applied sciences

MDPI

Article

Interactive Multifactorial Evolutionary Optimization Algorithm with Multidimensional Preference Surrogate Models for Personalized Recommendation

Weidong Wu [1], Xiaoyan Sun [1,*], Guangyi Man [1], Shuai Li [1] and Lin Bao [2]

[1] School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China
[2] School of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang 212000, China
* Correspondence: 3730@cumt.edu.cn

Abstract: Interactive evolutionary algorithms (IEAs) coupled with a data-driven user surrogate model (USM) have recently been proposed for enhancing personalized recommendation performance. Since the USM relies on only one model to describe the full range of user preferences, existing USM-based IEAs have not investigated how knowledge migrates between preference models to improve the diversity and novelty of recommendations. Motivated by this, an interactive multifactorial evolutionary optimization algorithm with multidimensional preference user surrogate models is proposed here to perform a multi-view optimization for personalized recommendation. Firstly, multidimensional preference user surrogate models (MPUSMs), partial-MPUSMs, and probability models of MPUSMs are constructed to approximate the different perceptions of preferences and serve for population evolution. Next, a modified multifactorial evolutionary algorithm is used for the first time in the IEAs domain to recommend diverse and novel items for multiple preferences. It includes initialization and diversification management of a population with skill factors, recommendation lists of preference grading and interactive model management of inheriting previous information. Comprehensive comparison studies in the Amazon dataset show that the proposed models and algorithm facilitate the mining of knowledge between preferences. Eventually, at the cost of losing only about 5% of the Hit Ratio and Average Precision, the Individual Diversity is improved by 54.02%, the Self-system Diversity by 3.7%, the Surprise Degree by 2.69%, and the Preference Mining Degree by 16.05%.

Keywords: evolutionary optimization; multitask optimization; interactive; surrogate model; personalized recommendation

check for updates

Citation: Wu, W.; Sun, X.; Man, G.; Li, S.; Bao, L. Interactive Multifactorial Evolutionary Optimization Algorithm with Multidimensional Preference Surrogate Models for Personalized Recommendation. *Appl. Sci.* **2023**, *13*, 2243. https://doi.org/10.3390/app13042243

Academic Editor: Vincent A. Cicirello

Received: 15 January 2023
Revised: 4 February 2023
Accepted: 8 February 2023
Published: 9 February 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In traditional recommendation algorithms [1], content-based, collaborative filtering-based and hybrid methods, it is difficult to extract user preference features effectively. As a result, deep learning has become a hot topic in the field of recommendation due to its powerful feature extraction [2]. Although deep learning effectively fuses multi-source heterogeneous data into recommendation systems [3,4], there are few existing studies on dynamic recommendation, most of which only consider the accuracy of recommendation and lack diversity and novelty.

In the realm of personalized recommendation [5–8], the interactive evolutionary algorithms (IEAs) based on the user surrogate model (USM) are a highly effective search paradigm. The information obtained from interaction with users allows it to effectively track dynamic changes in user preferences. Due to its good performance in dynamic recommendation, it has been applied to scenarios such as movie recommendation [6], personalized pattern designs [8], preference guides [9], and energy system designs [10]. However, studies

related to IEAs still consider the diversity and novelty of recommendations and the use of deep learning methods in the construction of user surrogate models less.

To increase the diversity and novelty of recommendations, items with new combinations of attributes can be obtained by fully exploiting the characteristics of the items themselves. Mining new items for a preference by transfer knowledge from other related preferences is an effective approach. The search process of IEAs is guided by a USM. Therefore, unless the USM can describe preference information from multiple perspectives, IEAs cannot effectively mine the implicit information between different preferences. Currently, it is common to build several models to solve the problem of multiple perspectives [11–13]. For example, using multiple local surrogate models, Li et al. [11] proposed an evolutionary algorithm that approaches the optimal solution of the original problem from different directions. Although surrogate models have been able to describe the problem from various perspectives, previous studies have not considered the interaction and sharing of information between models.

Although there are strategies, subspace alignment [14], and feature mapping [15] to achieve feature transfer, there is no doubt that high computation will affect the timeliness required for personalized recommendation. Transferring genetic information between individuals of population is a viable way to address knowledge sharing between USMs. A multifactorial evolutionary algorithm (MFEA) [16] is a recently emerged approach for multi-task search. In MFEA, the evolutionary search is performed on multiple optimization problems or multiple search spaces simultaneously to improve evolutionary search capabilities by exploiting the associative knowledge between the tasks. In many successful cases of multi-task optimization scenarios, MFEA has demonstrated remarkable solution quality and search efficiency over traditional evolutionary algorithms [17–19]. To the best of our knowledge, the MFEA has not been extended to the IEAs-based personalized recommendation area. As we mentioned before, IEAs with only one surrogate cannot fully explore user preference information to provide diverse and novel items. Further consideration should be given to USMs describing user preferences from different perspectives, which will be searched as multiple optimization tasks, simultaneously.

Keeping these in mind, here we propose a new IEA for enhancing diversity and novelty of personalized recommendations. Firstly, deep learning-based multidimensional preference surrogate models (MPUSMs), partial-MPUSMs, and probability models of MPUSMs are developed, all of which are composed of multiple models representing different dimensions of preferences and serving the population evolution process. Then, an interactive multifactorial evolutionary optimization algorithm with multidimensional preference user surrogate models (MPUSMs-IMFEOA) is investigated, which combines the advantages of IEA's ability to update USM and MFEA's ability to migrate the preference knowledge of the user. It is the first attempt to apply MFEA to mining user preference knowledge in IEAs for personalized recommendation.

Accordingly, the primary contributions of our research are as follows. (1) In order to properly evaluate the population individuals, partial-MPUSM and MPUSMs are created. (2) In the evolutionary search phase, the modified MFEA is applied to the field of IEAs to improve the diversity and novelty of recommendations. Moreover, the efficiency of population evolution is improved with the help of probabilistic models. In addition, the normal genetic operators are adjusted accordingly because they are not sufficient to provide sufficient item diversity. (3) In the recommendation phase, a pre-recommendation list is proposed to prevent the problem of insufficient diversity, and both Top-N list and pre-recommendation list use roulette to focus on recent preferences and distinguish the importance of different preferences. (4) Lastly, a model management method for proposed models is also investigated, which can guarantee the correctness of the models and also inherit the valid information from the previous models.

The remainder of this paper is organized as follows: Section 2 reviews related work on personalized recommendation, interactive evolutionary algorithms and multifactorial evolutionary algorithm, and finally mentions the underlying surrogate model used in this

paper. Section 3 details the proposed interactive multifactorial evolutionary algorithm with the multidimensional preference surrogate models. Section 4 presents comprehensive empirical comparison experiments of the proposed algorithm. Finally, this paper is concluded and discussed in Section 5.

2. Related Work

This section will introduce four elements. Firstly, the development and shortcomings of traditional algorithms for achieving personalized recommendations are presented. Secondly, interactive evolutionary algorithms and their recent advances are described. Thirdly, multifactorial evolutionary algorithms that are most suitable for implementing knowledge migration in IEAs are introduced. Finally, the underlying model used in this paper for the construction of multidimensional surrogate models is illustrated.

2.1. Personalized Recommendation

Personalized recommendation systems typically present users with options that match their preferences as they browse short videos, listen to music or shop online without a clear goal in mind, so the recommendations should be accurate, diverse, and novel. Traditional recommendation algorithms are mainly content-based [20], collaborative filtering [21] and hybrid recommendation [22]. Although they are widely used in the recommendation field, they suffer from the problem of not being able to build user models quickly and not being able to represent item features adequately due to the difficulty of performing adequate feature extraction. In particular, collaborative filtering also often faces the problems of data sparsity (low number of user rated items) and cold start (no rating data for new items or users).

Deep learning brings new opportunities to solve these problems due to its powerful ability to extract user requirements features [2]. Yuan et al. [23] proposed a deep and wide model to successfully extract features from multiple sources of information such as user–item interaction matrix, attributes, and context. To extract dynamic information about user behavior sequences, Wang et al. [24] developed a model called an Interval- and Duration-aware Long Short-Term Memory network to capture long-term and short-term user preferences.

From the perspective of evolutionary optimization, personalized recommendation is a kind of dynamic optimization problems guided by user knowledge. The IEAs have been widely used in personalized recommendation problems to quickly track changes in user behaviors and habits to meet the individual needs [25–27]. An example is the combination of a Kano model and an interactive genetic algorithm proposed by Dou et al. [26] for enhancing the effective user-interactive process of creating customer-oriented products and responding quickly to customer's needs.

IEAs have an inherent advantage in tracking changes in user preferences, but there is relatively little research on combining deep learning to model user preferences and consider the diversity and novelty of recommendations. Hence, how to use the user preference surrogate model built based on deep learning to further guide the evolution of the population and thus uncover more diverse and novel items becomes a problem that must be solved. Since there are hidden relationships between different preferences of the user, this problem can be solved if useful knowledge can be transferred from other preferences so that a certain preference can explore new items. Thus, a new IEA is explored to search for multiple preferences while also transferring knowledge between preferences.

2.2. Interactive Evolutionary Algorithms with User Preference Surrogate Model

IEAs combine a user's evaluation information with traditional evolutionary algorithms to solve the problem of personalized recommendation involving human–computer interaction. In traditional IEAs [28], a user is involved in evaluating population individuals to guide population evolution. To alleviate user fatigue, some studies [29,30] build a user surrogate model to reduce the number of user evaluations. However, in practice, users

only have a very small number of scoring behaviors. To this end, Sun et al. [31] obtained user preferences indirectly from clicks, browsing and other behaviors to build a better USM. After each population iteration, some solutions with high fitness values are selected from the USM-evaluated population and recommended to the user. Finally, the model is updated according to behavior information to ensure that the next recommended items are more in line with the user's preferences. As can be seen, the accuracy of USM is the key to ensure that the user's favorite items are searched.

From Section 2.1, it is clear that most studies using deep learning methods to model user preferences have built only one model. Nevertheless, it is difficult to use only one model to guide evolutionary algorithms to extract inter-preference information efficiently. While some studies [11–13] have constructed multi-model USM by describing the problem from different perspectives, they have not yet investigated how information can be transferred between models under the framework of IEAs. This can not only improve the efficiency of problem solving but also increase the diversity and novelty of recommendations.

In addition, to accommodate different optimization objectives, IEAs can adopt different optimization strategies to optimize population. For example, there are differential evolution algorithms [32], non-dominated sorting genetic algorithms II [33] and even the estimation of distribution algorithms [34] that directly obtain population individuals by sampling a probabilistic model describing the distribution of solutions and thus achieving population evolution.

It can be found that existing research on user preference model are not conducive to explicit expressing multiple preferences and to guiding the mining of relationships between preferences, so as to find more items with higher diversity and novelty. Accordingly, we are motivated to design a novel IEA that optimizes the population guided by multiple USMs while transferring preference knowledge when individuals are mated.

2.3. Multifactorial Evolutionary Algorithm

As mentioned in Section 2.2, we need a strategy for optimizing population that not only addresses multiple goals simultaneously, user preference surrogate models, but also transfers information between models during the evolution of the population. It is worth noting that finding satisfactory items for each dimension of preferences can provide a wider choice of perspectives, which can likewise improve the diversity of recommendations. Considering an optimization task of each preference model as a unit of work, the multifactorial evolutionary algorithm (MFEA) [16] capable of optimizing multiple related tasks simultaneously is a natural selection. Optimization of multiple tasks differs from optimization of multiple objectives [35] in that the former aims to find the optimal solution for each task, whereas the latter seeks to discover a set of Pareto solutions. MFEA is structured similarly to a classical evolutionary algorithm, making it more economical to find new items for each preference by migrating the features of different tasks in each population iteration. In the case of related problems, transferring knowledge between different problem solutions will improve performance over solving problems in isolation [36–38].

MFEA is a classic algorithm in the field of evolutionary multitasking, as described in Algorithm 1. One of the most essential innovations in MFEA is the skill factor (τ), which represents the task to which an individual is most suited to solve among all the tasks. Each task is a unique factor that affects the evolution of a single population of individuals, so the population is implicitly divided into multiple skill groups by skill factors, each of which is good at solving corresponding tasks. Two components of MFEA control the inter-tasks (different skill groups) and intra-tasks (identical skill groups) knowledge transfer for the population: assortative mating and vertical cultural transmission, as shown in Steps i and ii of Algorithm 1. Assortative mating controls two randomly selected parental candidates to perform the crossover. Two candidates with the same skill factor can mate directly. Conversely, crossovers are performed between candidates with different skill factors if the random mating probability (rmp) condition is satisfied; otherwise, mutation is performed.

Vertical cultural transmission means that individuals of the offspring randomly inherit the skill factors from their parents and evaluate only the task corresponding to the skill factor.

Algorithm 1 Basic structure of the multifactorial evolutionary algorithm

1: Generate an initial population of individuals and store it in *current-pop (P)*.
2: Evaluate every individual with respect to every optimization task in the multitasking environment.
3: Compute the skill factor (τ) of each individual.
4: **while** stopping conditions are not satisfied **do**
5: i. Apply genetic operators on current-pop to generate an *offspring-pop (C)*. (Assortative Mating)
6: ii. Evaluate the individuals in offspring-pop for selected optimization tasks only. (Vertical Cultural Transmission)
7: iii. Concatenate offspring-pop and current-pop to form an intermediate-pop $(P \cup C)$.
8: iv. Update the scalar fitness (φ) and skill factor (τ) of every individual in intermediate-pop.
9: v. Select the fittest individuals from intermediate-pop to form the next *current-pop (P)*.
10: **end while**

Since MFEA is effective at solving multiple tasks and has not been extended to IEA-based personalized recommendation, it is necessary to carry out a corresponding study to improve further the IEAs performance in optimizing personalized recommendation. Using the useful knowledge implied between preferences, knowledge migration is performed by gene selection, crossover, and mutation at the chromosome level. This is carried out to mine further desired items of the user under every preference.

2.4. DRBM-Social User Surrogate Model

Here is a brief description of the user surrogate model proposed in [13] that integrates a trained dual Restricted Boltzmann Machine model with the social knowledge of similar users (DRBM-Social). It is selected as the base component of multidimensional preference user surrogate model proposed in this paper. Restricted Boltzmann Machine (RBM) is an energy-based stochastic neural network with a two-layer network structure, as shown in part in Figure 1, that learns the probability distribution of the input data. The visible layer v contains n units and its input is the observed data, and the hidden layer h contains m units and acts as a feature extractor. Given the state (v, h) and the parameter set $\theta = \{w, a, b\}$ where w is the connection weight, a and b are biases, it is easy to determine the RBM network state based on the energy function, as shown in (1). With the input observed data, all parameters in the RBM can be obtained by the Contrastive Divergence algorithm [39]:

$$E_\theta(\mathrm{v}, \mathrm{h}) = -\sum_{i=1}^{n}\sum_{j=1}^{m} v_i w_{ij} h_j - \sum_{i=1}^{n} a_i v_i - \sum_{j=1}^{m} b_j h_j \tag{1}$$

where v_i and h_j are the states of the i-th visible unit and the j-th hidden one, respectively. w_{ij} represents the connection weight between them, and a_i and b_i are their respective biases.

The prediction score of DRBM-Social surrogate contains two parts: the score predicted by the dual-RBM model and the score calculated based on social knowledge. Figure 1 presents the structure of the Dual-RBM which consists of a positive RBM and a negative RBM. As a first step, the evaluated items are divided into dominant and inferior groups according to the magnitude of their scores. Then, these two groups of items are used to train the Dual-RBM separately to obtain two sets of RBM network structure parameters. The item scores predicted by DRBM-Social are derived by combining social knowledge and user preferences which contain both positive and negative preferences, as shown in (2):

$$\hat{f}(x) = \sigma(\alpha * social(x) + preference(x)) \tag{2}$$

where α is the weight used to adjust the social knowledge contribution; σ is the activation function, which can be either a sigmoid or tanh function; $social(x)$ indicates the weighted average score of M similar users on item x and will be defined in (3):

$$social(x) = \sum_{j=1}^{M} Sim(u, u_j) * R_{u_j x} \tag{3}$$

where $Sim(u, u_j)$ is the Pearson similarity coefficient between the current user u and the neighbor u_j, and $R_{u_j x}$ is the rating of item x given by the neighbor u_j.

Then, the $preference(x)$ score of the item x is calculated based on its positive and negative preferences obtained in the positive RBM and negative RBM, respectively, as presented in (4):

$$preference(x) = \frac{\max(E_{\theta_p}) - E_{\theta_p}(x, \mathrm{h})}{\max(E_{\theta_p}) - \min(E_{\theta_p})} - \beta * \frac{\max(E_{\theta_n}) - E_{\theta_n}(x, \mathrm{h})}{\max(E_{\theta_n}) - \min(E_{\theta_n})} \tag{4}$$

where $E_{\theta_p}(x, \mathrm{h})$ and $E_{\theta_n}(x, \mathrm{h})$ are the energy values of item x calculated by the hidden layer h in the positive and negative RBM models, respectively. In the positive RBM model, $\max(E_{\theta_p})$ and $\min(E_{\theta_p})$ are the maximum and minimum energy values of all items which are generated by the probabilistic model and then matched to the real space to obtain. β is used to adjust the proportion of the contribution of the negative preference. An item with J attributes is expressed as $x = \{x_1, x_2, \cdots, x_J\}$, where $x_{1,2,\ldots,J} \in \{0, 1\}$.

Furthermore, a probability model [13] deduced by positive RBM only, as expressed in (5), is designed to efficiently reflect the degree of user preference for each attribute of the item and generate a superior population of individuals. According to (6), individuals x is generated by sampling the probability model randomly:

$$P(x) = [p(x_1 = 1), p(x_2 = 1), \cdots, p(x_J = 1)] \tag{5}$$

$$x_j = \begin{cases} 1, & if \quad random(0, 1) \geq p(x_j = 1) \\ 0, & otherwise \end{cases} \tag{6}$$

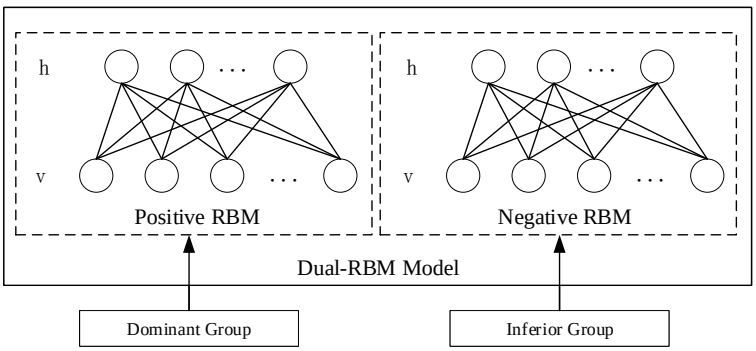


Figure 1. Structure of the Dual-RBM model.

3. Proposed MPUSMs-IMFEOA for Personalized Recommendation

In this section, the proposed Interactive multifactorial evolutionary optimization algorithm with multidimensional preference user surrogate models depicted in Figure 2 is presented in detail. The middle part of Figure 2 depicts the construction process of the multidimensional preference user surrogate model and its derived models. Firstly, K-means method and minimum Euclidean distance method are used to divide the historical evaluation items containing dominant and inferior items into K clusters representing different dimensional preference items. Secondly, K clusters of data are used to train K Dual-RBMs. Finally, based on K-trained dual-RBMs, partial-MPUSMs and MPUSMs for evaluating population individuals and items are obtained, as well as probabilistic models for generating initial population with skill factors. The proposed interactive multifactorial evolutionary optimization algorithm is illustrated in the left and right parts of Figure 2. The right side of the figure, from top to bottom, represents the improved MFEA algorithm, the two recommendation lists, and the user interaction process. The left part of the figure

shows the update of the K clusters to further update the MPUSMs, partial-MPUSMs, and probability models using a re-clustering or fine-tuning approach based on an updated historical item set incorporating the latest evaluation results. The mathematical symbols in this paper are summarized in Table 1.

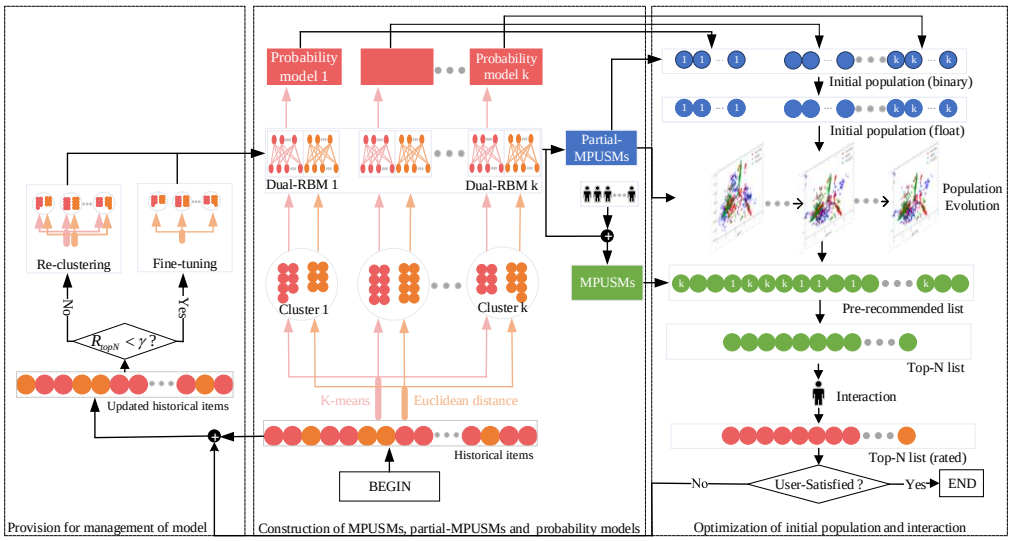


Figure 2. The framework diagram of the proposed MPUSMs-IMFEOA algorithm.

Table 1. Annotation of the symbols.

Symbol	Definition and Description
v	The visible layer of RBM
h	The hidden layer of RBM
α	Adjustment coefficient of $social(x)$
β	Adjustment coefficient of negative RBM
u	User
x	Real item
$x^{'}$	Virtual individuals representing items in the population
M	Number of similar users
U_{sum}	The number of users included in each dataset used for the experiment
K	Number of preference dimensions
r	Rating of item
ε	Threshold for dividing historical items into two parts
Q	Length of the pre-recommended list
N	Length of the Top-N list
rmp	Parameter to control crossover and mutation in MFEA
R_{topN}	RMSE of the Top-N list
γ	Threshold for managing the model
pop	Number of individuals in every skill group of the population
G	Number of evolutionary generations
W	Number of latest items to calculate the roulette probability

3.1. Construction of Multidimensional Preference User Surrogate Models and Their Derivatives

Different user preference behaviors can be simulated with multidimensional preference user surrogate models (MPUSMs) to evaluate unknown items. In addition, MPUSMs will be the base for running MFEA by viewing each surrogate as an optimized objective. Here, we combine multiple Dual-RBM models and social knowledge illustrated in Section 2.4 to describe multidimensional preferences. To properly evaluate the real items and unreal individuals of population, we design the MPUSMs and partial-MPUSMs, and to improve the evolutionary efficiency of MFEA, we derive probability models of MPUSMs. The construction process of MPUSMs and their derivatives is detailed in the middle part of Figure 2 and Algorithm 2. Key Innovative steps are marked with an asterisk.

Algorithm 2 Pseudocode for the procession of creating MPUSMs and their derivatives

Input: Historical evaluation items of user u and other M users
Output: partial-MPUSMs, MPUSMs, and probability models of MPUSMs
Provision of Social Knowledge
The $Social(x)$ for all items in the pool of items to be searched is obtained with (3).
***Clustering**
*Sort the historical items of the user u into dominant and inferior groups.
*The dominant group is clustered into K clusters by the K- means algorithm, which is repeated 10 times and the most balanced classification result is selected.
*Assign the inferior group to K clusters according to the distance of inferior items from the cluster centers.
Training
Train K Dual-RBM models with the data from K clusters.
*Form a partial-MPUSMs by combining K Dual-RBM models.
*Combine K Dual-RBM models and the user's social knowledge to form MPUSMs.
Calculate the K probability models based on positive RBMs of K Dual-RBM models using (5).

3.1.1. Clustering of Historical Items

Clustering the set of historical evaluation items is a critical step for dividing the user's different preference perceptions. To conform to the characteristics of the DRBM-social model, we first divided the historical data of user u into dominant group $(r \geq \varepsilon)$ and inferior group $(r < \varepsilon)$ based on the magnitude of the items' rating r and the threshold ε. For the dominant group, the K-means method is used to classify the historical items into K clusters. To prevent the number of samples in a cluster from being too small to train an accurate positive RBM model, we perform K-means clustering process several times so that the most evenly distributed result is selected from it. For the inferior group, the inferior items are assigned to K clusters based on the minimum Euclidean distance between the inferior item and the clustering centers. The negative RBM model only plays a subsidiary role, so even a small number of samples within the inferior cluster does not have much impact. At this point, K clusters are obtained as the base data for training K surrogates.

3.1.2. Training and Usage of MPUSMs and Their Derivatives

The middle part of Figure 2 and the training part of Algorithm 2 clearly depict the construction process of MPUSMs, partial-MPUSMs, and K probability models. As shown on the right in Figure 2, the proposed interactive multifactorial evolutionary optimization algorithm (IMFEOA) performs the MFEA evolution process for certain generations to obtain some relatively superior candidates and then eventually provides a Top-N list for interaction with the user. As the evolving individuals of population may not be real items, social knowledge derived from existing rated items cannot be used for this process. Hence, the evolving individuals are only evaluated with the partial-MPUSMs. After the evolution loop ends, the pre-recommended list will be filled with superior individuals and matched to real items, and the corresponding social knowledge could be added to provide more adequate evaluation information. To this end, two types of surrogates are designed for separately evaluating evolving individuals (or virtual items) and pre-recommended items.

Firstly, we train K dual-RBM models with K clustered data. Then, partial-MPUSMs consist of only K dual-RBMs to approximate the user's K dimensional preferences. Their formula for assessing individuals belonging to the k-th dimension of preferences is shown in (7) and is only used to evaluate the individuals x' in the MFEA evolutionary process. For evaluating the items x from the pre-recommended list, social knowledge is used together with K Dual-RBM models to form MPUSMs, as shown in (8). Finally, the probability models of MPUSMs derived from each positive RBM are calculated with (5) and an initial population that better matches the user preferences could be generated by (6). In these two types of models, the components $\max(E_{\theta_p})$, $\min(E_{\theta_p})$, $\max(E_{\theta_n})$ and $\min(E_{\theta_n})$ of the $preference(x)$ as defined in (4) are calculated based on the samples of each cluster, which is quite different from that in our previous work [13]:

$$\hat{f}_k^p(x^{'}) = \sigma(preference(x^{'})) \tag{7}$$

$$\hat{f}_k^c(x) = \sigma(\alpha * social(x) + preference(x)) \tag{8}$$

During the MFEA evolution process, (7) is used to directly evaluate the individuals with skill factor k. For the pre-recommended list, superior individuals are drawn from every skill group in the population, but their cluster features may have changed significantly as a result of evolution. To facilitate a reliable evaluation, these individuals should be converted to corresponding real items before finding the most appropriate preference surrogates for them to evaluate. Here, the Euclidean distance is used to find the real items for individuals in the set of items to be searched, as well as the most appropriate surrogate in the MPUSMs for the items in the pre-recommended list, for which the process is shown in (9):

$$r(x_i) = \hat{f}_k^c(x_i) \tag{9}$$

where $r(x_i)$ is the rating of the i-th item x; $k = \arg\min D_j$ denotes that the i-th item is best suited to be evaluated by the k-th surrogate model in the MPUSMs, of which $D_j = ||x_i - c_k||, (j = 1, 2, \cdots K)$ is the Euclidean distance between the i-th item and the k-th clustering center.

3.2. Interactive Multifactorial Evolutionary Optimization Algorithm with MPUSMs

In this section, we design a new IEA, Interactive multifactorial evolutionary optimization algorithm, compatible with the constructed MPUSMs to discover more diverse and novel items for recommendation. The main components of the IMFEOA include the following three parts: (1) Enhanced MFEA: For solving MPUSMs-aimed tasks and improving evolution efficiency, initial population with skill factors is obtained by sampling MPUSMs' probabilistic models that predicts the best search region. In addition, MFEA's knowledge sharing can increase the diversity of the items, but the use of common genetic operators in MFEA is still not sufficient to obtain sufficiently diverse items. MFEA with SBX crossover and Gaussian mutation is designed to optimize the personalized search to obtain more diverse and novel items. (2) MPUSM-based recommendation list: A pre-recommended list is creatively proposed in order to provide a Top-N list for user in line with diversity and current preferences. It is obtained by sampling evolved population with a roulette and resampling re-evolved population in case of insufficient diversity. (3) Interaction-assisted model management: MPUSMs and their derivatives are updated in accordance with the interaction results to learn useful information of the previous generation models while ensuring the accuracy of updated models. To the best of our knowledge, this is the first time to design an interactive MFEA for solving personalized searches. Its difference from common IEAs is that a user is not involved in the evaluation after each population iteration, which not only reduces user fatigue, but also improves the ability to uncover new items. The pseudocode of the specific process is shown in Algorithm 3. Asterisks indicate innovative key steps.

$$x_{ij}^{'} = \begin{cases} random(0-0.5) & x_{ij} = 0 \\ random(0.5-1) & x_{ij} = 1 \end{cases} \tag{10}$$

where x_{ij} is the value of the j-th gene locus of the i-th individual in the population.

Algorithm 3 Pseudocode of MPUSMs-IMFEOA

Input: Trained MPUSMs model, Partial-MPUSMs and probability models
Output: User-satisfied Top-N list

while Top-N list is not satisfied **do**
  while Number of items in the pre-recommended list $< Q$ **do**
    Enhanced MFEA:
    *By sampling pop times for each probability model, a better initial population is obtained and saved to current-pop.
    *Directly fill in skill factor for each individual of population.
    *Evaluate the initial population with partial-MPUSMs.
    *Encode the initial population according to (10)
    while *generation* $< G$ **do**
    Optimization of diversity population
      *Apply SBX and Gaussian mutation to current-pop to generate an offspring-pop.
      *Decode offspring-pop according to the inverse (10).
      *offspring-pop individuals are evaluated with partial-MPUSMs.
      Update the scalar fitness and skill factor of each individual in the merged current-pop and offspring-pop.
      Select the fittest individuals to form next current-pop.
    end while
    *Select the superior individuals from the evolved population with multiple skill groups by roulette to fill the pre-recommended list.
  end while
  ***Pre-recommended list:**
  Convert Q individuals to Q real items by matching them with the closest item in the item pool.
  The Q items are evaluated using the MPUSM.
  ***Top-N list:** Select the highest-rated N items in each category from the pre-recommended list using roulette.
  Interactive Evaluations: The Top-N list is submitted to the user for evaluation.
  ***Model Management:** Update the partial-MPUSMs, MPUSMs and probability models based on interaction results. In case $R_{topN} < \gamma$, then the models are fine-tuned. In case $R_{topN} > \gamma$, re-clustering and retraining the model are necessary.
end while

3.2.1. Enhanced MFEA

To further improve evolutionary efficiency, initial population with skill factors is generated based on multiple probability models that make individuals inherently suitable for solving the corresponding tasks. Thus, by sampling the corresponding probabilistic models with (6), this paper establishes K skill groups in the population, each with its own skill factor. In other words, the individual is created by the k-th probability model, and thereby its skill factor is k. Individuals are then evaluated with the partial-MPUSMs to prepare for the next step of population evolution.

In particular, the real items are represented as discrete binary vectors, and an item with J attributes is expressed as $x = \{x_1, x_2, \cdots, x_J\}$, where $x_{1,2,\ldots,J} \in \{0, 1\}$. They are encoded continuously with (10) so that continuous crossover and mutational operators can be accommodated to ensure population recommendation diversity. Consequently, individuals have two encoding representations: the discrete form for evaluation and the continuous form for evolution.

Given that personalized recommendations require both satisfying user's preferences and conforming to the diversity of recommendation, it is important to ensure population diversity in order to select a sufficient number of items for each preference dimension. As a result, a targeted adjustment must be made to the MFEA. After specific experiments, simulated binary crossover (SBX) and Gaussian mutation are chosen to accomplish the migration

of preference knowledge, as described in Section 4.2. It is noticeable that individuals in evolution are unrealistic, so we use partial-MPUSMs to evaluate them.

As crossover and mutation occurs between skill groups, knowledge transfer will occur between different dimensional preferences. Eventually, with the continuous iterative evolution of population, the implicit preferences of user will be uncovered and a population with excellent individuals in multiple preference dimensions will be formed.

3.2.2. MPUSM-Based Recommendation List

Two selection processes for Top-N items are designed here: coarse and fine selection to obtain a pre-recommended list and a Top-N list, respectively. During the coarse selection stage, although population diversity has already been enhanced by changing the coding form and adjusting the crossover and mutation operators, sometimes, an evolved population may not be sufficient to provide enough unduplicated individuals. For this reason, a new initial population generated by probability models is again evolved to provide more non-repeating individuals. In addition, MPUSM separates user's preferences, so that user's attention to different dimensions of preferences can be determined based on recently assessed items. Therefore, the roulette wheel selection is applied, which not only takes into account each preference's importance but also provides more randomness. According to (11), the probability of roulette is calculated. Moreover, the preference dimension to which the item belongs is determined by calculating the magnitude of the Euclidean distance of the item from the K clustering centers:

$$F = [f_1, f_2, \cdots, f_K] \tag{11}$$

where f_k denotes the probability of attention to the k-th dimensional preference and $f_k = w_k/W$. w_k represents the number of items belonging to the k-th cluster in the latest W historical evaluations.

The population is divided into K parts by skill factors, so it is easy to extract diverse and non-repeating individuals for the pre-recommended list using the roulette. At this time, individuals with higher rating are given priority for extraction. Once the population has no more non-replicated individuals, a newly created population is re-evolved for filling the pre-recommended list. After the pre-recommended list is filled, the individuals are transformed into nearest real items in an item pool using the minimum Euclidean distance. Then, they need to be evaluated with MPUSMs, at which point the list is still divided into K groups.

During the fine selection stage, a Top-N list containing the N most desired real items is generated. Currently, a pre-recommended list contains items on multiple dimensional preferences, and each item is evaluated with utmost precision MPUSMs. Thus, the roulette is used again to draw N items from the pre-recommended list to form the Top-N list.

3.2.3. Interaction-Assisted Model Management

Model updating is an essential part of tracking user preference changes. The role of this part is to recommend the Top-N list to the user and to update the MPUSMs, partial-MPUSMs and probability models according to the interaction results. In order to retain the helpful information from the previous models while updating the model, we keep information that correctly classifies preferences in case the MPUSMs are accurate. In this way, the user's preferences are retained, and the latest information can be added. That is, if the models are accurate, they will be fine-tuned, while if they are not accurate, they must be completely retrained. The specific work is as follows.

In this work, the root mean square error (RMSE) of the true and predicted ratings of the items for training MPUSMs is used as the threshold γ, and the RMSE of the Top-N list is noted as R_{topN}. If $R_{topN} < \gamma$; this means that the model preference classification and prediction is accurate, so the MPUSMs and their derivatives can be fine-tuned directly to extend the users preferences. In addition, if $R_{topN} < \gamma$, this indicates that the prediction

results of the models are inaccurate. That is, the classification of user preferences does not match user habits and needs to be re-clustered.

Following is a detailed explanation of the fine-tuning and re-clustering process. Firstly, the obtained evaluation results of the Top-N list are combined with the historical evaluation items. If fine-tuning is needed, then the updated historical evaluation items are grouped into K classes by following the previous clustering centers. If re-clustering is necessary, then the clustering process of Algorithm 2 is executed for the updated historical evaluation items. Finally, the models are updated by running the training process in Algorithm 2, either after fine-tuning or re-clustering.

3.3. Evaluation Indicators

Three types of evaluation metrics, namely accuracy, diversity, and novelty, are used to assess the Top-N list for comprehensively evaluating the performance of MPUSMs-IMFEOA. Among them, the accuracy [13] is evaluated by three indicators: Root Mean Square Error ($RMSE$), Hit Ratio (HR), and Average Precision (AP). The Diversity [40] metrics are set as individual diversity (Div) and Self-System Diversity (SSD). The novelty includes surprise degree (Sur) [40] and preference mining degree (PMD), where the PMD is firstly and specifically proposed to evaluate the effectiveness of our algorithm in mining inter-preference knowledge. Below are details of the evaluation metrics.

3.3.1. Root Mean Square Error (RMSE)

$RMSE$ calculates the error between the predicted rating and the user's real rating:

$$RMSE = \sqrt{\frac{\sum_{i \in S} (r_i - \hat{r}_i)^2}{|S|}} \tag{12}$$

where r_i denotes the user's true rating of item i, and $\hat{r}_i$ is the predicted one. S represents the set of items to be evaluated, and $|S|$ is the number of items in the set S.

3.3.2. Hit Ratio (HR)

HR is the ratio of the number of satisfying items in the Top-N list to all the items that the user likes in the item pool of items to be searched. The larger the HR, the more satisfying items in the Top-N list:

$$HR = \frac{Hit}{|GT|} \tag{13}$$

where Hit is the number of satisfying items in the Top-N list, and $|GT|$ is the number of items in the item pool that the user is satisfied with.

3.3.3. Average Precision (AP)

AP measures the impact of item ranking on user experience perception, and a higher AP value indicates a higher ranking of items for user satisfied items:

$$AP = \frac{1}{|L|} \sum_{i=1}^{|L|} \frac{i}{position(i)} \tag{14}$$

where $position(i)$ is the ranking position of the user preferred i-th item in the Top-N list. $|L|$ is the number of items in the Top-N list.

3.3.4. Individual Diversity (Div)

Div measures the diversity of items in Top-N list items, focusing on whether items with different dimensional preferences can be recommended to the user to avoid homogenization:

$$Div(L) = 1 - \frac{\sum_{i,j \in L, i \neq j} s(i,j)}{|L|(|L|-1)} \tag{15}$$

where $s(i,j) \in [0,1]$ defines the similarity of items in the Top-N list, and $s(i,j) = 1 - d(i,j)/\sqrt{D}$. Among them, $d(i,j)$ denotes the Euclidean distance between the i-th item and the j-th item, and D is the dimension of the item vector.

3.3.5. Self-System Diversity (SSD)

SSD represents the ratio of items recommended this time that are not similar to those recommended last time. The user's interest in the item changes over time, and they prefer new recommendations that differ from those in the past. The larger the SSD, the better the temporal diversity of the recommended list:

$$SSD = \frac{|L_t / L_{t-1}|}{|L_t|} \tag{16}$$

where L_{t-1} is the last recommendation list of L_t, and $L_t / L_{t-1} = \{i \in L_t | i \notin L_{t-1}\}$.

3.3.6. Surprise Degree (Sur)

Sur means that, if the recommended items are not similar to the user's historical preferences, but the user is satisfied with that, the recommendation is said to have a high degree of surprise:

$$Sur(x) = \frac{1}{|L||H|} \sum_{i=1}^{|L|} \sum_{h=1}^{|H|} \frac{r_i}{\frac{r_h}{r_{\max}} s(i,h)} \tag{17}$$

where r_i represents the user rating of the i-th recommended item. r_h represents the rating of the h-th historical item, and $r_{\max}$ is the maximum rating in the set of historical items. $|H|$ is the number of items projects evaluated historically.

3.3.7. Preference Mining Degree (PMD)

PMD measures the algorithm's ability to mine user preference information. By calculating the degree of dissimilarity between the Top-N list and the set of items evaluated historically, it evaluates whether the algorithm can predict the user's unexpressed preferences by learning the user's historical preferences. A larger PMD value indicates that the algorithm is more capable of mining user preferences:

$$PMD = 1 - \frac{1}{|L||H|} \sum_{i=1}^{|L|} \sum_{h=1}^{|H|} s(i,h) \tag{18}$$

where $|H|$ is the number of items that have been rated by the user. $s(i,h)$ represents the similarity between the item i in the Top-N list and the historically evaluated item h.

4. Experiments and Results

To verify the effectiveness of the proposed algorithm in the field of personalized recommendation, comprehensive empirical comparison studies are conducted in this section. The dataset sources and processing, as well as the experiment-related parameter settings and compared algorithms and surrogate models, are first described in Section 4.1. Then, four sets of experiments are designed. (1) Effect of genetic operators on population diversity; (2) Validate the accuracy of MPUSM evaluation items compared to DRBM-Social; (3) The recommended performance of the proposed MPUSMs-IMFEOA is verified by comparing it with SC-DRBMIEDA [13], which is also based on the DRBM-social model; (4) Visually demonstrate the search capability of MPUSMs-IMFEOA by comparing it to other IEAs.

4.1. Experimental Setup

Three datasets from Amazon [41] are selected for the experiment, and the statistical information of the datasets is described in Table 2. Based on the dataset, the experiments mainly use item IDs, user IDs, review time, categories, and rating information. Each experimental dataset is composed of evaluation data from U_{sum} randomly selected users in the dataset. In each experimental dataset, one user is selected as the experimental subject, and the remaining users who are similar will provide social knowledge for that user. If all items evaluated by a user have a total of J categories, the items are expressed as $x = \{x_1, x_2, \cdots, x_J\}$, where $x_{1,2,\ldots,J} \in \{0, 1\}$. The evaluation data of the experimental subjects are processed as follows.

Table 2. # of users, items and ratings of the datasets.

Dataset	# of Users	# of Items	# of Ratings
Digital_Music (Music)	478,235	266,414	836,006
CDs_and_Vinyl (CDs)	1,578,597	486,360	3,749,004
Kindle_Store (Kindle)	1,406,890	430,530	3,205,467

Considering the actual situation of interactive behavior, the set C of historical evaluation items of a user is sorted by review time. Then, it is divided to obtain multiple training sets (sliding window) and item pools containing the items to be searched at different interactions, as shown in Figure 3. In these experiments, the set C is divided into two parts, one for the first 70% of items as the set C_{HI} regarded as the historical interactive items, and the other as the fixed set C_F of items to be searched. The last 30% of the set C_{HI}, named C_I, is considered as the result of 10 actual interactions. The training items are obtained from the set C_{HI} through a sliding window with a fixed length of $0.4 * |C_{HI}|$. $0.1 * |C_I|$ items following the sliding window is seen as the actual result of the interaction, and all the data after the sliding window in C are regarded as an item pool to be searched, so that the item pool decreases gradually as the interaction proceeds. To be relevant, the updated history items in experiments include sliding window data and the Top-N list, except for the first interaction. However, the R_{topN} used for model management is still calculated from the Top-N list rather than real interaction results.

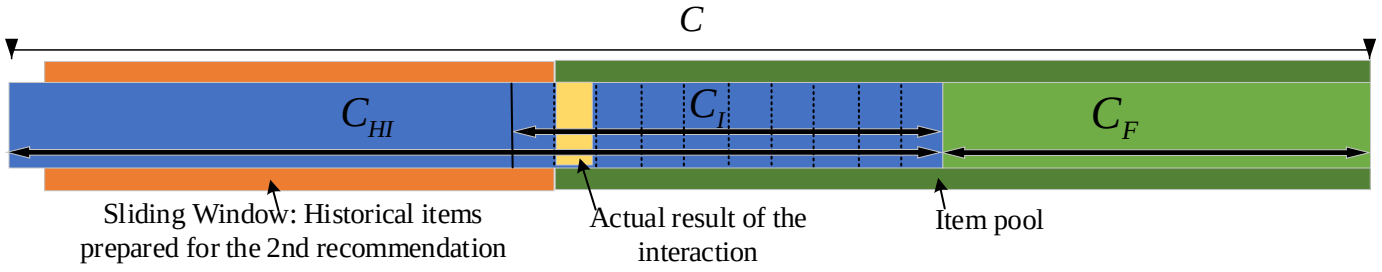


Figure 3. The data set division of the user.

In these experiments, MPUSMs and their derived models are created based on Dual-RBM [13], so the parameters (number of units in the hidden and visible layers) of the Dual-RBM and the training method are the same as in [13]. In addition, the parameters (α, β) related to the calculation of scores for MPUSMs and partial-MPUSMs are referred to [13]. Moreover, IMFEOA is built based on MFEA [16], so the parameter rmp is referenced to [16]. In addition, the population to be optimized by IMFEOA will evolve until no new individuals can be searched for, so that unknown items can be fully explored (G = 100). The remaining parameters are deliberately set to be identical to SC-DRBMIEDA [13] in order to be able to compare with it in the same context. Detailed parameter configurations are given

in Table 3. The user surrogate models and algorithm descriptions used for comparison in these experiments are shown in Table 4.

Table 3. Experimental parameters of the proposed algorithm.

Parameter	Value
α	0.3
β	0.2
K	3
ε	3
U_{sum}	200
rmp	0.3
pop	50
G	100
Q	100
N	10
W	50
Number of visible units of RBM	# of attributes
Number of hidden units of RBM	1.2 * # of attributes

1 The '# of attributes' represents the number of attributes of the items. 2 The '*' is the multiplication sign.

Table 4. The comparison models and algorithms used in this paper.

DRBM-Social [13]	It uses a model based dual-RBM and social knowledge to describe the user's preference information
RBM [42]	It uses an RBM-based model to describe the user's preference information
MPUSMs based on RBM	They use RBM as the underlying model for building MPUSMs
SC-DRBMIEDA [13]	It is an IEA that uses the same DRBM-Social as the proposed algorithm, but it does not implement knowledge transfer between preferences.
DRBM-Social-IGA	It is an IEA using a general genetic algorithm to optimize DRBM-Social.
RBM-IGA	It is an IEA using a general genetic algorithm to optimize RBM.
MPUSMs-IMFEOA based on RBM	It differs from the algorithm proposed in this paper in that it replaces the underlying surrogate model for constructing MPUSMs

4.2. Experiment 1—Effect of Genetic Operators on the Variety of Individuals in the Population

The genetic operators have a significant impact on the performance of the proposed IMFEOA algorithm. It directly determines the diversity of individuals in the population. As described in Section 3.2.1, individuals have two forms of coding. In this experiment, Two-Point Crossover, Uniform Crossover, and the method of sampling RBM probability model (Prob) applicable to binary vectors are chosen, as well as the Simulated Binary Crossover (SBX) and Gaussian Mutation (Gauss) methods suitable to float vector.

During this experiment, we randomly select the user with ID A9Q28YTLYRE07 in the music dataset, and the first sliding window data are selected after processing according to the method in Figure 3. Common recommended settings used for all genetic operators. Combining the three crossover operators and two mutation methods, the effect of the experimental genetic operators on the variety of individuals in the population is shown in Figure 4. It is clear that only the SBX and Gaussian Mutation operators are capable of maintaining the diversity of the population.

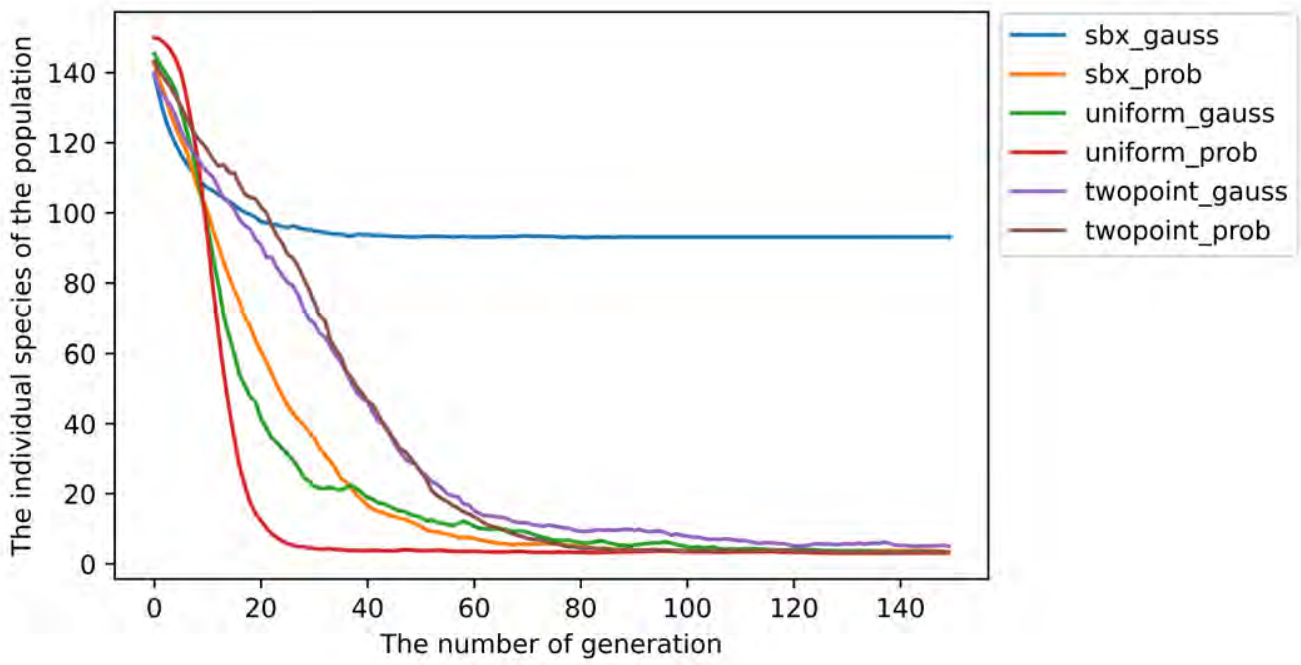
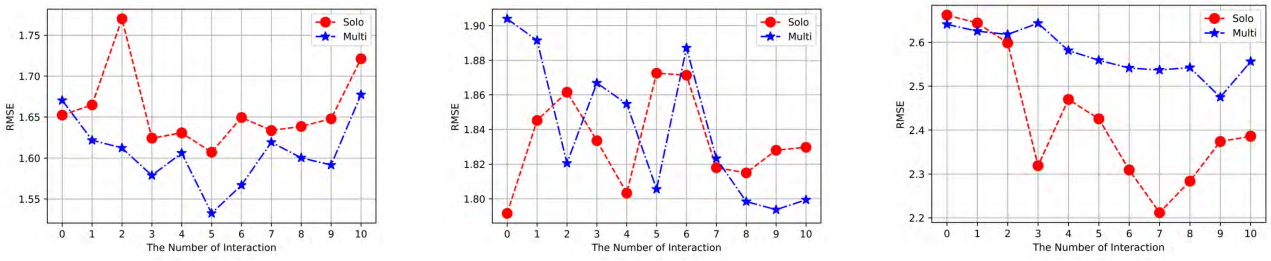


Figure 4. Effect of six gene operator combinations on the number of non-repeating individuals in the population.

4.3. Experiment 2—Evaluation of the Rating Accuracy of MPUSMs

Test the accuracy of MPUSMs with the DRBM-Social surrogate on different vector dimensions, and they represent the accuracy of multiple local models and a single overall model, respectively. In each dataset, a user with different numbers of categories is selected. The IDs of the three users are 'A9Q28YTLYRE07' (Music), 'A2582KMXLK2P06' (CDs), and 'A320TMDV6KCFU' (Kindle), and their corresponding number of categories are 187, 233, and 149, respectively. To enable a reasonable comparison, the local model of MPUSMs has the same parameters as DRBM-Social and the same parameters in different vector dimensions. As the user has been configured to perform ten interactions, there are ten historical item sets and item pools. Using historical item sets as a basis for rating items in the item pools. Figure 5 shows the situation for three users. Multi and Solo in the figure denote MPUSMs and DRBM-Social.



(**a**) Music (**b**) CDs (**c**) Kindle

Figure 5. RMSE performance of MPUSMs and DRBM-Social on three users with different vector dimensions during 10 interactions.

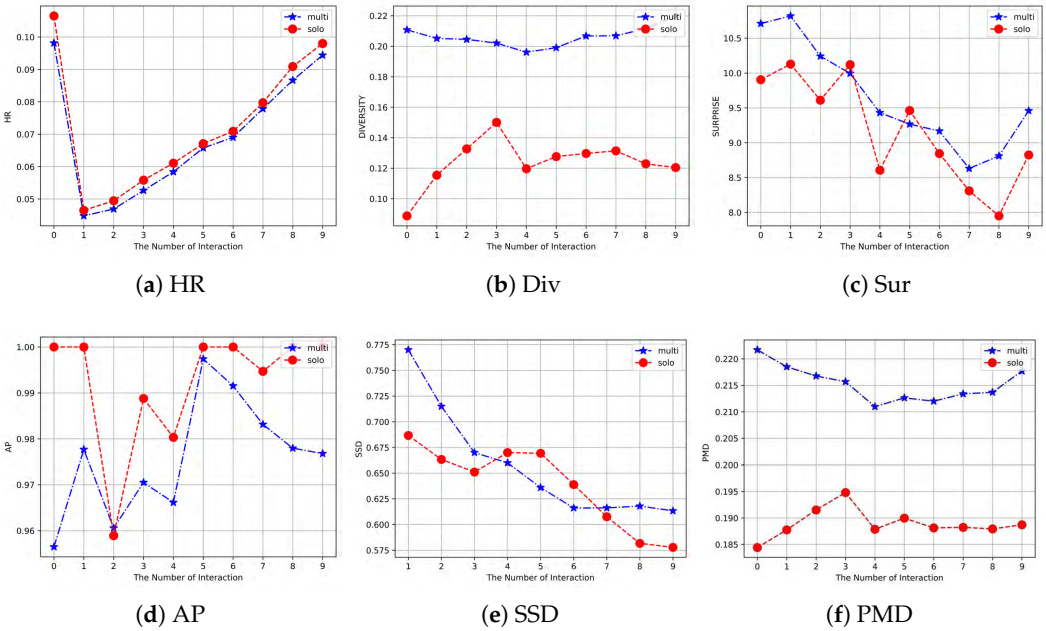
In Figure 5, three scenarios emerge in terms of predicting rating results: Better performance from MPUSMs (Figure 5a), Similar performance of MPUSMs and DRBM-Social (Figure 5b), and Better performance from DRBM-Social (Figure 5c). It can be seen that the accuracy of MPUSMs rating is not directly related to the vector dimension size. The better performance of MPUSMs is due to the fact that it is constructed by dividing user preferences in a manner similar to an attention mechanism. In this way, we can simulate the

behavior of users who evaluate items based on only a few of their preferences. Conversely, MPUSMs do not always perform better. Due to the fact that the K-means clustering may not correctly classify user's preferences or the limitations of the local surrogate model, this may sometimes lead to poor prediction. Nevertheless, in general, MPUSMs still have an accuracy comparable to the global surrogate model.

4.4. Experiment 3—Performance of Personalized Recommendation

To evaluate the performance of the proposed algorithm for personalized recommendation results, the TOP-N lists generated from 10 interactions are comprehensively evaluated using three types of evaluation metrics. This experiment uses the same dataset as Experiment 2 and compares it with the SC-DRBMIEDA algorithm, and the two algorithms are run 30 times independently. The biggest difference between the two algorithms is that the excellent initial population of the proposed algorithm evolves further to mine preference knowledge, while the population of SC-DRBMIEDA does not have this process.

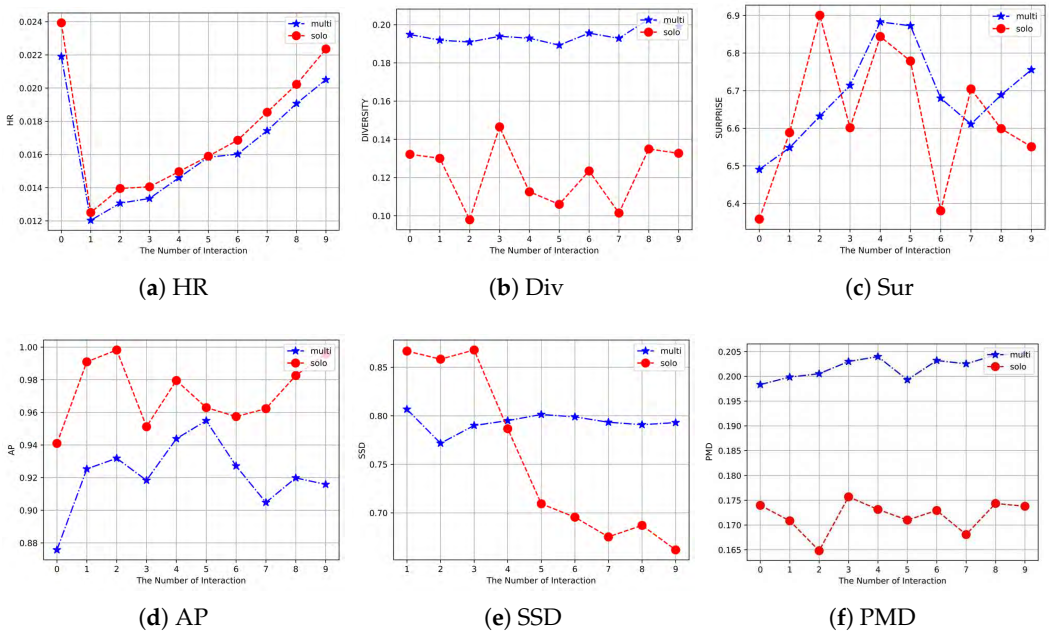
In comparison with it, the advantage of migrating inter-preference knowledge is easiest to manifest. In particular, to ensure the comparability of the two algorithms, the population size of SC-DRBMIEDA is the same as the size of the pre-recommended list, except that the model parameters are the same as in Experiment 2. That is, the source size of the TOP-N list is the same. Figures 6–8 depict the performance of the two personalized recommendation algorithms on Music, CDs, and Kindle datasets, respectively. A summary of the combined performance of each metric on the ten interaction results can be found in Table 5, where IMFEOA and IEDA stand for MPUSMs-IMFEOA and SC-DRBMIEDA, respectively. Best and Ave are the maximum and average of the ten interaction results, Growth represents the average percentage improvement of the proposed algorithm over SC-DRBMIEDA, and Total denotes the average growth rates of the combined 3 datasets. Bolded values in the table indicate higher performance. In this experiment, the Mann–Whitney U nonparametric test with a confidence level of 0.95 is used to demonstrate that the proposed algorithms are significantly different from each other, and the mark '*' indicates that the algorithm is significantly different from other algorithms.



(**a**) HR (**b**) Div (**c**) Sur

(**d**) AP (**e**) SSD (**f**) PMD

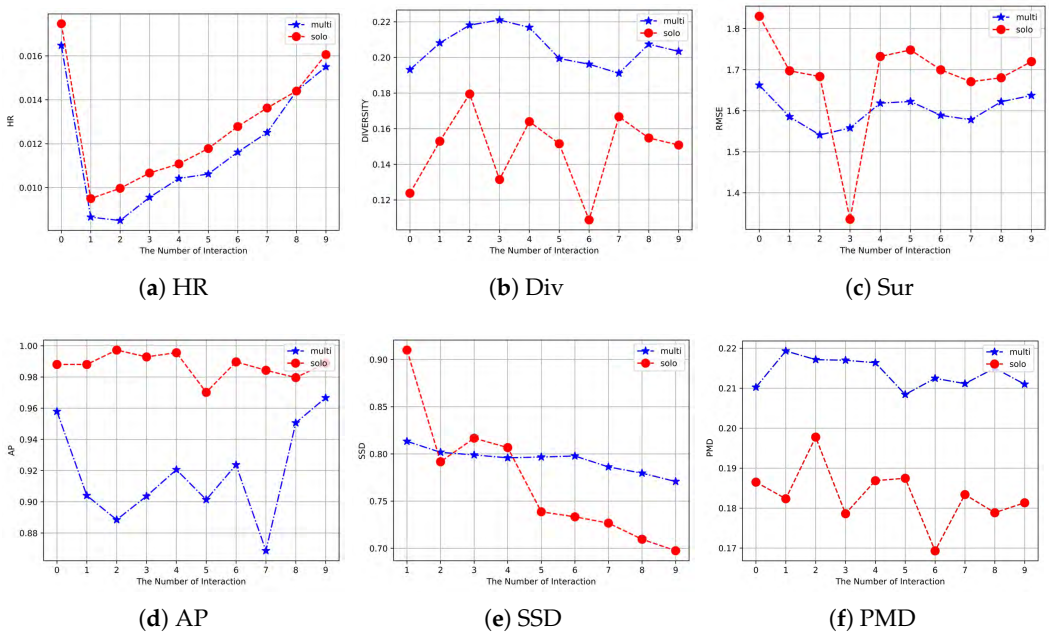
Figure 6. The recommendation performance of the two algorithms on six metrics when performing interactions in the Music dataset.



(**a**) HR (**b**) Div (**c**) Sur

(**d**) AP (**e**) SSD (**f**) PMD

Figure 7. The recommendation performance of the two algorithms on six metrics when performing interactions in the CD dataset.



(**a**) HR (**b**) Div (**c**) Sur

(**d**) AP (**e**) SSD (**f**) PMD

Figure 8. The recommendation performance of the two algorithms on six metrics when performing interactions in the Kindle dataset.

Combining Figures 6–8, the following conclusions could be drawn. For one, the MPUSMs-IMFEOA algorithm consistently excels in *Div* metric, and the curve is smoother. This indicates the effective improvement of the individual diversity and its stability of recommendation results by dividing preferences through MPUSMs and selecting appropriate genetic operators. It also suggests that the algorithm improves the search ability for each preference. Second, as shown by the performance of *SSD*, the temporal diversity of SC-DRBMIEDA decreases significantly with the reduction of the project pool size, while the proposed MPUSMs-IMFEOA is more likely to maintain temporal diversity and is less affected by the size of the project pool. Third, the proposed algorithm consistently performs

better on the PMD metric. The significant improvement in PMD implies that the algorithm is significantly more capable of mining unknown user preferences. It indicates that knowledge transfer between preferences can effectively facilitate the discovery of entirely new items.

Table 5. Combined evaluation results of six metrics for both algorithms.

Metric \ Daraset	Music					CDs					Kindle					Total
	IMFEOA		IEDA		Growth	IMFEOA		IEDA		Growth	IMFEOA		IEDA		Growth	
	Best	Ave	Best	Ave		Best	Ave	Best	Ave		Best	Ave	Best	Ave		
HR	0.098	0.069	**0.106**	**0.073** *	−4.33%	0.022	0.016	**0.024**	**0.017** *	−5.47%	0.016	0.012	**0.017**	**0.013** *	−7.14%	−5.65%
AP	0.997	0.976	**1**	**0.992** *	−1.66%	0.955	0.922	**0.998**	**0.972** *	−5.19%	0.967	0.919	**0.997**	**0.987** *	−6.98%	−4.61%
Div	**0.215**	**0.203** *	0.15	0.124	**63.98%**	**0.203**	**0.194** *	0.147	0.122	**59.65%**	**0.221**	**0.205** *	0.179	0.148	**38.42%**	**54.02%**
SSD	**0.77**	**0.591**	0.687	0.575	**2.93%**	0.807	**0.714** *	**0.868**	0.681	**4.88%**	0.813	**0.714** *	**0.91**	0.693	**3.30%**	**3.70%**
Sur	**10.938**	**9.686** *	10.128	9.175	**5.57%**	6.883	**6.687**	**6.9**	6.631	**0.86%**	**6.659**	**6.512** *	6.55	6.406	**1.65%**	**2.69%**
PMD	**0.222**	**0.215** *	0.195	0.183	**13.96%**	**0.205**	**0.202** *	0.176	0.172	**17.51%**	**0.219**	**0.214** *	0.198	0.183	**16.68%**	**16.05%**

1 The mark '*' indicates that the algorithm is significantly different from other algorithms. 2 Bolded values in the table indicate higher performance.

As shown in Table 5, the MPUSMs-IMFEOA has a lower accuracy according to the Ave of HR and AP metrics. Even so, the MPUSM-IMFEOA algorithm is still sufficient to recommend desired items. This is directly reflected in the AP values above 0.9 for all three datasets, as the AP values can laterally reflect the number of satisfying items of users in the recommendation list. Furthermore, the proposed algorithm performs poorly in HR and AP metrics for two reasons. One may be due to the reduced scoring accuracy of the MPUSMs. As seen in Figure 5, MPUSMs have the best accuracy in the Music dataset, and MPUSMs-IMFEOA also has the least degradation in the performance of HR and AP metrics in the Music dataset, as reflected in the Growth of the three datasets in Table 5. The other may be caused by the introduction of low-rated individuals into the pre-recommended list resulting in an insufficient number of competent individuals. In other words, the distribution of evolved population will be somewhat different from the distribution of actual excellent items. For example, Figure 5a indicates that MPUSMs perform better than DRBM-Social on $RMSE$, but proposed algorithm performance on HR and AP metrics is worse.

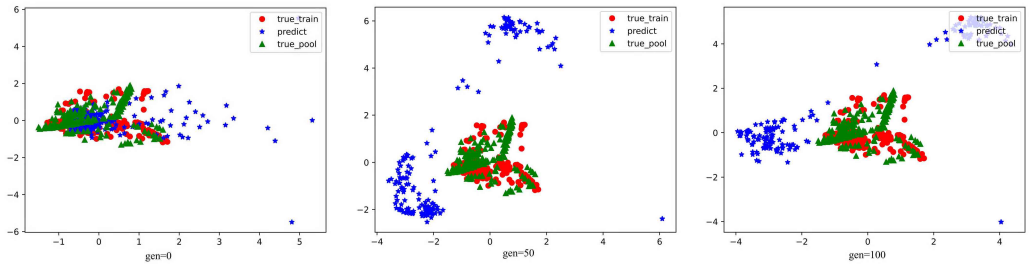
From the Growth rate of SSD in Table 5, it can be seen that the proposed algorithm shows a slight improvement in each dataset, which further demonstrates that MPUSMs' interactive dynamic update improves its ability to avoid homogeneous recommendations. Similarly, the improvement in Sur indicates that MPUSMs-IMFEOA is more likely to uncover items that are not similar to users' historical preferences but that they are satisfied with. It not only confirms the accuracy of MPUSMs, but also reveals the algorithm's ability to explore unknown items at the same time.

Overall, it can be seen from the Total values in Table 5 that, at the expense of 5% accuracy, MPUSMs-IMFEOA simultaneously improves the diversity and novelty of the recommended results. In addition, the reduction of HR and AP is limited and does not actually have a negative impact on the user experience. Hence, MPUSMs-IMFEOA is capable of guaranteeing the accuracy of recommendations, exploiting knowledge among preferences to improve diversity and novelty and ultimately bringing the user a more satisfying personalized recommendation experience.

4.5. Experiment 4—Search Performance of MPUSMs-IMFEOA

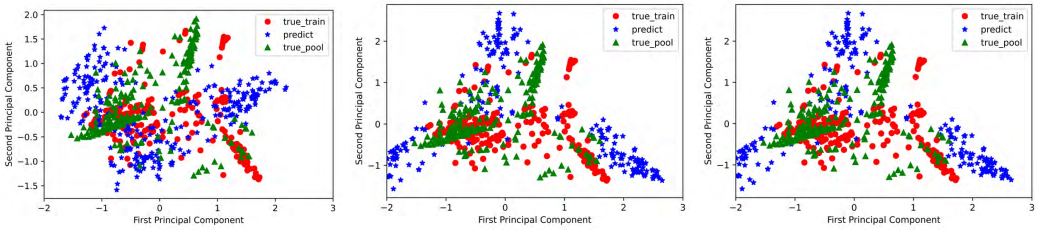
To demonstrate the search performance produced by the proposed IMFEOA in conjunction with the MPUSMs, the distribution of individuals in the population evolution process is observed (Figures 9–12). This experiment is validated by changing the underlying surrogate models of MPUSMs and the evolutionary algorithm. This experiment is validated by changing the underlying surrogate models of MPUSMs and the evolutionary algorithm. That is, single and multi-model guided population evolution under different underlying surrogate models exhibits performance suitable for personalized recommendations. Therefore, introduce Restricted Boltzmann Machine (RBM) [42] to build surrogates

and Genetic Algorithm (GA) [30] to optimize the population. This is because a surrogate model built with an RBM is equally easy to calculate a probabilistic model such that the starting point of the initial population being evolved is the same. In addition, a genetic algorithm can use the same crossover and mutation operators to optimize populations, suitable for comparison with MFEA.



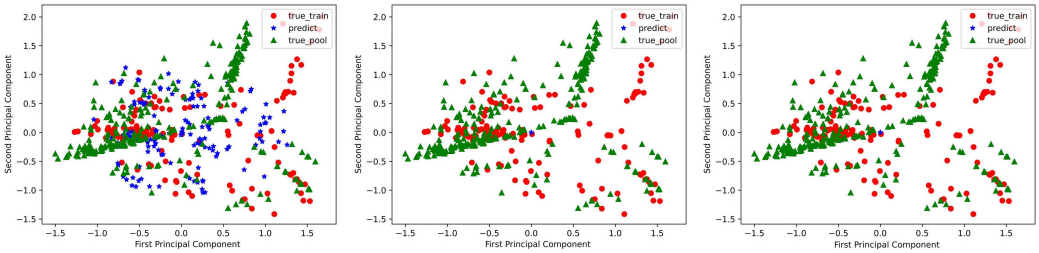
(**a**) gen = 0 (**b**) gen = 50 (**c**) gen = 100

Figure 9. Population evolution of DRBM-Social-IGA based on the DRBM-Social surrogate model.



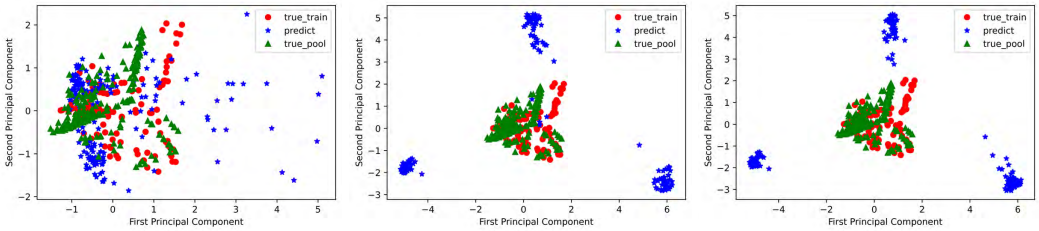
(**a**) gen = 0 (**b**) gen = 50 (**c**) gen = 100

Figure 10. Population evolution of MPUSMs-IMFEOA based on the DRBM-Social surrogate model.



(**a**) gen = 0 (**b**) gen = 50 (**c**) gen = 100

Figure 11. Population evolution of RBM-IGA based on the RBM surrogate model.



(**a**) gen = 0 (**b**) gen = 50 (**c**) gen = 100

Figure 12. Population evolution of MPUSMs-IMFEOA based on the RBM surrogate model.

Specifically, there are MPUSMs-IMFEOA and DRBM-Social-IGA based on the DRBM-Social surrogate model, and MPUSMs-IMFEOA and RBM-IGA based on the RBM surrogate model. To make the IMFEOA and IGA algorithms comparable, they both use the same crossover operator and mutation operator and set the same crossover probability ($p = 1/3 + 2/3 * rmp$) and mutation probability ($1 - p$) for IGA as for IMFEOA. In addition, the visible/hidden layer units of the RBM are the same as those of the Dual-RBM, and

probability models are set for them. The dataset is the same as Experiment 1. The true-train in the legend is the real historical items, and the true-pool denotes the items in the item pool, and the predict indicates the evolving individuals.

As can be seen from Figures 9 and 10, the MPUSMs-IMFEOA based on the DRBM-Social surrogate model is more likely to keep the population distribution similar to the actual item distribution, ensuring that the produced individuals are valid. From Figures 11 and 12, the evolved population in Figure 12 has more unduplicated individuals, indicating that the IMFEOA is more likely to maintain population diversity. Comparing Figures 10 and 12 shows that they differ only in the surrogate model, but both have a richly diverse population. It further indicates that IMFEOA can effectively enhance population diversity. However, MPUSMs-IMFEOA based on the RBM surrogate model is insufficient to maintain the correct true distribution of populations, suggesting that the performance of MPUSMs-IMFEOA is also highly dependent on the surrogate model.

5. Discussion and Conclusions

In this article, we introduce the knowledge transfer method into the interactive evolutionary algorithm for the first time to improve the diversity and novelty of personalized recommendations. With the aim of this, a new IEA (MPUSMs-IMFEOA) is proposed based on a modified MFEA and deep learning-driven user preference surrogate models.

Firstly, to describe preference information from different perspectives, MPUSMs and partial-MPUSMs are first developed. They are used to evaluate unknown items at different population evolutionary stages, and partial-MPUSMs are the objectives to be simultaneously optimized by MFEA for achieving knowledge transfer between preferences. Then, probability models of MPUSMs are constructed to generate initial populations with skill factors to improve the evolutionary efficiency of MFEA.

Secondly, a new IEA method based on MPUSMs and its derived models and joint multitasking learning is proposed. Accordingly, improved MFEA, recommendation lists generation method, and interaction-assisted model management method are designed for the characteristics of the models mentioned above.

Finally, comprehensive empirical studies are conducted in the Amazon dataset. Experimental results show that the proposed algorithm can increase diversity and novelty while maintaining accuracy, which demonstrates the effectiveness of building multiple surrogate models to facilitate the mining of inter-preference knowledge. Specifically, the MPUSMs ensure the accuracy of its evaluation results by conforming to the characteristic that users rely only on partial preferences to evaluate items. In addition, with a loss of only about 5% recommendation accuracy (HR and AP), individual diversity (Div) improved by 54.02%, temporal diversity (SSD) by 3.7%, surprise degree (Sur) by 2.69%, and preference mining degree (PMD) by 16.05%. Comparison with other evolutionary algorithms also shows that the combination of MPUSMs and its derived models with IMFEOA makes it easier to mine user's unknown preferences, and the search performance of IMFEOA is highly dependent on the underlying surrogate model.

MPUSMs-IMFEOA also has certain limitations. In terms of surrogate models, their model parameters cannot be automatically adjusted. As shown in Experiment 1, the accuracy of the models shows a difference in performance on different datasets with the same parameters. Although they can provide sufficient recommendation accuracy, it still has impact on the final recommendation results. In addition, the classification method of the data of the training models may not fit the user' habits. At the algorithmic level, IMFEOA may unearth items that are not actually possible to exist, and thus the corresponding items found in the item pool using the shortest distance method are equally inaccurate. Therefore, the search area of IMFEOA needs to be adjusted.

In the future, we hope to optimize three aspects under the framework of the MPUSMs-IMFEOA algorithm. The first is to apply other user surrogate models to the MPUSMs to address issues such as model accuracy, cold start, and inappropriate preference classification. Secondly, further improving the efficiency and effectiveness of knowledge transfer is a

research direction. Finally, it is possible to study how to update the model more effectively during the interaction.

Author Contributions: Conceptualization, W.W., X.S. and L.B.; Data curation, G.M.; Formal analysis, W.W.; Investigation, W.W.; Methodology, W.W.; Resources, L.B.; Software, W.W.; Supervision, X.S.; Validation, G.M. and S.L.; Visualization, S.L.; Writing—original draft, W.W.; Writing—review and editing, X.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, Grant No. 61876184.

Data Availability Statement: The data presented in this study are openly available in Amazon Review Data at https://nijianmo.github.io/amazon/index.html, accessed on 28 December 2021, reference number [41].

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Yu, M.; He, W.; Zhou, X.; Cui, M.; Wu, K.; Zhou, W. Review of recommendation systems. *Comput. Appl.* **2021**, 1–16
2. Batmaz, Z.; Yurekli, A.; Bilge, A.; Kaleli, C. A review on deep learning for recommender systems: Challenges and remedies. *Artif. Intell. Rev.* **2019**, *52*, 1–37. [CrossRef]
3. Huang, W.; Wu, J.; Song, W.; Wang, Z. Cross attention fusion for knowledge graph optimized recommendation. *Appl. Intell.* **2022**, *52*, 10297–10306. [CrossRef]
4. Khanal, S.S.; Prasad, P.; Alsadoon, A.; Maag, A. A systematic review: Machine learning based recommendation systems for e-learning. *Educ. Inf. Technol.* **2020**, *25*, 2635–2664. [CrossRef]
5. Bao, L.; Sun, X.; Gong, D.; Zhang, Y. Multi-Source Heterogeneous User Generated Contents-Driven Interactive Estimation of Distribution Algorithms for Personalized Search. *IEEE Trans. Evol. Comput.* **2021**, *26*, 844–858. . [CrossRef]
6. Chen, Y.; Sun, X.; Gong, D.; Yao, X. DPM-IEDA: Dual Probabilistic Model Assisted Interactive Estimation of Distribution Algorithm for Personalized Search. *IEEE Access* **2019**, *7*, 41006–41016. [CrossRef]
7. Gabor, T.; Altmann, P. Benchmarking Surrogate-Assisted Genetic Recommender Systems. In *Proceedings of the 2019 Genetic and Evolutionary Computation Conference Companion (Geccco'19 Companion)*; Assoc Computing Machinery: New York, NY, USA, 2019; pp. 1568–1575. [CrossRef]
8. Cai, H. User Preference Adaptive Fitness of Interactive Genetic Algorithm Based Ceramic Disk Pattern Generation Method. *IEEE Access* **2020**, *8*, 95978–95986. [CrossRef]
9. Trachanatzi, D.; Rigakis, M.; Marinaki, M.; Marinakis, Y. An Interactive Preference-Guided Firefly Algorithm for Personalized Tourist Itineraries. *Expert Syst. Appl.* **2020**, *159*, 113563. [CrossRef]
10. Aghaei Pour, P.; Rodemann, T.; Hakanen, J.; Miettinen, K. Surrogate Assisted Interactive Multiobjective Optimization in Energy System Design of Buildings. *Optim. Eng.* **2022**, *23*, 303–327. [CrossRef]
11. Li, G.; Zhang, Q. Multiple Penalties and Multiple Local Surrogates for Expensive Constrained Optimization. *IEEE Trans. Evol. Comput.* **2021**, *25*, 769–778. [CrossRef]
12. Ji, X.; Zhang, Y.; Gong, D.; Sun, X. Dual-Surrogate-Assisted Cooperative Particle Swarm Optimization for Expensive Multimodal Problems. *IEEE Trans. Evol. Comput.* **2021**, *25*, 794–808. [CrossRef]
13. Bao, L.; Sun, X.; Chen, Y.; Gong, D.; Zhang, Y. Restricted Boltzmann Machine-Driven Interactive Estimation of Distribution Algorithm for Personalized Search. *Knowl.-Based Syst.* **2020**, *200*, 106030. [CrossRef]
14. Liang, Z.; Zhu, Y.; Wang, X.; Li, Z.; Zhu, Z. Evolutionary Multitasking for Multi-objective Optimization Based on Generative Strategies. *IEEE Trans. Evol. Comput.* **2022**. [CrossRef]
15. Chen, Z.; Zhou, Y.; He, X.; Zhang, J. Learning task relationships in evolutionary multitasking for multiobjective continuous optimization. *IEEE Trans. Cybern.* **2020**, *52*, 5278–5289. [CrossRef]
16. Gupta, A.; Ong, Y.S.; Feng, L. Multifactorial Evolution: Toward Evolutionary Multitasking. *IEEE Trans. Evol. Comput.* **2016**, *20*, 343–357. [CrossRef]
17. Lin, J.; Liu, H.L.; Tan, K.C.; Gu, F. An Effective Knowledge Transfer Approach for Multiobjective Multitasking Optimization. *IEEE Trans. Cybern.* **2021**, *51*, 3238–3248. [CrossRef]
18. Zhou, L.; Feng, L.; Tan, K.C.; Zhong, J.; Zhu, Z.; Liu, K.; Chen, C. Toward Adaptive Knowledge Transfer in Multifactorial Evolutionary Computation. *IEEE Trans. Cybern.* **2021**, *51*, 2563–2576. [CrossRef]
19. Xu, Q.; Wang, N.; Wang, L.; Li, W.; Sun, Q. Multi-Task Optimization and Multi-Task Evolutionary Computation in the Past Five Years: A Brief Review. *Mathematics* **2021**, *9*, 864. [CrossRef]
20. Van Dat, N.; Van Toan, P.; Thanh, T.M. Solving distribution problems in content-based recommendation system with gaussian mixture model. *Appl. Intell.* **2022**, *52*, 1602–1614. [CrossRef]

21. Khojamli, H.; Razmara, J. Survey of similarity functions on neighborhood-based collaborative filtering. *Expert Syst. Appl.* **2021**, *185*, 115482. [CrossRef]
22. Song, C.; Yu, Q.; Jose, E.; Zhuang, J.; Geng, H. A Hybrid Recommendation Approach for Viral Food Based on Online Reviews. *Foods* **2021**, *10*, 1801. [CrossRef] [PubMed]
23. Yuan, W.; Wang, H.; Hu, B.; Wang, L.; Wang, Q. Wide and deep model of multi-source information-aware recommender system. *IEEE Access* **2018**, *6*, 49385–49398. [CrossRef]
24. Wang, D.; Xu, D.; Yu, D.; Xu, G. Time-aware sequence model for next-item recommendation. *Appl. Intell.* **2021**, *51*, 906–920. [CrossRef]
25. Zeng, D.; He, M.; Zhou, Z.; Tang, C. An Interactive Genetic Algorithm with an Alternation Ranking Method and Its Application to Product Customization. *Hum.-Centric Comput. Inf. Sci.* **2021**, *11*. [CrossRef]
26. Dou, R.; Zhang, Y.; Nan, G. Application of Combined Kano Model and Interactive Genetic Algorithm for Product Customization. *J. Intell. Manuf.* **2019**, *30*, 2587–2602. [CrossRef]
27. Zhu, X.; Li, X.; Chen, Y.; Liu, J.; Zhao, X.; Wu, X. Interactive Genetic Algorithm Based on Typical Style for Clothing Customization. *J. Eng. Fibers Fabr.* **2020**, *15*, 1558925020920035. [CrossRef]
28. Takagi, H. Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation. *Proc. IEEE* **2001**, *89*, 1275–1296. [CrossRef]
29. Tomczyk, M.K.; Kadziński, M. Decomposition-based interactive evolutionary algorithm for multiple objective optimization. *IEEE Trans. Evol. Comput.* **2019**, *24*, 320–334. [CrossRef]
30. Gong, D.; Yuan, J.; Sun, X. Interactive Genetic Algorithms with Individual's Fuzzy Fitness. *Comput. Hum. Behav.* **2011**, *27*, 1482–1492. [CrossRef]
31. Sun, X.; Lu, Y.; Gong, D.; Zhang, K. Interactive genetic algorithm with CP-nets preference surrogate and application in personalized search. *Control Decis.* **2015**, *30*, 1153–1161.
32. Funaki, R.; Sugimoto, K.; Murata, J. Estimation of Influence of Each Variable on User's Evaluation in Interactive Evolutionary Computation. In Proceedings of the 2018 9th International Conference on Awareness Science and Technology (iCAST), Fukuoka, Japan, 19–21 September 2018; pp. 167–174. [CrossRef]
33. Alizadeh, V.; Kessentini, M.; Mkaouer, M.W.; Ocinneide, M.; Ouni, A.; Cai, Y. An Interactive and Dynamic Search-Based Approach to Software Refactoring Recommendations. *IEEE Trans. Softw. Eng.* **2020**, *46*, 932–961. [CrossRef]
34. Chen, Y.; Sun, X.; Gong, D.; Zhang, Y.; Choi, J.; Klasky, S. Personalized Search Inspired Fast Interactive Estimation of Distribution Algorithm and Its Application. *IEEE Trans. Evol. Comput.* **2017**, *21*, 588–600. [CrossRef]
35. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
36. Gupta, A.; Ong, Y.S. Back to the Roots: Multi-X Evolutionary Computation. *Cogn. Comput.* **2019**, *11*, 1–17. [CrossRef]
37. Bali, K.K.; Ong, Y.S.; Gupta, A.; Tan, P.S. Multifactorial Evolutionary Algorithm With Online Transfer Parameter Estimation: MFEA-II. *IEEE Trans. Evol. Comput.* **2020**, *24*, 69–83. [CrossRef]
38. Gupta, A.; Zhou, L.; Ong, Y.S.; Chen, Z.; Hou, Y. Half a Dozen Real-World Applications of Evolutionary Multitasking, and More. *IEEE Comput. Intell. Mag.* **2022**, *17*, 49–66. [CrossRef]
39. Hinton, G.E. Training products of experts by minimizing contrastive divergence. *Neural Comput.* **2002**, *14*, 1771–1800. [CrossRef]
40. Jiang, S.; Song, J. Evaluation Metrics for Personalized Recommendation Systems. *J. Phys. Conf. Ser.* **2021**, *1920*, 012109. [CrossRef]
41. Ni, J.; Li, J.; McAuley, J. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*; Association for Computational Linguistics: Hong Kong, China, 2019; pp. 188–197.
42. Bao, L.; Sun, X.; Chen, Y.; Man, G.; Shao, H. Restricted Boltzmann Machine-Assisted Estimation of Distribution Algorithm for Complex Problems. *Complexity* **2018**, *2018*, 2609014. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.