IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 31 January 2023, accepted 12 February 2023, date of publication 16 February 2023, date of current version 23 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3246060

RESEARCH ARTICLE

Recommendation System Based on Deep Sentiment Analysis and Matrix Factorization

NING LIU[1,2] AND JIANHUA ZHAO[2]
[1]College of Economics Management, Shangluo University, Shangluo 726000, China
[2]Engineering Research Center of Qinling Health Welfare Big Data, Universities of Shaanxi Province, Shangluo 726000, China

Corresponding author: Ning Liu (liuning20122014@aliyun.com)

This work was supported in part by the Key Research and Development Program of Shaanxi Province under Program 2022GY-073; in part by the Science and Technology Planning Project of Shangluo City under Grant 2022-J-0009; and in part by the Shangluo Universities Key Disciplines Project, Discipline name: mathematics.

ABSTRACT In order to solve the problem of data sparsity and credibility in collaborative filtering, a recommendation system based on sentiment analysis and matrix factorization (SAMF) is proposed in this paper, which uses topic model and deep learning technology to fully mine the implicit information in reviews to improve the rating matrix and assist recommendation. Firstly, user topic distribution and item topic distribution are generated from reviews(consisting user reviews and item reviews) through LDA(Latent Dirichlet Allocation). The user feature matrix and item feature matrix are created based on topic probability. Secondly, user feature matrix and item feature matrix are integrated to create user-item preference matrix. Thirdly, the user-item preference matrix and the original rating matrix are integrated to create the user-item rating matrix. Fourthly, BERT(Bidirectional Encoder Representation from Transformers) is used to quantify the sentiment information contained in the reviews and integrate the sentiment information with the user-item rating matrix, to modify and update the user-item rating matrix. Finally, the updated user-item rating matrix is used to achieve rating prediction and Top-N recommendation. Experiments on Amazon datasets demonstrates that the proposed SAMF has better recommendation performance than other classical algorithms.

INDEX TERMS Reviews, sentiment analysis, matrix factorization, recommendation system.

I. INTRODUCTION

With the explosive increase of Internet information, the problem of information overloaded has become increasingly severe. To deal with overloading information, recommendation system(RS) is produced [1]. RS is a technique that seeks to predict the rating or preference for a user, and it is used to provide suggestions or items to users by exploiting various strategies [2].

Traditional RS mainly includes content-based RS, collaborative filtering RS and hybrid RS [3], [4], [5]. The content-based RS provides recommendations that depend on the user profile and the similarity of the item description. However, this method requires effective feature extraction, which is difficult to finish. At the same time, this method is only limited to text resource recommendation, and it is difficult to mine users' implicit interests [6], [7].

The core idea of collaborative filtering is that users with similar tastes share similar rating distributions toward the same items [8]. Because it won't bother about the content of the items during recommendations process, it can complete complex recommendations. Collaborative filtering makes significant success and plays a vital role in many applications. The main problem in collaborative filtering is data sparsity and the cold start issue. Without complete information, it is hard to recommend efficiently [9], [10]. A sparsity problem arises due to user interactions with a small portion of items in the particular application domain, and the cold start issue is due to the lack of data about new entities, i.e., a new item/new user.

A hybrid RS is an integration of collaborative filtering and content-based method. The hybrid RS can solve the cold start problem of items. While taking into account the

The associate editor coordinating the review of this manuscript and approving it for publication was Yiqi Liu.

preference relationship between users, it can build a better recommendation model. However, traditional RS relies on a large number of interactions between users and items to achieve recommendations, such as user purchase records or user rating records, the results of recommendations depend on the user item interaction matrix. Therefore, the sparsity of data and the cold start problem of new users pose challenges to the accuracy and interpretability of recommendation results [11], [12].

In recent years, with the rapid development of deep learning in AI(artificial intelligence) application field, deep learning has also become an important technology in the field of RS. Compared with the traditional RS, deep learning RS can better mine the latent features of data, obtain deep level feature description of users and items. The deep learning RS mainly uses some deep learning technologies, such as Auto-Encoders [13], Restricted Boltzmann Machine(RBM) [14], Convolutional Neural Networks (CNN) [15] and Recurrent Neural Network (RNN) [16] to build the recommendation model. However, they also face some limitations [17]. (1) They all use word2vec or glove to pretreat word vectors. However, this word vector belongs to a kind of static coding, the same word is the same parameter expression in different contexts. It leads to deviation in the model's understanding of semantics. (2) They only use review data as input, and fail to fully explore the internal relationship between reviews data and rating data. At the same time, the interpretability and scalability of the recommendation results are still the shortcomings of the deep learning RS.

In order to solve the problem of data sparsity and credibility in collaborative filtering, a recommendation system based on sentiment analysis and matrix factorization(SAMF) is proposed in this paper. SAMF uses topic model and deep learning technology to fully mine the implicit information in reviews and improve the rating matrix. Firstly, user topic distribution and item topic distribution are generated from reviews(consisting user reviews and item reviews) through LDA(Latent Dirichlet Allocation). The user feature matrix and item feature matrix are created based on topic probability. Secondly, user feature matrix and item feature matrix are integrated to create user-item preference matrix. Thirdly, the user-item preference matrix and the original rating matrix are integrated to create the user-item rating matrix. Fourthly, BERT(Bidirectional Encoder Representation from Transformers) is used to quantify the sentiment information contained in the reviews and integrate the sentiment information with the user-item rating matrix, to modify and update the user-item rating matrix. Then the updated user-item rating matrix is used to achieve rating prediction and recommendation. Finally, experiments are conducted on Amazon dataset to verify the effectiveness of the proposed algorithm.

The main contributions of this work are listed as follows:

(1) The review texts are used to mine the implicit information of users and items, the user-item preference matrix is constructed to assist recommendation. It can solve the sparsity problem of the rating matrix and reduce the error of rating prediction.

(2) Bert is used to mine the semantic information contained in the review texts. The user sentiment information is deeply integrated into the user-item rating matrix, and the user-item rating matrix is updated to improve the credibility of the rating matrix and the accuracy of recommendations.

The rest of this paper is arranged as follows. In Section II, we discuss the related works. Section III explains the methodology used in this study. In Section IV, we present experiment and simulation results, as well as discussion and influence of research results. Finally, Section V concludes the work and a suggestion is given for future extension of this work.

II. RELATED TECHNOLOGY

A. LDA

LDA is a commonly optimization model for document topic extraction, which belongs to an unsupervised learning algorithm. It is mainly applied to text topic recognition, text classification, text similarity calculation and other aspects in the field of text mining. By decomposing a collection of documents into multiple topics in the form of probability distribution,it performs topic clustering or text classification optimization according to the topic distribution.

Formally, the following terms are defined [18]:

A corpus is a collection of M documents denoted by $D = \{d_i | i \in \{1, 2, \ldots, M\}\}$, where d_i *is* the i_{th} document consisting of N_{i} words. Each word corresponds to a latent topic and the topic set $Z = \{z_i | i \in \{1, 2, \ldots, M\}\}$, where z_i is the topic set corresponding to d_i. It can be seen that the total number of topics in document set D is $l = \sum_{i=1}^{M} count(z_i)$, and the total number of words is $N = \sum_{i=1}^{M} N_i$. The joint distribution of all variables in the LDA model is defined as

$$P(w_i, z_i, \theta_i, \phi | \alpha, \beta) = \prod_{j=1}^{N_i} P(\theta_i | \alpha) \cdot P(z_{i_j} | \theta_i) \cdot P(\phi | \beta) \cdot P(w_{ij} | \varphi_{z_{i_j}}), \quad (1)$$

where α, β follows a prior Dirichlet distribution. w_{i} denotes the word set in the i_{th} document, θ_i denotes the "text-topic" distribution probability of the i_{th} document, and ϕ denotes the "topic-word" distribution matrix. $P(\theta_i|\alpha)$ is the "text-topic" distribution probability of the i_{th} document generating from Dirichlet's prior parameter $\alpha.P(z_{i_j}|\theta_i)$ is the topic probability generated in the j_{th} word of the i_{th} document, depending on the topic distribution θ_i. $P(\phi|\beta)$ is the "topic-word" distribution matrix $\varphi_{z_{i_j}}$ of the topic z_{ij} generating from the Dirichlet distribution parameter $\beta.P(w_{ij}|\varphi_{z_{ij}})$ is the probability corresponding to the word w_{ij} generated from the distribution $\varphi_{z_{i_j}}$.

The main computational problem of LDA topic model is to estimate the latent parameters θ, ϕ and z by Gibbs sampling algorithm and variational Bayesian method.

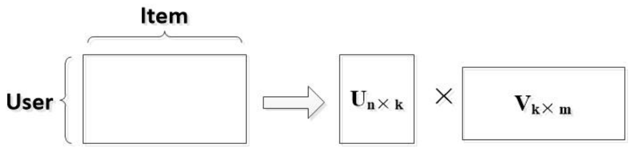


FIGURE 1. Schematic diagram of matrix factorization.

B. MATRIX FACTORIZATION

Among many collaborative filtering recommendation algorithms, matrix factorization has become the most popular one due to its scalability and easy implementation. It is based on the assumption that user preferences are affected by a small number of latent factors, and the user-item rating depends on how each characteristic factor is applied to user preferences [19], [20]. Matrix factorization can map the user-item rating matrix into two or more low dimensional matrices to reduce the dimensions. It can use low dimensional spatial data to study the properties of high-dimensional data, mainly including non negative matrix factorization (NMF) [21], generalized matrix factorization (GMF) and probabilistic matrix factorization (PMF).

Suppose the RS contains n users and m items. Denote R as the rating matrix with n rows and m columns, matrix factorization is to decompose the rating matrix $R_{n\times m}$ into two matrices $U_{n\times k}$ and $V_{k\times m}$ (k denotes the dimension of the latent factors and it is much smaller than m and n). Such that U^VT most closely reconstructs the existing ratings of R and performs well in predicting ratings for non-rated items of R. The schematic diagram is shown in FIGURE 1.

When using matrix factorization to model learners, we usually first construct a matrix with a value of 1/0 based on the interactive data, and then decompose the matrix into two low dimensional matrices. Any application scenario with interaction or rating behavior can consider using matrix decomposition method.

By learning U_i^T($1 \times k$ latent-factor vector for user i) and V_j ($k \times 1$ latent-factor vector for item j), the inner product $\hat{R}_{\mathrm{ij}}$ of U_i and V_j is approximated to the actual rating value R_{ij} assigned by user i for item j. The calculation method is defined as

$$\hat{R}_{ij} = U_i^T V_j, \tag{2}$$

where U_i^T denotes the i_{th} row of U^{T}, V_j denotes the j_{th} column of V, and $\hat{R}_{\mathrm{ij}}$ denotes the predicted rating of user i for item j.

In the process of learning, U and V are calculated by minimizing the regularized mean squared error as follows:

$$\min_{U,\,V} L(U, V) = \frac{1}{2M} \sum_{(i,j)\in O} (R_{ij} - U_i^T V_j)^2 + \frac{1}{2}\lambda(\|U_i\|^2 + \|V_i\|^2), \tag{3}$$

where R_{ij} is the rating value assigned by user i for item j, O is the set of user–item pairs whose rating value has already been generated in R, $M = |O|$ is the number of elements in O, and $\lambda > 0$ is the regularization parameter.

C. BERT

BERT is a pre-trained language representation model and it can directly obtain the global information of the text through the modeling of the self attention mechanism. Because it has no forgetting gate mechanism, all the word information can be retained. Therefore, BERT can better express the complete semantic information of the sentence, and also directly find the correlation features between words from the global word features [22].

BERT is composed of multiple Transformer layers. It conducts multi-level linear transformation on the input vectors to obtain different linear values, then inputs them to the attention module to calculate the attention weight. Finally, the output value of the multi-head attention mechanism is combined to make another linear change. Any vector input to the Transformer is processed and output, and Trans $(\cdot)$ represents all operation procedures in multiple Transformer layers, which is defined as

$$V_t = Trans(W_t X_a + b_t), \tag{4}$$

where X_a denotes the input vector, V_t denotes the output vector, w_t denotes the weight, and b_t denotes the offset.

BERT is composed of multiple transformers stacked together. Bert represents the calculation process in Bert, and it is defined as

$$V_b = Bert(W_b X_b + b_b), \tag{5}$$

where V_b denotes the output value of BERT, X_b denotes the input vector, w_t denotes the weight, and b_t denotes the offset.

III. OUR METHOD

A. ALGORITHM FRAMEWORK

The flow chart of the algorithm SAMF proposed in this paper is shown in FIGURE 2. SAMF mainly includes the following modules. (1) User topic distribution and item topic distribution are generated from reviews through LDA to create user feature matrix and item feature matrix based on topic probability. User feature matrix and item feature matrix are integrated to create user-item preference matrix. (2) The user-item preference matrix and the original rating matrix are integrated to create the user-item rating matrix. (3) Bert is used to quantify the sentiment information contained in the reviews, integrate the sentiment information with the user-item rating matrix, and modify and update the user-item rating matrix. (4) The updated user-item rating matrix is used to achieve rating prediction and Top-N recommendation.

B. IMPLEMENTATION PROCESS

1) GENERATION OF USER FEATURE MATRIX AND ITEM FEATURE MATRIX

User reviews and item reviews are preprocessed. The preprocessed review text set is trained by topic model using LDA, and the number of topics is selected according to the requirements for algorithm execution optimization. The topics are represented by the characteristics of the review

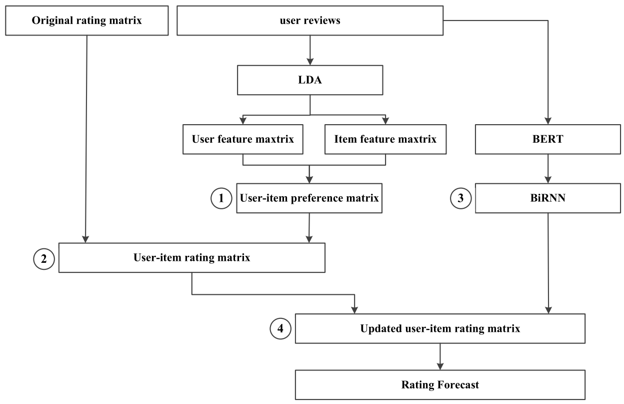


FIGURE 2. **Algorithm Flow Chart.**

text with a certain probability, the user topic feature matrix and the item topic feature matrix are created. The probability distribution of the i_{th} user review on k topics is define as

$$\theta(D^{(u_i)}) = [\theta_1^{(u_i)}, \theta_2^{(u_i)}, \ldots, \theta_k^{(u_i)}], \tag{6}$$

where u_i denotes the i_{th} user, $D^{(u_i)}$ denotes the review of the i_{th} user, k denotes the number of topics, F denotes the number of users, and θ denotes probability distribution.

For the convenience of expression, Equation (6) is transformed into Equation (7). At the same time, the user feature matrix is created, as shown in Equation (8), which denotes the probability distribution of |F| users on k topics.

$$H^{(u_i)} = [H_1^{(u_i)}, H_2^{(u_i)}, \ldots, H_k^{(u_i)}]. \tag{7}$$

$$H = [H^{(u_1)}, H^{(u_2)}, \ldots, H^{(u_{|F|})}]^T. \tag{8}$$

In the same way, the item feature matrix is define as

$$I = [I^{(v_1)}, I^{(v_2)}, \ldots, I^{(v_{|G|})}]^T, \tag{9}$$

where |G| denotes the number of items.

2) GENERATION OF USER-ITEM PREFERENCE MATRIX

The user feature matrix and item feature matrix are integrated on K topics, to generate the user-item preference matrix according to Equation (10). Here, S can effectively use the auxiliary information of user reviews to alleviate the sparsity of the original rating matrix.

$$S = HI^T \in R^{|F|\times|G|}, \tag{10}$$

where H denotes the user feature matrix, I denotes the item feature matrix, and S denotes the user-item preference matrix.

3) INTEGRATION OF USER-ITEM PREFERENCE MATRIX AND RATING MATRIX

The user-item preference matrix and the original rating matrix come from different data sources. One is the probability distribution, and the other is the rating of users for items. Here, the integration of the two is achieved. Since the ratings and reviews have the same user preferences, the user-item preference matrix and the original rating matrix share a common latent factor matrix in the rating space and the review space. In this study, the original rating matrix and the user-item preference matrix are factorized, and the common user latent factor matrix U is used. That is, the user-item preference matrix is decomposed into $U^T P$, and the original rating matrix is decomposed into $U^T V$, where U denotes the user latent factor matrix, P denotes the preference latent factor matrix, and V denotes the item latent factor matrix. The loss function is shown in Equation (11), and the minimum error is calculated.

$$\begin{aligned} L^{'}(U, V, P) &= \frac{1}{2}\sum_{i=1}^{|F|}\sum_{j=1}^{|G|}(R_{ij} - U_i^T V_j)^2 \\ &+ \frac{1}{2}\sum_{i=1}^{|F|}\sum_{l=1}^{|G|}(S_{il} - U_i^T P_l)^2 \\ &+ \frac{1}{2}\|U\|^2 + \frac{1}{2}\|V\|^2 + \frac{1}{2}\|P\|^2, \end{aligned} \tag{11}$$

where R_{ij} denotes the actual rating matrix.

Then, the gradient descent method is used to solve Equation (11), the optimized value of U, V, and P are U^*, V^* and P^*, respectively. The predicted rating matrix is calculated according to Equation (12).

$$\mathrm{R}^* = U^*(V^*)^T, \tag{12}$$

where R^* denotes the user-item rating matrix.

4) UPDATING OF USER-ITEM RATING MATRIX INTEGRATING SENTIMENT ANALYSIS TECHNOLOGY

The user-item rating matrix obtained by Equation (12) reduces the sparsity of the rating table to a certain extent through user reviews, but there are still limitations, such as low credibility. Based on the above problems, the sentiment information contained in user reviews is mined here. Specifically, the sentiment information contained in the reviews is quantified, and integrated with the user-item rating matrix. The user-item rating matrix is modified and updated, so that the rating matrix can more accurately and comprehensively represent the truest sentimental tendencies of users. The specific calculation process is shown below.

First, the user's review text is input into BERT for word vector extraction to obtain the word vector representation of the review text. In each review, each word is converted into a word segmentation vector $E_{\text{Token\_EMB}}$, segment vector $E_{\text{Segment\_EMB}}$ and position vector $E_{\text{Position\_emb}}$. For i_{th} review, three vectors are combined according to Equation (13). The input review sequence is converted into vector encoding according to Equation (14). After vector encoding is processed by multi-level transformer transformation, output word vector from BERT is obtained, as show in Equation (15).

$$E_j = E_{Token\_emb} + E_{segment\_emb} + E_{Position\_emb}. \tag{13}$$

$$E_{input} = \{E_0, E_1, \ldots E_k\}. \tag{14}$$

$$T_{output} = Trans(E_{input}). \tag{15}$$

TABLE 1. The experimental datasets.

Dataset	Amazon food	Amazon clothes
Number of users	1000	500
Number of items	1132	2134
Number of reviews	1919	2463
Average number of reviews	1.92	4.27
Average number of reviews per item	1.70	1.15
Percentage of rated products(%)	0.0017	0.0023

Second, the word vector is transferred to BiRNN layer to extract the sentiment features of user reviews, and the sentiment rating is obtained through the Softmax classifier. Given that the input vector of BiRNN is $\{T_0, T_1, \ldots, T_k\}$, the corresponding output sentiment feature vector is $\{H_0, H_1, \ldots, H_k\}$. The user's sentiment feature is numerically processed to obtain the sentiment rating of the review text, the computing method is defined as

$$p(s_i|H, w_s, b_s) = soft\max(w_s H + b_s), \tag{16}$$

where w_s denotes the weight, b_s denotes the offset in sentiment calculation, and s_i denotes the sentiment rating after calculation.

Third, the sentiment rating is combined with the user-item rating to achieve the correction and update of the user-item rating matrix. The updating method is shown in Equation (17). The sentiment rating s_i is calculated by Equation (16), it is weighted and summed with the rating r_i from user-item rating matrix.

$$r_i^t = (1-\alpha)\mathrm{r_i} + \alpha s_i, \tag{17}$$

where r_i denotes the value from R^* in Equation (12), α denotes the balance factor that measures the weight between the two points, and r_i^t denotes the user-item rating integrating sentiment factors.

Here, the new rating matrix $\mathrm{R_{new}}$ integrating sentiment factors is named updated user-item rating matrix.

5) RATING FORECAST

The updated user-item rating matrix is used to complete Top-N recommendation. Based on the predicted rating, a rating sequence from high to low for each item is established for each user. In the rating sequence, the items that the user has purchased are eliminated, and the first N items in the remaining rating sequence are recommended to the user.

IV. EXPERIMENT

A. EXPERIMENTAL DATA, EXPERIMENTAL SETTING, AND EVALUATION INDICATORS

1) EXPERIMENTAL DATA

In this study, two real datasets are selected to verify the experimental performance. They are the Amazon food dataset and the Amazon Clothing dataset. The detail of dataset is shown in Table 1.

The Amazon food dataset and Amazon Clothing dataset are offered by Kaggle (https://www. kaggle.com). The Amazon food dataset contains user reviews and user ratings about food and drinks sold on Amazon.com. The Amazon Clothing dataset consists of user reviews and user ratings for clothes, shoes and jewelry product on Amazon.com.

In order to better evaluate the performance of the proposed algorithm, user reviews and item reviews are preprocessed by word segmentation and stop words removal. The frequency of the words are also calculated. Reviews with words less than 3 will be deleted from the dataset, and words with frequencies less than 5 will be removed as noise.

2) EXPERIMENTAL SETTING AND EVALUATION INDEX

During the learning process, the dataset is randomly divided into training set and test set at a ratio of 7:3, and the average value is taken as the experiment result after five experiments.

In this study, MAE (Mean Absolute Error) is chosen to test the accuracy of the proposed model and other comparative models in rating and forecasting. MAE is the average of the absolute difference between the actual value and the predicted value in the dataset, and it measures the average of the residuals in the dataset. The calculation method of MAE is shown in Equation (18). The lower values of MAE indicate better performance of models in the accuracy.

$$MAE = \frac{1}{m}\sum_{i=1,j=1}^{m,n}\left|R_{ij} - \hat{R}_{ij}\right|, \tag{18}$$

where m denotes the number of users, n denotes the number of items, R_{ij} denotes the actual rating value of user i for item j, and $\hat{R}_{ij}$ denotes the predicted rating of user i for item j.

In order to verify the effect of the proposed algorithm in recommendation, F1-score is selected to test user tendencies and the recommendation accuracy. The F-score measure is presented for evaluation as a harmonic mean of retrieval and accuracy, and higher values of F1-Score indicate better performance of models in recommendation.

Precision is defined as Equation (19), Recall is defined as Equation (20), and F-score is defined as Equation (21), where $R(U)$ indicates N items recommended to user u and $T(U)$ indicates the related options for user u in test set.

$$\Pr ecision = \frac{\sum_u |R(U) \cap T(U)|}{\sum_u |R(u)|}. \tag{19}$$

$$Recall = \frac{\sum_u |R(U) \cap T(U)|}{\sum_u |T(u)|}. \tag{20}$$

$$F1 - Score = \frac{2 * \Pr ecision * Recall}{\Pr ecision + Recall}. \tag{21}$$

B. COMPARISON MODEL

In order to prove the superiority of the proposed algorithm, the proposed **SAMF** is compared with the other three classical recommendation algorithms.

(1) LFM(Latent Factor Model) [23]. LFM is one of the topic models, which decomposes the co-occurrence

TABLE 2. The MAE on Amazon food dataset.

method	K=10	K=15	K=20	K=25	K=30	K=35	K=40	K=45	K=50
LFM	0.47	0.48	0.45	0.44	0.51	0.5	0.49	0.48	0.47
SVD++	0.45	0.44	0.446	0.47	0.45	0.47	0.45	0.49	0.52
MFFR	0.47	0.42	0.35	0.34	0.35	0.34	0.32	0.37	0.36
SAMF	**0.42**	**0.40**	**0.30**	**0.30**	**0.30**	**0.30**	**0.29**	**0.32**	**0.30**

TABLE 3. The MAE on Amazon clothes dataset.

method	K=10	K=15	K=20	K=25	K=30	K=35	K=40	K=45	K=50
LFM	0.57	0.58	0.54	0.58	0.57	0.56	0.58	0.59	0.58
SVD++	0.55	0.53	0.55	0.54	0.52	0.54	0.52	0.51	0.52
MFFR	0.48	0.47	0.48	0.48	0.47	0.46	0.49	0.46	0.46
SAMF	**0.47**	**0.45**	**0.44**	**0.41**	**0.42**	**0.41**	**0.42**	**0.41**	**0.41**

matrix into user latent factor matrix and item latent factor matrix. It calculates the user's click rate or rating for each item through the inner product of user latent-factor vector and item latent-factor vector, and finally completes the recommendation according to the rating matrix.

(2) SVD(Singular Value Decomposition)++ [24]. On the basis of SVD, several optimizations have been made in SVD++. In addition to transforming the matrix decomposition problem into an optimization problem and adding a regularization term to the loss function, SVD++ also takes into account that the correlation between items evaluated by users may affect the prediction of rating. It also introduces the influence factor among historical items into the prediction function.

(3) MFFR(matrix factorization fusing reviews) [25]. MFFR is a recommendation algorithm based on matrix decomposition. It also integrates the rating matrix and text reviews to assist recommendation.

C. EXPERIMENTAL RESULTS AND ANALYSIS

Under the same experiment environment, **SAMF** proposed in this paper is compared with LFM, SVD++, and MFFR. Each algorithm is tested on Amazon food dataset and Amazon cloths dataset respectively. The experiment is divided into two cases, including rating prediction experiment and Top-N recommendation experiment.

1) RATING PREDICTION EXPERIMENT

Table 2 and Table 3 show the value of MAE in the rating prediction experiment results. FIGURE 3 shows the change of MAE with the number of recommendations k in the Amazon food dataset, and Figure 4 shows the change of MAE with the number of recommendations k in the Amazon clothes dataset.

The results are presented as shown in Table 2. Compared with LFM, SVD++ and MFFR, the MAE value of **SAMF** is also lower than the other three comparison algorithms under different k values on Amazon food dataset. It shows that the proposed algorithm **SAMF** has low error rate in rating prediction and good performance in rating prediction. Compared with LFM, the MAE value of **SAMF** decreases by 15% on average; compared with SVD++, the MAE value of **SAMF** decreases by 14.6% on average; compared with MFFR, the MAE value of **SAMF** decreased by 4.6% on average.

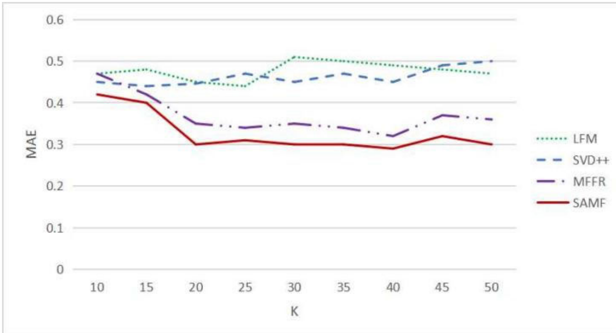


FIGURE 3. The comparison results of different methods on Amazon food dataset.

It can also be seen from Table 3 that the MAE value of **SAMF** is also lower than the other three comparison algorithms under different k values on Amazon Clothes dataset. It shows that **SAMF** has low error rate in rating prediction and good performance in rating prediction. Compared with LFM, the MAE value of **SAMF** decreases by 17% on average; compared with SVD++, the MAE value of **SAMF** decreases by 11% on average; compared with MFFR, the MAE value of **SAMF** decreases by 5% on average.

It can be seen from Figure 3 and Figure 4 that, compared with LFM, SVD++ and MFFR, **SAMF** is also stable. In the whole change process of K, the performance of **SAMF** is better than the other three algorithms. It shows that **SAMF** can make full use of text review information to assist recommendation, fill in the rating matrix, and mine the implicit sentiments of users. It can more truly and accurately understand users' preferences and has a higher performance of rating prediction.

2) TOP-N RECOMMENDED EXPERIMENT

The values of F1-Score are shown in Table 4 and Table 5. FIGURE 5 shows the change of F1-Score with the number of recommendations k in the Amazon food dataset, and Figure 6 shows the change of F1-Score with the number of recommendations k in the Amazon clothes dataset.

It can be calculated from Table 4 that on the Amazon food dataset, the F1-Score value of **SAMF** is higher than that

TABLE 4. F1-Score on Amazon food dataset.

method	K=10	K=15	K=20	K=25	K=30	K=35	K=40	K=45	K=50
LFM	0.75	0.75	0.76	0.76	0.77	0.78	0.78	0.79	0.79
SVD++	0.72	0.72	0.75	0.75	0.75	0.75	0.76	0.76	0.76
MFFR	0.64	0.66	0.67	0.80	0.81	0.82	0.82	0.85	0.85
SAMF	**0.75**	**0.76**	**0.80**	**0.82**	**0.84**	**0.84**	**0.86**	**0.89**	**0.90**

TABLE 5. F1-Score on Amazon clothes dataset.

method	K=10	K=15	K=20	K=25	K=30	K=35	K=40	K=45	K=50
LFM	0.37	0.37	0.43	0.47	0.48	0.51	0.52	0.53	0.54
SVD++	0.36	0.36	0.48	0.49	0.49	0.49	0.49	0.49	0.49
MFFR	0.55	0.57	0.60	0.62	0.63	0.64	0.65	0.65	0.65
SAMF	**0.57**	**0.59**	**0.65**	**0.66**	**0.66**	**0.67**	**0.68**	**0.69**	**0.70**

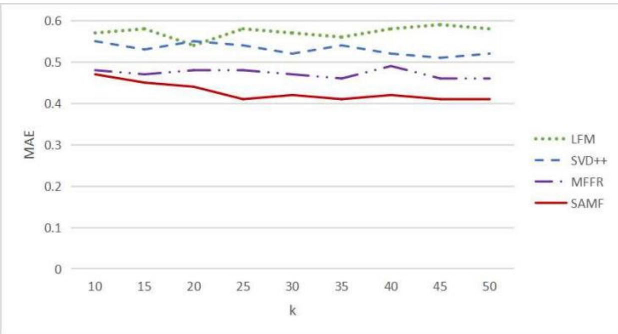


FIGURE 4. The comparison results of different methods on Amazon clothes dataset.

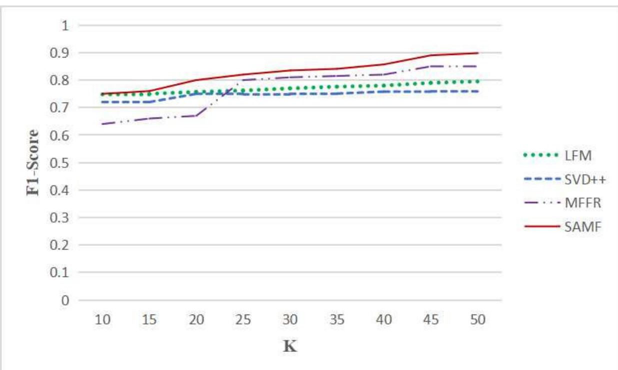


FIGURE 5. The comparison results of different methods on Amazon food dataset.

of the other three comparison algorithms under different k values. Compared with LFM, the F1-Score value of **SAMF** has an average increase of 5.8%; compared with SVD++, the F1-Score value of **SAMF** has an average increase of 8.2%; compared with MFFR, the F1-Score value of **SAMF** has an average increase of 5.9%.

It can be calculated from Table 5 that on Amazon Clothes dataset, **SAMF** is higher than the other three comparison algorithms under different k values. Compared with LFM, the F1-Score value of **SAMF** has an average increase of 18.4%; compared with SVD++, the F1-Score value of **SAMF** has an average increase of 19.2%; compared with MFFR, the F1-Score value of **SAMF** has an average increase of 3.5%.

There are several reasons. (1)Compared with LFM and SVD++, **SAMF** makes full use of review text data to

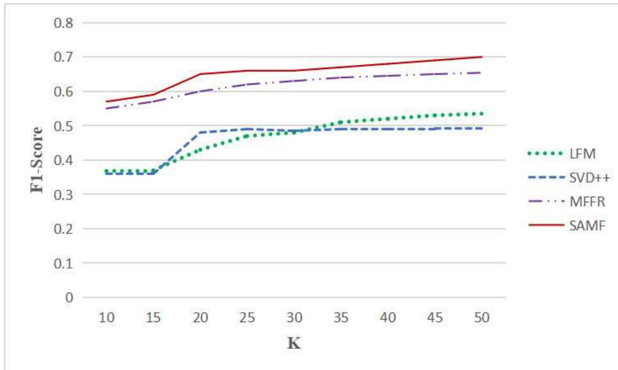


FIGURE 6. The comparison results of different methods on Amazon clothes dataset.

assist recommendation and fill in the rating matrix, and it can effectively solve the problem of data sparsity and improve the recommendation performance. (2)Compared with MFFR, **SAMF** can fully mine the implicit information in reviews, analyze the semantic features of reviews by using rating, reviews and other resources, and fully mine the semantic information of reviews. (3)**SAMF** uses the sentiment information contained in the reviews to update the rating table, so that the rating table can accurately reflect the user's preferences and recommend items that may be of interest to users. So **SAMF** has better recommendation performance.

It can be seen from Figure 5 and Figure 6 that **SAMF** is also stable compared with LFM, SVD++ and MFFR. The performance of **SAMF** is better than the other three algorithms in the whole change process of K. Especially, compared with MFFR, the performance of **SAMF** is always better. It shows that **SAMF** can make full use of the text review information, excavate the implicit sentiments of users, understand users' preferences more truly, and implement recommendations more accurately.

V. CONCLUSION

In order to further improve the performance of RS and solve the problem of data sparsity and credibility in collaborative filtering, we propose a deep learning RS based on sentiment analysis and matrix factorization in this study. The topic model LDA and deep learning technology are used to fully mine the implicit information in text reviews to improve the rating matrix and assist in recommendation. Experimental results show that the proposed algorithm has better performance.

In the future, we intend to extend our work into two main directions. The first direction is to design new methods to use tags and the knowledge graph as auxiliary information to improve recommendation performance. The other direction is to effectively describe the dynamic characteristics of user preferences in tag recommendation scenarios, and propose a tag recommendation model based on dynamic user preferences.

REFERENCES

[1] Z. Batmaz, A. Yurekli, A. Bilge, and C. Kaleli, "A review on deep learning for recommender systems: Challenges and remedies," *Artif. Intell. Rev.*, vol. 52, no. 1, pp. 1–37, Jun. 2018.

[2] E. Çano and M. Morisio, "Hybrid recommender systems: A systematic literature review," *Intell. Data Anal.*, vol. 21, no. 6, pp. 1487–1524, Nov. 2017.

[3] X. Yu, F. Jiang, J. Du, and D. Gong, "A cross-domain collaborative filtering algorithm with expanding user and item features via the latent factor space of auxiliary domains," *Pattern Recognit.*, vol. 94, pp. 96–109, Oct. 2019.

[4] B. Alhijawi, G. Al-Naymat, N. Obeid, and A. Awajan, "Novel predictive model to improve the accuracy of collaborative filtering recommender systems," *Inf. Syst.*, vol. 96, Feb. 2021, Art. no. 101670.

[5] F. Liang, E. Yang, W. Pan, Q. Yang, and Z. Ming, "Survey of recommender systems based on federated learning," *Scientia Sinica Informationis*, vol. 52, no. 5, pp. 713–741, 2022.

[6] Y. Zhang, X. Zhu, C. Xu, and S. Dong, "Hybrid recommendation approach based on deep sentiment analysis of user reviews and multi-view collaborative fusion," *Chin. J. Comput.*, vol. 42, no. 6, pp. 1316–1333, 2019.

[7] Z. Xie, C. Jin, and Y. Liu, "Personalized knowledge recommendation model based on constructivist learning theory," *J. Comput. Res. Develop.*, vol. 55, no. 1, pp. 125–138, 2018.

[8] X. Zheng, M. Wang, C. Chen, Y. Wang, and Z. Cheng, "EXPLORE: EXPLainable item-tag CO-REcommendation," *Inf. Sci.*, vol. 474, pp. 170–186, Feb. 2019.

[9] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web*, Hong Kong, Apr. 2001, pp. 285–295.

[10] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Comput.*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[11] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Sydney, NSW, Australia, Aug. 2015, pp. 1235–1244.

[12] K. Miyahara and M. Pazzani, "Collaborative filtering with the simple Bayesian classifier," in *Proc. 6th Pacific Rim Int. Conf. Artif. Intell.*, Berlin, Germany, 2000, pp. 679–689.

[13] S. Sedhain, A. Menon, S. Sanne, and L. Xie, "AutoRec: Autoencoders meet collaborative filtering," in *Proc. 24th Int. Conf. World Wide Web*, Florence, Italy, 2015, pp. 111–112.

[14] A. Biswal, M. Borah, and Z. Hussain, "Music recommender system using restricted Boltzmann machine with implicit feedback," *Adv. Comput.*, vol. 122, pp. 367–402, 2021.

[15] J. Cheng, P.-S. Wang, G. Li, Q.-H. Hu, and H.-Q. Lu, "Recent advances in efficient computation of deep convolutional neural networks," *Frontiers Inf. Technol. Electron. Eng.*, vol. 19, no. 1, pp. 64–77, Jan. 2018.

[16] L. Huang, B. Jiang, S. Lv, Y. Liu, and D. Li, "Survey on deep learning based recommender systems," *Chin. J. Comput.*, vol. 41, no. 7, pp. 1619–1647, 2018.

[17] S. Seo, J. Huang, H. Yang, and Y. Liu, "Interpretable convolutional neural networks with dual local and global attention for review rating prediction," in *Proc. 11th ACM Conf. Recommender Syst.*, Como, Italy, Aug. 2017, pp. 297–305.

[18] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[19] D. Lee and H. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, Denver, CO, USA, 2000, pp. 556–562.

[20] S. Zhang, W. Wang, J. Ford, and F. Makedon, "Learning from incomplete ratings using non-negative matrix factorization," in *Proc. SIAM Int. Conf. Data Mining*, Bethesda, MD, USA, Apr. 2006, pp. 549–553.

[21] D. Lee and H. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 10, pp. 789–791, 1999.

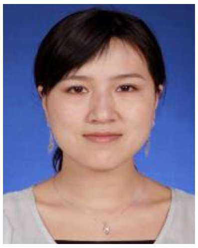
[22] N. Liu and J. Zhao, "A BERT-based aspect-level sentiment analysis algorithm for cross-domain text," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–11, Jun. 2022.

[23] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, vol. 2008, pp. 426–434.

[24] R. Kumar, B. K. Verma, and S. Sunder Rastogi, "Social popularity based SVD++ recommender system," *Int. J. Comput. Appl.*, vol. 87, no. 14, pp. 33–37, Feb. 2014.

[25] H. Wang, Z. Hong, and M. Hong, "Research on product recommendation based on matrix factorization models fusing user reviews," *Appl. Soft Comput.*, vol. 123, Jul. 2022, Art. no. 108971.

[26] X. Cai, L. Xie, R. Tian, and Z. Cui, "Explicable recommendation based on knowledge graph," *Expert Syst. Appl.*, vol. 200, Aug. 2022, Art. no. 117035.



NING LIU received the master's degree in computer software and theory from the School of Information Science and Technology, Northwestern University, China, in 2017. She is currently an Assistant Professor with the School of Economics and Management, Shangluo University, China. Her current research interests include natural language processing, recommendation system, sentiment analysis, semi-supervised learning, and deep learning.

JIANHUA ZHAO received the Ph.D. degree in computer science and technology from the School of Computer, Northwestern Polytechnical University, China, in 2014. He is currently a Professor with the Engineering Research Center of Qinling Health Welfare Big Data, Universities of Shaanxi, China. His current research interests include natural language processing, recommendation systems, sentiment analysis, deep learning, transfer learning, and semi-supervised learning.