

Intelligent Recommendation of Learning Resources Based on Kernel Canonical Correlation Analysis and Deep RBM

Li-Juan Luan*

Party and Government Office
Changchun Sci-Tech University
Changchun 130600, P. R. China
LLJ840502@126.com

Tao Jiang

College of Life Sciences
Changchun Sci-Tech University
Changchun 130600, P. R. China
496578103@qq.com

Da-Wei Sun

School of Management
SEGi University
Kuala Lumpur 47810, Malaysia
30969930@qq.com

*Corresponding author: Li-Juan Luan

Received September 9, 2024, revised April 6, 2025, accepted September 12, 2025.

ABSTRACT. *As online learning platforms evolve, the diversity of learning resources and the complexity of user needs continue to increase, making it difficult for traditional recommendation algorithms to effectively capture the deep associations between users and resources. As a result, this article proposes an intelligent recommendation method for learning resources based on KCCA and Depth Constrained Boltzmann Machine (RBM). Firstly, feature extraction is performed on learning resources through data mining methods, generating recommendation libraries for both learning resources and user personalities. The KCCA algorithm is used to analyze the characteristics of teaching resources and user personalities, selecting learning resources with high correlation coefficients to recommend to users. At the same time, this article also improved the visible layer of the restricted Boltzmann machine model by replacing binary elements with Gaussian distribution elements. The improved model can represent real values and reduce model complexity; Finally, this article integrates the scoring data generated by the improved restricted Boltzmann machine model and the scoring data generated by nuclear canonical correlation analysis through linear weighting. Through experiments, it has been proven that this hybrid recommendation algorithm reduces the average absolute error, improves recommendation accuracy, and alleviates the problem of data sparsity.*

Keywords: kernel canonical correlation analysis; depth RBM; recommendation system

1. **Introduction.** As information technology advances rapidly, online learning platforms have become an important component of modern education. These platforms greatly enrich learners' choices by providing a large variety of learning resources, including video

courses, e-books, and online quizzes, and interactive discussions. However, with the diversity and complexity of learning resources, users' personalized learning needs are also increasing. How to find learning content that matches user needs among massive resources has become an important challenge faced by online learning platforms. Therefore, personalized learning resource recommendation systems [1] have emerged, with the goal of providing accurate and personalized learning resource recommendations by analyzing users' learning behavior and interests, in order to improve learners' participation and learning efficiency.

Recommendation systems can generally be classified into three categories: collaborative filtering, content-based recommendations, and hybrid methods. Collaborative filtering algorithms [2] primarily use users' historical behavior data to assess user similarities or resource correlations, in order to achieve recommendations [3]. However, this method faces the problem of data sparsity, which makes it difficult for the system to make effective recommendations when there is limited interaction data between users and the system. In addition, collaborative filtering methods mainly rely on user historical ratings and ignore the content characteristics of resources, which limits their ability to handle multimodal data. Content based recommendation methods analyze the content features of resources (such as text, audio, video, etc.) to make recommendations, which can usually address the issue of cold start. However, recommendation methods that solely rely on content features are difficult to capture users' deep-seated preferences, which can easily lead to the singularity and limitations of recommended content. To address this issue, hybrid recommendation algorithms merge collaborative filtering with content-based techniques, enhancing recommendation accuracy and diversity by leveraging the strengths of both approaches [4]. Although the traditional methods mentioned above have achieved some success, they often struggle to effectively handle the nonlinear relationship between users and resources when faced with complex and multimodal learning resource data. Online learning resources have the characteristics of diversity, heterogeneity, and high dimensionality, and traditional linear models are difficult to capture the deep associations between users and resources. In addition, with the continuous changes in user needs and rapid updates of resources, how to dynamically adapt to users' personalized needs has become another challenge faced by recommendation systems.

In response to these issues, deep learning based recommendation systems have gradually gained attention in recent years. Deep learning models can automatically extract features from data through multi-layer neural network structures, particularly when handling complex nonlinear and high-dimensional data, showing significant advantages. For example, the RBM, as a deep learning model, can capture potential features of users and resources through unsupervised learning and generate user preference models for recommendation tasks [5]. However, traditional RBM models have certain limitations when dealing with continuous real valued data, as their visible layers typically use binary units, making it difficult to effectively

represent complex real valued features in learning resources. Therefore, how to improve the RBM model to better handle continuous data has emerged as a key research focus for enhancing the performance of recommendation systems. In addition, the correlation analysis of multimodal data is increasingly being emphasized in recommendation systems. The learning behavior, interest preferences, and content characteristics of learning resources of users are often distributed in different data modalities, such as text, images, and videos. In order to effectively capture the correlations between these modalities, Kernel Canonical Correlation Analysis (KCCA) provides an effective solution [6]. KCCA is a nonlinear mapping algorithm that can handle multimodal data by finding the maximum

correlation between different modal data in high-dimensional space, achieving accurate matching between user needs and resource features.

In light of these considerations, this paper presents a hybrid recommendation algorithm that integrates KCCA with an enhanced Depth RBM to enhance the personalized recommendation performance of learning resources on online learning platforms. Specifically, this article first extracts features from learning resources through data mining techniques, and constructs a learning resource library and a user personality feature library. Through the KCCA algorithm, the system can effectively analyze the nonlinear correlation between users and resources, and select learning resources that are highly relevant to user needs for recommendation. Meanwhile, in order to better handle the continuous real valued features in learning resources, this paper has improved the traditional RBM model. The visible layer of traditional RBM models uses binary units, which are suitable for discrete data, but have limited effectiveness in dealing with real valued features in learning resources such as resource ratings and time lengths. This article replaces the visible layer elements of the RBM model with Gaussian distribution elements to better represent continuous real valued features, thereby improving the model's performance and processing capabilities. In addition, the improved RBM model generates a user preference model through unsupervised learning to predict users' ratings of resources. In order to fully utilize the advantages of KCCA and RBM models, this paper proposes a scoring strategy based on linear weighted fusion. Specifically, the system weights and fuses the rating data generated by KCCA with the rating data generated by the improved RBM model, thereby further improving the accuracy and personalization of recommendations. Through this fusion strategy, the system can not only capture the deep nonlinear relationship between users and resources, but also effectively address the issue of data sparsity and improve the robustness of the recommendation system. The experimental results demonstrate that the hybrid recommendation algorithm based on KCCA and deep RBM proposed in this paper outperforms traditional recommendation algorithms in multiple evaluation metrics, particularly in reducing mean absolute error (MAE) and improving recommendation accuracy. In addition, this method has demonstrated good adaptability and robustness in dealing with data sparsity and large-scale complex data, providing a new and effective solution for personalized recommendation in online learning platforms.

1.1. Related work. At present, the research on recommendation systems in the field of education mainly began in the late 1980s abroad, and their practical application in China mainly emerged in the early 21st century. In 2013, Aher and Lobo [7] used a combination of clustering techniques and association rules to generate personalized recommendations for new users who

were just starting to learn. In 2018, Ibrahim et al. [8] introduced a hybrid filtering system framework based on ontology, which integrates collaborative filtering and content-based filtering to deliver personalized course recommendations to users. However, these traditional recommendation methods have poor performance and cannot fully explore users' deep preference features.

In recent years, deep learning technology, a key branch of artificial intelligence, has experienced rapid development. With ongoing innovation and optimization of deep learning models, it is widely used in fields such as image recognition, natural language processing, and recommendation systems. In 2017, Berg et al. proposed the Graph Convolutional Matrix Completion framework from the perspective of graph link prediction. This framework utilizes graph convolutional networks to directly model user and project/course ratings, but this method cannot effectively capture high-order collaboration information between users and projects. In 2018, Shu et al. [10] introduced a content-based recommendation

algorithm utilizing Convolutional Neural Networks (CNN). CNN is employed to predict latent factors from the textual information of multimedia resources. The process begins with determining the input and output for the CNN. A language model is used as the input, while a latent factor model is developed for the output. Additionally, the split Bregman iterative method is applied to solve the model. In 2021, Chen and Deng [11] investigated students' interests and hobbies based on their online learning behavior to recommend suitable video materials. For students lacking evaluation records, they used data mining association rule algorithms to suggest videos of interest. For those with evaluation records, they applied a machine learning-based collaborative filtering algorithm, leveraging the Pearson correlation coefficient to identify and recommend similar video materials. These methods cater to different scenarios, ensuring that all students on the platform receive personalized recommendations. Through this application, data search time can be reduced, platform stickiness can be improved, information overload can be solved, and personalized needs of learners can be met. In 2022, Zhong and Ding [12] focused on researching how to develop an effective personalized teaching resource recommendation system to enhance recommendation accuracy. Based on the data on learning behavior of online learning platforms, the author explored the classic cold start problem of popular collaborative filtering algorithms and improved the algorithm according to the data characteristics of the platform.

Although significant research achievements have been made in the area of learning resource recommendations, existing traditional recommendation algorithms are difficult to effectively capture deep level associations between users and resources, and data sparsity problems still exist. To tackle these issues, this paper introduces an intelligent learning resource recommendation approach using KCCA and a Depth Constrained RBM.

1.2. Contribution. This article proposes an intelligent learning resource recommendation approach using KCCA and an improved Depth Constrained Boltzmann Machine (RBM), and makes the following main contributions to the domain of recommendation systems:

1. Innovative recommendation method combining KCCA and RBM models: A novel hybrid recommendation algorithm is proposed by combining kernel canonical correlation analysis with an improved depth constrained Boltzmann machine model. This method fully utilizes KCCA's correlation analysis between user and learning resource features to optimize recommendation performance.
2. Improved RBM Model: The traditional RBM model has been improved by replacing binary units in the visible layer with Gaussian distribution units, thereby enhancing the model's performance in processing real valued data and reducing its complexity, making it more suitable for learning resource recommendation scenarios.

3. Multi source information fusion: By linearly weighting the scoring data of KCCA and improved RBM models, and considering the advantages of both methods, the recommendation accuracy is further improved.

4. Relieve the problem of data sparsity: By introducing the above hybrid recommendation method, the problem of data sparsity in the recommendation system has been effectively alleviated, and the recommendation performance of the system in the case of data scarcity has been improved.

5. Experimental verification and performance improvement: The results of the experiments indicate that this method has significant advantages in reducing mean absolute error (MAE) and improving recommendation accuracy, providing more personalized and accurate recommendation services for online learning platforms.

2. Theoretical analysis.

2.1. **Canonical correlation analysis.** CCA is widely utilized to measure the relationship

between two multidimensional datasets [13, 14]. There is a set of samples

$$X = (x_1, x_2, \dots, x_M) \tag{1}$$

and $Y = (y_1, y_2, \dots, y_M)$ that have undergone mean to zero processing. The correlation of typical variable X and the correlation of typical variable Y are referred to as correlation X^* , respectively. X^* and

$$X^* = \varphi^T(X) \tag{2}$$

$$Y^* = \varphi^T(Y) \tag{3}$$

where φ_x and φ_y represent the coefficients associated with the maximum correlation value, and T denotes the transpose. The objective of the CCA method is to combine Equations (1) and (2) to identify the maximum correlation between the two variables. If the covariance matrices of X and Y are represented by C_{xx} and C_{yy} , respectively, and the cross-covariance matrix is denoted as C_{xy} , then the maximum correlation coefficient ρ is: $\rho = \varphi_x^T C_{xy} \varphi_y$

$$C_{xx} = \text{cov}(x, x) = E(x^2) - (E(x))^2 \tag{4}$$

$$C_{yy} = \text{cov}(y, y) = E(y^2) - (E(y))^2 \tag{5}$$

$$C_{xy} = \text{cov}(x, y) = E(xy) - E(x)E(y) \tag{6}$$

where $E()$ denotes the expected value, and $\text{cov}()$ represents the covariance calculation. The constraints for the above calculation are:

$$\begin{cases} \max_{\varphi_x, \varphi_y} \varphi_x^T C_{xy} \varphi_y \\ \text{s.t.} \quad \varphi_x^T C_{xx} \varphi_x = 1 \\ \quad \quad \varphi_y^T C_{yy} \varphi_y = 1 \end{cases} \tag{7}$$

by applying the Lagrange function method to solve Equation (7), obtain the optimal values of φ_x and φ_y .

2.2. **Kernel canonical correlation analysis.** Building on CCA, the Kernel Canonical Correlation Analysis (KCCA) method introduces a kernel function to analyze the correlation between features of different dimensions [15,16]. The primary structure is shown in Figure 1. If the mapping function ϕ_x satisfies $K(x, y) = \langle \phi(x), \phi(y) \rangle$, then $K(x, y)$ is known as the kernel function. Apply a ϕ function mapping to the normalized samples $X = (x_1, x_2, \dots, x_N)$ and $Y = (y_1, y_2, \dots, y_N)$, and then solve the correlation coefficient between samples X and Y according to Equation (1). Equation (3) becomes:

$$\rho = \frac{\varphi_x^T K(x, y) \varphi_y}{\sqrt{\varphi_x^T K(x, x) \varphi_x \varphi_y^T K(y, y) \varphi_y}} \tag{8}$$

the constraint conditions for the above calculation are:

$$\begin{cases} \max_{\varphi_x, \varphi_y} \varphi_x^T K(x, y) \varphi_y \\ \text{s.t.} \quad \varphi_x^T K(x, x) \varphi_x = 1 \\ \quad \quad \varphi_y^T K(y, y) \varphi_y = 1 \end{cases} \tag{9}$$

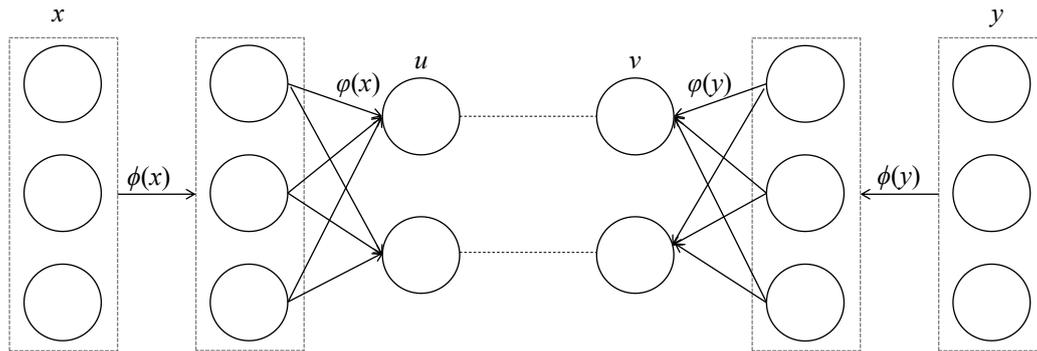


Figure 1. Schematic diagram of kernel canonical analysis

2.3. Restricted Boltzmann machine model. RBM originates from the BM, a stochastic recurrent neural network. The BM network architecture is shown in Figure 2. Each neuron has an undirected connection [17, 18]. Although BM has achieved good results in the actual unsupervised learning process, due to the unrestricted connection of neurons, the hidden and visible units are not only fully connected between layers, but also fully connected outside layers, resulting in the BM model being too complex. As the number of neurons grows, it becomes more difficult for the BM model to reach a steady-state distribution, resulting in excessively long training times and inability to be practically applied.

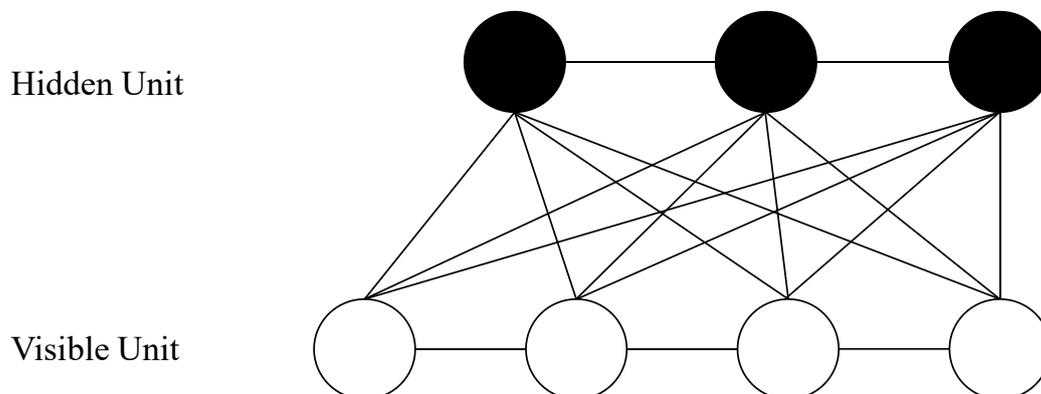


Figure 2. Simple BM network model

To address the issue of excessively long training time for BM models. Hinton [19] proposed an improved BM and named it RBM. The principle is to divide BM into two layers, namely the visible layer and the hidden layer, cancel intra layer connections, reduce model complexity, and enable effective learning and problem-solving. However, there was no fast training algorithm at that time, and RBM training was too time-consuming, so it did not receive widespread attention in the academic community until Professor Hinton introduced a rapid learning algorithm based on contrastive divergence [20], which accelerated the training speed of RBM models, improved their learning ability, and attracted the attention of many researchers. RBM is used for unsupervised learning. Figure 3 shows the RBM network architecture.

Throughout the RBM model's training process, the visible layer edge probability distribution, commonly referred to as the likelihood function, often affects the learning efficiency of the RBM model:

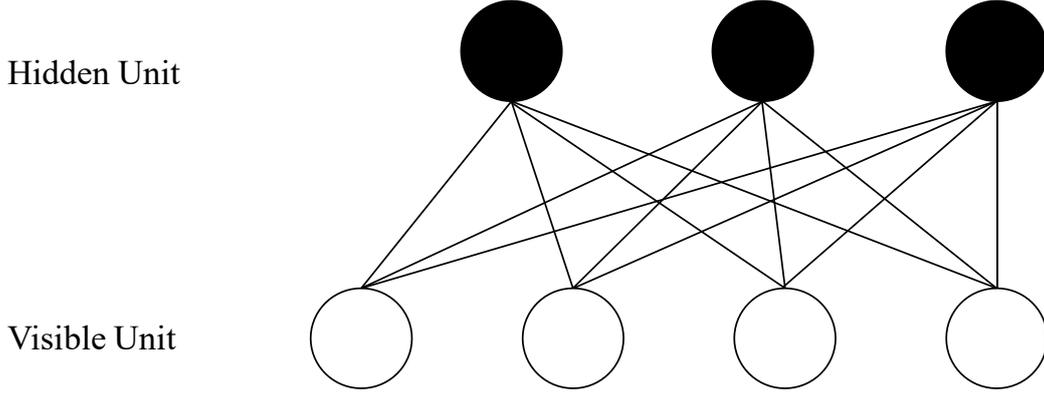


Figure 3. RBM network model diagram

$$p(v) = \sum_h p(v, h) = \frac{1}{Z} \sum_h e^{-E_\theta(v, h | \eta)} \quad (10)$$

where $\eta = (W, a, b)$ is the set of RBM model parameters. Since RBM has an undirected bipartite graph structure, the edge probability distribution of the hidden layer is similarly:

$$p(h) = \sum_v p(v, h) = \frac{1}{Z} \sum_v e^{-E_\theta(v, h | \eta)} \quad (11)$$

In actual process calculations, in order to calculate the edge probability distributions of visible and hidden layers, it is important to first establish the value of the allocation function Z . However, to determine the value of the partition function Z , it is necessary to calculate the probability the distribution of all visible and hidden neurons. The spatial complexity of this process is $O(2^{g+f})$, and the directly calculated spatial complexity is too high, resulting in high computational costs. Here, Gibbs sampling can be used to approximate the value of Z , which reduces space complexity but increases time complexity. The following text will introduce a learning method based on contrastive divergence to alleviate the problem of long training time for Gibbs sampling.

Because the RBM model is non connected within layers with full connections between layers, the expectations of visible layers and hidden layers are independent from one another. Once the expected value of the visible layer units is determined, the expected value of the hidden layer units can be computed using the visible layer units [21, 22]. Likewise, when the expected value of the hidden layer units is given, the expected value of the visible layer units can also be derived through the hidden layer units. Therefore, when the condition of each unit in the visible layer is determined, the likelihood of activation for a hidden layer unit is $p(h_j = 1 | v)$. Similarly, when each unit in the hidden layer is determined, the likelihood of activation for a visible layer unit is $p(v_i = 1 | h)$. The formula is as follows:

$$p(h_j = 1 | v) = \sigma \left(b_j + \sum_{i=1}^f v_i W_{ij} \right) \quad (12)$$

$$p(v_i = 1 | h) = \sigma \left(a_i + \sum_{j=1}^g h_j W_{ij} \right) \quad (13)$$

where f represents the number of neurons in the visible layer, g represents the number of neurons in the hidden layer, v_i represents the expected value of the i -th neuron in the

visible layer, a_i represents the bias of the i -th neuron in the visible layer, h_j represents the expected value of the j -th neuron in the hidden layer, b_j represents the bias of the j -th neuron in the hidden layer, and W_{ij} represents the connection weight between the i -th neuron in the visible layer and the j -th neuron in the hidden layer.

If the visible units are independent Gaussian variables, the visible layer units can take real values, as demonstrated below:

$$p(v_i | h) = \mathcal{N}\left(b_j + \sum_j h_j W_{ij}, 1\right) \quad (14)$$

where $\mathcal{N}\left(b_j + \sum_j h_j W_{ij}, 1\right)$ represents Gaussian distribution.

The CD method is now the standard training algorithm for RBM models. The parameter updates in RBM using the CD algorithm are as follows:

$$\Delta W_{ij} = \varepsilon (\langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}}) \quad (15)$$

$$\Delta a_i = \varepsilon (\langle v_i \rangle_{\text{data}} - \langle v_i \rangle_{\text{model}}) \quad (16)$$

$$\Delta b_i = \varepsilon (\langle h_j \rangle_{\text{data}} - \langle h_j \rangle_{\text{model}}) \quad (17)$$

where ε represents the learning rate, $\langle \cdot \rangle_{\text{data}}$ represents the expected value of the hidden layer once the state of the visible layer neurons is specified, $\langle \cdot \rangle_{\text{model}}$ represents the expected value of the model reconstructed with the CD algorithm.

2.4. Collaborative filtering recommendation algorithm based on RBM. In 2007, the RBM model was initially used in the field of recommendation to address the issue of data sparsity in collaborative filtering algorithms. At that time, the application of RBM model in recommendation systems had the following problems:

- The recommended range for scoring data in the dataset is [1, 5]. Traditional RBM models have binary units that can only take 0 and 1 as visible units. How to solve the problem of data input is a difficult point that researchers must address.
- Due to data sparsity, there are a large number of missing rating data in the user rating matrix of recommendation systems. What neurons should be used to represent the missing rating data is also a problem that researchers need to consider.

In response to the above issues, proposed a collaborative filtering algorithm based on RBM, which solves the data input problem by representing individual neurons in the visible layer as K -length neuron vectors; At the same time, missing scoring data is represented by special neurons (Missing) that are not connected to the RBM hidden layer units.

When the state of the hidden layer neuron is set, the activation probability for the i -th neuron in the visible layer can be determined using the following formula:

$$p(v_i^k = 1 | h) = \frac{\exp\left(a_i^k + \sum_{j=1}^g h_j W_{ij}^k\right)}{\sum_{l=1}^K \exp\left(a_i^l + \sum_{j=1}^g h_j W_{ij}^l\right)} \quad (18)$$

where f represents the number of visible layer neuron vector groups, which is the number of neurons in the visible layer in the original RBM model; K represents the maximum score of the dataset; G remains unchanged, indicating the number of hidden layer neurons; when $v_i^k = 1$, k indicates that the user's rating for item i is k ; the range of values is $k = \{1, 2, \dots, K\}$; a_i^k is the bias of the k -th neuron in the i -th visible layer neuron vector; b_j is the bias of the j -th neuron in the hidden layer; W_{ij}^k is the connection weight between

the k -th neuron in the i -th visible layer neuron vector and the j -th neuron in the hidden layer.

Professor Hinton proposed a fast learning algorithm based on contrastive divergence (CD) with the aim of solving the problem of increasing the number of sampling steps in Gibbs sampling when the feature dimension of data samples is too high in Monte Carlo algorithms. A fast learning algorithm based on CD only requires one step of Gibbs sampling to achieve satisfactory results, greatly improving training efficiency. At the beginning of the CD algorithm, the data samples are input into the visible units of the RBM model, and then the probability distribution function of the hidden units is calculated using Equation (13). Similarly, after knowing the state of the hidden layer unit, the value of the visible unit is obtained through Equation (14) and iteratively updated. Using the CD fast learning algorithm to adjust the parameters of the RBM collaborative filtering model, the formula is as follows:

$$\Delta W_{ij}^k = \varepsilon (\langle v_i^k h_j \rangle_{\text{data}} - \langle v_i^k h_j \rangle_{\text{model}}) \quad (19)$$

$$\Delta a_i^k = \varepsilon (\langle v_i^k \rangle_{\text{data}} - \langle v_i^k \rangle_{\text{model}}) \quad (20)$$

$$\Delta b_j = \varepsilon (\langle h_j \rangle_{\text{data}} - \langle h_j \rangle_{\text{model}}) \quad (21)$$

3. Hybrid recommendation algorithm based on kernel canonical correlation analysis and deep RBM. An RBM is a probabilistic model with a two-layer architecture, where the visible layer encodes the data and the hidden layer extracts features. Introducing real valued conditions into the RBM model simplifies the traditional RBM model. In big datasets, the recommendation performance based on project RBM model is much better than that based on user RBM model. This article proposes a KCCA-RBM model based on the above ideas.

Because visible layer units represent rating data in real values, a visible layer real value energy function has been added to the traditional RBM energy model. The energy calculation formula for the KCCA-RBM model is as follows:

$$E(v, h | \theta) = \frac{1}{2} \sum_{i=1}^f v_i^2 - \sum_{i=1}^f a_i v_i - \sum_{j=1}^g b_j h_j - \sum_{i=1}^f \sum_{j=1}^g v_i W_{ij} h_j \quad (22)$$

where f denotes the number of real-valued units in the visible layer, corresponding to the number of neurons in the original RBM model's visible layer. G remains unchanged, representing the number of hidden layer neurons; v_i is the value of the i -th visible layer real valued unit, and a_i is the bias of the i -th visible layer real valued unit; h_j is the value of the j -th neuron in the hidden layer, and b_j is the bias of the j -th neuron in the hidden layer; W_{ij} connects the weights between the i -th real-valued unit in the visible layer and the j -th neuron in the hidden layer; $\eta = (W, a, b)$ is the parameter set of KCCA-RBM model.

Due to the unique structure of the KCCA-RBM model, which lacks intra-layer connections but has fully connected inter-layers, the activation probabilities of the hidden layer units are independent. After inputting the rating data into the model, the activation probabilities of the hidden layer units can be calculated using the energy function. The formula is:

$$p(h_j = 1 | v) = \sigma \left(b_j + \sum_{i=1}^f v_i W_{ij} \right) \quad (23)$$

Given the state of a hidden layer unit, the value of the i -th real-valued visible layer unit is:

$$p(v_i | h) = \mathcal{N}\left(a_i + \sum_{j=1}^g h_j W_{ij}, 1\right) \quad (24)$$

The traditional contrastive divergence algorithm has the problems of too many iterations and easy generation of local optimal solutions. This article proposes a method of increasing the impulse factor to solve the above problems. When updating weights, adding the weights from the previous update can make the weight update method closer to the direction of model iteration, decrease the number of RBM iterations and enhance the model's resistance to overfitting. The parameter update formula is as follows:

$$\Delta W = \varepsilon(\langle vh_j \rangle_{\text{data}} - \langle vh_j \rangle_{\text{model}}) + \lambda \Delta W^{(n-1)} \quad (25)$$

$$\Delta a = \varepsilon(\langle v \rangle_{\text{data}} - \langle v \rangle_{\text{model}}) + \lambda \Delta a^{(n-1)} \quad (26)$$

$$\Delta b = \varepsilon(\langle h \rangle_{\text{data}} - \langle h \rangle_{\text{model}}) + \lambda \Delta b^{(n-1)} \quad (27)$$

where ε represents the learning rate, λ is the impulse factor, $\langle vh_j \rangle_{\text{data}}$ represents the expected value of the hidden layer given the visible layer, and $\langle vh_j \rangle_{\text{model}}$ represents the expected value of the reconstructed model.

4. Experimental results and analysis.

4.1. Dataset and evaluation indicators. The experimental dataset used in this article is provided by the MOOCCube repository publicly available at Tsinghua University. This dataset contains user historical behavior data, user attribute data, course attribute data, and user learning course time record data, all of which are saved in JSON format. By filtering the data, some data with interaction times greater than 10 were obtained and named the MOOCCube dataset.

MOOC dataset: including 58,760 browsing and learning records of 31,076 users, and 300 course data. The specific information is presented in Table 1.

Table 1. Dataset

Dataset	User	Course	Interactive data
MOOCCube	6,549	689	97,833
MOOC	31,076	331	58,760

The experimental data is divided into training, testing, and validation sets with an 8:1:1 ratio. The embedding vector size is 64, the learning rate is 10^{-3} , the batch size is 2,048, and the total number of training epochs is 150. For experimentation, the maximum time interval T in the aggregator layer is set to $T = \{1, 16, 32, 64, 256\}$. The algorithm's performance is evaluated using recall rate (Recall@K) and normalized discounted cumulative gain (NDCG@K).

In top- k recommendations, Recall measures the proportion of courses included in the recommendation list of the test set that users have actually learned. A higher proportion indicates better recommendation performance. The calculation method is as follows:

$$\text{Recall}@k = \frac{|C_k(u) \cap C^+|}{|C^+|}, \quad (28)$$

where $C_k(u)$ represents the optimal course list recommended for a random user u in the test set, and C^+ represents the set of all courses learned by the user.

NDCG (Normalized Discounted Cumulative Gain) is used to evaluate the degree to which the position of courses in the recommendation list affects the recommendation effect. A higher NDCG value indicates that the recommended items more closely match user preferences, reflecting better recommendation quality. The calculation method is as follows:

$$\text{NDCG}@k = \frac{\text{DCG}_k}{\text{IDCG}_k} = \frac{\sum_{n=1}^{|c|} D_k(n) [c_n \in C_u^+]}{\sum_{n=1}^{|c|} D_k(n)} \quad (29)$$

$$D_k(n) = \frac{2^{\text{rel}_n} - 1}{\text{lb}(n + 1)} \quad (30)$$

where rel_n is the relevance (score) for the item at position n .

4.2. Comparison algorithm. To assess the recommendation performance of our model on the dataset, we conducted experimental analysis with different recommendation models, and the results are presented in Table 2.

Table 2. Comparative algorithm performance analysis

Model	MOOCCube				MOOC		
	Recall@5	NDCG@5	Recall@10	NDCG@10	Recall@5	NDCG@5	Recall@10
BPR	0.1218	0.1217	0.1243	0.1235	0.2557	0.2263	0.1218
DMF	0.3568	0.1674	0.3033	0.1824	0.3116	0.2697	0.3568
GC-MC	0.2635	0.1520	0.2788	0.1703	0.3385	0.2720	0.2635
IBCF	0.2294	0.1654	0.2620	0.1904	0.2419	0.1940	0.2294
NAIS	0.3568	0.2324	0.3815	0.2351	0.3615	0.3394	0.3568
NGCF	0.3570	0.3090	0.3596	0.3010	0.3532	0.2154	0.3570
LGCN	0.3620	0.2857	0.3812	0.3017	0.3923	0.3219	0.3620
KCCA-RBM	0.3958	0.3212	0.4125	0.3354	0.4219	0.3591	0.3958

According to the results in Table 2, the performance of our model is optimal in both MOOCCube and MOOC datasets. Compared to the suboptimal method LGCN, our model performs better on MOOCCube and MOOC datasets; Recall@5 increased by 9.2% and 7.5% respectively, NDCG@5 increased by 12.4% and 9.3% respectively. Among them, the BPR model has the worst performance because it does not consider the impact of neighboring nodes on performance and requires large-scale parameter training, which cannot achieve convergence quickly. The performance of DMF and NAIS models is slightly higher than BPR, but these two methods still extract explicit interaction features between users and courses to achieve predictive recommendations, ignoring the deep collaborative information hidden in the interaction sequence between users and courses. Compared with the IBCF method, this method only uses traditional similarity calculation to compare the correlation between courses, and cannot fully extract the deep attribute features of courses. Therefore, its recommendation performance is inferior to DMF and NAIS models. In addition, these models did not model the interaction sequence, position relationship, and time interval information coherently when making MOOC recommendations, resulting in slightly worse recommendation performance than the model proposed in this paper.

GC-MC, NGCF, and LGCN models are all graph convolutional neural network-based models that aggregate predictions by capturing the neighborhood of interacting nodes.

Compared with GC-MC, the model introduced in this paper achieves better on the MOOCCube dataset; Recall@5 and NDCG@5 increased by 33.4% and 52.6% respectively. Meanwhile, the performance of the GC-MC model is significantly lower than that of NGCF and LGCN. Based on this analysis, the KCCA-RBM algorithm proposed in this article offers high recommendation accuracy and strong resistance to overfitting.

5. Conclusion. This paper presents a recommendation system for learning resources algorithm that combines kernel canonical correlation analysis (KCCA) and improved depth constrained Boltzmann machine (RBM), aiming to improve the performance of online learning platform recommendation systems. This article first extracts attributes of users and learning resources through data mining techniques, and constructs a personalized recommendation library. Subsequently, KCCA is used to analyze the deep correlation between users and learning resources, and to select resources with high relevance for recommendation to users. At the same time, the paper improved the traditional RBM model by replacing binary units in the visible layer with Gaussian distribution units to reduce model complexity and enhance its ability to process continuous real valued data. Finally, the scoring data of KCCA and the improved RBM model are fused through linear weighting to enhance accuracy of the recommendation. The experimental results demonstrate that the enhanced algorithm considerably decreases the average absolute error, enhances recommendation accuracy, and to some extent alleviates the problem of data sparsity. Overall, this study provides an effective hybrid algorithm for personalized learning resource recommendation, with good application prospects.

REFERENCES

- [1] H. Ko, S. Lee, Y. Park, and A. Choi, "A survey of recommendation systems: recommendation models, techniques, and application fields," *Electronics*, vol. 11, no. 1, pp. 141, 2022.
- [2] Y. Ma, Y. Peng, and T.-Y. Wu, "Transfer learning model for false positive reduction in lymph node detection via sparse coding and deep learning," *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 2, pp. 2121–2133, 2022.
- [3] L. Zhang, T. Luo, F. Zhang, and Y. Wu, "A recommendation model based on deep neural network," *IEEE Access*, vol. 6, pp. 9454–9463, 2018.
- [4] S.-M. Choi, S.-K. Ko, and Y.-S. Han, "A movie recommendation algorithm based on genre correlations," *Expert Systems with Applications*, vol. 39, no. 9, pp. 8079–8085, 2012.
- [5] N. Zhang, S. Ding, J. Zhang, and Y. Xue, "An overview on restricted Boltzmann machines," *Neurocomputing*, vol. 275, pp. 1186–1199, 2018.
- [6] W. Zheng, X. Zhou, C. Zou, and L. Zhao, "Facial expression recognition using kernel canonical correlation analysis (KCCA)," *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 233–238, 2006.
- [7] S. B. Aher, and L. Lobo, "Combination of machine learning algorithms for recommendation of courses in E-Learning System based on historical data," *Knowledge-Based Systems*, vol. 51, pp. 1–14, 2013.
- [8] M. E. Ibrahim, Y. Yang, D. L. Ndzi, G. Yang, and M. Al-Maliki, "Ontology-based personalized course recommendation framework," *IEEE Access*, vol. 7, pp. 5180–5199, 2018.
- [9] M. E. Ibrahim, Y. Yang, D. L. Ndzi, G. Yang, and M. Al-Maliki, "Ontology-based personalized course recommendation framework," *IEEE Access*, vol. 7, pp. 5180–5199, 2018.
- [10] J. Shu, X. Shen, H. Liu, B. Yi, and Z. Zhang, "A content-based recommendation algorithm for learning resources," *Multimedia Systems*, vol. 24, no. 2, pp. 163–173, 2018.
- [11] X. Chen, and H. Deng, "Research on personalized recommendation methods for online video learning resources," *Applied Sciences*, vol. 11, no. 2, pp. 804, 2021.
- [12] M. Zhong, and R. Ding, "Design of a personalized recommendation system for learning resources based on collaborative filtering," *International Journal of Circuits, Systems and Signal Processing*, vol. 16, no. 1, pp. 122–31, 2022.
- [13] D. R. Hardoon, and J. Shawe-Taylor, "Sparse canonical correlation analysis," *Machine Learning*, vol. 83, pp. 331–353, 2011.

- [14] A. Klami, S. Virtanen, and S. Kaski, "Bayesian Canonical correlation analysis," *Journal of Machine Learning Research*, vol. 14, no. 4, 2013.
- [15] Y. Li, and J. Shawe-Taylor, "Using KCCA for Japanese–English cross-language information retrieval and document classification," *Journal of Intelligent Information Systems*, vol. 27, pp. 117–133, 2006.
- [16] M. Ashad Alam, and K. Fukumizu, "Higher-order regularized kernel canonical correlation analysis," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 29, no. 04, pp. 1551005, 2015.
- [17] M. H. Amin, E. Andriyash, J. Rolfe, B. Kulchytskyy, and R. Melko, "Quantum boltzmann machine," *Physical Review X*, vol. 8, no. 2, pp. 021050, 2018.
- [18] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for Boltzmann machines," *Cognitive Science*, vol. 9, no. 1, pp. 147–169, 1985.
- [19] G. E. Hinton, "Learning multiple layers of representation," *Trends in Cognitive Sciences*, vol. 11, no. 10, pp. 428–434, 2007.
- [20] G. E. Hinton, and R. R. Salakhutdinov, "A better way to pretrain deep boltzmann machines," *Advances in Neural Information Processing Systems*, vol. 25, 2012.
- [21] Y. Chen, "Mineral potential mapping with a restricted Boltzmann machine," *Ore Geology Reviews*, vol. 71, pp. 749–760, 2015.
- [22] X. Peng, X. Gao, and X. Li, "On better training the infinite restricted Boltzmann machines," *Machine Learning*, vol. 107, pp. 943–968, 2018.