



Research on multi-context aware recommendation methods based on tensor factorization

Shulin Cheng¹ · Huimin Jiang² · Wanyan Wang¹ · Wei Jiang³

Received: 23 April 2022 / Accepted: 28 April 2023 / Published online: 15 May 2023
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

Abstract

Compared to the traditional recommender systems, context-aware recommender systems are more in line with actual application contexts. However, the existing researches are mostly focused on single context-aware recommendation, such as time-aware recommendation or location-aware recommendation, and lack of in-depth research on multi-context-aware recommendation. Therefore, we proposed a recommendation method of high-order tensor factorization based on multi-context-aware. First, on the basis of analyzing the influence of context on users' interest preferences, the sensitivity of users to multiple contexts was detected using statistical methods. For context-sensitive users, four-dimensional tensors and feature matrices used to solve data sparsity were constructed based on rating matrix and situational information. And then the stochastic gradient descent algorithm was used for iterative calculation to fill in missing data values and carry out parameter optimization. For context-insensitive users, we used matrix factorization to predict users' interest preferences. Finally, we tested and validated our method on a multi-context-aware movie dataset, and the experimental results show that the proposed method could effectively reduce the prediction error and improve the recommendation quality.

Keywords Recommender systems · Multi-context-Aware · Tensor factorization · Matrix factorization

1 Introduction

With the advent of big data, to obtain information on the Internet accurately and quickly under the condition of information overload has become a hot research direction for scholars. The recommender systems can filter the redundant information according to users' interest preferences and

recommend relevant items for them. However, in the context-aware recommender systems, users' interest preferences are also affected by the context. In different contexts, there may be significant differences in users' interest preferences.

The traditional recommendation algorithm is usually a two-dimensional implementation of User-Item relationship. It assumes that users' interest preferences and the attributes of items are static and ignored the influence of contexts. In the real world, users' interest preferences are influenced by the surrounding environment, and different contexts may have different degrees of influence on users. So, taking into account context can improve the accuracy of determining users' interest preferences. Adomavicius et al. [1] pointed out that in addition to the User-Item interaction matrix, users' interest preferences are also influenced by different types of contexts, and integrating contexts into recommendation is beneficial to improve the performance of recommender systems. Therefore, they proposed the concept of Context-Aware Recommender Systems (CARS).

At present, most researchers have recognized the importance of contexts and attempted to incorporate relevant contexts in their research. Some believed that time is one of the situational information that cannot be ignored [2, 3]. Users'

✉ Shulin Cheng
chengshulin@shu.edu.cn

✉ Wei Jiang
42517985@qq.com

Huimin Jiang
15345229576@163.com

Wanyan Wang
wy.wang0115@gmail.com

¹ The University Key Laboratory of Intelligent Perception and Computing of Anhui Province, Anqing Normal University, Anqing 246000, Anhui, China

² School of Computer and Information, Anqing Normal University, Anqing 246000, Anhui, China

³ Anqing Data Resources Management Bureau, Anqing 246000, Anhui, China

interest preferences change over time. With the emergence of new selection, users' cognition and popularity of items would change. In addition to time, space is also a very important factor in the context. Users' interest preferences vary from region to region, especially in the location-aware recommender system, where users' interest preferences are affected by spatial location information [4–6]. Studies have shown that recommendation algorithms that consider either time or space outperform traditional recommendation algorithms. However, how to select the correct relevant context and how to integrate multiple contextual factors into the recommendation algorithm or model has become one of the main focuses of future research. In CARS, there may be a lot of contexts, and the impact of different context may be different. Therefore, it is very important to make effective use of multiple contexts for recommendation.

To address the above problems, we proposed a recommendation method based on multi-context-aware higher-order tensor factorization, which is based on detecting multiple valid contexts of users and integrating their interest preferences for multi-context-aware recommendations. Since users have different degrees of sensitivity to contexts. Firstly, we used the Chi-square test to detect user's sensitivity to contexts, multiple effective contexts were found. Then, we integrated the sensitive contexts with users' interest preferences matrix to build the multi-dimensional tensor model. At the same time, a joint tensor factorization of multiple feature matrices is constructed to alleviate data sparse. For context-insensitive users, the traditional matrix factorization method is used to predict their preferences.

2 Related work

2.1 Context-aware recommender system

Context is a very complex concept with different definitions in different application contexts. Initially of the study, Schilit et al. [7] referred to contexts as locations, the collection of nearby people and objects, as well as the changes to these objects over time. Brown et al. [8] consider contexts as location, identification of people around the user, time of day, season, temperature, etc. Dey [9] enumerates the context as the user's emotional state, concerns, location and direction, date and time, goals, people in the user's environment, etc. The current definition of context in the field of context-aware computing widely cites the concept proposed by Dey et al., who argue that a context can be information that describes the characteristics of any context of an entity. The entity here can be a person, place, or object, as long as the entity is considered relevant to the interaction between the user and the application, including the user and the application system [10].

Context-aware is one of the most important research aspects of pervasive computing, in which people are able to

access and process information at anytime, anywhere, and in any way. The context-aware enables the system to automatically discover and utilize contextual factors such as location and surroundings. The CARS introduce context into the traditional recommender system and applies many contexts such as time, location, device, and surroundings to generate recommendations for users, extending the traditional two-dimensional recommendations to the multi-dimensional recommendations. Compared with the two-dimensional relational recommendation, CARS are based on the three elements of User-Item-Context.

CARS can be expressed formally as follows.

The utility function to measure the preference of user u for item i in the CARS is given by equation 1, given that U is the set of users, I is the set of items, and C is the set of included contexts.

$$R : U \times I \times C \longrightarrow \text{Rating} \quad (1)$$

Rating is a full sequence.

One of the main issues to be studied in CARS is how to integrate context with traditional User-Item two-dimensional recommendation. Adomavicius et al. proposed three paradigms of context-aware recommendation according to which process the context is integrated in the recommendation, namely pre-context filtering, post-context filtering and context modeling [2], as shown in Fig. 1.

2.2 Matrix factorization based methods

Traditional recommender system regarded the User-Item rating matrix as the primary processing data and attempt to predict users' interest preferences for unrated items. In real life, recommender systems deal with massive data items. Matrix factorization technology is used to reduce the dimension of data so as to speed up calculation without losing important data items. Liu et al. [11] added consideration of temporal and social to the matrix factorization approach. Shi et al. [12] then used matrix factorization to

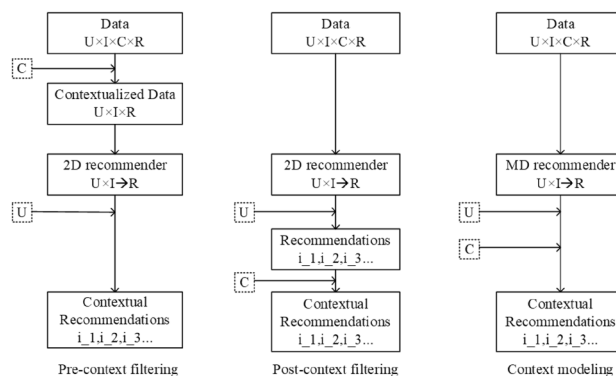


Fig. 1 Three paradigms of context-aware recommender systems

mine emotion-specific movie similarities to obtain context-aware recommendations. Baltruna et al. [13] then argued that matrix factorization with context-aware could improve the accuracy of standard matrix factorization. Zheng et al. [14] proposed a matrix factorization method based on a sparse linear method in which a user’s rating for an item is coalesced by that user’s ratings for other items. Kim et al. [15] incorporated the neural network KNN into the matrix factorization technique, also known as convolutional matrix factorization. The method applies the maximum a posteriori estimation method to optimize the parameters of the document latent vector model, the user latent vector model and the item latent vector model.

2.3 Tensor factorization based methods

It is common to refer to scalars as 0 tensors, vectors as 1st tensors, matrices as 2nd tensors, and so on to refer to multi-dimensional data as tensors. Consider the tensor of order $d \geq 3$ as a d -dimensional generalized form of the matrix. Essentially, tensor factorization is a higher-order generalization of matrix factorization that provides a flexible and versatile integration of contextual information. It does not use any post-filtering or pre-filtering techniques, which increases the significant complexity of the model.

Tucker decomposition and CP decomposition are the most commonly used methods for tensor factorization [16]. The CP decomposition is a special representation of the Tucker decomposition, which is essentially a decomposition of a tensor into a sum of tensors of finite rank 1. Suppose that given a third-order tensor $\mathcal{X} \in \mathbb{R}^{(I \times J \times K)}$, then the CP decomposition can be expressed by equation 2:

$$\mathcal{X} \approx \sum_{r=1}^R a_r \circ b_r \circ c_r \tag{2}$$

Where \circ denotes the vector outer product and R is a positive integer and $a_r \in \mathbb{R}^I, b_r \in \mathbb{R}^J, c_r \in \mathbb{R}^K$. The factorization process of the third-order tensor is shown in Fig. 2.

At present, tensor factorization is widely used in recommender systems for its advantage of being able to solve the problem of multi-dimensional data [17]. The context-aware

recommender system will inevitably lead to the problem of multi-dimensional space, and the tensor factorization technology will be more convenient to integrate context. Most of them are used to recommend points of interest. Cai et al. [18] constructed the three-dimensional tensor of User-Item-Label for label recommendation, improved the statistical information among users, items and labels by using low-order polynomials, and solved the problem of data sparse at the same time. Luan et al. [19] proposed a cooperative tensor factorization method, which utilized a three-dimensional tensor with three feature matrices to recommend points of interest. They used an element-level gradient descent optimization algorithm to solve the problem. Meanwhile, many scholars combine tensor factorization with neural network to solve the multi-type information in the CARS. Chen et al. [20, 21] proposed a model that combines tensor factorization and adversarial learning for context-aware recommendations. They combined deep neural networks and tensor algebra to capture nonlinear interactions among multi-aspect factors. Wu et al. [22] proposed a Neural network based Tensor Factorization model for predictive tasks on dynamic relational data. They argued that users’ preferences would change over time and the underlying factors driving the user project relationship would also change over time.

3 User multi-context sensitivity detection

Existing related studies have shown that there are significant differences in users’ sensitivity to different types of contexts [23], that is, users are context-sensitive. In terms of movie recommendations, some users are sensitive to their own emotions. When they are in a good mood, they will choose relaxing or cheerful movies, otherwise they will choose sad movies. We believe that in the same context with different dimensions, when there are significant changes in user interest preferences, this context is the sensitive context for the user. In this paper, the chi-square test of significance test method is used to detect whether users are sensitive to a certain context. Chi-square test is generally applicable to fitness test, independence test and unity test, and usually represents the deviation degree between the observed value and the theoretical value. The statistical value of each observed context for the corresponding preference in each dimension is used as the observation value of the Chi-square test. The average value of the user’s evaluation number in a single situation is taken as the theoretical value. The calculation formula is as in equation 3:

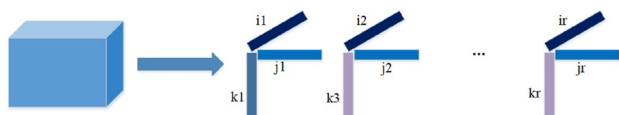


Fig. 2 Third-order tensor factorization process

$$\begin{aligned}
 X^2 &= \sum \frac{(A_i - E_i)^2}{E_i} = \sum_{i=1}^k \frac{(A_i - E_i)^2}{E_i} \\
 &= \sum_{i=1}^k \frac{(A_i - np_i)^2}{np_i} (i = 1, 2, 3, \dots, I)
 \end{aligned}
 \tag{3}$$

Where A_i is the observed count at level i , E_i is the expected count at level i , n is the total count, and p_i is the expected frequency at level i . When n is relatively large, the X^2 statistic approximately obeys a cardinal distribution with $I-1$ degrees of freedom.

Based on the LDOS-CoMoDa movie rating dataset [24], we select users who have evaluated more than 5 items and more than 10 items as test targets. The Chi-square value of each user for a single situation is calculated. If the Chi-square value is greater than the critical value compared with the critical value table of Chi-square test, it indicates that the user is sensitive to the context, otherwise it is not sensitive. Then the number of individual contexts judged as sensitive contexts is counted, and the high number is considered as user sensitive contexts. Individual contexts fall under the user sensitive context statistics as shown in Fig. 3.

In the figure, the horizontal axis indicates the results of sensitive statistics for users evaluating more than 5 and more than 10 items. Each part is represented as 12 different contexts in the dataset, including time, daytype, season, location, weather, social, endEmo, dominantEmo, mood, physical, decision, and interaction. As can be seen from the figure, the results of both parts are high in the statistics of daytype and season, then it is considered that the user sensitive context is daytype and season.

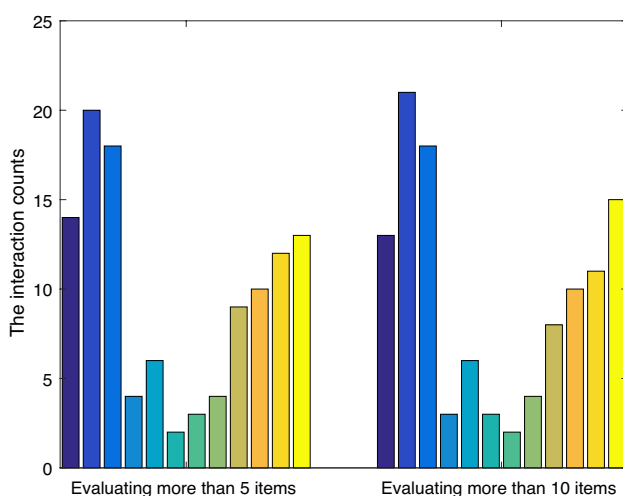


Fig. 3 Single context belongs to user sensitive context statistics (The higher the interaction count, the more likely the context is to belong to a sensitive context.)

4 Tensor factorization for multi-context-aware recommendation methods

4.1 User interest model based on four-dimensional high-order Tensor

From the user multi-context sensitivity detection, it is concluded that most users are sensitive to both daytype and season. We constructed a tensor $\mathcal{X} \in \mathbb{R}^{(U \times T \times D \times S)}$ of User-Item-Daytype-Season to represent the users' interest preferences for items in different dimension of contexts. Where U, T, D , and S denote the number of users, items, daytype dimensions and season dimensions, respectively. For ease of reading, Table 1 lists the key notations of this article.

The four modules are described in detail as follows.

Module1 (User): $U = [u_1, u_2, \dots, u_U]$ represents the presence of U different users; Module2 (Item): $T = [t_1, t_2, \dots, t_T]$ represents T different items; Module3 (Daytype): $D = [d_1, d_2, \dots, d_D]$ represents that there are D different dimensions of daytypes; Module4 (Season): $S = [s_1, s_2, \dots, s_S]$ represents that there are S different dimensions of seasons.

We use the rating of the item as an indication of the users' interest preferences, with higher ratings indicating that the user likes the item more. For each $\mathcal{X}(u, t, d, s)$ denotes the rating of user u for item t with daytype d and season s . If the user does not interact with this item in that context, then $\mathcal{X}(u, t, d, s) = 0$.

As in Fig. 4, a schematic diagram of the constructed four-dimensional tensor is shown. The constructed tensor can be represented as a combination of three-dimensional (User-Item-Daytype) tensor in different seasons.

Table 1 Notations used in the paper

Notation	Description
A_i	The observed count at level i
E_i	The expected count at level i
n	Total count
p_i	The expected frequency at level i
U, T, D, S	The set of users, items, daytype, seasons
M_1, M_2, M_3	Feature matrices
Q_j	The value of item Q in dimension j
G_j	The value of item G in dimension j
u_r, t_r, d_r, s_r	The rank-one vectors of user, item, daytype, and season
$\mathcal{X}_{(1)}, \mathcal{X}_{(2)}, \mathcal{X}_{(3)}$	Expand tensor \mathcal{X} according to different modes
λ_0	The regularization coefficient
$\lambda_1, \lambda_2, \lambda_3$	Weight of feature matrix
α	The learning rate
k	The number of hidden factors

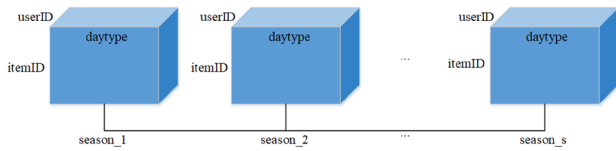


Fig. 4 Schematic diagram of four-dimensional tensor construction

4.2 Construction of feature matrices

The user is limited to the evaluation of a few items in contextual conditions, and the amount of data is extremely sparse, and populating it with zero values using only the data present in the tensor would greatly reduce the accuracy of the prediction. To reduce the data sparsity, we further constructed three feature matrices Item-Item similarity matrix, User-Daytype matrix, and User-Season matrix. And used the tensor for collaborative factorization. All three feature matrices are common to at least one dimension of the constructed four-dimensional tensor.

- (1) Item-Item similarity matrix **M1** Driven by interest, the categories in which users watch movies have greater similarity. For the same context, movies with high similarity in recommendation categories are more capable of satisfying user interest preferences. For example, if the user prefers to watch mystery movies, the system can recommend mystery movies to meet the user’s needs better than comedy movies. Therefore, in this paper, the first three attribute types (gener1, gener2, gener3) of item features are used as item category features, and the Item-Item similarity feature matrix **M1** is constructed based on the cosine similarity. The similarity between items is calculated as shown in equation 4:

$$\begin{aligned}
 \text{cosSim}(Q_j, G_j) &= \frac{Q \cdot G}{\|Q\| \|G\|} \\
 &= \frac{\sum_{j=1}^J Q_j \times G_j}{\sqrt{\sum_{j=1}^J (G_j)^2} \times \sqrt{\sum_{j=1}^J (Q_j)^2}}
 \end{aligned}
 \tag{4}$$

- (2) User-Daytype matrix **M2** According to the previous multi-context sensitivity detection, the daytype is a sensitive context for most users. Therefore, this paper constructed a User-Daytype context matrix to represent user interest preference on the daytype. To simplify the calculation, we used the average ratings of users on each dimension of the daytype to construct the User-Daytype feature matrix **M2**. An example of this matrix is as follows.

$$\begin{bmatrix}
 3.9605 & 3.6346 & 3.4285 \\
 3.8613 & 3.7258 & 4 \\
 \vdots & \ddots & \vdots \\
 3.7105 & 4.1667 & 4 \\
 4 & 3.2857 & 3.6363
 \end{bmatrix}$$

- (3) User-Season matrix **M3** Similarly, season is also a user-sensitive context through user multi-context sensitivity detection. The User-Season matrix can also reflect users’ interest preferences in different dimensions of the context from the perspective of the season. The User-Season matrix **M3** is constructed based on the average ratings of user interest preferences of the item on each dimension of the season. A partial example of this matrix is as follows.

$$\begin{bmatrix}
 3.8 & 4 & 3.5589 & 4.0588 \\
 3.9 & 3.7272 & 3.7083 & 3.9697 \\
 \vdots & \vdots & \vdots & \vdots \\
 4 & 5 & 3.6052 & 4.3846 \\
 0 & 0 & 3 & 3.6667
 \end{bmatrix}$$

4.3 Recommendation methods for multi-context-aware

4.3.1 Context-aware collaborative Tensor factorization

The ultimate goal of both tensor factorization and matrix factorization is to fill in the missing items based on the existing data items. Tucker decomposition and CP decomposition are the most commonly used methods in tensor factorization [16]. CP decomposition is a special representation of Tucker decomposition, which can be applied to massive data and is more convenient to calculate. Therefore, we used CP decomposition. The tensor $\mathcal{X} \in \mathbb{R}^{(U \times T \times D \times S)}$ constructed in our experiments can be decomposed into $U \in \mathbb{R}^{(u \times k)}, T \in \mathbb{R}^{(t \times k)}, D \in \mathbb{R}^{(d \times k)}, S \in \mathbb{R}^{(s \times k)}$. The expression for its decomposition is given in equation 5.

$$\mathcal{X} \approx [U, T, D, S] = \sum_{r=1}^R \lambda_r \circ u_r \circ t_r \circ d_r \circ s_r
 \tag{5}$$

Where **U**, **T**, **D**, and **S** are called factor matrices and are combinations of rank-one vectors. $u_r \in \mathbb{R}^U, t_r \in \mathbb{R}^T, d_r \in \mathbb{R}^D, s_r \in \mathbb{R}^S (r=1, 2, \dots, R)$ denote the rank-one vectors of user, item, daytype, and season, respectively. R is a positive integer that represents the number or rank of components. Usually, we correspond the rows of the factor matrix to each dimension of the tensor, and the columns to the rank R .

The three constructed feature matrices **M1**, **M2**, and **M3** can be decomposed into two smaller matrices using matrix

factorization, respectively. In general, the factorization of the matrix can be expressed by equation 6.

$$M = YZ^T (M \in R^{n \times m}, Y \in R^{n \times k}, Z \in R^{m \times k}) \tag{6}$$

Where k denotes the rank of the factorization and represents the matrix containing k implied features. We considered that users' interest preferences for items are mainly determined by k hidden features.

To implement the calculation between the tensor and the matrix, it is first necessary to transform the tensor matrix into the same dimension as the matrix. For a third-dimension tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ can be expanded in three modules. Accordingly, each modal expansion yields a matrix, as shown in equation 7–9:

$$\mathcal{X}_{(1)} = [\mathcal{X}(:, :, 1), \mathcal{X}(:, :, 2), \dots, \mathcal{X}(:, :, n_3)] \in R^{n_1 \times (n_2 n_3)} \tag{7}$$

$$\mathcal{X}_{(2)} = [\mathcal{X}(:, :, 1)^T, \mathcal{X}(:, :, 2)^T, \dots, \mathcal{X}(:, :, n_3)^T] \in R^{n_2 \times (n_1 n_3)} \tag{8}$$

$$\mathcal{X}_{(3)} = [\mathcal{X}(:, 1, :)^T, \mathcal{X}(:, 2, :)^T, \dots, \mathcal{X}(:, n_2, :)^T] \in R^{n_3 \times (n_1 n_2)} \tag{9}$$

The created feature matrix **M1** shares the item dimension with tensor \mathcal{X} . **M2** shares the user and daytype dimensions with the tensor. And **M3** shares the user and season dimensions with the tensor. Thus, the data in the matrix can be fused into the tensor by a shared matrix dimension. Given the tensor $\mathcal{X} \in \mathbb{R}^{(U \times T \times D \times S)}$ and the feature matrices **M1**, **M2**, **M3**, construct the objective function as shown in equation 10.

$$F(U, T, D, S) = \frac{1}{2} \|\mathcal{X} - U \circ T \circ D \circ S\|_F^2 + \frac{\lambda_1}{2} \|M_1 - TT^T\|_F^2 + \frac{\lambda_2}{2} \|M_2 - UD^T\|_F^2 + \frac{\lambda_3}{2} \|M_3 - US^T\|_F^2 + \frac{\lambda_0}{2} (\|U\|_F^2 + \|T\|_F^2 + \|D\|_F^2 + \|S\|_F^2) \tag{10}$$

Where $\|\dots\|$ denotes finding its two-parametric number and $\frac{1}{2} \|\mathcal{X} - U \circ T \circ D \circ S\|_F^2$ denotes the least square error loss function that decomposes the four-dimensional tensor into four factor matrices **U**, **T**, **D**, and **S**. Then this part of the formula can be further expressed as equation 11.

$$F = \frac{1}{2} [\mathcal{X}_{(1)} - U(S \circ D \circ T)^T]^2 = \frac{1}{2} [\mathcal{X}_{(2)} - T(S \circ D \circ U)^T]^2 = \frac{1}{2} [\mathcal{X}_{(3)} - D(S \circ T \circ U)^T]^2 = \frac{1}{2} [\mathcal{X}_{(4)} - S(D \circ T \circ U)^T]^2 \tag{11}$$

Tensor factorization can usually be computed using Alternating Least Squares (ALS) and Gradient Descent (GD), and considering their limitations, we used Stochastic Gradient Descent (SGD) for optimization. To implement the multiplication of a tensor and a matrix, the tensor needs to be matrixed first. Matricization is the rearrangement of the elements of an n-dimensional array into a matrix. Matricization of a tensor means transforming the tensor into a matrix from different dimensions of the tensor. The product is calculated by multiplying the matrix formed by the n-dimensional matricization by the matrix. The derivative of our four-dimensional tensor for the matrix **U** when $n = 1$ is chosen is given in equation 12.

$$\frac{\partial F}{\partial U} = 2 \times \frac{1}{2} (\mathcal{X}_{(1)} - U(S \circ D \circ T)^T) \times \frac{\partial [\mathcal{X}_{(1)} - U(S \circ D \circ T)^T]}{\partial U} = -[\mathcal{X}_{(1)} - U(S \circ D \circ T)^T](S \circ D \circ T) \tag{12}$$

Then, for the **U** matrix in tensor \mathcal{X} can be updated based on the following equation 13, where α denotes the learning rate.

$$U = U + \alpha [\mathcal{X}_{(1)} - U(S \circ D \circ T)^T](S \circ D \circ T) \tag{13}$$

By the same token, **T**, **D**, and **S** are updated as follows in equation 14–16.

$$T = T + \alpha [\mathcal{X}_{(2)} - T(S \circ D \circ U)^T](S \circ D \circ U) \tag{14}$$

$$D = D + \alpha [\mathcal{X}_{(3)} - D(S \circ T \circ U)^T](S \circ T \circ U) \tag{15}$$

$$S = S + \alpha [\mathcal{X}_{(4)} - S(D \circ T \circ U)^T](D \circ T \circ U) \tag{16}$$

$\|M_1 - TT^T\|_F^2$ denotes the least square error of decomposing the **M1** matrix into **T** and **T** matrices. $\|M_2 - UD^T\|_F^2$ denotes the least square error of decomposing the **M2** matrix into **U** and **D** matrices. $\|M_3 - US^T\|_F^2$ denotes the least square error of decomposing the **M3** matrix into **U** and **D** matrices.

$\frac{\lambda_0}{2} (\|U\|_F^2 + \|T\|_F^2 + \|D\|_F^2 + \|S\|_F^2)$ is a regularization term to prevent overfitting. λ_0 is the regularization coefficient, and $\lambda_1, \lambda_2, \lambda_3$ are the model parameters controlling the weights of different parts of the objective function.

The learning process of this algorithm is shown in Algorithm 1, where the input is a four-dimensional tensor \mathcal{X} with three feature matrices **M1**, **M2**, and **M3**. In the algorithm, the four-factor matrix is first initialized using the minimum random value, then the optimal values of the parameters are learned using the stochastic gradient

descent method, and finally the dense four-factor matrix is output.

Algorithm 1 Tensor joint matrix factorization algorithm

Input: original tensor \mathcal{X} , identity matrices M1, M2, M3

Output: factor matrices U, T, D, S

- 1: Initialization of the factor matrix using the minimum random value according to the size of each dimension of the tensor \mathcal{X}
- 2: Iteration starts, set the difference between loss values β less than 0.0001 stop
- 3: Calculate each gradient as follows:
- 4: $\nabla_U F = - \left[\mathcal{X}_{(1)} - U(S \odot D \odot T)^T \right] (S \odot D \odot T) + \lambda_2 (M_2 - UD^T) + \lambda_3 (M_3 - US^T) + \lambda_0 U$
- 5: $\nabla_T F = - \left[\mathcal{X}_{(2)} - T(S \odot D \odot U)^T \right] (S \odot D \odot U) + \lambda_1 (M_1 - TT^T) + \lambda_0 T$
- 6: $\nabla_D F = - \left[\mathcal{X}_{(3)} - D(S \odot T \odot U)^T \right] (S \odot T \odot U) + \lambda_2 (M_2 - UD^T) + \lambda_0 D$
- 7: $\nabla_S F = - \left[\mathcal{X}_{(4)} - S(D \odot T \odot U)^T \right] (D \odot T \odot U) + \lambda_3 (M_3 - US^T) + \lambda_0 S$
- 8: Update the values according to the new gradient:
- 9: $U_{i+1} = U_i - \alpha_i \nabla_U F$
- 10: $T_{i+1} = T_i - \alpha_i \nabla_T F$
- 11: $D_{i+1} = D_i - \alpha_i \nabla_D F$
- 12: $S_{i+1} = S_i - \alpha_i \nabla_S F$
- 13: End the iteration according to the stop condition
- 14: Return U, T, D, S

The complexity of the algorithm is analyzed below. Assuming that the tensor $\mathcal{X} \in \mathbb{R}^{(n \times n \times n \times n)}$, where R denotes the rank of \mathcal{X} , it is known that $S \in \mathbb{R}^{(n \times R)}, D \in \mathbb{R}^{(n \times R)}, T \in \mathbb{R}^{(n \times R)}$, then the time complexity of computing $(S \odot D \odot T)^T$ is about $O(n^3 R)$; obviously, $\mathcal{X}_{(1)} \in \mathbb{R}^{(n \times n^3)}$ and $(S \odot D \odot T)^T \in \mathbb{R}^{(n \times n^3)}$, then the computation of $[\mathcal{X}_{(1)} - U(S \odot D \odot T)^T](S \odot D \odot T)$ is approximately $O(n^4 R)$. The time complexity of the algorithm is about $O(n^4 R)$ when the value of the rank R in the calculation is taken to be smaller. The algorithm needs to store the matrix of four-dimensional tensor expansion, the feature matrix and the associated product values, then the space complexity is about $O(n^4)$.

4.3.2 Matrix factorization

In this paper, context-insensitive users are predicted using the traditional matrix factorization method [25] for User-Item ratings. The constructed User-Item rating matrix is denoted as \mathbf{M} , where M(u, t) ratings the users' interest preferences of the item, and the prediction model corresponds to the objective function in equation 17.

$$F(U, T) = \frac{1}{2} \|M - UT^T\|_F^2 + \frac{\lambda_1}{2} \|U\|_F^2 + \frac{\lambda_2}{2} \|T\|_F^2 \quad (17)$$

4.3.3 Top-N recommends

In this paper, the recommendation method used the Top-N recommendation strategy. In the corresponding contexts, users sensitive to multiple contexts use the outer product of the output four-factor matrix to recover the sparse tensor based on

the higher-order tensor factorization, and users not sensitive to contexts use the matrix factorization method for the scoring prediction array. The reconstructed tensor \mathcal{X}_{new} , matrix M_{new} expressions are as in equation 18–19.

$$\mathcal{X}_{new} = U \odot T \odot D \odot S \quad (18)$$

$$M_{new} = UT^T \quad (19)$$

Where $\mathcal{X}_{new}(u, t, d, s)$ is the preference pre-rating of context-sensitive user u for item t in the context with daytype d and season s , and $M_{new}(u, t)$ is the preference rating of context-insensitive user u for item t . Finally, the Top-N items are recommended for different users by pre-rating the location items in reverse order.

5 Experiments

5.1 Multi-context movie dataset

The experimental dataset in this paper was chosen from the real movie dataset LDOS-CoMoDa, collected by Prof. Ante Odi'c, which contains multiple contexts [24]. The dataset not only contains basic information about users and items such as age, sex, city, country, director, country, language, year, genre, actor, budget, but also collects information about 12 contexts. The relevant information of the dataset is shown in the following Table 2.

The 12 contexts in the dataset with different categories are time, daytype, season, location, weather, social, endEmo, dominantEmo, mood, physical, decision, and interaction. The specific description of each context is shown in Table 3.

5.2 Baseline methods

- (1) Standard-CP [26]: Only the four-dimensional tensor is used as input, and the influence of feature lifting on it is not considered. By setting $\lambda_1, \lambda_2, \lambda_3$ in the determined objective function to zero can be obtained as in equation 20.

$$F(U, T, D, S) = \frac{1}{2} \|\mathcal{X} - U \odot T \odot D \odot S\|_F^2 + \frac{\lambda_0}{2} (\|U\|_F^2 + \|T\|_F^2 + \|D\|_F^2 + \|S\|_F^2) \quad (20)$$

- (2) HOSVD [27]: HOSVD is a common method applied to tensor factorization, which is to fill the data after

Table 2 Statistics of the dataset

Total	User	Item	Rating	Ave-rating
2296	121	1232	1–5	3.83

Table 3 Description of situational factors

Context	Dimensions
Time	Morning (1), afternoon (2), evening (3), night (4)
Daytype	Working day (1), weekend (2), holiday (3)
Season	Spring (1), summer (2), autumn (3), winter (4)
Location	Home (1), public place (2), friend’s house (3)
Weather	Sunny / clear (1), rainy (2), stormy (3), snowy (4), cloudy (5)
Social	Alone (1), my partner (2), friends (3), colleagues (4), parents(5),public (6), my family (7)
EndEmo	Sad (1), happy (2), scared (3), surprised (4), angry (5), disgusted (6), neutral (7)
DominantEmo	Sad (1), happy (2), scared (3), surprised (4), angry (5), disgusted (6), neutral (7)
Mood	Positive (1), neutral (2), negative (3)
Physical	Healthy (1), ill (2)
Decision	User decided which movie to watch (1), user was given a movie (2)
Interaction	First interaction with a movie (1), n-th interaction with a movie (2)

decomposing the tensor by different modes and then using the SVD method in turn. It is often used in contextual recommender systems because of its applicability to higher-order data.

- (3) NMF [28]: The method of non-negative matrix factorization is often used for the recommendation of two-dimensional data items and, like traditional SVD, does not consider the effect of context on the results.

5.3 Evaluation Metrics

We used the classical Root Mean Square error (RMSE) and Mean Absolute Error (MAE) as evaluation metrics. RMSE and MAE are used to measure the deviation between the actual and predicted values and are calculated as in equation 21–22.

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (\mathcal{X}_{(u,t,d,s)} - \mathcal{X}'_{(u,t,d,s)})^2}{N}} \tag{21}$$

$$MAE = \frac{\sum_{i=1}^N |\mathcal{X}_{(u,t,d,s)} - \mathcal{X}'_{(u,t,d,s)}|}{N} \tag{22}$$

$\mathcal{X}_{(u,t,d,s)}$, $\mathcal{X}'_{(u,t,d,s)}$ denote the actual rating and predicted rating, respectively, and N denotes the number of predicted ratings.

5.4 Experiment results

5.4.1 Parameter optimization

We first conducted optimization experiments on the parameters of the multi-context-aware recommendation method. The parameters to be optimized include the learning rate α and $\lambda_0, \lambda_1, \lambda_2, \lambda_3$. In order to ensure the accuracy of parameter optimization and prevent overfitting caused by the complexity of the model, we adopt a three-fold cross-validation

experimental method to calculate the experimental results. Both context-sensitive and context-insensitive users selected 80% of the random data as the training set and the remaining 20% as the test set.

- (1) Optimization of learning rate α When optimizing the learning rate α , it is necessary to ensure that $\lambda_0, \lambda_1, \lambda_2, \lambda_3$ take relatively fixed values. The experimental results for different learning rates α are shown in Fig. 5. It can be seen from the figure, as the value of α increases both RMSE and MAE decrease first and then increase. When the value of α is $1e-4$, both RMSE and MAE are minimal, and the recommendation model reaches the relatively optimal results.
- (2) Optimization of $\lambda_0, \lambda_1, \lambda_2, \lambda_3$ The parameters $\lambda_0, \lambda_1, \lambda_2, \lambda_3$ control the feature matrix and the degree of regularization influence, which have a large impact on the recommendation accuracy. We chose 0.2–1.6

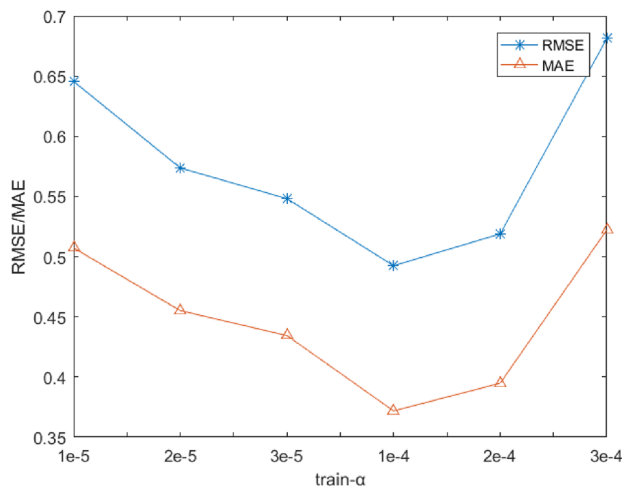


Fig. 5 Effect of learning rate α on the model

step 0.2 as the value of each parameter, and the specific degree of impact is shown in Fig. 6. From Figure 6, we can see that the model can achieve better results when λ_0 is 0.4, λ_1 is 0.4, λ_2 is 1.2, and λ_3 is 1.2. The values chosen for λ_2 and λ_3 are larger than those for λ_0 and λ_1 because λ_2 and λ_3 control the degree of influence of daytype and season.

When performing matrix factorization for context-insensitive users, the number of hidden factors k has a large impact on their results, and we choose the optimal value of k by the RMSE. It can be seen from Fig. 7 that RMSE decreases and flattens out as the value of k increases, and the RMSE is minimal when $k = 14$ in the selected interval.

5.4.2 Method Comparison

Based on the parameter optimization, the proposed method was compared with three baseline methods (Standard-CP, HOSVD and NMF), and the experimental results on RMSE and MAE are shown in Fig. 8.

It can be seen from the figure that the NMF has higher values on both RMSE and MAE than the other methods. Since it only considers users' interest preferences and does not take into account contexts. Standard-CP and HOSVD are the most commonly used tensor factorization methods, and their values are closer. Our method adds the feature matrix based on tensor factorization to alleviate the problem of data sparsity, and the values of RMSE and MAE are 0.4765 and 0.3988, which are 5.09% and 5.32% lower than optimal HOSVD. Therefore, the method proposed

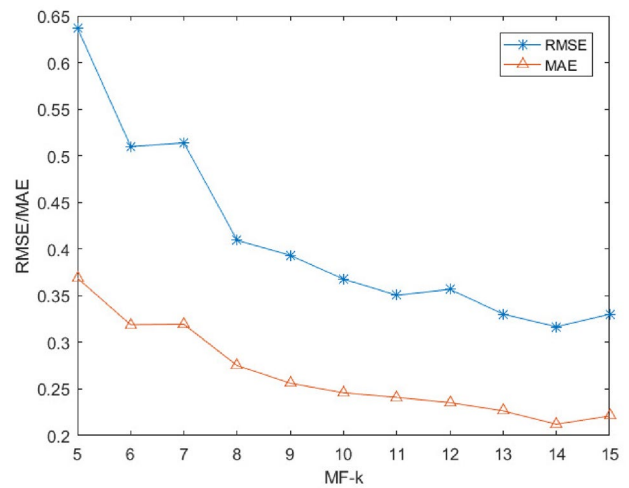


Fig. 7 Effect of hidden factor k on MF

in this paper can reduce the recommendation error to a certain extent and effectively improve the recommendation accuracy.

6 Conclusions

In the context-aware recommender system, making full use of multiple context information can effectively improve the accuracy of recommendation. In this paper, we proposed a recommendation method based on multi-context-aware higher-order tensor factorization. Firstly, users' sensitivities to the contexts were tested, and then users were divided into two categories. For the context-sensitive users, a four-dimensional tensor was constructed to simulate the relationship between users, items, daytypes and seasons.

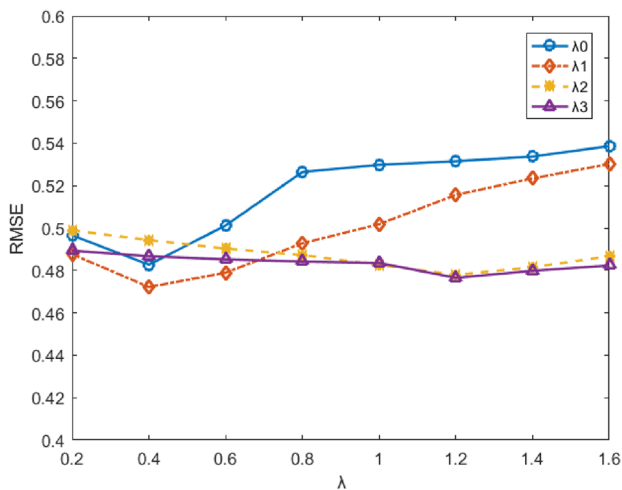


Fig. 6 Effect of different parameters on the model

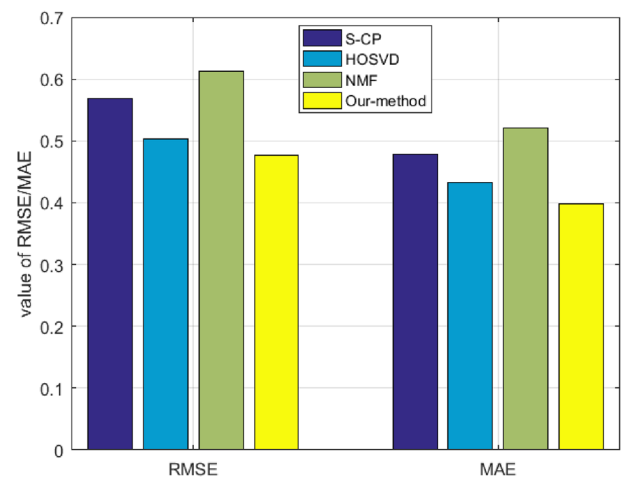


Fig. 8 Comparison with baseline method

Further, three feature matrices were constructed combined with different dimensions to solve the problem of data sparsity. Compared with the standard tensor factorization, high-order matrix decomposition and traditional matrix decomposition methods, this method has higher accuracy and better recommendation results. In the future work, we will deeply study the multi-context-aware recommender system from the influence difference of each dimension of high-order tensor.

Acknowledgements The work was supported by grants from the Nature Science Foundation of Anhui Province in China, No.2008085MF193, the Outstanding Young Talents Program of Anhui Province in China, No.gxyqZD2018060, the Major Science and Technology Project of Anhui Province, No.201903a06020006, the Provincial Quality Project of the Anhui Province Education Department in China, No.2019jyxm0285. And we thank all the anonymous reviewers for their hard work and valuable comments.

References

- Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions[J]. *IEEE Trans. Knowl. Data Eng.* **17**(6), 734–749 (2005). <https://doi.org/10.1109/TKDE.2005.99>
- Guo, Y., Ling, Y., Chen, H.: A Time-Aware Graph Neural Network for Session-Based Recommendation[C], *IEEE Access*, **8**, 167371–16738 (2020). <https://doi.org/10.1109/ACCESS.2020.3023685>
- Xian, J., Qin, Z., Sun, L.: An optimization of collaborative filtering personalized recommendation algorithm based on time context information. Springer International Publishing (2015). <https://doi.org/10.1007/978-3-319-16274-4-15>
- Yin, H., Cui, B., Sun, Y., et al.: LCARS: a spatial item recommender system. *Acem Trans. Inf. Syst.* **32**(3), 11–11 (2014). <https://doi.org/10.1145/2629461>
- Wang, F., Li, D., Xu, M.: A location-aware TV show recommendation with localized semantic analysis. *Multimed. Syst.* **22**(4), 535–542 (2016). <https://doi.org/10.1007/s00530-015-0451-z>
- Lu, J., Indeche, M.A.: Multi-context-aware location recommendation using Tensor decomposition. *IEEE Access* (2020). <https://doi.org/10.1109/ACCESS.2020.2983555>
- Schilit, B.N., Theimer, M.M.: Disseminating active map information to mobile hosts. *IEEE Netw.* **8**(5), 22–32 (1994). <https://doi.org/10.1109/65.313011>
- Brown, P.J., Bovey, J.D., Chen, X.: Context-aware applications: from the laboratory to the marketplace. *IEEE Pers. Commun.* **4**(5), 58–64 (1997). <https://doi.org/10.1109/98.626984>
- Dey, A.K.: Context-aware computing: the cyber desk project. AAAI Press, (1998)
- Abowd, G.D., Dey, A.K., Brown, P.J., et al.: Towards a better understanding of context and context-awareness. Springer, Berlin Heidelberg (1999)
- Liu, N.N., Cao, B., Zhao, M., et al.: Adapting neighborhood and matrix factorization models for context aware recommendation[M]. ACM, (2010)
- Shi, Y., Hanjalic, A., Larson, M.: Mining mood-specific movie similarity with matrix factorization for context-aware recommendation[J]. In Proceedings of the Workshop on Context-Aware Movie Recommendation (CAMRa'10). ACM, New York, NY, 34–40 (2010)
- Baltrunas, L., Ludwig, B., Ricci, F.: Matrix factorization techniques for context aware recommendation, in: Proceedings of the Fifth ACM conference on recommender systems, ACM, pp. 301–304 (2011)
- Yong, Z., Mobasher, B., Burke, R.: CSLIM: contextual SLIM recommendation algorithms[C]// ACM, conference on recommender systems. ACM, (2014)
- Kim, D., Park, C., Oh, J., Lee, S., Yu, H.: Convolutional matrix factorization for document context-aware recommendation, in: Proceedings of the 10th ACM conference on recommender systems, ACM, pp. 233–240 (2016)
- Kolda, T.: Tensor decompositions and applications. *Siam Rev.* (2009). <https://doi.org/10.1137/07070111X>
- Hong, M., Jung, J.J.: Multi-Sided recommendation based on social tensor factorization. *Inf. Sci.* (2018). <https://doi.org/10.1016/j.ins.2018.03.019>
- CAI, Y., ZHANG, M., LUO, D., et al.: Low-order tensor decompositions for social tagging recommendation[C]//The forth ACM international conference on web search and web data mining. ACM, 695–704. (2011) <https://doi.org/10.1145/1935826.1935920>
- Luan, W., Liu, G., Jiang, C.: Collaborative tensor factorization and its application in POI recommendation, 2016 IEEE 13th international conference on networking, sensing, and control (ICNSC), pp. 1–6 (2016)
- Chen, H., Li, J.: Neural Tensor model for learning multi-aspect factors in recommender systems[C]// International joint conference on artificial intelligence. International joint conferences on artificial intelligence organization, (2020)
- Chen, H., Li, J.: Adversarial tensor factorization for context-aware recommendation[C]// The 13th ACM conference. ACM, (2019)
- Xian, W., Shi, B., Dong, Y., et al.: Neural Tensor factorization for temporal interaction learning[C]// The twelfth ACM international conference. ACM, (2019)
- Chen, S.L., Zhang, B.F., et al.: Valid context detection based on context filter in context-aware recommendation system, (2018)
- Kovsir, A., Odi' C, A., Kunaver, M., et al.: Database for contextual personalization. *Electrotech. Rev.* **78**(5), 270–274 (2011)
- Paterek, A.: Improving regularized singular value decomposition for collaborative filtering, pp. 39–42. ACM Press, Proc. KDD Cup and Workshop (2007)
- Xu, Y., Hao, R., Yin, W., et al.: Parallel matrix factorization for low-rank tensor completion. *Inverse Probl. Imaging* **9**(2), 601–624 (2015). <https://doi.org/10.1007/s11464-012-0194-5>
- Symeonidis, P.: ClustHOSVD: item recommendation by combining semantically enhanced Tag clustering with Tensor HOSVD. *IEEE Trans. Syst. Man Cybern. Syst.* **46**(9), 1240–1251 (2016). <https://doi.org/10.1109/TSMC.2015.2482458>
- Lin, C.J.: On the convergence of multiplicative update algorithms for nonnegative matrix factorization. *IEEE Trans. Neural Netw.* **18**(6), 1589–1596 (2007). <https://doi.org/10.1109/TNN.2007.895831>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.