Nearest-Neighbor Restricted Boltzmann Machine for Collaborative Filtering Algorithm

Xiaodong Qian(✉) and Guoliang Liu

Lanzhou Jiaotong University, Lanzhou 730070, China
qianxd@mail.lzjtu.cn, 1185169269@qq.com

Abstract. Based on the restricted Boltzmann machine (RBM) collaborative filtering algorithm in recommendation phase easy to weaken the needs of individual users, and the model has poor ability of anti over-fitting. In this paper, the traditional nearest neighbor algorithm is introduced into the recommendation stage of RBM, use the characteristics of interest similarity, the nearest neighbor's interest is used as the target user's, strengthen the individual needs of users: First, using the traditional K-mean algorithm to find out the user's n nearest neighbors; Then, using nearest neighbor to calculate the probability of users rating grades for the non rating items; Finally, weighted average score probability to the RBM model in the process of recommendation. Using benchmark data set Movielens experimental results show that the improved RBM model with nearest neighbor can not only improve the accuracy of the model results, but also increase the ability to resist over-fitting.

Keywords: Restricted Boltzmann Machine · Nearest neighbor
Collaborative filtering · Accuracy · Over-fitting

Mathematics Subject Classification 2010: 68-W99

1 Introduction

With the rapid development of information technology in social, economic and other areas, data is increasing with hitherto unknown speed, according to the report released by the IDC show that [1], the total network data based on scale, diversity, real-time and low value density will increase from 1.8 ZB in 2011 to 35 ZB in 2020. Faced with such huge data, users can't accurately get information they want: from the point of consumers' view, consumers are overwhelmed by a flood of information, unable to find what they really need or surprise goods; from the point of business' view, the increasing amount of data led to the business can not dig out the user's real interest preferences and can not make accurate recommendations for the user's current interest, gradually lost the trust of users and the viscosity, resulting in the loss of customer resources. The above phenomena show that the increase of the amount of data results in the difficulty of data

G. Sun and S. Liu (Eds.): ADHIP 2017, LNICST 219, pp. 387–398, 2018.
https://doi.org/10.1007/978-3-319-73317-3_45

mining and reduces the efficiency of information usage, leading to the problem of information overload [2].

At present, recommendation system is one of the most common methods to solve the problem of information overload. Collaborative filtering is the most widely used and successful recommendation strategy. According to the classification of collaborative filtering algorithm by Breese [3], collaborative filtering algorithm is mainly divided into two categories: memory based collaborative filtering and model-based collaborative filtering. The recommended process of memory based collaborative filtering is carried out through the analysis of the whole user item rating matrix, as if the whole score matrix exists in the memory, the core of the method is the calculation of similarity, the similarity calculation method commonly used Pearson Correlation Coefficient [4], Vector Space Similarity [3] and Jaccard Similarity Coefficient. The process of model-based collaborative filtering based on a model obtained by learning user item rating matrix, after the recommendation of the use of the model to replace the original user rating matrix, so the core of this method is to establish a user model, commonly used models including Bayesian Belief Networks model and Clustering model, Regression model, Latent Factor model, Singular Value Decomposition model and Restricted Boltzmann Machine model etc. In recent years, the Restricted Boltzmann Machine (RBM) because of its high accuracy and can be used as the underlying of deep learning, has attracted wide attention of scholars and research.

RBM is a two layer network which is composed of a softmax visible units and a binary hidden units. The RBM model is successfully applied to collaborative filtering recommendation for the first time by Salakhutdinov et al. [5], and puts forward the Conditional Restricted Boltzmann Machine (CRBM) can highlight the importance of rating data; Georgiev and Nakov [6] directly use real values in the visible unit of RBM model as opposed to multinomial variables, reduce the training parameters in the model, and the model can directly deal with the real data; Louppe [7] analysis the impact of various parameters in the RBM model on the Netflix data set and make a detailed comparison and experiment, and in the MapReduce to realize the parallel model; Zhang et al. [8] detailed introduction the RBM model for the training and learning process, parameter selection and evaluation model based on RBM algorithm; Luo [9] analysis of RBM model from the perspective of collaborative filtering, explain the intrinsic link between the RBM and collaborative filtering; He and Ma [10] based on Real_valued CRBM (R_CRBM) training prediction score, and then applied the nearset trusted relationships to the R_CRBM model in the recommended process to improve the accuracy of prediction and parallelization scheme is proposed based on Spark platform; Chen et al. [11] using multi-layer RBM building the depth of structure model, combined the abstract feature extracted from the model with nearest neighbor recommendation method formed a recommendation algorithm which can fast convergence and have high accuracy of recommendation.

This paper analyses from point of the internal principle of the RBM model prediction process view that the excessive growth of partial weight of RBM model is the main cause of poor model discrimination. The poor discrimination of the model leads to the lack of recognition of the individual needs of the user in the final recommendation stage based on the RBM model, thus weakening the user's personalized needs in the recommendation results, eventually resulting in reduced recommendation accuracy. Aiming at the above problems, this paper solves the problem by using the nearest neighbor method, and analyzes the internal mechanism of feasibility used nearest neighbor to improve the prediction accuracy of RBM model, and gives the method to implement the improved model. Unlike Chen Da and He Jieyue, this paper innovatively utilizes the nearest neighbor calculated score probability items which not score by the target user, the probability integrated into the prediction process in RBM model. The experimental results on the MovieLens data sets show that this method can effectively improve the prediction accuracy, it is proved that this method is helpful to solve the problem of poor model discrimination caused by excessive weight growth in RBM model, enhancing the individual needs of the users, improves the recommendation accuracy; at the same time proved by experiments the model of anti overfitting ability has been greatly improved.

The Sect. 2 introduces the collaborative filtering framework based on RBM model, and analyzes the problem existed in the model; Sect. 3 gives the improved RBM model and algorithm description; Sect. 4 show the experimental results of the algorithm and analysis of the results; finally summarized the work of the paper and the existing shortcomings.

2 Collaborative Filtering Framework Based on RBM Model

The main problem of applying RBM model to collaborative filtering algorithm is how to deal with the non scoring items effectively. The literature [5] first improved the visible units of the traditional RBM model, using Softmax cell as a visible units then introduced a special visible units "Missing" to represent the user with no score project, this kind of visible units is not connected with any hidden units. Each user has a separate RBM, but all RBM corresponding to a common hidden unit, and the weights and biases between all RBM are shared (i.e. if the user U_1 and U_2 at the same time scored the film M_1, and the scores were the same, then the two users in visible units and hidden units are used in connection with a same weight). The model is shown in Fig. 1. The RBM model is an energy model, define its energy function that its energy function is Eqs. 1 and 2:

$$E(V,h) = -\sum_{i=1}^{M}\sum_{j=1}^{F}\sum_{k=1}^{K} W_{ij}^{k} h_j v_i^k + \sum_{i=1}^{M} \log Z_i - \sum_{i=1}^{M}\sum_{k=1}^{K} v_i^k a_i^k - \sum_{j=1}^{F} h_j b_j \tag{1}$$

$$Z_i = \sum_{l=1}^{K} \exp(b_i^l + \sum_{j=1}^{F} h_j W_{ij}^l) \tag{2}$$

where W_{ij}^{l} is a symmetric interaction parameter between feature j and rating k of movie i; h_j is the binary values of hidden variables j; v_i^k is the user rated movie i as k; a_i^k is the bias of rating k for movie i; b_j is the bias of feature j.

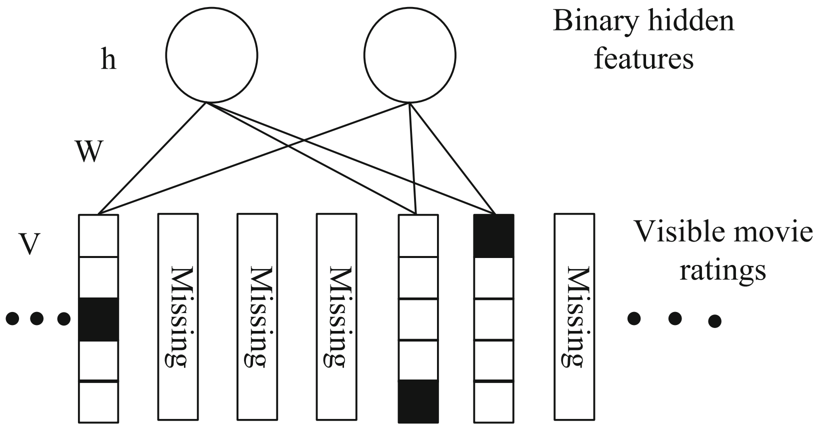

Fig. 1. Restricted Boltzmann Machine used in collaborative filtering

According to the Eqs. 1 and 2, we use the conditional probability (activation probability) for modeling 'hidden' user features h and the conditional probability (activation probability) for modeling 'visible' binary rating matrix V:

$$p(h_j = 1\,|V\,) = \sigma(b_j + \sum_{i=1}^{M}\sum_{k=1}^{K} v_i^k W_{ij}^k) \tag{3}$$

$$p(v_i^k = 1\,|h\,) = \frac{\exp(a_i^k + \sum_{j=1}^{F} h_j W_{ij}^k)}{\sum_{l=1}^{K} \exp(a_i^l + \sum_{j=1}^{F} h_j W_{ij}^l)} \tag{4}$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the logistic function.

According to Eqs. 3 and 4 we can see that the training of RBM model is to maximize the generating probability. So we use Eqs. 5, 6 and 7 to update parameters:

$$\Delta W_{ij}^k = \frac{\partial \log p(V)}{\partial W_{ij}^k} = (<v_i^k h_j>_{data} - <v_i^k h_j>_{cd-\bmod el}) \tag{5}$$

$$\Delta a_i^k = \frac{\partial \log p(V)}{\partial a_i^k} = (<v_i^k>_{data} - <v_i^k>_{cd-\bmod el}) \tag{6}$$

$$\Delta b_j = \frac{\partial \log p(V)}{\partial b_j} = (<h_j>_{data} - <h_j>_{cd-\bmod el}) \tag{7}$$

where ∂ is the learning rate. $<\bullet>_{data}$ is an expectation with respect to the distribution defined by the user-rating data, v_i^k is movie i with rating k and h_j is feature j which is computed using Eq. 3. $<\bullet>_{cd-model}$ represents a distribution of samples from running the Gibbs sampler, using Contrastive Divergence (CD) algorithm present by Hinton [12] in 2002.

After training, the Mean Field Method is used to approximate the estimation of a user's score on the non-rating movies.

$$\overset{\wedge}{p_j} = p(h_j = 1 \,|V\,) = \sigma(b_j + \sum_{i=1}^{m} \sum_{k=1}^{K} v_i^k W_{ij}^k) \tag{8}$$

$$p(v_i^k = 1|\hat{p}) = \frac{\exp(a_i^k + \sum\limits_{j=1}^{F} p_j \overset{\wedge}{W_{ij}^k})}{\sum\limits_{l=1}^{K} \exp(a_i^l + \sum\limits_{j=1}^{F} p_j \overset{\wedge}{W_{ij}^l})} \tag{9}$$

The key of RBM used in collaborative filtering is how to predict the scores of Missing items. In order to solve this problem, the above model with each user has a separate RBM, all RBM corresponds to a common hidden units, and the weights between visible units and hidden units and the respective bias of all RBM is shared. By using the method of weight and bias sharing is considered the number of movies each user has rated is far less than all the movies, so the number of identical films that have been rated among different users is less, embodied in the model that the weights of the RBM model for different users are only partially overlapped.

However, in practical applications, the data tend to show the characteristics of the "long tail", "popular movie" will be viewed and rated by more users. When a "popular movie(i)" was repeatedly score and score most of r, due to the weight of all users are shared, every user who select the "popular movie" to enter the model training and the weight of $w_{i.}^r$ will be update. The RBM model tends to reconstruct the score of r so that the model is suitable for most users.

3 Improvement RBM and Algorithm Description Based on Nearest Neighbor

For the problems raised in the Sect. 2, analysis of RBM model training and prediction process discovery: in the training phase, the CD algorithm uses the parameter update, while CD algorithm aims at learning the characteristics of reduce the reconstruction error. When the user who scored r for "popular movie(i)" enters the model, in order to reduce the reconstruction error of the model, need the corresponding weight $w_{i.}^r$ is large enough to ensure that the reconstructed data is suitable for most users. Due to the weight sharing, when the score is r for many users, after CD algorithm $w_{i.}^r$ will be updated to very large, and other weight will be significantly less than $w_{i.}^r$ for the movie; in the stage of RBM model, using the mean field method for the prediction of film score, according to Eqs. 8 and 9 we can see the size of the weight can significantly affect the prediction of film scores. When the weight of $w_{i.}^{r}$ is very large, the prediction score will tend to score r. This makes it difficult to identify some special users, resulting in the model has poor ability to identify and reduce the accuracy of prediction.

Take <the godfather> as an example, many users have seen and its evaluation is very high (assuming that most users score 5 points). In order to reduce the reconstruction error during the learning process of model, learning the weight of $w_{i.}^{5}$ will be large to apply to most users. So, namely <the godfather> of the film's score of 5 corresponds to the weight will be great, and the weight will be other scores is very small. When using the RBM model to predict the users who did not see the movie, the majority of the ratings would be 5. This leads to the fact that even users who do not like this kind of movie, but its forecast score will tend to 5 points.

According to the above analysis, reconstruction of CD algorithm in training phase and mean field method in the prediction stage is the main cause of discrimination is poor, so we can consider how to improve from two aspects of the RBM model training and prediction stage. But in the training stage of RBM model has a great influence on the model when changing its parameters, and more suitable parameter learning algorithms are also difficult to find. Therefore, this paper considers the improvement of the model prediction stage, in order to get good results.

The model is based on the mean field method in prediction, which is similar to the prediction results from the global perspective. In order to highlight the individual needs of users, should be from the user's point of view, taking into account the user's own unique interests, similar from the local point of view to strengthen user personalization. It is difficult to find out the unique interests of each user by using the user movie evaluation matrix as the historical data, so an indirect method (nearest neighbor) is used to estimate the user's interest. Users and their nearest neighbors have similar interests, so the interests of the user's nearest neighbor as a user's interest. Still take <the godfather> as an example, if the target user doesn't like this type of film, the target user may score lower on the film (2–3 points), there is a big gap between the apparently predicted by RBM model to score 5 points and the target user's true interest. In the process of model prediction model integration the nearest neighbor. The nearest neighbor, which is similar to the user's interest, does not like the film, they make score the film between 2–3 points, and according to the nearest neighbor prediction target users may also lower the score (2–3 points). Obviously lower score than predicted by the RBM model more accurate. Therefore, this paper considers the nearest neighbor into the RBM model to improve the accuracy of model prediction.

3.1 Improvement Ideas

According to the neighbor, calculate the rating level probability of the target user's un-rating film.

$$p_i^k = \frac{num^k}{sum} \tag{10}$$

where, p_i^k represents the probability of rating k of movie i which target user un-rating (The un-rating films restricted to target users who do not score and score in the nearest neighborhood, the rest of the films that nearest neighbor also did not score the probability of the film was 0); num^k is user number of rating k of movie i in nearest neighbor; sum represents the number of users in the nearest neighbor for all ratings of the movie i.

Then, the probability is added to the RBM model in the prediction process by the form of mixed weighting.

$$Q_i^k = \lambda * p(v_i^k = 1 \,|\hat{p}) + (1 - \lambda) * p_i^k \tag{11}$$

where, $p(v_i^k = 1 \,|\hat{p})$ and p_i^k respectively calculated by Eqs. 9, 10; λ is the weight of the calculated probability of the two calculation methods in the final results.

The estimated value of the target user's score for all the films is calculated based on Q_i^k.

$$R(u,i) = \sum_{k=1}^{K} Q_i^k * k \tag{12}$$

3.2 Algorithm Pseudo-code Description

Step 1. Compute nearest neighbor

Algorithm 1. k-Nearest Neighbor

1: Set $Nearest\ Neighbor's\ number \leftarrow n_{neighbor}$;
2: Use Pearson correlation coefficient compute each user's $similarity \leftarrow Sim(i,j)$;
3: Use Top $n_{neighbor}$ users with $Sim(i,j)$ as $user_i$ nearest neighbor;

Step 2. Initialization RBM

Algorithm 2. RBM-Initialize algorithm

1: Set Parameter $mini\_batches$, max_epoch, θ, ρ, CD_step;
2: Initialize W_{ij}^k with small values sampled from a zero-mean normal distribution;
3: Initialize a_i^k to the log of their respective base rates;
4: Initialize b_j with zeroes.

Determine the training set S, and according to the number of data in mini_batches data, the training sample set is divided into $S = \bigcup_{i=1}^{m} S_i$ and no intersection between S_i; Adding momentum term to update parameters not only dependent on the gradient direction of the likelihood function in the current sample, but also depends on the direction of the last parameter modification, which helps to avoid premature convergence to local optima. Literature [5] has proved that in the practical application, the parameter step is very small usually can get satisfactory results even in step 1.

Step 3. Training RBM

Algorithm 3. RBM-Training algorithm

---

repeat
  epoch=1:max_epoch
  for all mini_batch of users in S_{batch} and $S_{batch} \in S$ **do**
    for all $user \in S_{batch}$ **do**
      Translate the ratings of user to Softmax as visible units v_i^k;
      Eq. 3 compute all the hidden units h_j;
      Record samples $v_i^k h_j,\ v_i^k,\ h_j$
      Run CD algorithm to the Gibbs sampler;
      for $step = 1 : CD\_step$ **do**
        Gibbs sampler all the hidden units $<h_j>^{step}$;
        Use Eq. 4 compute all the visible units $P(v_i^k = 1\,|h\,)$;
        Gibbs sampler all the visible units $<v_i^k>^{step}$;
        Use Eq. 3 compute all the hidden units h_j;
      end for
      Record samples$<v_i^k h_j>^{step}$, $<v_i^k>^{step}$, $<h_j>^{step}$;
    end for
    Average the first samples to get $<v_i^k \cdot h_j>_{data}$, $<v_i^k>_{data}$, $<h_j>_{data}$;
    Average the second samples to get $<v_i^k \cdot h_j>_{cd-model}$, $<v_i^k>_{cd-model}$, $<h_j>_{cd-model}$;
    Use Eqs. 5,6,7 compute ΔW_{ij}^k, Δa_i^k, Δb_j;
    Update $W_{ij}^k = \rho * W_{ij}^k + \theta * \Delta W_{ij}^k$;
    Update $a_i^k = \rho * a_i^k + \theta * \Delta a_i^k$;
    Update $b_j = \rho * b_j + \theta * \Delta b_j$;
  end for
  $epoch = epoch + 1$;
  Compute the error Err_{epoch};
until $Err_{epoch-1} - Err_{epoch} > \varepsilon$ *or* $epoch = max\_epoch$

Step 4. Prediction

Algorithm 4. RBM-Initialize algorithm

1: Translate the ratings of user u to Softmax units;
2: Use Eq. 8 compute $\hat{p}_j$ for all hidden units j;
3: Use Eq. 9 compute $p(v_q^k = 1|\hat{p})$ for all $k = 1, 2..., K$;
4: Use Eq. 10 compute p_i^k;
5: Use Eq. 11 compute Q_i^k;
6: Use Eq. 12 compute $R(u, i)$;

---

4 Experimental Analysis

4.1 Data Sources

The experiment using Matlab 2015b, the data set using MovieLens 100K data set (http://www.grouplens.org) developed by Minnesota University GroupLens research group. Movielens data set is a film rating system, according to user preference score after viewing of the film are scores between 1 5, but also includes the theme of the film and user information. MovieLens 100K includes 943 users, 1682 movies and the score of 100000.

In the experiment, 80% of the data sets were randomly selected as the training set, and the remaining 20% were used as the test set. Each randomly divided data set using the standard RBM collaborative filtering algorithm as compared with the experimental reference algorithm, taking the average of the results of 10 experiments as the final prediction results. The experimental results are compared to test in the training and test sets are exactly the same situation.

4.2 Evaluating Indicator

At present, Root Mean Square Error (RMSE) is common measurement methods for evaluating the accuracy of recommender systems. The formula is as follows:

$$RMSE = \sqrt{\frac{\sum\limits_{(u,i)\in R_{test}} \left(R_{u,i} - \hat{R}_{u,i}\right)^2}{|N_{R_{test}}|}} \tag{13}$$

where, R_{test} is test data set; $R_{u,i}$ is user u actual score for movie i; $\hat{R}_{u,i}$ is user u prediction score for the movie i; $N_{R_{test}}$ represents the number of data in the test data set; The smaller the calculated results of the two evaluation indexes, the higher the accuracy of the recommendation.

4.3 Experimental Results and Analysis

Before the RBM model training and the paper algorithm, we must first determine the parameters of the model. The literature [7,8] on the choice of model parameters are introduced in detail. In this paper, we use the same experimental parameters for the paper algorithm and the RBM algorithm to ensure the accuracy and contrast of the experimental results. And the parameters are set in Table 1.

To determine the values of model parameters, because this algorithm contains the user's nearest neighbor, so it needs to consider the effect of different number of nearest neighbor users on the experimental results. So we need find the optimal user number of nearest neighbor. The calculation results are shown in Fig. 2.

From Fig. 2, the number of users nearest neighbor after reaching 10, its impact on RMSE tends to be stable. Therefore, this paper set up the user's nearest neighbor number to 20.

Table 1. Main parameters of the model

Parameter	Parameter values
Number of hidden units node	60
Weight decay coefficient	0.0005
Weight learning rate	0.001
Bias of visible units learning rate	0.001
Bias of hidden units learning rate	0.01
Iterations times	100
Iterations times of CD algorithm	3

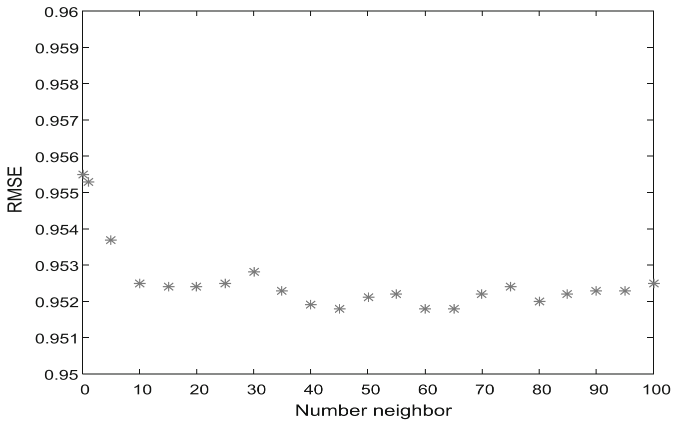

Fig. 2. Effect of different nearest neighbor number on RMSE

Figure 3 show that: the algorithm RMSE value has been less than RBM algorithm RMSE value shows that the accuracy of this algorithm is higher than that of the RBM algorithm; The improvement effect can be seen from Fig. 3, when the number of iterations is smaller and the number of iteration to achieve optimal effect (this is 40–50 times), the improved effect is more obvious. When the number of iterations reached 50, subsequent iterations will cause the overfitting problem and the value of RMSE to become larger. The recommendation accuracy of RBM algorithm will decrease rapidly, while this algorithm the recommendation accuracy decreasing speed was less than that of RBM algorithm. This show that the algorithm against over fitting ability is superior to RBM algorithm. show that: the algorithm RMSE value has been less than RBM algorithm RMSE value shows that the accuracy of this algorithm is higher than that of the RBM algorithm; The improvement effect can be seen from Fig. 3, when the number of iterations is smaller and the number of iteration to achieve optimal effect (this is 40–50 times), the improved effect is more obvious. When the number of iterations reached 50, subsequent iterations will cause the overfitting prob-

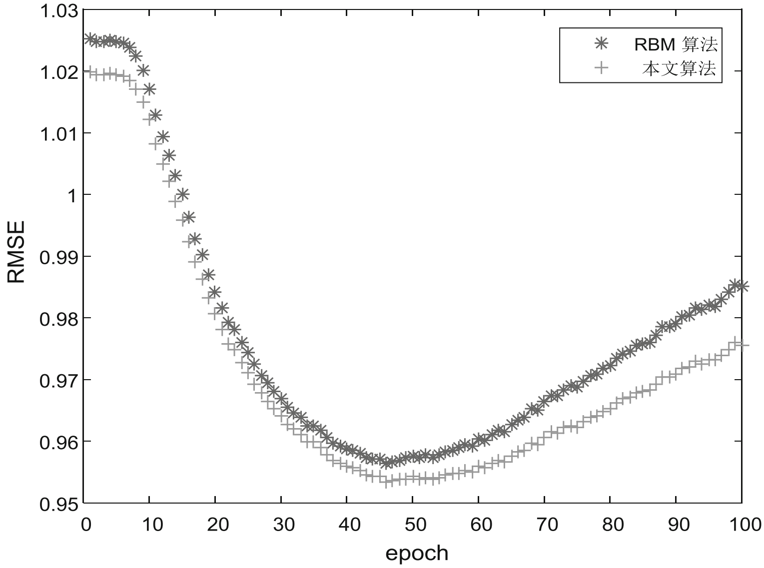

Fig. 3. Contrast experiment between the algorithm and RBM algorithm

lem and the value of RMSE to become larger. The recommendation accuracy of RBM algorithm will decrease rapidly, while this algorithm the recommendation accuracy decreasing speed was less than that of RBM algorithm. This show that the algorithm against over fitting ability is superior to RBM algorithm.

5 Conclusions

A good recommendation algorithm must first ensure the accuracy of recommendation. Therefore, improving the accuracy of recommendation is an important research direction. To provide users with the goods in line with their interests, can increase the user's satisfaction with the recommendation system, enhance the user's adhesion to the recommendation system. In this paper, the nearest neighbor is added to improve the discriminative ability of the RBM model. The experimental results show that the accuracy of the improved RBM model is better than that of the original model, and the over fitting ability of the model is improved. But this method is still not fully reflect the user interest, the target user's interest is calculated according to the nearest neighbor, and there are still some differences between the actual user and the individual interest. In the following work, will consider starting from the user's actual interest, fully tap the user's personal interests.

References

1. Gantz, J., Reinsel, D.: Digital Universe Study: Extracting Value from Chaos. IDC Go-to-Market Services (2011)
2. Wang, G.X., Liu, H.P.: Survey of personalized recommendation system. Comput. Eng. Appl. **48**(746(07)), 66–76 (2012)
3. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. New Page **7**(7), 43–52 (1998)
4. Resnick, P., Iacovou, N., Suchak, M.: GroupLens: an open architecture for collaborative filtering of netnews. In: ACM Conference on Computer Supported Cooperative Work ACM, pp. 175–186 (1994)
5. Salakhutdinov, R., Mnih, A., Hinton, G.: Restricted Boltzmann machines for collaborative filtering. In: Proceedings of the Twenty-Fourth International Conference on Machine Learning, vol. 227, pp. 791–798 (2007)
6. Georgiev, K., Nakov, P.: A non-IID framework for collaborative filtering with restricted Boltzmann machines. In: International Conference on Machine Learning, vol. 28, pp. 1148–1156 (2013)
7. Louppe, G.: Collaborative Filtering: Scalable Approaches Using Restricted Boltzmann Machine. Liége University Press, Liége (2010)
8. Zhang, C.X., Ji, N.N., Wang, G.W.: Restricted Boltzmann machines. Chin. J. Eng. Math. **32**(2), 159–173 (2015)
9. Lou, H.: Restricted Boltzmann Machines: A Collaborative Filtering Perspective. Shanghai Jiao Tong University Press, Shanghai (2011)
10. He, J.Y., Ma, B.: Based on real-valued conditional restricted Boltzmann machine and social network for collaborative filtering. Chin. J. Comput. **39**(1), 183–195 (2016)
11. Chen, D., Gao, S., Lin, Z.Q.: A survey on recommendation system algorithm based on restricted Boltzmann machine. Software **34**(12), 156–159 (2013)
12. Hinton, G.: Training products of experts by minimizing contrastive divergence. Neural Comput. **14**(8), 1771–1800 (2002)