

Research Article

Personal Recommendation Using a Novel Collaborative Filtering Algorithm in Customer Relationship Management

Chonghuan Xu^{1,2}

¹ College of Business Administration, Zhejiang Gongshang University, Hangzhou 310018, China

² Center for Studies of Modern Business, Zhejiang Gongshang University, Hangzhou 310018, China

Correspondence should be addressed to Chonghuan Xu; talentxch@gmail.com

Received 2 May 2013; Revised 21 June 2013; Accepted 6 July 2013

Academic Editor: Tinggui Chen

Copyright © 2013 Chonghuan Xu. 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 customer relationship management, more and more user recommendation technologies are used to enhance the customer satisfaction. Although there are many good recommendation algorithms, it is still a challenge to increase the accuracy and diversity of these algorithms to fulfill users' preferences. In this paper, we construct a user recommendation model containing a new method to compute the similarities among users on bipartite networks. Different from other standard similarities, we consider the influence of each object node including popular degree, preference degree, and trust relationship. Substituting these new definitions of similarity for the standard cosine similarity, we propose a modified collaborative filtering algorithm based on multifactors (CF-M). Detailed experimental analysis on two benchmark datasets shows that the CF-M is of high accuracy and also generates more diversity.

1. Introduction

With the great development of enterprise informatization, customer relationship management (CRM) has been an indispensable part of supply chain. More and more entrepreneurs and scholars generate the growing interests in applications of CRM. This trend is partly attributable to the availability of an overwhelming amount of customer transaction data and the necessary data-mining tools to obtain managerially useful insights [1]. CRM is a model for managing company's interactions with current and future customers. It aims to maximize the benefits gained from relationships with customers and enhance the enterprises competitive power. The most important expected outcomes of CRM can be listed as follows[2]: improvements in efficiency, cost reduction, improved profitability, increases in sales, enhanced customer value, customer satisfaction, and improved customer loyalty.

Nowadays, customer satisfaction becomes more crucial among researchers and practitioners alike. How to enhance the customer satisfaction, there are many approaches such as setting lower price and better quality of products, providing

better service for customers. Specifically, with the mushroomed development of E-commerce applications, the size and complexity of business websites grow rapidly. For the users of these websites it becomes increasingly difficult and time consuming to find the information or products they are looking for. As a consequence, how to efficiently help users filter out the unwanted information and find what is really useful for them is a challenging problem for customer service. Recommendation technologies are used to provide individual marketing decisions for each user. The main task of them is to recommend good products to users, and their performance metric is the number of recommendations made to users until good products are recommended, as well as the number of users that are eventually satisfied. Crucially, they do not require detailed keywords provided by users. Instead, they use the users' historical activities and possible personal profiles to uncover their preferences or potential interests. Today, some good recommendation technologies have been used to recommend books and CDs at Amazon.com, movies at Netflix.com, and news at VERSIFI Technologies [3].

With the development of the recommendation technologies, various kinds of approaches are proposed, including collaborative filtering (CF) [4, 5], content-based filtering [6, 7],

K-Nearest Neighbor (K-NN) [8–11], diffusion approach [12–14], and spectral analysis [15, 16]. CF is a class of information filtering technique which can predict what users will like according to their similarity to other users based on collecting and analyzing a large amount of information on users' behaviors, activities, or preferences. Content-based filtering method selects items based on the correlation between the content of the items and the users' preferences as opposed to a collaborative filtering system that chooses items based on the correlation between people with similar preferences. K-NN is one of the most commonly used algorithms for classifying objects based on the properties of its closest neighbors in the feature space. In K-NN, an object is classified through a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors. Diffusion approach applies the three-step mass diffusion starting from the target user on a user-object bipartite network. Spectral analysis is a new recommendation algorithm that relies on the singular value decomposition (SVD) of the rating matrix.

Due to the fact that the CF method has been the most widely and successfully used in many applications, more and more scholars devote themselves to improve this technology. In this paper, we construct a user recommendation model and present a novel method to compute the similarities among users on bipartite network. Comparing the proposed method with other standard similarity computation methods, the advantage of our method is that it takes into account the influence of each object node including popular degree, preference degree, and trust relationship. Detailed numerical analysis on two benchmark datasets, *MovieLens* and *Book-Crossing*, indicates that our modified collaborative filtering based on multifactors (CF-M) outperforms other algorithms. Specifically, it can not only provide more accurate recommendations, but also generate more diverse recommendations by precisely recommending less popular objects.

The main contribution of this paper is that we provide a better recommendation algorithm to improve the quality of individual customer service. This proposed algorithm meets high accuracy and certain diversity and also can solve the problems existed in the current CF algorithms which consider little factors of objects in the process of similarity computation. Through detailed experiment, the proposed algorithm demonstrates its superiority. A review of related work is given in Section 2. In Section 3, our recommendation model and modified collaborative filtering algorithm based on multifactors (CF-M) are described. Section 4 provides experimental results and analysis of the CF-M algorithm on two benchmark datasets. Finally, we draw conclusions in Section 5.

2. Related Work

Collaborative filtering is the most widely used technique to produce user-specific recommendations of items based on patterns of ratings or usage without need for exogenous information about either items or users. Since the method was proposed, many scholars attempt to improve it to enhance the

quality of recommendation results. As a result, it generates lots of modified CF algorithms.

Liu et al. [17] proposed a novel method to compute the similarity between congeneric nodes on bipartite network. They considered the influence of a node's degree and then presented a modified collaborative filtering (MCF) to substitute the standard cosine similarity. Yang et al. [18] proposed an approach based on the fact that any two users might have some common interest genres as well as different ones. Different from most existing methods, this approach introduced a more reasonable similarity measure metric, considering users' preferences and rating patterns. Zhao et al. [19] presented a shared collaborative filtering approach to alleviate the sparse problem. The proposed approach leveraged the data from other parties to improve CF performance and did not compromise the privacy of other parties. Bobadilla et al. [20] provided a detailed formulation of the method proposed and an extensive set of experiments and comparative results which showed the superiority of designed collaborative filtering compared to traditional collaborative filtering in (a) the number of recommendations obtained, (b) quality of the predictions, and (c) quality of the recommendations. Liu et al. [21] proposed a sequence-based trust model based on users' sequences of ratings on documents. The model considered two factors in computing the trustworthiness of users. It also enhanced the similarity of user profiles and was incorporated into a standard collaborative filtering method to discover trustworthy neighbors for making predictions. Kim et al. [22] proposed a collaborative approach to user modeling for enhancing personalized recommendations to users. Their approach first discovered some useful and meaningful user patterns and then enriched the personal model with collaboration from other similar users. López-Nores et al. [23] presented a new strategy called property-based collaborative filtering (PBCF) to address problems of recommender systems by introducing a new filtering strategy, centered on the properties that characterized the items and the users. Tsai and Hung [24] assessed the applicability of cluster ensembles to collaborative filtering recommendation. They used two well-known clustering techniques and three ensemble methods. The experimental results based on the *MovieLens* dataset showed that cluster ensembles could provide better recommendation performance than single clustering techniques in terms of recommendation accuracy and precision. Choi et al. [25] proposed a hybrid online-product recommendation method combining implicit rating-based collaborative filtering and sequential pattern analysis. They considered the objective of their research by two ways: one was to derive implicit ratings so that CF could be applied to online transaction data even when no explicit rating information was available, and the other was to integrate CF and SPA for improving recommendation quality. Dao et al. [26] proposed a new recommendation model called Context-Aware Collaborative Filtering using genetic algorithm (CACF-GA) for location-based advertising (LBA) based on both users' preferences and interaction's context. They first defined discrete contexts and then applied the concept of context similarity to conventional CF to create the context-aware recommendation model. Eckhardt [27]

proposed a collaborative filtering model which could provide clear information about preferences and then used this model as user similarity measure instead of traditional ratings-based similarity. Kant and Bharadwaj [28] developed an effective content-based filtering (CBF) by introducing an item representation scheme and fuzzy similarity measures and incorporating collaborative diverse predictions for alleviating its recommendation diversity. Lai et al. [29] proposed a hybrid personal trust model which adaptively combined the rating-based trust model and explicit trust metric to resolve the drawback caused by insufficient past rating records; after that, they presented a recommendation method based on a hybrid model of personal and group trust to improve recommendation performance. Choi and Suh [30] proposed a new similarity function in order to select different neighbors for each different target item. In the new similarity function, the rating of a user on an item was weighted by the item similarity between the item and the target item.

The target of our work is to construct a recommendation model containing an effective method to give users high accuracy and certain diversity recommendation results and also improve the quality of customer service. Finally, the experimental results show that our method is better than many other recommendation algorithms. In addition, our research result can be applied to CRM improvement or electronic commerce construction.

3. Modified Collaborative Filtering Recommendation Model

In this section, we introduce the similarity computation of the traditional collaborative filtering algorithm and then construct an effective recommendation model and derive out our modified collaborative filtering based on multifactors (CF-M).

Figure 1 shows the framework of our proposed recommendation model. In this model, we use a new similarity computation method considering more factors to compute the similarity between target user and other users and then obtain the objects collected by the similar users but not by the target user. Finally, we generate a recommendation list made up of these objects and then recommend them to the target user.

There are several phases in this framework.

- (1) *Data Preprocessing.* Data preprocessing is an important step in recommendation model. As far as we know, data-gathering methods are often loosely controlled, resulting in out-of-range values, impossible data combinations, missing values, and so forth. Analyzing data that has not been carefully screened for such problems can produce misleading results. So, in this phase, we process the primary data to fulfill the requirement of the recommendation method before similarity computation. For example, we normalize the users' ratings in order that our proposed recommendation method can compute these data directly. Since the existence of some implicit evaluations does not indicate users' preferences but meanwhile they

cannot be ignored as well, we assign some suitable values to them.

- (2) *Similarity Computation.* This phase is the key procedure. First, we give some definitions. We assume that there is a recommendation model which consists of m users and n objects, and each user has selected some objects. The relationship between users and objects can be described by a bipartite network. Let $U = \{u_1, u_2, \dots, u_m\}$ denote users set and $O = \{o_1, o_2, \dots, o_n\}$ denote objects set; the recommendation model can be fully described by an $m \times n$ adjacency matrix $A = \{a_{ij}\}$, where $a_{ij} = 1$ when object j is selected by user i ; otherwise, $a_{ij} = 0$. After that we use CF-M to compute the similarity between two users. The detailed process of this algorithm will be described in Section 3.2.
- (3) *Recommendation.* In the previous phase, we use the CF-M algorithm to compute the similarity between target user and others based on the influence of each object node including popular degree, preference degree, and trust relationship.

In this step, we calculate the comprehensive preference degree of each product unselected by the target user. Finally, the products with high comprehensive preference degree are used to compile a recommendation list in descending order. At last we recommend top L products to the target user. In general, the number L is no more than 100.

3.1. Similarity Computation of Traditional Collaborative Filtering. Traditional collaborative filtering method usually adopts the standard cosine similarity or Pearson correlation to compute the similarity between two users. For arbitrary users u_i and u_j , the number of common objects shared by them can be expressed as

$$c_{ij} = \sum_{l=1}^n a_{li} a_{lj}. \quad (1)$$

Generally, for standard cosine similarity computation, let s_{ij} denote the similarity between u_i and u_j and let $k(u_i)/k(u_j)$ denote the degree of the user u_i/u_j ; namely, how many objects are collected by this user? So we can formulate the expression as

$$s_{ij} = \frac{c_{ij}}{\sqrt{k(u_i)k(u_j)}} = \frac{\sum_{l=1}^n a_{li} a_{lj}}{\sqrt{k(u_i)k(u_j)}}. \quad (2)$$

The problem of (2) is that it has not taken into account the influence of an object's degree, so that objects with different degrees have the same contribution to the similarity. If users u_i and u_j both have selected object o_l , then they have a similar preference for object o_l .

3.2. Modified Collaborative Filtering Recommendation Method. As we know, in real recommender system, the similarity computation between two users is not simple but influenced by many factors. So we need to improve the traditional

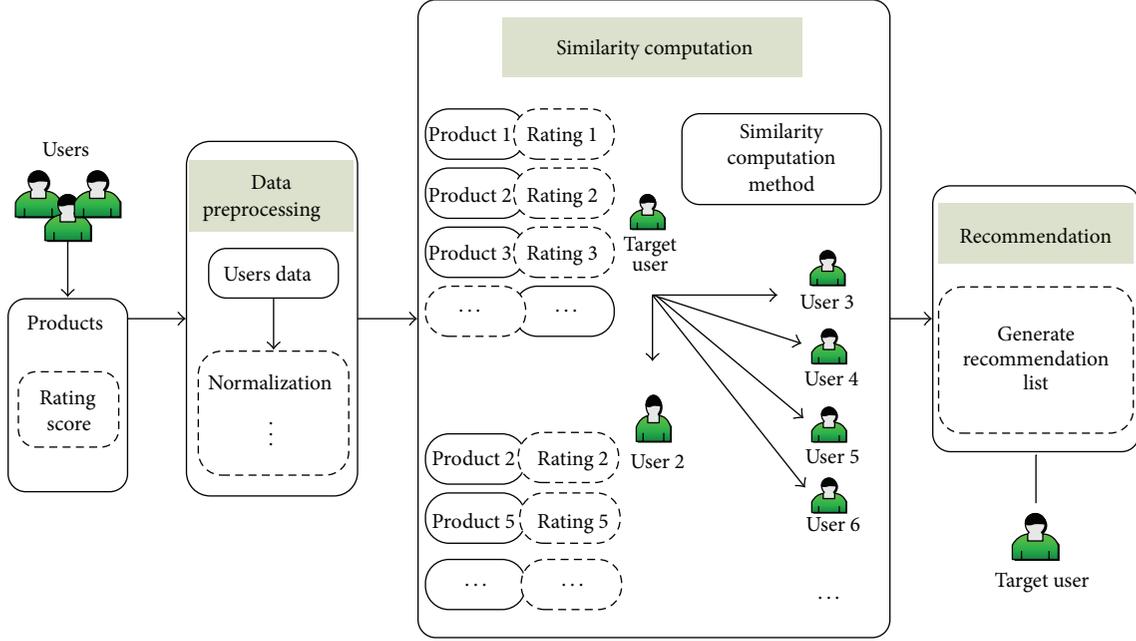


FIGURE 1: The framework of recommendation model based on CF-M.

collaborative filtering method in order to fit the complex conditions. Generally, the similarity between two users should be somewhat relative to their degrees, preference degree, and trust relationship. According to these features, we propose a modified collaborative filtering algorithm based on multifactors (CF-M) including all factors mentioned previously to increase the accuracy and certain diversity of the recommendation results.

Through analyzing these factors, we can make the conclusion that each object node's degree and preference degree are related to its popular degree and corresponding users' comments or ratings, respectively. For preference degree computation, we need to make quantification of them when encountering users' comments and then distinguish the degrees of preferences. In other cases, we can divide the degrees of preferences directly. Trust relationship is derived from two users' past ratings on corated products by adopting Hwang and Chen's [31] trust computation method. In other words, trust relationship relates to users' evaluations or scores. Generally, a recommender is more trustworthy if he or she has contributed more precise predictions than other users do.

We assume that the similarity computation on user-object bipartite networks is affected by an influence degree that is proportional to $[(1 - (|v_{li} - v_{lj}|/M))/k(o_l)]^\alpha$, with α being a freely adjustable parameter. Accordingly, the contribution of object o_l to the similarity s_{ij} should be negatively correlated with its degree $k(o_l)$ and positively correlated with its preference degree and trust relationship. It means that it is not very meaningful if two users both select a popular object, while if a very unpopular object is simultaneously selected by two users, there must be some common tastes shared by these two users.

So we suppose that the object o_l contribution to s_{ij} is inversely proportional to $k(o_l)$ and directly proportional to $|v_{li} - v_{lj}|$. The formulation of s_{ij} can be expressed as

$$s_{ij} = \frac{1}{\sqrt{k(u_i)k(u_j)}} \sum_{l=1}^n a_{li}a_{lj} \left[\frac{(1 - (|v_{li} - v_{lj}|/M))}{k(o_l)} \right]^\alpha. \quad (3)$$

M is the range of the rating score, which equals the difference of the maximum and minimum rating scores. v_{li}/v_{lj} represents the preference degree that object o_l obtained from user u_i/u_j . $k(o_l)$ denotes the degree of object o_l ; namely, how many users select this object? $k(u_i)$ denotes the degree of the user u_i ; namely, how many objects are selected by this user?

Despite the users similarity computation, we can find the products unselected by target user but selected by test users who have much similarity to target user. Then we may predict the comprehensive preference degree of these products. Let p_{ij} represent the comprehensive preference degree of object o_j obtained from target user u_i . The formulation of p_{ij} can be expressed as

$$p_{ij} = \sum_{l=1, l \neq i}^m s_{il}a_{jl}. \quad (4)$$

In the process of recommendation, we get the elements of p_{ij} uncollected by target user and then sort them in descending order, as target user prefers the objects in the top, so we recommend top L objects to this user.

The pseudocode of the modified collaborative filtering algorithm based on multifactors (CF-M) algorithm is shown in Pseudocode 1.

```

Algorithm CF-M: Calculating the similarity between users
begin
  get A;
  n = size(A, 2), m = size(A, 1);
  parameter ar;
  preference degree v();
  range of the rating score M;
  S = zeros(m, m), o = sum(A), u = sum(A');
  for i = 1 : m
    for j = 1 : m
      x = (u(i) * u(j)) ^ (-0.5);
      for z = 1 : n
        y = y + a(i, z) * a(j, z) * ((1 - abs(v(i, z) - v(j, z)))/M)/o(z) ^ ar,
      end
      S(i, j) = x * y;
      x, y = 0;
    end
  end
End

```

PSEUDOCODE 1

3.3. Recommendation Performance Metrics. In this paper, we adopt some standard metrics to measure the accuracy and diversity of the proposed method, in which accuracy is the most important aspect in evaluating the recommendation algorithmic performance.

Five metrics: ranking score, precision, recall, intrasimilarity, and Hamming distance. The first three are used to test accuracy and the rest are used to test diversity. The detailed descriptions of these metrics are as follows.

- (1) Ranking score is used for an arbitrary user u_i if the recommendation o_j is in the test set (according to the training set, o_j is an unselected object for u_i) and ranked in R_{ij} position in the ordered recommendation list L_i . We can formulate the expression as $r_{ij} = R_{ij}/L_i$. For example, if the length of L is 200, namely, there are 200 unselected objects for u_i and o_j is the 10th from the top, we say that the position of o_j is 10/200, denoted by $r_{ij} = 0.05$. The average of r_{ij} of the overall user-object pairs in the test set defines the average ranking score $\langle r \rangle$, which can be used to evaluate the algorithmic accuracy. The smaller the ranking score is, the higher the algorithmic accuracy is.
- (2) Precision is defined as the ratio of the number of recommended objects collected by users appearing in the test set to the total number of recommended objects. This measure is used to evaluate the effectiveness of a given recommendation list. The precision can be formulated as a/L , in which a represents the number of recommended products collected by users appearing in test set and L is the total number of recommended products.
- (3) Recall is defined as the ratio of the number of recommended objects collected by users appearing in the test set to the total number of objects actually

collected by users. The larger recall corresponds to the better performance. The recall can be formulated as a/M , in which a represents the number of recommended products collected by users appearing in test set and M is the total number of users' actual buying.

- (4) Intrasimilarity evaluates the similarities between objects inside users' recommendation lists. A good recommendation algorithm is expected to give fruitful recommendation results and has the ability to guide or help the users to exploit their potential interest fields. Therefore, it calls for a lower intrasimilarity. There are many similarity metrics between objects. Here we adopt the widely used one, that is, cosine similarity to measure objects' similarity. For two objects o_t and o_k , their similarity is defined as

$$S_{tk} = \frac{1}{\sqrt{k(o_t)k(o_k)}} \sum_{l=1}^m a_{tl}a_{kl}. \quad (5)$$

For an arbitrary user u_i , the number of recommendation objects is L . Firstly, we need to calculate $L(L-1)/2$ couple of objects' similarity and then average these values to get $I_i = \langle S_{tk} \rangle$. Finally, we use the mean value of I of the overall users to measure the diversity in recommendation lists.

- (5) Hamming distance can measure the strength of personalization. If the overlapped number of objects in u_i and u_j 's recommendation lists is Q , their Hamming distance is

$$H_{ij} = 1 - \frac{Q}{L}. \quad (6)$$

Generally speaking, a more personalized recommendation list should have long Hamming distances to other lists. Accordingly, we use the mean value

of Hamming distance $S = \langle H_{ij} \rangle$, averaged over all the user-user pairs, to measure the strength of personalization.

4. Experimental Results and Analysis

To test the recommendation algorithmic performance, we use two benchmark datasets. The *MovieLens* [32] dataset consists of 1682 movies and 943 users. Each user has rated at least 20 movies by using a discrete number in the scale of 1 to 5. The original data contains 100,000 ratings. In the dataset, there are three kinds of information tables: demographic information about the users, information about the items (movies), and the score about the movies. The *Book-Crossing* dataset [33] contains 278,858 users (anonymized but with demographic information) providing 1,149,780 ratings (explicit/implicit) about 271,379 books. Ratings (Book-Rating) are either explicit, expressed on a discrete number in the scale of 1 to 10 (higher values denoting higher appreciation), or implicit, expressed by 0.

In our experiment, we need to preprocess the datasets. For *MovieLens*, only the links with ratings no less than 3 are considered and $v_{li} = \{3, 4, 5\}$. For *Book-Crossing*, only the links with ratings no less than 5 or equal to 0 are considered and $v_{li} = \{10, 9, 8, 7, 6, 5, 1\}$. We divide each processed dataset into two parts: the training set which contains 80% of the data and the remaining 20% of the data for the test.

Firstly, we need to predict the range of the optimal values of α in order to reduce the computational costs in determining the optimal value through our approach. According to some works in the literature on CF approaches, we predict that our optimal values of α are located in the range of 1 to 2. To find the optimal value of parameter α rapidly, we execute the iterative computation based on the strategy of binary search. In the process of iterative computation, we set the interval between α as 0.01. All these computational definitions and steps lead to lower computational costs. Figure 2 shows the algorithmic accuracy, measured by the ranking score, as a function of α . We note that for the two benchmark datasets the best performance of this algorithm occurs around $\alpha = 1.86$. Certainly, people can adjust the parameter's value by themselves in practice.

Recall and precision can be used to realize the balance of two competitive factors: cost and efficiency. The efficiency can be improved by increasing the number of recommended products; however, the cost is increasing at the same time. That is to say, the cost can be decreased by reducing the recommendations, while the efficiency may be decreased correspondingly. At a certain length of recommendation list L , precision tests whether the cost is deserved or necessary, while recall tests whether the efficiency is sufficient. Based on these two measures, one can find a certain L as a tradeoff for cost and efficiency. In general, the number of recommended L is no more than 100.

Figure 3 shows the precision and recall in different value of parameter α . What we can know from this figure is that the algorithm reaches the highest precision and maximum recall when the parameter $\alpha = 1.86$ for the *MovieLens* dataset.

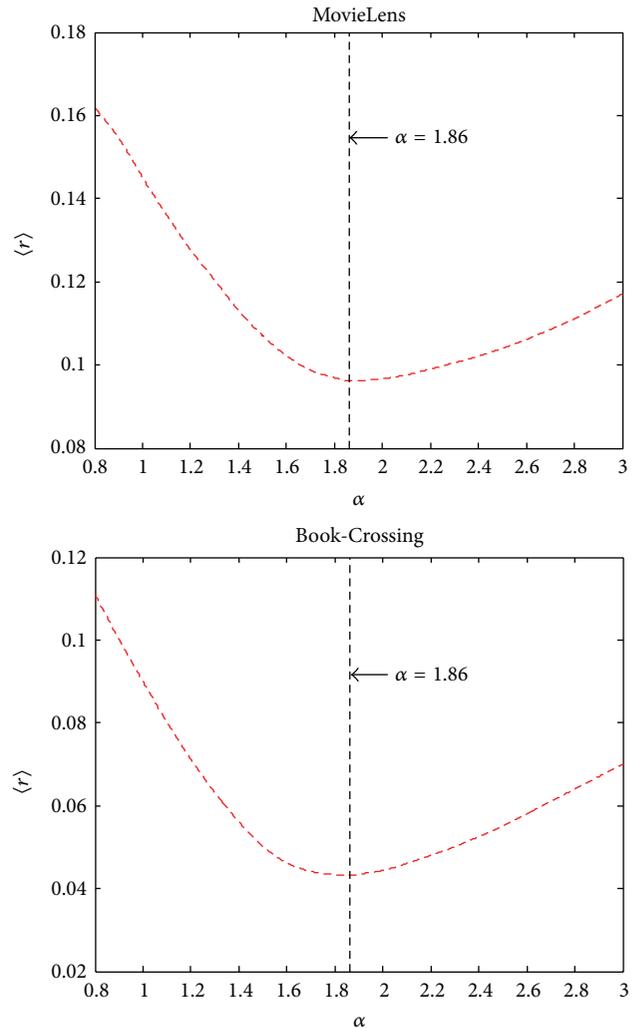


FIGURE 2: The effect of parameter α on this algorithmic accuracy. The optimal parameter α for *MovieLens* and *Book-Crossing*, corresponding to the minimal $\langle r \rangle$, is equal to 1.86. Recommendation number $L = 50$. Present results are obtained by averaging over five independent 80% versus 20% divisions.

In addition, the precision reaches a good level if $\alpha \in (1.84, 1.88)$ and so does the recall.

For *Book-Crossing* dataset, what we can know from Figure 4 is that the algorithm reaches the highest precision and maximum recall when the parameter α is also equal to 1.86. Furthermore, the precision and recall reach a good level if $\alpha \in (1.84, 1.88)$.

After that, we compare CF-M with three other widely used recommendation algorithms: CF, MCF, and NBI [34] in all five metrics. Different from CF and MCF, NBI is a diffusion algorithm based on homogeneous diffusion process on networks; that is, each object distributes its resource to its neighbors equally. In addition, it has been demonstrated to be more accurate than the classical CF algorithm, with lower computational complexity. We summarize the algorithmic performance in Table 1 for *MovieLens* and Table 2 for *Book-Crossing*.

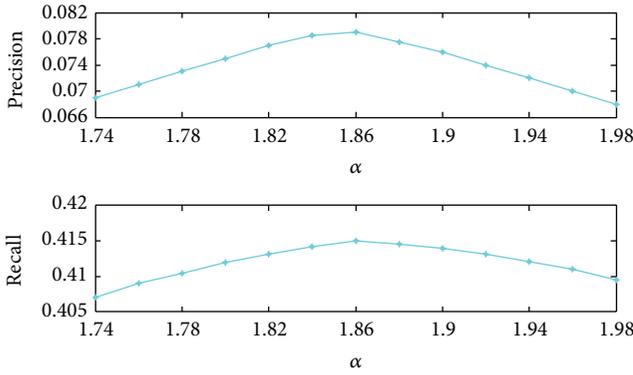


FIGURE 3: The effect of parameter α on CF-M with recommendation number $L = 50$. For *MovieLens*, the precision is at its highest at about $\alpha = 1.86$, at almost the same point where the recall achieves its maximum. Present results are obtained by averaging over four independent 80% versus 20% divisions.

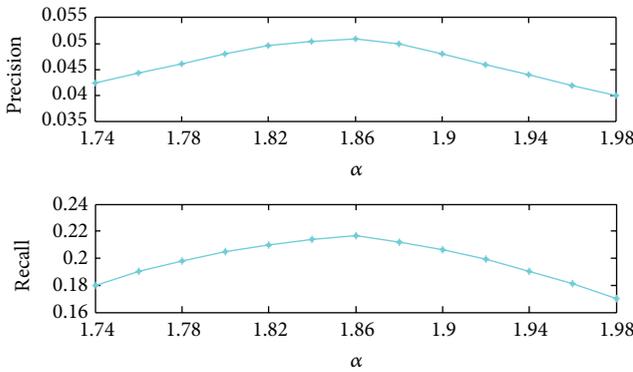


FIGURE 4: The effect of parameter α on CF-M with recommendation number $L = 50$. For *Book-Crossing*, the precision is at its highest at about $\alpha = 1.86$, at almost the same point where the recall achieves its maximum. Present results are obtained by averaging over four independent 80% versus 20% divisions.

Comparing CF-M with the standard CF, as is seen in Table 1 in the condition of recommendation number $L = 50$, the ranking score can be further reduced by 23.2%, and with MCF the ranking score can be reduced by 12.7%. Similarly, our algorithm has lower ranking score than NBI algorithms. For the rest of metrics, our algorithm is also the best. Although *Book-Crossing* dataset is similar to *MovieLens*, it is much sparse. So we set the number of recommended products no less than 50.

Table 2 shows that our algorithm exceeds the other three algorithms in all the five criterions: lower ranking score, higher precision, bigger recall, lower intrasimilarity, and larger Hamming distance.

CF-M algorithm adjusts the accuracy via parameter α . When $\alpha < 0$, the comprehensive preference degree of each product is inversely proportional to $|v_{i_i} - v_{i_j}|$ and directly proportional to $k(o_l)$. Thus, the algorithm tends to recommend popular products to users. But it is not what people want. On the other hand, when $\alpha > 0$, the comprehensive

TABLE 1: Algorithmic performance for *MovieLens* dataset. The ranking score, precision, recall, intrasimilarity, and hamming distance are corresponding to $L = 30, 40, 50$. And the value of parameter α is 1.86. Each number presented in this table is obtained by averaging over five runs, each of which has an independently random division of training set and test.

Algorithms	Ranking score	Precision	Recall	Intrasimilarity	Hamming distance
$L = 30$					
CF	0.148	0.077	0.321	0.316	0.704
MCF	0.131	0.087	0.360	0.304	0.751
NBI	0.120	0.089	0.379	0.291	0.778
CF-M	0.116	0.097	0.387	0.275	0.798
$L = 40$					
CF	0.137	0.071	0.332	0.328	0.698
MCF	0.121	0.080	0.373	0.317	0.743
NBI	0.112	0.081	0.392	0.303	0.771
CF-M	0.107	0.088	0.401	0.286	0.790
$L = 50$					
CF	0.125	0.066	0.343	0.342	0.692
MCF	0.110	0.072	0.385	0.330	0.735
NBI	0.101	0.073	0.406	0.315	0.762
CF-M	0.096	0.079	0.415	0.297	0.781

TABLE 2: Algorithmic performance for *Book-Crossing* dataset. The ranking score, precision, recall, intrasimilarity, and hamming distance are corresponding to $L = 50, 60, 70$. And the value of parameter α is 1.86. Each number presented in this table is obtained by averaging over five runs, each of which has an independently random division of training set and test.

Algorithms	Ranking score	Precision	Recall	Intrasimilarity	Hamming distance
$L = 50$					
CF	0.056	0.039	0.181	0.374	0.519
MCF	0.052	0.044	0.192	0.338	0.547
NBI	0.049	0.047	0.204	0.316	0.613
CF-M	0.043	0.051	0.217	0.263	0.764
$L = 60$					
CF	0.046	0.037	0.201	0.395	0.511
MCF	0.044	0.042	0.214	0.352	0.536
NBI	0.042	0.045	0.233	0.331	0.604
CF-M	0.037	0.048	0.248	0.278	0.752
$L = 70$					
CF	0.042	0.032	0.228	0.421	0.497
MCF	0.040	0.036	0.243	0.375	0.522
NBI	0.034	0.041	0.262	0.349	0.587
CF-M	0.030	0.044	0.296	0.293	0.731

preference degree of each product is inversely proportional to $k(o_l)$ and directly proportional to $|v_{i_i} - v_{i_j}|$. In this case, the algorithm tends to recommend unpopular products to users. The experimental results show that it is more suitable

to recommend unpopular and reliable products to users in fact. Finally, for an online recommender system, we need to consider the processing time and memory consumption of its recommendation algorithm. If we denote $\langle k_u \rangle$ and $\langle k_o \rangle$ by the average degree of users and objects on the bipartite network, the computational complexity of CF-M is $O(m^2 \langle k_u \rangle + mn \langle k_o \rangle)$ and the memory store is m^2 , which is the same as CF and MCF. For NBI algorithm, its computational complexity is $O(m \langle k_u^2 \rangle + mn \langle k_u \rangle)$ and the memory store is n^2 . When the number of objects is much larger than the number of users, the CF-M may be more practicable; otherwise NBI may be more practicable. Although the accuracy and diversity of the proposed method CF-M are increased, it is still confronted with the cold-start problem.

5. Conclusions

Recommendation model predicts users' potential future likes and interests by using users' past preferences data. An excellent recommendation algorithm meets high accuracy and certain diversity and can enhance the quality of personalized service. Since the collaborative filtering approach was proposed, it has attracted much attention for its convenience, high accuracy, and certain diversity as well as low computational complexity.

In this paper, we construct an effective recommendation model in order to improve the current recommender system for better customer service. We analyze the collaborative filtering algorithm and propose a modified one based on multiple influence factors. We compute the similarity between two users on bipartite network. Comparing the proposed method with other standard similarity computation methods, the key feature or superiority is that our method takes into account the influence of each object node including popular degree, preference degree, and trust relationship. All the factors are governed by a parameter which is derived from optimal value calculation. Certainly, people can adjust the parameter's value by actual requirement. Detailed numerical analysis on two benchmark datasets, *MovieLens* and *Book-Crossing*, indicates that the presented algorithm is of high accuracy and also generates certain diversity.

Concerning future work, we will improve our recommendation model and pay more attention to algorithmic structure. The research covers the following aspects: how to compute the similarity while engaging in the user context factors; how to introduce other technologies to improve the accuracy and diversity; how will the recommendation algorithm keep its robustness when meeting hostile attack; and how to alleviate the influence of the cold-start problem.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grants nos. 71071140 and 71071141), Natural Science Foundation of Zhejiang Province (Grant no. LQ12G01007); and Ministry of Education, Humanities and Social Sciences project (Grant no. 13YJCZH216).

References

- [1] S. Singh, B. P. S. Murthi, and E. Steffes, "Developing a measure of risk adjusted revenue (RAR) in credit cards market: implications for customer relationship management," *European Journal of Operational Research*, vol. 224, no. 2, pp. 425–434, 2013.
- [2] B. Öztayşi, T. Kaya, and C. Kahraman, "Performance comparison based on customer relationship management using analytic network process," *Expert Systems with Applications*, vol. 38, no. 8, pp. 9788–9798, 2011.
- [3] 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.
- [4] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, no. 1, pp. 5–53, 2004.
- [5] Z. Huang, H. Chen, and D. Zeng, "Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 116–142, 2004.
- [6] M. Balabanovi and Y. Shoham, "Fab: content-Based, collaborative recommendation," *Communications of the ACM*, vol. 40, no. 3, pp. 66–72, 1997.
- [7] M. J. Pazzani, "A framework for collaborative, content-based, and demographic filtering," *Artificial Intelligence Review*, vol. 13, no. 5-6, pp. 393–408, 1999.
- [8] Y. M. Yang, "An evaluation of statistical approaches to text categorization," *Information Retrieval*, vol. 1, no. 1-2, pp. 69–90, 1999.
- [9] M. L. Zhang and Z. H. Zhou, "ML-kNN: a lazy learning approach to multi-label learning," *Pattern Recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.
- [10] J.-Y. Jiang, S.-C. Tsai, and S.-J. Lee, "FSKNN: multi-label text categorization based on fuzzy similarity and k nearest neighbors," *Expert Systems With Applications*, vol. 39, no. 3, pp. 2813–2821, 2012.
- [11] G. Adomavicius and Z. Jingjing, "Stability of Recommendation Algorithms," *ACM Transactions on Internet Technology*, vol. 10, no. 4, 2012.
- [12] T. Zhou, L.-L. Jiang, R.-Q. Su, and Y.-C. Zhang, "Effect of initial configuration on network-based recommendation," *Europhysics Letters*, vol. 81, no. 5, Article ID 58004, 2008.
- [13] Y.-C. Zhang, M. Medo, J. Ren, T. Zhou, T. Li, and F. Yang, "Recommendation model based on opinion diffusion," *Europhysics Letters*, vol. 80, no. 6, Article ID 68003, 2007.
- [14] R. Liu, J.-G. Liu, C.-X. Jia, D. Sun, and B.-H. Wang, "Personal recommendation via unequal resource allocation on bipartite networks," *Physica A*, vol. 389, no. 16, pp. 3282–3289, 2010.
- [15] S. Maslov and Y.-C. Zhang, "Extracting hidden information from knowledge networks," *Physical Review Letters*, vol. 87, no. 24, Article ID 248701, 2001.
- [16] J. Ren, T. Zhou, and Y.-C. Zhang, "Information filtering via self-consistent refinement," *Europhysics Letters*, vol. 82, no. 5, Article ID 58007, 2008.
- [17] R. Liu, C.-X. Jia, T. Zhou, D. Sun, and B.-H. Wang, "Personal recommendation via modified collaborative filtering," *Physica A*, vol. 388, no. 4, pp. 462–468, 2009.
- [18] J.-M. Yang, K. F. Li, and D.-F. Zhang, "Recommendation based on rational inferences in collaborative filtering," *Knowledge-Based Systems*, vol. 22, pp. 105–114, 2009.

- [19] Y. Zhao, X. Feng, J. Li, and B. Liu, "Shared collaborative filtering," in *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys '11)*, pp. 29–36, ACM, 2011.
- [20] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, "Generalization of recommender systems: collaborative filtering extended to groups of users and restricted to groups of items," *Expert Systems with Applications*, vol. 39, no. 1, pp. 172–186, 2012.
- [21] D.-R. Liu, C.-H. Lai, and H. Chiu, "Sequence-based trust in collaborative filtering for document recommendation," *International Journal of Human-Computer Studies*, vol. 69, no. 9, pp. 587–601, 2011.
- [22] 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.
- [23] M. López-Nores, Y. Blanco-Fernández, J. J. Pazos-Arias, and A. Gil-Solla, "Property-based collaborative filtering for health-aware recommender systems," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7451–7457, 2012.
- [24] C.-F. Tsai and C. Hung, "Cluster ensembles in collaborative filtering recommendation," *Applied Soft Computing*, vol. 12, no. 4, pp. 1417–1425, 2012.
- [25] K. Choi, D. Yoo, G. Kim, and Y. Suh, "A hybrid online-product recommendation system: combining implicit rating-based collaborative filtering and sequential pattern analysis," *Electronic Commerce Research and Applications*, vol. 11, no. 4, pp. 309–317.
- [26] T. H. Dao, S. R. Jeong, and H. Ahn, "A novel recommendation model of location-based advertising: context-Aware Collaborative Filtering using GA approach," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3731–3739, 2012.
- [27] A. Eckhardt, "Similarity of users' (content-based) preference models for Collaborative filtering in few ratings scenario," *Expert Systems with Applications*, vol. 39, no. 14, pp. 11511–11516, 2012.
- [28] V. Kant and K. K. Bharadwaj, "Enhancing recommendation quality of content-based filtering through collaborative predictions and fuzzy similarity measures," *Procedia Engineering*, vol. 38, pp. 939–944, 2012.
- [29] C.-H. Lai, D.-R. Liu, and C.-S. Lin, "Novel personal and group-based trust models in collaborative filtering for document recommendation," *Information Sciences*, vol. 239, no. 1, pp. 31–49, 2013.
- [30] K. Choi and Y. Suh, "A new similarity function for selecting neighbors for each target item in collaborative filtering," *Knowledge-Based Systems*, vol. 37, pp. 146–153, 2013.
- [31] C.-S. Hwang and Y.-P. Chen, "Using trust in collaborative filtering recommendation," in *New Trends in Applied Artificial Intelligence*, pp. 1052–1060, Springer, Berlin, Germany, 2007.
- [32] MovieLens dataset, <http://www.grouplens.org>.
- [33] Book-Crossing dataset, <http://www.informatik.uni-freiburg.de/~cziegler/BX/>.
- [34] T. Zhou, J. Ren, M. Medo, and Y.-C. Zhang, "Bipartite network projection and personal recommendation," *Physical Review E*, vol. 76, no. 4, Article ID 046115, 2007.



Hindawi

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

