Intelligent Automation And Soft Computing, 2019
Copyright © 2019, TSI® Press
Vol. 25, no. 3, 595–604
https://doi.org/10.31209/2019.100000114

A Recommendation Approach Based on Product Attribute Reviews: Improved Collaborative Filtering Considering the Sentiment Polarity

Min Cao[1], **Sijing Zhou**[1] **and Honghao Gao**[1,2,3]

[1]School of Computer Engineering and Science, Shanghai University, Shanghai, China;
[2]Computing Center, Shanghai University, Shanghai, China;
[3]Shanghai Key Laboratory of Intelligent Manufacturing and Robotics, Shanghai, China;

ABSTRACT
Recommender methods using reviews have become an area of active research in e-commerce systems. The use of auxiliary information in reviews as a way to effectively accommodate sparse data has been adopted in many fields, such as the product field. The existing recommendation methods using reviews typically employ aspect preference; however, the characteristics of product reviews are not considered adequate. To this end, this paper proposes a novel recommendation approach based on using product attributes to improve the efficiency of recommendation, and a hybrid collaborative filtering is presented. The product attribute model and a new recommendation ranking formula are introduced to implement recommendation using reviews. Experimental results show that the proposed method outperforms baselines in terms of sparse data.

KEY WORDS: Product Recommendation, Reviews, Hybrid Collaborative Filtering, Product Attributes.

1 INTRODUCTION

RECOMMENDATION system is dedicated to discovering user preferences and recommending suitable items (Chen, Chen, and Wang, 2015; Lu, Wu, Mao, Wang, and Zhang, 2015). Because of the increasing users and products, the expression of user preference is not limited to user-item-rating-matrix, such as aspect preference in recommendation methods using reviews (Chen et al., 2015; Lu et al., 2015). However, aspect preference, as the mainstream expression in personalized recommendation using reviews, does not consider the characteristics of product reviews for more precise user preferences.

Researchers tend to use clustering technology to extract centralized aspects from reviews, so the implicit condition is that reviews are not scattered. Product reviews span multiple categories and are not concentrated, which leaves out some valuable information through aspect preference. (Chen et al., 2015; Lei, Qian, and Zhao, 2016; Lu et al., 2015; Ma, Chen, and Wei, 2017; Zhao et al., 2016). Product attributes (including product performance, appearance, quality and other aspects) are considered to establish recommendation model because they can be used for all kinds of products and are often commented by users. Furthermore, these attributes are proved to affect consumers' desire for consumption (Alton and Snehasish, 2016; Su, Ewa, and Edward, 2018).

Recommendation model based on aspect preference was once focused on the calculation of weight values, such *aspect need* and *aspect importance* (Chen et al., 2015; Ma et al., 2017; Zhao et al., 2016). Lacking product perspective, user preference simulation is not comprehensive, which adversely affects the recommendation performance. Upon the introduction of matrix factorization theory, modeling method based on the multi-irrelevant-model form is an idea worth learning (Ma et al., 2017; Zhao et al., 2016). Meanwhile, the addition of product perspective makes the new model need an applicable formula to generate recommendation results.

In response to the above issues, this paper presents a hybrid collaborative filtering approach based on product attributes, namely, product attribute collaborative filtering (PACF), to obtain accurate user preferences. Then, we use constant product attributes to perform initial preprocessing of reviews. The preprocessing not only addresses the characteristics of product reviews but also lays the foundation of an accurate recommendation model (Lei et al., 2016). To implement the model from the two angles of user and product, a product attribute model (PAM) based on the

CONTACT Honghao Gao gaohonghao@shu.edu.cn
© 2019 TSI® Press

matrix factorization vector multiplication idea is discussed. Subsequently, important elements of the *product attribute weight* and *product attribute score* for the PAM are defined for the users and products, respectively. An applicable formula is sought to construct a new model to integrate these factors; thus, a new hybrid collaborative filtering formula Γ_{PAM} is proposed to generate the recommendation results for the PAM.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 provides formal definitions. Section 4 introduces the model PAM and the Γ_{PAM} formula to achieve PACF. Section 5 discusses the experimental analysis, and Section 6 presents conclusions and addresses future research directions.

2 RELATED WORK

E-SERVICE personalization service technology is represented by the recommendation system based on user-item ratings (Chen et al., 2015). To improve the user experience and recommendation performance, a variety of valuable data such as reviews are introduced into the recommendation (Lei et al., 2016). Prior work in the area of recommendation methods based on reviews can be further divided into using reviews as additional information to achieve accurate ratings and modeling reviews to achieve virtual ratings (Chen et al., 2015; Li, Liu, Cao, Liu, and Li, 2017; Lu et al., 2015). Regarding the research issues of concern, the related literature focuses on the modeling from reviews to virtual ratings on the product attribute perspective.

The concept of product attributes can be traced back to the preference-based product ranking algorithm. User preferences can be elicited in the form of a weight criterion assigned to each of the attributes (Chen et al., 2015). The use of reviews as a virtual rating changes the presentation of product attributes. Ma et al. (2017) combined sentiment analysis to calculate two measures based on the user's relevance to the average user, namely, *aspect need* and *aspect importance*. Meng et al. (2016) presented weight-based matrix factorization (WMF), which captured the weight value of each app for the specific user by the TF-IDF algorithm. Wang et al. (2017) proposed VFDSR, which demonstrated user preferences on a personalized distribution of each feature from a value standpoint. Weight values relying solely on product attributes could not adequately model user preferences. Li et al. (2017) integrated tag, topic, co-occurrence and popularity factors derived by the relational topic model and factorization machines to recommend Web APIs. Furthermore, the above review analysis methods are mainly associated with the restaurant and movie fields, while product recommendation favors feature-mentioning measures (Jiang, Cai, Olle and Qin, 2015).

The use of only the description of weight values using product attributes cannot provide a satisfactory user experience (Chen et al., 2015). Researchers have introduced matrix factorization theory to establish multi-irrelevant-recommendation models (Ma et al., 2017; Yu, Xu, Yang, and Guo, 2016; Zhao et al., 2016). Ma et al. (2017) calculated the multiplicative product of the *aspect importance* and *aspect need* through the relevance based on a regression model. Zhao et al. (2016) derived the product adopter model from online reviews. The product adopter model can be divided into a user preference model and a product distribution model. Every product was given a vector with six elements by the product distribution model. This vector essentially characterized the demographics of the product by the users who have actually used it.

In contrast with the above works, we consider the characteristics of product reviews and take advantage of irrelevant and multi-perspective models to improve the recommendation performance. The approach proposed in this paper is intended to complement the existing product recommendation approaches using reviews.

3 FOUNDATION

IN this section, we describe the symbols used in our research and the problem to be solved.

3.1 Formal Definition

Aimed at the characteristics of product reviews, product attributes are first fixed to facilitate feature consistency. Then, sentiment polarity is introduced to accurate user preferences. Product attributes can be defined in terms of the quality, performance, appearance and other aspects. Positive polarity and negative polarity are embodied in sentiment polarity. For example, 'good packaging' means the sentiment positive polarity of the product attribute 'packaging'.

The data preprocessing in this paper can be described as introducing a static product attribute parameter and a sentiment polarity parameter. The specific symbols are clearly defined in Table 1.

Table 1. Symbols and Their Meanings

Symbol	Meaning
$R=\{r_1, r_2,\ldots, r_{\|R\|}\}$	Review set.
$U=\{u_1, u_1,\ldots, u_{\|U\|}\}$	User set.
$P=\{p_1, p_2,\ldots, p_{\|p\|}\}$	Product set.
$PA=\{pa_1, pa_2,\ldots, p_{\|pa\|}\}$	Product attribute set; the specific definition is shown in Table 2.
$F_k=\left\{f_1, f_2,\ldots f_{\|F_k\|}\right\}$	F_k is a set of feature words f_i of the product attribute pa_k. The specific definition is shown in Table 2.
$\delta_i\in\{1,-1\}$	δ_i is the sentiment polarity corresponding to the product attribute feature word f_i. Within the set, -1 is a negative sentiment, and 1 is a non-negative sentiment (including positive and neutral).

The product attribute parameter is fixed through building the dictionary, shown in Table 2. Through four basic characteristics of all commodities, we can process data uniformly avoid any regions of sparsity resulting from the user not reviewing a product feature. When $r_i \in R$ matches the product attribute pa_k's feature words f_i, δ_i is obtained through the sentiment analysis technique. For example, if {*food is fresh* $\in$ R} and *fresh*$\in F_{Performance}$, then $\delta_i = 1$. The structured two-tuple (pa_k,δ_i) is acquired after the preprocessing of review in the research, while $pa_k \in PA$, and $\delta_I \in \{1,-1\}$.

Table 2. Product Attributes and Feature Words

Symbol	Meaning
PA	*PA = {Quality, Service, Performance, Package}*
$F_{Quality}$	$F_{Quality}$ = *{nature, product, greener, brand, etc.}*
$F_{Service}$	$F_{Service}$ = *{communication, efficient, responsive, etc.}*
$F_{Performance}$	$F_{Performance}$ = *{fresh, flavor, awful, taste, etc.}*
$F_{Package}$	$F_{Package}$ = *{delivery, ship, on-time, speed, etc.}*

3.2 Technology Overview

Integrating the valuable information embedded in reviews not only promotes the user experience in the recommendation system but also improves the recommendation performance (Chen et al., 2015). A virtual rating can be generated through users' implicit preference information from reviews. The research problem is to address the following challenges, as shown in Figure 1. First, how do we reliably model inference user preferences from two-tuples (pa_k,δ_i)? Second, how do we effectively incorporate product attribute information to generate recommendation results? This problem is based on the relevance of the user model and product model.

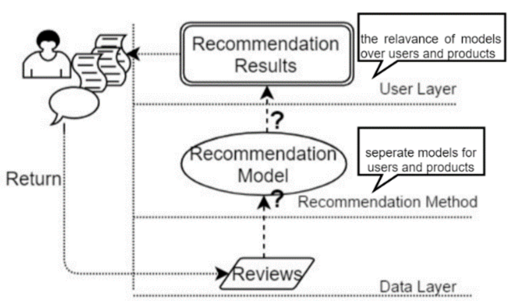

Figure 1. The problem of our recommender approach research.

The recommendation method needs structured data. In general, product reviews have a variety of content and a wide range of features. To accommodate the characteristics of product reviews, we consider constant product attributes and introduce emotional polarity. Under this condition, PACF is proposed as an applicable approach that simulates user preferences more fully in the product field to address the above issues. The PAM and calculation formula Γ_{PAM} are included. Based on the matrix factorization vector multiplication idea, the *product attribute weight* and *product attribute score* employed in the PAM model are proposed from the perspective of users and products, respectively. Multi-perspective combination leads to more accurate recommendation. A new hybrid collaborative filtering formula Γ_{PAM} is proposed for the PAM model to generate the recommendation. The relevance between the user model and product model is established through a shopping records.

4 PRODUCT ATTRIBUTE COLLABORATIVE FILTERING

THIS section presents the recommendation framework of PACF as shown in Figure 2. After data preprocessing, a collector of two-tuples (pa_k,δ_i) in reviews is obtained. The data preprocessing is the precursor of our recommendation approach to change raw data to structured data.

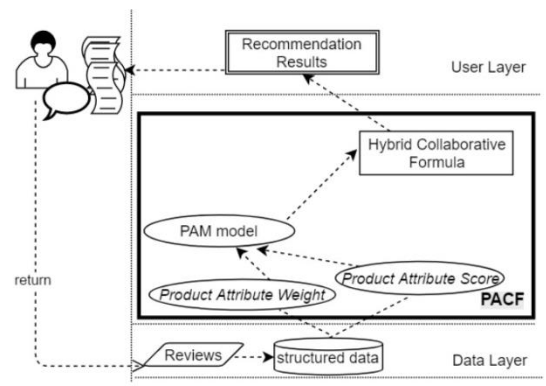

Figure 2. Recommendation framework of PACF

Then, a novel recommendation approach is used to generate recommendations. We formalize PACF with the PAM based on reviews and a hybrid collaborative filtering formula Γ_{PAM}. The *product attribute weight* proposed in PAM characterizes the user weight, while the *product attribute score* describes the product score. The specific modeling is detailed in Section 4.1. Developing the idea of a hybrid collaborative filtering approach, Section 4.2 proposes a new formula Γ_{PAM} for the PAM. The premise of the formula is to relate the *product attribute weight* to the *product attribute score*.

Finally, the recommendation results are generated through the formula.

4.1 PAM Based on Reviews

The use of reviews helps simulate user preferences and improves the recommendation performance. The primary issue in recommendation is how to express user preference models numerically. The challenge to model the different parameters after data preprocessing should be addressed. A novel approach is provided by vector multiplication in the matrix factorization algorithm (Zhao et al., 2016). Drawing lessons from this idea, users and products are modeled separately.

The reviews are generally modeled on the product attribute weight point of view (Chen et al., 2015; Lu et al., 2015). First, different reviews of the same user can measure the weight of a feature (Ma et al., 2017). Second, different reviews on the same product can measure certain features of the product (Zhao et al., 2016). In summary, from the perspective of the user weight and the product score, the PAM is subdivided into the *product attribute weight* and the *product attribute score*. The *product attribute weight* describes user preferences through the proportion of reviews on different product attributes, whereas the *product attribute score* represents product features as they relate to accurate user preferences.

4.1.1 Product Attribute Weight Analysis

The first measure, *product attribute weight*, recorded as W, is the degree of attention given to the attributes of the product. In this paper, the user preferences are inferred by weight. Given a user u_i and a product attribute pa_k, the formula for the *product attribute weight* is defined as follows.

$$W\left(u_i, pa_k\right) = \frac{|F_{ik}|}{|F_i|} = \frac{|\delta_{ik}|}{|\delta_i|} = \frac{\sum_{p_j \in R_i} |\delta_{ijk}|}{\sum_{p_j \in R_i} |\delta_{ij}|} \quad (1)$$

In formula (1), R_i is the set of u_i 's reviews, $p_j \in R_i$ represents the products that the user reviewed, $|F_{ik}|$ represents the frequency that the feature word is mentioned by the user with respect to product attribute pa_k, $|F_i|$ represents the number of times that all product attribute feature words are mentioned by the user, and δ indicates the sentiment polarity value. $|\delta_{ijk}|$ is the user's sentiment polarity set of product attributes pa_k for product $p_j \in R_i$. The number of user comments on the product attribute pa_k is $|\delta_{ik}|$. $|\delta_{ij}|$ represents the user's sentiment polarity set for $p_j \in R_i$ on all product attributes PA. $|\delta_i|$ is the frequency of all product attributes PA in the comments. When the value of W is zero or unknown, the *product attribute weight* is 0.1.

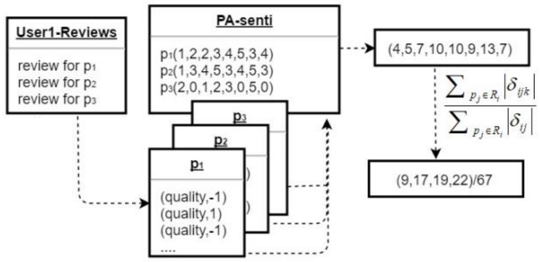

Figure 3. Calculation example for *W*

According to formula (1), the number of product attributes reviewed by the user can be statistically summarized. When a user comments on one product attribute more frequently than on other product attributes, the weight value is significantly greater than that of the other weights. The specific calculation flow is shown in Figure 3. First, the two-tuple (pa_k, δ_i) for the current product is transformed to an eight-tuple that corresponds to the positive and negative emotional levels of four product attributes, as shown in Table 2. We use eight-tuples in implementation to improve the calculation efficiency. The first element 1 in (1, 2, 2, 3, 4, 5, 3, 4) is the total number of positive sentiment polarities regarding the *quality* for the product p_1. After accumulation of all items, (4, 5, 7, 10, 10, 9, 3, 7) is obtained. Finally, through formula (1), the current user's W is (9, 17, 19, 22)/67.

4.1.2 Product Attribute Score Analysis

The second measure, *product attribute score*, is similar to the user rating of the product. The user's views of product attributes are scored and recorded as S. Given a product p_j and a product attribute pa_k, the *product attribute score* can be defined as follows.

$$S\left(p_j, pa_k\right) = \frac{\sum_{u_i \in R_j, \delta_{ijk}=1} |\delta_{ijk}|^*}{\sum_{u_i \in R_j} |\delta_{ijk}|} = \frac{\sum_{u_i \in r_j, \delta_{ijk}=1} |\delta_{ijk}|}{|\delta_{jk}|} \quad (2)$$

In formula (2), R_j is the review set for product p_j, u_i is the user who commented on it, and δ indicates the sentiment polarity value. δ_{ijk} is the user's sentiment polarity set of product attribute pa_k. * expresses the size of the sentiment polarity set when $\delta_{ijk} = 1$. When the value of S is zero or unknown, the *product attribute score* is 0.1.

The sentiment polarity of each product attribute in the user reviews is calculated by formula (2). The score is affected by the relative value of the positive sentiment polarity over the product attributes. The concrete calculation flow is shown in Figure 4. Every eight-tuple from different users is calculated from the two-tuple (pa_k, δ_i) as in the case of W (Figure 3). The tuple (4, 5, 5, 10, 7, 9, 10, 7) is obtained after accumulation of all items. The current product's S is (4/9, 5/15, 7/16, 10/17) as obtained through formula (2).

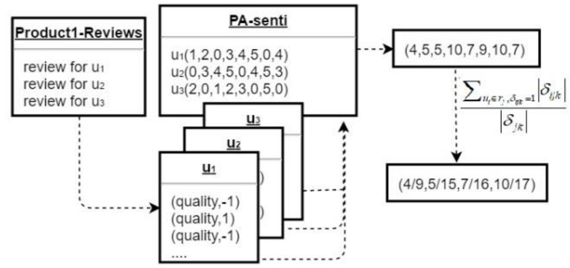

Figure 4. Calculation example for *S*

4.1.3 Product Attribute Model

Drawing on our experience of the multiplication form of matrix factorization, the PAM model is divided into different models corresponding to users and products. With the two formulas proposed above, the PAM can be defined:

$$PAM(u_i, p_j, PA) = \begin{cases} (W_1, W_2, ..W_{|PA|}) & for \quad users \\ (S_1, S_2, ..S_{|PA|}) & for \quad products \end{cases} \quad (3)$$

Therefore, the vector W and S over users and products separately are obtained. The user preference is expressed as the user's weight W on the product's attributes. The product score S is described through the positive sentiment polarity of product reviews.

4.2 Hybrid Collaborative Filtering Formula for PAM: Γ_{PAM}

The ultimate goal of a recommendation system is to generate recommendations. The typical idea of the calculation method is collaborative filtering (Chen et al., 2015; Lee, Oh, Yang, and Park, 2016; Lu et al., 2015; Musa and Hasan, 2016; Zou, Wang, Wei, Li, and Yang, 2014). Such methods can be divided into user-based, item-based, hybrid and matrix decomposition (Kassak, Kompan, and Bielikova, 2016; Koren, Bell, and Volinsky, 2009). SVD synthesized by matrix decomposition and basic prediction belongs to the category of matrix decomposition (Bakir, 2018).

In general, there are direct and direct ways to sort using collaborative filtering methods (Chen et al., 2015). The input of the collaborative filtering algorithm is the user-item-rating matrix. The PAM is transformed into vectors corresponding to users and projects. A formula that applies to the PAM is required because the constraints entered do not apply to the PAM. Considering the computational cost, the idea of collaborative filtering is used to generate recommendations for the PAM in an indirect way.

Hybrid collaborative filtering has proven to be a better recommendation approach than user-based approaches (Lu et al., 2015). Kassak et al. (2016) generated two sort lists based on the user or project in the multimedia group recommendation and summarized the results according to the lists' ranking order. Based on that work, Hammou and Lahcen (2017) further calculated the user similarity and product similarity in a user-item-rating matrix and sorted the results based on probabilistic values. Although the form of previous research results (Hammou and Lahcen, 2017) and (Kassak et al., 2016) differs in the specific calculation formulas and scenarios, the idea of sorting is similar. The specific explanation is that the user-based collaborative filtering parameter values and the item-based collaborative filtering parameter values are obtained and summed through scalar multiplication. These methods are used to calculate and summarize different aspects with collaborative filtering idea. After summarizing all the above research, the generalization formula we proposed is extracted for implementing the hybrid collaborative filtering:

$$\Gamma_{HCF} = UBCF \times IBCF \quad (4)$$

The vectors corresponding to users and products separately in the PAM are unrelated. To solve the problem, the average *product attribute score* of users, recorded as $\overline{S}$, is introduced based on the shopping history in reviews. The average *product attribute weight* of the product is calculated similarly and designated $\overline{W}$.

$$PAM'(u_i, p_j, PA) = \begin{cases} (W_1, W_2, ...W_{|PA|}) \\ (S_1, S_2, ...S_{|PA|}) \end{cases} \xrightarrow{reviews} \begin{cases} (\overline{S_1}, \overline{S_2}, ...\overline{S_{|PA|}}) & for \quad users \\ (\overline{W_1}, \overline{W_2}, ...\overline{W_{|PA|}}) & for \quad products \end{cases} \quad (5)$$

From formula (5), we obtain the average *product attribute score* of users and the average *product attribute weight* of products. Through $\overline{W}$ and $\overline{S}$, the credible relationship between users and products is established. A new hybrid collaborative filtering formula (6) for the PAM is obtained by deriving formula (4) through combining formula (5) and the cosine formula.

$$\Gamma_{PAM}(u_i, p_j) = UBCF \times IBCF$$

$$\xrightarrow{\cos} \cos_{UBCF} \times \cos_{IBCF}$$

$$\xrightarrow{based \quad on \quad PAM'} \frac{\sum_1^{|PA|}\left(W_i \times \overline{W_j}\right)}{\sqrt{\sum_1^{|PA|} W_i^2} \times \sqrt{\sum_1^{|PA|} \overline{W_j}^2}} \times \frac{\sum_1^{|PA|}\left(\overline{S_i} \times S_j\right)}{\sqrt{\sum_1^{|PA|} \overline{S_i}^2} \times \sqrt{\sum_1^{|PA|} S_j^2}}$$

$$\Gamma_{PAM}(u_i, p_j) = \frac{\sum_1^{|PA|}\left(W_i \times \overline{W_j}\right)}{\sqrt{\sum_1^{|PA|} W_i^2} \times \sqrt{\sum_1^{|PA|} \overline{W_j}^2}} \times \frac{\sum_1^{|PA|}\left(\overline{S_i} \times S_j\right)}{\sqrt{\sum_1^{|PA|} \overline{S_i}^2} \times \sqrt{\sum_1^{|PA|} S_j^2}} \quad (6)$$

For the current user, the cosine cos_{UBCF} is obtained by the vector $W = (W_1, W_2, \dots W_{|PA|})$ and the average user weight value $\overline{W} = (\overline{W_1}, \overline{W_2}, ...\overline{W_{|PA|}})$. The same approach is used to obtain cos_{IBCF}.

5 EXPERIMENTS

TO evaluate the performance of PACF, various experiments are carried out and compared by analyzing our approach relative to other algorithms using an offline dataset.

5.1 Experimental Data and Settings

We used the Amazon fine-food reviews dataset from SNAP (http://snap.stanford.edu/data/web-Amazon.htm). The dataset comprises 568,454 reviews posted by 256,059 users regarding 74,258 items of food. The format contains the UserId, the ProductId, the Text, and the Score attributes, which are required for the experiment. The Score attribute is an integer between 1 and 5. In large shopping sites, the number of users and products increases daily. However, the dataset of actual purchases is sparse, usually below 0.1%. Five representative datasets are selected to simulate the scene. As the reviews accumulate, the level of sparsity increases. Among them, Data3 and Data4 reduce the level of sparsity with additional reviews, which helps analyze the correlation between methods and the number of reviews or the level of sparsity. The datasets are described in Table 3.

Table 3. Descriptive Statistics of Datasets

Dataset	*Num. of users*	*Num. of products*	*Num. of ratings and reviews*	*Sparsity*
Data1	1232	754	1250	0.1346%
Data2	2420	1193	2500	0.0866%
Data3	4719	1791	5000	0.0592%
Data4	9051	1765	10000	0.0626%
Data5	17139	3148	20000	0.0371%

In offline experiments, evaluation metrics include the root-mean-square error (RMSE) and the coverage (Chen et al., 2015; Lu et al., 2015). The RMSE is used to evaluate the ability of recommendation methods to simulate user preferences. Evaluation metrics with the same purpose as the RMSE include the mean absolute error (MAE) and mean relative error (MRE). The recommendation performance has many aspects. The coverage is used to measure the ability of the methods to discover products. The RMSE, MRE, MAE and coverage are generally recognized and important evaluation metrics in the recommendation system. Moreover, the recommendation methods need practical feasibility. The space and time required by the methods are the variables that are primarily sought.

The PACF method contains the idea of collaborative filtering and matrix factorization. Therefore, the following comparison is considered: UBCF, IBCF, SVD and MF_PAM (Zhao et al., 2016). UBCF and IBCF are classical algorithms for collaborative filtering, SVD is a representative algorithm for matrix factorization, and the PAM model is substituted into the MF rewriting formula in (Zhao et al., 2016) by MF_PAM. The UBCF, IBCF and SVD algorithms are available from Cambridge Coding (http://online.cambridgecoding.com/notebooks).

In the specific experiment, Table 3 is used as the dataset. The user preference simulation of algorithms under sparse data is validated with the RMSE, MRE and MAE. The ability of PACF and other algorithms to explore products is verified by using the coverage. Using the space complexity and time complexity as metrics, the feasibility of practical application of the PACF model is evaluated.

5.2 Experimental Results and Discussion

5.2.1 Rating Prediction

Rating prediction, the most important experimental parameter in the recommendation system, measures the ability of the recommendation methods to predict user behavior. The smaller the error, the more accurate the rating and the better the recommendation performance. Thus, we carry out experiments to compare UBCF, IBCF, SVD and MF_PAM to measure the rating prediction accuracy through the RMSE, MAE and MRE. The dataset in Table 3 is used as the experimental data, and 1/4 of the dataset is used as the target test set.

From Figure 5, Figure 6 and Table 4, all evaluation metrics of MF_PAM and PACF using the PAM model are superior to those of UBCF, IBCF and SVD. The input of UBCF, IBCF and SVD is the user-item-rating matrix. The three algorithms are not suitable for rating predictions for sparse data. The input sources of MF_PAM and PACF are reviews and are not affected by sparsity. The recommendation methods based on PAM are better than the baseline models. Furthermore, MF_PAM and PACF slowly reduce the error as the reviews accumulate. Obviously, our proposed PAM is more suitable for large but sparse data.

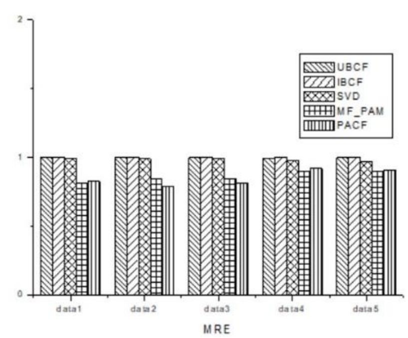

Figure 5. Experimental results in terms of the MRE

Table 4. Experimental results in terms of the MAE

MAE	Dataset				
	Data1	*Data2*	*Data3*	*Data4*	*Data5*
UBCF	3.9729	4.0605	4.0325	4.0647	4.1113
IBCF	3.9776	4.0685	4.0421	4.0788	4.1223
SVD	3.9553	4.0415	4.0222	3.9962	4.0039
MF_PAM	1.1566	1.1707	1.1607	1.1835	1.1864
PACF	1.1054	1.0447	1.0576	1.1156	1.0930

The MREs of all algorithms are relatively high in Figure 5 and are greatly affected by the data sparsity.

The PACF MRE values are generally lower than those of other methods, which is a promising result. In addition, the MAE of MF_PAM shows an increasing trend. The MAE of MF_PAM increases with decreasing sparsity. As shown in Table 4, the results also show that the MAE of PACF is always lower than that of MF_PAM. The MAE of PACE is in a state of volatility but ultimately declines. Despite the double impact of the number of reviews and sparsity, our proposed PACF is significantly better than MF_PAM.

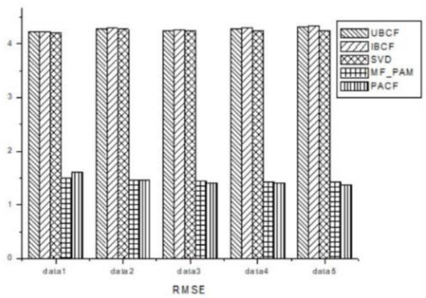

Figure 6. Experimental results in terms of the RMSE

In dataset Data1 of Figure 6, the PACF RMSE is higher than that of MF_PAM. The current number of reviews is 1250. Hybrid collaborative filtering is influenced by fewer reviews. Because the reviews are scarce, W and S are not accurate. MF_PAM is better than PACF with fewer reviews. However, with the exception of dataset Data1 of Figure 6, the RMSE of PACF performs better than that of MF. As we expect, the trend of error reduction is greater than that of MF. When reviews are not sparse, W and S are accurate. Relative to MF_PAM, our proposed PACF reduces the error faster while filtering users and products through Γ_{PAM}. The formula Γ_{PAM} contributes to generate accurate recommendation results for the PAM. In summary, PACF is more suitable for sites characterized by large and sparse reviews, such as shopping sites.

5.2.2 Recommendation Coverage Experiment

The rating data have a sparseness of less than 0.1% on shopping sites. Furthermore, the product category is in the billions of units. Thus, using TOP-N sorting to evaluate accuracy is not an appropriate method. Our experiment simulates the data from actual shopping sites. The extremely low accuracy of this site does not have a value; in this case, the coverage is more appropriate. The experiment aims to demonstrate that proposed PACF achieves better coverage performance than UBCF, IBCF and SVD. First, we take the datasets in Table 3 as the experimental dataset. Then, the evaluation metric coverage is calculated under N = 1, 5, 10 and 20. The experimental results are shown in Table 5. The coverage unit is %. The next part describes the analysis of the experimental results.

Table 5. Experimental results in terms of coverage

Data set	Coverage				
	Methods	*N = 1*	*N = 5*	*N = 10*	*N = 20*
Data 1	UBCF	0.1326	0.0119	0.0146	0.0291
	IBCF	0.2652	0.0093	0.0172	0.0305
	SVD	0.6631	0.0186	0.0358	0.0650
	PACF	1.3263	0.0597	0.0941	0.1552
Data 2	UBCF	0.0008	0.0042	0.0092	0.0176
	IBCF	0.0017	0.0050	0.0101	0.0192
	SVD	0.0117	0.0268	0.0360	0.0762
	PACF	0.0142	0.0386	0.0695	0.1215
Data 3	UBCF	0.0006	0.0036	0.0430	0.0122
	IBCF	0.0017	0.0061	0.0089	0.0151
	SVD	0.0168	0.0329	0.0061	0.0642
	PACF	0.0101	0.0274	0.0519	0.0966
Data 4	UBCF	0.0017	0.0045	0.0073	0.0130
	IBCF	0.0017	0.0062	0.0102	0.0164
	SVD	0.0221	0.0567	0.0771	0.1082
	PACF	0.0096	0.0306	0.0515	0.0816
Data 5	UBCF	0.0004	0.0016	0.0030	0.0048
	IBCF	0.0006	0.0022	0.0042	0.0076
	SVD	0.0072	0.0198	0.0358	0.0550
	PACF	0.0056	0.0150	0.0260	0.0398

As shown in Table 5, with increasing N, the coverage of the recommendation algorithms also increases. None of the algorithms are linearly proportional to N. It can be concluded that the key parameter that affects the coverage is not N.

Furthermore, the coverage of UBCF and IBCF shows a downward trend in Table 5, while the coverage of SVD shows an upward trend. Apart from the idea of different algorithms, SVD is also affected by the amount of data. With increasing amount of data, the recommendation performance improves. SVD is suitable for large and sparse datasets.

The overall coverage of PACF is increasing in Table 5. PACF depends on the number of reviews. The larger the review, the better the performance. PACF also applies to scenarios where SVD is suitable. In Data1, Data2 and Data3, PACF performs better than SVD. In Data4 and Data5, PACF’s coverage is lower than that of SVD. In further analysis, the PACF formula is similarly affected by the purchase record. With a sparsity of 0.05%, PACF sacrifices coverage. Moreover, when the sparseness is higher than 0.05%, the coverage performance of our proposed PACF is satisfactory.

5.2.3 Complexity Analysis

Recommendation methods such as algorithms based on time and space are also worth considering. In the big data environment, customer satisfaction is influenced by the speed of the recommendation system response. From the point of view of the algorithm and user experience, the time complexity and spatial complexity of recommendation methods should be evaluated. Although a variety of forms of optimization are used in algorithms, the frequency of the statement can be replaced by the number of basic operations of the algorithm. Under non-optimized conditions, if the time complexity of method A is lower than that of method B, the computational cost of method A is proved to be smaller.

SVD is computationally expensive due to dimensionality reduction. Because UBCF is more computationally expensive than IBCF, IBCF has the lowest computational cost among the three. Therefore, PACF is compared with this method to verify the practical application of PACF. The calculation order of IBCF and PACF is shown in Table 6. In the table, $S_P^{\cos}$ is the project similarity matrix, X is the user project rating matrix, and $\sum |S_p^{\cos}| \quad \forall p \in P$ is the sum of the rows of $S_P^{\cos}$ [20].

Table 6. IBCF and PACF Calculation Order

Steps	IBCF	PACF
Step 1	$S_P^{\cos} \to S_p^{\cos} \quad \forall p \in P$ $S_p^{\cos}(p_m, p_n) = \frac{p_m \cdot p_n}{\lvert p_m \rvert \lvert p_n \rvert}$	*PAM*
Step 2	$X \cdot S_P^{\cos}$	*PAM′*
Step 3	$\sum \lvert S_p^{\cos} \rvert \quad \forall p \in P$	$\cos_{UBCF}$ $\cos_{IBCF}$
Step 4	$(X \cdot S_p^{\cos}) / \sum \lvert S_p^{\cos} \rvert \quad \forall p \in P$	ΓPAM

The number of arithmetic operations (i.e., +, –, ×, ÷) is used to determine the sentence frequency. First, the time complexity of the IBCF T^{IBCF} is calculated.

$$T_{S_p^{\cos}}^{IBCF} = T_+ + T_\times + T_\div + T_{\sqrt{}} = 3(m+1) + m + 1 + 3$$

The matrix of similarity is a symmetric matrix. Duplicate calculation terms in the matrix are removed when calculating T_{Step1}^{IBCF}. The time complexity of IBCF TIBCF is obtained through the accumulation. In addition, the largest matrix involved in IBCF operations is m rows and n columns, so the space complexity is *O(mn).*

$$\begin{aligned} T_{Step1}^{IBCF} &= T_+ + T_\times + T_\div + T_{\sqrt{}} \\ &= [(n-1)(n-2)/2 + n] T_{S_p^{\cos}}^{IBCF} \\ &= [(n-1)(n-2)/2 + n][3(m+1) + m + 1 + 3] \end{aligned}$$

$$T_{Step2}^{IBCF} = T_+ + T_\times + T_\div + T_{\sqrt{}} = mn(n-1) + mn + 0 + 0$$

$$T_{Step3}^{IBCF} = T_+ + T_\times + T_\div + T_{\sqrt{}} = n(n-1) + 0 + 0 + 0$$

$$T_{Step3}^{IBCF} = T_+ + T_\times + T_\div + T_{\sqrt{}} = 0 + 0 + mn + 0$$

$$T^{IBCF} = T_{Step1}^{IBCF} + T_{Step2}^{IBCF} + T_{Step3}^{IBCF} + T_{Step4}^{IBCF} = O(mn^2)$$

$$S^{IBCF} = S_{Step1}^{IBCF} + S_{Step2}^{IBCF} + S_{Step3}^{IBCF} + S_{Step4}^{IBCF} = O(mn)$$

The time complexity T^{PACF} and space complexity S^{PACF} of PACF are calculated as described above. r is the number of reviews; k is the number of product attributes.

$$\begin{aligned} T_{Step1}^{PACF} &= T_+ + T_\times + T_\div + T_{\sqrt{}} = [(r-m)k + r(2k-1) \\ &\quad + (r-n)k + (2r-m)k] + 0 + (mk + nk) + 0 \end{aligned}$$

$$\begin{aligned} T_{Step2}^{PACF} &= T_+ + T_\times + T_\div + T_{\sqrt{}} \\ &= [(r-m)k + (r-n)k] + 0 + (mk + nk) + 0 \end{aligned}$$

$$\begin{aligned} T_{Step3}^{PACF} &= T_+ + T_\times + T_\div + T_{\sqrt{}} \\ &= mn(k+1) + 3mnk + mn + mn \end{aligned}$$

$$T_{Step4}^{PACF} = T_+ + T_\times + T_\div + T_{\sqrt{}} = 0 + mn + 0 + 0$$

$$T^{PACF} = T_{Step1}^{PACF} + T_{Step2}^{PACF} + T_{Step3}^{PACF} + T_{Step4}^{PACF} = O(mn)$$

$$S^{PACF} = S_{Step1}^{PACF} + S_{Step2}^{PACF} + S_{Step3}^{PACF} + S_{Step4}^{PACF} = O(mn)$$

Finally, the results are shown in Table 7. We conclude that the space complexity of PACF is not higher than that of IBCF. The time complexity of PACF is significantly lower than that of IBCF. Because the IBCF calculation cost is less than that of SVD and UBCF, PACF is superior to the comparative methods and is feasible in practical application.

Table. 7. Complexity of IBCF and PACF

Complexity	IBCF	**PACF**
Time	*O(mn²)*	*O(mn)*
Space	*O(mn)*	*O(mn)*

6 CONCLUSION AND FUTURE WORK

CONSIDERING the characteristics of product reviews, this paper uses constant product attributes and introduces sentiment polarity. The objective is to solve the problem of complex product reviews and to refine information on user preferences. Based on the above conditions, the hybrid collaborative filtering approach called product attribute collaborative filtering (PACF) is proposed. The product attribute model (PAM) and the Γ_{PAM} formula are integrated to achieve PACF. The PAM consists of the *product attribute weight* and *product attribute score*. The perspectives of users and products can effectively simulate user preferences. The experiments have showed that PACF achieves better recommendation performance in accommodating large and sparse reviews. The coverage is superior to that of other methods at a sparsity higher than 0.05%. In addition, the calculation cost of PACF is lower than that of UBCF, IBCF and SVD, which implies feasibility in practical applications.

In the future, we plan to study the issue that PACF sacrifices coverage when the sparsity is less than 0.05%. Using user relations and social information from other platforms can further improve the performance of the recommendation on sparse data.

Moreover, the implementation and effective response of a large-scale electronic website platform is worthy of further work. To this point, cloud computing and the use of clusters will be considered to accelerate the reaction speed.

7 DISCLOSURE STATEMENT

NO potential conflict of interest was reported by the authors.

8 ACKNOWLEDGMENTS

THE authors thank the anonymous reviewers for their valuable and constructive comments. This work is supported by the National Key Research and Development Plan of China under grant no. 2017YFD0400101, the National Natural Science Foundation of China under grant no. 61502294, and the IIOT Innovation and Development Special Foundation of Shanghai under Grant No. 2017-GYHLW-01037.

9 REFERENCES

Alton, Y. K. C., & Snehasish, B. (2016). Helpfulness of user-generated reviews as a function of review sentiment, product type and information quality. *Computers in Human Behavior*, 54, 547-554.

Bakir, C. (2018). Collaborative filtering with temporal dynamics with using singular value decomposition. *Technical Gazette*, 25(1), 130-135.

Chen, L., Chen, G. L., & Wang, F. (2015). Recommender systems based on user reviews: the state of the art. *User Modeling And User-adapted Interaction*, 25(2), 99-154.

Hammou, B. A., & Lahcen, A. A. (2017). FRAIPA: a fast recommendation approach with improved prediction accuracy. *Expert Systems with Applications*, 87, 90-97.

Jiang, S. M., Cai, S. Q., Olle, G. O., & Qin, Z. Y. (2015). Durable product review mining for customer segmentation. *Kybernetes*, 44(1), 124-138.

Kassak, O., Kompan, M., & Bielikova, M. (2016). Personalized hybrid recommendation for group of users: Top-N multimedia recommender. *Information Processing & Management*, 52(3), 459-477.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8), 30-37.

Lei, X. J., Qian, X. M., & Zhao, G. S. (2016). Rating prediction based on social sentiment from textual reviews. *IEEE Transactions on Multimedia*, 18(9), 1910-1921.

Lee, J., Oh, B., Yang, J., & Park, U. (2016). RLCF: a collaborative filtering approach based on reinforcement learning with sequential ratings. *Intelligent Automation and Soft Computing*, 23(3), 439-444.

Li, H. C., Liu, J. X., Cao, B. Q., Liu, X. Q., & Li, B. (2017, June). Integrating tag, topic, co-occurrence, and popularity to recommend web APIs for mashup creation. In *SCC2017*. *Proceedings of IEEE International Conference on Services Computing* (pp.84-91). Honolulu, HI: IEEE Computer Society.

Lu, J., Wu, D. S., Mao, M. S., Wang, W., & Zhang, G. Q. (2015). Recommender system application developments: a survey. *Decision Support Systems*, 74, 12-32.

Ma, Y., Chen, G. Q., & Wei, Q. (2017). Finding users preferences from large-scale online reviews for personalized recommendation. *Electronic Commerce Research*, 17(1), 3-29.

Meng, J. K., Zhen, Z. B., Tao, G. H., & Liu, X. Z., (2016, June). User-Specific rating prediction for mobile applications via weight-based matrix factorization. In *ICWS2016*. *Proceedings of IEEE 23rd International Conference on Web Services* (pp.728-731). San Francisco, CA: IEEE Computer Society.

Musa, M, & Hasan B. (2016). The effect of neighborhood selection on collaborative filtering and a novel hybrid algorithm. *Intelligent Automation and Soft Computing*, 23(2), 261-269.

Su, J. K., Ewa, M., & Edward, C. M. (2018). Understanding the effects of different review features on purchase probability. *International Journal of Advertising*, 37(1), 29-53.

Wang, H. F., Chi, X., Wang, Z. J., Xu, X. F., & Chen, S. P. (2017, June). Extracting fine-grained service value features and distributions for accurate service recommendation. In *ICWS2017*. *Proceedings of IEEE 24th International Conference on Web Services* (pp.277-284). Honolulu, HI: IEEE Computer Society.

Yu, Z. W., Xu, H., Yang, Z., & Guo, B. (2016). Personalized travel package with multi-point-of-interest recommendation based on crowdsourced user footprints. *IEEE Transactions on Human-Machine Systems*, 46(1), 151-158.

Zhao, W. X., Wang, J. P., He, Y. L., Wen, J. R., Chang, E. Y., & Li, X. M. (2016). Mining product adopter information from online reviews for improving product recommendation. *ACM Transactions on Knowledge Discovery from Data*, 10(3), article no.29.

Zou, T. F., Wang, Y., Wei, X. Y., Li, Z. L., & Yang, G. C. (2014). An effective collaborative filtering via enhanced similarity and probability interval prediction. *Intelligent Automation and Soft Computing*, 20(4), 555-566.

10 NOTES ON CONTRIBUTOR

Prof. Min Cao, received the Ph.D. degree in Control Theory and Control Engineering from Shanghai University, China, in 2000. Now, she is an associate professor at the School of Computer Engineering and Science, Shanghai University. Her research interests include software architecture, distributed computing and database.

Sijing Zhou received the BS degree form Jilin University, China, in 2015. Currently, she is studying in the School of Computer Engineering and Science of Shanghai University in China as a postgraduate student. Her research interests include product recommendation using reviews and the sparsity problem of recommendation.

Prof. Honghao Gao received the Ph.D. degree in Computer Science and started his academic career at Shanghai University in 2012. He is an IET Fellow, BCS Fellow, EAI Fellow, IEEE Senior Member, CCF Senior Member, and CAAI Senior Member. Prof. Gao is currently a Distinguished Professor with the Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, China. His research interests include service computing, model checking-based software verification, and sensors data application.