG dropping below comparison algorithms. When $k > 800$, the gap between CMF($\beta = 0$) and CMF becomes substantial. Overall, the proposed algorithm far outperforms the other four algorithms.

3) Impact of Content Weight on Collaborative Information
Parameter α controls the importance of content versus collaborative information. As shown in Figure 3 [Figure 3: see original paper], both proposed models achieve high and stable performance when $\alpha \in [0.1, 0.5]$, indicating that X_U factorization is more important during collaborative factorization. CMF reaches optimal performance at $\alpha = 0.2$, and CMF consistently outperforms CMF($\beta = 0$) across all α values.

4) Impact of Solution Smoothness
Parameter λ controls solution smoothness through regularization. With $k = 600$, $\alpha = 0.2$, and $\beta = 0.25$, λ is varied in $(0, 1)$. Figure 4 [Figure 4: see original paper] shows NDCG at different λ values. Stable high performance is achieved when $\lambda \in [0.15, 0.35]$, with CMF slightly outperforming CMF($\beta = 0$) and both significantly surpassing comparison algorithms. However, performance degrades when $\lambda > 0.5$, falling below the other four algorithms.

5) Impact of Local Smoothness
The CMF model computes local smoothness through a nearest-neighbor graph adjacency weight matrix A, with parameter β controlling its importance. Figure 5 [Figure 5: see original paper] shows algorithm performance at different β val-

ues. Good performance is observed when $\beta \in [0.1, 0.9]$, with the highest NDCG at $\beta = 0.2$. Performance degrades when $\beta > 1$. Therefore, this experiment constrains β within $[0, 1]$.

6) Impact of Nearest Neighbor Count
The CMF model computes document similarity through nearest-neighbor graphs, while CMF($\beta = 0$) does not. Therefore, only CMF is studied for nearest neighbor impact, as shown in Figure 6 [Figure 6: see original paper]. High performance is maintained with 1-4 nearest neighbors, but recommendation accuracy declines when the neighbor count exceeds 5.

7) Algorithm Runtime
With parameters set to $\alpha = 0.3$, $\lambda = 0.25$, $\beta = 0.25$, runtime is tested for both models under different complexities. Each model runs 10 times at different k values, with averages shown in Figure 7 [Figure 7: see original paper]. CMF demonstrates clear advantages when $k > 300$, with significantly reduced runtime compared to CMF($\beta = 0$)—over 2,100 seconds difference at $k = 800$. This proves that CMF with nearest-neighbor weighting and local smoothness effectively reduces runtime. When $k > 800$, performance is poor and runtime is long, so these cases are not considered.

---

4 Conclusion

This paper proposes a hybrid recommendation algorithm combining content and collaborative information. Content and collaborative matrices are factorized in a common space to obtain item topics and topic degree matrices. Local structure is considered by constructing nearest-neighbor graphs to compute user preference similarity, forming a weighted adjacency matrix to measure local smoothness. Non-negativity constraints are applied, resulting in interpretable and sparse latent representations.

Multiplicative update rules are used for parameter learning to optimize the objective function and find stable performance points. The evaluation metrics MAP and NDCG incorporate position factors and increased relevance weighting to better assess ranking performance. After extensive experiments and comparison with cold-start algorithms, results show the proposed algorithm significantly outperforms comparison algorithms, effectively alleviating cold-start problems and improving recommendation accuracy. Comparison between the two proposed models proves the effectiveness of introducing local structure.

Although the proposed content and matrix factorization fusion algorithm demonstrates good performance on this dataset, its scalability to large-scale datasets requires further investigation. Future research may consider incorporating more item attributes, such as age and gender, to improve user preference similarity and recommendation accuracy.

References

[1] Chen Jiemin, Tang Yong, Li Jianguo, et al. Survey of personalized recommendation algorithms [J]. Journal of South China Normal University: Natural Science Edition, 2014, 46(5): 8-15.

[2] Wang Licai, Meng Xiangwu, Zhang Yujie. Context-aware recommender systems [J]. Journal of Software, 2012, 23(1): 1-20.

[3] Breese J S, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering [C]// Proc of the 14th Conference on Uncertainty in Artificial Intelligence. San Francisco: Morgan Kaufmann Publishers Inc, 1998: 43-52.

[4] De Campos L M, Fernández-Luna J M, Huete J F, et al. Combining content-based and collaborative recommendations: a hybrid approach based on Bayesian networks [J]. International Journal of Approximate Reasoning, 2010, 51(7): 785-799.

[5] Yu Hong, Li Junhua. Algorithm to solve the cold-start problem in new item recommendations [J]. Journal of Software, 2015, 26(6): 1395-1408.

[6] Zou Lingjun, Chen Ling, Li Juan. Time-weighted hybrid recommender algorithm [J]. Computer Science, 2016, 43(S2): 451-454.

[7] Wu Yanwen, Wang Jie. Similarity recommendation based on hybrid of GA and PSO [J]. Computer Engineering and Design, 2017, 38(2): 405-408.

[8] Li Kechao, Lan Dongmei. A collaborative filtering hybrid recommendation algorithm for attribute and rating [J]. Computer Technology and Development, 2013, 23(7): 116-119,123.

[9] Soboroff I, Nicholas C. Combining content and collaboration in text filtering [C]// Proc of the IJCAI'99 Workshop on Machine Learning for Information Filtering. 1999: 86-91.

[10] Singh A, Gordon G. Relational learning via collective matrix factorization [C]// Proc of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2008: 650-658.

[11] Gantner Z, Drumond L, Freudenthaler C, et al. Learning attribute-to-feature mappings for cold-start recommendations [C]// Proc of IEEE International Conference on Data Mining. Washington DC: IEEE Press, 2010: 176-185.

[12] Wang J H, Chen Yihao. A distributed hybrid framework to address the new-user cold-start problem [C]// Proc of the 13th IEEE International Conference on Ubiquitous Intelligence and Computing. Piscataway, NJ: IEEE Press, 2015: 1686-1691.

[13] Ahn H J. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem [J]. Information Sciences, 2008, 178(1): 37-51.

[14] Rosen-Zvi M, Griffiths T, Steyvers M, et al. The author-topic model for authors and documents [C]// Proc of Uncertainty in Artificial Intelligence. 2004: 487-494.

[15] Felfernig A, Ninaus G, Grabner H, et al. An overview of recommender systems in requirements engineering [M]// Maalej W, Thurimella A K. Managing Requirements Knowledge. Berlin: Springer, 2013: 315-332.

[16] Cai Deng, He Xiaofei, Wu Xiaoyun, et al. Non-negative matrix factorization on manifold [C]// Proc of the 8th IEEE International Conference on Data Mining. Pisa: IEEE Press, 2008: 63-72.

[17] Yin Jun, Lai Zhihui, Yan Hui. Dimension reduction using kernel collaborative representation based projection [J]. AEUE-International Journal of Electronics and Communications, 2017, 81(11): 23-30.

[18] Liu Jianwei, Liu Yuan, Luo Xionglin. Semi-supervised learning methods [J]. Chinese Journal of Computers, 2015, 38(8): 1592-1617.

[19] Allab K, Labiod L, Nadif M. Multi-manifold matrix decomposition for data co-clustering [J]. Pattern Recognition, 2017, 64(4): 386-398.

[20] Katsnelson V. The matrix function e^{tA+B} representable as the Laplace transform of a matrix measure [J]. Integral Equations and Operator Theory, 2016, 86(3): 439-452.

[21] Cichocki A, Zdunek R, Phan A H, et al. Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation[M]. [S.l.]: Wiley, 2009.

[22] Dehghan M, Mohammadi V. Error analysis of method of lines (MOL) via generalized interpolating moving least squares (GIMLS) approximation [J]. Journal of Computational and Applied Mathematics, 2017, 321(9): 540-554.

[23] Wang Meiling, Ma Jun. A novel recommendation approach based on users' weighted trust relations and the rating similarities [J]. Soft Computing, 2015, 20(10): 3981-3990.

[24] Willmott C J, Matsuura K. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance[J]. Climate Research, 2005, 30: 79-82.

[25] Keyhanipour A H, Moshiri B, Rahgozar M. CF-Rank: learning to rank by classifier fusion on click-through data [J]. Expert Systems With Applications, 2015, 42(22): 8597-8608.

[26] Mueller J. Combining aspects of genetic algorithms with weighted recommender hybridization [C]// Proc of the 19th International Conference on Information Integration and Web-based Applications & Services. New York: ACM Press, 2017: 13-22.

Note: Figure translations are in progress. See original paper for figures.