International Journal for Housing Science and Its Applications
Publish August 6, 2025. Volume 46, Issue 3 Pages 5168-5178

https://doi.org/10.70517/ijhsa463444

Optimization design of online platform courses in university mathematics teaching reform based on genetic algorithm

Jing Liang[1,*]
[1] Huainan Normal University, Huainan, Anhui, 232001, China
Corresponding authors: (e-mail: ahsdlj@126.com).

Abstract This study proposes an optimization model that integrates adaptive genetic algorithm and probabilistic matrix decomposition. The category features are quantified by WOE coding, and the global search capability of the genetic algorithm is improved by combining adaptive cross-mutation strategy, simulated annealing algorithm and orthogonal table initialization to filter out a subset of highly discriminative features. Further, the student-course nearest neighbor similarity is embedded into the probability matrix decomposition model, and the feature distribution is constrained by the logistic Steele function to optimize the personalized recommendation accuracy. Experiments based on the real MOOC dataset of Academy Online show that the model in this paper achieves optimal performance when the crossover probability Pc is 0.9, with HR and NDCG of 60.58% and 32.99%, respectively. When the variation probability Pm is 0.001, HR is 59.67% and NDCG is 32.78%, which performs close to the optimal value. In the Top-K recommendation, the Precision@5 and mAP@15 increased to 60.44% and 96.32%, respectively, which was significantly better than the XGBoost of the traditional model (34.15% and 59.48%). The results show that the multi-strategy fusion of genetic algorithm and probabilistic matrix decomposition model can effectively solve the course recommendation problem in highly sparse scenarios.

Index Terms genetic algorithm, course optimization design, simulated annealing algorithm, probabilistic matrix decomposition

I. Introduction

Since the “13th Five-Year Plan”, the process of modern education informatization has been continuously promoted. Various universities and colleges of higher education mostly adopt network teaching platform to support students' online course learning, and the use of modern information technology to assist classroom teaching has become one of the important means to improve teaching quality [1], [2]. The construction of network teaching platform is an important part of course construction and reform. Through the application of network teaching platform, teaching resources can be balanced, teaching space can be expanded, teaching methods can be enriched, and it becomes an effective supplement to traditional teaching methods and approaches [3]-[5]. With the support of network teaching platform, teachers can on conveniently provide teaching materials, and communicate and interact with teachers and students through teaching mailboxes and course forums [6], [7].

Meanwhile, students, on the other hand, can view course information, submit assignments, ask and answer questions, participate in course surveys and online tests [8], [9]. It is easy to see that the online teaching platform provides a new teaching and learning environment for teachers and students, actively promotes the reform of education and teaching, and improves the teaching quality of the “Optimal Design” course [10], [11]. However, the course content design of most online teaching platforms still stays in the use of digital media, repeating traditional course content and simply expanding online teaching resources, which cannot effectively reflect the characteristics of information classroom interactivity, diversity, personalization and resource integration [12]-[15]. College mathematics, as an important basic course in colleges and universities, plays an important role in cultivating students' innovative consciousness and innovative ability [16]. For this reason, the effective use of online teaching platforms to design the content of mathematics courses in line with the characteristics of higher education and thus optimize the effect of online classroom teaching is an important issue that colleges and universities must pay attention to [17], [18].

This study proposes an optimization model integrating adaptive genetic algorithm and probabilistic matrix decomposition, aiming to improve the performance of online course recommendation through multi-strategy improvement. For the feature coding problem, WOE coding is used to effectively quantize the key features in the course selection data and enhance the model's ability to interpret non-numerical features. To address the defects of genetic algorithm in global search and convergence efficiency, adaptive cross-variance probability is introduced, the mutation characteristics of simulated annealing algorithm are fused, and the orthogonal table is used to optimize the initial population generation strategy. Adaptive cross-variance probability adjustment mechanism is proposed, the global search ability of simulated annealing algorithm is fused, and the Lévy flight strategy is introduced to solve the problem of low efficiency of traditional genetic algorithm's late search. On this basis, the orthogonal table is used to generate the initial population, the fitness function that takes into account the number of features and model loss is designed, and the data dictionary is combined to optimize the problem of repeated computation and screen the optimal feature subset. Further combined with the probabilistic matrix decomposition model, the near-neighbor similarity between students and courses is embedded in the matrix decomposition process, the feature vector distribution is constrained by the logistic Steele function, and the model parameters are optimized by combining with the gradient descent method, so as to realize Top-K personalized recommendation.

II. Optimization model construction of online courses based on genetic algorithm and probability matrix decomposition

II. A. Data pre-processing

Since most machine learning algorithms cannot handle category-based features, they need to be converted to numerical features before model training, a process known as feature encoding. Two common feature coding schemes are One-Hot coding and WOE coding. Where One-Hot uses binary coding, each bit corresponds to a level of the feature's value, and the bit is 1 indicating that the feature's value is WOE coding, which is the weight of evidence, is also called a kind of coding of the independent variable, and is defined as in equation (1)

$$WOE_i = \ln\left(\frac{Bad_i}{Bad_T}\right) - \ln\left(\frac{Good_i}{Good_T}\right) \tag{1}$$

where is the value of i a feature, Bad_i is the number of Bad tags when the feature takes the value of i, and Bad_T is the total number of tags for the feature.

For example, in the course selection dataset, the category feature of "grade" includes four values: "freshman", "sophomore", "junior", and "senior", how to quantify them into numbers? The following describes the WOE encoding for the "grade" feature. First, the number of courses selected by students in each grade level was counted from the data set.

According to equation (1), the corresponding WOE coding value of each grade is calculated. The WOE encoding value for the "freshman" is -1.21, the encoding value for the "sophomore" is -0.80, the encoding value for the "junior" is 1.14, and the encoding value for the "senior" is 3.36. Then, the value of "grade" is rewritten into the corresponding WOE code value in the course selection dataset to complete the WOE coding of "grade" characteristics.

II. B. Genetic Algorithm Improvement Strategy

After completing the feature encoding and preprocessing of course selection data, for the problem of poor local search ability and low search efficiency in the later stage of the genetic algorithm, this chapter further improves the genetic algorithm in multiple dimensions, and enhances its global search and local optimization ability through the integration of adaptive strategy and simulated annealing algorithm.

II. B. 1) Introducing Adaptive Strategies

The crossover probability P_c and the mutation probability P_m in genetic algorithms have a great influence on the diversity of the algorithm in the early stage and the convergence in the later stage, generally speaking, in traditional genetic algorithms, it is more reasonable to set the value of P_c at 0.4-0.99 and the value of P_m at 0.0001-0.1. However, no matter what the P_c and P_m values are set to, they will not change from the beginning, which is obviously unreasonable, for example, in the iterative process to produce more adapted individuals should be retained to the next generation as much as possible, which needs to reduce the crossover probability of variation, and less adapted individuals should be increased to make the crossover probability of variation so that they change their inferiority status. Another example is that a high crossover and mutation probability should be needed in the early stage of the algorithm to enrich the diversity of the population, while a low crossover and mutation probability should be needed in the later stage of the algorithm to facilitate the convergence of the algorithm. Therefore, a constant crossover and mutation probability will affect the efficiency of the algorithm.

To address the above problems, adaptive genetic algorithm (AGA) is invoked in this paper for adaptively adjusting P_c values and P_m values according to the fitness of individuals:

$$P_c = \begin{cases} P_{c1} - \dfrac{(P_{c1} - P_{c2})(f_{\max} - f')}{f_{\max} - f_{avg}}, & f' \geq f_{avg} \\ P_{c1}, & f' < f_{avg} \end{cases} \tag{2}$$

$$P_m = \begin{cases} P_{m1} - \dfrac{(P_{m1} - P_{m2})(f_{\max} - f)}{f_{\max} - f_{avg}}, & f \geq f_{avg} \\ P_{m1}, & f < f_{avg} \end{cases} \tag{3}$$

where, f_{max}: maximum fitness value in each generation of the population; f_{avg}: average fitness value in each generation of the population, f': the larger fitness value of the two individuals to be crossed; f: the fitness value of the individual to be mutated, $P_{c1}, P_{c2}, P_{m1}, P_{m2}$ are constant parameters, in this paper, after many experiments, we set $P_{c1} = 0.9$, $P_{c2} = 0.5$, $P_{m1} = 0.1$, $P_{m2} = 0.001$.

II. B. 2) Incorporating and improving the simulated annealing algorithm

Simulated annealing algorithm is a common global optimization algorithm that solves complex optimization problems by simulating the formation of crystal structures during the cooling process of a solid object. The main idea is to avoid the search falling into local optimal solutions through randomness, in the expectation of finding better global optimal solutions. The specific steps are as follows:

Initialization: set the initial state s, the initial temperature T_O, and the cooling rate k and the termination temperature T_{end}.

Generate neighborhood solution: perform a random tour in the neighborhood of the current state s to obtain a new state s'.

Calculate the energy difference: calculate the energy difference $\Delta E = E(s') - E(s)$ for the transfer from the current state s to the neighborhood solution s', where $E(s)$ is the value of the objective function taken in state s.

Metropolis criterion determines whether to accept the neighborhood solution: if $\Delta E < 0$, the neighborhood solution is accepted; otherwise, the neighborhood solution is accepted or not according to the probability $P = e^{-\Delta E / \alpha T}$.

Cooling: The effect of slow "cooling" is achieved by lowering the temperature T, with the rate of cooling k controlling the rate of cooling.

Determining the end condition: When the temperature T is reduced below the termination temperature T_{end}, the iterative process of the algorithm is stopped and the current state s is output as the solution of the optimization problem.

In the simulated annealing algorithm, there are three parameters that are closely related to the effect of the algorithm, i.e., the initial temperature T_0, the cooling rate k and the termination temperature T_{end}. These parameters need to be set and adjusted reasonably according to the specific problem. The flow of the simulated annealing algorithm is shown in Fig. 1.

The algorithm has the ability to get rid of local optimums because of its own glitching property, which has a certain probability of accepting worse solutions even if they are temporarily obtained, and therefore has a strong local search capability. However, the simulated annealing algorithm's limited ability to search each time and its dependence on the quality of the randomly generated initial solution makes the algorithm run inefficiently and evolve slowly. In this paper, the higher quality solution obtained after the genetic algorithm has finished running is used as the initial solution of the simulated annealing algorithm in order to improve the line evolution efficiency of the simulated annealing algorithm.

II. C.Feature selection based on genetic algorithm

The improvement of genetic algorithm lays an efficient search foundation for feature selection. On this basis, this chapter combines orthogonal table initialization with dynamic fitness function to construct a feature selection model, aiming to extract key information from redundant features and reduce the complexity of subsequent model training.

Feature selection can filter redundant features, improve model accuracy, and facilitate model construction and optimization. In the process of feature selection, the three commonly adopted feature search strategies include global search, random search and heuristic search. The update frequency of the course selection dataset in higher education is slow and usually occurs at a fixed time (e.g., at the beginning of each semester). When there are more features but still expect to obtain the optimal feature subset, the random search strategy is usually used. In this paper, we use a genetic algorithm as the feature selection method and use orthogonal tables to generate the initial population for the genetic algorithm.

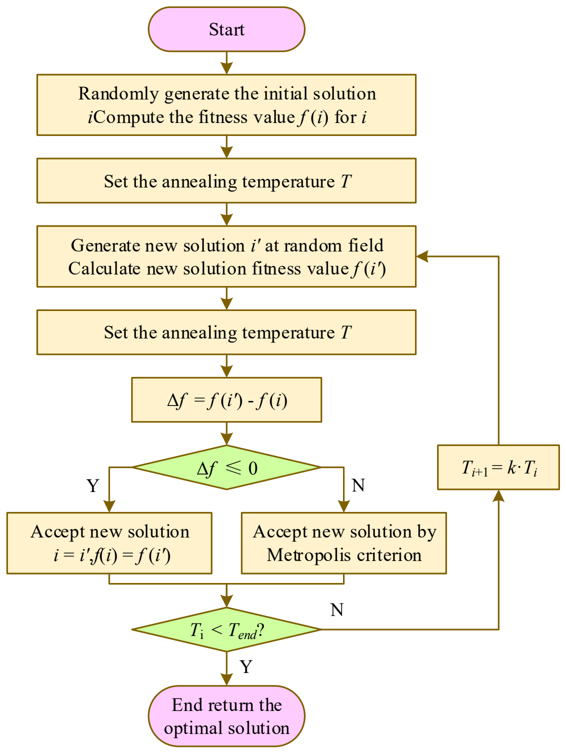
Start
Randomly generate the initial solution
*i*Compute the fitness value *f* (*i*) for *i*
Set the annealing temperature *T*
Generate new solution *i'* at random field
Calculate new solution fitness value *f* (*i'*)
Set the annealing temperature *T*
Δ*f* = *f* (*i'*) - *f* (*i*)
Δ*f* ⩽ 0
Y
N
Accept new solution
i = *i'*,*f*(*i*) = *f* (*i'*)
Accept new solution by Metropolis criterion
T
T_{i+1} = k·T_i
N
T_i < T_end?
Y
End return the optimal solution

Figure 1: Flow diagram of the simulated annealing algorithm

Since before feature selection, we do not know which features are good quality features that are beneficial for recommendation model training. Therefore it is necessary to use orthogonal tables to distribute the initial population uniformly and discretely throughout the feature subset space. In this way, not only the size of the initial population is reduced making the model easy to converge, but also the initial population is more representative.

The key to using genetic algorithms for feature selection is to design a fitness function that balances the number of features against the model loss. We want the feature selection algorithm to search for a subset of features that have good classification ability and a small number of features. Therefore define the fitness function (4)

$$f(x) = \frac{\alpha}{1 + L(x)} + \frac{(1-\alpha)}{1 + F_n} \tag{4}$$

where $L(x)$ is the log loss obtained by the model for a classification model trained on the feature vector x. F_n denotes the number of feature dimensions. $\alpha \in [0,1]$ is a given equilibrium factor, where the fitness function depends on the number of feature dimensions when $\alpha = 0$, and the fitness function depends on the log loss of the classification model when $\alpha = 1$. Thus the balancing factor α regulates the weighting between model loss and the number of features. Since the log-loss value of the classification model is used in the fitness function, this can lead to the computation of fitness during the population iteration to take a long time, thus affecting the convergence time of the genetic algorithm. For this problem this paper proposes the following two optimization measures from the perspective of avoiding repeated calculation of fitness:

(1) In the core process of genetic algorithm, for the new population generated by iteration, we need to calculate the fitness of each genotype in the new population, however, the new population may contain genotypes that have appeared before, and the repeated calculation of the fitness of these genotypes affects the convergence speed of the genetic algorithm. To address this problem, this paper utilizes a data dictionary to record the fitness of each newly appeared genotype, and when calculating the fitness, if the genotype exists in the data dictionary, the fitness value recorded in the data dictionary is returned.

(2) Crossover and mutation operations improve the nonlinear search ability of the genetic algorithm, but crossover and mutation operations may also produce inferior genotypes with very low fitness, and these inferior genotypes will reduce the overall fitness of the population, thus affecting the convergence speed of the genetic algorithm. Therefore, in this paper, the individuals in the data dictionary whose fitness is lower than the minimum fitness of the current population are regarded as the set of eliminated samples, and when the samples produced by crossover and mutation are in the set of eliminated samples, the two samples are re-selected to generate new

offspring.

II. D.Probability matrix decomposition model

The quality subset of features screened by genetic algorithm provides high-quality input to the probabilistic matrix decomposition model. This section further incorporates the student-course nearest-neighbor similarity into the feature vector learning, which is optimized by Bayesian framework with gradient descent to finally achieve accurate course recommendation.

The obtained set of nearest neighbors is fused into the probabilistic matrix decomposition based model, at which time the feature vectors of student-course selection are affected by their nearest neighbors, and the similarity feature vectors of a specific student-course selection are:

$$\tilde{U}_j = \sum_{s \in N_j} P_{sj} U_s \tag{5}$$

$$\tilde{V}_j = \sum_{t \in N_j} Q_{tj} V_t \tag{6}$$

where $\tilde{U}_i$, $\tilde{V}_j$ are the proximate feature vectors of student u_i and course-selection v_j, respectively; N_i and N_j are the sets of neighbors of student u_i and course-selection v_j, respectively.

In this paper, the algorithm not only considers the feature vectors of student-course selection itself in the process of feature vector learning, but also considers the influence of the set of neighbors of student-course selection on the feature vectors of student-course selection, i.e., the feature vector of each student-course selection not only obeys the Gaussian distribution with zero mean, but also has to be similar to that of the nearest neighboring student-course selection.

$$\begin{aligned} & p\left(U \mid P, \sigma_U^2, \sigma_r^2\right) \propto p\left(U \mid \sigma_U^2\right) \times p\left(U \mid P, \sigma_r^2\right) \\ & = \prod_{i=1}^{m} N\left(U_i | 0, \sigma_U^2 I\right) \times \prod_{i=1}^{m} N\left(U_i | \sum_{s \in N_i} P_{si} U_s, \sigma_r^2 I\right) \end{aligned} \tag{7}$$

$$\begin{aligned} & p\left(V \mid Q, \sigma_V^2, \sigma_q^2\right) \propto p\left(V \mid \sigma_V^2\right) \times p\left(V \mid Q, \sigma_q^2\right) \\ & = \prod_{j=1}^{n} N\left(V_j | 0, \sigma_V^2 I\right) \times \prod_{j=1}^{n} N\left(V_j | \sum_{t \in N_j} Q_{tj} V_t, \sigma_q^2 I\right) \end{aligned} \tag{8}$$

where P is the student similarity matrix and Q is the course selection similarity matrix; σ_{U}^2, σ_V^2 are the variances of the U, V distributions, respectively, σ_P^2, σ_Q^2 are the variances of the P, Q distributions respectively, and I is the unit matrix.

From Bayes' theorem, the posterior distribution function of the feature matrices U, V are

$$\begin{aligned} & p\left(U, V \mid R, P, Q, \sigma_R^2, \sigma_P^2, \sigma_Q^2\right) \propto p\left(R \mid U, V, \sigma_R^2\right) \\ & \times p\left(U \mid P, \sigma_U^2, \sigma_r^2\right) \times p\left(V \mid Q, \sigma_V^2, \sigma_q^2\right) \\ & = \prod_{i=1}^{m} \prod_{j=1}^{n} \left[N(r_{ij} \mid g(U_i^T V_j), \sigma_R^2)\right]^{I_{ij}^R} \times \prod_{i=1}^{m} N\left(U_i | 0, \sigma_U^2 I\right) \\ & \times \prod_{j=1}^{n} N\left(V_j | 0, \sigma_V^2 I\right) \times \prod_{i=1}^{m} N\left(U_i | \sum_{s \in N_i} P_{si} U_s, \sigma_r^2 I\right) \\ & \times \prod_{j=1}^{n} N\left(V_j | \sum_{t \in N_j} Q_{tj} V_t, \sigma_q^2 I\right) \end{aligned} \tag{9}$$

where $N\left(r_{ij} \mid g(U_i^T V_j), \sigma_R^2\right)$ denotes a Gaussian distribution with a mean of $g(U_i^T V_j)$ and a variance of σ_R^2 with a Gaussian distribution; I_{ij}^R is a variable that represents the relationship between students and course selection, with a value of 1 when a student chooses a course and 0 otherwise.

On this basis, the values of $U_i^T V_j$ are further normalized by the logistic Steele function $g(x) = 1/(1+\exp(-x))$, which restricts the values of $U_i^T V_j$ to the interval $[0,1]$.

Logarithmic processing of equation (9) yields

$$\begin{aligned}&\ln p\left(U,V|\ R,P,Q,\sigma_R^2,\sigma_P^2,\sigma_Q^2,\sigma_U^2,\sigma_V^2\right)\\&=-\frac{1}{2\sigma_R^2}\sum_{i=1}^{m}\sum_{j=1}^{n}I_{ij}^R(r_{ij}-g(U_i^TV_j))^2-\frac{1}{2\sigma_U^2}\sum_{i=1}^{m}U_i^TU_i\\&-\frac{1}{2\sigma_V^2}\sum_{j=1}^{n}V_j^TV_j-\frac{1}{2\sigma_r^2}\sum_{i=1}^{m}\left(U_i-\sum_{s\in N_i}P_{si}U_s\right)^T\left(U_i-\sum_{s\in N_i}P_{si}U_s\right)\\&--\frac{1}{2\sigma_q^2}\sum_{j=1}^{n}\left(V_j-\sum_{t\in N_j}Q_{tj}V_t\right)^T\left(V_j-\sum_{t\in N_j}Q_{tj}V_t\right)\\&-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}I_{ij}^R\ln\sigma_R^2-\frac{1}{2}(m\times l)\ln\sigma_U^2\\&+(n\times l)\ln\sigma_V^2+(m\times l)\ln\sigma_P^2+(m\times l)\ln\sigma_Q^2+C\end{aligned}\tag{10}$$

Then maximizing this posterior probability is equivalent to minimizing the following objective function, i.e., there are

$$\begin{aligned}E\left(R,P,Q,U,V\right)=&\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}I_{ij}^R\left(r_{ij}-g(U_i^TV_j)\right)^2+\frac{\lambda_U}{2}\sum_{i=1}^{m}U_i^TU_i\\&+\frac{\lambda_V}{2}\sum_{j=1}^{n}V_j^TV_j+\frac{\lambda_P}{2}\sum_{i=1}^{m}\left(U_i-\sum_{s\in N_i}P_{si}U_s\right)^T\left(U_i-\sum_{s\in N_i}P_{si}U_s\right)\\&+\frac{\lambda_Q}{2}\sum_{j=1}^{n}\left(V_j-\sum_{t\in N_j}Q_{tj}V_t\right)^T\left(V_j-\sum_{t\in N_j}Q_{tj}V_t\right)\end{aligned}\tag{11}$$

where $\lambda_U=\sigma_R^2/\sigma_U^2$, $\lambda_V=\sigma_R^2/\sigma_V^2$, $\lambda_r=\sigma_R^2/\sigma_r^2$, $\lambda_Q=\sigma_R^2/\sigma_Q^2$. By gradient descent method, the partial derivatives of Eq. (11) are found with U_i and V_j as parameters, respectively

$$\begin{aligned}\frac{\partial E}{\partial U_i}=&\sum_{j=1}^{n}I_{ij}^Rg'(U_i^TV_j)(g(U_i^TV_j)-r_{ij})V_j+\lambda_UU_i\\&+\lambda_P\left(U_i-\sum_{s\in N_i}P_{si}U_s\right)-\lambda_P\sum_{s\in N_i}P_{is}\left(U_s-\sum_{w\in N_s}P_{ws}U_w\right)\end{aligned}\tag{12}$$

$$\begin{aligned}\frac{\partial E}{\partial V_j}=&\sum_{i=1}^{m}I_{ij}^Rg'(U_i^TV_j)(g(U_i^TV_j)-r_{ij})U_i+\lambda_VV_j\\&+\lambda_Q\left(V_j-\sum_{t\in N_j}Q_{tj}V_t\right)-\lambda_Q\sum_{t\in N_j}Q_{jt}\left(V_t-\sum_{y\in N_t}Q_{yt}V_y\right)\end{aligned}\tag{13}$$

In equation C, $g'(x)=\exp(x)/(1+\exp(x))^2$ is the derivative of the logistic Stee function. At each iteration, U_i, V_j are adjusted as follows

$$U_i\leftarrow U_i-\gamma\cdot\frac{\partial E}{\partial U_i}\tag{14}$$

$$V_j\leftarrow V_j-\gamma\cdot\frac{\partial E}{\partial V_j}\tag{15}$$

where γ is the predefined step size. Repeat the above training process, after each iteration, calculate and verify the average absolute error, and terminate the iterative process after 10 iterations. Based on the obtained U_i, V_j can predict the unknown rating value of the students on the selected courses, and then get an alternative course sorting, and then personalized recommendation through the Top-K recommendation list.

The algorithm first reads in the student's rating information of the alternative courses, that is, constructs the student-course selection scoring matrix, and then uses the cosine similarity measurement method to calculate the nearest neighbor sets $\overline{u}$ and $\overline{v}$ of each student (course selection) in turn, and applies the nearest neighbor set to the probability matrix factorization model, and obtains the student's feature vector U and the feature vector v of course selection through the probability matrix factorization model. Finally, the scoring matrix $\overline{R}$ is reconstructed according to the feature vector prediction, and the calculated prediction score values are sorted, and then the Top-K recommendation results are given according to the constraints.

III. Experimental validation and comparative analysis of course recommendation based on genetic algorithm

Chapter 2 constructs an optimized recommendation framework for online courses by improving the genetic algorithm and probability matrix decomposition model. To further validate the actual efficacy of the model, this chapter conducts experiments based on real MOOC datasets in two dimensions, namely, parameter sensitivity analysis and multi-model comparison, to comprehensively evaluate the performance of the recommender system.

III. A. Sensitivity analysis of genetic algorithm parameter optimization on recommendation performance

In order to verify the model performance of the genetic algorithm under different crossover probabilities P_c and variance probabilities P_m, this section compares the performance under different parameter settings using a common evaluation metric on the real MOOC dataset of Academy Online.

III. A. 1) Experimental data set

In order to quantify the recommendation effect of the model under the data-driven model, this paper selects a real MOOC dataset from Xueyuan Online for validation. This MOOC dataset records 517,243 <learner ID, course ID> valid course enrollment learning behaviors generated by 102,534 learners for 1,288 courses on the Xue Tang Online platform from 2020-2024. The data selected therein ensures that each learner registers for at least 3 courses, but the longest historical registration behavior record of extremely active learners is as high as 351 courses, and the specific distribution of learning behavior data is shown in Figure 2.

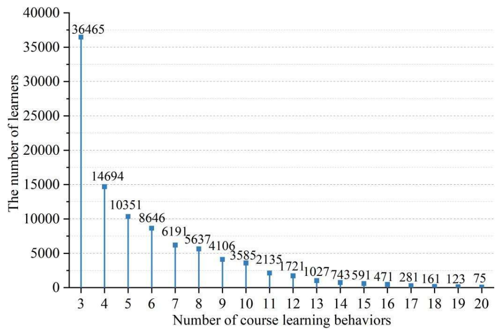

Figure 2: Learning behavior data distribution

As can be seen from Figure 2, the number of learners registering for learning behaviors is the largest number of 3 courses, 36,465 people, with the increase in the number of elective courses the number of people gradually decreases, and basically maintains the registration within 10 courses belongs to the normal range. According to the data distribution shown in Figure 2, through the data sparsity formula

$$Sparsity = 1 - \frac{Interactions}{Users \times Courses} \tag{16}$$

The sparsity of the MOOC data is derived to be as high as 99.61%. Considering the above and combining with the problem of learning time period, the experiment adopts the time node division method, which divides the data between 2020 and 2023 as the training set, and the data in 2024 as the test set to ensure better training model.

III. A. 2) Experimental parameterization

The experiments use Python3 and TensorFlow framework to realize the model construction and run the experiments under Linux system. The setting of parameters is very important in deep learning, in which the learning rate is set to take values from 0.5-0.0001 for experiments; the embedding dimension is set to take values from [64,32,16,8] for experiments in order; the number of Epcho is set to 15; the discard rate is set to 0.75; the Batch_size is equal to 128; and γ is equal to 0.2, and other appropriate and relevant parameters are used to train the model to make the model to obtain the relative best performance.

III. A. 3) Analysis of experimental results

In the model evaluation method, the widely used evaluation methods in Top-K recommendation are used. One is the hit rate HR based on recall and the other is the normalized discount gain NDCG based on location prediction.

Figures 3 and 4 show the effects of different crossover probabilities P_c and variance probabilities P_m on HR and NDCG, respectively

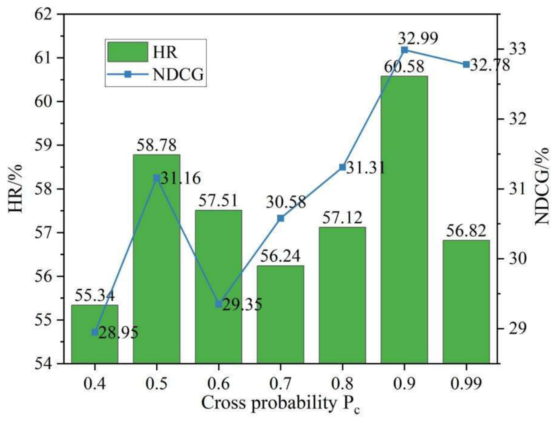

Figure 3: The influence of different crossover probabilities on HR and NDCG

The experimental results show that the model recommendation performance shows nonlinear changes with the adjustment of crossover probability. When the crossover probability is raised from 0.4 to 0.9, the hit rate HR gradually rises from 55.34% to 60.58%, and the normalized discount gain NDCG rises from 28.95% to 32.99%, indicating that higher crossover probability helps to improve the global search capability. However, when the crossover probability reaches 0.99, HR and NDCG decrease to 56.82% and 32.78%, respectively, indicating that too high crossover probability may lead to uncontrolled population diversity and affect convergence efficiency. Overall, the model performance was optimal when the crossover probability was 0.9, which verified the necessity of the adaptive strategy.

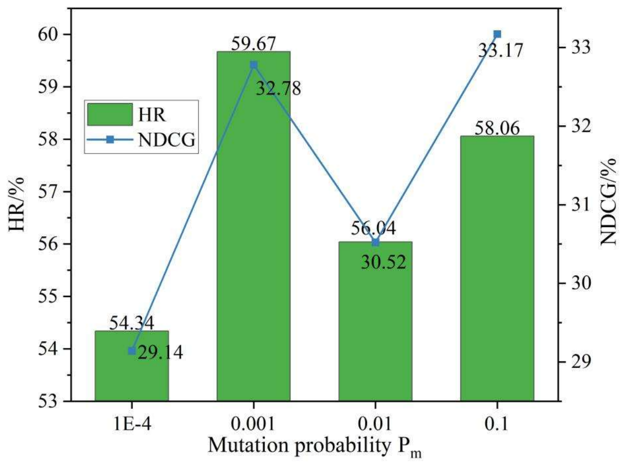

Figure 4: The influence of different mutation probabilities on HR and NDCG

The adjustment of the variance probability has a significant impact on the model performance. When the variance probability was raised from 0.0001 to 0.1, both HR and NDCG showed a fluctuating upward trend, with NDCG reaching the highest value of 33.17% when the variance probability was 0.1. It is worth noting that when the mutation probability is 0.001, HR of 59.67% and NDCG of 32.78% perform close to the optimal value, but too high mutation probability (0.1) may lead to some high-quality individuals being destroyed, resulting in a slight decrease of HR to 58.06%. Experiments showed that the model performance was optimal when the variance probability was 0.001.

III. B. Comparison of Multiple Recommendation Models and Comprehensive Assessment of Top-K Evaluation Indicators

In the previous section, the optimization effect of genetic algorithm on recommendation performance was verified by adjusting the crossover probability and variance parameters. On this basis, in order to further highlight the comprehensive advantages of this paper's model, this section will continue to conduct relevant experiments on the MOOC dataset, and will compare the traditional machine learning model with the fusion model proposed in this paper, and systematically analyze the applicability of the different methods in course recommendation scenarios by combining the Top-K evaluation indexes (Precision@K, mAP@K).

III. B. 1) Methods of comparison

Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM), Extreme Gradient Boosting (XGB) and Plain Bayes (NB) models are used for course recommendation in the course recommendation module respectively.

III. B. 2) Evaluation indicators

This study uses two evaluation metrics commonly used in the course recommendation domain to assess the recommendation effectiveness of each method. The first one is Precision@K, i.e., the precision of the first K courses recommended, which measures the proportion of recommended courses that the user has actually taken, and the larger its value means the better the recommendation effect. The second one is mAP@K, i.e., the average of the mean accuracy of the first K courses recommended, which takes into account both the accuracy and the sorting position of the recommended courses in the list, and the larger its value means the better the recommendation effect.The formula of Precision@K is shown in Equation (17) and Equation (18).

$$\Pr ecision@K = \frac{\sum_{u\in U} \Pr ecision_u @K}{|U|} \tag{17}$$

$$\Pr ecision_u @K = \frac{TP@K}{TP@K + FP@K} \tag{18}$$

where $|U|$ is the number of all users, Precisionu@K represents the precision of recommending the first K courses to user u, TP@K is the number of samples of the first K courses recommended to user u that have both actual and predicted positive cases, and FP@K is the number of samples that have actual negative cases but predicted positive cases.

The definition of mAP@K is shown in equations (19) and (20).

$$mAP = \frac{\sum_{u\in U} AP_u @K}{|U|} \tag{19}$$

$$AP_u @K = \frac{1}{\min(n_u, K)} \sum_{p=1}^{K} Hit_{u,p} \times \Pr ecision_u @p \tag{20}$$

where APu @K is the average precision of each user u, nu denotes the total number of courses that user u has actually taken in the test set, and K is the size of the recommendation sequence. Hitu,p∈{0,1} is a binary variable indicating whether the true label of the recommended course at position 1 in the user's recommendation list is 1. If the true label is 1, Hitu,p1=1, and vice versa. Precisionu@p while denotes the precision of the recommended course before the position (contains) in the user's recommendation list.

III. B. 3) Comparison of the effectiveness of different recommendation models

This study further compares the effects of different recommendation models on the recommendation effect of online courses, and the recommendation results of each model are shown in Figures 5 and 6.

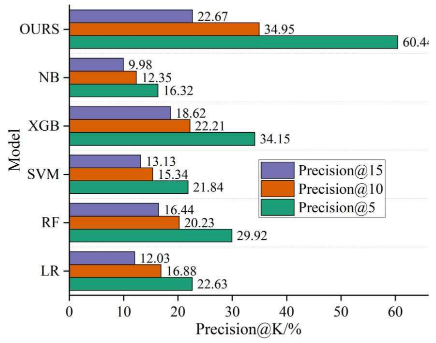

Figure 5: Precision@K metric performance of different recommendation models

In the Precision@K metrics comparison, the model proposed in this paper achieves 60.44%, 34.95%, and 22.67% on Precision@5, @10, and @15, respectively, which is significantly better than the traditional models, such as 34.15%, 22.21%, and 18.62% for XGB. Although Random Forest RF and XGBoost perform better, the accuracy decreases significantly in high K-value scenarios. Plain Bayes NB performs the worst, indicating its lack of adaptability to high-dimensional sparse data. The model in this paper effectively improves the recommendation accuracy by fusing genetic algorithm and probability matrix decomposition.

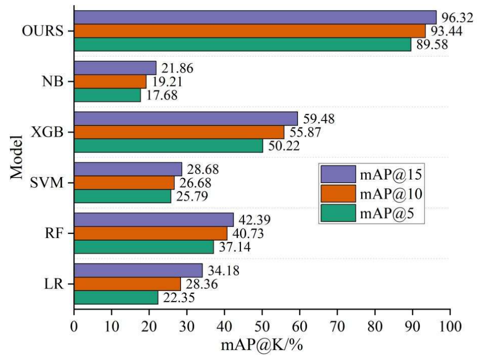

Figure 6: mAP@K metric performance of different recommendation models

In the comparison of mAP@K metrics, this paper's model leads with an absolute advantage of 89.58%, 93.44%, and 96.32% of average accuracy mAP at recommended courses of 5, 10, and 15, respectively. Although XGBoost reaches 59.48% in mAP@15, it is still far lower than the model in this paper, indicating that the traditional method has limitations when considering the sorting position. Logistic regression LR and SVM perform weakly due to the lack of nonlinear modeling capability. The experiments verify the superiority of the model in this paper in terms of integrated sorting accuracy.

IV. Conclusion

In this study, an online course recommendation framework for highly sparse data is constructed by improving the genetic algorithm with a probabilistic matrix decomposition model. The experimental validation shows that:

(1) When the adaptive crossover probability is 0.9 and the variation probability is 0.001, the global search and local convergence efficiency can be balanced, and the HR and NDCG can be increased to 60.58% and 33.17%, respectively.

(2) The probabilistic matrix factorization model with near-neighbor similarity performs well in Top-K recommendation, with Precision@5 and mAP@15 improving by 26.29% and 36.84% respectively compared with XGBoost, which verifies the robustness of the model under sparse data.

Funding

This work was supported by the Anhui Provincial quality Engineering Project of higher education in 2023(No:2023jcjx021).

References

[1] Zhang, J., & Zhou, J. (2023). An introduction to education Informatization for educational equity: an example of online education platform. Journal of Education and Educational Research, 3(2), 114-118.
[2] Dong, X., & Chen, X. (2021). Research on online teaching of college teachers under the background of education informatization. In MATEC Web of Conferences (Vol. 336, p. 05005). EDP Sciences.
[3] Liu, H. (2024). The Network Teaching Platform Based on Information Technology. Procedia Computer Science, 243, 1214-1221.
[4] Wu, J. (2021, September). Design of Distance Network Teaching Platform Based on Information Technology. In International Conference on Cognitive based Information Processing and Applications (CIPA 2021) Volume 1 (pp. 994-1002). Singapore: Springer Singapore.
[5] Huang, R. (2018). Development of a Cloud-based Network Teaching Platform. Int. J. Emerg. Technol. Learn., 13(4), 176-186.
[6] Zhang, Y. (2018). Design and development of WEB-based remote network physical education teaching platform in colleges and universities. International Journal of Emerging Technologies in Learning (Online), 13(4), 150.
[7] Zhu, X. (2023). Study on regional digital teaching resource sharing platform based on internet of things and big data. International Journal of Industrial and Systems Engineering, 44(4), 458-474.
[8] Zhao, J., Mao, H., Mao, P., Hao, J., & Mao, M. (2024). Research on Network Teaching Collaboration Platform Using Flipped Classroom Teaching Mode. Journal of Information & Knowledge Management, 23(02), 2450015.
[9] Lu, J., & Churchill, D. (2014). The effect of social interaction on learning engagement in a social networking environment. Interactive learning environments, 22(4), 401-417.

[10] Wang, Y., & Zhang, N. (2021). The optimization of classroom teaching in colleges and universities based on network topology. Scientific Programming, 2021(1), 1271438.

[11] Ma, Y., Wang, S., Sun, W., Yu, Y., & Bian, J. (2021). Research on the construction and optimization of distributed Moodle course platform. Computer Applications in Engineering Education, 29(2), 474-480.

[12] Wang, J. (2025). Optimizing Learning Path Design in Mobile Learning Platforms for Online Courses. International Journal of Interactive Mobile Technologies, 19(1).

[13] Jing, L., Bo, Z., Tian, Q., Xu, W., & Shi, J. (2020). Network education platform in flipped classroom based on improved cloud computing and support vector machine. Journal of Intelligent & Fuzzy Systems, 39(2), 1793-1803.

[14] Zheng, L., Wang, C., Chen, X., Song, Y., Meng, Z., & Zhang, R. (2023). Evolutionary machine learning builds smart education big data platform: Data-driven higher education. Applied Soft Computing, 136, 110114.

[15] Torres, J. A. G., Lau, S. H., Anchuri, P., Stevens, J. M., Tabora, J. E., Li, J., ... & Doyle, A. G. (2022). A multi-objective active learning platform and web app for reaction optimization. Journal of the American Chemical Society, 144(43), 19999-20007.

[16] Gao, Y., Yuan, R., & Li, J. (2022). Towards Data-Driven Teaching Strategies to Develop Mathematical Thinking. Enthusiastic: International Journal of Applied Statistics and Data Science, 68-81.

[17] Liang, J. (2025). Web-based optimization algorithm for web-based platform course design in teaching reform of university mathematics. J. COMBIN. MATH. COMBIN. COMPUT, 127, 5639-5654.

[18] Lv, Z. (2021). The Design of Mathematics Teaching Optimization Scheme Based on Data‐Driven Decision‐Making Technology. Scientific Programming, 2021(1), 5377784.