Mathematical Biosciences and Engineering

MBE, 19(10): 10673–10686.
DOI: 10.3934/mbe.2022499
Received: 13 May 2022
Revised: 14 July 2022
Accepted: 22 July 2022
Published: 27 July 2022

http://www.aimspress.com/journal/MBE

Research article

Hybrid recommendation algorithm based on real-valued RBM and CNN

Jue Wu[1], Lei Yang[1], Fujun Yang[2,*], Peihong Zhang[2] and Keqiang Bai[1]

[1] School of Computer Science and Technology, Southwest University of Science and Technology, Mianyang, China

[2] Computational Aerodynamics Institute at China Aerodynamics Research and Development Center, Mianyang, China

* **Correspondence:** Email: fintan_yang@yeah.net, Tel: +8613989287057.

Abstract: With the unprecedented development of big data, it is becoming hard to get the valuable information hence, the recommendation system is becoming more and more popular. When the limited Boltzmann machine is used for collaborative filtering, only the scoring matrix is considered, and the influence of the item content, the user characteristics and the user evaluation content on the predicted score is not considered. To solve this problem, the modified hybrid recommendation algorithm based on Gaussian restricted Boltzmann machine is proposed in the paper. The user text information and the item text information are input to the embedding layer to change the text information into numerical vector. The convolutional neural network is used to get the latent feature vector of the text information. The latent vector is connected to rating vector to get the item and the user vector. The user vector and the item vector are fused together to get the user-item matrix which is input to the visual layer of Gaussian restricted Boltzmann Machine to predict the ratings. Some simulation experiments have been performed on the algorithm, and the results of the experiments proved that the algorithm is feasible.

Keywords: convolutional neural network; Gaussian restricted Boltzmann machine; recommendation algorithm

1. Introduction

The recommendation system is very important for alleviating the problem of information overload arising out of rapid development of big data. To provide a better recommendation system, predicting the rating score of the users for the goods is a vital problem to solve. Recommendation system is a main method of data selection. Recommendation system is mainly based on the characteristics of users' daily behavior. Recommendation system can accurately recommend interesting items to users by analyzing the behaviors of them. Recently, recommendation systems have developed rapidly [1–6]. Many researchers have proposed many recommendation algorithms [7–10]. The hybrid recommendation algorithm is an important research point. The neural networks have been widely used in the recommendation system. Compared with the traditional algorithms, the neural networks model can better match the data. The restricted Boltzmann machine (RBM) [11] has been used in the recommendation system and exhibits good performance [12–17]. Majority of the recommendation algorithms based on RBM use only the rating data in which the information of the user and item are not taken into account.

The convolutional neural network (CNN) is a neural network which uses the convolution structure to extract the valuable information. The convolution structure is used to get the local feature. In order to reduce the number of the parameters, the weights share and the pooling operator are used in the model which however, alleviates the over-fitting problem. The CNN was originally developed to solve computer vision problems. CNN architecture is very important for CNNs' performance [18–20]. Off late, CNN has been successfully applied in various fields of natural language processing [21–24]. Recently, the CNN was also used in the recommendation system [25–28] and proved that it can extract the context information of the texts.

The modified hybrid recommendation algorithm based on the Gaussian RBM (GRBM) and CNN is proposed in this paper. The user text information and the item text information are taken into account in the algorithm. The word2vec is used to change the text data into numerical vector. The CNN is used to get the latent feature of the text data. The GRBM is used to predict the ratings.

The structure of the paper is as follows. An extensive literature review is presented in Section 2. The conceptual model is elaborated in Section 3. Section 4 describes the simulation experiments and the analysis result. The findings are concluded in Section 5.

2. Related work

The restricted Boltzmann machine (RBM) was first proposed by Hinton [11] and was widely used for recommendation system because of its high accuracy. In order to further improve its performance many optimization methods were proposed. A hybrid recommendation algorithm based on RBM and term frequency-inverse document frequency (TF-IDF) was proposed by Wang and Li [12]. The RBM was used to fill in the rating matrix to alleviate the problem of data sparsity in this paper. A novel recommendation method based on the trust-distrust relationship (TDA-RBM) was suggested by Hu et al. [13], etc. The trust relationship was applied in the algorithm, and the trust-distrust supervision mechanism was constructed in the model. A hybrid recommendation model based on RBM was advocated by Wang et al. [14]. The RBM was used to generate the candidate sets where the LFM was used to sort the candidate results. The IRC-RBM algorithm was proposed by He et al. [15] in which the rating matrix based on the cluster method and the IRC-RBM model were fused together to get the recommendation. A hybrid recommendation algorithm based on restricted Boltzmann machine and weighted slope one was proposed (IR-RBM-WSO) by Shen et al. [29] where a hybrid item similarity calculation method was used in this algorithm. A user-based BM model with time information was recommended by Du and Zhou [30] where the time information bias was added into the existing RBM model in the algorithm. Kuo and Chen [31] proposed a hybrid algorithm and the cluster-based RBM

(CRBM) with differential evolution method to optimize the parameters. A new item-based restricted Boltzmann machine method for collaborative filtering was propounded by Du et al. [32] where the multilayer RBM structure can extract feature very well. A Recommendation Algorithm based on RBM and the item type was put up by He et al. [33] where the type similarity of the item was introduced to modify the original neighbors in the algorithm where the type similarity of the item was used in the algorithm to get neighbors which were closer to the target, and this model is better than algorithms based on RBM. Conditional restricted Boltzmann machine (CRBM) for item recommendation was proposed by Chen et al. [34] where the users' feedback data were converted into three different matrixes and the CRBM was used to handle the matrixes. Real-Valued conditional restricted Boltzmann machine (R_CRBM) was put forward by He and Ma [35] wherein the nearest trusted relationships were used in the algorithm to improve the performance of CRBM.

When the limited Boltzmann machine RBM was used for collaborative filtering, only the scoring data matrix was considered, and the influences of item content, user characteristics and user evaluation content on the predicted score were not considered leading to low accuracy. We proposed a novel collaborative filtering algorithm based on RBM which combine the item content and the users feature. The word2vec is used to get the items and the users vector subsequently, the vectors were input into the CNN to get the latent vectors of items and users. Combining the latent vector and the score matrix, we get the combined matrix, which is then input into the Gaussian RBM (GRBM) to get the score predict model. The model is denoted as TCNN-GRBM in this paper.

3. Algorithm model

3.1. Model structure

The text information of items and users are put into the respective embedded layer. The embedded layer changes the text information into the data matrix. The user data matrix and the item data matrix are put into the CNN to get the users latent vector and the items latent vector, respectively. Then the item latent feature vector is connected with the item score vector given by all the users. Later, the user latent feature vector is connected with the score vector given by the user. The connected vector is fused together in the fusion layer to get the new user-item matrix. which is input into the GRBM. The model structure is shown in Figure 1.

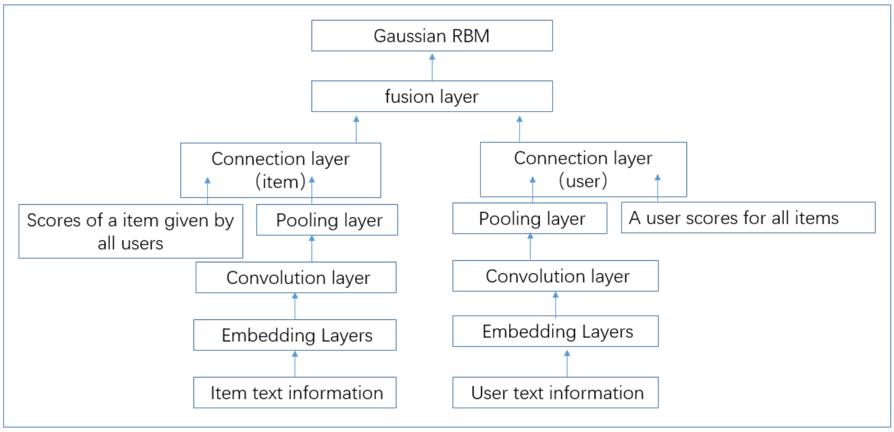

Figure 1. Algorithm model structure.

3.2. Latent feature learning

Convolutional neural network (CNN) is generally used in computer vision for image classification, detection, and nature language processing. Off late, CNN has been introduced into the recommendation systems to achieve good results. The CNN is used to train the text information of users and items to obtain the latent features of the items and the users in this paper.

3.2.1. Data normalization

The scores are integers 1–5 or 1–10, and so the data needed can be normalized. Assumed that 0 means that the users did not rate the item; we normalize the score data to a floating point value. The normalization formula is as follows:

$$r_{norm} = \frac{\mathrm{r} - r_{\min}}{r_{\max} - r_{\min}} \tag{1}$$

Where, r_{norm} is the normalized data, r is the source data, r_{max} and r_{min} are the maximum and the minimum value of the score matrix respectively.

3.2.2. Convolutional neural network

The word2vec is used in the paper to get the vectors of the item text information and the user text information. Multi-layer convolution in CNN model can obtain the interrelationship among the words in the documents, and so it can learn the relationships among the contexts. The CNN is used to get the latent item feature and the latent user feature in this paper. The specific processes followed are as follows.

(1) Embedding layer

The word embedding layer transforms the documents into the numerical matrix to extract its semantic information and the vectors are input to the next layer. The word2vec model is used in this paper to extract the word vector. The user text data and the item text data are input to the word2vec. The user text data include the score, comment, click, and social data. The item text data include the name, size, color, quality, image and other data that describe the item.

The text is taken as a sequence of words, and the word vectors are connected together to form a text matrix. The matrix can be initialized arbitrarily, or it can be initialized with the trained vector and embedded into the model.

(2) Convolution layer

The convolution operation can get the latent feature of the context. The convolution operation on a text matrix can enhance some features of the text, reduce the noise and get the latent feature of the text. The different sizes convolution kernels are used in the algorithm to get the relationship among the word vectors. The convolution operation is described as:

$$\mathrm{A}_i = f_1(W_1 \otimes X_1 + b_1) \tag{2}$$

Where, $\otimes$ represents the convolution operator, W_1 denotes the weight, X_1 represents the vector matrix, and b_1 denotes the bias, f_1 denotes the activate function.

After the convolution operation, the output of the convolution layer is given by:

$$\mathrm{A} = \{a_1, a_2, ..., a_i\} \tag{3}$$

Where, *A* is the latent feature of the text, used as the input of pooling layer and is composed of different convolution kernels.

(3) Pooling layer

Pooling layer can reduce the number of features and parameters. This further reduces the amount of calculation, controls over-fitting and extracts representative features. Maximum pooling is used in this paper. Its main function is to maximize the features observed from different angles. That is, to capture the most important information of the feature and the formula is expressed as:

$$y_1 = \max\{0, a\} \tag{4}$$

(4) Output layer

After convolution and pooling operations, the data is input to a full connection layer. The latent feature of the text can be obtained in this layer. The process can be expressed by the following formula:

$$y_2 = f_2(w_2 y_1 + b_2) \tag{5}$$

Where, w_2 denotes the weight, b_2 denotes the bias and f_2 denotes the activate function. In this paper the ReLU is used as the activate function.

(5) Feature connection layer

The text latent feature is connected with the score data in this layer. The network is added in the structure in order to make the connected item feature and the connected user feature have the same dimension. The process can be expressed in following formulae (6)–(8).

$$\mathrm{y}_{3cI} = \begin{bmatrix} s_I \\ y_{2I} \end{bmatrix}, \ \mathrm{y}_{3cu} = \begin{bmatrix} s_u \\ y_{2u} \end{bmatrix} \tag{6}$$

$$\mathrm{y}_{3\mathrm{u}} = f_3(w_3 y_{3cu} + b_3) \tag{7}$$

$$\mathrm{y}_{3I} = f_3(w_3 y_{3cI} + b_3) \tag{8}$$

Where, y_{2cI} denotes the item score vector which is composed by the scores given by all the users, y_{3cu} denotes connection vector of the user, s_I denotes an item score vector given by all the users, s_u denotes the score vector that composed by all the items score given by a user, y_{2I} denotes the item latent features vector, y_{2u} denotes the user latent feature vector, y_{3u} denotes the connected user vector, y_{3I} denotes the connected item vector, w_3 denotes the weight, and b_3 denotes the bias.

(6) Feature fusion layer

The user feature matrix and the item feature matrix are fused in this layer. The final score matrix can be obtained by the fusion operation by using the formula:

$$\mathrm{y}_4 = f_4((y_{3u} \ \mathrm{e} \ y_{3I}) w_4 + b_4) \tag{9}$$

Where, e denotes the inner product operation, w_4 denotes the weight, b_4 denotes the bias, and f_4 denotes the activate function. The ReLU is used in this layer.

3.3. Gaussian RBM

RBM is generally suitable for processing the binary data. Many scholars have proposed a series of variant algorithms of RBM to deal with the real valued data. The Gaussian Binary RBM (GRBM) is used in the paper to deal with the score data. GRBM is the earliest RBM variant model for real value data. The traditional visible layer units are regarded as Gaussian variables with diagonal covariance. The energy function can be defined as:

$$E(v,h)=-\sum_{\mathrm{i}=1}^{m}\sum_{j=1}^{\mathrm{n}}\frac{v_i}{\sigma_i}\omega_{ik}h_j-\sum_{j=1}^{n}b_jh_j-\sum_{i=1}^{m}\frac{(v_i-c_i)^2}{2\sigma_i^2} \tag{10}$$

Where, v denotes the feature vector of the visible layer, h denotes the feature vector of the hidden layer, *E(v,h)* denotes the energy function, ω denotes the connection weight between visible layer and hidden layer, *b* and *c* denote the bias of visible layer and hidden layer respectively. σ_i denotes variance of the unit *i* in the visible layer, m denotes unit number in visual layer, and *n* denotes unit number in hidden layer. When either visible layer or hidden layer in GRBM is fixed, the conditional probability distribution of the other layer can be calculated as follows:

$$P(v_i|h)=\frac{1}{\sigma_i\sqrt{2\pi}}\exp(-\frac{1}{2\sigma_i^2}(v_i-c_i-\sigma_i\sum_j\omega_{ij}h_j)) \tag{11}$$

$$P(h_j=1|v)=s(\sum_i\frac{v_i}{\sigma_i}\omega_{ij}+b_j) \tag{12}$$

Where, $s(x)=\frac{1}{1+\exp(-x)}$ is the activated function.

$P(v_i|h)$ can be described as $N(\sigma_i\sum_j\omega_{ij}h_j+c_i,\sigma_i^2)$. That is, it is a Gaussian distribution, with mean $\sigma_i\sum_j\omega_{ij}h_j+c_i$ and the variance σ^2.

3.4. Model training

We jointly train the two sub-modules. The contrastive divergence (CD) algorithm is used in the paper to train the model. The gradient of the parameters (ω,b,c,σ) can be calculated according to the formulae (12)–(15).

$$\Delta\omega_{ij}=\frac{\varepsilon(\sum_{i=1}^{D}\frac{v_i}{\sigma_i}h_{jdata}-\sum_{i=1}^{D}\frac{v_i}{\sigma_i}h_{j\,\mathrm{mod}el})}{N} \tag{13}$$

$$\Delta b_j=\frac{\varepsilon(\sum_{i=1}^{D}h_{jdata}-\sum_{i=1}^{D}h_{j\,\mathrm{mod}el})}{N} \tag{14}$$

$$\Delta c_i=\frac{\varepsilon(\sum_{i=1}^{D}\frac{v_{idata}}{\sigma_i}-\sum_{i=1}^{D}\frac{v_{i\,\mathrm{mod}el}}{\sigma_i})}{N} \tag{15}$$

$$\Delta\sigma_i=\frac{\varepsilon}{D}(\sum_{i=1}^{D}(\frac{(v_{i\mathrm{data}}-c_i)^2}{\sigma_i^3}-\sum_{j=1}^{n}\frac{v_{i\,\mathrm{mod}el}\omega_{\mathrm{ij}}h_{jdatal}}{\sigma_i^2})-\sum_{i=1}^{D}(\frac{(v_{i\mathrm{data}}-c_i)^2}{\sigma_i^3}-\sum_{j=1}^{n}\frac{v_{i\,\mathrm{mod}el}\omega_{\mathrm{ij}}h_{j\,\mathrm{mod}el}}{\sigma_i^2})) \tag{16}$$

Where, D denotes Sample number of the training set, ε denotes the learning rate, v_{idata} denotes the data variance, v_{imodel} denotes the variance predicted by the model, h_{idata} denotes the hidden layer data, h_{imodel} denotes the hidden layer data predicted by the model, v_{idata} denotes visual layer data, and h_{imodel} denotes the visual layer data predicted by the model.

The model detailed training process is as follows.

Table1. Training process of the model.

```
Input: Train dataset
Output: the parameters $\omega, b, c, \sigma$
    Step 1: Use the word2vec to get the text matrix. And normalize the score matrix using formula (1).
    Step 2: Initialize all the parameters $\omega, b, c, \sigma$.
For t = 1 to epoch
    Step 3: Do the convolution operation according to formula (2)
    Step 4: Do the pooling operation according to formula (4)
    Step 5: Calculate $y_2$ according to formula (5)
    Step 6: Calculate $y_{3u}$, $y_{3I}$ according to formulae (6) and (7) and get the items and users latent feature.
    Step 7: Calculate $y_4$ according to formula (8)
    Step 8: Calculate the probability according to formula (11)
    Step 9: Calculate the probability according to formula (12)
    Step10: Calculate according to formulae (13)–(16) to get the gradient $V\omega_{ij}, Vb_i, Vc_i, V\sigma_i$
    $\omega_{ij} = \omega_{ij} + V\omega_{ij}$
    $b_i = b_i + Vb_i$
    $c_i = c_i + Vc_i$
    $\sigma_i = \sigma_i + V\sigma_i$
End for
```

4. Experiment

4.1. Datasets

The data sets used in this paper are Movielnes 1M and Moviellens 10M. MovieLens collects a large number of users' movie evaluation data, which is suitable for the experiment and comparative analysis of recommendation algorithms. The dataset is mainly composed of user, rating, movie and other information, which at present is one of the commonly used dataset of the recommendation algorithms. The ratings.dat file contains user rating information, the file movies.dat contains the movie information, and the file users.dat contains the users' information. The user information includes userid, gender, age, profession and so on. The movie information includes movie id, name, movie type, some introduction. The rating data concludes the ratings given by users, it is the integers between 1 and 5. The rating 0 means that users did not rate the movie. The detailed information of the datasets is summarized in Table 2.

Table 2. The datasets used in the experiment.

Dataset	Number of users	Number of items	Number of scores
Movielens 1M	6039	3544	993482
Movielens 10M	69878	10073	9945875

4.2. Evaluation index

There are many evaluation indexes for the recommendation algorithm. The mean absolute error (MAE) and root mean square error (RMSE) are used in this paper. The MAE is calculated between the predicted value and the user's actual rating value. The MAE can well reflect the actual situation of the predicted value. The smaller the MAE value the higher is the recommended accuracy. The calculation is done by using following formula:

$$MAE = \frac{\sum_{i=1}^{D} |p_i - r_i|}{D} \tag{17}$$

Where, p_i denotes the predicted score, r_i denotes the actual score, and D denotes the number of samples.

In general, the MAE value is relatively small, but in the case of high error dispersion or outliers, RMSE is magnified because RMSE squares the deviation. RMSE is sensitive to large or small errors and can better reflect the accuracy of the predicted values, which is helpful to discover the existence of the abnormal value. The smaller the RMSE value, the better the recommendation effect. The RMSE is calculated by using formula:

$$RMSE = \frac{1}{D}\sqrt{i = \sum_{i=1}^{D} (p_i - r_i)^2} \tag{18}$$

4.3. Contrast experiment

In order to verify the performance of the algorithm, it is compared with hybrid recommendation algorithms based on RBM model. The R-CRBM [35], Real UI-RBM [16], TDA-RBM [13] and S-RBM [17] are selected to perform the contrast experiments in this paper.

(1) R-CRBM: The visible layer is a real-value conditionally constrained Boltzmann machine model. The model uses conditions on the user's rate items.

(2) Real UI-RBM: This model is RBM model based on the real value. The visible layer of this model can directly be represented by real value data, and the missing score data does not participate in the training.

(3) TDA-RBM: The trust and distrust relationship were introduced into the recommendation system.

(4) RBM: This model is based on the real-value RBM model which improves the CD learning algorithm, and its input training data is the original score matrix.

(5) The model proposed in this paper.

4.4. Experiment result

All the experiments are carried out under the Win10 system, using Python coding. The computer is powered by an Intel Core I5 processor with 3.15 GHz frequency and 8 GB memory. The RBM part of the algorithm is constructed by TensorFlow. Set the number of neurons in the hidden layer as 20, the learning rate as 0.8 and calculate the MAE and RMSE. The results of the experiment are shown in Figures 2 and 3.

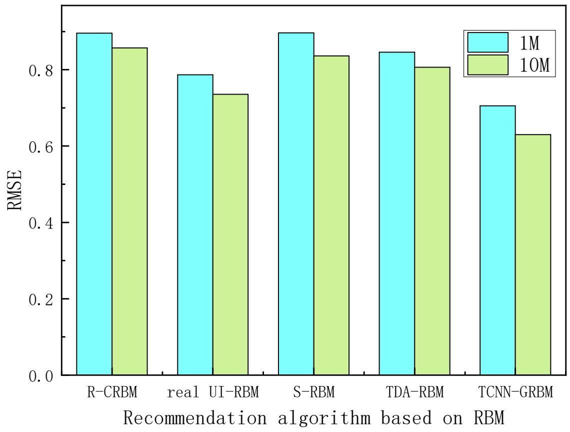

Figure 2. RMSE comparison of different hybrid recommendation algorithms based on RBM.

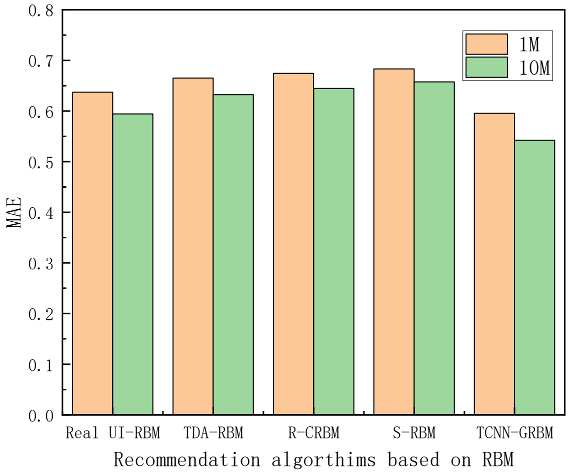

Figure 3. MAE comparison of different hybrid recommendation algorithms based on RBM.

Experimental results show that the RMSE and MAE values of S-RBM and R-CRBM algorithms are the largest, and the recommendation accuracy is the lowest, because these two algorithms only consider the score matrix. The RMSE and MAE values of TDA-RBM and Real UI-RBM algorithms are better than those of S-RBM and R-CRBM, because they consider the trust relationship between the users and the underlying characteristics of the project. The algorithm proposed in this paper shows the best performance on both RMSE and MAE, since the latent characteristics of the user information and the project information are considered in the deep learning algorithm, which provides more accurate recommendation results.

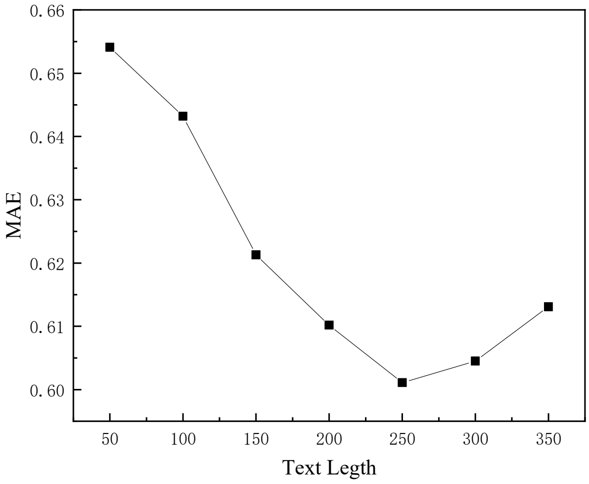

Figure 4. MAE value with different text length.

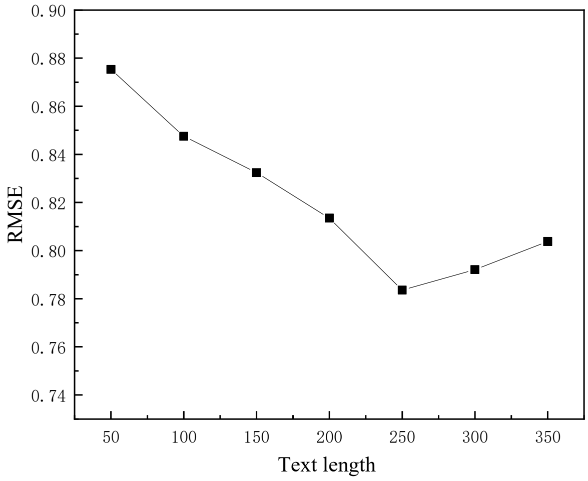

Figure 5. RMSE value with different text length.

In order to explain the influence of the user text information and the item text information on recommendation performance, we analyzed the influence of the maximum document length on RMSE value and MAE value. The experimental results are shown in Figures 4 and 5. It is evident from Figures 3 and 4 that when the maximum length of the item document is smaller, the value of RMSE and MAE is higher whereas, the accuracy is lower. When the maximum length of the project document gradually increases, the value of RMSE and MAE gradually reduces Whereas, when the maximum length of the project document Max length reaches 300, the value of RMSE and MAE instead begin to increase. It is speculated that with the increase in the text length, the noise in the text increases leading to decrease in model accuracy.

5. Conclusions

The modified hybrid recommendation algorithm based on GRBM is proposed in this paper. The text information of the items and the users are used in this algorithm. The word2vec is used to change the text data into numerical data. The CNN is used to get the latent feature of the text information. The latent feature vectors are fused with the score vector to get the user-item matrix. The GRBM is used to predict the user score. The similarity of the items and the users are considered in the algorithm which gives better recommendation performance. The result of the experiments indicates that by taking the user information and the item information into account in the scoring process improves the recommendation accuracy. Thus, we conclude that the text information used in the algorithm model helps to improve the recommendation performance. There are some shortcomings in this model. The data sparsity was not taken into account in the model. And this may affect the performance of the model. In the future, we will study the effect of data sparsity in the model and improve performance of the model.

Acknowledgments

This work was supported by the National Numerical Wind Tunnel Project, the Doctoral Fund of the Southwest University of Science and Technology (13ZX7102) and the Sichuan Science and Technology Program (Nos. NO.2021YJ0334).

Conflict of interest

The authors declare that there is no conflict of interest.

References

1. L. Qi, C. H. Hu, X. Y. Zhang, M. R. Khosravi, S. Sharma, S. N. Pang, et al., Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment, *IEEE Trans. Ind. Inf.*, **17** (2021), 4159–4167. https://doi.10.1109/TII.2020.3012157
2. Y. Tian, R. Zheng, Z. Liang, S. Li, F. X. Wu, M. Li, A data-driven clustering recommendation method for single-cell RNA-sequencing data, *Tsinghua Sci. Technol.*, **26** (2021), 772–789. https://doi.10.26599/TST.2020.9010028
3. L. Shen, Q. Liu, G. Chen, S. Ji, Text-based price recommendation system for online rental houses, *Big Data Min. Anal.*, **3** (2020), 143–152. https://doi.10.26599/BDMA.2019.9020023
4. L. Qi, W. Lin, X. Zhang, W. Dou, X. Xu, J. Chen, A correlation graph based approach for personalized and compatible Web APIs recommendation in mobile APP development, *IEEE Trans. Knowl. Data Eng.*, **2022** (2022). https://doi.10.1109/TKDE.2022.3168611
5. P. Nitu, J. Coelho, P. Madiraju, Improvising personalized travel recommendation system with recency effects, *Big Data Min. Anal.*, **4** (2021), 139–154. https://doi.10.26599/BDMA.2020.9020026

6. T. Li, C. Li, J. Luo, L. Song, Wireless recommendations for Internet of vehicles: Recent advances, challenges, and opportunities, *Intell. Converged Networks*, **1** (2020), 1–17. https://doi.10.23919/ICN.2020.0005
7. Z. Y. Ji, M. D. Wu, H. Yang, J. E. A. Íñigo, Temporal sensitive heterogeneous graph neural network for news recommendation, *Future Gener. Comput. Syst.*, **125** (2021), 324–333. https://doi.10.1016/j.future.2021.06.007
8. A. H. Nabizadeh, J. P. Leal, H. N. Rafsanjani, R. R. Shah, Learning path personalization and recommendation methods: A survey of the state-of-the-art, *Expert Syst. Appl.*, **159** (2020), 1–20. https://doi.10.1016/j.eswa.2020.113596
9. C. F. Xu, J. Feng, P. P. Zhao, F. Z. Zhuang, D. Wang, Y. Liu, et al., Long-and short-term self-attention network for sequential recommendation, *Neurocomputing*, **423** (2021), 580–589. https://Doi.10.1016/j.neucom.2020.10.066
10. J. Zhang, B. Qin, Y. F. Zhang, J. H. Zhou, H. W. Wang, A knowledge extraction framework for domain-specific application with simplified pre-trained language model and attention-based feature extractor, *Serv. Oriented Comput. Appl.*, **16** (2022), 121–131. https://doi.10.1007/s11761-022-00337-5
11. G. Hinton, A practical guide to training restricted Boltzmann machines, *Momentum*, **9** (2010), 926–947.
12. C. Wang, Q. M. Li, Research on hybrid recommendation algorithm based on restricted Boltzmann machine and term frequency-inverse document frequency, *J. Nanjing Univ. Sci. Technol.*, **45** (2021), 551–557. https://doi.10.14177/j.cnki.32-1397n.2021.45.05.005
13. C. H. Hu, X. Q. Tong, W. Liang, The real-valued restricted Boltzmann machine recommendation algorithm based on trust-distrust relationship, *Syst. Eng. Theory Pract.*, **39** (2019), 1817–1830. https://doi.10.12011/1000-6788-2018-2504-14
14. W. B. Wang, L. C. Zhang, Q. Xu, A recommendation algorithm based on restricted Boltzmann machine, *J. Harbin Univ. Sci. Technol.*, **25** (2020), 62–67. https://doi.10.15938/j.jhust.2020.05.009
15. D. P. He, W. Y. Zhang, H. Huang, A hybrid recommendation algorithm based on multi-source information clustering and IRC-RBM, *Comput. Eng. Sci.*, **42** (2020), 1089–1095. https://doi.10.3969/j.issn.1007-130X.2020.06.107
16. L. L. Pei, *An improved real UI-RBM collaborative filtering recommendation algorithm*, Ph.D thesis, Lan Zhou University, 2016.
17. K. Georiev, P. Nakov, A non-IID framework for collaborative filtering with restricted Boltzmann machine, in *Proceedings of the 30th International Conference on Machine Learning*, (2013), 1148–1156.
18. Y. Xue, Y. K. Wang, J. Y. Liang, A. Slowik, A self-adaptive mutation neural architecture search algorithm based on blocks, *IEEE Comput. Intell. Mag.*, **16** (2021), 67–78. https://doi.10.1109/MCI.2021.3084435
19. Y. Xue, P. Jiang, F. Neri, A multiobjective evolutionary approach based on graph-in-graph for neural architecture search of convolutional neural networks, *Int. J. Neural Syst.*, **31** (2021), 1–17. https://doi.10.1142/S0129065721500350
20. D. O'Neill, B. Xue, M. Zhang, Evolutionary neural architecture search for high-dimensional skip-connection structures on densenet style networks, *IEEE Trans. Evol. Comput.*, **25** (2021), 1118–1132. https://doi.10.1109/TEVC.2021.3083315

21. B. Jang, M. Kim, G. Harerimana, S. Kang, J. W. Kim, Bi-LSTM model to increase accuracy in text classification: Combining Word2vec CNN and attention mechanism, *Appl. Sci.*, **10** (2020), 1–14. https://doi.10.3390/app10175841
22. S. H. Yu, D. L. Liu, W. F. Zhu, Y. Zhang, S. M. Zhao, Attention-based LSTM, GRU and CNN for short text classification, *J. Intell. Fuzzy Syst.*, **39** (2020), 333–340. https://doi.10.3390/app10175841
23. F. Xu, J. Luo, M. W. Wang, G. D. Zhou, Speech-driven end-to-end language discrimination toward chinese dialects, *ACM Trans. Asian Low Resour. Lang. Inf. Process.*, **19** (2020), 1–23. https://doi.10.1145/3389021
24. Y. S. Zhao, Y. X. Duan, Convolutional neural networks text classification model based on attention mechanism, *J. Appl. Sci.*, **37** (2019), 541–550. https://doi.10.3969/j.issn.0255-8297.2019.04.011
25. G. Kim, I. Choi, Q. L. Li, J. Kim, A CNN-Based Advertisement recommendation through real-time user face recognition, *Applied Sci.*, **11** (2021). https://doi.10.3390/app11209705
26. Y. J. Yan, G. Yu, X. B. Yan, Online doctor recommendation with convolutional neural network and sparse inputs, *Comput. Intell. Neurosci.*, **2020** (2020), 1–10. https://doi.10.1155/2020/8826557
27. Z. F. Liao, H. Y. Yang, T. H. Song, S. Yu, X. F. Qi, Developer project recommendation model based on CNN-LSTM in GitHub, *Acta Electron. Sin.*, **48** (2020), 2202–2207. https://doi.10.3969/j.issn.0372-2112.2020.11.015
28. K. Saraswathi, V. Mohanraj, Y. Sruesh, J. Senthilkumar, A hybrid multi-feature semantic similarity based online social recommendation system using CNN, *Int. J. Uncertainty Fuzziness Knowl. Based Syst.*, **29** (2021), 333–352. https://doi.10.1142/S0218488521400183
29. X. L. Shen, C. H. He, X. F. Meng, Research on hybrid recommendation algorithm of restricted Boltzmann machine and weighted Slope One, *Appl. Res. Comput.*, **37** (2020), 684–687. https://doi.10.19734 /j.issn.1001-3695.2018.08.0619
30. D. Q. Du, F. Zhou, Hybrid collaborative filtering algorithm based on TimeRBM and item attribute clustering, *Appl. Res. Comput.*, **35** (2018), 239–353. https://doi.10.3969/j.issn.1001-3695.2018.02.007
31. R. J. Kuo, J. T. Chen, An application of differential evolution algorithm-based restricted Boltzmann machine to recommendation systems, *J. Int. Technol.*, **21** (2020), 701–712. https://doi.10.3966/160792642020052103008
32. Y. P. Du, C. Q. Yao, S. H. Huo, J. X. Liu, A new item-based deep network structure using a restricted Boltzmann machine for collaborative filtering, *Front. Inf. Technol. Electron. Eng.*, **18** (2018), 658–666. https://doi.10.1631/FITEE.1601732
33. F. He, N. Li, Z. G. Zhang, Recommendation algorithm based on restricted Boltzmann machine and item type, in *Proceedings of the 2018 3rd International Conference on Automation, Mechanical Control and Computational Engineering (AMCCE 2018)*, (2018), 238–244. https://dx.doi.org/10.2991/amcce-18.2018.42
34. Z. X. Chen, W. Q. Ma, W. Dai, W. K. Pan, Z. Ming, Conditional restricted Boltzmann machine for item recommendation, *Neurocomputing*, **385** (2020), 269–277. https://doi.10.1016/j.neucom.2019.12.088

35. J. Y. He, B. Ma, Based on real-valued conditional restricted Boltzmann machine and social network for collaborative filtering, *Chin. J. Comput.*, **39** (2016), 183–195. https://doi.10.11897/SP.J.1016.2016.00183