Hybrid matrix factorization recommendation algorithm based on item similarity

Lu Min¹, Xuejian Huang², Gensheng Wang¹,²
¹School of Humanities, Jiangxi University of Finance and Economics, Nanchang 330013, China
²Computer Practice Teaching Center, Jiangxi University of Finance and Economics, Nanchang 330013, China

Corresponding author and e-mail: Gensheng Wang, wgs74@126.com

Abstract. In order to solve the problem of sparse data in matrix factorization recommendation algorithm, a hybrid matrix factorization recommendation algorithm based on item similarity is proposed. First of all, Word2vec is used to mine the content information of items in user's comment data to get the content similarity of items; secondly, the attribute similarity of items is calculated according to the attribute information of items; then, the content similarity and attribute similarity are fused to get the final item similarity; finally, the similarity of item is integrated into the objective function of matrix factorization algorithm. The experimental results show that the hybrid recommendation algorithm has higher accuracy, recall and coverage than the traditional matrix factorization recommendation algorithm.

1. Introduction

To solve the problem of information overload, recommendation algorithm is an important solution. [1]. Matrix factorization recommendation algorithm has become a very popular recommendation algorithm after it stands out in Netflix Grand Prix. Although the algorithm can achieve better recommendation results than other collaborative filtering algorithms, it still faces the problem of matrix sparsity [2]. In view of this problem, many scholars have put forward relevant improvement methods. Such as reference [3] proposed an algorithm to integrate item attribute information into matrix factorization model; reference [4] introduced additional information (such as browsing, collecting, sharing, etc) based on the explicit scoring information; Reference [5] integrates the social status information of users in the matrix factorization recommendation algorithm.

Through analysis, it is found that adding additional information is the main idea to alleviate matrix sparsity. With the rapid development of Internet, data presents diversity and heterogeneity [6]. How to integrate these heterogeneous data is of great significance to improve the performance of recommendation algorithm [7]. Therefore, this paper proposes a hybrid matrix factorization recommendation algorithm which integrates user comment data, rating data and attribute data to alleviate the problem of matrix data.

2. Correlation theory

2.1. Matrix factorization recommendation algorithm
The matrix factorization recommendation algorithm decomposes the rating matrix $R$ into item feature matrix $V$ and user feature matrix $U$. The expression is shown in Eq. (1):

$$ R \approx UV^T $$

The prediction rating of user $i$ for item $j$ is shown in Eq. (2):

$$ \hat{R}_{ij} = U_i V_j^T $$

The objective optimization function is established by linear regression as shown in Eq. (3):

$$ J = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} I_{ij} (U_i V_j^T - R_{ij})^2 + \frac{\lambda}{2} (\|U\|^2 + \|V\|^2) $$

Where $I_{ij}$ is the indicator parameter; $R_{ij}$ is the user's actual score; $U_i V_j^T$ is the predicted score; $\lambda(\|U\|^2 + \|V\|^2)$ is added to prevent regularization by over fitting. $\lambda$ is the regularization parameter.

2.2. Word2vec model

Based on deep learning, word2vec maps words to low dimensional real space containing semantic relations. Words with similar semantics are similar in this space [8]. Word2vec has been widely used in natural language processing, which has changed the traditional one-hot coding problems such as high vector dimension, sparse matrix and lack of semantics. Word2vec includes two training models: Skip-Gram and CBOW. The Skip-Gram predicts context $S_{tw} = (w_{t-k}, \ldots, w_t, \ldots, w_{t+k})$ by inputting word $w_t$, where $k$ is the size of a context window. On the contrary, The CBOW predicts $w_t$ according to context $S_{tw}$. The objective optimization functions are shown in Eq. (4) and (5), respectively.

$$ L_{\text{Skip-Gram}} = \sum_{tw \in C} \sum_{-k \leq j \leq k, j \neq 0} \log p(w_{r_{tw}} | w_t) $$

$$ L_{\text{CBOW}} = \sum_{tw \in C} \log(p(w_t | S_{tw})) $$

Where $C$ is all the words in the text library. Word2vec model is not built to deal with new prediction tasks, but to get the parameter matrix of hidden layer in the trained neural network. These hidden layer parameters are word vectors learned by word2vec.

3. Hybrid matrix factorization recommendation algorithm

The core of this algorithm is to construct a hybrid matrix factorization based on User-Item rating matrix and item similarity matrix. The algorithm flow is shown in Figure 1:

![Figure 1. Algorithm flow.](image)

3.1. Item similarity matrix

The calculation of item similarity is based on the feature description of items. How to get the accurate description of item content becomes a key, which is also the core of content-based filtering recommendation. In many recommendation fields, it is impossible to obtain the description of the content of the object directly, such as movie recommendation. At present, the calculation of movie similarity is based on the attribute information of the movie (such as genres, actor, director, etc.). With
the development of online movie review website, each movie has accumulated a large number of user comment data. Through text analysis technology, we can mine the content information of the movie from these comment data. In this paper, taking movie recommendation as an example, the similarity of movie is calculated by combining movie attribute information and comment data. The algorithm flow is shown in Figure 2.

![Figure 2. Calculation process of item similarity matrix.](image)

### 3.1.1. Movie content similarity
The data set of comment is segmented and pronouns, conjunctions, prepositions and stop words are deleted. Using word2vec model, word vectors containing semantic relations are obtained. The key words of each movie are extracted by TextRank algorithm. Textrank [9] is based on the improvement of PageRank algorithm, which is widely used to extract keywords and keyword groups of text. After getting the keyword group \( W_m = \{w_1, w_2, \cdots, w_k\} \) which represents the content of movie \( m \), combining with the word vector, each keyword \( w_n \) in \( W_m \) is replaced by the corresponding word vector \( V_n = [v_1, v_2, \cdots, v_k] \), and \( k \) is the dimension of the word vector. The content similarity calculation of movie \( m_i \) and \( m_j \) is shown in Eq. (6):

\[
S_c(m_i, m_j) = \sqrt{\frac{\sum_{i=1}^{n} \sum_{j=1}^{m} \cos(V_{w_i}, V_{w_j})}{nm}}
\]

Where \( n \) and \( m \) represent the length of keyword groups of movie \( m_i \) and \( m_j \), respectively. \( \cos(V_{w_i}, V_{w_j}) \) is the cosine similarity of word vectors \( V_{w_i} \) and \( V_{w_j} \).

### 3.1.2. Movie attribute similarity
In reference [11], a detailed similarity calculation method of movie attributes is proposed. The attributes of a movie are defined as \( A = \{\text{Genres, Country, Actors, Director, Release Date, Awards, Label, Score}\} \). The attribute similarity calculation of movie \( m_i \) and \( m_j \) is shown in Eq. (7):

\[
S_a(m_i, m_j) = \frac{\sum_{i=1}^{8} w_i \cdot \text{sim}(a_i)}{\sum_{i=1}^{8} w_i}
\]

Where \( \text{sim}(a_i) \) is the similarity between the two films in the \( i \)-th attribute. \( w_i \) is the weight of the \( i \)-th attribute. Refer to [10] for the calculation of similarity \( \text{sim}(a_i) \) and weight \( w_i \) of each attribute.
3.1.3. Movie similarity. After getting the movie content similarity $S_c(m_i, m_j)$ and movie attribute similarity $S_a(m_i, m_j)$, the two are fused to get the final movie similarity. Fusion calculation is shown in Eq. (8):

$$S(m_i, m_j) = \beta S_c(m_i, m_j) + (1 - \beta) S_a(m_i, m_j)$$

Where $\beta$ is the fusion weight, which controls the proportion of the influence of movie content similarity and movie attribute similarity on the overall movie similarity. Using Eq. (8) to calculate the similarity of all movies, the movie similarity matrix $D$ is obtained.

3.2. User-Item rating matrix

$U = \{u_1, u_2, \ldots, u_m\}$ is user set. $I = \{i_1, i_2, \ldots, i_n\}$ is item set. User-Item rating matrix $R$ is shown in Figure 3.

![Figure 3. User-Resource rating matrix.](image)

$r_{mn}$ is the rating of user $u_m$ on resource $i_n$. When the number of resources is large, users may only rate a small part of the resources, resulting in the whole matrix sparse.

3.3. Hybrid matrix factorization

The movie similarity matrix $D$ and the User-Item rating matrix $R$ are decomposed jointly. The feature matrix of users and items is obtained by integrating multiple heterogeneous information. Make up for the problem of data sparsity in single rating matrix. The objective function of fusion matrix factorization is shown in Eq. (9):

$$J = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}^2 (U^T V - g(R_{ij}))^2 + \frac{\lambda_1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}^2 (V H^T - D_{ij})^2 + \frac{\lambda_2}{2} (\|U\|^2 + \|V\|^2 + \|H\|^2)$$

The function $g(R_{ij}) = (R_{ij} - \text{min}) / (\text{max} - \text{min})$ normalizes the rating. $\lambda_1$ is the balance coefficient. If it is 0, the algorithm degenerates to the traditional matrix factorization algorithm which only uses the rating information.

3.4. Recommended results

The rating of item $j$ by user $i$ is predicted by Eq. (10). Recommend items with prediction rating greater than the threshold to users.

$$\hat{R}_{ij} = (\text{max} - \text{min}) U^T V_j + \text{min}$$

(10)

Eq. (10) is the inverse process of normalization $g(R_{ij})$.

4. Experimental verification and result analysis

4.1. Experimental data
The experimental data set is derived from Douban film reviews, including 7815 users' 214920 comments on 1593 films. The data consistency is 1.73%, which is between movielens10m (4.3%) and movielens10m (1.2%).

4.2. Evaluating indicator
The performance of the algorithm is measured by Precision, Recall and Coverage. The calculation is shown in Eq. (11), (12) and (13) respectively.

\[
\text{Precision} = \frac{TP}{TP + FP} \quad (11)
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \quad (12)
\]

\[
\text{Coverage} = \frac{N_d}{N} \quad (13)
\]

4.3. Experimental result analysis

4.3.1. Experimental comparison of different users and movie feature dimensions \(d\). When decomposing the matrix, we need to set the feature dimension \(d\) of users and movies. Different feature dimension \(d\) may get different experimental results. In this experiment, 9 groups of experiments with dimension \(D = \{10, 15, 20, 25, 30, 35, 40, 45, 50\}\) were set for comparison. The experimental results are shown in Figure 4.

![Figure 4. Comparison of experimental results in different feature dimensions \(d\).](image)

The experimental results show that when the user and movie feature dimensions are 25, the performance of the algorithm is good.

4.3.2. Experimental comparison of different balance coefficient. The balance coefficient \(\lambda_i\) in Eq. (9) controls the proportion of the influence of the similarity degree of item on the results. This experiment set up \(a(\lambda_i = 0)\), \(b(\lambda_i = 0.5)\), \(c(\lambda_i = 1)\), \(d(\lambda_i = 1.5)\), \(e(\lambda_i = 2)\), \(f(\lambda_i = 2.5)\) 6 groups of experiments for comparison. The experimental results are shown in Figure 5.

![Figure 5. Comparison of experimental results in different balance coefficients.](image)
Figure 5. Comparison of experimental results in different balance coefficient.
The experimental results show that when the balance coefficient is 1.5, the performance of the algorithm is good.

4.3.3. Experimental comparison with other algorithms. The in order to further verify the effectiveness of this algorithm, this algorithm (HMF) is compared with user based collaborative filtering (UCF), item based collaborative filtering (ICF) and traditional matrix factorization recommendation algorithm (FunkSVD). The experimental results are shown in Figure 6.

Figure 6. Comparison of experimental results with other algorithms.
The experimental results show that FunkSVD has better performance than UCF and ICF. The performance of HMF is better than FunkSVD. It is proved that the algorithm in this paper is effective.

5. Conclusions
The traditional matrix factorization recommendation algorithm is very dependent on the density of the rating data. In order to solve this problem, this paper proposes an improved matrix factorization recommendation algorithm which integrates user comment data, user item rating data and item attribute information. The problem of sparse rating data can be alleviated by integrating multiple heterogeneous information. Finally, the effectiveness of the algorithm is proved by the comparison of several experiments. Although the traditional matrix factorization recommendation algorithm has been improved in this paper, there are still shortcomings. For example, when faced with massive data, the efficiency of matrix factorization is low. With the rapid development of the Internet, there will be more and more data in the future. How to design efficient recommendation algorithm in massive data is the direction of further research.

Acknowledgement
Fund: NSFC funded project (Project No.: 71461012); Science and technology project of Jiangxi Provincial Department of Education (GJJ181550)

References
[1] Xiong Hui-xiang, Li Yue-yan. Personalized Information Recommendation Based On Weighted Cliques[J]. Information Science, 2018,36(01):67-74. (In Chinese)
[2] Xuejian Huang, Lu Min, Gensheng Wang, Tian Gan, Zhipeng Li. Matrix factorization recommendation algorithms based on knowledge map representation learning[C]. Proceedings of 2019 International Conference on Applied Machine Learning and Data Science (ICAMLDS 2019), 2019:462-467.
[3] Yu Y, Wang C, Wang H, et al. Attributes coupling based matrix factorization for item recommendation[J]. Applied Intelligence, 2017, 46(3):521-533.
[4] Shulong C, Yuxing P. Matrix Factorization for Recommendation with Explicit and Implicit
7

Feedback[J]. Knowledge-Based Systems, 2018,158:109-117.
[5] Yu Yong-hong, Gao Yang, Wang Hao, et al. Integrating User Social Status and Matrix Factorization for Item Recommendation[J]. Journal of Computer Research and Development, 2018,55(1):113-124. (In Chinese)
[6] Zhang H, Ganchev I, Nikolov N S, et al. A trust-enriched approach for item-based collaborative filtering recommendations[C]// IEEE, International Conference on Intelligent Computer Communication and Processing. IEEE, 2016:65-68.
[7] Huang Li-wei, Jiang Bi-tao, Lv Shou-ye, et al. Survey on Deep Learning Based Recommender Systems[J]. Chinese Journal of Computers, 2018,41(7):1619-1647. (In Chinese)
[8] Wang Gen-sheng, Huang Xue-jian. Convolution Neural Network Text Classification Model Based on Word2vec and Improved TF-IDF[J]. Journal of Chinese Computer Systems, 2019,40(05):1120-1126. (In Chinese)
[9] Mihalcea R, Tarau P. TextRank: Bringing order into texts[C]// Proceeding of the Conference on Empirical Methods in Natural Language Processing. EMNLP 2004:404-411.
[10] Zhou Wen-le, Zhu Ming, Chen Tian-hao. A Film Recommendation Method Based on Website Aggregation and Semantic Knowledge[J]. Computer Engineering, 2014, 40(08):277-281. (In Chinese)