International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18 (2026) pp. 1-19

https://doi.org/10.70917/ijcisim-2026-0149
Article

Research on Student Sports Training Data Modeling and Personalized Training Program Optimization Based on Multi-Layer Perceptron

Menglong Lin [1,*], Wiradee Eakronnarongchai [2], Jakrin Duangkam [2] and Jinchuan Lin [1]

[1] Teacher Education College, Zhangzhou City Vocational College, Zhangzhou, Fujian, 363000, China
[2] Udon Thani Rajabhat University, Muang.Udon Thani 41000, Thailand; wyxylml2007@163.com

Abstract: In this paper, based on the Gray Wolf Optimization (GWO) algorithm, Cauchy variation operator and cosine convergence factor are added, and the convergence speed of the algorithm is enhanced by the position updating formula to shorten the training time, and the improved multilayer perceptron is used for modeling student sports training data. After that, from the perspective of user groups, user-based collaborative filtering algorithm (UB-CF) is selected to model the sportsmen. Then from the perspective of sports, CB recommendation algorithm is used to build recommendation object model based on sports features, and finally UB-CF algorithm and CB algorithm are combined to form a personalized sports recommendation algorithm, so as to achieve the purpose of personalized recommendation to users. The results show that when the overlap rate is 75% and the window size is 3/4 of the original data length, the classification accuracy of the x-y dataset can reach 97.32%. The recommendation effect in the personalized training program of student sports shows that the value of RMSE of this paper's recommendation algorithm (0.1016) is much lower than that of the comparison method, and its predicted value is highly consistent with the actual value. It can be seen that the method in this paper makes the improved recommendation algorithm more perfect and the recommendation accuracy higher by deeply mining the user's preference.

Keywords: Gray wolf optimization algorithm; Cauchy variational operator; Cosine convergence factor; Multilayer perceptron; Sports training; Personalized recommendation

1. Introduction

Sports training digitization refers to the comprehensive collection of information on athletes' physical condition, health status, and training data through modern technological means, and the presentation of this information using various visualization methods. Through scientific analysis of training data [1-3], coaches can gain a better understanding of athletes' training status and develop more precise and reasonable training plans. For example, by collecting underlying visual information from sports training, coaches can assist in posture correction analysis for athletes [4]. By digitally monitoring physical fitness data during basketball training, coaches can customize more suitable training plans for athletes [5]. It is evident that training digitization, by introducing modern data management technologies into traditional sports training, can effectively enhance the competitive level of sports teams [6]. However, upon reviewing the current development of training digitization, several pain points are identified. The first is the inadequacy in managing training data, as the collection of training data characteristics is incomplete, leading to low efficiency in retrieving training data [7]. Second, in the presentation of training data, tables or simple charts are typically used, resulting in a limited presentation format that fails to fully showcase the multidimensional characteristics and relationships of the data [8-10]. Additionally, data reports often use fixed templates, which cannot meet the personalized needs of coaches [11].

Sports training data modeling refers to the process of collecting, processing, analyzing, and applying

various information during sports training using information technology, covering athlete information, training data, training plans, competition information, and other content [12-13]. Qian and Liu applied data mining technology to the modeling and analysis of sports training indicators, and found that compared to traditional manual analysis methods, their comprehensive analysis capability improved by 37.14% [14]. Lei et al. designed a real-time data analysis system for sports training using data mining techniques, aiming to improve teaching methods and student performance by identifying patterns [15]. Knobbe et al. introduced a data analysis method that extracts actionable patterns from historical data, and applying two aggregation techniques to optimize training parameters for professional speed skaters, thereby providing coaches with reference criteria for adjusting training plans and avoiding under-training or over-training [16]. Hu et al. combined the K-nearest neighbor method with the Alphapose network to propose a novel multi-object tracking framework, significantly improving multi-object tracking accuracy (MOTA) and global multi-object tracking accuracy in sports training environments [17]. As sports training data continues to accumulate, conventional statistical analysis techniques may be insufficient for analyzing training data. The emergence of multi-layer perceptrons provides an optimized method for discovering scientific patterns and correlations in large amounts of complex training data [18].

Gray wolf optimization algorithm (GWO) is a newly proposed meta-heuristic algorithm, by analyzing the inadequacy of the gray wolf algorithm for position updating in this predation process, this paper proposes an improved Cauchy-variant gray wolf optimization algorithm, IGWO.The improved algorithm is used as a trainer for multilayer perceptron in classification experiments for classification problems of varying complexity, which is in turn evaluated to assess the optimization performance of the improved gray wolf optimization algorithm and the robustness. After that, from the perspectives of users and sports, based on the sports user model and sports model, the collaborative filtering (CF) algorithm and the content-based recommendation (CB) algorithm are combined to recommend sports programs that meet the user's characteristics to the user and analyze the recommendation effect of the model.

2. Multi-Layer Perceptron-Based Modeling of Student Sports Training Data

2.1. Gray Wolf Optimization Algorithm

In the Gray Wolf Optimization Algorithm [19] (GWO), assuming that the population size of gray wolves is N and the search space is d -dimensional, the position of the i th gray wolf in the space is $X_i = (x_1, x_2, \cdots, x_d)$, and the spatial prey position is the global optimal solution. The gray wolf is commanded and controlled by the alpha wolf $(\alpha\, wolf)$, beta wolf $(\beta\ wolf)$ and sigma wolf $(\delta\, wolf)$ during the predation process and surrounds the prey by encircling the attack method, so the position of the gray wolf encircling the prey is updated as:

$$D = \left| C \cdot X_p(t) - X(t) \right| \tag{1}$$

$$X(t+1) = X_p(t) - A \cdot D \tag{2}$$

where t is the current iteration number, $X_p = (x_1, x_2, \cdots, x_d)$ is the prey position, $A \cdot D$ is the encircling step, and vectors A and C are defined as:

$$A = 2(r_1 - E) \cdot a \tag{3}$$

$$C = 2r_2 \cdot a \tag{4}$$

where r_1 and r_2 are the 1 row d column random vector between the interval $[0,1]$, E is a 1-row d column vector where each element is 1, and a is the convergence factor vector, which decreases linearly from 2 to 0 with the increase of the number of iterations, that is:

$$a = 2\left(1 - t / t_{\max}\right) \cdot E^T \tag{5}$$

From Eqs. (1)-(5), the predation position of other individual gray wolves guided and directed by alpha wolves $(\alpha\, wolf)$, beta wolves $(\beta\ wolf)$, and sigma wolves $(\delta\, wolf)$ during predation is updated as follows:

$$D_\alpha = \left|C_1 \cdot X_\alpha - X\right|, D_\beta = \left|C_2 \cdot X_\beta - X\right|, D_\delta = \left|C_2 \cdot X_\delta - X\right| \tag{6}$$

$$X_1 = X_\alpha - A_1 \cdot D_\alpha, X_2 = X_\beta - A_2 \cdot D_\beta, X_3 = X_\delta - A_3 \cdot D_\delta \tag{7}$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3} \tag{8}$$

2.2. Multilayer Perceptron Based on Improved Gray Wolf Optimization Algorithm

2.2.1. Improved Cauchy Variant Gray Wolf Optimization (IGWO) Algorithm

(1) Cosine convergence function

The convergence factor a affects the global and local search ability of Wolf. Different deceleration rates of $a(t)$ correspond to different search performance of the algorithm. The convergence factor a decreases linearly in the gray wolf optimization algorithm, and the convergence rate changes from slow to fast as the number of iterations increases, which balances the algorithm's global exploration and local exploitation. In training the MLP, this paper cites a cosine-based formula for the convergence factor:

$$a(t) = 2 \times \cos\left(\left(t / t_{\max}\right) * (\pi / 2)\right) \tag{9}$$

where $a(t)$ is the t th generation convergence factor and $t_{\max}$ is the maximum number of iterations. Using the cosine convergence factor balances the global exploration and local exploitation capabilities of the GWO algorithm.

(2) Introducing the Cauchy Variation Operator

The Gray Wolf Optimization algorithm is prone to premature maturity and falling into local optimum, so this paper introduces the Cauchy variation operator in the algorithm [20]. The Cauchy variation is based on the Cauchy probability density function:

$$g\left(x; x_0, \gamma\right) = \frac{1}{\pi\gamma\left[1 + \left(\frac{x - x_0}{\gamma}\right)^2\right]} \tag{10}$$

where x_0 is the location parameter, γ is a random variable greater than 0, and x is a real number. In this paper, we take $x_0 = 0$ and $\gamma = 1$, which is a standard Cauchy distribution. By analyzing its probability density function, it can be seen that it does not have a specific mean and variance, but the multitude and median are equal to the location parameter, x_0, and its distribution function is:

$$G(x) = \frac{1}{2} + \frac{1}{\pi}\arctan(x) \tag{11}$$

The overall distribution of the Cauchy distribution is more homogeneous, the maximum of the symmetry axis is flatter relative to the Gaussian distribution, and the 2-sided curve corresponds to a higher probability of trailing. In this paper, we add the perturbation formula:

$$X_{best}(t) = X_i(t) + X_i(t) \cdot G(x) \tag{12}$$

$$X_i(t+1) = \begin{cases} X_{best}(t), f\left(X_{best}(t)\right) > f\left(X_i(t)\right) \\ X_i(t), f\left(X_{best}(t)\right) \le f\left(X_i(t)\right) \end{cases} \tag{13}$$

where $f(X_i(t))$ denotes the fitness value of i -wolf at the t th iteration. The algorithm is guided to jump out of the local optimum by local perturbations.

(3) Adaptive position update formulation

In this paper, an adaptive adjustment strategy is invoked to increase the positional advantage of the 3 head wolves by using the inverse of the fitness value as the weight coefficient of the update formula. Namely:

$$X_i(t+1)=\begin{cases}\dfrac{\frac{1}{f_\alpha}X_\alpha+\frac{1}{f_\beta}X_\beta+\frac{1}{f_\delta}X_\delta}{\frac{1}{f_\alpha}+\frac{1}{f_\beta}+\frac{1}{f_\delta}}, f\left(X_i(t)\right)\le f_{avg}\\ \dfrac{X_\alpha+X_\beta+X_\delta}{3}, f\left(X_i(t)\right)> f_{avg}\end{cases} \tag{14}$$

Among them, $f(X_i(t))$ represents the fitness value of i wolf in the t generation, and f_α, f_β and f_δ represent the fitness values of α , β and δ in the first t generation, respectively. f_{avg} represents the average fitness of all wolves in the t generation population.

2.2.2. Multilayer Perceptron Based on Kersey Variant Gray Wolf Optimization Algorithm

In this paper, we use an intelligent optimization algorithm to optimize the training of a multilayer perceptron (MLP) by encoding the weights and biases of each node as an input vector V :

$$V=\{W,\theta\}=\left\{w_{1,1},w_{1,2},\cdots,w_{n,n},\theta_1,\theta_2,\cdots,\theta_n\right\} \tag{15}$$

where $w_{i,j}$ denotes the weights connected for node i and node j . θ_j denotes the bias of node j and n is the total number of nodes. To test the effectiveness of the algorithm to train the MLP, the following variables are defined:

Definition 1: Mean Square Error MSE

By feeding the data of the sample into the input layer of the MLP and comparing the output data of the MLP with the target result, if the output data is closer to the target result, the better the training effect is, and vice versa the worse the training effect is. Namely:

$$MSE=\frac{\sum_{i=1}^{PP}(o_i^k-d_i^k)^2}{PP} \tag{16}$$

where o_i^k is the desired output of the output node, d_i^k is the actual output, and PP is the total number of training samples.

Definition 2: Trainer classification accuracy is defined as follows:

$$CA=\left(\sum_{i=1}^{n}T_i\Big/\sum_{i=1}^{n}S_i\right)\times 100\% \tag{17}$$

Where S_i represents the number of samples with the i th attribute in the set of hypotheses S to be classified. T_i represents the number of correctly categorized samples in the dataset.

2.3. Analysis of the Training Effect of Multilayer Perceptron

One metric for assessing the effectiveness of MLP training is the mean square error mse, defined as:

$$mse=\sum_{i=1}^{m}(o_{i,k}-d_{i,k})^2 \tag{18}$$

Where: m is the output, $o_{i,k}$ is the actual output of input i at training sample k , and $d_{i,k}$ is the desired output of input i . Then the average MSE over all training samples is:

$$MSE=\frac{\sum_{k=1}^{N}\sum_{i=1}^{m}(o_{i,k}-d_{i,k})^2}{N} \tag{19}$$

Where: N is the number of training samples.

By inputting the test samples to the MLP input layer, the predicted output values of the MLP are compared with the actual values, and the closer the results are, the better the training effect is. Therefore, the training objective of the MLP is to minimize the objective function MSE, while the decision variables are the connection weights and deviations of the neural network.

The second evaluation metric is the trainer's classification accuracy CA, which can be used to measure the accuracy of the algorithm and is defined as:

$$CA = \sum_{i=1}^{n} T_i \Big/ \sum_{i=1}^{n} H_i \times 100\% \tag{20}$$

Where: T_i is the number of correctly classified samples in the dataset, and H_i is the total number of attributes to be classified in the samples.

The algorithm is iterated 500 times, the population size is set to 25, and other parameters of this paper's algorithm are the same as those of the previous benchmark function optimization test. The test data set is shown in Table 1; the classification accuracy performance is shown in Table 2. It can be seen that this paper's algorithm gets the highest average value of classification accuracy in all 4 of the datasets. At the same time, compared with the other three algorithms, the standard deviation of this paper's algorithm is also the smallest, which indicates the better robustness and stability of the algorithm. This case fully demonstrates that using this paper's algorithm to train MLPs is also effective and feasible, and verifies that the algorithm's optimization performance for optimization problems can be guaranteed.

Table 1. Test data set.

Data set	Data object number	Training sample number	Test sample number	Characteristic number	Classification number
Balance	625	467	157	3	4
Cancer	563	423	144	31	3
Cancer-int	701	520	176	10	2
Credit	688	525	174	14	2
Diabetes	769	580	188	9	1

Table 2. Classification accuracy.

Data set	Statistical index	Training algorithm			
		Icboa algorithm	Laboa algorithm	Iboa algorithm	Our algorithm
Balance	Mean accuracy	71.96	76.71	83.54	95.9
	Standard deviation	11.63	12.39	12.47	9.06
	Optimal value	89.01	90.88	95.54	97.57
Cancer	Mean accuracy	73.93	74.1	80.56	92.55
	Standard deviation	9.51	10.51	8.88	7.94
	Optimal value	76.19	80.24	87.79	95.05
Cancer-int	Mean accuracy	74.92	75.26	79.71	82.78
	Standard deviation	15.12	15.13	13.13	10.09
	Optimal value	78.86	81.46	87.01	91.29
Credit	Mean accuracy	86.6	89.32	92.37	92.03

	Standard deviation	12.42	11.69	11.49	9.64
	Optimal value	88.85	93.37	94.61	95.11
Diabetes	Mean accuracy	85.07	89.12	90.82	91.63
	Standard deviation	14.47	13.62	10.19	8.65
	Optimal value	88.01	92.16	93.18	97.07
Average training hours/s		315.207	275.172	282.404	223.086

3. Personalized Recommendation Model for Student Sports Based on MLP and Application Effect

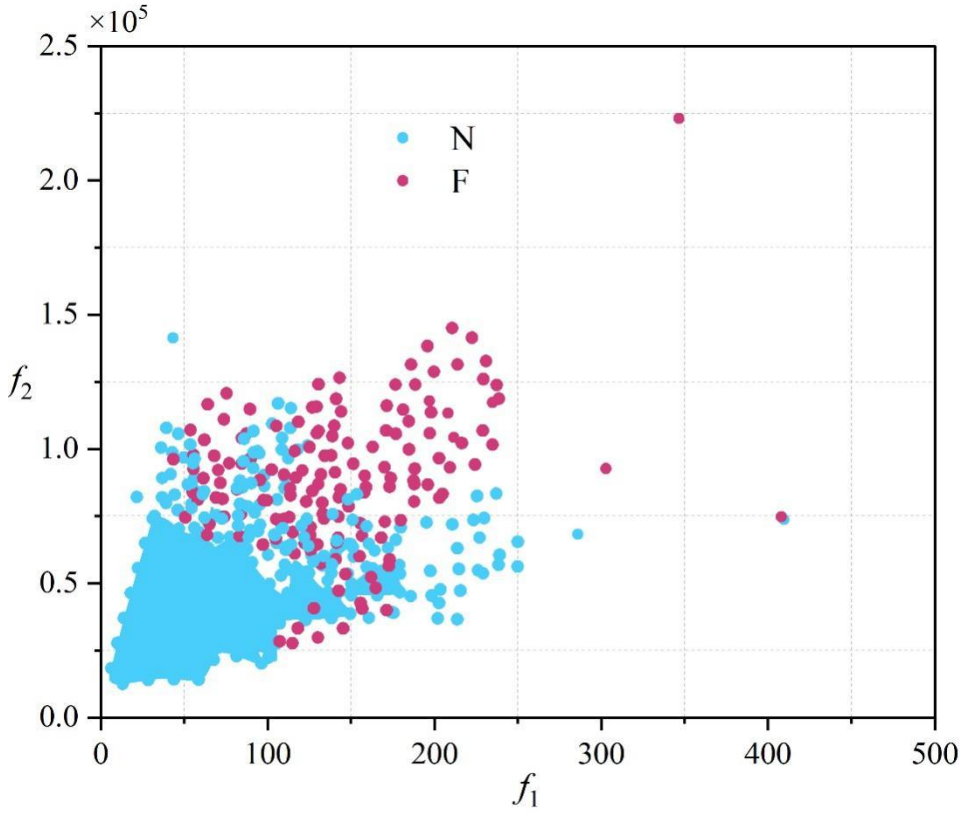
3.1. MLP-Based Human Physiological Signal Classification Study

The physiological signals of the physically trained human body are abbreviated as F, and the physiological signals of the physically untrained human body are abbreviated as N. There are 3680 samples in each category, and in each sample there are two columns of data (denoted as x and y) that are set in the same region of the two neighboring channels, and x-y are new datasets consisting of the two columns of data.

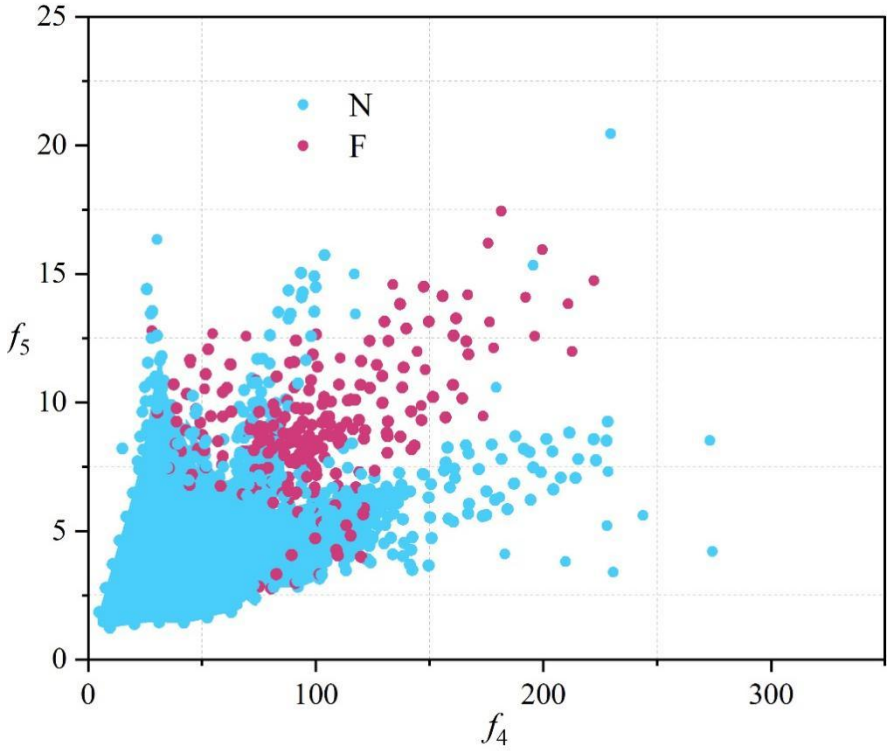
For the brainwave dataset during students' sports training, the time domain features $f_1, f_2, f_3, f_4, f_5, f_6$, and f_7 are taken, and the EMD features $f_{11}, f_{12}, f_{13}, f_{14}, f_{15}, f_{16}, f_{17}, f_{18}$, GAFDS features $f_{19}, f_{20}, f_{21}, f_{22}, f_{23}, f_{24}$.

Two sets of features were extracted from each of the three feature extraction methods and combined into different 2D feature spaces. The distribution of data in the feature space of the three methods is shown in Fig. 1-Fig. 3. In Fig. 1, (a) and (b) represent $\{f_1, f_2\}$ and $\{f_4, f_5\}$, respectively; in Fig. 2, (a) and (b) represent $\{f_{11}, f_{12}\}$ and $\{f_{13}, f_{14}\}$, respectively; in Fig. 3, (a) and (b) represent $\{f_{19}, f_{20}\}$ and $\{f_{21}, f_{22}\}$.

In the $\{f_1, f_2\}$ and $\{f_4, f_5\}$ feature spaces, it can be found that the distributions of the two types of data are somewhat different, but there is also a large amount of overlap. In the $\{f_{11}, f_{12}\}$ and $\{f_{13}, f_{14}\}$ feature spaces, there is likewise some overlap. However, there are fewer differences between the two types of data relative to the $\{f_1, f_2\}$ feature space. In the $\{f_{19}, f_{20}\}$ and $\{f_{21}, f_{22}\}$ feature spaces, there is also more overlap between the two types of data, which makes it harder to differentiate. Therefore, from the above analysis, it is difficult to distinguish these two types of data by simply using the two-dimensional feature space.

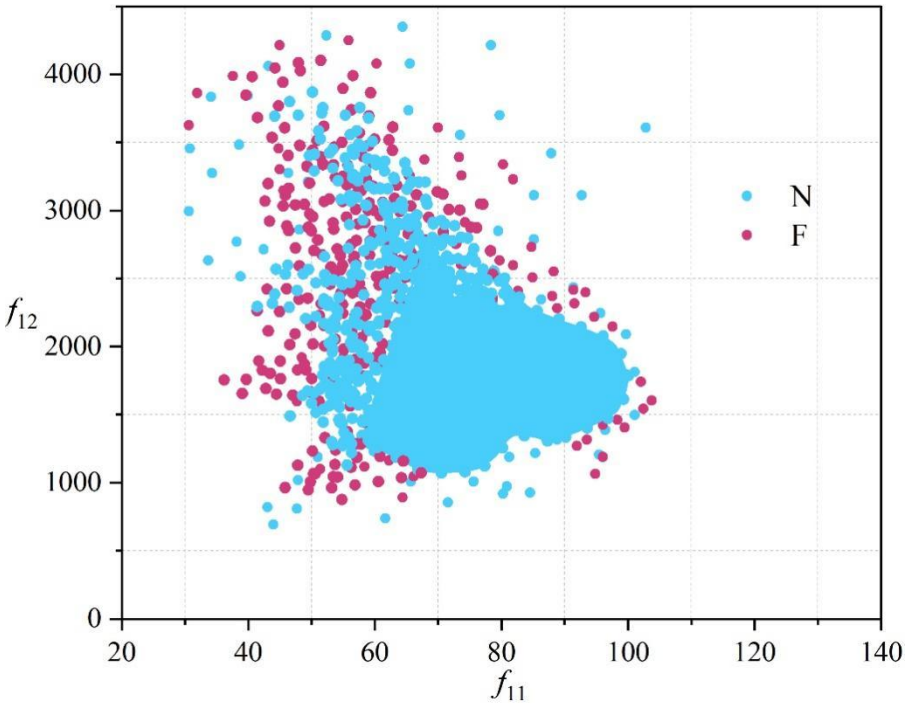


(a) $\{f_1, f_2\}$

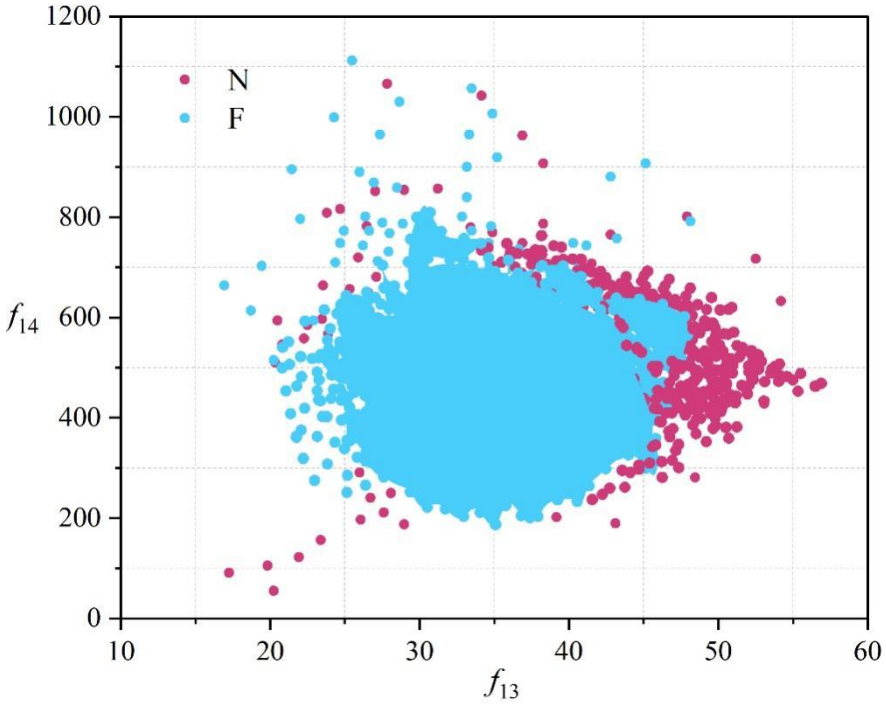


(b) $\{f_4, f_5\}$

Figure 1. Data distribution in $\{f_1, f_2\}$ and $\{f_4, f_5\}$ feature space.

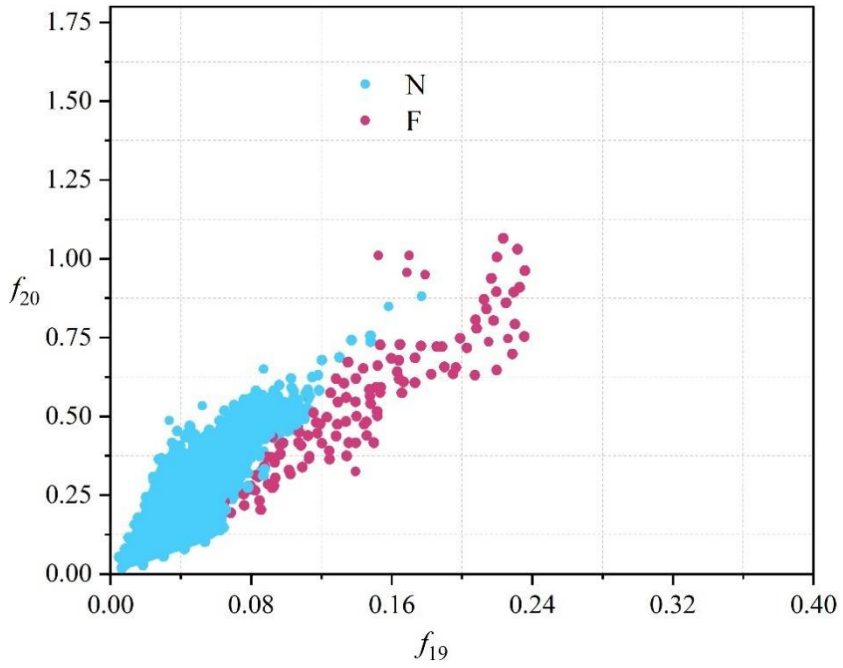


(a) $\{f_{11}, f_{12}\}$

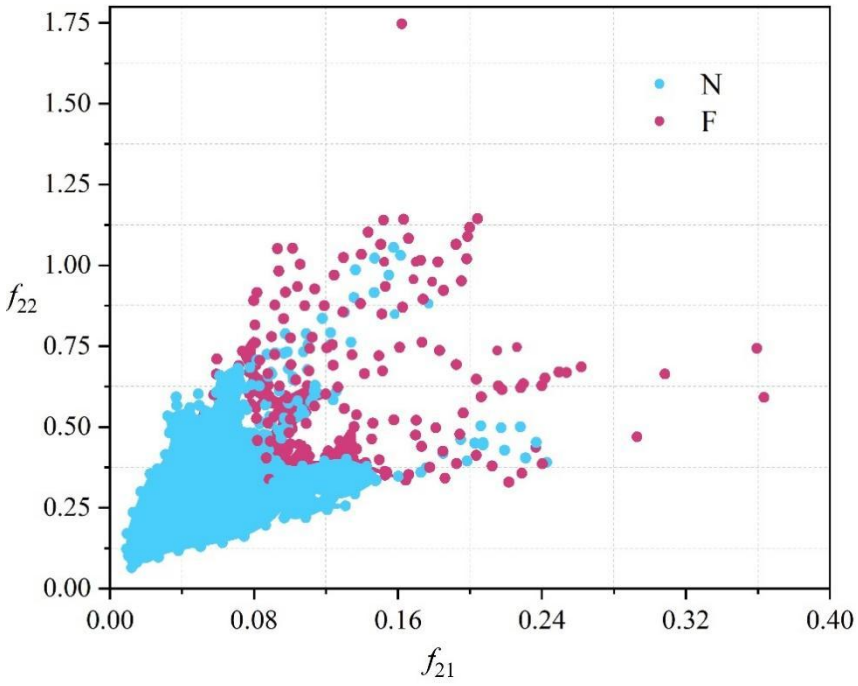


(b) $\{f_{13}, f_{14}\}$

Figure 2. Data distribution in $\{f_{11}, f_{12}\}$ and $\{f_{13}, f_{14}\}$ feature space.



(a) $\{f_{19}, f_{20}\}$



(b) $\{f_{21}, f_{22}\}$

Figure 3. Data distribution in $\{f_{19}, f_{20}\}$ and $\{f_{21}, f_{22}\}$ feature space.

Contour coefficients combine cohesion and separation for clustering and are often used to assess the effectiveness of clustering. Here the contour coefficient values are used to assess the distribution of the sample in the feature space and the validity of the features. There are only two categories in the sample i.e. two clusters, denoted as x^1 & x^2. The $x_i^1 (i = 1, \cdots, S^1)$ and $x_j^2 (j = 1, \cdots, S^2)$ denote the elements in the two clusters, where S^1 and S^2 denote the number of samples in the two categories respectively. The contour coefficients are computed as follows:

$$S(x_i^1)=\frac{avgDis\tan ce2(x_i^1)-avgDis\tan ce1(x_i^1)}{\max(avgDis\tan ce1(x_i^1),avgDis\tan ce2(x_i^1))} \tag{21}$$

Ditto for the contour coefficients of the elements in the cluster x^2. The contour coefficients of all points are averaged to give the total contour coefficient S.

The values of contour coefficients in different feature spaces are shown in Table 3. As the value of contour coefficient is between ±1, the closer to 1, the better the cohesion in the category and the better the separation between categories. The table shows that the contour coefficient is greater than 0 in three different feature spaces, indicating that the average distance between each element and the element of the cluster it belongs to is greater than the average distance between the element of the cluster it is closest to, which indicates that the two classes of samples can be better separated in the high-dimensional feature space.

Table 3. The value of the contour coefficient in different feature space.

Eigenspace	$f_1,f_2,f_3,$ f_4,f_5,f_6,f_7	$f_{11},f_{12},f_{13},f_{14},$ $f_{15},f_{16},f_{17},f_{18}$	$f_{19},f_{20},f_{21},$ f_{22},f_{23},f_{24}
S	0.0254	0.0576	0.01638

In this paper, we have cross-validated the datasets of x, y and $(x-y)$ with 2, 5, and 10 discounts, respectively. Similarly. The classification accuracies after the three types of feature extraction methods and optimized feature selection are shown in Table 4, where x, y and $(x-y)$ represent different experimental datasets. From the results, it can be seen that the resulting classification accuracy of the feature extraction method in this paper is higher compared to the other two feature extraction methods. From the final results, the feature set $\{f_1,f_2,f_4,f_6,f_7,f_{11},f_{12},f_{17},f_{19},f_{20},f_{21},f_{22},f_{23},f_{24},f_{24}\}$ the classification accuracy can reach 93.07%, which indicates that the feature optimization can further improve the classification accuracy.

Table 4. The classification accuracy of the selected characteristics is optimized.

Cross validation	MLP classifier		
	x(%)	y(%)	x-y(%)
2-fold	83.38	81.07	91.58
5-fold	84.05	82.67	92.66
10-fold	84.15	83.96	93.07

The results of accuracy comparison under different classification methods are shown in Table 5. In the case of targeting the same dataset, the accuracy of time domain feature extraction using the traditional classifier approach ranges from 77.46% to 81.24%, while the MLP classifier used in this paper achieves an accuracy of 93.07% for the time domain feature extraction, which shows that it can get better classification results.

Table 5. The accuracy of different classification methods is compared.

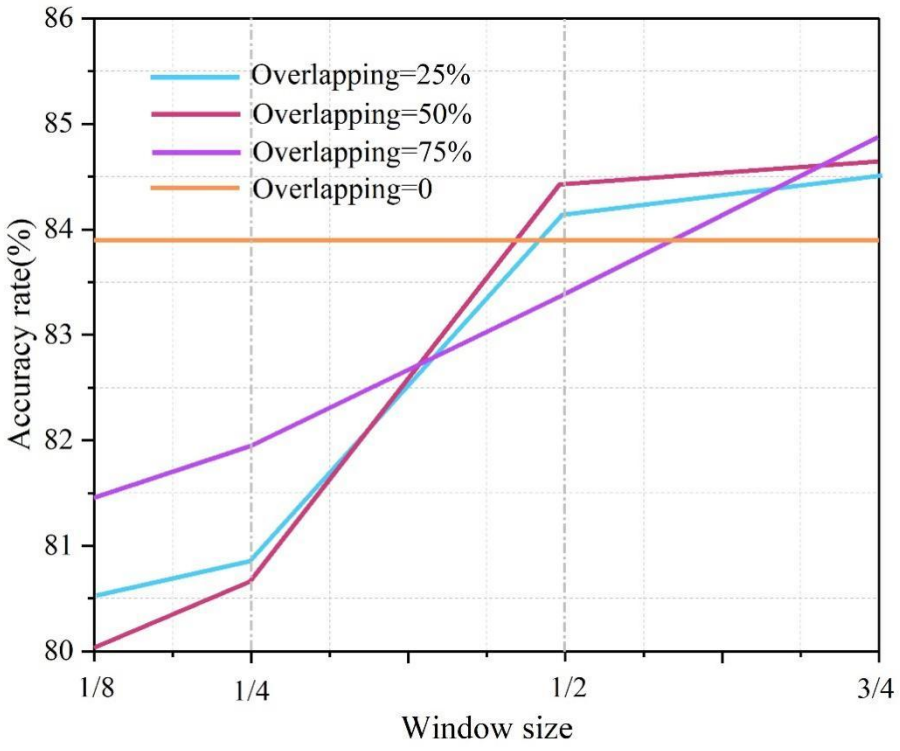
Classification method	Accuracy rate(%)
SVM algorithm based on RBF kernel function	77.46
Least squares support vector machine	79.03
KNN	81.24
MLP	93.07

In this paper, based on the original experiments, the original data set is preprocessed by window

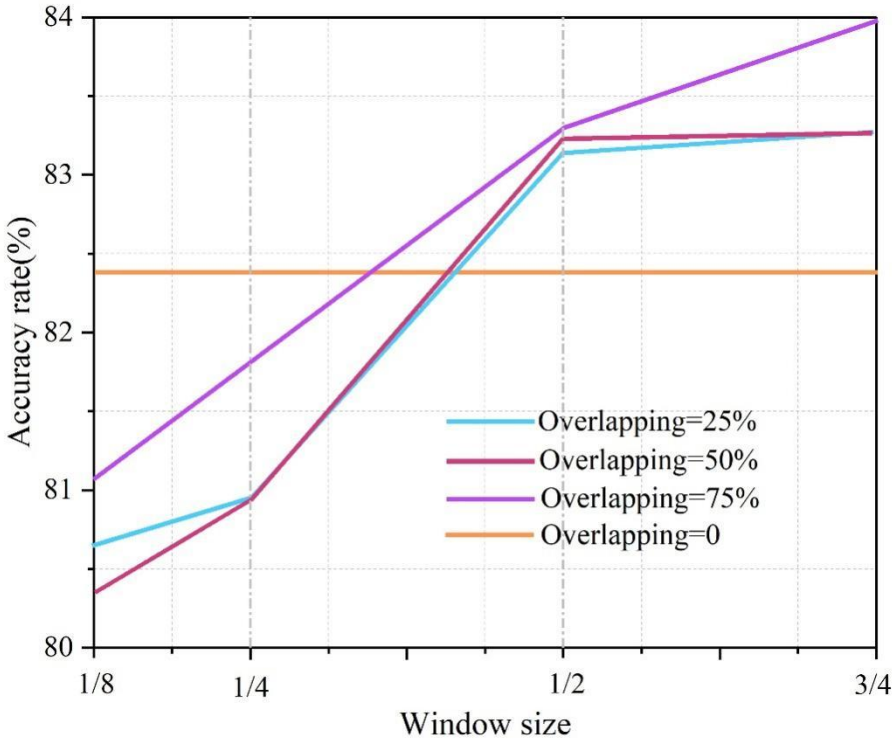
cutting. The sampling period of the original dataset is 20s, and the data length of each sample is 10250, and now a sliding window with the sizes of 1/8, 1/4, and 3/4 of the original data length is used to cut the original dataset, and in order to reduce the influence caused by the changes between the window and the window, the overlap is taken between the adjacent windows, and the overlap rate is taken to be 25%, 50%, and 75%, respectively. The final experimental results are obtained after using three types of feature extraction methods: time domain, GAFDS and EMD and optimizing the selection of features.

The experimental accuracies of signals x, y, and x-y with different overlap rates and different window sizes are shown in Fig. 4, where (a) to (c) represent x, y, and x-y, respectively.It can be found from the figure that, under a certain overlap rate, the size of the sliding window has an effect on the accuracy of the experimental results. In the case where the window size is 1/2 and 3/4 of the length of the original data, the accuracy of the classification results based on the three groups of data x, y and x-y are more satisfactory. For example, when the overlap rate is 75% and the window size is 3/4 of the length of the original data, the classification accuracy of the x-y group of data can reach 97.32%, which is a big improvement over the highest accuracy of 93.07% before uncut. Meanwhile, the experimental accuracies of all three data sets are lower than those before uncut when the window size is 1/4 and 1/8 of the original data length. This means that the original data was cut too short and the signal volume per sample was reduced, resulting in lower accuracy.

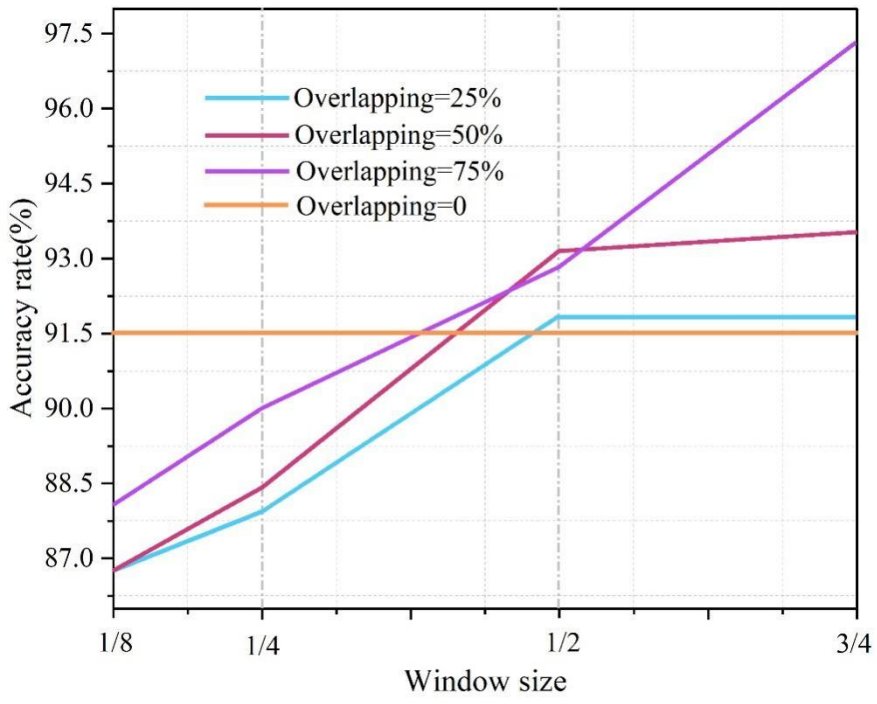
Considering that the amount of data per sample is on the large side, it is already enough to characterize students' physical training, thus the data can be cut to increase the number of samples. The increase in the number of samples helps to avoid model overfitting and can improve the classification accuracy of the model. A larger overlap rate means that the sample size after cutting is also larger, which can better improve the classification accuracy of the model.



(a)x



(b)y



(c)x-y

Figure 4. Accuracy of different overlapping rates and different window sizes.

3.2. Personalized Sports Recommendation System

3.2.1. Sport user modeling

The sports user model includes a model of the user's basic physical characteristics and a model of the user's rating matrix for the program. The basic user characteristics model includes the user's basic

attribute information and interests. Here the user keywords are extracted, each keyword is a basic information of the user, the basic information of the user is age, gender, BMI, sports category of interest, etc., the user model is:

$$\overrightarrow{U_u} = \left\{\left(k_u^1, w_u^1\right), \left(k_u^2, w_u^2\right), \left(k_u^3, w_u^3\right), \cdots, \left(k_u^n, w_u^n\right)\right\} \tag{22}$$

Where $\overrightarrow{U_u}$ denotes the feature vector of user u, k_u^n denotes the n th keyword of user u, and w_u^n denotes the weight of the n th keyword of user u.

3.2.2. Sport modeling

In this paper, from the perspective of sports, according to the role of sports can be in the human body can be divided into the upper limbs, trunk and lower limbs. If the sports have both upper extremity and lower extremity sports, the keyword of the object is set to 1, then the torso sports is set to 0. If the sports can have the three major sports qualities of strength, speed, and flexibility, their weights are set to 1, then the two major sports qualities of endurance and agility are set to 0. Eventually, the sports model is:

$$\overrightarrow{l_u} = \left\{(p_u^1, w_u^1), (p_u^2, w_u^2), (p_u^3, w_u^3), \ldots, (p_u^7, w_u^7), (p_u^8, w_u^8)\right\} \tag{23}$$

Where p_u^n denotes the n th keyword, and w_u^n denotes the weight of the n th keyword, which is either 0 or 1, i.e., whether or not the sport has this keyword.

3.2.3. User-based collaborative filtering algorithms

The UB-CF algorithm is to measure and score the degree of liking of these sports based on the historical behavior of the sportsman, and then calculate the relationship between the user's attitude and preference for the same sport, the principle of the UB-CF algorithm is shown in Fig. 5. In the figure, sport 1 and sport 2 are recommended to user C because in the figure, both user A and user C have the same behavior, i.e., they have liked sport 2 and sport 3, and user B has not liked sport 3 or sport 4, so sport 1 and sport 2 are recommended to user C.

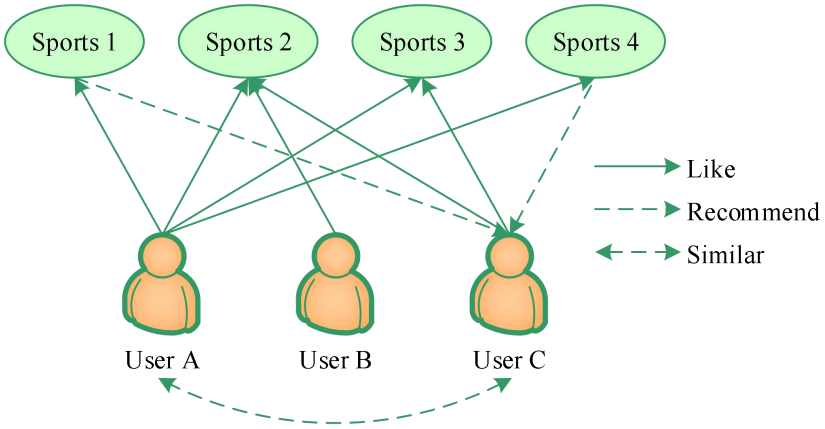


Figure 5. Schematic diagram of the UB-CF algorithm.

In this paper, Pearson's correlation coefficient is used to calculate the correlation between two users, and the calculated correlation result should be in the range of $[-1,+1]$, -1 means that there is an inverse influence between the two, +1 means a positive influence, and 0 means that there is no correlation, that is:

$$sim(u,v) = \frac{\sum_{p\in P_{u,v}} (r_{u,p} - \overline{r}_u)(r_{v,p} - \overline{r}_v)}{\sqrt{\sum_{p\in P_{u,v}} (r_{u,p} - \overline{r}_u)^2} \sqrt{\sum_{p\in P_{u,v}} (r_{v,p} - \overline{r}_v)^2}} \tag{24}$$

Where $P_{u,v}$ denotes the evaluation of the item by user u and user v together, $\overline{r}_u$ and $\overline{r}_v$ denote the average evaluation of user u and user v, p is the rating item of the item, and $r_{u,p}$ denotes the rating of

user u on item p.

The set of similar users or the nearest neighbor set of users of the target user to be recommended can be calculated by the above formula, and the set is denoted by N, then the ratings of the user u on the N set for the item p can be predicted to be $pred(u, p)$, and the $pred(u, p)$ is computed by the formula:

$$pred(u,p) = \overline{r}_u + \frac{\sum_{v \in N} sim(u,v) * (r_{v,p} - \overline{r}_v)}{\sum_{v \in N} sim(u,v)} \tag{25}$$

Where $sim(u,v)$ denotes the similarity of user u and user v, $\overline{r}_u$ and $\overline{r}_v$ denote the average rating of user u and user v, and $r_{v,p}$ denotes the rating of user v on item p.

Finally, based on the user's predicted value of the project, Top-N programs can be selected to the target user based on the descending list.

3.2.4. Content-based recommendation algorithms

The CB recommendation algorithm requires only one important feature, i.e., labeling, which entails decomposing the sports into a set of features that are sufficiently indicative of the sport and creating a relationship between the sport and the user based on the user's behavior in the system.The cosine similarity formula for the CB recommendation algorithm is:

$$sim(A,B) = \frac{\sum_{i=1}^{n} A_i * B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} * \sqrt{\sum_{i=1}^{n} (B_i)^2}} \tag{26}$$

Where A_i denotes the degree of user's preference for the type of sports, and B_i denotes the type of sports to which each sport belongs, and this belonging relationship is non-zero-i.e., one.

Finally, according to the similarity of sports in descending order, the Top-N can be selected to the user.

3.2.5. Hybrid recommendation algorithms

The more popular hybrid recommendation methods are as follows:

(1) Weighted hybrid: the recommended results of each algorithm are combined in accordance with a certain proportion of weights, and finally the results of the two combinations of the set of results in the screening of the top N results are recommended to the user.

(2) Switching hybrid: In the same project, for different scenarios, there are appropriate recommendation strategies, but in order to achieve better recommendation results, so the switching recommendation strategy, in different scenarios to switch the corresponding recommendation mechanism for the results of the recommendation.

(3) Partitioned hybrid: common in the front-end interface of the website, different regions show different recommendation results, this recommendation method uses a variety of recommendation mechanisms, the different results will be displayed to the user, the user can see the full range of recommended results, but also to facilitate the user to get the goods they want.

(4) Hierarchical mixing: Hierarchical combination of multiple recommendation methods, using the model generated by one recommendation method as the input of another recommendation method, thus combining the advantages of multiple recommendation methods and making the recommendation results more accurate.

(5) Waterfall mixing: Using multiple recommendation mechanisms, the first recommendation method is used to generate approximate recommendation results, and the second recommendation method makes more accurate recommendations based on the results of the previous recommendation.

(6) Feature combination: combining features from different recommendation mechanisms, and then inputting the combined features into a unique recommendation mechanism.

(7) Feature Enhancement: The feature information generated by one recommendation method is embedded in the features of another recommendation method, so as to achieve the effect of accurate

recommendation.

3.3. Application effect analysis of recommendation algorithm

The advantages and disadvantages between this paper's algorithm and the SVD decomposition method are examined by conducting experiments using both implicit feedback data (user's personal attributes, user's exercise data, and user's level of adoption) and explicit feedback data (user's ratings of the exercise instruction) actually collected by the system, and by using the root-mean-square error (RMSE) as a performance metric for evaluating the accuracy of the algorithm's recommendations.

3.3.1. Experimental data

This experiment requires a dataset of a certain size size to be extracted from the HDFS file system, including user ratings of exercise instruction, user personal attributes, user exercise data and user adoption level. The details of the specific database are as follows:

(1) User ratings: contains rating id, user id, exercise instruction id, rating value and timestamp (when a user uses the system a timestamp is generated which updates the rating value in the table).

(2) User Personal Attributes: contains user id, age age, gender sex, and sport level grade.

(3) User exercise data: contains id, user id, distance, steps, calories, and exercise program.

(4) User adoption level: contains id, user id, exercise instruction id, and adoption rate (i.e., the weight of user adoption level).

3.3.2. Experimental Procedures

Assuming that the difference between the user's real rating and the predicted rating obeys a Gaussian distribution, the corresponding loss function is jointly constructed using the rating matrix R (explicit feedback data) and the user's data behavior weighting matrix W (implicit feedback data), and then the loss function is optimized by the stochastic gradient descent algorithm (SGD), and the parameter values in the matrices P and Q are computationally solved using the existing historical rating data, so as to obtain the corresponding matrix P and matrix Q, i.e., matrix P is the user's hidden class matrix, representing the user's preference, and matrix Q is the exercise guidance hidden class matrix, representing the features of exercise guidance.

Since the data sets extracted from the HDFS file system in this experiment are all .txt files, which need some conversion processing, and at the same time, the method used is machine learning, which requires the use of a large number of third-party learning libraries, so the Python programming language is chosen for the implementation here. The following is the specific experimental process:

The first step is to extract a certain size of the data set from the HDFS file system, use the read function in Python to read the original file and convert the .txt format to .csv format, and then convert it to the matrix form through the DataFrame data structure common function under the pandas package.

In the second step, the weights of user data behaviors are calculated separately to generate the user data behavior weighting matrix W, which is then combined with the scoring matrix R to jointly construct the corresponding loss function.

In the third step, the loss function is optimized using the stochastic gradient descent algorithm (SGD), which determines the fastest descent direction by finding the partial derivatives of the parameters p_{uf} and q_{fi}, and then iteratively calculates the parameters to continuously optimize them until they converge. Here, the number of iterations is about 500 when the parameters converge and the iteration ends, outputting two low-dimensional matrices P and Q.

Part of the user hidden class matrix P is shown in Table 6, where each value indicates the degree of preference of the user u for the hidden class f motion guidance.

Table 6. Partial user implicit matrix P

user id	Aerobic exercise	Anaerobic motion	Indoor movement	Outdoor sports
382	0.5308	0.4696	0.3711	0.6287
13	0.7326	0.2681	0.2628	0.7374
385	0.3619	0.6396	0.6243	0.3706
62	0.2851	0.7215	0.7593	0.2369

107	0.5113	0.49	0.4809	0.5245
238	0.8333	0.1768	0.1914	0.8125
10	0.3922	0.6138	0.5893	0.4094
…	…	…	…	…

Part of the user exercise guidance hidden class matrix Q is shown in Table 7, and each value in the table indicates the weight of each exercise guidance i in the hidden class f.

Table 7. Some user movement guidance cryptomatrix Q

user id	Aerobic exercise	Anaerobic motion	Indoor movement	Outdoor sports
25	0.3593	0.6395	0.6318	0.3729
14	0.1774	0.8153	0.8061	0.1929
40	0.5389	0.4583	0.4488	0.5492
116	0.646	0.351	0.341	0.6632
40	0.2804	0.7216	0.7145	0.2936
102	0.7175	0.2828	0.2687	0.7298
141	0.8062	0.195	0.182	0.8184
204	0.4533	0.5505	0.5404	0.4576
88	0.4166	0.5843	0.5679	0.4327
…	…	…	…	…

In the fourth step, the two generated low-dimensional matrices P and low-dimensional matrix Q are cross-multiplied $P_u Q_i^T$ to obtain the corresponding predicted scoring values, and finally the corresponding recommendation results are given according to the predicted scoring values, and the predicted values of the exercise guidance $\hat{R}$ scores are shown in Table 8.

Table 8. The motion guidance predictive value R score

score_dataFrame_id	user_id	guide_id	score
54360	390	6	0.9486
54361	16	14	0.6999
54362	353	39	0.9806
54363	353	111	0.9204
54364	57	35	1.1973
54365	122	86	1.0063
54366	122	135	1.0183
54367	225	199	0.941
54368	39	97	1.0302
…	…	…	…

According to the predicted value $\hat{R}$ score table, the user-exercise guidance is sorted according to the score, and the exercise guidance with a high score is selected as the final result to be pushed to the user, and the details of some of the user recommendation results are shown in Table 9. Obviously after the exercise guidance, the duplicate user IDs are eliminated to make the user recommendation results more accurate.

Table 9. Some users recommend the results

User_id	Guide_id
390	6
16	14
353	39
57	35
122	86
225	199
39	97
…	…

3.3.3. Analysis of experimental results

The experimental results in this paper use the Root Mean Square Error (RMSE) as a performance metric to assess the accuracy of the algorithm recommendation, the smaller the value, the higher the accuracy of the algorithm recommendation. The following is a comparison between the predicted score and the actual score. The results of the comparison between the predicted score and the actual score under the method of this paper are shown in Table 10. Similarly, using the same experimental data, the user rating table is updated using the SVD decomposition method, and the results of predicted scores and actual scores under the SVD decomposition method are shown in Table 11. The results of comparing the trend of predicted scores with actual scores for both algorithms are shown in Fig. 6.

The results show that the value of RMSE for the recommended algorithm in this paper is 0.1016 and the value of RMSE for the SVD decomposition method is 0.4129; And the curve of the predicted value and the actual value under this paper's method matches well, and the curve of the SVD method deviates to a larger extent, which indicates that the improved recommendation algorithm has a higher recommendation accuracy, i.e., this paper's method is better than the traditional SVD decomposition method in terms of recommendation accuracy. This is because the singular value matrix decomposition model first needs to fill in the mean value of the historical rating data, and then carry out matrix decomposition to obtain the correlation sub-matrix, and such an approach reduces the model credibility, and the accuracy of the recommendation results will naturally be low. On the contrary, the method in this paper fully utilizes the user item rating matrix as the explicit feedback data, and at the same time introduces the idea of weighting to construct the user data behavioral weighting matrix as the implicit feedback data, which digs into the user's preference from both explicit and implicit aspects, making the improved recommendation algorithm more complete and the recommendation accuracy rate higher.

Table 10. The prediction score is compared with the actual score

User_id	Guide_id	P_score	R_score
390	6	0.9498	-
16	14	0.6995	0.6562
353	39	0.9806	-
353	111	0.9218	-
57	35	1.0006	0.9937
122	86	1.0106	-
122	135	1.0183	0.9172
225	199	0.9091	0.7377
39	97	1.0313	0.9196
…	…	…	…

Table 11. The SVD method predicts the scoring and the actual score

User_id	Guide_id	P_score	R_score
390	6	0.3008	-
16	14	0.8003	0.5956
353	39	1.0816	-
353	111	1.3014	-
57	35	0.7296	0.9362
122	86	1.3007	-
122	135	1.4202	0.9109
225	199	1.2703	0.7251
39	97	0.5207	0.8122
…	…	…	…

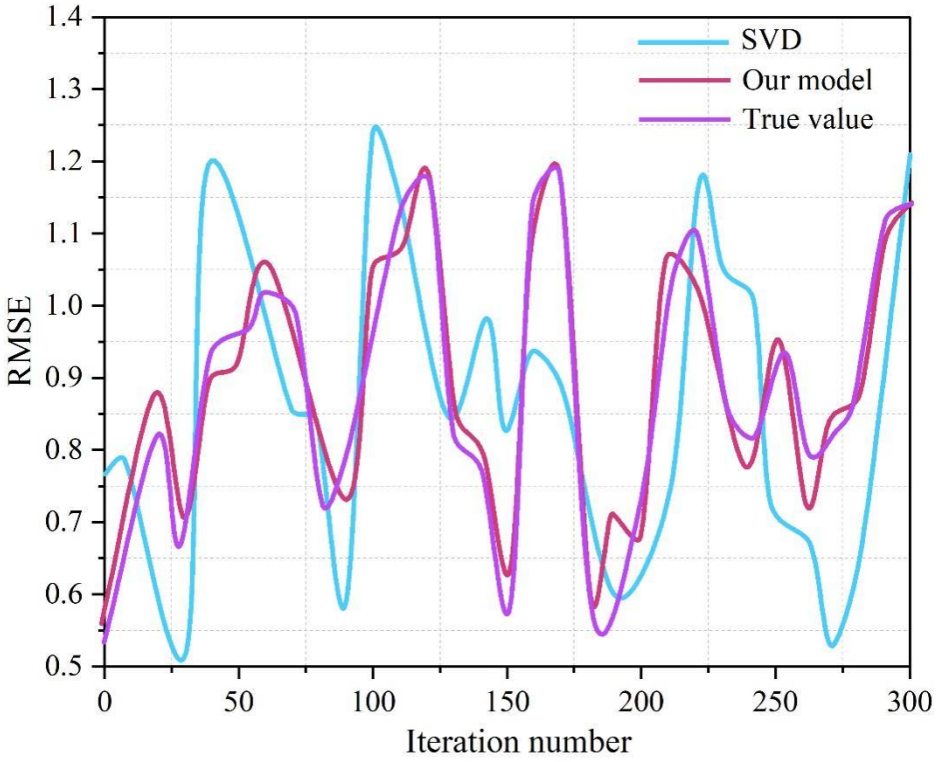


Figure 6. Two algorithms predict the comparison of the actual scoring trend

4. Conclusion

In this paper, the improved Cauchy variant gray wolf optimization algorithm is used to analyze the effect of students' sports training based on multilayer perceptron, and then the collaborative filtering algorithm is used to optimize the students' sports training program.

The accuracy of time domain feature extraction using the MLP classifier in this paper can reach more than 90%, especially in the group of x-y data, the classification accuracy can reach up to 97.32%, which is nearly 20% higher than that of the traditional classifiers, which shows that the method in this paper can get better classification results. The error between the predicted value and the real value under this paper's algorithm is the smallest (0.1016), and the curve between its predicted value and the actual value matches better, which reduces the RMSE value by 75.39% than that of the traditional SVD decomposition method, so it is obvious that this paper's method has a better recommendation effect.

References

1. Chen, K., Cao, F., Hao, L., Xiang, M., & Kamruzzaman, M. M. (2022). Application analysis of digital neural network-based data mining method in maximizing the performance of sports training. Revista Brasileira de Medicina do Esporte, 29, e2022_0152.
2. Lopatiev, A., Ivashchenko, O., Khudolii, O., Pjanylo, Y., Chernenko, S., & Yermakova, T. (2017). Systemic approach and mathematical modeling in physical education and sports. Journal of Physical Education and Sport (JPES), 17(1), 146-155.
3. Zelong, S., & Chunlan, F. (2024, December). Sports Training Data Analysis and Model Research Based on Artificial Intelligence. In 2024 IEEE 2nd International Conference on Electrical, Automation and Computer Engineering (ICEACE) (pp. 666-669). IEEE.
4. Lu, L. (2024, March). Intelligent Training Assistance System for Sports Posture Correction Feedback Based on Artificial Intelligence Algorithms. In 2024 International Conference on Distributed Computing and Optimization Techniques (ICDCOT) (pp. 1-6). IEEE.
5. Luo, L. (2022). A sports digital training system based on middle and bottom visual information. Mobile Information Systems, 2022(1), 6765954.
6. Yao, W., & Zhihai, Z. (2022). Design of sports training data monitoring system based on wireless internet of things. Mobile information systems, 2022(1), 4162088.
7. Rajšp, A., & Fister Jr, I. (2020). A systematic literature review of intelligent data analysis methods for smart sport training. Applied Sciences, 10(9), 3013.
8. Du, M., & Yuan, X. (2021). A survey of competitive sports data visualization and visual analysis. Journal of Visualization, 24, 47-67.
9. Liu, H., Zhang, Y., Lian, K., Zhang, Y., Martínez, O. S., & Crespo, R. G. (2022). Health care data analysis and visualization using interactive data exploration for sportsperson. Science China Information Sciences, 65(6), 162101.
10. Perin, C., Vuillemot, R., Stolper, C. D., Stasko, J. T., Wood, J., & Carpendale, S. (2018, June). State of the art of sports data visualization. In Computer Graphics Forum (Vol. 37, No. 3, pp. 663-686).
11. Yuan, R., Zhang, Z., Song, P., Zhang, J., & Qin, L. (2020). Construction of virtual video scene and its visualization during sports training. IEEE Access, 8, 124999-125012.
12. Dan, J., Zheng, Y., & Hu, J. (2022). Research on sports training model based on intelligent data aggregation processing in internet of things. Cluster Computing, 25(1), 727-734.
13. Kostiukevych, V., Imas, Y., Borysova, O., Dutchak, M., Shynkaruk, O., Kogut, I., ... & Stasiuk, I. (2018). Modeling of the athletic training process in team sports during an annual macrocycle. Journal of Physical Education and Sport, 18, 327-334.
14. Qian, L., & Liu, J. (2020). Application of data mining technology and wireless network sensing technology in sports training index analysis. EURASIP Journal on Wireless Communications and Networking, 2020(1), 121.
15. Lei, Q. (2023). Design of an Instant Data Analysis System for Sports Training Based on Data Mining Technology. International Journal of Web-Based Learning and Teaching Technologies (IJWLTT), 18(2), 1-15.
16. Knobbe, A., Orie, J., Hofman, N., Van der Burgh, B., & Cachucho, R. (2017). Sports analytics for professional speed skating. Data Mining and Knowledge Discovery, 31(6), 1872-1902.
17. Hu, M., Zhang, M., & Yu, K. (2024). Design of sports training information analysis system based on a multi-target visual model under sensor-scale spatial transformation. PeerJ Computer Science, 10, e2030.
18. Al Bataineh, A., Kaur, D., & Jalali, S. M. J. (2022). Multi-layer perceptron training optimization using nature inspired computing. IEEE Access, 10, 36963-36977.
19. Zimei Tu,Yichen Xie,Jinhua Jiang & Qin Qin. (2025). Point cloud registration based on surface feature extraction and an improved Grey Wolf Optimization algorithm. Scientific Reports,15(1),19199-19199.
20. Bao Yin Yin,Xing Cheng,Wang Jie Sheng,Zhao Xiao Rui,Zhang Xing Yue & Zheng Yue. (2023). Improved teaching–learning-based optimization algorithm with Cauchy mutation and chaotic operators. Applied Intelligence,53(18),21362-21389.