

Research Article

A Probability-Based Hybrid User Model for Recommendation System

Jia Hao,¹ Yan Yan,¹ Guoxin Wang,¹ Lin Gong,¹ and Bo Zhao²

¹School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China

²Beijing Institute of Astronautic System Engineering, Beijing 310027, China

Correspondence should be addressed to Jia Hao; haojia632@bit.edu.cn

Received 13 July 2015; Revised 16 December 2015; Accepted 20 December 2015

Academic Editor: Matteo Gaeta

Copyright © 2016 Jia Hao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the rapid development of information communication technology, the available information or knowledge is exponentially increased, and this causes the well-known information overload phenomenon. This problem is more serious in product design corporations because over half of the valuable design time is consumed in knowledge acquisition, which highly extends the design cycle and weakens the competitiveness. Therefore, the recommender systems become very important in the domain of product domain. This research presents a probability-based hybrid user model, which is a combination of collaborative filtering and content-based filtering. This hybrid model utilizes user ratings and item topics or classes, which are available in the domain of product design, to predict the knowledge requirement. The comprehensive analysis of the experimental results shows that the proposed method gains better performance in most of the parameter settings. This work contributes a probability-based method to the community for implement recommender system when only user ratings and item topics are available.

1. Introduction

Many researchers believe that comprehensive utilization of accumulated knowledge is vital to maintaining the competitive advantage in today's knowledge-based economy, particularly within research and development organizations [1–3]. According to the statistic, over 90% of product design is variant design [4], which means new requirements can be satisfied by simply revising the accumulated design knowledge like solution report, 3D models, and others [5]. Therefore, the design knowledge acquisition and utilization are of great importance for completing design tasks. However, the process of knowledge acquisition usually takes up over half of the design time, which significantly affects the design efficiency and extends the design cycle. This is caused by the separation between design knowledge and designers; in that situation the designers are inevitably to ask or search required information and design knowledge. Therefore, a recommender system is an urgent demand for decrease in the time spent on design knowledge acquisition.

To manage design knowledge, a number of design knowledge management systems, like experience design

knowledge, 3D model base and standard parts base, and so forth, are deployed in many organizations. These knowledge management systems have recorded information for implementing a recommender system. On one hand, in most of the knowledge management systems, a set of carefully predefined knowledge topics or classes is used to organize design knowledge existing in the knowledge base, and the most commonly used is the product structure (Bill of Material (BOM)). On the other hand, many knowledge management systems record the ratings that designers have given to the design knowledge. Before this research work, we have developed a recommender system for designers based on the method by Al-Shamri and Bharadwaj [6], which combines the user ratings and product topics, and the system is deployed to the practical environment. After a period of running, there is a requirement for further promoting the performance of the recommender system. Therefore, this research work is an attempt to promote the performance of the recommender systems which is based on the topics and ratings of design knowledge. In this work, we present a probability-based hybrid user model; based on that, the corresponding similarity calculation method as well as

the recommendation method is proposed. The experimental results show that this model obtains better performance in most parameter settings. The main contributions of this research include (1) combining the ratings and topics by a simple probability formula; (2) providing a new choice for implementing recommender system when only ratings and topics are available. The remainder of this paper is structured as follows. In Section 2, the related techniques are reviewed and analyzed. Then Section 3 presents the proposed method for constructing the probability based hybrid recommendation model. Following that, Section 4 gives a comprehensive experiments analysis. At last, Section 5 makes the conclusion and discusses the future works.

2. Related Works

In modern enterprises, tremendous amounts of data and information are stored and even more are generated than ever before. One side effect for the staff is information overload, which means the staff has been provided more information than they can process efficiently [7–9]. Recommendation systems have emerged as a research methodology in response to this problem [10]. According to [7], there are four kinds of filtering techniques that recommender systems may leverage, including content-based method (CB) [11], collaborative filtering method (CF) [12], knowledge-based method (KB) [13], and hybrid method [14]. Each method has its own advantages and disadvantages; for example, the CF method has the advantage of independence of item's content and the disadvantage of sparsity rating data. To overcome the disadvantages and combine the advantages of different methods, the hybrid recommender system is proposed [15]. The hybrid recommender system indicates the recommender system that combines multiple recommendation techniques together to produce better performance [7, 16]. According to [15], there are seven strategies for combining different recommender techniques, including weighted, mixed, switching, feature combination, feature augmentation, cascade, and metalevel.

In recent years, CF has become one of the most widely implemented recommender methods [6, 17, 18]. Researchers in the area tend to combine CB method to achieve high scalability while maintaining relatively high prediction accuracy [6, 10, 19]. For examples, Rastin and Zolghadri Jahromi [20] proposed a hybrid method which integrates collaborative filtering and content-based methods. They believe that two users are similar if their ratings on items that have similar context are similar. Yao et al. [21] built a collaborative filter and content-based filter hybrid user model to dynamically recommend Web services. Ronen et al. [22] presented a framework for automated selecting of informative content-based features, and this framework is in dependent on the type of recommender system, which means the method is generalized well to different recommender systems. Baragáns-Martínez et al. [23] developed a Web 2.0 TV recommendation system which is named *queveo.tv*. They proposed a hybrid method which combined content-filtering techniques with collaborative filtering; the social network also integrated together to make better recommendation

performance. Berkovsky et al. [24] proposed a method to merge different user models from different systems and use the merged user model to make recommendations. The same authors transformed this merged user model from a collaborative filtering system to a content-based recommendation system [25]. Degenmis et al. [26] proposed a new content-collaborative hybrid recommender that computes similarities between users, relying on their content-based profiles instead of comparing their rating styles and the content-based user profiles play a key role in the proposed hybrid recommender. Many of the existing researches rely on the description content of the items and this is unavailable in many situations, especially in the product design domain.

When the description content of the items is sometimes unavailable, the knowledge related to users or items can be leveraged for recommender system. Ontology-based or semantic-driven recommender system is a research branch to utilize the knowledge about items for recommendation [7, 16]. In these methods, ontology is used to model a big amount of related knowledge for recommender systems [27]. If knowledge related to users or items can be obtained, this method will contribute to the performance of recommender systems. Moreno et al. [28] developed a Web-based system that provides a personalized recommender system of touristic activities in the region of Tarragona, and the ontology is used to classify and label the touristic activities for further reasoning process.

The above-mentioned methods cannot well satisfy the situation we are facing. On one hand, most of the content-based and collaborative-based hybrid methods require description content, which is unavailable in our problem. On the other hand, we do not have complicated knowledge about items except the predefined simple topics or classes, and the ontology construction itself is an extremely complex process, which requires many experts to work collaboratively. Therefore, the requirement of building an ontology for recommender systems is not urgent in this research work. Among the literature, Al-Shamri and Bharadwaj [6] developed an approach that uses the ratings and topics to build a hybrid model which well meets our requirement and situation. Based on their work, we develop a new probability-based method, and the experimental results show that the newly developed method is better in most of the situation. Therefore, we contribute a method to the research community, which can be used when they have only the topics or classes and ratings.

3. Proposed Method

3.1. Symbols Definition. We first defined the symbols that might be used in the following sections. Let $U = u_1, u_2, \dots, u_{|U|}$ be the set of all users, let $I = i = i_1, i_2, \dots, i_{|I|}$ be the set of all items, and let $T = t = t_1, t_2, \dots, t_{|T|}$ be the set of all topics. We defined R as the two-dimensional user-item rating matrix and the element $R_{u,i} \in \{S, 0\}$ represents the rating of user $u \in U$ on item $i \in I$. The value of $R_{u,i}$ is on either the numerical ordinal scale (e.g., 1 to 5) or zero, which indicates an unknown rating (not rated). Here, $S = s_1, s_2, \dots, s_{|S|}$ is defined as the ratings. The matrix R can be explained either

using a row vector or a column vector. The row vector $r_u = [R_{u,1}, \dots, R_{u,|T|}]$ represents the rating of user $u \in U$ to all items. I_u has been defined as a set of items rated by user $u \in U$. The column vector $c_i = [R_{1,i}, \dots, R_{|U|,i}]$ represents the rating of item $i \in I$ by all users. We define F as a two-dimensional item-topic matrix, and the element $F_{i,t} \in \{0, 1\}$ is set to 1 when the item $i \in I$ belongs to topic $t \in T$ and 0 otherwise. The row vector $r_i = [F_{i,1}, F_{i,2}, \dots, F_{i,|T|}]$ represents which topics item $i \in I$ belongs to. One item $i \in I$ may belong to more than one topic, which means $\sum r_i \geq 1$. The following list shows details regarding the symbols used in this paper:

I : the set of items.

$|I|$: the number of items.

U : the set of users.

$|U|$: the number of users.

T : the set of topics.

$|T|$: the number of topics.

S : the set of scores.

$|S|$: the number of scores.

R : the rating matrix.

r_u : the row vector of R .

c_i : the column vector of R .

F : the item and topic matrix.

G : the user and topic matrix.

3.2. Database Validation. In many situations, items are classified into one or more specific topics according to their content; the number of topics is normally far less than the number of items. In this paper, we have attempted to formulate a hybrid user model for item recommendation by combining the rating data and the topic data. Before we develop the hybrid user model, one question should be addressed first: whether the topics could be used to express the interests of the users.

For a given database D , which includes D and F , we cannot ensure the user interests can be expressed by the item topics. An obvious example is that when most of the items belong to a small number of topics, the topics are unable to distinguish user interest. Therefore, we need to analyze and confirm whether the topics are capable of expressing user interest before building the user model.

We determine the relationship between users and topics using $G = R \times F$; each element $G_{u,t}$ in the matrix indicates the total score that user $u \in U$ rates for the topic $t \in T$. The row vector $G_u = [G_{u,1}, G_{u,2}, \dots, G_{u,|T|}]$ represents the total scores that user $u \in U$ rates for every topic. For a specific user $u \in U$, we define cap_u as the metric to measure the coverage of the user's interests:

$$\text{cap}_u = \frac{|T_{\text{OPT}_u}|}{|T|}, \quad (1)$$

where $|T_{\text{OPT}_u}|$ is a set of topics that the user may be interested in (the total score is higher than other topics) and it meets the following condition:

$$\frac{\sum_{t' \in T_{\text{OPT}_u}} G_{u,t'}}{\sum_{t \in G_u} G_{u,t}} \geq \varphi, \quad (2)$$

where φ is the threshold value that determines how many interested topics the user has. Based on that, we define cap_D as the metric to measure the topic's capability of expressing the interests of all users:

$$\text{cap}_D = \frac{1}{|U|} \sum_{u \in U} \text{cap}_u. \quad (3)$$

When most of cap_u and cap_D are small compared with $|T|$, we infer that the user's interests are concentrated on a small number of topics. In other words, the topics are able to distinguish the user's interests.

3.3. User Model Construction. In the traditional CF method, user model is simply represented by a rating vector in which all ratings that the user has given to item are recorded. Similarly, we use a row vector of G to represent user model, which records the sum of ratings that the user has given to all items which belong to every topic. However, this method is unreasonable because the sum of ratings may be very close although each individual rating is different. For example, user₁ rates 5 items which belongs to topic₁ and the scores are 1, 4, 5, 1, and 1. User₂ also rates 5 items which belongs topic₁ and the scores are 5, 1, 1, 3, and 2. Although the sum of both ratings is 12, we cannot believe the two users have similar interest.

Based on the above analysis, we adopts probability to construct the user model. The basic idea is that two similar users should have similar possibilities of rating a specific score on every topic. For example, the possibilities of user₁ rates 1, 2, 3, 4, and 5 on items which belong to topic₁ are 10%, 10%, 20%, 40%, and 20%, and if user₂ and user₁ have similar interest, user₂ should have similar possibilities of rating a specific score on items which belong to topic₁.

In this work, a matrix is used to represent user model (UM), as shown in formula (4). The UM is $|S| \times |T|$ matrix in which row vectors indicate the possibility that a user may rate a specific score on topics:

$$\text{UM} = \begin{bmatrix} p(s_1 | t_1) & \cdots & p(s_1 | t_{|T|}) \\ \vdots & \ddots & \vdots \\ p(s_{|S|} | t_1) & \cdots & p(s_{|S|} | t_{|T|}) \end{bmatrix}, \quad (4)$$

where $p(s_1 | t_1)$ means the possibility of user rates score s_1 to topic t_1 . The value of $p(s_1 | t_1)$ is calculated according to Bayes formula:

$$p(s | t) = \frac{p(s \cap t)}{p(t)}, \quad (5)$$

where $p(s \cap t)$ means the probability that user $u \in U$ gives a score $s \in S$ when the item $i \in I$ belongs to the topic $t \in T$; $p(t)$

indicates that the probability of an item $i \in I$ belongs to the topic $t \in T$, which user $u \in U$ ever rated; and both $p(s \cap t)$ and $p(t)$ may be determined out based on matrix R and matrix F .

3.4. Neighborhood Calculation. In this research work, we use CF as the method to make predication. Therefore, the calculation of neighborhood is a critical step. The number of neighborhood can be fixed or floated [6]. The fixed neighborhood means to select the top k users which have the biggest similarities with the active user while the floated neighborhood means to determine the neighborhood by a similarity threshold. We will use the fixed number of neighborhood in the experimental section.

The essential problem of calculating neighborhood is the measurement of similarity between two user models. The measurement of the similarity is highly depended on the user model, which means that different user model requires corresponding similarity measurement. In this work, the user model is represented by a probability matrix and this is very different with traditional method, which represents user model as a rating vector. When the user is a rating vector, the similarity is figured out easily by many existing methods, like Euclidean distance, Manhattan distance, Cosine, and so on.

In this work, the basic idea of measuring the similarity of two user models is the number of topics which are interested by both of the two users, as shown in formula (6). For example, if user₁ and user₂ share five topics while user₁ and user₃ share three topics, we believe that the similarity between user₁ and user₂ is higher than the similarity between user₁ and user₃:

$$\text{sim}(u_x, u_y) = \frac{|\text{pos}(u_x) \cap \text{pos}(u_y)|}{|T|}, \quad (6)$$

where $\text{sim}(u_x, u_y)$ means the similarity of user u_x and user u_y ; $\text{pos}(u_x)$ represents the interest topics of user u_x ; and $|T|$ means the total number of interest topics of u_x and u_y . The $\text{pos}(u_x)$ is computed by the following formula:

$$\text{pos}(u) = \{t \in T : \text{pos}p(u, t) \geq \theta\}, \quad (7)$$

where θ is a threshold used to determine whether a specific topic is an interested topic. When θ is high, it will be more difficult for a topic to be selected as the interested topic. $\text{pos}p(u, t)$ means the sum of positive possibilities.

Here, the positive probability indicates the probability that the user will rate a high score for some topics, referring to the degree that the user may interest in the topics. For example if the value $p(s = 5 | t)$ is high, we may infer the user interest in the topic t . On the other hand, a passive probability indicates the probability that the user will rate a low score for some topics, which refers to the probability that the user dislikes a specific topic. For example, if the value $p(s = 1 | t)$ is high, we believe the user dislikes topic t . In this research work, only the positive probability is considered to calculate the similarity of two user models. In general, the high score ranges from 3 to 5 when the score ranges from 1 to 5.

TABLE 1: Statistics of the final database.

# of users	497
# of items	903
# of topics	18
# of ratings	79432
Average ratings # of users	159.82
Average ratings # of items	87.96
Average topics # of items	1.96

3.5. Recommendation. The last step is to figure out recommendations, which is a list of top- N items that would most interest the user. The primary task of recommendation is to predict which scores the active user might give to unrated items according to the neighbor set, which gives us the top- N highest score items as the recommendation. In the classical CF, we can predict the score using the average of neighbor ratings. However, the average value ignores the different rating characters of different users. The widely used method is the weighted sum [29, 30], which is also Resnick's prediction formula [31]. Therefore, in this work, Resnick's prediction formula is adopted for all different methods to validate the presented method.

4. Experiments

4.1. Evaluation Database and Performance Metrics. In this research, the research question was arisen from the product design domain, and the solution has been discussed in this paper which is trying to deal with the information overload problem in the product design domain. Because of that, the best way to validate the method is to use a database in the product design domain. However, there is still no unanimous database for validating the recommender systems in this domain. Therefore, we turn to finding widely used database that has similar characteristics of the problem in the product design domain. Another reason for using widely used database is that we believe the proposed method has the potential to deal with many problems in the different domains.

After considering many different databases, the MovieLens is selected because the ratings and topics of items are recorded properly. MovieLens (<http://www.movielens.umn.edu>) is a widely used evaluation database. The database consists of 100,000 ratings, 943 users, and 1682 items. All ratings follow the following numerical scale: (1) bad, (2) average, (3) good, (4) very good, and (5) excellent. Each user in this database has rated at least 20 items. The dataset also contains genre (topic) features for the items that our method needs. A single item can belong to one or more movie genres, which include action, adventure, animation, children's, comedy, crime, documentary, drama, fantasy, film-noir, horror, musical, mystery, romance, sci-fi, thriller, war, or western. To avoid the influence of cold start items and cold start users, we have extracted a denser database from the original database. We only considered items that had been rated more than 20 times. Table 1 details the statistics of the extracted database.

In this paper, the 10 times 10-fold cross-validation process was conducted to evaluate our method's performance. The metrics we used included the mean absolute error (MAE) [30] and the coverage rate [32].

The MAE measured the difference between the predicted ratings generated by the recommendation algorithm and the real ratings given by the user; the lower the value, the better the method's performance. This metric was obtained by aggregating the MAE_i for all users. The MAE_i could be figured out using the following formula [30]:

$$MAE_i = \frac{1}{|R_i|} \sum_{r_{ij} \in R_i} |pr_{ij} - r_{ij}|, \quad (8)$$

where R_i is the validating split and $|R_i|$ is the size of the validating split. pr_{ij} denotes the predicted ratings for the j th item in the validating split, and r_{ij} denotes the real ratings for the j th item in the validating split. Based on that, the overall MAE can be calculated using the following formula:

$$MAE = \frac{1}{N_u} \sum_{i=1}^{N_u} MAE_i, \quad (9)$$

where N_u is the number of users in the validating split.

The MAE can indicate the prediction accuracy of the method; however, the MAE cannot reflect the real utility of a recommendation algorithm, which makes the coverage rate an important metric [32]. There are two ways to define the coverage rate. The first is to calculate the fraction of successful rating predictions for all unrated items; the second considers the items that have been rated by each user. In this paper, we have defined the coverage rate using the second definition, since it better meets the user's need [32]. The value can be found using the following formula:

$$CR = \frac{|PI|}{|VI|} \times 100\%, \quad (10)$$

where CR is the coverage rate and PI is a set of items that have been predicted correctly using the recommendation algorithm. Here, the correctly predicted items do not indicate that the predicted ratings and real ratings are identical. It means that the algorithm correctly distinguished whether or not the user had interest in the item. The VI denotes all items in the validating dataset.

4.2. Database Analysis. Before validating the performance, we must first examine whether the database is capable of expressing the users' interests. According to the method proposed in Section 2, we examined the value of cap_u for all users. Figure 1 shows the histogram of all cap_u ; as we can see, most of cap_u is small (range from 0.05 to 0.27) compared with 1, which means that most of the users' interests focused on a small number of topics. cap_D is 0.21, which also implies that the topics in this database adequately express the user's interest. Therefore, we can use this database to build a probability-based rating and topic hybrid user model.

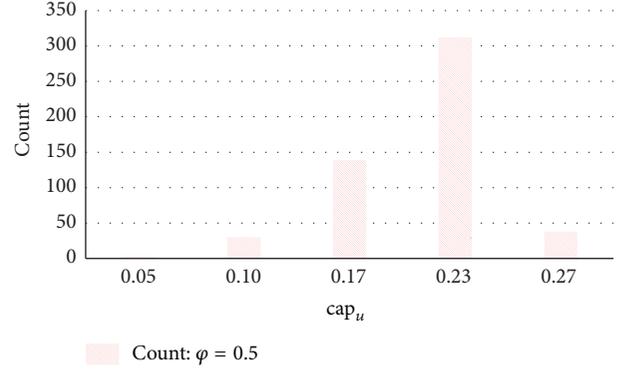


FIGURE 1: The cap distribution of the database.

4.3. Evaluation of System Components. In this section we have assessed the influence of two parameters, including the threshold value θ and the neighborhood number, on the method's performance.

First, we assessed the effect of the threshold value θ on the method's performance by setting the value from 0.3 to 0.9. For each setting, we set the neighborhood number at 4. Later, we will assess the robustness of neighborhood number over a range of value.

As shown in Figure 2, the MAE of our method is approximately 0.819, which refers to the absolute distance between the predicted ratings and real ratings. The standard deviation of MAE is about 0.0045, which implies the MAE is not affected by the threshold value θ . This may be because most of the users' interests focused on a small number of topics; no matter the variation of θ , the positive topics will not change. It is worth noting that $\theta = 0.3$ produced the highest MAE value (0.829); this may be because the small threshold value brings some topics that may fall within the user's interests in the positive topics set. When θ increased to 0.4 or greater values, the MAE value dropped and remained invariant. Figure 2(b) shows that the coverage rate was about 91.7%, which implies our method can recommend over 90% of the items correctly. The standard deviation (0.0045) was also small, which tells us the threshold value has little influence on the coverage rate. We also noticed that $\theta = 0.4$ produced the highest coverage rate; this is because some topics the user is interested in are lost when $\theta = 0.3$ and some topics in which the user is not interested remain when θ is bigger than 0.4. In short, we can conclude that both the MAE and the coverage rate are not obviously affected by the threshold value, and the smaller value (e.g., 0.4) of the threshold value tends to obtain a high coverage rate and small MAE.

Second, we assessed the method's performance when the neighborhood number was set at different values (from 1 to 10). The threshold value was set to 0.4 in this experiment. As shown in Figure 3, when the neighborhood number increased from 1 to 10, the method's MAE first decreased to the lowest value (0.812) at 4 and then continuously increased. It is noteworthy that the MAE was high when neighborhood number was 1 because a single neighborhood is not enough to provide correct rating predictions. Furthermore, when

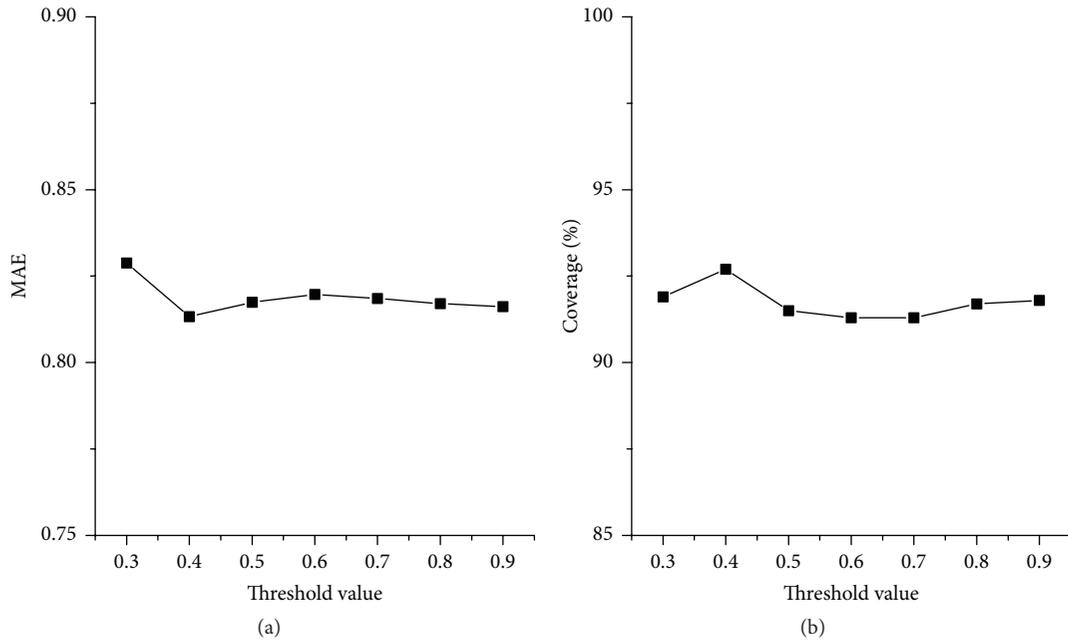


FIGURE 2: The influence of θ on the performance.

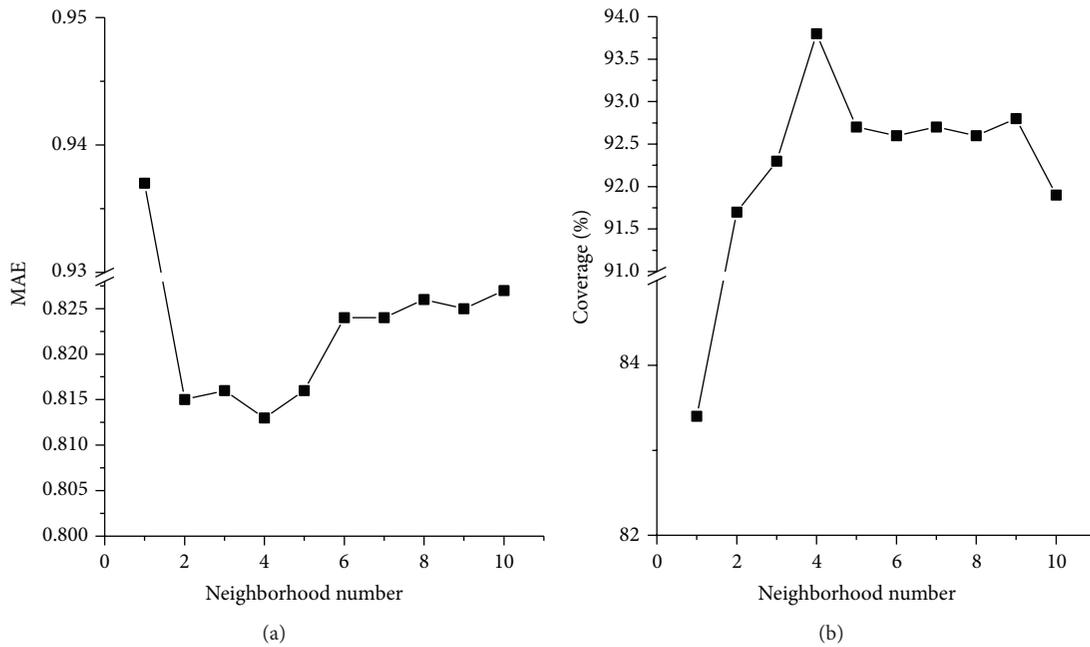


FIGURE 3: The influence of neighborhood number on the performance.

the neighborhood number is greater than 4, the method also obtains a high MAE. This is because some dissimilar neighbors could be introduced if the neighborhood number is too big; therefore, the predicted ratings can be inaccurate. A similar situation occurred relating to the coverage rate in Figure 3(b); when the neighborhood number increased from 1 to 10, the coverage rate first increased to the highest value (0.938) at 4 and then dropped. Therefore, we know that when the neighborhood number is 4, the method performs

at its best. We can conclude from this experiment that the method is sensitive to the neighborhood number; neither a too small value nor a too large value can obtain the best result. Therefore, the selection of an appropriate value for the neighborhood number is critical for our method.

4.4. Performance Comparison. In this section, we have compared our method with the other two methods: the classical

collaborative filtering (CF) method and the GIM method proposed by Al-Shamri and Bharadwaj [6]. The GIM method was selected as a comparison method since this process uses the same information to predict user interest. During the experiment, our method is denoted by PF, the probability-based filtering method. The classical collaborative filtering has been taken as the experiment baseline. Because the classical collaborative filtering method and the GIM method are described in detail in related papers, we can then replicate the methods and compare the performances.

We have compared these three methods' performance on different parameter settings. Two parameters (neighborhood number, dataset density) were selected in order to compare their performance. The first parameter was used to validate the performance in the previous section; therefore, we know the meanings of the parameters. The second parameter is the dataset density, which is used to represent how many ratings the users have in the database. For example, setting the dataset density to 20 means only users who have rated at least 20 items have been selected in the final database. With this parameter we can compare the method's performance using different databases.

First, we compared the method's performance after changing the neighborhood number from 1 to 10. The dataset density was set at 20. This experiment was intended to check whether our PF method could obtain a better performance given different neighborhood numbers. Figure 4 shows the comparison results. We can see from Figure 4(a) that the PF method always obtains the lowest MAE when the neighborhood number has increased from 2 to 10. Although the value is much higher than the GIM method when the neighborhood number is 1, this does not affect the utility of the PF method because we will never select 1 as the neighborhood number in real recommendation systems. In terms of the accuracy of rating prediction, the PF method always outperforms the GIM method and both the GIM and the PF methods are superior to the baseline CF method. In Figure 4(b), the experiment shows that the PF method has a higher coverage rate than the GIM method when the neighborhood number is from 2 to 10, although the gap is not large. The baseline CF method's coverage rate is low from start to finish. We can conclude from this experiment that the neighborhood number affects the performance only slightly and does not change the performance ranking of these three methods.

Second, we compared the performance when providing different database densities to the three methods. During the experiment, the neighborhood number was set at 4. Unlike the previous two experiments, this experiment's result, shown in Figure 5, is much more complicated. In Figure 5(a) we can see that the baseline CF method does not always hold a higher MAE. When the dataset density is over 210, the CF method obtains a lower MAE than the GIM method; afterwards, when the dataset density is over 280, the CF method obtains a lower MAE than the PF method. This phenomenon tells us that with the increase of the user's rating number the CF method's performance increases rapidly, performing even better than the GIM and the PF methods, which are rating and topic hybrid methods. The occurrence of this phenomenon may

be because the accuracy of user similarities using CF is higher than the accuracy obtained using the GIM and PF methods with a high-density database. It is also noteworthy that the PF method gets a lower MAE than the GIM method from beginning to end. If there are recommendation systems in which the prediction accuracy is critical, constructing a hybrid user model is not always useful; we can still use the classical CF method when the dataset density is high enough.

In Figure 5(b), the coverage rates for the three methods demonstrate a trend similar to the downward parabola, although the values fluctuate with a great deal. However, the three methods reach their highest coverage rates at different dataset densities. The similar downward parabola trend in the coverage rate is reasonable and reflects the real world situation. If the dataset density is very small, there may not be enough ratings to predict user ratings. As the dataset density increases, the coverage rate will also increase up to its highest value. However, afterward, if the dataset density continues to increase, the method may be incapable of dealing with the noise brought about by the high-density database; therefore, the coverage rate would begin to drop. For the baseline CF method, the highest coverage rate appears when the dataset density is at about 300. Although the coverage of the CF is pretty small at the beginning, it increases rapidly and exceeds the GIM method when the dataset density is about 170; it catches up with the PF method when the dataset density is about 270. For the GIM method, the coverage rate gets a slight increase as the dataset density increases and obtains its highest value when the dataset density is about 130, decreasing rapidly afterward. When the dataset density is over 170, the GIM method's coverage rate decreases until it is lower than the other two methods. The PF method maintains the highest performance throughout. The coverage rate increases slightly when the dataset density rises from 30 to 120, but after that it presents a highly uneven downward trend. When the dataset density is over 270, the coverage rate drops rapidly, along with the CF method.

Considering both the MAE and the coverage rate, this experiment demonstrates that the dataset density can significantly influence the performance of the three methods. When the dataset density is small, the GIM and PF outperform the classical CF method. However, the CF method can produce a lower MAE and a higher coverage rate prediction than the GIM method if the dataset density is high. The PF method obtained a relatively better result with both a low dataset density and a high dataset density. Therefore, we found that our PF method is more suitable than the GIM method when dealing with a high-density database.

5. Conclusion and Future Work

In an attempt to alleviate information overload in the domain of new product design domain, this paper proposed a probability-based hybrid user model that combined collaborative filtering and content-based filtering to implement a recommendation system. This research contributes a new method to the research community for implement recommender system when only user ratings and item topics or classes are available. Based on the extensive analysis

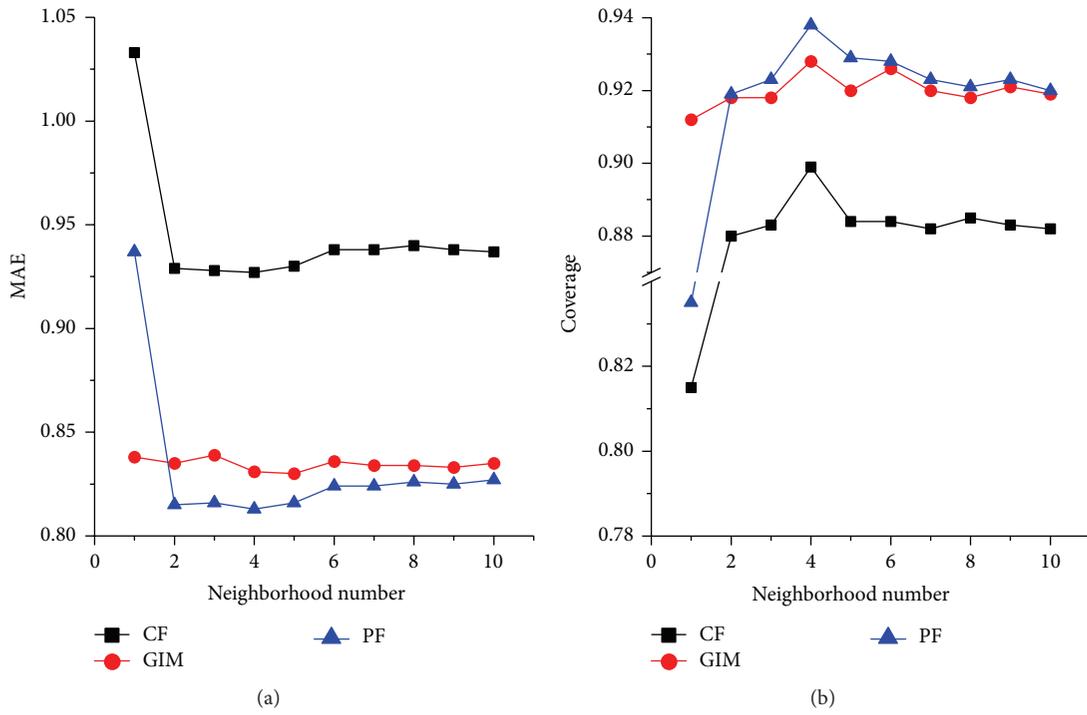


FIGURE 4: The effect of the neighborhood number on performance comparison.

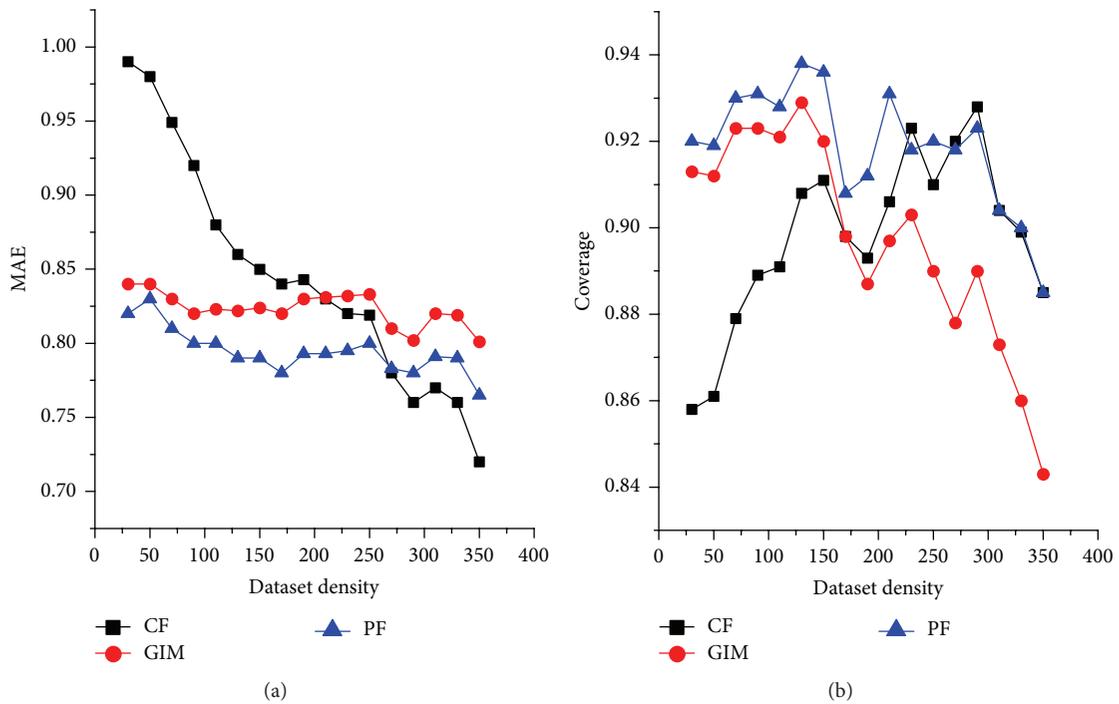


FIGURE 5: The effect of the dataset density on performance comparison.

of several experimental results, we got the conclusion that the proposed method obtains better performance in most parameter settings. Despite the progress of the method, some issues still need to be addressed in future work.

- (1) First, the performance of the proposed method dropped if the dataset density was high, which means the ability of fault tolerance needs to be promoted. Therefore, in our future work, we will develop techniques to filter the noise generated by the dense database.
- (2) Second, the performance of the proposed method was validated only by one database; more databases are necessary, especially a database from new product design domain.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and thank the strong support provided by National Ministries Projects (2015BAF18B01) and National Natural Science Foundation of China (51505032, 51375049).

References

- [1] A. A. Ammar, D. Scaravetti, and J. P. Nadeau, "Knowledge reuse: towards a design tool," in *Proceedings of the 11th International Design Conference (DESIGN '10)*, pp. 1451–1460, Dubrovnik, Croatia, May 2010.
- [2] Y. S. Kim and K.-Y. Kim, "DCR-based causal design knowledge evaluation method and system for future CAD applications," *CAD Computer Aided Design*, vol. 44, no. 10, pp. 947–960, 2012.
- [3] N. Reed, J. Scanlan, G. Wills, and S. T. Halliday, "Knowledge use in an advanced manufacturing environment," *Design Studies*, vol. 32, no. 3, pp. 292–312, 2011.
- [4] J. H. Ge, Y. P. Wang, J. Zhang, and H. Gao, "Research on method of product configuration design based on product family ontology model," *International Journal of Database Theory and Application*, vol. 6, no. 4, pp. 169–178, 2013.
- [5] J. Hao, Y. Yan, G. Wang, L. Gong, and J. Lin, "A user-oriented design knowledge reuse model," *ISRN Industrial Engineering*, vol. 2013, Article ID 928013, 10 pages, 2013.
- [6] M. Y. H. Al-Shamri and K. K. Bharadwaj, "Fuzzy-genetic approach to recommender systems based on a novel hybrid user model," *Expert Systems with Applications*, vol. 35, no. 3, pp. 1386–1399, 2008.
- [7] J. Lu, D. S. Wu, M. S. Mao, W. Wang, and G. Zhang, "Recommender system application developments: a survey," *Decision Support Systems*, vol. 74, pp. 12–32, 2015.
- [8] S. Bergamaschi, F. Guerra, and B. Leiba, "Guest editors' introduction: information overload," *IEEE Internet Computing*, vol. 14, no. 6, pp. 10–13, 2010.
- [9] S. Wan, C. Paris, and R. Dale, "Supporting browsing-specific information needs: Introducing the Citation-Sensitive In-Browser Summariser," *Journal of Web Semantics*, vol. 8, no. 2-3, pp. 196–202, 2010.
- [10] H.-N. Kim, I. Ha, K.-S. Lee, G.-S. Jo, and A. El-Saddik, "Collaborative user modeling for enhanced content filtering in recommender systems," *Decision Support Systems*, vol. 51, no. 4, pp. 772–781, 2011.
- [11] J. P. Michael and B. Daniel, "Content-based recommendation systems," in *The Adaptive Web*, pp. 325–341, Springer, Berlin, Germany, 2007.
- [12] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," in *The Adaptive Web*, pp. 291–324, Springer, Berlin, Germany, 2007.
- [13] R. Burke, "Knowledge-based recommender systems," *Encyclopedia of Library and Information System*, vol. 69, no. 2000, pp. 175–186, 2000.
- [14] S. K. Shinde and U. Kulkarni, "Hybrid personalized recommender system using centering-bunching based clustering algorithm," *Expert Systems with Applications*, vol. 39, no. 1, pp. 1381–1387, 2012.
- [15] R. Burke, "Hybrid web recommender systems," in *The Adaptive Web*, vol. 4321 of *Lecture Notes in Computer Science*, pp. 377–408, Springer, Berlin, Germany, 2007.
- [16] E. M. Stuart, D. R. David, and R. S. Nigel, "Ontology-based recommender systems," in *Handbook on Ontologies*, pp. 779–796, Springer, Berlin, Germany, 2009.
- [17] T. Bogers, M. Koolen, and I. Cantador, "Workshop on new trends in content-based recommender systems," in *Proceedings of the 8th ACM Conference on Recommender systems (RecSys '14)*, pp. 379–380, ACM, Foster City, Calif, USA, October 2014.
- [18] A. H. Mohd, A. J. Omar, and S. Ramachandram, "Collaborative filtering based recommendation system: a survey," *International Journal on Computer Science and Engineering*, vol. 4, no. 5, pp. 859–876, 2012.
- [19] C. C. Chen, Y.-H. Wan, M.-C. Chung, and Y.-C. Sun, "An effective recommendation method for cold start new users using trust and distrust networks," *Information Sciences*, vol. 224, pp. 19–36, 2013.
- [20] N. Rastin and M. Zolghadri Jahromi, "Using content features to enhance the performance of user-based collaborative filtering," *International Journal of Artificial Intelligence & Applications*, vol. 5, no. 1, pp. 53–62, 2014.
- [21] L. Yao, Q. Z. Sheng, A. Segev, and J. Yu, "Recommending web services via combining collaborative filtering with content-based features," in *Proceedings of the IEEE 20th International Conference on Web Services (ICWS '13)*, pp. 42–49, IEEE, Santa Clara Marriott, Calif, USA, July 2013.
- [22] R. Ronen, N. Koenigstein, E. Ziklik, and N. Nice, "Selecting content-based features for collaborative filtering recommenders," in *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys '13)*, pp. 407–410, Hong Kong, October 2013.
- [23] A. B. Barragáns-Martínez, E. Costa-Montenegro, J. C. Burguillos, M. Rey-López, F. A. Mikic-Fonte, and A. Peleteiro, "A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition," *Information Sciences*, vol. 180, no. 22, pp. 4290–4311, 2010.
- [24] S. Berkovsky, T. Kuflik, and F. Ricci, "Mediation of user models for enhanced personalization in recommender systems," *User*

- Modeling and User-Adapted Interaction*, vol. 18, no. 3, pp. 245–286, 2008.
- [25] S. Berkovsky, T. Kuflik, and F. Ricci, “Cross-representation mediation of user models,” *User Modeling and User-Adapted Interaction*, vol. 19, no. 1-2, pp. 35–63, 2009.
- [26] M. Degemmis, P. Lops, and G. Semeraro, “A content-collaborative recommender that exploits WordNet-based user profiles for neighborhood formation,” *User Modeling and User-Adapted Interaction*, vol. 17, no. 3, pp. 217–255, 2007.
- [27] F. Ricci, L. Rokach, and B. Shapira, “Introduction to recommender systems handbook,” in *Recommender Systems Handbook*, pp. 1–35, Springer, New York, NY, USA, 2011.
- [28] A. Moreno, A. Valls, D. Isern, L. Marin, and J. Borràs, “SigTur/E-destination: ontology-based personalized recommendation of tourism and leisure activities,” *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 633–651, 2013.
- [29] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.
- [30] E. Vozalis and K. G. Margaritis, “Analysis of recommender systems algorithms,” in *Proceedings of the 6th Hellenic European Conference on Computer Mathematics and its Applications*, Athens, Greece, 2003.
- [31] P. Resnick, N. Iacovou, M. Suchak, and P. Bergstrom, “GroupLens: an open architecture for collaborative filtering of networks,” in *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW '94)*, pp. 175–186, ACM, Chapel Hill, NC, USA, October 1994.
- [32] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, “Evaluating collaborative filtering recommender systems,” *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5–53, 2004.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

