A Film Recommendation Algorithm Based On Tensor Decomposition

Yulong Han, Junfeng Liang, Jingyi Liu and Ruxuan Wang

June 10, 2019

A Film Recommendation Algorithm Based On Tensor Decomposition

Han Yulong, Junfeng Liang, Jingyi Liu and Ruxuan Wang

North China University of Technology

Beijing, China

***Abstract*-- In today's increasingly complex Internet information, the emergence of a recommendation system provides a list of information for users with unclear goals. The recommenddation system belongs to the category of information filtering systems. When a large amount of useful information comes together, data mining and prediction algorithms are used to predict the user's possible preferences. Algorithms based on project recommenddation, content recommendation and integration are the main recom-mended methods used by people.**

The recommendation algorithm is the core of the recommendation system. The commonly used general recommendation algorithms are content-based collaborative filtering algorithms and hybrid recommendation algorithms. This article uses a new algorithm based on traditional algorithms, that is, a three-dimensional tensor is used to describe the potential correlation model of three entities: user, item, and tag. Based on the use of primary metadata to construct the initial high-order tensor, the high-order singular value decomposition (HOSVD) is applied to the tensor. The dimensionality reduction is performed to correlate the latent semantics among the three types of entities and improve the accuracy of the tag recommendation system.

In order to verify the effectiveness of the algorithm used in this paper, we selected a certain amount of data in the real database to test the algorithm. The test results show that the proposed method has significantly improved the RMS error performance index than the item-based recommendation algorithm.

***Keywords*-- Recommended system; Tensor decomposition;**
Tag recommendation; High–order singular value decomposition

I. INTRODUCTION

The recommendation technology consists of three parts: the interactive page of the foreground, the log system of the background, and the algorithm system. The core is an algorithm system that can automatically provide personalized recommendations based on the historical records of user activities. These activities are usually represented by user-project two-dimensional data. The main algorithms currently applied include content-based, collaborative filtering and hybrid recommendation algorithms, as well as recommendation algorithms based on data mining such as association rules and application clustering[1].

In this paper, by constructing a user-item-label three-dimensional model[2], the data is subjected to three-dimensional analysis of {user, item, label} to improve the accuracy of recommendation, and a three-dimensional tensor model is proposed, which uses three dimensions of tensor to describe Users, items, and tags are collected in three different physical objects, and they are looking for possible relationships. They use high-order singular value decomposition (HOSVD) to capture the relationship between them and use them to solve the data sparsity problem by least squares.

II. RECOMMENDED ALGORITHM

Karatzoglou et al.[3] proposed a tensor decomposition recommendation algorithm that constructs user, project, and context information as a user-project-context third-order tensor. Through the tensor decomposition algorithm, the user's preference for the project in different context information is obtained.

In order to explain the realization of tensor dimensionality reduction and to explore the potential semantic association between three types of entities, we illustrate by example, as shown in Figure 2.1, there are 5 users who scored 5 different items. The arrow between the user and the movie indicates that the user has given a label to the corresponding movie (in this case, the rating). The arrow between the movie and the rating indicates that the movie has been scored by the user, and the number on the arrow indicates the relationship chain between the user, the movie, and the rating.

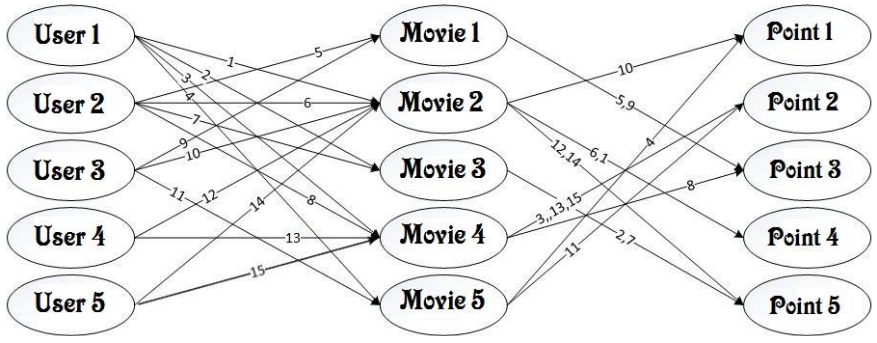


Figure II.1 Input simulation of the tag's original metadata

The following is a three-dimensional tensor constructed from the tag metadata, and the user in the triad scores the items under each tag as elements of tensor A, as listed in Table 2.1.

Table II.1 Initial tensor A constructed from tag metadata in the example

	User	Movie	Tag	Associated weight
1	U1	M2	4	1
2	U1	M3	5	1
3	U1	M4	2	1
4	U1	M5	1	1
5	U2	M1	3	1
6	U2	M2	4	1
7	U2	M3	5	1
8	U2	M4	3	1
9	U3	M1	3	1
10	U3	M2	1	1
11	U3	M5	2	1
12	U4	M2	5	1
13	U4	M4	2	1
14	U5	M2	5	1
15	U5	M4	2	1

The approximate tensor A' needs to be obtained by a tensor decomposition algorithm, as shown in Table 2.2 and Figure 2.2. Table 2.2 describes the predicted scores in the approximate tensor A', and Figure 2.2 visually describes the relationship between users, items, and labels in the approximate tensor A'. After applying the tensor decomposition algorithm, we can see that the output of the algorithm newly generates five associated data links {U1, M1, T4}, {U3, M3, T2}, {U3, M4, T2}, {U4, M1, T4}. And {U5, M1, T5}, as shown in bold text in lines 16-20 of Table 2.2, and in the dotted line position in Figure 2.2

Table II.2 Approximate tensor A' reconstructed from the tag metadata in the example

	User	Movie	Tag	Associated weight
1	U1	M2	4	1
2	U1	M3	5	1
3	U1	M4	2	1

4	U1	M5	1	1
5	U2	M1	3	1
6	U2	M2	4	1
7	U2	M3	5	1
8	U2	M4	3	1
9	U3	M1	3	1
10	U3	M2	1	1
11	U3	M5	2	1
12	U4	M2	5	1
13	U4	M4	2	1
14	U5	M2	5	1
15	U5	M4	2	1
16	**U1**	**M1**	**4**	**0.392**
17	**U3**	**M3**	**2**	**0.210**
18	**U3**	**M4**	**2**	**0.688**
19	**U4**	**M1**	**4**	**0.640**
20	**U5**	**M1**	**5**	**0.392**

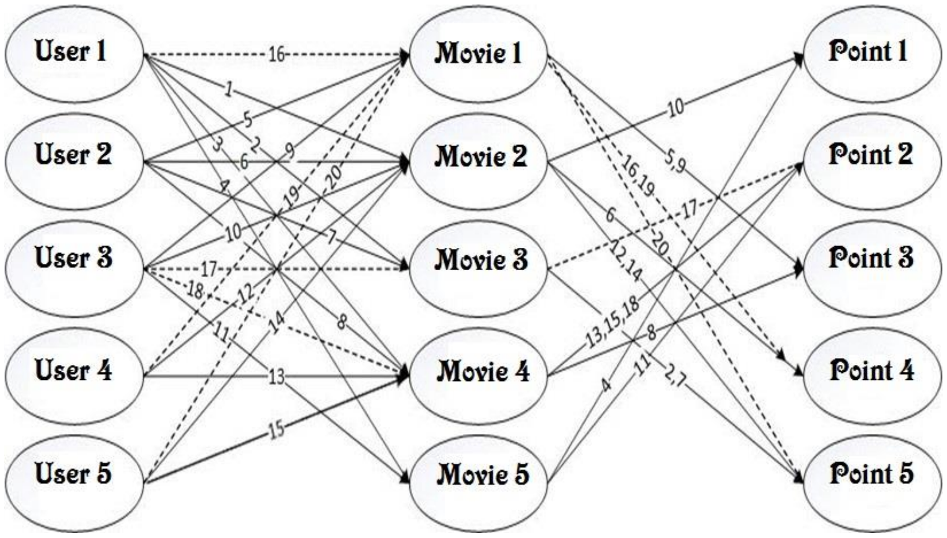


Figure II.2 Simulation simulation based on tensor decomposition algorithm

Based on the newly generated associated data, the system automatically recommends movie M1 to users U1, U4 and U5. Since there is no user U1 and item M1, user U4 and item M1, and associated data between user U5 and item M1 in the original tensor A, the data in Table 2.2 clearly indicates that after applying the tensor decomposition algorithm, the system Successfully utilizing the potential association between users, items and tags, and predicting the rating of user M1 for movie M1, user U4 scores for movie M1 for movie M1 and user U5, and there is no user U1 for item M1 in the data. The user U4 records the other items on the item M1 for the item M1 and the user U5. Therefore, the movie M1 will be recommended to the users U1, U4 and U5 by the system.

The results prove that the tensor decomposition algorithm can successfully mine the potential semantic association between multiple types of entities (users, items, tags), thus improving the performance of the tag recommendation system.

III. TENSOR DECOMPOSITION

The tensor decomposition algorithm first constructs an initial tensor A based on the tag metadata triplet {u,i,t} of the user, the item and the tag, including all users in the recommendation system, between the project and the tag. All associated data; secondly, the tensor A is expanded by n-module matrix using HOSVD to generate three new matrices A1, A2 and A3. Then, the two-dimensional matrix singular value decomposition (SVD) calculation is performed on the A1, A2 and A3 matrices respectively, and finally a new core tensor is constructed and the estimated tensor A' is calculated. The method calculation steps are as follows:

Step 1 constructs tensors ***A*** based on tag data {user, item, tag};

Step 2 According to the HOSVD method, the tensor A is expanded in a 1 to 3 mode n-modulus matrix to produce three expansion matrices A_1, A_2, A_3;

Step 3 Apply singular value decomposition based on SVD calculation on matrix expansion A_1, A_2 and A_3 to obtain three corresponding left singular matrices $U^{(1)}$, $U^{(2)}, U^{(3)}$;

Step 4 construct core tensor ***S***;

$$S = A \times_1 (U_{c1}^{(1)})^T \times_2 (U_{c2}^{(2)})^T \times_3 (U_{c3}^{(3)})^T \tag{3. 1}$$

Step 5 constructs an approximate tensor ***A'***;

$$A' = S \times_1 U_{c1}^{(1)} \times_2 U_{c2}^{(2)} \times_3 U_{c3}^{(3)} \tag{3. 2}$$

Step 6 recommends the label to the user based on the associated weight in the approximate tensor A'. The first five steps are the process of building the model, which needs to be implemented offline; the sixth step needs to be implemented online.

The details of each step of the algorithm are described in detail in the following subsections.

III. I Tensor construction

This paper mainly discusses the construction of the third-order tensor. Firstly, the user constructs the scoring matrix of the article under the five labels, and then the matrix is spliced into a 2-module expansion matrix. Finally, the matrix is expanded by the inverse matrix, so that the expansion matrix is realized in a folded form. The construction of the tensor.

III.II N module expansion

According to the following matrix expansion formula, the initial *A* tensor constructed in the image is expanded into three matrices *A1, A2, A3*.

$A_1 \in \mathbb{R}^{I_u \times I_i I_t}$ （1-module）， $A_{(1)} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$ ；

$A_2 \in \mathbb{R}^{I_i \times I_u I_t}$ （2-module）， $A_{(2)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$ ；

$A_3 \in \mathbb{R}^{I_u I_i \times I_t}$ （3-module）， $A_{(3)} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$ 。

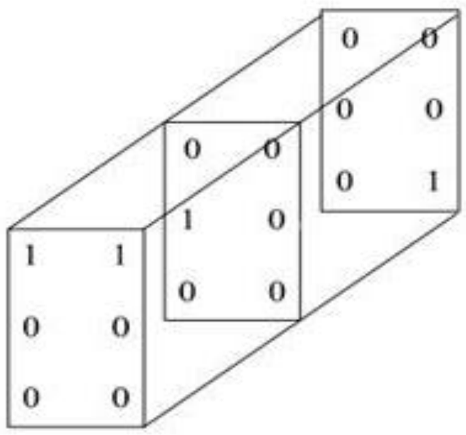


Figure 3.1　Initial tensor ***A***

III.III Singular value decomposition of the expanded matrix

On three matrix expansions **A1**，**A2**，**A3**, SVD decomposition is performed according to formulas (3.3), (3.4), and (3.5), Dimensionality reduction and noise reduction for the matrix, resulting in a left singular matrix $U^{(1)}$, $U^{(2)}$, $U^{(3)}$ and it's Corresponding Singular value diagonal matrix S_1, S_2, S_3。

$$A_1 = U^{(1)} S_1 V_1^T \tag{3. 3}$$

$$A_2 = U^{(2)} S_2 V_2^T \tag{3. 4}$$

$$A_3 = U^{(3)} S_3 V_3^T \tag{3. 5}$$

Since the dimension of the core tensor S is ultimately determined by the dimensionality parameter $c_i (i \in [1,3])$ of the left singular matrix, Therefore, it is necessary to adjust the initial diagonal matrix S_1, S_2, S_3, determining the dimensionality parameters c_1, c_2, c_3 of the left singular matrix in the three modes (1 mode, 2 mode, 3 mode) by filtering out the sparse noise data in the original matrix (by default retaining 80% of the original matrix information).

III. IV Constructing core tensors

The core tensor controls the interaction between users, items, and tags. Therefore, we need to substitute the matrix $U^{(1)}$, $U^{(2)}$, $U^{(3)}$ after the rank reduction, and then into the formula(3.6) to construct the approximate core tensor $\boldsymbol{S'}$.

$$S = A \times_1 (U_{c1}^{(1)})^T \times_2 (U_{c2}^{(2)})^T \times_3 (U_{c3}^{(3)})^T \tag{3. 6}$$

Where $\boldsymbol{A}$ is the initial tensor in this example, And $U^{(1)^T}$, $U^{(2)^T}$, $U^{(3)^T}$ is a transposed matrix of the left singular matrix $U^{(1)}$, $U^{(2)}$, $U^{(3)}$, obtained by SVD decomposition in the first to third modes, respectively.

III.V Constructing approximate tensors

Approximate tensors ***A*** is obtained by approximating the approximate core tensor and three left singular matrices $U^{(1)}$, $U^{(2)}$, $U^{(3)}$ obtained by reducing the rank, and modular multiplication. The formula is as follows:

$$A = S \times_1 (U_{c1}^{(1)})^T \times_2 (U_{c2}^{(2)})^T \times_3 (U_{c3}^{(3)})^T \tag{3.7}$$

Where ***S*** is an approximate core tensor, and $U_{c1}^{(1)}$, $U_{c2}^{(2)}$, $U_{c3}^{(3)}$ are the reduced rank matrices of a single left matrix $U^{(1)}$, $U^{(2)}$, $U^{(3)}$ with the dimensional parameters c_1, c_2 and c_3.

III.VI Generate a list of recommendations

The approximate tensor ***A′*** of the reconstruction reflects the relationship between the newly created user, item and label based on the initial tensor. The element of the tensor ***A*** is represented by a quad {u, i, t, p} and the probability (correlation weight) of the item i being tagged by the user u is denoted by p. Based on this, whether the item i is recommended under the label t is determined based on the predicted score. If we want to recommend n items under the t tag to user u, we only need to select the top n items with the highest score.

III.VII Gradient descent

Gradient Descent Optimization It is the most commonly used optimization algorithm for forming neural network models. In the machine learning model, the gradient descent algorithm is mainly used for optimization training. The principle of the gradient descent algorithm is that the gradient of the objective function $J(\theta)$ relative to the parameter θ will be the fastest upward direction of the target function. To minimize optimization problems, only one parameter can be pushed in the other direction of the gradient to reduce the objective function. This step is also known as the learning rate η. The parameter update formula is as follows:

$$\theta \leftarrow \theta - \eta \bullet \nabla_{\theta} J(\theta) \tag{3.8}$$

Among them, $\nabla_{\theta} J(\theta)$ is the gradient of the parameters. According to the different data quantities used to calculate the objective function $J(\theta)$, Batch Gradient Descent, Stochastic Gradient Descent and Mini-batch Gradient Descentare three transformations of the gradient descent algorithm.

Regarding $J(\theta)$'s superimposed gradient descent algorithm calculated on the entire training device, If the class is big data, you may encounter many mismatches and the convergence speed will be slow. The probability gradient descent algorithm represents other extremums, $J(\theta)$ is calculating for the learning sample in the learning series, also known as online learning. The samples and parameters obtained can be updated. As a result, the convergence speed will be faster, but the value of the objective function may fluctuate because the update of the high frequency parameters will change greatly.

Since the small batch gradient descent algorithm is a compromise solution, it selects many small sample calculations in the training sequence for calculating $J(\theta)$, so the training process is more stable and matrix calculation can be used in the batch training method. This is the most commonly used gradient descent algorithm. The ideal gradient attenuation algorithm must satisfy two points of fast convergence and global convergence. For this ideal, many variations of the classical gradient descent algorithm have emerged, which will be described separately below.

When the parameters of the impulse gradient descent method are updated, the accumulating term (pulse) and the hyperparameter γ close to 1 are added in consideration of the value of the current gradient. The falling pulse gradient algorithm speeds up the convergence compared to the original falling gradient algorithm. When the gradient coincides with the direction of the pulse, the pulse term increases, and when it is the opposite, the pulse term decreases, so the algorithm for reducing the pulse gradient can reduce the unit oscillation process.

$$\begin{aligned} m &\leftarrow \gamma \bullet m + \eta \bullet \nabla_\theta J(\theta) \\ \theta &\leftarrow \theta - m \end{aligned} \tag{3.9}$$

Hinton's RMSprop algorithm mainly solves the problem of a sharp decline in learning speed. This idea is similar to the concept of Momentum, which introduces a hyperparameter that is decomposed by accumulating squared gradient elements:

$$\begin{aligned} s &\leftarrow \gamma \cdot s + (1-\gamma) \cdot \nabla_\theta J(\theta) \odot \nabla_\theta J(\theta) \\ \theta &\leftarrow \theta - \frac{\eta}{\sqrt{s+\varepsilon}} \odot \nabla_\theta J(\theta) \end{aligned} \tag{3.10}$$

The RMSprop algorithm can be thought of as a cumulative gradient closer to a time of 0.9, which is the mean of the exponential decay, which reduces the occurrence of an explosion and thus helps solve the problem of avoiding a sharp drop in learning speed.

This paper chooses the rate adaptive RMSprop algorithm.

III.VII Experimental environment for tensor decomposition

By applying the tensor decomposition algorithm to reduce the tensor dimension and the potential semantic analysis between multiple entity types, two links are needed to make the label recommendation more accurate, namely offline and online. The end calculation of HOSVD is done by the offline link, The online link solves the problem by retrieving new relevant weight real-time recommendations from the reconstructed proximity tensor $\boldsymbol{A'}$. For the offline link, in the actual application system (the tag metadata contains a large number of users, items, and tag data), the main feature is that the data is sparse. That is, most tensor elements are zero and empty. Therefore, the following two points are equivalent to the HOSVD calculation.

(1) Calculate the singular values of the matrix expansions $\mathbf{A_1}$、$\mathbf{A_2}$、$\mathbf{A_3}$ in the three modes and the left singular matrix (these are large in size and sparse in data).

(2) In order to get a new tensor core $\boldsymbol{S'}$, The transposed matrix of the three reduced rank matrices $U^{(1)^T}$, $U^{(2)^T}$, $U^{(3)^T}$ obtained by performing the low rank approximation calculation, and the initial tensor matrix are multiplied by calculation.

IV.EXPERIMENTAL DESIGN

In this section, this article compares Tensor Decomposion with Item based CF to get the RMSE experimental results for each algorithm.

Dataset

In order to evaluate the performance of the algorithm, we select the real data set MoviesLens and apply the core subset of the data set for experimental verification.

MoviesLens: The dataset triples {u, I, t} represent users, movies, and public tags, respectively, and the number of users, items, and tags in the data set are 1104, 28599, and 87096, respectively.

Experiment Setup

In the traditional classical evaluation index, the root mean error RMSE (Root Mean Square Error) is the square root of the ratio of the square of the deviation between the observed value and the true value and the number m of observations. It is used to measure the deviation between the observed value and the true value.

Based on this, this article will include all test users to label the film into two groups. The test set is set to test in proportion to the 25% number of tags.

Group 1: Test the label that the user has previously played (defined as $T_1(u)$);

Group 2: Predict the label that the test user will play in the future (defined as $T_2(u)$).

The RMSE is defined as follows:

$$RMSE=\sqrt{\frac{1}{m}\sum_{i=1}^{m}(h(x^{(i)})-y^{(i)})^2} \quad (4.1)$$

Experimental Result

We use the root mean square error as an indicator of the comparison between Tensor Decomposion and Item based CF.

The RMSE of the two algorithms based on the data set MovieLens is as follows:

Tensor Decomposion：RMSE = 0.97091；

Item based CF：RMSE = 3.45430。

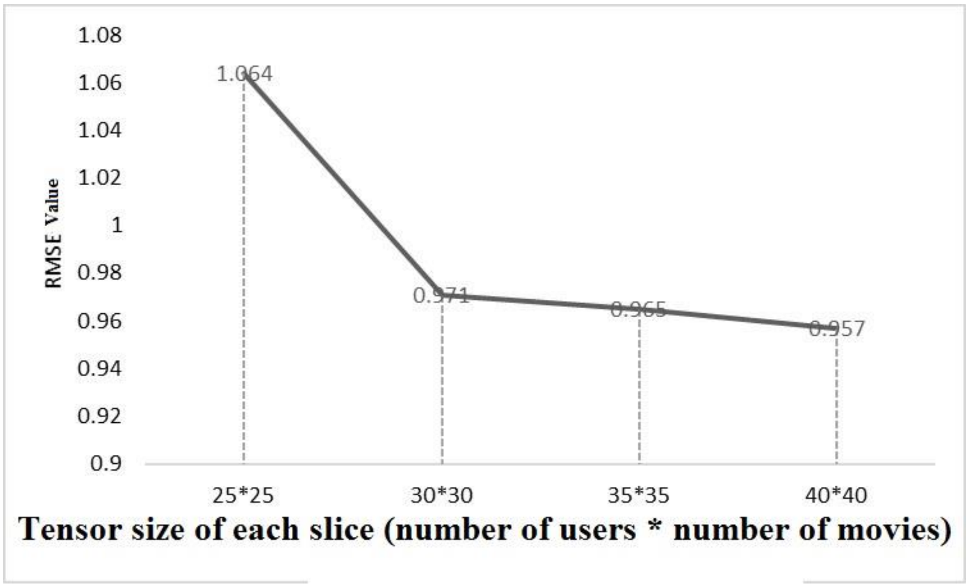
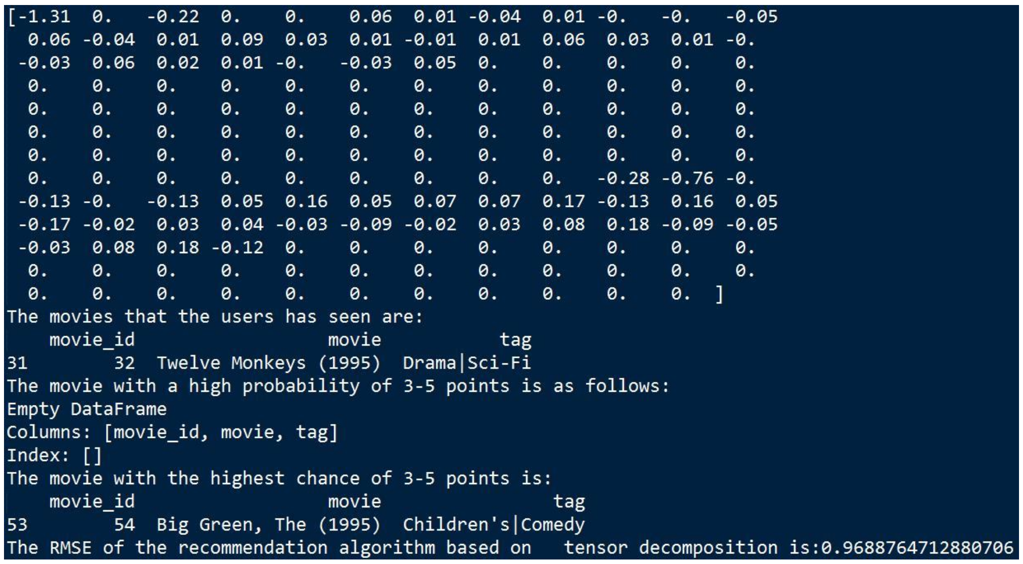


Figure 4.1 Trend of RMSE value under different user movie numbers



```
[-1.31  0.    -0.22  0.    0.    0.06  0.01 -0.04  0.01 -0.   -0.   -0.05
  0.06 -0.04  0.01  0.09  0.03  0.01 -0.01  0.01  0.06  0.03  0.01 -0.
 -0.03  0.06  0.02  0.01 -0.   -0.03  0.05  0.    0.    0.    0.    0.
  0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
  0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
  0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
  0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
  0.    0.    0.    0.    0.    0.    0.    0.    0.   -0.28 -0.76 -0.
 -0.13 -0.   -0.13  0.05  0.16  0.05  0.07  0.07  0.17 -0.13  0.16  0.05
 -0.17 -0.02  0.03  0.04 -0.03 -0.09 -0.02  0.03  0.08  0.18 -0.09 -0.05
 -0.03  0.08  0.18 -0.12  0.    0.    0.    0.    0.    0.    0.    0.
  0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.
  0.    0.    0.    0.    0.    0.    0.    0.    0.    0.    0.  ]
The movies that the users has seen are:
    movie_id                  movie           tag
31        32  Twelve Monkeys (1995)  Drama|Sci-Fi
The movie with a high probability of 3-5 points is as follows:
Empty DataFrame
Columns: [movie_id, movie, tag]
Index: []
The movie with the highest chance of 3-5 points is:
    movie_id                  movie                tag
53        54  Big Green, The (1995)  Children's|Comedy
The RMSE of the recommendation algorithm based on   tensor decomposition is:0.9688764712880706
```

Figure 4.2 Schematic decomposition recommendation algorithm recommendation result schematic diagram

The recommended results of the algorithm are shown in Figure 4.1 and Figure 4.2 . From the data obtained, the root mean square error of the tensor decomposition algorithm is slightly lower than the collaborative filtering algorithm based on the item. The error rate increases with the increase of the number of users and the number of movies. The reduction, and the tensor decomposition increase the dimension of the label based on the collaborative filtering of the item, that is, the synergy relationship between the {user, the item, the label} is increased, and the auxiliary function of the label is more fully utilized.

CONCLUSION

Now that we have entered the era of big data, the data in the era of big data reflects the characteristics of diversity, multimodality, heterogeneity and multiple associations, and the tensor model is a way to visually explain the relationship between things in today's world. And an effective tool for efficient expression and mining. It is very consistent and can present diverse and high-dimensional data. In this paper, we present a three-dimensional tensor model and use this model to represent the three types of entities: user, element, and tag in the social tag recommendation system. The three-dimensional tensor is used in the system for the storage of metadata entities to facilitate the discovery of hidden associations between three different types of entities, and then to reduce the dimensions of the three-dimensional tensor by the HOSVD method. This paper compares the two algorithms of tensor decomposition algorithm and collaborative filtering algorithm. On the experimental platform, we applied the real MovieLens series dataset. The results obtained prove that the tensor decomposition recommendation algorithm has a certain reduction in RMSE.

In this paper, when importing the database construction tensor, it is necessary to make each facet of the tensor is a square matrix of equal size. When the number of users is much larger than the number of movies, it is necessary to further divide the user into several parts according to the number of movies to form a square matrix. Further calculations are needed to make further calculations.

REFERENCE

[1] Adomavicius G， Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. on Knowledge and Data Engineering， 2005，17:734 749.

[2] Lathauwer L D， Moor B D， Vandewalle J. A Multilinear Singular Value Decomposition[J]. Siam J.matrix Anal.appl， 2006， 21(4):1253-1278.

[3] Lieven De Lathauwer， Bart De Moor， Joos Vandewalle. A Multilinear Singular Value Decomposition[J]. Siam J.matrix Anal.appl， 2006， 21(4):1253-1278.