Personalized Movie Recommendation System Based on Support Vector Machine and Improved Particle Swarm Optimization

Xibin WANG†, Fengji LUO††††, Chunyan SANG††††, Jun ZENG††††, Nonmembers, and Sachio HIROKAWA††††, Member

SUMMARY With the rapid development of information and Web technologies, people are facing ‘information overload’ in their daily lives. The personalized recommendation system (PRS) is an effective tool to assist users extract meaningful information from the big data. Collaborative filtering (CF) is one of the most widely used personalized recommendation techniques to recommend the personalized products for users. However, the conventional CF technique has some limitations, such as the low accuracy of similarity calculation, cold start problem, etc. In this paper, a PRS model based on the Support Vector Machine (SVM) is proposed. The proposed model not only considers the items’ content information, but also the users’ demographic and behavior information to fully capture the users’ interests and preferences. An improved Particle Swarm Optimization (PSO) algorithm is also proposed to improve the performance of the model. The efficiency of the proposed method is verified by multiple benchmark datasets.

key words: personalized recommendation, support vector machine, particle swarm optimization, service computing

1. Introduction

With the advances in information technology, electronic commerce (E-commerce) has been undergoing rapid development in recent years. The explosion of today’s data also imposes some significant challenges on the provision of accurate and timely information to the user. The Personalized Recommendation System (PRS) provides a promising solution to this problem. By analyzing historical data, the PRS automatically provides suggestions to users based on their preferences. The core techniques used in the state-of-the-art PRSs can be classified into the content-based recommendation (CBR) technique and collaborative filtering (CF) technique. CBR selects most relevant items to the targeted user by comparing the representations of the item’s content and the user’s interest model [1]. CBR is limited in their applicability as they are purely based on the item’s textual information. Typically, a profile is formed for an individual user by analyzing the item’s content in which he/she is interested, and the additional items can be inferred from this profile. However, in many cases, the contents of the items are difficult to analyze.

CF uses explicit or implicit ratings drawn from users to recommend items to a given user [2]. It can be further classified into memory-based or model-based algorithms. The memory-based algorithms find neighbors of an active user and use the neighbors’ preferences to predict the unknown preferences of the active user. The limitations of the memory-based algorithms include the low accuracy of similarity calculation, high computational complexity, and so on. The model-based methods firstly develop a model based on the historical data, and then use it to predict the new preferences for an active user. Currently, many machine learning methods have been used for the model-based CF, such as the Backward propagation (BP) neural network [3], Adaptive learning [4], and Linear Classifier [5]. Currently, CF based recommendation techniques have been applied in a variety of areas, such as music recommendation [6], news recommendation [7], product recommendation [8], etc.

Support Vector Machine (SVM) is a widely adopted machine learning method. Compared with other machine learning approaches, SVM has some advantages. It assures that once a solution is reached, it is the global optimum. Since the forecasting accuracy of SVM is parameter sensitive, it would be important to choose the kernel function and tune the kernel parameters to achieve the desired performance [9]. Many heuristic algorithms have thus been used for the parameter optimization of SVM [10], such as the Grid Search (GS), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO). Comparing with other algorithms, PSO is recognized to have merits of strong global search capability and ease of implementation [10]. But the standard PSO also has some demerits. It often pre-matures into the local optimum and has slow convergence speed.

In this paper, we propose a novel improved PSO algorithm, and then propose a personalized movie recommendation system based on the proposed improved PSO and SVM. In particular, the major contributions of this paper include:

(1) To overcome the shortcomings of the conventional PSO, in this study we propose an improved PSO with the contraction factor and dynamic adaptively inertia weight
(CF-IWA PSO). CF-IWA PSO is embedded with a self-adaptively parameter adjustment mechanism to enhance both the global search ability and convergence speed. We then use the CF-IWA PSO to do the parameter optimization of SVM.

(2) Based on CF-IWA PSO and SVM, we propose a personalized movie recommendation system. Comparing with the traditional CF methods which only use the historical score data to calculate similarity, the proposed system not only utilizes the user’s demographic information, but also their rating information. These two kinds of information can well reflect the user’s preferences.

The rest of the paper is organized as follows. In Sect. 2, some technical backgrounds of SVM and PSO are introduced; Section 3 presents the principles of the proposed CF-IWA PSO; in Sect. 4, the mechanism of the proposed movie recommendation system is described; experiments are discussed in Sect. 5; finally, conclusions are drawn in Sect. 6.

2. Background

2.1 Introduction of SVM

The SVM classification model can be described as following:

Let the training data set as \{ (x_i, y_i) \}, where \( x_i \in \mathbb{R}^n \) denotes the input sample and \( y_i \in \{-1, +1\}, i = 1, \ldots, l \) represent the forecasted outputs, then the generalized linear SVM finds an optimal separating hyper-plane (shown as Eq. (1)) by solving the optimization problem formulated as Eq. (2),

\[
\begin{align*}
  f(x_i) &= (\omega \cdot x_i) + b \\
  \min_{\omega, b, \xi_i} &\frac{1}{2} ||\omega||^2 + C \sum_{i=1}^{l} \xi_i \\
  \text{s.t.} &\ y_i ((\omega \cdot x_i) + b) \geq 1 - \xi_i \\
  &\ \xi_i \geq 0
\end{align*}
\]

In Eqs. (1) and (2), \( c \) controls the equilibrium between the complexity of model and training error; \( \varepsilon \) is a preset constant which controls the tube size; \( \delta \) denotes the width of Gaussian kernel function and it affects the complexity of the sample data distribution in the high dimensional space. By introducing the Lagrange multipliers \( \alpha \) and solving the dual problem, we can obtain the classification hyper-plane as,

\[
f(x) = \text{sgn} \left( (\omega^* \cdot x + b^*) \right) = \text{sgn} \left( \sum_{i=1}^{l} \alpha_i^* y_i (x_i, x) + b^* \right)
\]

For nonlinear classification, assuming there is a transform from \( \phi : \mathbb{R}^n \rightarrow H \), then the nonlinear classification function \( K(x, x') = \phi(x) \cdot \phi(x') \), then the non-linear classification function can be obtained as,

\[
f(x) = \text{sgn} \left( \sum_{i=1}^{l} \alpha_i^* y_i K(x_i, x) + b^* \right)
\]

where \( \omega^* = (\omega_1^*, \omega_2^*, \ldots, \omega_l^*) \) is the solution. If the \( \alpha^* \) component of \( \alpha_i \) or \( \alpha_i^* \) is nonzero, then \( (x_i, y_i) \) are called the support vectors. Different functions can be applied as the kernel function of SVM, such as the Gaussian kernel function \( K(x, x') = \exp \left( -\frac{||x - x'||^2}{\sigma^2} \right) \), where \( \sigma \) is the kernel parameter.

It has been shown that in SVM, the choices of \( C, \varepsilon \), and \( \delta \) can significantly affect the performance of SVM. Where, \( C \) controls the equilibrium between the complexity of model and training error; \( \varepsilon \) is insensitive loss function, and it is a preset constant that controls tube size; \( \delta \) denotes the width of Gaussian kernel function and affects the complexity of the sample data distribution in the high space. Therefore, how to optimally select above parameters is an important issue.

2.2 Principles of Original PSO Algorithm

PSO is a heuristic based optimization algorithm proposed by Kennedy and Eberhart [11], and has been applied in many industrial applications such as our previous works [12, 13].

PSO maintains a swarm consisting of \( n \) particles. Each particle has a position vector \( \mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iD}) \) and a velocity vector \( 
\mathbf{v}_i = (v_{i1}, v_{i2}, \ldots, v_{iD}) \), where \( i = 1, 2, \ldots, n \). Each particle represents a potential solution for the optimization problem in a \( D \)-dimensional search space. In each generation, each particle is accelerated toward its previously visited best position and the global best position of the swarm. The best previously visited position of the \( i \)th particle is denoted as \( \mathbf{P}_i = (p_{i1}, p_{i2}, \ldots, p_{iD}) \); the best previously visited position of the swarm is denoted as \( \mathbf{P}_g = (p_{g1}, p_{g2}, \ldots, p_{gD}) \). The new velocity value is then used to calculate the next position of the particle in the search space. This process repeats until the preset termination criterion is achieved. The updates of the velocity and position vectors of a particle are based on following formulas,

\[
\begin{align*}
  \mathbf{v}_{id}^{l+1} &= w \times \mathbf{v}_{id}^l + c_1 \times \mathbf{r}_{1d}^l \times (p_{id}^l - x_{id}^l) \\
  &\quad + c_2 \times \mathbf{r}_{2d}^l \times (p_{gd}^l - x_{id}^l) \\
  x_{id}^{l+1} &= x_{id}^l + \mathbf{v}_{id}^{l+1}
\end{align*}
\]

where \( d = 1, 2, \ldots, D \); \( w \) is nonnegative inertial weight coefficient; \( c_1 \) and \( c_2 \) are learning factors; \( \mathbf{r}_{1d}^l \) and \( \mathbf{r}_{2d}^l \) are positive random number in the range of \([0, 1]\); \( l \) is the iteration index; \( x_{id}^l \) is the position of the \( i \)th particle in the \( d \)-dimensional space.

\( w \) controls the impact of the previous history of velocities on the current velocity. A larger value of \( w \) facilitates the global exploration, while a small value tends to facilitate the local exploration. To balance the global exploration and local exploration capabilities, the linear decreasing inertia weight [14–17] has been widely used to adjust the value of \( w \). This updating process can be described as Eq. (7),

\[
w(l) = w_{start} - l \times (w_{start} - w_{end}) / T_{max}
\]

where \( T_{max} \) is the maximum number of iteration; \( w_{start} \) and \( w_{end} \) are the maximum and minimum values of the inertia weight, respectively.
3. Introduction of CF-IWA PSO

3.1 Principles of CF-IWA PSO Algorithm

Based on the principles of the standard PSO, following realizations can be obtained:

Firstly, in the iteration process, the current best solution \( P_g \) is important, because near the \( P_g \) there could exists the real global optimal solution. The update of \( P_g \) is only determined by \( w \). Secondly, a good evolutionary algorithm should has following properties: it should have strong global search ability in early iterative and has strong local search ability in later iterations. However, in the original PSO, the key factor in balancing algorithm’s global and local search abilities is \( w \).

Based on this, it can be seen that it is necessary to design a strategy to update the value of \( w \), so as to enhance the algorithm’s performance.

The linear weighting decreasing strategy has some disadvantages. It is easy to fall into the local optimum; its global search ability will decrease in later iterations; and it has slow convergence speed. To overcome these disadvantages, the self-adaptively update strategy of the inertia weight \( w \) is proposed in this paper. In our method, in the early iterations of the algorithm, \( w \) is maintained as larger values to make the algorithm has strong global search ability and also accelerates the convergence speed. In the later iterations, \( w \) is maintained as smaller values to improve the algorithm’s local search ability and search accuracy. The updated criteron is shown as below.

\[
w(l) = \begin{cases} 2k^2(w_{\text{end}} - w_{\text{start}})/T_{\text{max}}^2 + w_{\text{start}}, & \text{when } l \leq 1/2T_{\text{max}} \\ 2\left(k^2 - T_{\text{max}}^2\right)(w_{\text{start}} - w_{\text{end}})/T_{\text{max}}^2 + w_{\text{end}}, & \text{when } l > 1/2T_{\text{max}} \end{cases}
\]

(8)

where \( w(l) \in [0.4, 0.9] \); \( l \) is the current iteration number.

To further improve the convergence speed of the PSO, this paper also introduces the contraction factor \( \chi \), and the formula (5) is then converted into the following form:

\[
v_{ij}^{t+1} = \chi \left(v_{ij}^t + c_1 \times \delta l_t \times (p_{ij}^t - x_{ij}^t) + c_2 \times \delta r_t \times (p_{ij}^t - x_{ij}^t)\right) 
\]

(9)

where the expression of \( \chi \) is shown as,

\[
\chi = \frac{1}{2 - \phi - \sqrt{\phi^2 - 4\phi}}, \quad \phi = c_1 + c_2, \phi > 4 \quad (10)
\]

Finally, we called this improved PSO algorithm as CF-IWA PSO.

3.2 Principles of CF-IWA PSO Based SVM

In this study, parameters \( \delta, c, \varepsilon \) are optimized by the proposed CF-IWA PSO algorithm. The procedures of the SVM classification model integrated with the CF-IWA PSO algorithm are shown as below:

1. **Step 1**: Read the sample data; prepare the training set and testing set; pre-process the sample data.

2. **Step 2**: PSO Population Initialization. Initialize the velocity and position vectors of each particle; set the values of the control parameters;

3. **Step 3**: Set the values of \( P_i \) and \( P_g \). Set the current optimal position of the \( i \)th particle as \( X_i = (x_{i1}, x_{i2}, \cdots, x_{in}) \) (that is \( P_i = X_i (i = 1, \cdots, n) \)) and the optimal individual in group as the current \( P_g \);  

4. **Step 4**: Define and evaluate fitness function. Use the classification accuracy as the fitness function value, which is shown as the following formula. At the same time, the 5-fold cross validation is used to evaluate the fitness.

\[
Acc = \frac{\text{The number of correctly classified samples}}{\text{The total number of samples}}
\]

Then, calculate the current fitness function value of each particle. Determine the best individual adapt position \( P_i \) and \( P_g \) based on the fitness function value, the particle’s historical optimal value, and the global optimal value.

5. **Step 5**: Update velocity and position of each particle. Search for the better \( c, \sigma \) and \( e \) according to the formulas (6) and (9);

6. **Step 6**: Update the number of iteration. Let \( t = t+1 \);

7. **Step 7**: Judge the stop condition. If \( t > T_{\text{max}} \) or \( Acc_j > acc \), then stop the iteration and \( P_g \) is the optimal solution which represents the best parameters for SVM. Otherwise, go to Step 4.

8. **Step 8**: Decoding the obtained optimal solution and get the optimized parameters.

4. CF-IWA PSO-SVM Based Movie Recommendation System

We select 2000 users’ score data from the MovieLens 1M data set[18] as the experimental data set. For each user, we randomly select 10 data as testing data, add them to the test data set, and the remaining data are used as the training set.

To establish the personalized movie recommendation model, the user’s demographic information, user’s behavioral information (“ratings”), and movie’s content information are integrated to form a “user-movie” correlation matrix. The correlation matrix is then trained by a training model, and finally the movies are classified (or “recommend”). The proposed PRS performs the movie recommendation based on the classification method instead of the similarity calculation of the traditional CF methods. Before establishing a classification model, movies are divided into two categories: “like” (recommended) and “dislike” (not recommended), based on the users’ ratings. We classified the “like” category as the movies with 4 or 5 stars, and the “dislike” category as the movies with 1, 2 or 3 stars. The procedures of building the proposed personalized recommendation model are as below.
4.1 “User-Movie” Correlation Feature Extraction

In our movie recommendation system, the relationship between the user and movie is essential for establishing the classification model. In this study, based on the MovieLens dataset, we use the user’s demographic information, movie’s information, and user’s ratings information about movies to realize the correlation between the user’s preference characteristics and movie’s information, shown in Fig. 1. There are 3 files in the MovieLens data set: movie.data, ratings.data and users.data. As implied by the file names, these files store the information of movies, users, and users’ ratings on movies, respectively. The primary and foreign keys of the 3 data tables provide the correlation relationships of above 3 categories of information. By analyzing the correlation relationships, we extract the users’ behavior and their preference information about the movies, and the ‘User-Movie’ relationship feature vector can be formed.

4.2 Personalized Movie Recommendation System

As discussed before, the CF methods have some limitations. The user-based collaborative filtering (UserCF) method needs to calculate the similarity between two users based on the items’ rating matrix; the item-based collaborative filtering (ItemCF) needs to calculate the similarity between two items based on the items’ rating matrix. The computational complexity of the UserCF is related to the number of users, which is proportional to the square of the number of users. For the ItemCF, when the number of items is large, its computational cost is also very high, which is proportional to the product of the square of the number of items and the sparsity. Comparing with the conventional CF methods, the machine learning based approach can significantly reduce the computational complexity. Moreover, taking into account the user’s demographic information can also alleviate the “cold start” problem to a large extent.

The personalized movie recommendation model is shown in Fig. 2. We first divide the ‘User-Movie’ relationship matrix (i.e. feature vector) into training data set and testing data set respectively, and perform the feature transformation on each of them. Then, we apply the proposed CF-IWA PSO to optimize the parameters of SVM. After that, we train the personalized movie recommendation model based on the SVM classifier and make predictions on the relationships between users and movies. Based on the prediction results, the movie recommendation list can be formed.

5. Experiments

5.1 Experimental Dataset

To test the classification accuracy of SVM, we adopt 5 data sets from the UCI data set [19], which are Diabetes, Wine, Iris, Sonar, and Vehicle.

In order to further verify the advantages of the proposed CF-IWA PSO, we firstly conduct a series of comparison experiments based on 7 well-known benchmark functions ($f_1$: Sphere, $f_2$: Rosenbrock, $f_3$: Rastrigrin, $f_4$: Schwefler, $f_5$: Ackley, $f_6$: Giewank, and $f_7$: Schwefel). We compared CF-IWA PSO with 6 PSO variants: standard PSO (SPSO), linear decreasing inertia weight PSO (LINW PSO), fine grained inertia weight PSO (FGIW PSO), double exponential self-adaptive inertia weight PSO (DESIW PSO), and dynamic inertia weight PSO (DEDIW PSO). The mathematical descriptions of the 7 benchmark functions can be found in literature [20]–[22].

To test the performance of PSO-SVM model in terms of the personalized movie recommendations, we select 2,000 users’ score data from the MovieLens 1M data set as the experimental data set. The MovieLens data set includes 2,000 users’ score data from the MovieLens 1M data set as the experimental data set. The MovieLens data set includes three files: usr.dat, movie.dat, and ratings.dat. For each user, we randomly select 10 data as the testing data, add them to the test data set, and the remaining data are used as the training set.

5.2 Performance Evaluation of CF-IWA PSO on Benchmark Functions

Firstly, we test the performance of the proposed CF-IWA
Table 1 Comparison results of CF-IWA PSO and other algorithms

| Function | SPSO  | CF-IWA PSO | LINW PSO | FGW PSO | DESIW PSO | DEDIW PSO |
|----------|-------|------------|----------|---------|-----------|-----------|
| $f_1$    | Mean  | $6.73 \times 10^{-27}$ | $6.73 \times 10^{-29}$ | $2.49 \times 10^{-41}$ | $20.2 \times 10^{182}$ | $16.73 \times 10^{229}$ | $12.03 \times 10^{211}$ |
|          | MSE   | $1.25 \times 10^{-26}$ | 0        | $1.21 \times 10^{-40}$ | 0          | 0         | 0         |
|          | OV    | $2.01 \times 10^{-28}$ | $4.78 \times 10^{-37}$ | $5.37 \times 10^{-28}$ | $1.98 \times 10^{-24}$ | $3.36 \times 10^{-26}$ | 0         |
| Mean     | 9.32  | 8.13       | 10.87    | 9.02    | 8.64      | 8.58      |
| $f_2$    | MSE   | 5.79       | 3.41     | 6.32    | 3.73      | 3.72      | 3.67      |
|          | OV    | $6.51 \times 10^{-6}$ | 0        | $1.57 \times 10^{-13}$ | 0          | 0         | 0         |
| Mean     | 2.27  | 1.27       | 1.93     | 1.23    | 1.16      | 1.18      |
| $f_3$    | MSE   | 5.23       | 1.20     | 1.18    | 1.15      | 1.09      | 1.06      |
|          | OV    | 0          | 0        | 0       | 0         | 0         | 0         |
| Mean     | 6.47  | $10^{-4}$  | 0        | 0       | 0         | 0         | 0         |
| $f_4$    | MSE   | $2.46 \times 10^{-3}$ | 0        | 0       | 0         | 0         | 0         |
|          | OV    | 0          | 0        | 0       | 0         | 0         | 0         |
| Mean     | 1.52  | $10^{-12}$ | 0        | 0       | 0         | 0         | 0         |
| $f_5$    | MSE   | $1.03 \times 10^{-2}$ | $2.16 \times 10^{-15}$ | 0.23 | $2.23 \times 10^{-15}$ | $2.18 \times 10^{-15}$ | $2.21 \times 10^{-15}$ |
|          | OV    | $4.02 \times 10^{-3}$ | $7.94 \times 10^{-13}$ | $2.75 \times 10^{-11}$ | $7.69 \times 10^{-13}$ | $7.45 \times 10^{-13}$ | $7.21 \times 10^{-15}$ |
| Mean     | 0.17  | $1.18 \times 10^{2}$ | $1.50 \times 10^{2}$ | $1.12 \times 10^{2}$ | $1.05 \times 10^{2}$ | $1.01 \times 10^{2}$ |
| $f_6$    | MSE   | $0.16 \times 10^{-2}$ | $1.07 \times 10^{-2}$ | $1.24 \times 10^{-2}$ | $1.05 \times 10^{-2}$ | $0.97 \times 10^{-2}$ | $0.89 \times 10^{-2}$ |
|          | OV    | $1.63 \times 10^{-2}$ | 0        | $6.66 \times 10^{-16}$ | 0          | 0         | 0         |
| Mean     | 5.25  | $2.05 \times 10^{-2}$ | $2.92 \times 10^{-5}$ | $2.48 \times 10^{-22}$ | $2.45 \times 10^{-22}$ | $2.39 \times 10^{-22}$ |
| $f_7$    | MSE   | $5.76 \times 10^{-2}$ | $8.26 \times 10^{-2}$ | $5.12 \times 10^{-5}$ | $8.47 \times 10^{-22}$ | $8.42 \times 10^{-22}$ | $8.38 \times 10^{-22}$ |
|          | OV    | $6.15 \times 10^{-3}$ | $1.30 \times 10^{28}$ | $8.53 \times 10^{8}$ | $1.41 \times 10^{28}$ | $1.38 \times 10^{28}$ | $1.34 \times 10^{28}$ |

PSO on the 7 benchmark functions with the 6 PSO variants. The number of variables (dimensions) for all test problems used in the experiments varies from 2 to 30. The swarm size is fixed to 30 and the maximum number of iteration is set to 3,000. The stopping criterion is set to allow the algorithms to run for the maximum number of iterations. The performances of six algorithms were measured by using the following metrics: Mean, Mean square error (MSE), and Optimal value (OV). For each algorithm, 50 independent runs are performed and the averaged results are obtained. The experimental results are shown in Table 1.

From Table 1, it can be seen clearly that the proposed CF-IWA PSO algorithm outperforms the SPSO algorithm and LINW PSO algorithm, and shows comparable performance with other three algorithms (FGIW PSO, DESIW PSO, DEDIW PSO). In particular, in $f_1$, DEDIW PSO algorithm shows the best performance; in $f_2$, CF-IWA PSO algorithm has the best results in the optimal value, average, and mean square error; in $f_3$, all algorithms converge to the optimal value, but the optimal value of CF-IWO PSO is better than several other algorithms.

From the above experiments, it can be found that the proposed CF-IWO PSO algorithm has good performance in both accuracy and convergence speed, and is suitable for real-world applications.

5.3 Validation of CF-IWA PSO on SVM’s Kernel Parameter Optimization

In this section, we validate the performance of the proposed CF-IWA PSO algorithm on the UCI data sets. We compare our proposed algorithm with some proven heuristic algorithms, including the PSO, genetic algorithms (GA), and grid search (GS). For PSO and CF-IWA PSO, the parameter settings are as follows: $c_1 = c_2 = 1.5$, $w_{end} = 0.4$; the initial speed range of the particles is set to be $[-5, 5]$; the population size is set to be 20; the maximum iteration number is set to be 100. For the SVM prediction model, the Gaussian kernel is used and the corresponding parameters are set as follows: $c \in [0, 100]$, $\delta \in [2^{-10}, 2^{10}]$, $\epsilon \in [2^{-10}, 5]$, and the settings are the same with [23]. For GS, the ranges of grid coordinates are set as $a = [-10, 10]$ and $b = [-10, 10]$; the model parameters for the grid points are set to be $c = 2a$, $\delta = 2b$; the search step of parameters is set to be 1. For GA, the population size is set to be 20; the maximum iteration number is set to be 100; the binary-coding is applied with the crossover probability of 0.8 and mutation probability of 0.2.

Normally, the performance of parameter combinations is significantly affected by the training data. For the same parameter pair of $(c, \delta)$, the fitting performance would often vary under different training data sets. This phenomenon is more evident on the small scale datasets, where the selection of the optimal parameters is affected largely by the randomness of the selected samples. For this reason, in this paper, 5-fold cross validation is adopted in PSO, GA and GS to comprehensively evaluate the performance of each combination $(c, \delta)$. Meanwhile, to avoid random error generated by experiments, each algorithm on the UCI data sets are executed 5 times, and the average results are taken and reported.

Table 2 reports the classification results of the four algorithms on the 5 UCI data sets, together with the optimal parameter combinations of each method. Since GS is a two-dimensional search algorithm and can only optimize two parameters, the PSO-SVM, GA-SVM and GS-SVM model are only used to optimize two parameters: $c$ and $\delta$. Based on the optimization results of CF-IWA PSO-SVM, the value of $\epsilon$ of PSO-SVM, GA-SVM and GS-SVM model are set to be 1.0.

Table 2 clearly shows that the proposed CF-IWA PSO
has better classification accuracy than other methods. When comparing with the standard PSO, CF-IWA PSO obtains higher classification accuracy on each data set. Specifically, for the Diabetes data set, the CF-IWA PSO has higher classification accuracy than PSO. This might because that PSO encounters the premature in later iterations, and thus could not find the global optima.

GS is a 2-dimensional optimization technique. It firstly sets the search ranges of $\delta$ and $c$ based on experience, and then evaluates the different discrete combinations of $\delta$ and $c$, and selects the optimal one. Some limitations of GS include: GS is often computational intensive when the training set is large; GS does not use the heuristic information of the model to accelerate the speed of searching the optimal parameters. When comparing with GS, in many cases CF-IWA PSO and GS are with approximately equal classification accuracies. However, the CF-IWA PSO does not have the shortcomings of GS presented above.

GA is a biological inspired heuristic algorithm, which encodes each solution of a given problem as a multi-dimensional vector (called ‘chromosome’), and applies crossover and mutation operators to change the chromosomes. GA iteratively performs the crossover and mutation operations, and update the population based on the fitness evaluation of the chromosomes. CF-IWA PSO’s information sharing mechanism is quite different with GA. In GA, chromosomes share information with each other, so the whole population is relatively evenly to move towards the optimal region; in CF-IWA PSO, the information flow is one-way. That is, only $gbest$ information is shared to other particles, which makes the whole search and update process follow the current optimal solution. By comparing with GA, the proposed CF-IWA PSO is easier to implement and has better global searching capability.

The comparative analysis of CF-IWA PSO and PSO is also performed on the convergence and fitness changes over the whole iterations. Figures 3-4 show the fitness profiles (classification accuracies) of the two algorithms on the Wine data set. From Figs. 3-4, it can be seen that CF-IWA PSO has stronger optimization ability than PSO. It can find the optimal parameter combinations with higher accuracy than PSO. Specifically, after 20 iterations, CF-IWA PSO adjusts search strategy, making the algorithm search the better parameter combinations than PSO in the iteration. These results show that the proposed method has good practicality and effectiveness.

### Movie Recommendation Evaluation

In this study, the personalized recommendation is converted into a binary classification problem. To verify the performance of the CF-IWA PSO based SVM classification model in the personalized recommendation, several other methods are also tested for comparison purpose. These methods include the item-based collaborative filtering (ItemCF), user-based collaborative filtering (UserCF), PSO-SVM model, GA-SVM model, GS-SVM model, and BP neural network model.

ItemCF method only considers the item’s unilateral information, but not considers the user’s demographic information in the similarity calculation. Therefore, we combine the item’s information with the user’s information, and establish a user-item relationship model to implement the preference relation forecast between the user and item.

### Table 2 Comparison of 4 algorithms on UCI dataset

| Data set | CF-IWA PSO-SVM | PSO-SVM | GA-SVM | GS-SVM |
|----------|----------------|---------|--------|--------|
| Accuracy | $c=1.376, \delta=1.076, \epsilon=0.82$ | $c=1.526, \delta=1.72$ | $c=1.526, \delta=0.92$ | $c=1.526, \delta=0.92$ |
| Diabetes | 85.2% | 78.5% | 82.3% | 81.7% |
| Wine | $c=2.04, \delta=4.15, \epsilon=0.82$ | $c=2.27, \delta=5.39$ | $c=2.27, \delta=5.66$ | $c=2.27, \delta=5.66$ |
| Iris | $c=14.13, \delta=2.35, \epsilon=0.96$ | $c=2.05, \delta=2.16$ | $c=2.05, \delta=2.16$ | $c=2.05, \delta=2.16$ |
| Sonar | 100.0% | 98.2% | 100.0% | 100.0% |
| Vehicle | $c=15.5, \delta=1.086, \epsilon=0.635$ | $c=1.283, \delta=1.163$ | $c=1.283, \delta=1.163$ | $c=1.283, \delta=1.163$ |
| Accuracy | 95.2% | 95.6% | 84.5% | 95.3% |

---

![Fig. 3](image1.png)

**Fig. 3** Classification accuracy of CF-IWA PSO on Wine dataset

![Fig. 4](image2.png)

**Fig. 4** Accuracy of PSO on Wine dataset

---

In this study, the personalized recommendation is converted into a binary classification problem. To verify the performance of the CF-IWA PSO based SVM classification model in the personalized recommendation, several other methods are also tested for comparison purpose. These methods include the item-based collaborative filtering (ItemCF), user-based collaborative filtering (UserCF), PSO-SVM model, GA-SVM model, GS-SVM model, and BP neural network model.

ItemCF method only considers the item’s unilateral information, but not considers the user’s demographic information in the similarity calculation. Therefore, we combine the item’s information with the user’s information, and establish a user-item relationship model to implement the preference relation forecast between the user and item.
Figure 5 shows the offline results on MovieLens data set of the recommendation based on the CF-IWA PSO-SVM, ItemCF, UserCF, PSO-SVM, GA-SVM, GS-SVM model, and BP neural network, respectively. The results show that with the increasing number of the training data, the classification accuracy of each method also rises. Meanwhile, the proposed CF-IWA PSO-SVM based PRS model has higher classification accuracy than other methods. When the training samples reach 90% of the entire training set, the classification accuracy of CF-IWA PSO-SVM reach 74.9%, higher than other methods: POS-SVM (73.7%), GA-SVM (69.2%), GS-SVM (75.5%), BP (66.3%), UserCF (51.7%), and ItemCF (52.8%). These encouraging results clearly show that the well-designed machine learning method can significantly improve the prediction accuracy of the recommendation system.

Figure 6 shows the evaluation results of the different methods in terms of the F-Score. As shown in Fig. 6, the CF-IWA PSO-SVM method provides better results than other 6 methods. In particular, when the number of training samples reaches 90% of the entire training set, the values of the F-Score of all the 7 methods reach the highest values.

In many cases, users are more interested in several or dozens of items on top of the recommendation list. In this study, the accuracy analysis is restricted to the top 10 recommended movies. From Fig. 7, it can be seen that the classification accuracy of the CF-IWA PSO-SVM was significantly higher than the other methods. When training samples reach 90% of the entire training set, the classification accuracy of CF-IWA PSO-SVM reach 79.6%, higher than other methods: POS-SVM (73.5%), GA-SVM (69.2%), GS-SVM (75.5%), BP (66.3%), UserCF (51.7%), and ItemCF (52.8%). These encouraging results clearly show that the well-designed machine learning method can significantly improve the prediction accuracy of the recommendation system.

6. Conclusions

In this paper, in order to overcome the limitations of the traditional collaborative filtering, a personalized movie recommendation model based on SVM is proposed. The proposed model not only considers the item’s content information, but also the user’s demographic and behavior information to fully capture the users’ interests and preferences.
a more direct manner, the model’s accuracy could be further improved.

Acknowledgments

This work was supported in part by the National Science Foundations of China under Grant No. 61379158; in part by the Scientific and Technological Research Program of Chongqing Municipal Education Commission under Grant No. KJ1500434; and in part by the Visiting Scholarship of State Key Laboratory of Power Transmission Equipment & System Security and New Technology (Chongqing University, China) (2007DA10512716401).

References

[1] M. Pazzani and D. Billsus, “Learning and revising user profiles: the identification of interesting websites,” Machine Learning, vol.27, no.3, pp.313–331, 1997.

[2] J.L. Herlocker, J.A. Konstan, A. Borchers, and J. Riedl, “An algorithmic framework for performing collaborative filtering,” Proc. 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.230–237. ACM, New York, 1999.

[3] D. Billsus and M.J. Pazzani, “Learning collaborative information filters,” Proc. 15th International Conference on Machine Learning (ICML’98), vol.98, pp.46–54, New York, 1998.

[4] W.-Y. Deng, Q.-H. Zheng, S. Lian, and L. Chen, “Adaptive personalized recommendation based on adaptive learning,” Neurocomputing, vol.74, no.11, pp.1848–1858, 2011.

[5] T. Zhang and V.S. Iyengar, “Recommender systems using linear classifiers,” J. Machine Learning Research 2, pp.313–334, 2002.

[6] M.S. Reddy and T. Adilakshmi, “Music recommendation system based on matrix factorization technique-SVD,” In Computer Communication and Informatics (ICCCI), 2014 International Conference on, IEEE, pp.1–6, 2014.

[7] Y. Seroussi, F. Bohnert, and I. Zukerman, “Personalised rating prediction for new users using latent factor models,” Proc. 22nd ACM Conference on Hypertext and hypermedia, New York, NY, USA, pp.47–56, 2011.

[8] W. Hong, L. Li, and T. Li, “Product recommendation with temporal dynamics,” Expert Systems with Applications, vol.39, no.16, pp.12398–12406, 2012.

[9] W. Hong, L. Li, and T. Li, “Product recommendation with temporal dynamics,” Expert Systems with Applications, vol.39, no.16, pp.12398–12406, 2012.

[10] J. Gu, M. Zhu, and L. Jiang, “Housing price forecasting based on genetic algorithm and support vector machine,” Expert Systems with Applications, vol.38, no.4, pp.3383–3386, 2011.

[11] J. Kennedy and R. Eberhart, “Particle swarm optimization,” Encyclopedia of Machine Learning, pp.760–766, 2010.

[12] F. Luo, J. Zhao, J. Qiu, J. Foster, Y. Peng, and Z. Dong, “Assessing the transmission expansion cost with distributed generation: an Australia case study,” IEEE Transactions on Smart Grid, vol.5, no.4, pp.1892–1904, 2014.

[13] Y. Zheng, Z.Y. Dong, F.J. Luo, K. Meng, J. Qiu, and K.P. Wong, “Optimal allocation of energy storage system for risk mitigation of DISCOs with high renewable penetrations,” IEEE Trans. Power Syst., vol.29, no.1, pp.212–220, 2014.

[14] Y. Shi and R.C. Eberhart, “Empirical study of particle swarm optimization,” Proc. 1999 Congress on Evolutionary Computation, vol.3, pp.101–106, 1999.

[15] A. Rathnaweera, S.K. Halgamuge, and H.C. Watson, “Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients,” Evolutionary Computation, IEEE Transactions on, vol.8, no.3, pp.240–255, 2004.

[16] J.C. Bansal, P.K. Singh, M. Saraswat, A. Verma, S.S. Jadon, and A. Abraham, “Inertia weight strategies in particle swarm optimization,” In Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on. IEEE, pp.633–640, 2011.

[17] M.A. Arasomwan and A.O. Adewumi, “On the performance of linear decreasing inertia weight particle swarm optimization for global optimization,” The Scientific World Journal, vol.2013, pp.1–12, 2013.

[18] MovieLens Datasets. http://grouplens.org/datasets/movielens/

[19] I.H. Witten and E. Frank, “Data mining: Practical machine learning tools and techniques,” Morgan Kaufmann, 2005. http://prdownloads.sourceforge.net/weka/datasets-UCI.jar

[20] M.A. Arasomwan and A.O. Adewumi, “Improved particle swarm optimization with a collective local unimodal search for continuous optimization problems,” Scientific World Journal, 3, 798129-798129, 2014.

[21] P. Chauhan, K. Deep, and M. Pant, “Novel inertia weight strategies for particle swarm optimization,” Memetic Computing, vol.5, no.3, pp.229–251, 2013.

[22] R. Mendes, J. Kennedy, and J. Neves, “The fully informed particle swarm: simpler, maybe better,” IEEE Transactions on Evolutionary Computation, vol.8, no.3, pp.204–210, 2004.

[23] X. Wang, J. Wen, F. Luo, W. Zhou, and H. Ren, “Personalized Recommendation System Based on Support Vector Machine and Particle Swarm Optimization,” In: Proceedings of the Knowledge Science, Engineering and Management, vol.9403, pp.489–495. Springer International Publishing, 2015.
Chunyan Sang received the MS and Ph.D degree from Chongqing University in 2010 and 2014. Since 2015 she has been a he lec-turer in the school of software engineering of Chongqing University of Posts and Telecommu-nications. Her research interests include data mining, machine learning, and Cyber Physical systems.

Jun Zeng received the BS and MS degree in software engineering from Chongqing University in 2007 and 2010. Since 2010, he has been a PhD student in advanced information technology of Kyushu University. His research interest includes Web data extraction, data mining, and search engine.

Sachio Hirokawa received the BS and MS degree in Mathematics and PhD degree in Interdisciplinary Graduate School of Engineering Sciences from Kyushu University in 1977, 1979, and 1992. Since 1997, he has been a professor in research institute for information technology of Kyushu University. His research interest includes search engine, text mining, and computational logic.