Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org

Volume 26, Issues 3, pp. 1017–1034, DOI 10.12694/scpe.v26i3.4166

RESEARCH ON IMPROVED RBM RECOMMENDATION ALGORITHM BASED ON GIBBS SAMPLING

QIAN XIAODONG* AND LAN JIABAO†

Abstract. Restricted Boltzmann Machine (RBM) is an important tool for personalized recommendation prediction, but it ignores the power-law distribution of the Restricted Boltzmann Machine data set, the RBM algorithm can not focus on the tail data sampling of the recommended data set. Therefore, firstly, the recommended data are obtained and the data characteristics are analyzed, then the random Gibbs Sampling initial value of RBM is changed to random selection in the early iteration and the last sampling value in the later iteration, the fixed Gibbs sampling steps were replaced by single-step sampling (CD-1) and multi-step sampling (CD-5),which is Periodic Gibbs Sampling (PGS). The experiment shows that the improved Gibbs sampling initial value and the changed Gibbs sampling steps can effectively improve the sampling performance, the improved RBM algorithm is also more accurate than the original RBM algorithm, the cyclic time Restricted Boltzmann Machine (RTRBM) algorithm and the Probability Matrix Factorization (PMF) algorithm. It shows that the improved RBM algorithm is suitable for the power-law distribution of recommendation data sets, and effectively improves the accuracy of recommendation.

Key words: Recommendation Algorithm, RBM model, Gibbs Sampling, power-law distribution polynomial

1. Introduction. With the rapid development of e-commerce, the scale of consumer behavior data has been growing exponentially, making it difficult for consumers to locate satisfactory products among the vast amounts of product data. Recommendation algorithms, known for their simplicity and robustness, have become indispensable tools for assisting in recommendation decisions. To address the problem of "information overload", many companies use recommendation algorithms to intelligently mine and predict large-scale recommendation data, thereby increasing user engagement and consumption. However, as the complexity of recommendation data increases, it becomes challenging to balance the efficiency and accuracy of recommendation algorithms. Therefore, improvements to the efficiency and accuracy of the algorithms themselves are necessary.

Regarding the necessity for improvements in recommendation algorithms, this study focuses on the Restricted Boltzmann Machine (RBM) recommendation algorithm, based on the theories of complex networks and Markov chains. Considering the characteristics of recommendation data, the research aims to summarize and improve the parameter iteration algorithm of RBM, specifically the Gibbs sampling principle, to explore a more reasonable and efficient RBM algorithm. Comparative experiments between the improved RBM recommendation algorithm and the original RBM will be conducted to provide feasible references for the enhancement of recommendation algorithms.

2. Literature Review. As a widely used neural network model in practical applications, the RBM algorithm is capable of making effective recommendation predictions in recommendation scenarios. Therefore, it has been extensively studied by scholars both domestically and internationally. The analysis of improvements to the RBM recommendation algorithm can be primarily divided into two aspects: recommendation data and recommendation algorithms. The improvements are further categorized into the characteristics of recommendation data, the integration and enhancement of the RBM algorithm, and the inherent improvements of the RBM algorithm itself. Each of these aspects will be introduced and evaluated in detail.

2.1. Analysis of Recommendation Data Characteristics. Before conducting research on recommendation algorithms, analyzing the characteristics of recommendation data or the networks they form, and quantifying these features within the recommendation algorithms, can improve the efficiency of algorithm enhancements. Consequently, many scholars have carried out relevant research in this area. Garima and Rahul[1]

*School of Economics and Management, Lanzhou Jiaotong University, Lanzhou Gansu 730070, China (qianxd@mail.lzjtu.cn).
†School of Transportation, Lanzhou Jiaotong University, Lanzhou Gansu 730070, China

used text mining and sentiment analysis to extract relevant information from the text information of food and aperitif wine, and concluded the power law characteristics of the data set.Ralph and Patrycja[2] analyse the characteristics of the data set based on two-dimensional image, the relevant scaling parameters are extracted accurately, and the power-law distribution is proved. Fang et al [3] proposed a contrastive meta-learning framework (CM-HIN) based on heterogeneous information networks. This framework utilizes meta-paths and network motifs to capture both high-order and local structure information of heterogeneous information networks, thereby improving the precision of recommendation network construction. Wang et al [4] also noted the heterogeneity in recommendation data and proposed a commodity recommendation framework based on self-attention mechanism for attribute heterogeneous information network embedding. This framework learns the latent information contained in different edge types and attribute embeddings to increase the effective information in the recommendation network.

Besides focusing on the heterogeneity of recommendation networks, many scholars also delve into the latent features of these networks. For instance, Ambikesh et al [5] proposed methodology, termed GOA-k-means, amalgamates the Grasshopper Optimization Algorithm (GOA) with k-means clustering to navigate the dynamic nature of user preferences. Facilitating real-time calibration, GOA-k-means yields recommendations that adapt to users' shifting interests. By combining neural network Doc2vec and word bag BOW, Hafez et al [6] construct a multi-standard recommendation system.

2.2. Integration and Improvement of the RBM Algorithm. In the process of recommendation prediction, most studies on recommendation systems do not distinctly categorize improvements into recommendation data and recommendation algorithms. For instance, Jha et al [7] proposed a hyper-tuned Restricted Boltzmann Machine (RBM), using a contrastive divergence learning algorithm to regenerate tabular data models for enhancing recommendation accuracy. Harshvardhan et al [8] introduced a time-aware recommendation system based on unsupervised Boltzmann Machines (UBMTR) to detect latent hidden features related to the time of each rating in user movie rating data. Fachechi et al [9] calculated the relevance of recommendation data and constructed intra-layer connections for the neurons in the hidden layer of the RBM, thereby creating the Dream Boltzmann Machine (DBM). Xie et al [10] extracted user and resource features from the recommendation system to construct a multi-layer RBM network, forming a deeply stable personalized recommendation model using the Restricted Boltzmann Machine to compute recommendation results. Wu et al [11] proposed an improved hybrid recommendation algorithm based on Gaussian RBM. They used a convolutional neural network to obtain latent feature vectors of text information and rating information, merged user vectors and item vectors into a user-item matrix, and input this matrix into the visual layer of the Gaussian RBM to predict ratings.

From the aforementioned literature, it is evident that most personalized recommendation approaches using the RBM model have focused on improving the recommendation datasets or combining RBM with other algorithms, without addressing the intrinsic time costs, accuracy, and other aspects of the RBM model itself. Therefore, further research on the RBM recommendation algorithm needs to supplement and refine these aspects to enhance its overall performance.

2.3. Improvements to the RBM Algorithm. To enhance the performance of the RBM algorithm, scholars have started focusing on improving its efficiency and accuracy, particularly in two main areas: the initialization of sampling values and the optimization of parameter gradients. The original Gibbs sampling initialization uses the initial training sample values, but randomly selected training samples can lead to increased training time costs and reduced accuracy. Therefore, Tieleman [12], building on the CD algorithm, proposed the Persistent Contrastive Divergence (PCD) algorithm. PCD used the sampled values from the previous iteration as the initial values for the next sampling iteration, accelerating the convergence speed of the Gibbs sampling chain. To further speed up the PCD algorithm, Tieleman [13] introduced the Fast Persistent Contrastive Divergence (FPCD) algorithm, which includes additional acceleration parameters to enhance sampling speed.

To improve the effectiveness of initial value selection, Li et al [14] proposed the Dynamic Initial Value Algorithm (DIS), which dynamically improves the initial values for Gibbs sampling. Savitha et al [15] introduced the Online Restricted Boltzmann Machine (O-RBM), which adjusted the initial values for Gibbs sampling, constructing a probability distribution of data information to achieve unsupervised learning for recommendation predictions.

In the realm of optimizing parameter gradients in Gibbs sampling, Li et al [16] conducted an analysis of the numerical and directional errors between the approximate gradient and the true gradient of the RBM model. They devised two algorithms to mitigate these errors: the Gradient Fixed Gibbs Sampling (GFGS) training algorithm and the Gradient Fixed Parallel Tempering (GFPT) algorithm. These methods aim to adjust the numerical values and directional aspects of the approximate gradient, thereby reducing errors during training. Ma and Wang [17] identified biases in the parameter Gibbs sampling of RBM models, which fail to achieve maximum likelihood parameters. Leveraging the principle that the average of random variables approximates the expected value, they introduced the Averaged Contrastive Divergence (ACD) algorithm to mitigate the bias in maximum likelihood parameters. Kirubahari and Amali [18] utilized Bayesian Optimization (BO) to enhance the hyperparameters of Restricted Boltzmann Machines. By optimizing the number of sampling steps, they aimed to improve prediction quality. Wang et al [19] proposed the Three-Phase Gibbs Sampling (PGS) method, which involves training RBMs using different data distributions across phases to achieve more effective parameter extraction and feature reconstruction.

In addition to these advancements, research on the number of Gibbs sampling steps in RBM algorithms remains relatively limited. Li et al [14] conducted detailed research on the selection of sampling steps. However, since the choice of Gibbs sampling steps significantly impacts the time cost and training accuracy of RBM models, it remains a crucial area requiring further investigation.

2.4. Literature Review Summary. Through the analysis of the aforementioned studies, it is evident that many scholars analyze recommendation data and then quantify these features into recommendation algorithms to improve recommendation accuracy. However, due to the vastness of recommendation data and the inherent accuracy limitations of recommendation algorithms, current research still has several shortcomings. Firstly, most studies on recommendation data construct data networks and then investigate the characteristics of these networks based on their properties. However, when the characteristics of recommendation data are difficult to quantify within a network, it becomes challenging to construct a network that accurately reflects these characteristics to obtain meaningful insights from the recommendation data [1, 2, 3, 4, 5, 6]. And then, although the RBM model is widely used in the field of recommendations, most research [7, 8, 9, 10, 11]focuses on improving the data inputted into the RBM model rather than enhancing the operational speed or accuracy of the RBM model itself. Even within studies aimed at improving the RBM model [16, 17, 18, 19], which tends to overlook issues such as the problem of important data information not being learned due to the random initialization of Gibbs sampling, as well as the drawback of fixed sampling steps, which makes it difficult to improve prediction accuracy in the later stages of algorithm iteration.

To address the first issue, user social attention information is statistically analyzed to obtain the power-law distribution characteristics of the recommendation data. To tackle the second issue, the Gibbs sampling approach is adjusted by incorporating these power-law distribution characteristics. Specifically, the initial values of Gibbs sampling are set to be random in the early stages of iteration and are replaced by the previous sampling results in the later stages. Additionally, fixed Gibbs sampling steps are replaced with periodic Gibbs sampling (PGS).

3. Analysis of RBM Algorithm Improvements. Most recommendation datasets exhibit a power-law distribution, indicating that the recommendation data is primarily concentrated in the tail [20, 21]. The main algorithm in RBM (Restricted Boltzmann Machine) is Gibbs sampling, where the initial values are randomly selected from the recommendation data. This random selection fails to focus on tail data, lacking deep iterative analysis of tail data and not aligning with the long-tail characteristics of recommendation networks. Similarly, Gibbs' fixed number of sampling steps processes both the head and tail of the recommendation dataset with the same number of steps, collecting an equal amount of recommendation data. This approach does not allow for concentrated learning of tail information, resulting in insufficient learning and representation of tail user information.

Therefore, the Gibbs sampling method will be improved in terms of sampling initial values and sampling steps to enhance the recommendation performance of the RBM algorithm. The process will be as follows:

1. Provide a brief overview of the RBM algorithm principles.
2. Perform a characteristic analysis of the recommendation data in conjunction with the relevant theories of power-law distribution.

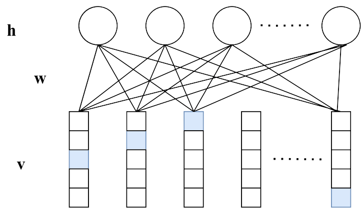


Fig. 3.1: The structure of the RBM Algorithm

3. Modify the Gibbs sampling initial values in the algorithm according to the power-law distribution characteristics of the data. Specifically, in the early stages of iteration, the initial values will be selected randomly, while in the later stages, the initial values will be the results of the previous sampling step.

4. Change the fixed number of Gibbs sampling steps to Periodic Gibbs Sampling (PGS). These improvements aim to offer a reference for enhancing the accuracy of the RBM recommendation algorithm.

3.1. Improvement of Initial Values in Gibbs Sampling for RBM.

3.1.1. Selection of Initial Values in Gibbs Sampling for RBM. Before improving the selection of initial values in the RBM algorithm, it is necessary to briefly introduce the working principles of the RBM algorithm. RBM (Restricted Boltzmann Machine) is a generative stochastic neural network based on an energy function. It is capable of transferring data through visible and hidden layers, deeply learning the latent features of users and items, making it suitable for recommendation problems. The structure of the RBM algorithm is illustrated in Fig. 3.1.

As shown in figure 3.1, v and h represent the visible and hidden units in the visible layer V and the hidden layer H, respectively. a denotes the biases of the visible units, b denotes the biases of the hidden units, and W represents the weights connecting the visible and hidden layers. In the visible layer, a node x_i is multiplied by a weight $W_{i,j}$,then a bias term b is added. The result is then passed through an activation function σ (the sigmoid function) to produce the output of the node x_i.

The energy function for each unit is:

$$E(\nu, h) = -\sum_i a_i \nu_i - \sum_j b_j h_j - \sum_i \sum_j h_j w_{i,j} \nu_i \tag{3.1}$$

Using this energy function, the joint probability distribution between the visible layer and the hidden layer can be obtained:

$$P(\nu, h) = \frac{1}{Z} e^{-E(\nu,h)} \tag{3.2}$$

In equation 3.2, Z is the normalization function that ensures the sum of probabilities over all possible states of the node set $e^{-E(\nu,h)}$ equals 1.

The units within the visible layer and the hidden layer are mutually independent. With the joint probability distribution defined, we can derive the marginal probability distribution, thereby obtaining the activation probabilities of the nodes in the visible layer and the hidden layer.

The units within the visible layer and the hidden layer are mutually independent. With the joint probability distribution defined, we can derive the marginal probability distribution, thereby obtaining the activation probabilities of the nodes in the visible layer and the hidden layer.

$$p(h_j = 1 \mid v) = \sigma(b_j + \sum_{i=1}^{m} w_{i,j} v_i) \tag{3.3}$$

$$p(v_i = 1 \mid h) = \sigma(a_i + \sum_{j=1}^{n} w_{i,j} h_j) \tag{3.4}$$

In equation 3.3 and 3.4, σ represents the sigmoid function. The phase where the hidden layer is computed based on visible layer data during training is referred to as the Positive phase, while the reverse is termed the Negative phase.

Using the visible layer input data again as the starting point, with K ($K \geq 1$) iterations of Gibbs sampling from the visible layer data known, randomly initialize $w_{i,j}$, randomly select the sampling initial value, iterate between the visible and hidden layers using equation 3.3 and 3.4, loop iterate K times, and stop the iteration. The parameter iteration formula is:

$$\begin{cases} \nabla W_{ij} = P(h_j = 1 \mid v^{(0)}) v_i^{(0)} - P(h_j = 1 \mid v^{(k)}) v_i^{(k)} \\ \nabla a_i = v_i^{(0)} - v_i^{(k)} \\ \nabla b_j = P(h_j = 1 \mid v^{(0)}) - P(h_j = 1 \mid v^{(k)}) \end{cases} \tag{3.5}$$

In equation 3.5, $\nu_i^{(0)}$ denotes the sample value, $\nu_i^{(k)}$ represents the sample value obtained after K sampling steps.

Based on the operational process of the RBM algorithm described above, it is evident that Gibbs sampling is the primary iterative algorithm used for recommendation computation in the RBM algorithm. Specifically, the initial values for Gibbs sampling are randomly selected variables from the sample data, and subsequent Gibbs sampling iterations are also based on these initial values, without including updates from the previous iteration steps.

However, the Yelp dataset, after preprocessing using the GRU model to enhance temporal characteristics, indicates that the improved dataset includes contextual feature information. On the other hand, Gibbs sampling parameter updates only consider the parameter update values from the previous step and do not incorporate earlier parameter updates. Consequently, within a limited number of iterations, it is unable to consider the effective information contained in previous parameters.

Therefore, there is a need to enhance the iterative updating method and utilization of information contained in Gibbs sampling parameters.

3.1.2. Improvement of Gibbs Sampling Initialization in RBM. Before improving the Gibbs sampling initialization in RBM, it is necessary to analyze the recommendation data and then refine the initialization based on its characteristics to enhance the accuracy of the CD algorithm.

Analysis of Recommendation Data Characteristics. When making recommendations, users recommend products to other users directly or indirectly based on social relationships. It is necessary to study the characteristics of recommendation data. Using the Yelp dataset, we examine whether there are inherent patterns such as power-law distribution, whose probability distribution is shown in equation 3.6.

$$p(x) = Cx^{-a} \tag{3.6}$$

Power-law distribution refers to a phenomenon where a small number of key items in any given entity contribute to the majority of outcomes or benefits, while the vast majority of items contribute minimally. If recommendation data exhibits power-law distribution characteristics, it indicates that only a small portion of the data contains substantial information, whereas the majority contains minimal information. Therefore, following the approach proposed by Víctor Navas-Portella et al [22], using maximum likelihood estimation to assess the cumulative degree of networks under power-law is recommended. Specifically, for practical datasets, formula 3.7 is employed to estimate the power-law distribution.

$$\alpha \simeq 1 + n[\sum_{i=1}^{n} \ln \frac{x_i}{x_{\min} - 0.5}]^{-1} \tag{3.7}$$

In equation 3.7, $\boldsymbol{\alpha}$ represents the power law exponent, and x_i represents the sample data.

Using the user social information from the Yelp dataset, user a following user b is defined as out-degree, and user a being followed by user b is defined as in-degree. Then, the power-law distribution of the frequency of user following and being followed is judged by maximum likelihood estimation. After obtaining and filtering the Yelp dataset, the power-law distribution results are shown in Fig. 3.2.

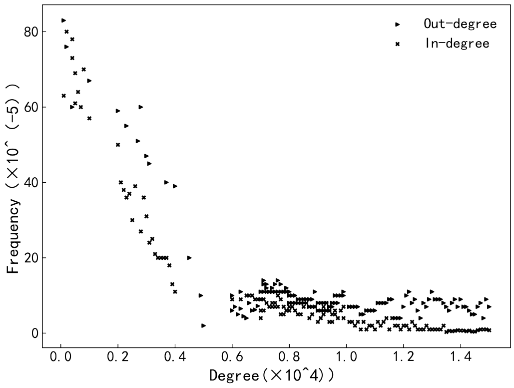


Fig. 3.2: The degree distribution of user attention

From Fig. 3.2, it can be observed that when the number of users is small, there is a higher probability of users following and being followed. As the number of users increases, the frequencies of out-degree and in-degree decrease, and a 'long tail phenomenon' appears at the distribution's tail. This is because when users make hotel choices, the majority of users only follow unfamiliar users who provide more valuable information or reciprocate with friends. Specifically, users with many followers do not necessarily follow all those who follow them. Hence, a small number of users have a high degree of followers, while the probabilities of following and being followed for the majority of users are low. Therefore, the user social information in the Yelp dataset exhibits a power-law distribution.

In summary, the user engagement data in the Yelp recommendation dataset exhibits a long-tail distribution, indicating that a small number of top users have limited social connections, while the majority of users in the tail contribute significantly to the dataset. However, in Gibbs sampling, the random selection of initial values means that if the initial value at time t is sampled from the head of the distribution, the sample at time t+1 could be from either the head or the tail. This randomness across iterations prevents Gibbs sampling from concentrating on gathering data from the tail, thereby limiting the thorough extraction of information from tail-end users.

Similarly, fixed Gibbs sampling steps treat the head and tail segments of the dataset equally, preventing deeper learning from tail-end data. Therefore, when applying the RBM model to predict recommendations from this dataset, it is essential to enhance the parameter iteration and sampling methods of the RBM algorithm to account for the dataset's long-tail characteristics effectively.

Improved Strategy for Gibbs Sampling Initialization in RBM. Due to the long-tail nature of recommendation data, it is evident that the majority of recommendations are concentrated towards the tail end. However, the current method of initializing Gibbs sampling involves randomly selecting training data from the recommendation network. This random selection could pick either head or tail data as initial values, failing to concentrate on tail data and thereby lacking in-depth analysis of this segment, which contradicts the long-tail characteristic of recommendation networks.

Therefore, it is necessary to enhance the strategy for randomly selecting initial values in Gibbs sampling as follows: during the initial training phase, use the original training data as initial values, and during subsequent phases, use the previous Gibbs sampling values as initial values. When the initial value is the original training data, the update method for Gibbs sampling initialization is shown in equation 3.8. When the original data is the previous training data, the Gibbs sampling initialization update method is shown in equation 3.9.

$$\nu^{(0)} \to \nu^{(k)} \tag{3.8}$$

$$\mathrm{v}^{(k-1)} \to \nu^{(k)} \tag{3.9}$$

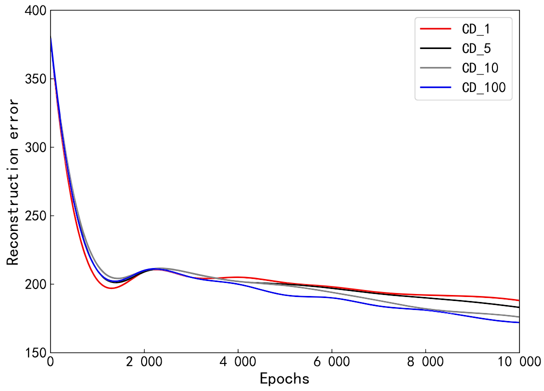


Fig. 3.3: Comparison of CD-K iterative reconstruction error

In equation 3.8 and 3.9, $V^{(0)}$ represents the randomly selected initial training value, and $V^{(k)}$ represents the training value after K steps of Gibbs sampling.

To determine the threshold for changing the Gibbs initial value sampling method, it is necessary to analyze the reconstruction error line chart of CD-K. From Fig.3.3, it can be observed that CD-1, CD-5, CD-10, and CD-100 algorithms show a gradual increase in reconstruction error after approximately 2000 iterations, followed by a slow decrease. This indicates that randomly selecting training data as initial values in the early iterations can lead to rapid convergence of the Gibbs sampling network. However, after about 2000 iterations, the training effectiveness of the RBM model decreases due to slower network convergence. Therefore, the threshold for the number of Gibbs sampling iterations is set at 2000. During iterations 1-2000, random training data is selected as the initial value for sampling, and from 2001 to 10000 iterations, the initial value for sampling is selected as the parameter sampled from the previous Gibbs sampling, ensuring rapid convergence in the early iterations and higher precision convergence in the later iterations.

3.1.3. Analysis of Improvements in Gibbs Sampling Initialization in RBM. Previous sections provided both theoretical and experimental analyses of the characteristics of recommendation data and the improvement strategy for Gibbs sampling initialization in RBM. It was demonstrated that the initial sampling values should be changed from purely random selection to using random values in the early stages of iteration and using the previous step's sampling results in the later stages. This section will validate the effectiveness of the improved Gibbs sampling initialization strategy through parameter gradient verification.

Based on the energy function of the RBM in Equation 3.1 and the marginal probability distributions of the visible and hidden layers in Equations (3.3-3.4), the parameter gradient for iterative parameter updating in the RBM network using Gibbs sampling is given by:

$$\nabla\hat{\theta}_1 = -\sum_h P(h \mid \nu^{(0)} \frac{\partial E(\nu^{(0)}, h)}{\partial\theta}) + E_{P(\nu^{(k)}|\nu^{(0)})}[\sum_h P(h \mid \nu^{(k)} \frac{\partial E(\nu^{(k)}, h)}{\partial\theta})] \tag{3.10}$$

In Equation 3.10,θ represents the general term for the parameters of the RBM.

According to the strategy for improving Gibbs sampling initialization, the updated parameter gradient is given by:

$$\begin{aligned}\nabla\hat{\theta}_2 = &-\sum_h P(h \mid v^{(0)} \frac{\partial E(v^{(0)}, h)}{\partial\theta}) + E_{P(v^{(i)}|v^{(0)})}[\sum_h P(h \mid v^{(i)} \frac{\partial E(v^{(i)}, h)}{\partial\theta})] \\ &-\sum_h P(h \mid \nu^{(i)} \frac{\partial E(\nu^{(i)}, h)}{\partial\theta}) + E_{P(\nu^{(i+1)}|\nu^{(i)})}[\sum_h P(h \mid \nu^{(i+1)} \frac{\partial E(\nu^{(i+1)}, h)}{\partial\theta})]\end{aligned} \tag{3.11}$$

According to Equation 3.11,$-\sum_h P(h \mid \nu^{(0)} \frac{\partial E(\nu^{(0)},h)}{\partial \theta}) + E_{P(\nu^{(i)}|\nu^{(0)})}[\sum_h P(h \mid \nu^{(i)} \frac{\partial E(\nu^{(i)},h)}{\partial \theta})]$represents the Gibbs sampling initialization as the original random sampling data $v^{(0)}$, with i steps of iteration, where i denotes the number of iterations and $0 < i+1 \leq k$.On the other hand, $-\sum_h P(h \mid \nu^{(i)} \frac{\partial E(\nu^{(i)},h)}{\partial \theta}) + E_{P(\nu^{(i+1)}|\nu^{(i)})}[\sum_h P(h \mid \nu^{(i+1)} \frac{\partial E(\nu^{(i+1)},h)}{\partial \theta})]$ represents the Gibbs sampling initialization as the sampling result from the previous step $v^{(i)}$, with $k-1$ steps of iteration. When Gibbs sampling reaches the i-th step, the initialization value is transformed from $v^{(0)}$ to $v^{(i)}$.

The optimization of the RBM model is achieved by finding the optimal parameters through gradient descent.Persistent Contrastive Divergence (PCD) is one of the benchmark algorithms used in RBM training. PCD is proved to be able to approach the network distribution with a small enough learning rate of network parameters. Therefore, using the result of the last parameter iteration as the initial value of the next iteration can make the training parameters change little, and make the parameter gradient decline faster and stabilize in a smaller interval.

Therefore, the parameter gradient of the improved method is smaller than the original randomly selected initial values:

$$-\sum_h P(h \mid \nu^{(i)} \frac{\partial E(\nu^{(i)},h)}{\partial \theta}) + E_{P(\nu^{(i)}|\nu^{(0)})}[\sum_h P(h \mid \nu^{(i)} \frac{\partial E(\nu^{(i)},h)}{\partial \theta})] + E_{P(\nu^{(i+1)}|\nu^{(i)})}[\sum_h P(h \mid \nu^{(i+1)} \frac{\partial E(\nu^{(i+1)},h)}{\partial \theta})] <$$

$$-\sum_h P(h \mid \nu^{(0)} \frac{\partial E(\nu^{(0)},h)}{\partial \theta}) + E_{P(\nu^{(i)}|\nu^{(0)})}[\sum_h P(h \mid \nu^{(i)} \frac{\partial E(\nu^{(i)},h)}{\partial \theta})] + E_{P(\nu^{(i+1)}|\nu^{(0)})}[\sum_h P(h \mid \nu^{(i+1)} \frac{\partial E(\nu^{(i+1)},h)}{\partial \theta})], i+1 \leq k \tag{3.12}$$

In Equation (12),

$$E_{P(\nu^{(i)}|\nu^{(0)})}[\sum_h P(h|\nu^{(i)} \frac{\partial E(\nu^{(i)},h)}{\partial \theta})] + E_{P(\nu^{(i+1)}|\nu^{(0)})}[\sum_h P(h|\nu^{(i+1)} \frac{\partial E(\nu^{(i+1)},h)}{\partial \theta})] = E_{P(\nu^{(k)}|\nu^{(0)})}[\sum_h P(h|\nu^{(k)} \frac{\partial E(\nu^{(k)},h)}{\partial \theta})]$$

That is $\nabla \overset{\wedge}{\theta}_2 < \nabla \overset{\wedge}{\theta}_1$.

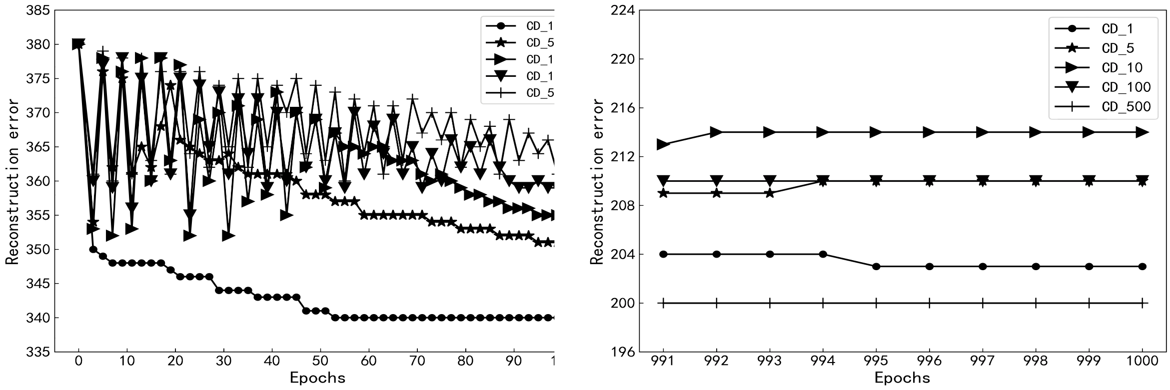
From the perspective of parameter gradients, the improvements to Gibbs sampling initialization are demonstrated to be effective.

3.2. Improving Gibbs Sampling Steps in RBM. Section 3.1.2 analysis highlighted the foundational characteristics of power-law distribution in the Yelp dataset's complex network. However, fixed Gibbs sampling steps collect an equal amount of data from both the head and tail of the dataset, failing to concentrate on learning tail-end information. This results in insufficient characterization of user information in the tail. Therefore, there is a need to improve and adjust the Gibbs sampling steps.

3.2.1. Comparison of Single-step and Multi-step Gibbs Sampling. Section 3.1 has already introduced and analyzed the principles of Gibbs sampling and the long-tail characteristics of recommendation data. Therefore, this section compares single-step Gibbs sampling with multi-step Gibbs sampling to assess their performance advantages and disadvantages. Additionally, leveraging the classical momentum algorithm (CM) to determine decision points for varying Gibbs sampling steps and formulate Gibbs sampling strategies. Finally, from the perspective of Markov chain theory, analyze and justify the rationality of improving Gibbs sampling steps.

Comparison of training errors between single-step Gibbs sampling (CD-1) and various multi-step Gibbs samplings (CD-5, CD-10, CD-100, CD-500) at epochs 1-100 and 991-1000. Utilizing the concept of reconstruction, original data is obtained from trained data, and reconstruction error serves as the evaluation metric to compare CD-K sampling results, thereby assessing the performance of the RBM network at different iteration steps. The comparison results are shown in Fig.3.4.

As shown in Figures 3.4(a) and (b), during the early stages of RBM parameter iteration, the reconstruction error of single-step Gibbs sampling (CD-1) decreases rapidly and vertically, outperforming the training error of multi-step sampling (CD-K, where ($K > 1$). This indicates that single-step sampling provides better fitting of the training data. In the initial stages of RBM training, with fewer iterations and larger recommendation



(a) Gibbs pre-sampling error

(b) Gibbs post-sampling error

Fig. 3.4: Gibbs sampling error comparison

errors, simple single-step sampling can quickly reduce the recommendation error without the need for time-consuming multi-step sampling. On the other hand, multi-step Gibbs sampling exhibits higher and oscillating reconstruction errors, suggesting larger errors during the early iteration stages. In the later stages of RBM parameter iteration, both single-step and multi-step sampling errors stabilize. However, the training error of single-step sampling is higher compared to multi-step sampling, indicating that random sampling alone can no longer significantly improve recommendation accuracy and emphasizes the need to focus on sampling the tail-end data.

Therefore, Gibbs sampling exhibits strong training capabilities for the RBM model, but its sampling steps significantly impact the algorithm's performance. So the following sections will analyze the influence of Gibbs sampling steps on algorithm performance, combining theoretical analysis with the characteristics of recommendation networks to improve Gibbs sampling.

3.2.2. Improvement Strategies for Gibbs Sampling Steps in RBM. The primary distinction between CD-1 and CD-K lies in their iteration steps, which result in different parameter iteration gradients. Specifically, CD-1 sampling concludes after Gibbs sampling step 1, while CD-K sampling involves K Gibbs sampling steps before terminating the CD algorithm. This leads to the following analysis: CD-1 exhibits good early-stage effectiveness but lacks high precision in later stages, whereas CD-K shows initial error oscillation but achieves higher accuracy in the later stages.

The magnitude of parameter gradients can measure the effectiveness of training methods for parameters. How to divide the sampling steps within the iteration interval can be judged based on the magnitude of gradient ascent to determine the Gibbs sampling steps, thus the classic momentum algorithm (CM) can be used to determine the decision points for Gibbs sampling step changes. CM adjusts the difference between accumulated velocity and current gradient $\nabla g(\theta_t)$ to decrease the target gradient, thereby accelerating the convergence speed of parameter learning. The RBM model is trained based on gradient ascent, hence CM's gradient update formula under the RBM model is shown in equation 3.13 and equation3.14.

$$\nu_{t+1} = \mu\nu_t + \varepsilon\nabla g(\theta_t) \tag{3.13}$$

$$\theta_{t+1} = \theta_t + \nu_{t+1} \tag{3.14}$$

In equation 3.13 and equation 3.14, v_t represents the accumulated velocity, $\nabla g(\theta_t)$ denotes the gradient of the objective function at the current point, θ denotes the parameters of the model, μ represents the accumulated velocity parameter, and ε represents the learning rate.

When performing single-step Gibbs sampling in the initial training stages of the RBM model, as the number of iterations increases, the numerical values of the network weights also increase. As the network weights expand, denoted by $|w| \to +\infty$, according to equation (3.3-3.4).

$$b_j + \sum_{i=1}^{m} w_{i,j} v_i \to \infty \tag{3.15}$$

$$a_i + \sum_{j=1}^{n} w_{i,j} h_j \to \infty \tag{3.16}$$

Therefore, the corresponding probabilities for the hidden layer nodes and visible layer nodes change to:

$$P(h_k = 1 \mid \nu) \to 0 or 1 \tag{3.17}$$

$$P(\mathrm{v}_k = 1 \mid h) \to 0 or 1 \tag{3.18}$$

As the number of iterations increases, the sampling probabilities of the Gibbs sampling chain gradually approach 0 or 1. That is, during each sampling, the values at each point are either 0 or 1. At this point, the transition operator of the sampling chain loses its randomness, indicating that the parameter gradient optimization direction is not the fastest. Moreover, the mixing rate of the Gibbs sampling chain decreases as the randomness of its transition operator decreases[23]. This means that as the number of iterations increases and the network weights grow, the mixing rate of the single-step Gibbs sampling chain gradually decreases, leading to reduced accuracy in later stages. Similarly, multi-step Gibbs sampling involves K repetitions of single-step Gibbs sampling, which confirms that multi-step Gibbs sampling may experience slower convergence in the early iterations.

Based on the analysis above and the results in Fig. 3.3, it is evident that improving the parameter iteration method of the RBM model in conjunction with the characteristics of the recommendation data can enhance the efficiency of the recommendation algorithm[24, 25]. Specifically, reducing the number of sampling steps in the sparse head of the data and increasing the sampling steps in the tail can yield more accurate data information.

Comparing the Gibbs sampling of CD-1, CD-5, CD-10, and CD-100 as shown in Fig.3.3, CD-1 exhibits better reconstruction error during the early iterations (1-2000 iterations) compared to multi-step Gibbs sampling. However, beyond this range, CD-10 consistently outperforms other step sizes in sampling.Therefore, the strategy for improving Gibbs sampling steps is as follows.

1. For iterations 1 to 2000, set Gibbs sampling steps $\mathrm{K}_1 = 1$. Execute single-step Gibbs sampling, using equation3.3 and equation 3.4, to compute the probability distributions of visible and hidden layers.
2. For iterations 2001 to 10000, set Gibbs sampling steps $\mathrm{K}_2 = 5$. Execute 5-step Gibbs sampling, using equation3.3 and equation 3.4, to compute the probability distributions of visible and hidden layers.

3.2.3. Analysis of the Improvement Characteristics of Gibbs Sampling Steps in RBM. Gibbs sampling is a type of Markov Chain Monte Carlo (MCMC) sampling algorithm. This section analyzes the number of Gibbs sampling steps using relevant theories from Markov chains, providing a theoretical justification for the improvement in the number of sampling steps.

In RBM model training, hidden layer nodes and input layer nodes are sampled alternately, as described in Equations 3.3 and 3.4. According to the Markov chain convergence theorem, if the number of possible states for the parameters is finite, the transition probabilities of the chain are fixed, and the parameter states can transition from any state to any other state. Therefore, when the number of steps $n \to +\infty$, the Gibbs sampling chain will converge to a stationary distribution:

$$\pi_i(x) = \pi_{i-1}(x) P = \pi_0 P^n, i \in S \tag{3.19}$$

In Equation 3.19, $\pi_i(x)$ represents the stationary distribution of the sample x. i denotes an arbitrary state, and S is the state space.

Furthermore, according to the detailed balance criterion of Markov chains, it can obtain:

$$\pi(x_i)P_{ij} = \pi(x_j)P_{ji}, \forall i, j \in S \tag{3.20}$$

In Equation 3.20,x_i and x_j represent the training data. P_{ij} and P_{ji} denote the Markov transition probabilities.

According to Equation 3.20, the stationary distribution achieved by Gibbs sampling is independent of the initial sampling values and depends only on the Markov transition probabilities. Combining this with the alternating sampling probability formulas for the visible and hidden layers in the RBM algorithm (Equations 3.3 and 3.4), it can be seen that when using Gibbs sampling for iterative training of RBM model parameters, the stationary distribution is a function of the network parameters:

$$\pi(x) = f(a, b, w) \tag{3.21}$$

In Equation 3.21, $\pi(x)$represents the stationary distribution of the sample x. $f(\theta)$ denotes the distribution of the parameters as a function.

The trained parameter values are denoted as $\hat{\theta} = (\hat{a}, \hat{b}, \hat{w})$, while the true parameter values are $\theta = (a, b, w)$. The goal of training the RBM is to adjust the network parameters such that the trained parameter values are as close as possible to the true parameter values.

$$\begin{cases} \Delta a = \hat{a} - a \\ \Delta b = \hat{b} - b \\ \Delta w = \hat{w} - w \end{cases} \tag{3.22}$$

Therefore, in the early stages of RBM iterative learning, when the trained parameter values are significantly different from the true values, multiple steps of Gibbs sampling can cause the sampled values to deviate further from the true values, resulting in multiple oscillations in the parameter values during early iterations. Single-step sampling, however, allows for faster convergence of the parameters to the true values. In the later stages of sampling, as the trained parameter values approach the true values, multiple-step sampling can more deeply capture the latent features of the recommendation data, thereby improving the accuracy of parameter training. On the other hand, single-step sampling may lead to a path with significant deviation from the true values, making CD-1 susceptible to local minima and resulting in lower parameter accuracy in the later stages of single-step sampling.

From the above analysis, it is evident that both single-step Gibbs sampling and multi-step Gibbs sampling have limitations in the RBM network training process, as demonstrated by MCMC algorithms. The experimental results, as shown in Figure 3.4, further validate this from the perspective of Markov chains. This highlights the necessity of changing the fixed sampling step size in Gibbs sampling to a phase-based variable step size.

4. The whole process of improving RBM algorithm is introduced. The improvements made to the RBM model itself primarily focus on refining the initial values and sampling steps of Gibbs sampling. Specifically, the random selection of Gibbs sampling initial values has been adjusted to a combination of random selection and the previous sampling value, and the fixed Gibbs sampling steps have been changed to staged Gibbs sampling steps. The specific improvement process is as follows: First, determine a model iteration of 10,000 steps. Then, during iterations 1-2,000, initialize sampling with randomly selected training data from the recommendation set and set the sampling step to 1. For iterations 2,001-10,000, use the previous Gibbs sampling result as the sampling value and set the sampling step to 5. The algorithm process is illustrated in Algorithm 1.

Combined with the Data pre-processing analysis, the overall process of the recommender system improvement is shown in Fig.4.1.

Algorithm 1

Step 1: Set initial values for RBM model parameters.
Step 2: Input the Yelp dataset into the visible layer of the RBM model, randomly select initial sampling values, and adjust the fixed sampling steps to staged sampling steps, setting the staged sampling step to 1.
Step 3: Within iterations 1-2000, perform interactive sampling between the RBM visible and hidden layers using equations 3.3 and 3.4, and update parameters using equation 3.5 with the sampled values.
Step 4: Repeat steps 2-3 for 2000 times to complete single-step Gibbs sampling with a sampling step of 1.
Step 5: Within iterations 2001-10000, change the randomly selected initial sampling value to the previous sampling value, and adjust the 1-step Gibbs sampling step from step 2 to 5.
Step 6: Repeat steps 3 and 5 for 8000 times. Stop parameter training at iteration 10000, completing Gibbs sampling with a step of 5.

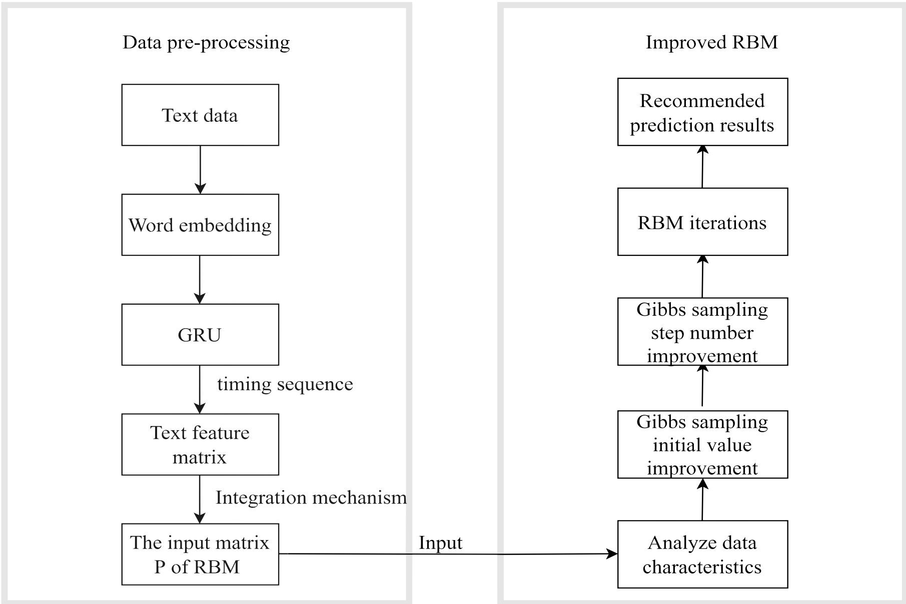


Fig. 4.1: Personalize the recommendation process

5. Experimental analysis. The previous section analyzed the power-law characteristics of the Yelp dataset and improved the Gibbs sampling's random initial values and the number of sampling steps based on the characteristics of the recommendation data. The characteristics of the improved initial values and sampling steps were analyzed, and the effectiveness of the improved RBM algorithm was discussed from a theoretical perspective. Therefore, this section aims to conduct an empirical analysis of the Gibbs sampling and RBM recommendation algorithms before and after the improvements, to further explore the effectiveness of the improved RBM recommendation algorithm.

The dataset selected for the empirical analysis is the improved Yelp dataset, which has been refined according to the data preprocessing steps shown on the left side of Figure 4.1. This includes GRU sequential processing of the textual information in the dataset, quantifying the different contributions of the text data using the attention mechanism, and integrating the text data with the rating data based on user preferences. The details of the improved Yelp dataset are shown in Table5.1.

Compare the reconstruction error metrics of the initial values and sampling steps of Gibbs sampling before and after the improvements. Additionally, analyze the recall@K, MAE, and RMSE metrics using classical recommendation algorithms such as RBM, RTRBM, PMF, and the improved RBM algorithm. Finally, analyze the experimental results to determine the effectiveness of the improved RBM recommendation algorithm strategy.

Table 5.1: Features of the improved Yelp data set

Features	Improved Yelp dataset
number of users	15328
number of items	37335
number of ratings	106821
number of comments	13796
number of mutual attention	4626

5.1. Introduction to the Dataset, Metrics, and Comparison Algorithms.

5.1.1. Introduction to the Dataset. The Yelp dataset is an online service and business review platform where users can post reviews and ratings for businesses. It includes a wealth of data such as user reviews, ratings, user information, and business information. The details of the improved Yelp dataset are shown in Table 5.1.

5.1.2. Introduction to Metrics. The effectiveness of the improvements to the RBM algorithm is measured using the following three metrics. recall@K indicates the proportion of correctly predicted positive samples out of all positive samples.

$$recall@k = \frac{TP@k}{TP@k + FN@k} \tag{5.1}$$

In Equation 5.1,$TP@k$ represents the number of correctly recommended items in the top-K recommendation list, and $FN@k$ represents the number of incorrectly recommended items in the top-K recommendation list. A higher recall@K value indicates better performance of the model's recommendations.

MAE stands for Mean Absolute Error, and RMSE stands for Root Mean Squared Error. Both metrics are used to measure the difference between predicted values and observed values in recommendation algorithms.

$$MAE = \frac{1}{T} \sum_{(u,i) \in T} \mid r_{ui} - r_{ui}^{\wedge} \mid \tag{5.2}$$

$$RMSE = \sqrt{\frac{1}{T} \sum_{(u,i) \in T} (r_{ui} - r_{ui}^{\wedge})^2} \tag{5.3}$$

In Equations 5.2 and 5.3, T represents the test set, r_{ui} and $r_{ui}^{\wedge}$ denote the true rating and the predicted rating for item by user, respectively. Generally, smaller MAE and RMSE values on the test set indicate better accuracy in the rating predictions of the recommendation algorithm.

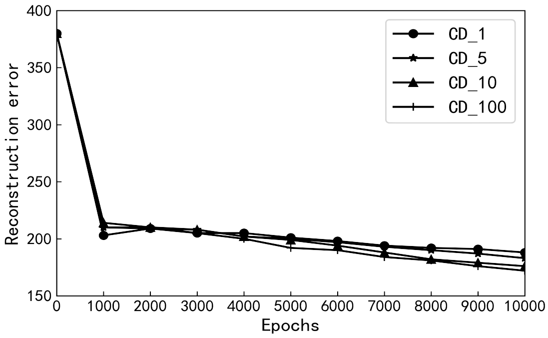
5.1.3. Introduction to Comparison Algorithms.

(1) Restricted Boltzmann Machine (RBM) Algorithm. Using the Yelp recommendation dataset, RBM processes data through visible and hidden layers, employing Gibbs sampling to deeply learn the latent features of users' hotel choices, thereby generating hotel recommendation predictions.

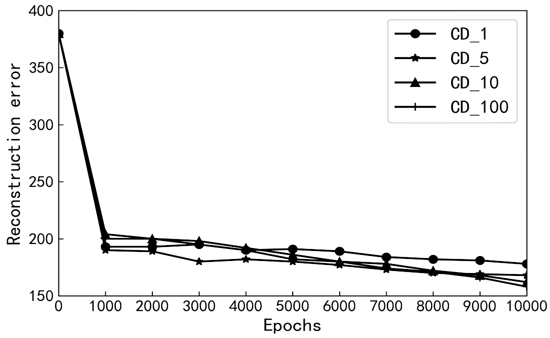
(2) Improved Restricted Boltzmann Machine (RBM) Algorithm. Building upon RBM, this improved version adjusts the sampling initial values and sampling steps of Gibbs sampling to phased sampling initial values and phased sampling steps.

(3) Recurrent Temporal Restricted Boltzmann Machine (RTRBM) Algorithm. This algorithm can be seen as an extension of RBM, where several RBMs are horizontally concatenated. It utilizes continuously sampled differences from previous sampling results to effectively handle temporal information about changes in user preferences for hotels.

(4) Probabilistic Matrix Factorization (PMF) Algorithm. This algorithm leverages both user social information and user rating and review information. It decomposes the "user-social attention" matrix into user implicit factor matrices and social attention implicit factor matrices. Using these implicit factor matrices, it predicts user ratings for hotels and generates hotel recommendation lists.



(a) Reconstruction error of Gibbs sampling initial value before improvement



(b) Reconstruction error of improved Gibbs sampling initial value

Fig. 5.1: Comparison of reconstruction error of Gibbs sampling random initial value before and after improvement

5.2. Comparative analysis of Gibbs sampling initial values. By comparing the random initial values of Gibbs sampling with the initial values set as random for the early iterations (1-2000 iterations) and as the values from the previous Gibbs sampling for the later iterations (2001-10000 iterations), the effectiveness of the improved Gibbs sampling initial values can be demonstrated. Therefore, under the premise of 10,000 iterations, the reconstruction error metrics of the initial values of Gibbs sampling before and after the improvements for CD-1, CD-5, CD-10, and CD-100 are calculated to determine the effectiveness of the improved Gibbs sampling initial values strategy. The experimental results are shown in Fig.5.1.

1. As shown in Fig.5.1(a), the selection of random initial values for Gibbs sampling does not satisfy the power-law distribution characteristics of the recommendation data. This results in higher reconstruction errors for both single-step and multi-step sampling during the 10,000 iterations compared to the improved initial values of Gibbs sampling shown in Fig.5.1(b). This demonstrates that the improvement to the Gibbs sampling initial values—using random initial values in the early iterations (1-2000 iterations) and the previous sampling results in the later iterations (2001-10000 iterations)—enables the early iterations to converge quickly and the later iterations to converge to higher precision, effectively reducing the reconstruction error throughout the entire training cycle.
2. Meanwhile, the change in CD-1 reconstruction error before and after the improvement is minimal, decreasing from 188 to 178 in the later stages. This is because single-step sampling converges extremely quickly within just one iteration, resulting in a slower decrease in reconstruction error. However, the greater the number of multi-step samplings, the larger the reduction in reconstruction error. This indicates that multi-step sampling focuses on collecting tail data in the later stages, allowing the tail data of the recommendation data to be intensively learned through iterations, thereby improving the accuracy of Gibbs sampling.
3. Considering the characteristics of the Yelp dataset, users' ratings and reviews of hotels are quite sparse, with a sparsity level exceeding 90%. Only a few users who frequently patronize hotels and actively review them receive a lot of attention. When the RBM recommendation algorithm learns the users' rating and review information for hotels, it processes all user information one by one, leading to the loss of hotel review data for the majority of users. This results in a decrease in the recommendation accuracy of RBM. Therefore, by changing the strategy of random initial values in Gibbs sampling and shifting the algorithm's attention to users with more social information and hotel ratings and reviews, the reconstruction error of Gibbs sampling can be reduced, thereby enhancing the predictive performance of the RBM recommendation algorithm.

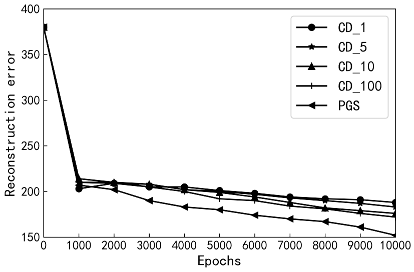


Fig. 5.2: Comparison of reconstruction error between fixed Gibbs sampling steps and staged Gibbs sampling steps

In summary, after improving the initial values of Gibbs sampling, the RBM algorithm can adjust sampling based on the characteristics of the concentrated tail data in the recommendation dataset, making the advantages of the improved RBM algorithm more apparent. The improved RBM algorithm is suitable for large-scale and sparse recommendation data, enhancing the learning speed of the algorithm while ensuring the accuracy of the recommendations.

5.3. Comparative analysis of Gibbs sampling steps. Based on the empirical analysis of the Gibbs sampling initial values mentioned above, a similar empirical comparative analysis is conducted for the Gibbs sampling steps before and after the improvements. By comparing the performance of fixed Gibbs sampling steps with the strategy of single-step sampling in the early iterations (1-2000 iterations) and multi-step sampling (CD-5) in the later iterations (2001-10000 iterations), with the initial values being random in both cases, we aim to validate the effectiveness of phased sampling steps. Therefore, under the premise of 10,000 iterations, the reconstruction error of fixed Gibbs sampling steps CD-1, CD-5, CD-10, CD-100, and phased sampling steps is calculated to determine the effectiveness of the phased Gibbs sampling strategy. The experimental results are shown in Fig.5.2.

1. As shown in Fig.5.2, during the early iterations, the phased Gibbs sampling (PGS) employs single-step sampling, resulting in significant decreases in reconstruction errors for various fixed Gibbs sampling steps (CD-K) and PGS. This rapid convergence indicates comparable performance across these methods. In the later iterations, the reconstruction error of PGS is lower than that of fixed Gibbs sampling steps. This suggests that PGS facilitates rapid convergence in the early iterations while focusing on improving sampling precision in the later iterations, effectively enhancing the operational efficiency and accuracy of the RBM algorithm. Furthermore, it demonstrates PGS's ability to adapt effectively to the power-law distribution characteristics of recommendation data by conducting multi-step sampling in the tail data, thereby improving the accuracy of Gibbs sampling.
2. According to the analysis of the reconstruction error characteristics of Gibbs sampling before improvement, fixed Gibbs sampling steps do not conform to the power-law characteristics of the Yelp dataset. From the reconstruction error characteristics of PGS shown in Fig.5.2, it is evident that throughout the 10,000 iterations, PGS consistently maintains lower reconstruction errors compared to fixed sampling steps. This indicates that PGS can effectively leverage the rich information of users in the Yelp dataset, enabling better selection and prediction of hotels.
3. During the initial 2000 iterations, whether using single-step or multi-step Gibbs sampling, the reconstruction error shows a significant decrease followed by a slight increase. This pattern is determined by the characteristics of the Yelp dataset. In the early stages of recommendation prediction, the Yelp dataset allows RBM to learn and predict recommendations based on limited information. This initial learning involves deep learning of relevant information from scratch, acquiring data features, and thus achieving a significant decrease in reconstruction error after a few Gibbs iterations. However, the majority of users in the dataset have sparse information, providing minimal evaluations on aspects

Table 5.2: Comparison of performance indicators of different recommendation algorithms

Indicators \ Recommendation algorithm	RBM		**Improved RBM**		RTRBM		PMF	
recall@K	0.728		**0.763**		0.731		0.712	
MAE	0.801	0.833	0.741	0.702	0.778	0.764	0.862	0.848
RMSE	1.291	1.227	1.114	1.031	1.201	1.126	1.429	1.354

such as hotel location and cuisine. Due to the drawbacks of fixed single-step and multi-step Gibbs sampling namely, insufficient precision in the later stages and high time costs respectively the original RBM algorithm struggles to efficiently predict recommendations based on the characteristics of the Yelp dataset.

In conclusion, throughout the entire iteration cycle, PGS demonstrates its ability to balance the parameter iteration of the RBM algorithm, allowing the network to converge to a higher precision. Specifically, in the early stages of the algorithm's recommendation learning, single-step Gibbs sampling facilitates faster parameter convergence, enabling the parameters to approach the true values. In the later stages, rapid parameter convergence through multi-step sampling also helps the parameters to converge to the true values. This indicates that the improvement in Gibbs sampling steps is meaningful and can effectively enhance the recommendation efficiency of the RBM.

5.4. Performance Comparison Analysis of Different Recommendation Algorithms. In addition to comparing the initial values and sampling steps of Gibbs sampling before and after improvement, it is also essential to conduct a comparative analysis between the improved RBM algorithm and other classical recommendation algorithms to assess the performance of the improved RBM. This section will compare the original RBM algorithm, Recurrent Temporal Restricted Boltzmann Machine (RTRBM) algorithm, Probabilistic Matrix Factorization (PMF) algorithm, and the improved RBM algorithm based on recall@K, MAE, and RMSE metrics. This comparison aims to demonstrate the effectiveness of the improved RBM algorithm.

5.4.1. The results of the experiment. A comparison was made between the classic recommendation algorithms and the improved RBM. The performance of the algorithms was evaluated using metrics such as recall@K, MAE, and RMSE. The performance comparison results for different recommendation algorithms are shown in Table 5.2. The performance of the algorithms was tested using different test set sizes, with the test sets being 70% and 80%, and the prediction sets being 30% and 20%. In Table 5.2, the MAE and RMSE values on the left side correspond to the predictions with the 70% test set, while those on the right side correspond to the predictions with the 80% test set.

5.4.2. Analysis of experimental results.

1. As shown in Table 5.2, the recall@K value of the improved RBM algorithm is 0.763, which is higher than that of the other algorithms. In terms of MAE and RMSE, when the test set is 70%, the improved RBM algorithm yields values of 0.741 and 1.114, respectively, which are lower than those of the other algorithms. When the data sparsity is further reduced, the improved RBM algorithm values decrease to 0.702 and 1.031. This clearly demonstrates that the improvements made to the sampling initial values and sampling steps of the RBM algorithm, considering the power-law characteristics of the dataset, effectively enhance both the efficiency and accuracy of recommendation predictions.
2. All metrics in Table 5.2 demonstrate that the performance of the improved RBM algorithm surpasses that of the original RBM algorithm. This indicates that compared to the original RBM algorithm, the improved RBM algorithm can deeply learn various aspects of information from the Yelp dataset, including user preferences for hotel location, cuisine, service, and social interactions, thereby achieving superior recommendation performance. During the early stages of algorithm iteration, RBM recommendation involves the transmission and updating of data between visible and hidden layers to learn

relevant features based on user preferences for hotel selection. Therefore, when randomly selecting recommendation data during Gibbs sampling to gather diverse user hotel preference information and mitigate the limitations of single sampling, it effectively reduces algorithm runtime and enhances recommendation efficiency.Similarly, in the later stages of iteration, due to the continuity of user information, the RBM algorithm needs to learn temporal user characteristic information, acquiring hotel consumption information and characteristics over time to better understand changes in user hotel preferences. Hence, considering the temporal aspects and refining the interpretation of user preference features during Gibbs sampling can effectively improve algorithm performance.

3. The recommendation prediction performance of the RTRBM algorithm lies between that of the improved RBM algorithm and the RBM algorithm. The recall@K value of RTRBM is 0.731, which is greater than the values of 0.728 for RBM and 0.712 for PMF. For both the 70% and 80% test sets, the MAE and RMSE values are 0.764 and 1.126, respectively, both of which are higher than the values of 0.833 and 1.227 for RBM, and 0.848 and 1.345 for PMF. This suggests that user preferences for hotels in the Yelp dataset change over time, showing temporal dynamics where preferences in one period influence those in subsequent periods. RTRBM demonstrates efficient capabilities in collecting and organizing user feature information and capturing temporal changes in hotel preferences. This enables it to make highly accurate recommendation predictions within a relatively short timeframe.
4. The recommendation prediction performance of the PMF algorithm is relatively poor. the recall@K, MAE, and RMSE values are 0.712, 0.848 and 1.354, respectively when the test set is 80%.This could be attributed to the PMF algorithm's reliance solely on implicit factor matrices to predict user hotel preferences, without adequately addressing the temporal dynamics of user data. Furthermore, influenced by the sparsity of the dataset, the matrix sparsity in PMF leads to decreased accuracy in prediction results.

In summary, regardless of which metric is used to assess recommendation accuracy, the improved RBM algorithm consistently outperforms others. This demonstrates that the improved RBM algorithm, which considers the characteristics of the input dataset, achieves the best recommendation performance.

6. Conclusion. Most of the studies do not consider the characteristics of real data sets when making recommendation prediction, so this paper studies the power-law distribution characteristics of recommendation data, according to this characteristic, a novel recommendation and prediction algorithm based on improved RBM model is proposed.

According to the long tail characteristic of the recommendation data, the recommendation algorithm is required to collect the recommendation tail data and to study and analyze the tail data deeply. Therefore, the main algorithm in RBM, Gibbs sampling, has been modified: random sampling for initial stages and using the previous sampling results as initial values in later stages, alongside phased sampling steps. This approach aims to concentrate on collecting data from the tail end of recommendations, iteratively analyzing this data to enhance algorithm performance.

Subsequently, the improved Yelp dataset is selected as the training data for the RBM algorithm, and ablation experiments are conducted on Gibbs sampling. The improved RBM is then compared and analyzed against the original RBM, RTRBM, and PMF algorithms. Experimental results demonstrate that the improved RBM algorithm outperforms the other three algorithms in prediction accuracy. It accurately predicts user hotel preferences, effectively enhancing the recommendation prediction capability of the RBM algorithm.

7. Acknowledgment. Natural Science Foundation of Gansu Province: Research on the Evolution of Power Law Distribution of Decentralized Social Networks (Project No. 23JRRA898)

REFERENCES

[1] Garima Gupta, Rahul Katarya., *A computational approach towards food-wine recommendations[J].* , Expert Systems with Applications. 2024, 238(15):121766.

[2] Ralph Bulanadi and Patrycja Paruch., *Outperforming RBM Feature-Extraction Capabilities by "Dreaming" Mechanism[J]. IEEE Transactions on Neural Networks and Learning Systems*, Expert Systems with Applications., 2024, 238(15):121766.

[3] Fang Yang, Tan Zhen, Chen Ziyang, et al., *Meta-learning of heterogeneous information networks for cold start recommendations[J]*, Journal of Software, 2023, 34(10): 4548-4564.
[4] Wang Honglin, Yang Dan, Nie Tiezheng, et al., *A computational approach towards food-wine recommendations[J]*, Computer Research and development,2022, 59(07): 1509-1521.
[5] Ambikesh, G., Rao, S.S. & Chandrasekaran, K., *A grasshopper optimization algorithm-based movie recommender system[J].* , Multimed Tools And Applications., 2024, 83(7), 54189-54210 .
[6] Hafez MM, Redondo RPD, Vilas AF, Pazó HO., *Multi-Criteria Recommendation Systems to Foster Online Grocery[J].* , Expert Systems with Applications., Sensors. 2021, 21(11):3747.
[7] Jha, G.K, Gaur, M, Ranjan, P, et al. , *A trustworthy model of recommender system using hyper-tuned restricted boltzmann machine[J]*, Multimed Tools and Applications., 2022, 82(2): 8261–8285.
[8] GM Harshvardhan, Mahendra Kumar Gourisaria, Siddharth Swarup Rautaray, et al., *UBMTR: Unsupervised Boltzmann machine-based time-aware recommendation system[J]*, Journal of King Saud University - Computer and Information Sciences., 2022, 34(8): 6400-6413.
[9] A. Fachechi, A. Barra, E. Agliari and F. Alemanno(delete) ,et al., *Outperforming RBM Feature-Extraction Capabilities by "Dreaming" Mechanism[J]*, IEEE Transactions on Neural Networks and Learning Systems,2022, 4(34): 1-10.
[10] Xie Miao, Deng Yulin, Lv Jie., *Personalized recommendation algorithm based on depth Restricted Boltzmann machine[J]*, Data acquisition and processing, 2022, 37(02): 456-462.
[11] Jue Wu, Lei Yang, Fujun Yang, et al., *Hybrid recommendation algorithm based on real-valued RBM and CNN[J]*,Mathematical Biosciences and Engineering, 2022, 19(10): 10673-10686.
[12] Tieleman T., *Training restricted Bo;tzmann machines using approximations to the likelihood gradient[C]// Proceedings of the 25th International Conference on Machine Learning*, New York: ACM,2008: 1064-1071.
[13] Tieleman T, Hinton G. Montreal, , *Using fast weights to improve persistent contrastive divergence[C]// Proceedings of the 26th International Conference on Machine Learning (ICML)* , Quebec, Canada: ACM, 2009: 1033-1040.
[14] Li Fei, Gao Xiaoguang, Wan Kai Fang. , *Research on RBM training algorithm based on dynamic Gibbs Sampling[J]*, ACTA automatica, 2016, 42(06): 931-942.
[15] Ramasamy Savitha, ArulMurugan Ambikapathi, Kanagasabai Rajaraman. , *Online RBM: Growing Restricted Boltzmann Machine on the fly for unsupervised representation[J]*, Applied Soft Computing, 2020, 92(12): 422-450.
[16] Li Fei, GAO Xiaoguang and WAN Kaifang., *Training Restricted Boltzmann Machine Using Gradient Fixing Based Algorithm[J]*, Chinese Journal of Electronics, 2018, 27(4): 694-703.
[17] Ma X, Wang X. , *Average Contrastive Divergence for Training Restricted Boltzmann Machines[J]*, Entropy, 2016; 18(1):35.
[18] Kirubahari, R., Amali, S.M.J., *An improved restricted Boltzmann Machine using Bayesian Optimization for Recommender Systems[J]*, Evolving Systems, 2023, 21(66): 34-45.
[19] [Wang, Q., Gao, X., Li, X. et al. , *A precise method for RBMs training using phased curricula[J]*, Multimed Tools Appl, 2022,82(3): 8013–8047.
[20] He Ying,Wang Zhuoran, Zhou Xu,et al. , *Point of interest recommendation algorithm based on weighted matrix decomposition of social geographic information[J]*, Journal of Jilin University Science (engineering edition),2023,53(09):2632-2639.
[21] Shao Changcheng, Chen Pinghua., *Points of interest recommendations that incorporate social networking and graphic content[J]*, Computer Applications, 2019, 39(05): 1261-1268.
[22] Víctor Navas-Portella, Álvaro González, Isabel Serra, et al., *Universality of power-law exponents by means of maximum-likelihood estimation[J]*,Physical Review,2019, 100(6): 062106.
[23] Bengio Y, Delalleau O., *Justifying and generalizing contrastive divergence[J]*, Neural Computation, 2009, 21(6): 1601-1621.
[24] Zhang Shusen, Wei Yudang, Liang Xun, et al. , *Power-law distribution of mobile social networks and identification of kinship[J]*, Chinese Journal of information, 2018, 32(06): 114-123.
[25] Yin H, Cui B, Li J, et al., *Challenging the long tail recommendation[J]*, Proceedings of the VLDB Endowment, 2012, 5(9): 896-907.

Edited by: Jingsha He
Special issue on: Efficient Scalable Computing based on IoT and Cloud Computing
Received: Jun 24, 2024
Accepted: Oct 15, 2024