Research Frontier

Yi Zuo, Maoguo Gong, Jiulin Zeng,
Lijia Ma, and Licheng Jiao
*Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, Xidian University, Xi'an, China*

# Personalized Recommendation Based on Evolutionary Multi-Objective Optimization

**Abstract**

Traditional recommendation techniques in recommender systems mainly focus on improving recommendation accuracy. However, personalized recommendation, which considers the multiple needs of users and can make both accurate and diverse recommendations, is more suitable for modern recommender systems. In this paper, the task of personalized recommendation is modeled as a multi-objective optimization problem. A multi-objective recommendation model is proposed. The proposed model maximizes two conflicting performance metrics termed as accuracy and diversity. The accuracy is evaluated by the probabilistic spreading method, while the diversity is measured by recommendation coverage. The proposed MOEA-based recommendation method can simultaneously provide multiple recommendations for multiple users in only one run. Our experimental results demonstrate the effectiveness of the proposed algorithm. Comparison experiments also indicate that the proposed algorithm can make more diverse yet accurate recommendations.

## I. Introduction

With the rapid development of science and technology, we human beings have entered an era of information explosion. People are inundated with enormous amount of information nowadays. Accordingly, it becomes an urgent problem to find out useful information for us efficiently. Recommender systems (RSs) [1], which use statistical and knowledge discovery techniques to provide recommendations automatically, are considered to be the most promising tools to alleviate the overload of information. Ever since their advent, RSs have attracted considerable attention in both theoretical research and practical applications [2]. Researches on RSs cover various topics, including movies [3], books [4], songs [5], jokes [6], tourism [7], web search [8], and so on. Moreover, RSs are now a key component of many e-commerce sites, such as Youtube.com, Yahoo.com and Amazon.com.

IMAGE LICENSED BY INGRAM PUBLISHING

Usually, the major aim of traditional RSs is to maximize accuracy as much as possible in predicting items which are likely be appreciated by a particular user. For example in October 2006, the online DVD rental company Netflix announced the Netflix Prize [9], a competition for movie recommendation. The competition challenged researchers to develop RSs that could beat the company's RS, Cinematch in accuracy. The grand prize of $1,000,000 was awarded to the winner of the contest, whose recommendation accuracy was 10% higher than that of Cinematch. However, as discussed in recent studies [10]–[14], only considering the accuracy of recommendations may not be enough to suggest the most relevant items to users. Other performance metrics, such as the diversity, should also be taken into account to meet users' multiple requirements. It can be observed that a high accuracy of recommendations can be easily obtained by safely recommending popular items to users [15]. However, it will undoubtedly lose recommendation diversity. Likewise, to recommend diverse items to users may lead to a decrease in recommendation accuracy, since diversity and accuracy are two conflicting metrics for RSs. As a consequence, a pressing challenge for RSs is how to develop personalized recommendation techniques that can generate recommendations with both high accuracy and diversity.

To achieve a proper balance between accuracy and diversity, a variety of recommendation techniques have been developed. Zhang et al. [14] modeled the tradeoff between accuracy and diversity as a quadratic programming problem and developed several strategies to solve this

*Digital Object Identifier 10.1109/MCI.2014.2369894*
*Date of publication: 14 January 2015*



1556-603X/15©2015IEEE

optimization problem. A control parameter should be used to determine the importance of diversification in the recommendation lists. Zhou et al. [16] proposed a hybrid recommendation algorithm, which combines Heat-spreading (HeatS) algorithm specifically to address the challenge of diversity and probabilistic spreading (ProbS) algorithm to focus on accuracy. Note that the hybrid algorithm is produced by using a basic weighted linear aggregation method. As a result, the weight parameter should be appropriately tuned so as to make accurate and diverse recommendations. Adomavicius et al. [15], developed a number of item ranking techniques to generate diverse recommendations while maintaining comparable levels of recommendation accuracy.

In this paper, we propose a general multi-objective recommendation model to address the challenge of striking a balance between accuracy and diversity. The model considers two conflicting objectives. The first one is the measurement of recommendation accuracy, which can be estimated by an accuracy-based recommendation technique, and the other one is a diversity metric. The task of personalized recommendation is thus modeled as a multi-objective optimization problem (MOP). A multi-objective evolutionary algorithm (MOEA) is then performed to evolve the population to maximize these two objectives. Finally, a set of different recommendations can be provided for users.

ProbS [17], as a simple yet effective recommendation technique, is adopted as the accuracy estimator in this paper. Coverage, which measures the ability of a recommendation algorithm to suggest distinct items. Low coverage indicates that only a small fraction of items in the system are recommended, whereas high coverage means that the algorithm is more likely to suggest diverse recommendations [18]. From this viewpoint, coverage can be considered as a diversity metric [19]. NSGA-II [20] is applied to optimize the two conflicting objectives simultaneously so as to make recommendations to users. For convenience, the proposed MOEA-based recommendation algorithm is termed as MOEA-ProbS. Note that recommendations to multiple users are encoded in one individual. Therefore, multiple recommendations can be provided simultaneously in one run for multiple users. To reduce the computational complexity, a clustering technique is firstly introduced to divide users into several clusters. Then, the proposed algorithm can simultaneously provide recommendations for all users in each cluster. To investigate the performance of the proposed MOEA-ProbS, we will compare it with some widely used recommendation techniques.

The main contributions of this paper are as follows.

1) A general multi-objective recommendation model is proposed to balance recommendation accuracy and diversity.
2) Different from traditional recommendation techniques, the proposed algorithm can simultaneously provide multiple recommendations for multiple users in only one run.
3) Experimental results show that the proposed algorithm can provide a set of diverse and accurate recommendations. Specially, the coverage of recommendations is greatly improved, which is a promising property for RSs.
4) A clustering technique is employed to improve the computational efficiency.

The remainder of this paper is organized as follows. In Section II, some backgrounds including the problem definition of recommendation, some preliminaries of multi-objective optimization, the related work on RSs and the introduction to ProbS are presented. Section III describes the proposed MOEA-based recommendation algorithm in detail. In Section IV, experimental studies are presented. Finally, conclusions are given in Section V.

## II. Background

### A. Problem Definition of Recommendation

Generally, the problem of recommendation can be formalized as follows. Assume that set *Users* contains all the users in a system, and set *Items* contains all possible items that can be recommended. A rating matrix $R$ is used to measure the preferences of users to items. For instance, the preference of a user $i \in Users$ to an item $\alpha \in Items$ is measured by $R(i,\alpha)$, which is often a non-negative integer or a real number within a certain range [21]. Usually, each user rates few items and each item is rated by few users in practice, so the rating matrix $R$ is rather sparse. The first step of recommendation techniques is to predict the unknown ratings in $R$. Then, items will be recommended to users based on the obtained ratings. More specifically, one or a set of items $\alpha$ that maximize $R(i,\alpha)$ will be selected as the recommendation to user $i$ i.e.,

$$\forall i \in \text{Users}, \alpha = \arg\max_{\alpha \in \text{Items}} R(i,\alpha). \quad (1)$$

In most RSs, only a single-criterion value is considered. However, the preference of a particular user may depend on more than one criterion. The additional information provided by multi-criteria ratings can improve the quality of recommendations [22].

### B. Multi-Objective Optimization

Multi-objective optimization is to optimize a vector of functions [23]

$$\min F(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))^T \quad (2)$$

where $\boldsymbol{x} = [x_1, x_2, \ldots, x_d] \in \Omega$ is called the decision vector, and $\Omega$ is the $D$-dimensional decision space.

Without loss of generality, we consider the minimization problem as in (2), since the maximization problem can be easily transformed into the minimized form. Given two decision vectors $\boldsymbol{x}_A \in \Omega$ and $\boldsymbol{x}_B \in \Omega$, it is said that $\boldsymbol{x}_A$ dominates $\boldsymbol{x}_B$ (written as $\boldsymbol{x}_A \succ \boldsymbol{x}_B$ if $f_i(\boldsymbol{x}_A) \le f_i(\boldsymbol{x}_B)$ for all $i = 1, 2, \ldots, m$, and $F(\boldsymbol{x}_A) \neq F(\boldsymbol{x}_B)$.

A vector of decision variables $\boldsymbol{x} \in \Omega$ is called a Pareto-optimal solution if there is no $\boldsymbol{x}^* \in \Omega$ such that $\boldsymbol{x}^* \succ \boldsymbol{x}$.

The set of all the Pareto optimal solutions is called the Pareto set, which can be defined as

$$PS = \{\boldsymbol{x} \in \Omega \mid \neg\exists \boldsymbol{x}^* \in \Omega, \boldsymbol{x}^* \succ \boldsymbol{x}\}. \quad (3)$$

The image of the Pareto set under the objective function space is called the Pareto front, defined as

$$PF = \{F(\boldsymbol{x}) | \boldsymbol{x} \in PS\}. \tag{4}$$

The goal of an MOEA is to find a set of non-dominated solutions approximating the true Pareto front.

### C. Related Work on RSs

As described in [21], recommendation techniques can be classified into three major types: content-based filtering, collaborative filtering (CF) and hybrid methods. Content based filtering methods [24] recommend items to a target user based on the content of items already preferred by the target user. Thus, items with the most similar content will be suggested to the target user. Unlike content-based methods, CF methods [25] do not require any content information. They recommend items to a given user based on information provided by those users having similar preferences with the given user. Hybrid methods combine different recommendation techniques in order to take advantages of them. A plenty of hybrid recommendation methods have been proposed and proved to produce better results in real applications [26]–[28]. A survey focused on hybrid RSs can be found in [29].

Although many efforts have been dedicated to generating accurate RSs [30]–[33], some studies [10]–[12] have shown that other metrics, such as the diversity, contribute as importantly as the accuracy to the success of RSs. Some new recommendation methods were developed to increase individual diversity [13], [14], which is measured by an average dissimilarity between all pairs of recommended items to a given individual user. In contrast to individual diversity, aggregate diversity of recommendations across all users was also investigated [15], [34], [35]. In [36], the notion of "topic diversification" was introduced to balance the diversity of recommendations across different topics. A different measure of item diversity was proposed to assess the extent to which the same items are recommended to users over and over again [37]. In [38], the authors designed a novelty measure, which assumes that an item with lower popularity is deemed to have higher novelty. By hypothesizing that the degree of user's surprise is proportional to the estimated time used for searching the item, a new metric for item novelty was introduced in [39]. A thorough review of extensive measures used for evaluating RSs is presented in [40].

Real-world optimization problems often involve a number of characteristics, some of which may be conflicting, resulting in MOPs. So far, there have been a variety of mathematical programming techniques that can be used to solve MOPs [23]. However, these techniques may have several limitations [41]. For example, most of them require the differentiability of the objective functions and the constraints. In addition, many of them are susceptible to the shape of the Pareto front of the MOP. They may be invalid when the MOP has a concave or disconnected Pareto front. Moreover, they usually generate only one solution from one run. Therefore, many runs are necessary to generate multiple solutions. Different from traditional optimization techniques, MOEAs can generate many Pareto optimal solutions in one single run. Also, they are less susceptible to the shape or continuity of the Pareto front, and can be used for solving MOPs without good mathematical properties. Since the pioneering work of Schaffer [42], a number of MOEAs have been developed and applied in many fields [43]. The typical representatives of MOEAs include Strength Pareto Evolutionary Algorithm (SPEA) [44] and its improved version (SPEA2) [45], Non-dominated Sorting Genetic Algorithm (NSGA) [46] and its improved version (NSGA-II) [20], Multi-objective Particle Swarm Optimization (MOPSO) [47], Multi-objective Evolutionary Algorithm Based on Decomposition (MOEA/D) [48] Non-dominated Neighbor Immune Algorithm (NNIA) [49] and Hypervolume Estimation Algorithm for Multi-objective Optimization (HypE) [50].

Moreover, some new studies have been reported to use MOEAs to improve the capacity of RSs. Demir et al. [51] employed MOEAs for clustering web user sessions, and then generated web recommendations based on the obtained clusters. Their experimental results show that the use of MOEAs can improve the accuracy of recommendations. Rana et al. [52] proposed a multi-objective evolutionary clustering method based on temporal features for dynamic RSs. Tyagi et al. [53] developed a multi-objective particle swarm optimization algorithm for association rule mining in the collaborative filtering framework. Ribeiro et al. [54] designed a Pareto-efficient hybridization recommendation approach, where MOEAs are used to optimize a vector of weights assigned to different recommendation methods.

### D. ProbS

The ProbS method [17] is suitable for RSs without explicit ratings, i.e., each element $R(i,\alpha)$ in rating matrix $R$ is either 0 or 1, denoting that user $i$ has not collected or collected item $\alpha$ respectively. Explicit ratings can be easily mapped to this form, albeit losing some information in the process [19]. The whole process of ProbS is composed of two steps. First, a user-item bipartite network is constructed according to the relationship between users and items. Second, the initial resource placed on each item is equally distributed to all neighboring users, and then redistributed back to those users' neighboring items in the same way. The resource allocation process in a simple bipartite network is illustrated in Fig. 1. After two resource-distribution steps, a column normalized transition matrix can be obtained according to (5), where $M$ is the total number of users, and $k_i$ denotes the degree of item node $i$, i.e., the number of edges connected to node $i$. The element $w_{\alpha\beta}$ in this matrix represents the fraction of the initial resource of $\beta$ transferred to $\alpha$.

$$w_{\alpha\beta} = \frac{1}{k_\beta} \sum_{i=1}^{M} \frac{r_{i\alpha} r_{i\beta}}{k_i}. \tag{5}$$

Then, the ratings of items can be obtained by

$$f'_\alpha = \sum_{\beta=1}^{N} w_{\alpha\beta} f_\beta \qquad (6)$$

where $N$ is the total number of items. Finally, items are recommended to users according to the obtained ratings. To further understand ProbS, please refer to [17].

## III. The Proposed MOEA-Based Recommendation Algorithm

In order to balance recommendation accuracy and diversity, the recommendation problem is modeled as an MOP. In this paper, NSGA-II [20] is adopted as the MOEA to solve the modeled MOP, due to its robustness and effectiveness. In this section, we describe the proposed MOEA-based recommendation algorithm in detail, including the clustering method, the two objectives, the individual representation and the genetic operators.

### A. User Clustering

To reduce the computational complexity, a clustering technique is used to split a large number of users into several clusters. Since the users from different clusters have different habits and preferences, diverse recommendations can be easily provided for these users by RSs. In contrast, the users belonging to the same cluster are similar, and therefore similar items tend to be recommended to these users. The aim of the proposed algorithm is to improve the diversity of recommendations to these similar users. In particular, the quality of recommendations can be improved by using a clustering technique sometimes. Several experiments are conducted to show the impact of clustering in Section IV-C4.

As discussed in [55], there exist many clustering techniques, such as k-means, fuzzy c-means (FCM), and hierarchical clustering. In addition, community-based methods [56], which are able to mine potential relationship between different users, can be also used for clustering. In this paper, the k-means clustering method is employed. The performance of clustering will be influenced by the used similarity strategy. Here we use the cosine index [21] to measure the similarity $s_{ij}$ between two users $i$ and $j$, which is defined as

$$s_{ij} = \frac{r_i \cdot r_j}{|r_i||r_j|} \qquad (7)$$

where $r_i$ and $r_j$ are rating vectors given by $i$ and $j$ on items, respectively.

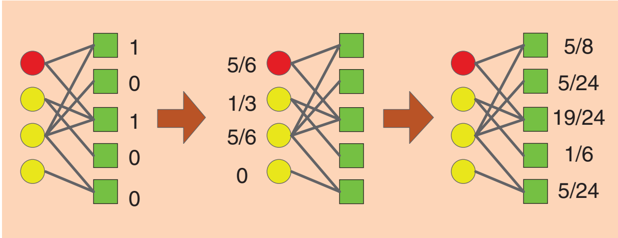


**FIGURE 1** Illustration of the process of ProbS in a simple bipartite network. The squares denote items and the circles denote users. The target user is denoted by the red circle.

### B. The Two Objectives

In this paper, two conflicting objectives are considered. The first one measures the accuracy of recommendations. In fact, it is impossible to compute the true preferences of users in the training stage. Therefore, the estimated ratings of items are used. For a user $i$ and an item $\alpha$ the predicted rating of $\alpha$ given by $i$ is $pr_{i\alpha}$. For all the users in one cluster $S$, the predicted rating is defined as:

$$PR = \frac{\sum_{i \in S} \sum_{\alpha=1}^{L} pr_{i\alpha}}{|S| \times L} \qquad (8)$$

where $|S|$ is the number of users in $S$, and $L$ is the length of the recommendation list.

The second one is to measure the diversity of recommendations. There are several diversity metrics [19], such as inter-user diversity, intra-user diversity, and coverage. Due to its simplicity, coverage is used in this paper. The specific definition is given as follows:

$$CV = \frac{N_{\mathrm{dif}}}{N} \qquad (9)$$

where $N_{\mathrm{dif}}$ is the number of different items in the recommendation lists for the users in the same cluster, and $N$ is the total number of items. Obviously, within a certain level of accuracy, a higher value of coverage indicates a better recommendation.

### C. Individual Representation

Directly, items recommended to a user are encoded by a vector of integer values, each of which represents the corresponding item number. Since we aim at providing recommendations for all the users in the same cluster, the chromosome is encoded by a matrix. Assuming that $L$ items will be recommended to each user and there are $K$ users in the cluster, the scale of the matrix is thus $K \times L$. An illustration of the encoding method is given in Table 1, where rows represent users and columns represent items. Usually, RSs will not recommend one item to one user twice. This means that duplicate alleles are not allowed in the same row. In addition, for a given user, there is no need to suggest items rated by the given user in the past. However, different users often rate different items, leading to different search spaces of decision variables. Hence, the modeled optimization problem can be considered as a complex discrete MOP.

**TABLE 1** Illustration of chromosome encoding.

| | ITEM 1 | ITEM 2 | ... | ITEM $L$ |
|---|---|---|---|---|
| USER 1 | 5 | 3 | | 13 |
| USER 2 | 16 | 27 | | 8 |
| ... | | ... | | |
| USER $K$ | 19 | 5 | | 7 |

### D. Genetic Operators

The genetic operators used in our algorithm include crossover and mutation, which are performed to produce new solutions. In this paper, we adopt the uniform crossover. Since one item is not allowed to be suggested to one user twice, an additional operation should be executed to avoid generating invalid solutions. The procedure of the crossover operator can be described as follows. Firstly, the same items from two parents are identified and propagated to the child. Then, the remaining alleles perform crossover. A random number in

**TABLE 2** Properties of the test data sets.

| DATA SET | USERS | ITEMS | SPARSITY |
|---|---|---|---|
| MOVIELENS 1 | 200 | 1682 | $1.39 \times 10^{-2}$ |
| MOVIELENS 2 | 258 | 1682 | $5.17 \times 10^{-2}$ |
| MOVIELENS 3 | 227 | 1682 | $2.38 \times 10^{-2}$ |
| MOVIELENS 4 | 258 | 1682 | $6.89 \times 10^{-2}$ |

[0, 1] is produced for each remaining position in the child. If the number is larger than 0.5, the child receives the corresponding allele from the first parent. Otherwise, it receives the allele from the second parent. An illustration of the crossover operator is shown in Fig. 2. Note that the crossover operator is performed row by row and then the two parent matrices complete crossover.

The mutation operator is applied to a single individual. If one allele in the parent matrix is to be mutated, another available item is randomly selected from the item set to replace the initial one. An item available means that the item does not exist in the parent. In this way, the mutation operator can always generate feasible solutions.

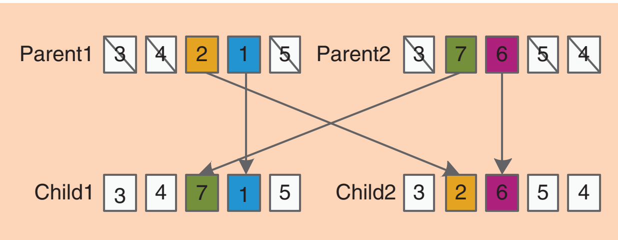


**FIGURE 2** Illustration of the crossover operator. Only the positions without slash perform crossover. Two generated random numbers are 0.1 and 0.8, respectively.

## IV. Experimental Studies

### A. Experimental Settings

To evaluate the performance of the proposed algorithm, we use a classical benchmark data set, Movielens. The Movielens data set can be downloaded from the web site of GroupLens Research (http://www.grouplens.org/). This data set contains 943 users and 1682 movies. Here, we consider a binary rating system ("like" or "dislike"). Since Movielens uses a rating system (ratings 1-5), we preprocess the data set with the same method applied in [17]. An item is considered to be liked by a user, if the user rated this item at least 3. Then we randomly select 80% of the data as the training set, and the remaining data constitutes the probe set. The training set is treated as known information for generating recommendations, while the probe set is used to evaluate the performance of RSs. In order to accelerate the search process, the users are divided into several clusters. In our experiments, we divide the users into four clusters, thus generating four different relatively small data sets. The properties of these data sets are presented in Table 2, where the sparseness of each data set is defined as the number of links divided by the total number of user-object pairs [16]. A sparse data set indicates that only a few items are rated by users.

As is known to all, some common parameters in the MOEA need to be predetermined. The specific values of the parameters used in the computational experiments are listed in Table 3. All experiments are implemented in Matlab on an Intel(R) Core i3 computer with 2.13GHz CPU and 4.00GB memory. To obtain statistical results, 30 independent runs are performed for each data set.

### B. Performance Metrics

Precision is widely used to measure the accuracy of recommendations [19]. For a given user $i$, precision $P_i(L)$ is defined as

$$P_i(L) = \frac{d_i(L)}{L} \tag{10}$$

where $d_i(L)$ is the number of relevant items, which are in the recommendation list and also preferred by user $i$ in the probe set. $L$ is the length of the recommendation list. The obtained mean precision of all users can reveal recommendation accuracy of RSs generally.

Coverage denotes the ability of a recommendation algorithm to recommend diverse items. It is considered as one of the two conflicting objectives in our proposed algorithm. Here we also adopt it as a performance metric to measure the diversity of recommendations.

Novelty is used to measure how well RSs recommend unknown items to users. To measure the unexpectedness of

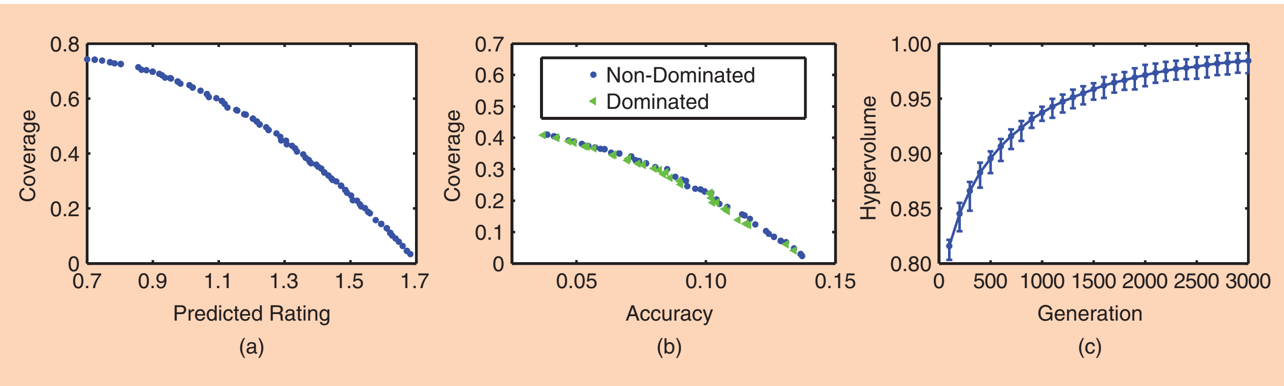


**FIGURE 3** Results of MOEA-ProbS on Movielens 1. (a) Plots of final non-dominated solutions with the highest hypervolume. (b) Plots of final solutions in the accuracy-coverage space (c) The error-bar of hypervolume metric of population among 30 independent runs with different generations.

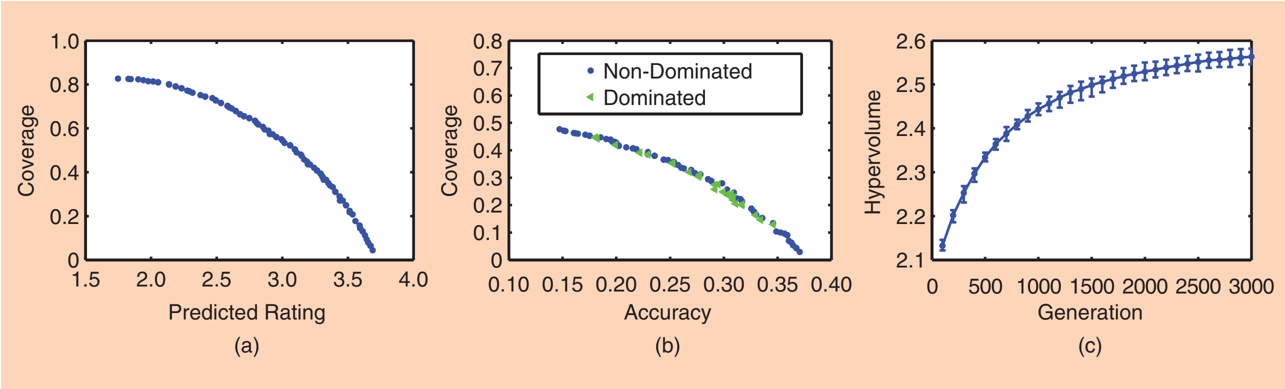


**FIGURE 4** Results of MOEA-ProbS on Movielens 2. (a) Plots of final non-dominated solutions with the highest hypervolume. (b) Plots of final solutions in the accuracy-coverage space (c) The error-bar of hypervolume metric of population among 30 independent runs with different generations.

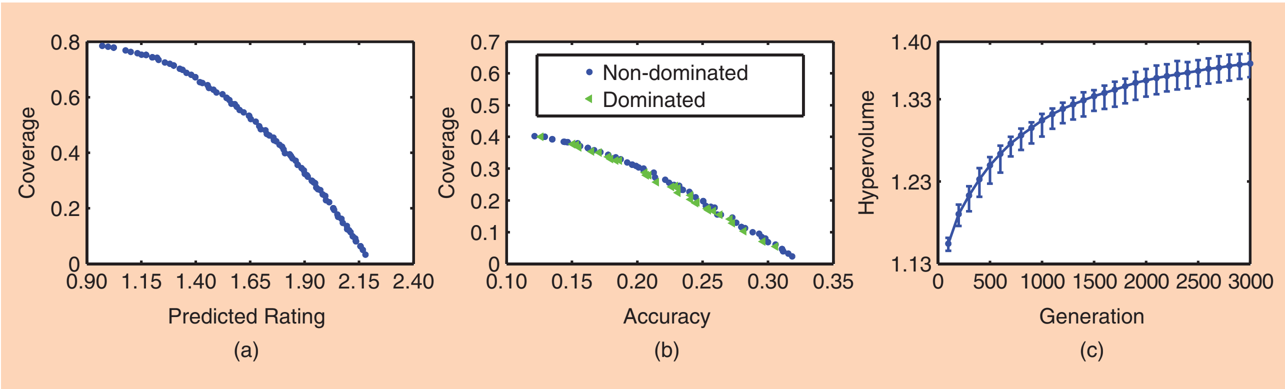


**FIGURE 5** Results of MOEA-ProbS on Movielens 3. (a) Plots of final non-dominated solutions with the highest hypervolume. (b) Plots of final solutions in the accuracy-coverage space (c) The error-bar of hypervolume metric of population among 30 independent runs with different generations.

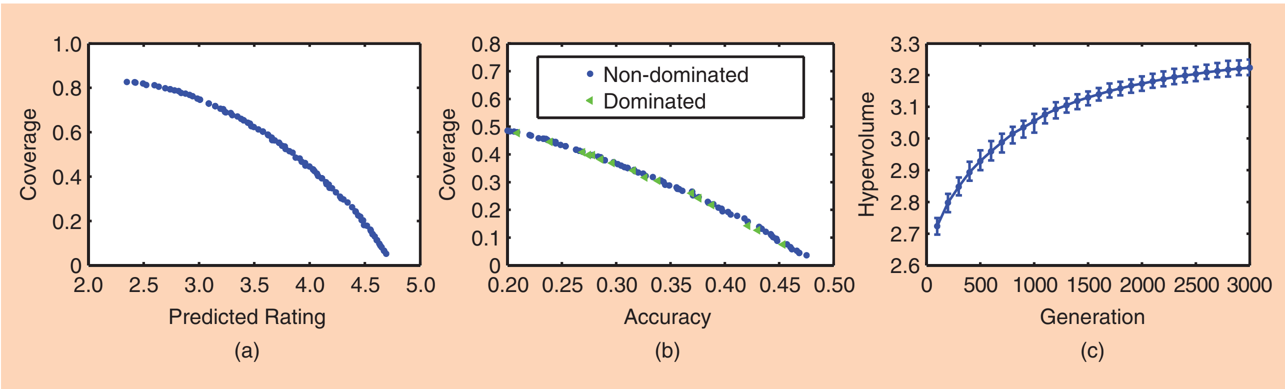


**FIGURE 6** Results of MOEA-ProbS on Movielens 4. (a) Plots of final non-dominated solutions with the highest hypervolume. (b) Plots of final solutions in the accuracy-coverage space (c) The error-bar of hypervolume metric of population among 30 independent runs with different generations.

the recommended items, we use self-information. Given an item $\alpha$, the probability to collect it by a random-selected user is $k_\alpha/M$, where $M$ is the total number of users, and $k_\alpha$ is the degree of item $\alpha$ (i.e., the popularity of item $\alpha$) [19]. The self-information of item $\alpha$ is thus:

$$N_\alpha = \log_2\left(\frac{M}{k_\alpha}\right). \tag{11}$$

A user-relative novelty is obtained by calculating the average self-information of items in the target user's recommendation list. Then the mean novelty $N(L)$ over all users can be obtained according to:

$$N(L) = \frac{1}{ML}\sum_{i=1}^{M}\sum_{\alpha\in O_L^i} N_\alpha \tag{12}$$

where $O_L^i$ is the recommendation list of user $i$ and $L$ is the length of the recommendation list.

### C. Experimental Results

1) *Effectiveness of MOEA-ProbS:* In this subsection, we present the experimental results of MOEA-ProbS on the four data sets. To show the effectiveness of the proposed algorithm, the final non-dominated

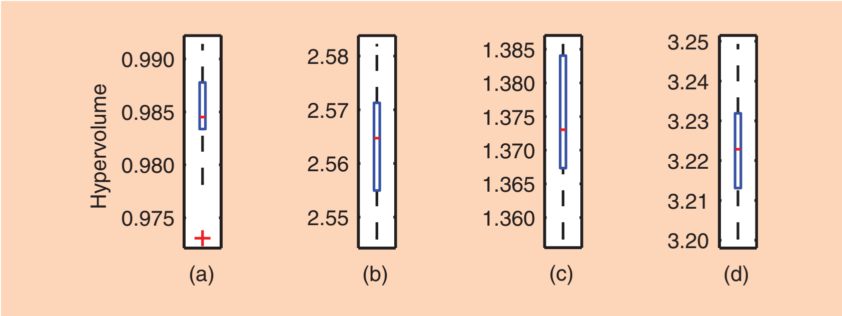


**FIGURE 7** Statistical values of hypervolume for four data sets. (a) Movielens 1. (b) Movielens 2. (c) Movielens 3. (d) Movielens 4. On each box, the central mark is the median and the edges of the box mean the 25th and 75th percentiles. The whiskers extending to the most extreme data points are not outliers. The outliers are denoted by symbol "+".

solutions with the highest hypervolume[1] for each data set are displayed. Since each solution represents recommendations to all the users in the same cluster, the accuracy and coverage of these recommendations can be calculated to examine the quality of the solution. The final solutions of MOEA-ProbS in the accuracy-coverage space are then plotted. In addition, to observe the convergence trend of the MOEA, we record the hypervolume values of the non-dominated solutions among 30 independent runs with different generations. Since the two objectives are non-negative obviously, the reference point for computing hypervolume is set to the origin.

From Figs. 3-6, it can be concluded that there exists a tradeoff between recommendation accuracy and diversity. After a certain number of generations, the proposed MOEA-ProbS can generate a set of recommendations. Note that the accuracy and coverage of different recommendations determined by a set of non-dominated solutions can also form a non-dominated front. However, there may exist some dominated points in the accuracy-coverage space. The reason is that the predicted rating is not exactly equal to the true accuracy of recommendations. For example in Fig. 3 (b), the points denoted by green left triangle represent the dominated solutions. According to the error bars of hypervolume metric in Figs. 3-6, the proposed MOEA-ProbS is robust and effective, which can be proved further by the box plots in Fig. 7.

2) *Comparison results:* To show the advantages of the proposed MOEA-ProbS, we compare it with several well-known recommendation techniques, including CF [58] and matrix factorization method (MF) [59], which can produce only one solution. In addition, a

[1]Hypervolume is an often-used performance metric for multi-objective optimization problems, which can measure both convergence and diversity of the solutions. Moreover, it is the only unary indicator that is strictly monotonic with Pareto dominance. That is to say larger values of hypervolume indicate better solutions [57].

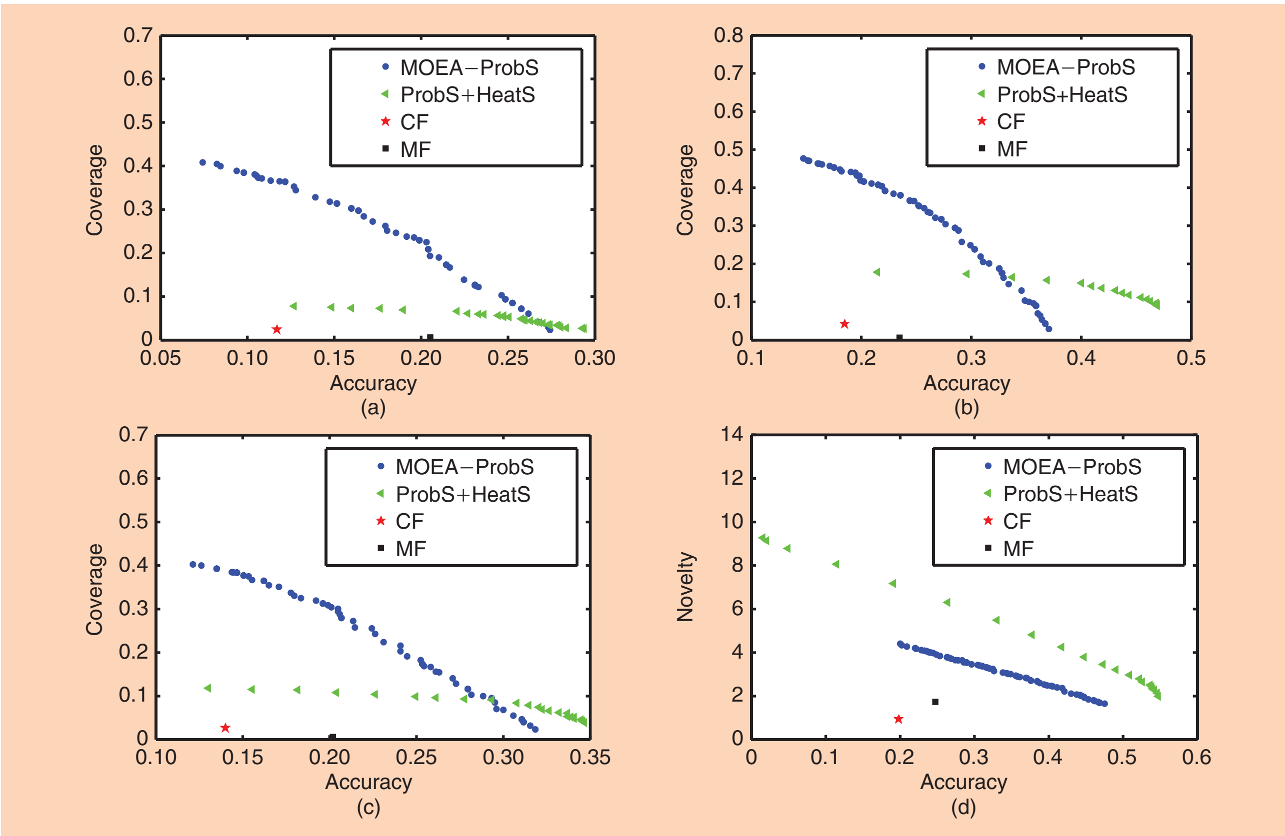


**FIGURE 8** Final non-dominated solutions of CF, MF, MOEA-ProbS and ProbS+HeatS in the accuracy-coverage space. (a) Movielens 1. (b) Movielens 2. (c) Movielens 3. (d) Movielens 4.

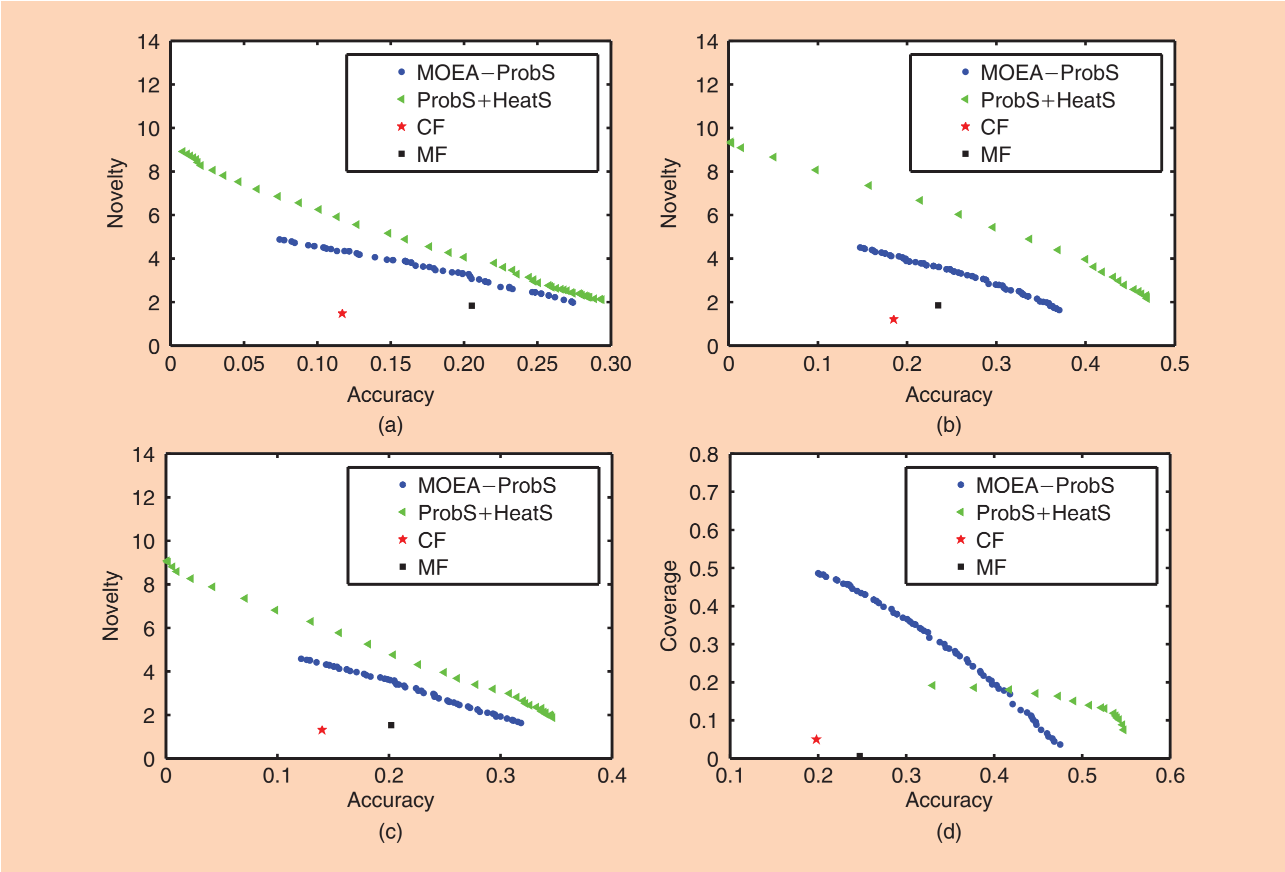


**FIGURE 9** Final non-dominated solutions of CF, MF, MOEA-ProbS and ProbS+HeatS in the accuracy-novelty space (a) Movielens 1. (b) Movielens 2. (c) Movielens 3. (d) Movielens 4.

hybrid recommendation algorithm [16] is selected as a comparative algorithm, which combines an accuracy-based method (ProbS) and a diversity-focused method (HeatS). For convenience, the hybrid algorithm is denoted by ProbS+HeatS. A weight parameter $\lambda \in [0,1]$ is used to incorporate these two algorithms with completely different features. Different from searching for one optimal $\lambda$ through extensive experiments in [16], we generate a number of $\lambda$ evenly sampled in $[0,1]$. Then the experiments with different $\lambda$ are conducted to get a set of recommendations. A non-dominated front can be obtained by eliminating the dominated points in the accuracy-diversity or accuracy-novelty space. For fair comparison, the number of different $\lambda$ is equal to the size of population used in our MOEA.

In the experiments, three performance metrics are considered, namely, accuracy, coverage and novelty. As displayed in Figs. 8 and 9, the solutions of CF and MF are dominated by those of MOEA-ProbS on all the data sets, which demonstrates the effectiveness of our algorithm. Fig. 8 shows that MOEA-ProbS is able to generate multiple recommendations with higher coverage and similar accuracy compared to ProbS+HeatS. This is a promising property, especially for online business. Diverse items can be discovered to stimulate the purchase desire of customers. However, MOEA-ProbS is beaten by ProbS+HeatS according to the accuracy metric. The reason is twofold. First, the performance of our algorithm is mainly influenced by the introduced accuracy-based recommendation technique. Hybrid recommendation methods, which have been proved to provide more accurate recommendations [29], can be employed in our model. Second, the large-scale search space may cause difficulty. In order to improve the search ability, MOEAs should be elaborately designed and some local search methods can be taken into account. According to the

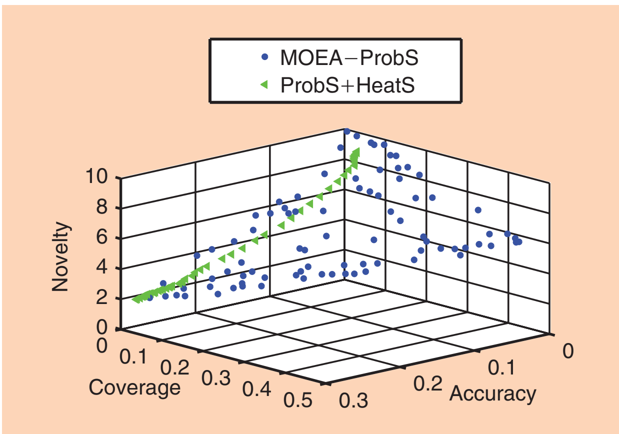


**FIGURE 10** Final non-dominated solutions of MOEA-ProbS and ProbS+HeatS in the accuracy-novelty-coverage space on Movielens 1.

values of hypervolume reported in Table 4, the proposed MOEA-ProbS gains a slight superiority over ProbS+HeatS in terms of accuracy and coverage.

However, Fig. 9 shows our deficiency in generating novel recommendations compared to ProbS+HeatS. This is due to the use of HeatS, which is inclined to suggest less popular items. In fact, high novelty can be obtained by recommending items as less popular as possible to users. Particularly, the value of novelty reaches the maximum, when only HeatS works ($\lambda = 0$). Nevertheless, it will result in a low accuracy, which can be observed by the extreme point close to y-axis in Fig. 9. In contrast, the accuracy of recommendations obtained by MOEA-ProbS is well maintained. Note that the performance of ProbS+HeatS is determined by the parameter $\lambda$, which varies with the data sets. However, it is difficult and computationally expensive to choose a suitable parameter $\lambda$ in the training stage. Moreover, several runs need to be performed to get multiple recommendations. Different from ProbS+HeatS, the proposed MOEA-ProbS can make multiple recommendations in one run without additional parameters. To improve the novelty of recommendations obtained by our algorithm, a good way is to introduce the novelty as the third objective to be optimized, which will be discussed in the following subsection.

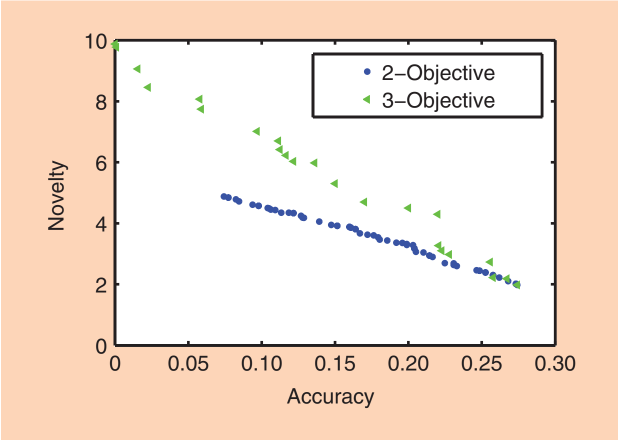

**FIGURE 11** Final non-dominated solutions of MOEA-ProbS with two objectives and MOEA-ProbS with three objectives in the accuracy-novelty space on Movielens 1.

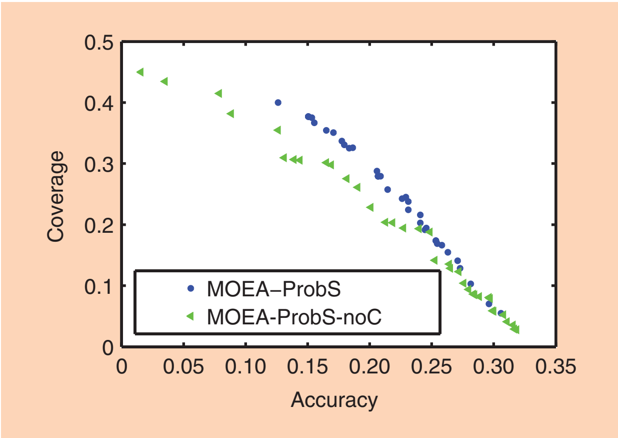

**FIGURE 12** Final non-dominated solutions of MOEA-ProbS and MOEA-ProbS-noC in the accuracy-coverage space on Movielens 3.

In addition, the computational time in seconds (Time) required by each algorithm is given in Table 4, where the execution time of ProbS+HeatS is the total time used by multiple runs to get multiple recommendations. It is evident that ProbS+HeatS and MOEA-ProbS are much more time-consuming than CF and MF. For ProbS+HeatS, the combination of two algorithms increases the computational burden, and the implementation of multiple runs brings about expensive time cost further. Due to the complexity of the modeled MOP, some computational cost is needed for our algorithm to search for a set of optimal solutions. Note that the computational time of ProbS+HeatS for selecting suitable parameter $\lambda$ is not involved. The efficiency of the proposed MOEA-ProbS is thus comparable to that of ProbS+HeatS.

**TABLE 3** Parameter settings of the algorithm.

| PARAMETER | MEANING | VALUE |
|---|---|---|
| *L* | THE LENGTH OF THE RECOMMENDATION LIST | 10 |
| *NP* | THE SIZE OF POPULATION | 100 |
| *pc* | THE CROSSOVER PROBABILITY | 0.8 |
| *pm* | THE MUTATION PROBABILITY | 1/*L* |
| *gmax* | THE NUMBER OF GENERATIONS | 3000 |

3) *Discussions of Objectives:* In this subsection, we consider the extended three-objective model, by introducing the novelty as the third objective. The maximum number of generation is set to be 6000, and other parameters remain the same as in Table 3. The results of MOEA-ProbS and ProbS+HeatS on Movielens 1 are displayed in Fig. 10. It can be observed that the solutions of ProbS+HeatS in the three-dimensional space form a curve, while those of MOEA-ProbS form a two-dimensional surface. This reveals the effectiveness of the extended three-objective model. Similar results are also observed for other data sets. In addition, to test the performance of recommendation novelty, the results of bi-objective and three-objective models in the accuracy-novelty space are presented in Fig. 11. It is obvious that the novelty of recommendations is greatly improved by introducing the third objective. However, little effect is obtained for the solutions with a relatively high precision. Moreover, it will inevitably increase the computational cost in the three-objective model.

4) *Impact of the clustering method:* To investigate the impact of the clustering method, several experiments are carried out in this subsection. Firstly, we use CF and ProbS to test the rationality of using the clustering method. More specifically, we compare the results obtained by CF and ProbS with those by their variants using the clustering method, denoted as CF_C and ProbS_C, respectively. The k-means clustering method is also applied. The detail results, including the accuracy

**TABLE 4** Results of CF, MF, ProbS+HeatS and ProbS-MOEA on four data sets. HV denotes hypervolume in the accuracy-coverage space.

| | MOVIELENS 1 | | MOVIELENS 2 | | MOVIELENS 3 | | MOVIELENS 4 | |
|---|---|---|---|---|---|---|---|---|
| METHOD | HV | TIME (s) | HV | TIME (s) | HV | TIME (s) | HV | TIME (s) |
| CF | – | 7.36 | – | 10.95 | – | 8.30 | – | 10.43 |
| MF | – | 3.51 | – | 6.27 | – | 4.78 | – | 5.62 |
| ProbS+HEATS | 0.0198 | 927.51 | 0.0772 | 1538.42 | 0.0361 | 1156.27 | 0.0958 | 1573.05 |
| MOEA-ProbS | 0.0794 | 3636.76 | 0.1384 | 4233.58 | 0.0958 | 3869.20 | 0.1763 | 4198.32 |

**TABLE 5** Results of CF_C, CF, ProbS_C, and ProbS on Movielens.

| | MOVIELENS 1 | | MOVIELENS 2 | | MOVIELENS 3 | | MOVIELENS 4 | | MOVIELENS |
|---|---|---|---|---|---|---|---|---|---|
| METHOD | ACCURACY | TIME (s) | ACCURACY | TIME (s) | ACCURACY | TIME (s) | ACCURACY | TIME (s) | TIME (s) |
| CF_C | 0.115 | 7.36 | 0.185 | 10.95 | 0.140 | 8.30 | 0.198 | 10.43 | 37.04 |
| CF | 0.070 | – | 0.164 | – | 0.128 | – | 0.192 | – | 104.72 |
| ProbS_C | 0.270 | 12.48 | 0.416 | 15.24 | 0.317 | 12.98 | 0.490 | 15.75 | 56.45 |
| ProbS | 0.274 | – | 0.371 | – | 0.318 | – | 0.475 | – | 82.19 |

and the computational time required by the related algorithms, are reported in Table 5. Note that the last column in Table 5 displays the computational time of the related algorithms run on the complete Movielens data set. For CF_C and ProbS_C, the time presented in the last column is the sum of time used on the four data sets. From Table 5, it can be concluded that the clustering method can reduce the computational time greatly. Actually, the computational time required by CF is more than the sum of time used by CF_C on the four data sets. Similar performance is also observed for ProbS. In particular, the accuracy of CF is improved for all the four data sets by using the clustering method. Since CF is based on the similarity of users, more effective information from similar users in the same cluster are available to produce better results. For ProbS, by using the clustering method, the accuracy is improved on Movielens 2 and Movielens 4 while slightly decreased on other data sets.

Next, we compare the proposed MOEA-ProbS with its variant without clustering (MOEA-ProbS-noC). MOEA-ProbS-noC is tested on Movielens 3 with a computational time limit, which is set to be twice the sum of time required by MOEA-ProbS on the four data sets. As shown in Fig. 12, MOEA-ProbS-noC performs slightly better than MOEA-ProbS in terms of coverage metric. It is easy to understand that as more users participate in rating items, more items will be found and recommended to users, leading to higher coverage. Nevertheless, the accuracy of recommendations becomes worse by using the clustering method.

## V. Conclusions

In this paper, we developed a general multi-objective recommendation model to simultaneously optimize recommendation accuracy and diversity. The accuracy was predicted by the ProbS method while the diversity was measured by recommendation coverage. NSGA-II was adopted to solve the modeled MOP for personalized recommendation. To reduce the computational cost, we used the k-means clustering technique to split the users into several relatively small clusters. The proposed MOEA-based recommendation algorithm can make multiple recommendations for multiple users in only one run. The experimental results show that the proposed algorithm can provide a set of diverse and accurate recommendations for users. In addition, the experiments for the clustering method indicate that it can improve the algorithmic efficiency.

However, recommendation accuracy of our proposed algorithm can still be improved. Our future work will focus on a more in-depth analysis of RSs, including the following aspects: 1) hybridizing other recommendation techniques to further improve the performance of the proposed algorithm; 2) developing specialized MOEAs to solve the optimization problem quickly and robustly; and 3) applying the proposed algorithm to large-scale data sets.


## Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant nos. 61273317, 61422209, 61473215), the National Top Youth Talents Program of China, the Specialized Research Fund for the Doctoral Program of Higher Education (Grant no. 20130203110011) and the Fundamental Research Fund for the Central Universities (Grant nos. K50510020001 and K5051202053).


## References


[1] P. Resnick and H. R. Varian, "Recommender systems," *Commun. ACM*, vol. 40, no. 3, pp. 56–58, 1997.

[2] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowl.-Based Syst.*, vol. 46, pp. 109–132, July 2013.

[3] P. Winoto and T. Y. Tang, "The role of user mood in movie recommendations," *Expert Syst. Applicat.*, vol. 37, no. 8, pp. 6086–6092, 2010.

[4] R. G. Crespo, O. S. Martínez, J. M. C. Lovelle, B. García-Bustelo, J. E. L. Gayo, and P. O. de Pablos, "Recommendation system based on user interaction data applied to intelligent electronic books," *Comput. Human Behav.*, vol. 27, no. 4, pp. 1445–1449, July 2011.

[5] S. K. Lee, Y. H. Cho, and S. H. Kim, "Collaborative filtering with ordinal scale-based implicit ratings for

mobile music recommendations," *Inf. Sci.*, vol. 180, no. 11, pp. 2142–2155, June 2010.
[6] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: A constant time collaborative filtering algorithm," *Inf. Retr.*, vol. 4, no. 2, pp. 133–151, July 2001.
[7] S. T. Cheng, G. J. Horng, and C. L. Chou, "The adaptive recommendation mechanism for distributed group in mobile environments," *IEEE Trans. Syst., Man, Cybern. C,* vol. 42, no. 6, pp. 1081–1092, Nov. 2012.
[8] S. H. Ha, "Helping online customers decide through Web personalization," *IEEE Intell. Syst.*, vol. 17, no. 6, pp. 34–43, Nov. 2002.
[9] J. Bennett and S. Lanning, "The Netflix prize," in *Proc. 13th ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining ACM*, 2007, pp. 3–6.
[10] S. M. McNee, J. Riedl, and J. A. Konstan, "Being accurate is not enough: How accuracy metrics have hurt recommender systems," in *Proc. Conf. Human Factors Computing Systems*, New York, 2006, pp. 1097–1101.
[11] N. Hurley and M. Zhang, "Novelty and diversity in top-N recommendation—Analysis and evaluation," *ACM Trans. Internet Technol.*, vol. 10, no. 4, pp. 1–30, Mar. 2011.
[12] S. Vargas and P. Castells, "Rank and relevance in novelty and diversity metrics for recommender systems," in *Proc. 5th ACM Conf. Recommender System*, New York, 2011, pp. 109–116.
[13] B. Smyth and P. McClave, "Similarity vs. diversity," in *Case-Based Reasoning Research and Development*, D. Aha and I. Watson, Eds. Berlin Heidelberg, Germany: Springer-Verlag, 2001, pp. 347–361.
[14] M. Zhang and N. Hurley, "Avoiding monotony: Improving the diversity of recommendation lists," in *Proc. ACM Conf. Recommender Systems*, New York, 2008, pp. 123–130.
[15] G. Adomavicius and Y. Kwon, "Improving aggregate recommendation diversity using ranking-based techniques," *IEEE Trans. Knowledge Data Eng.*, vol. 24, no. 5, pp. 896–911, May 2012.
[16] T. Zhou, Z. Kuscsik, J. G. Liu, M. Medo, J. R. Wakeling, and Y. C. Zhang, "Solving the apparent diversity-accuracy dilemma of recommender systems," *Proc. Natl. Acad. Sci.*, vol. 107, no. 10, pp. 4511–4515, 2010.
[17] T. Zhou, J. Ren, M. Medo, and Y. C. Zhang, "Bipartite network projection and personal recommendation," *Phys. Rev. E*, vol. 76, no. 4, p. 046115, Oct. 2007.
[18] L. Lü and W. Liu, "Information filtering via preferential diffusion," *Phys. Rev. E*, vol. 83, no. 6, p. 066119, June 2011.
[19] L. Lü, M. Medo, C. H. Yeung, Y. C. Zhang, Z. K. Zhang, and T. Zhou, "Recommender systems," *Phys. Rep.*, vol. 519, no. 1, pp. 1–49, Oct. 2012.
[20] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[21] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowledge Data Eng.*, vol. 17, no. 6, pp. 734–749, June 2005.
[22] G. Adomavicius, N. Manouselis, and Y. Kwon, "Multi-criteria recommender systems," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Berlin Heidelberg, Germany: Springer-Verlag, 2011, pp. 769–803.
[23] K. Miettinen, *Nonlinear Multiobjective Optimization*. Berlin Heidelberg, Germany: Springer-Verlag, 1999.
[24] M. Pazzani and D. Billsus, "Content-based recommendation systems," in *The Adaptive Web*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Berlin Heidelberg, Germany: Springer-Verlag, 2007, pp. 325–341.
[25] J. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," in *The Adaptive Web*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Berlin Heidelberg, Germany: Springer-Verlag, 2007, pp. 291–324.
[26] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model," *IEEE Trans. Audio, Speech, Language Processing*, vol. 16, no. 2, pp. 435–447, Feb. 2008.
[27] S. H. Choi, Y. S. Jeong, and M. Jeong, "A hybrid recommendation method with reduced data for large-scale application," *IEEE Trans. Syst., Man, Cybern. C*, vol. 40, no. 5, pp. 557–566, Sept. 2010.
[28] E. Amolochitis, I. T. Christou, and Z. H. Tan, "Implementing a commercial-strength parallel hybrid movie recommendation engine," *IEEE Intell. Syst.*, vol. 29, no. 2, pp. 92–96, Mar. 2014.
[29] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Model. User-Adapt. Interact.*, vol. 12, no. 4, pp. 331–370, Nov. 2002.
[30] R. Bell, Y. Koren, and C. Volinsky, "Modeling relationships at multiple scales to improve accuracy of large recommender systems," in *Proc. 13th ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining*, New York, 2007, pp. 95–104.
[31] M. Jahrer, A. Töscher, and R. Legenstein, "Combining predictions for accurate recommender systems," in *Proc. 16th ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining*, New York, 2010, pp. 693–702.
[32] Q. Liu, E. Chen, H. Xiong, C. Ding, and J. Chen, "Enhancing collaborative filtering by user interest expansion via personalized ranking," *IEEE Trans. Syst., Man, Cybern. B*, vol. 42, no. 1, pp. 218–233, Feb. 2012.
[33] M. Sarwat, J. Levandoski, A. Eldawy, and M. Mokbel, "LARS: An efficient and scalable location-aware recommender system," *IEEE Trans. Knowledge Data Eng.*, vol. 26, no. 6, pp. 1384–1399, June 2014.
[34] D. Fleder and K. Hosanagar, "Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity," *Manage. Sci.*, vol. 55, no. 5, pp. 697–712, Mar. 2009.
[35] D. Jannach, L. Lerche, F. Gedikli, and G. Bonnin, "What recommenders recommend—An analysis of accuracy, popularity, and sales diversity effects," in *User Modeling, Adaptation, and Personalization*, vol. 7899, S. Carberry, S. Weibelzahl, A. Micarelli, and G. Semeraro, Eds. Berlin Heidelberg, Germany: Springer-Verlag, 2013, pp. 25–37.
[36] C. N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," in *Proc. 14th Int. Conf. World Wide Web*, New York, 2005, pp. 22–32.
[37] N. Lathia, S. Hailes, L. Capra, and X. Amatriain, "Temporal diversity in recommender systems," in *Proc. 33rd Int. ACM SIGIR Conf. Research Development Information Retrieval*, New York, 2010, pp. 210–217.
[38] F. Fouss and M. Saerens, "Evaluating performance of recommender systems: An experimental comparison," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intelligent Agent Technology*, Washington, D.C.: IEEE Computer Society, 2008, pp. 735–738.
[39] N. Kawamae, "Serendipitous recommendations via innovators," in *Proc. 33rd Int. ACM SIGIR Conf. Research Development Information Retrieval*, New York, 2010, pp. 218–225.
[40] G. Shani and A. Gunawardana, "Evaluating recommendation systems," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Berlin Heidelberg, Germany: Springer-Verlag, 2011, pp. 257–297.
[41] C. A. C. Coello, "Evolutionary multi-objective optimization: A historical view of the field," *IEEE Comput. Intell. Mag.*, vol. 1, no. 1, pp. 28–36, Feb. 2006.
[42] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proc. 1st Int. Conf. Genet. Algorithms*, Hillsdale, NJ: L. Erlbaum Associates Inc., 1985, pp. 93–100.
[43] K. Deb, "Multi-objective optimization," in *Search Methodologies*, E. K. Burke and G. Kendall, Eds. Berlin Heidelberg, Germany: Springer-Verlag, 2014, pp. 403–449.
[44] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.
[45] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," Swiss Federal Inst. Technol., Zurich, Switzerland, Tech. Rep. TIK-Report 103, May 2001.
[46] N. Srinivas and K. Deb, "Muiltiobjective optimization using nondominated sorting in genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, Dec. 1994.
[47] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 256–279, June 2004.
[48] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.
[49] M. Gong, L. Jiao, H. Du, and L. Bo, "Multiobjective immune algorithm with nondominated neighbor-based selection," *Evol. Comput.*, vol. 16, no. 2, pp. 225–255, June 2008.
[50] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, Feb. 2011.
[51] G. N. Demir, A. Ş. Uyar, and Ş. Gündüz-Öğüdücü, "Multiobjective evolutionary clustering of web user sessions: A case study in Web page recommendation," *Soft Comput.*, vol. 14, no. 6, pp. 579–597, Apr. 2010.
[52] C. Rana and S. K. Jain, "An evolutionary clustering algorithm based on temporal features for dynamic recommender systems," *Swarm Evol. Comput.*, vol. 14, pp. 21–30, Feb. 2014.
[53] S. Tyagi and K. K. Bharadwaj, "Enhancing collaborative filtering recommendations by utilizing multi-objective particle swarm optimization embedded association rule mining," *Swarm Evol. Comput.*, vol. 13, pp. 1–12, Dec. 2013.
[54] M. T. Ribeiro, A. Lacerda, E. S. de Moura, I. Hata, A. Veloso, and N. Ziviani, "Multi-objective Pareto-efficient approaches for recommender systems," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 1, pp. 1–20, Feb. 2013.
[55] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, May 2005.
[56] M. Gong, Q. Cai, X. Chen, and L. Ma, "Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 82–97, Feb. 2014.
[57] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms—A comparative case study," in *Parallel Problem Solving from Nature—PPSN V*, A. Eiben, T. Bäck, M. Schoenauer, and H. P. Schwefel, Eds. Berlin Heidelberg, Germany: Springer-Verlag, 1998, pp. 292–301.
[58] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, Jan. 2004.
[59] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.