Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 14, no. 7, July 2018, pp. 1401-1410
DOI: 10.23940/ijpe.18.07.p3.14011410

A Personalized Recommendation Algorithm based on Text Mining

Ningbin Zhang*

School of Information Engineering, Xi'an University, Xi'an, 710062, China

Abstract

The recommendation system is a new technology used to recommend products for customers from huge amounts of products by inferring objective users' preferences based on their personal information or online behavior. This paper studied the main personalized recommendation technology for current e-commerce. It proposed a hybrid recommendation algorithm based on opinion mining. This system combines web data mining technology, i.e., takes advantage of user-generated content by mining customers' online reviews. It is well known that online reviews can directly reflect a customer's real emotions and expectations, so it is appropriate to extract a customer's latent interest and preference from his/her reviews, thus refining recommendations and improving accuracy. Meanwhile, an experiment was conducted and the result demonstrated that our system could generate a reliable and realistic recommendation.

Keywords: recommendation system; personal recommendation; e-commerce; text mining

(Submitted on March 29, 2018; Revised on May 3, 2018; Accepted on June 19, 2018)

1. Introduction

In the e-commerce environment, in the face of tremendous commodities and information provided by internet merchants [4,11,12], people have plenty of choices. However, it is impossible to check and select commodities as in a real environment, resulting in a dilemma of difficult selection and a waste of time and energy. Traditional search engines like Baidu and Google and commodity search engines like Etao [1,7] retrieve enormous information according to user input keywords. Even so, the searched commodity information is universal and is selected by the user with lots of time and energy. Statistics show that without a recommendation system helping make decisions, users need to check on average 11.7 goods before finding satisfactory merchandise. After the introduction of the recommendation mechanism, users need to check on average only 6.6 items, reducing workload by almost 50%. If internet merchants want to increase client loyalty and possession rate, they must consider how to let users reduce online shopping time and purchase commodities to their hearts' desire [10]. The individualized recommendation system attempts to convert the commodity browser into a purchaser, increase cross-selling to boost sales and increase user adhesiveness to enhance user satisfaction and loyalty. On the e-commerce platform, it is not hard to find that websites that can provide a personalized recommendation system can yield more throughput than common e-business websites [8]. The individualized recommendation based on online review opinion mining aims to recommend merchandise which accords the most to users' preferences with the use of real user review data on internet e-commerce websites to find their interests or preferences through analysis of their review information [5].

The internet has dramatically changed the way people express their opinions. Many commercial sites such as Amazon and Taobao allow users to provide product evaluations [3,6]. In addition, BBS, blogs, micro-blogs, and other Internet applications can also see the user to express their views. These reviews and opinions are often referred to as user-generated content or user-generated media. These texts can reveal a variety of valuable information, but because the text is unstructured data, this makes it as easy to handle as structured data [2,9]. Opinion mining is a cross type technology of data mining, text mining, Natural Language Processing, and artificial intelligence. It also has strong practicability in the process of text data. The paper takes opinion mining technology, from the perspective of users' online reviews, to extract commodity features that are interesting to users as to predict their preference for some article, then recommend their items which are in exact accordance with their interests by providing an intelligent shopping guide. To be specific, it offers them items that can meet

* Corresponding author.
E-mail address: ningbinzhangnbz@sina.com

their fuzzy requirements, tries to understand users better, helps raise website flow, and increases user's adhesiveness, so it has very strong and practical value.

2. Individualized Recommendation Model based on Online Review Mining

The objective of individualized recommendations is to boost sales on the basis of recommending items that can satisfy users' preferences through collection and analysis of individual consumers' online behaviors and purchase record data and withdraw their potential preferences [13]. The paper proposes an individual recommendation algorithm based on users' online reviews, i.e., full use of users' reviews of items. Since a majority of users would make more reviews of content to their attention or show stronger affection when they are reviewing items, we believe product features that are marked too much by reviews on an item in one category may concern current users. The idea of the algorithm: from reviews, obtain a feature of an item of the kind interesting to the user to search horizontally for other users who have higher interest similarity with them; then, compare vertically the item of the kind which gains the highest score among those product features and recommend it to the user.

The major task of this paper is to offer personalized recommendations to the user with relative recommendation system technology by analyzing user interest and similarity between users after comparing product similarity and good/bad points. This is done by decomposing user review data into the granularity of product features, with the use of such data about one product for one e-commerce website [15]. The main thinking is to create one review analysis and user recommendation model based on product feature and analyze key features of each kind of product. The prerequisite of the assumption of this method is: when a user wants to buy one item and refers to another user's review, he cares more about article features which are more interesting to him. Likewise, when a user reviews an item, he would make more specific reviews on features that he is most concerned about. Therefore, we assume that for all the reviews made by a user for the same category, the product feature he reviews the most is his concerned feature. In the meantime, when a user's rating of a concerned item is generally lower, he would make a higher request. Unlike other relative work on the traditional recommendation system, this paper discusses mainly text data about product review through the relative model, rather than merely relying on users' overall rating data on products. Main contributing factors that allow us to do this are users' reviews or opinions that can be specifically studied on the level of product features, the system's accurate analyses of products and users, and meticulously defined users' interests. We introduce the working flow of proposed individual recommendation model: feature extraction, user similarity calculation, commodity similarity calculation, and generate the recommendation.

2.1. Feature Extraction

During feature extraction, when we only have a few commodity features, we can use the Bootstrapping method to learn the text. Bootstrapping is a semi-supervised learning technique, with the main thinking: use a few main defined commodity feature words as the feature's seed set, then according to learned pattern, withdraw relative glossaries from unlabeled text. Labeled words are used as the seed set which can join the next cycling process. Through a specified iteration time, or until no new seed is produced, the cycle ends and outputs extracted words and patterns. The merit is the lower requirement for the training set. With limited sample information, it is possible to learn automatically patterns relevant with annotated information.

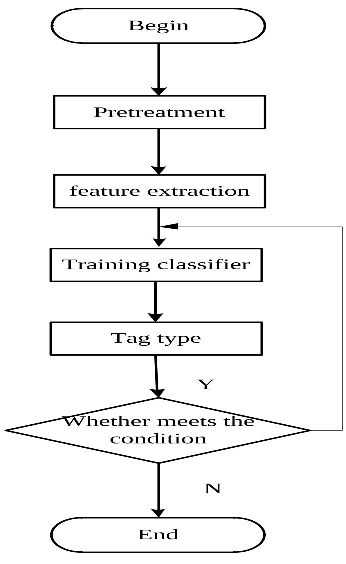

Figure 1. Framework of semi-supervised learning theory

Figure 1 shows the theoretical framework of semi-supervised learning. Semi-supervised learning is distinguished from supervised learning and unsupervised learning. Supervised learning is made in the presence of a large number of classified labels. Set up the classification model to classify the unclassified data. However, in the case of very large amounts of data, manually labeling a large number of class labels is very unrealistic. Unsupervised learning is in the absence of classification labels. The computer learns how to generate and label class labels, such as through the clustering algorithm. Semi-supervised learning is the case between supervised learning and supervised learning. It is a part of the data label: first, use the supervised learning method to learn the existing annotation data and then attempt to label those not yet annotated data. The labeled data is included, and the cycle repeats to increase the labels.

2.2. Calculation of Commodity Similarity

Commodity features derive not only from user review data but also from products' ontology structures, which in turn reflects details of one item. To make it simpler, after extracting product features, we only use item features fetched from user reviews, which can help easily generate the Item-Feature model. For an item of one category, all its features are expressed with F vector. $F = \{f_1, f_2, ..., f_p\}$. There are many methods for calculating item similarity, of which cosine similarity is adopted to compute Item-Feature matrix, i.e., use commodity feature item to represent commodity. The matrix value can be scored for an item, acquired from one of its feature items or weight of one feature item.

2.3. Calculation of User Similarity

The core of the collaborative filtering method is the calculation of similarity. The paper proposed a personalized recommendation based on opinion mining, which is based on the product feature. When similarity between users is calculated, we introduced the User-Feature model, hereafter named the UF model. The UF model records each user's rating information about items. User-feature matrix represents the matrix of reviews by user i for feature k, marked as $UF_{m\times p} = \{UF_{ik} \in UF \mid 1 < i < m, 1 < k < p\}$. The value of UF_{ik} is decided by the experimental requirement. If UF_{ik} records rating result by user for one feature k, UF matrix reflects user's visual evaluation of one item's specific feature, if UF_{ik} records weight of user i for feature k, UF matrix reflects the degree of its preference for the feature. We explain by citing UF as the weighting matrix of user for one feature to calculate user similarity with Pearson correlation coefficient in Equation (1).

$$sim(i, j) = \frac{\sum_{k=1}^{n} (UF_{i,k} - \overline{UF}i)(UF_{j,k} - \overline{UF}j)}{\sqrt{\sum_{p=1}^{n} (UF_{i,p} - \overline{UF}i)^2} \sqrt{\sum_{q=1}^{n} (UF_{j,p} - \overline{UF}j)^2}} \tag{1}$$

If we use the given item feature for analysis, i.e., all users' item features of the same dimension, then user similarity can be computed by cosine similarity method in Equation (2).

$$sim(i, j) = \cos(i, j) = \frac{UF_i \times UF_j}{\| UF_i \|^2 \times \| UF_j \|^2} = \frac{\sum_{k=1}^{n} UF_{ik} UF_{jk}}{\sqrt{\sum_{p=1}^{n} UF_{ip}^2} \sqrt{\sum_{q=1}^{n} UF_{jq}^2}} \tag{2}$$

2.4. Generate Recommendations

The use-item matrix is the most common matrix used to stand for user i 's rating of item j, put as $R_{m\times n} = \{R \in R_{ij} \mid 1 < i < m, 1 < j < n\}$. User collection is $U_m = \{u_1, u_2, ..., u_m\}$ and item collection is $I_N = \{i_1, i_2, ..., i_N\}$.

Generally speaking, the R matrix records rating numbers of a user for one item j. $R_{m\times n}=\begin{bmatrix} r_{11} & \cdots & r_{1n} \\ & & \\ r_{m1} & \cdots & r_{mn} \end{bmatrix}$.

With the user similarity formula and R matrix, we can predict the target user's possible scoring value for an item that has not been purchased, as seen in Equation (3):

$$p_{u,s}=\overline{r}_u+\frac{\sum_{u^{'}\in U^{'}} sim(u,u^{'})\times(r_{u^{'},s}-\overline{r}_{u^{'}})}{\sum_{u^{'}\in U^{'}} sim(u,u^{'})} \tag{3}$$

3. Online Recommendation in Combination with Multi-Objective Decision Making

Because the user's online review can be processed as < feature, emotion > pair, the decision matrix is shown in Table 1.

Table 1. Fuzzy decision matrix

Emotional polarity	Feature 1	Feature 2	Feature 3	…	Feature n
Goods 1	$\overline{f}_1(a_1)$	$\overline{f}_2(a_1)$	$\overline{f}_3(a_1)$	…	$\overline{f}_n(a_1)$
Goods 2	$\overline{f}_1(a_2)$	$\overline{f}_2(a_2)$	$\overline{f}_3(a_2)$	…	$\overline{f}_n(a_2)$
Goods 3	$\overline{f}_1(a_3)$	$\overline{f}_2(a_3)$	$\overline{f}_3(a_3)$	…	$\overline{f}_n(a_3)$
…	…	…	…	…	…
Goods m	$\overline{f}_1(a_m)$	$\overline{f}_2(a_m)$	$\overline{f}_3(a_m)$	…	$\overline{f}_n(a_m)$

Regardless of the language form of online reviews, pre-processing steps are required before parsing the text data. After extracting the commodity features, combining the multi-objective decision-making of online recommendations will need to dig out the goods and features of the opinions with relevant words, and then determine the emotions expressed in the words to determine the decision matrix and facilitate the subsequent calculation. Due to the fuzziness of semantics, this paper proposes a method of fuzzy PROMETHEE for the online recommendation. Fuzzy PROMETHEE is divided into the following six steps:

1. Define semantic values and their corresponding fuzzy Number, and identify candidate commodities and their indicators. Let A represent the candidate product set. $A=\{a_1,a_2,...,a_m\}$, C is the index set, $f_i(a_i)$ is the value of c_j of the candidate product a_i. It is shown in Equation (4) and Equation (5).

$$F=[f(a_i)]_{m\times n}=\begin{pmatrix} \overline{f}_1(a_1)......\overline{f}_n(a_1) \\ ... \qquad\qquad ... \\ \overline{f}_1(a_m)......\overline{f}_n(a_m) \end{pmatrix} \tag{4}$$

$$\overline{f}_i(a_i)=(f_j(a_i)^l, f_j(a_i)^m, f_j(a_i)^r) \tag{5}$$

2. Determine the weight value of each feature/indicator. It is shown in Equation (6).

$$w=[w_1,w_2,...,w_n]=\frac{1}{\sum_{i=1}^{n} DF_i}[DF_1,DF_2,...,DF_n] \tag{6}$$

3. The PROMETHEE method is a method based on pairwise comparison. There are six common priority function

construction methods, and we use the most common v-type indicator priority function. It is shown in Figure 2.

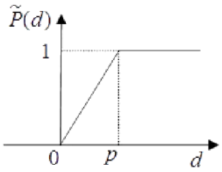

Figure 2. V-type indicator precedence function.

The a 's priority for b is defined in Equation (7).

$$\bar{P}(d)=\begin{cases}0, & d<0\\ \dfrac{d}{p} & 0<d<p\\ 1, & d>p\end{cases} \tag{7}$$

In Equation (7), p is a threshold of strict priority, in model and experiment, set to p =2. Then, on the index j , the priority of a for b is shown in Equation (8).

$$\begin{aligned}P_j(a,b) &= P_j(d_j(a,b)) = P_j(\bar{f}_j(a_i)-\bar{f}_j(b_i))\\ &= P_j((a^l,a^m,a^u)-(b^l,b^m,b^u))\\ &= P_j(a^l-b^u,a^m-b^m,a^u-b^l)\\ &= (P_j(a^l-b^u),P_j(a^m-b^m),P_j(a^u-b^l))\end{aligned} \tag{8}$$

4. The fuzzy preference index is shown in Equation (9) and Equation (10).

$$\bar{\pi}(a,b)=\sum_{j=1}^{n} w_j \bar{P}_j(a,b) \tag{9}$$

$$\bar{\pi}(b,a)=\sum_{j=1}^{n} w_j \bar{P}_j(b,a) \tag{10}$$

$\bar{\pi}(a,b)$ is a priority for b on all indicators, and $\bar{\pi}(b,a)$ vice versa.

5. The calculation formulas for fuzzy positive output flow, fuzzy negative output flow, and fuzzy net output flow are shown respectively in Equation (11), Equation (12), and Equation (13).

$$\bar{\phi}^{+}(a)=\frac{1}{m-1}\sum_{b\in A}^{m}\bar{\pi}(a,b) \tag{11}$$

$$\bar{\phi}^{-}(a)=\frac{1}{m-1}\sum_{b\in A}^{m}\bar{\pi}(b,a) \tag{12}$$

$$\bar{\phi}(a)=\bar{\phi}^{+}(a)-\bar{\phi}^{-}(a)=(r^l,r^m,r^u) \tag{13}$$

6. Calculate the expected value of the net output flow. Since the model users triangular fuzzy Number, the fuzzy formula is adopted to calculate the expected value of net output stream to make a final comparison. It is shown in Equation (14).

$$E(\bar{\phi}(a)) = (r^l + 2r^m + r^u)/4 \quad (14)$$

4. Experiment Design and Discussion

4.1. Experimental Environment

Most experiments here are completed in Eclipse compilation environment with Java language. Since the experiments use both Chinese and English reviewing data, the author pre-processes Chinese data by applying the Chinese morphological analysis system ICTCLAS during word segmentation and tagging, which is an open source software furnished by the CAS Institute of Computer Technology, with the Hierarchical Hidden Markov Model HHMM as the core of word segmentation technology. This includes the Chinese segmentation module, part of the speed tagging module, the named entity recognition module, and the new word recognition module. The system has wide applicability and supports user custom dictionary and several coding schemes including traditional Chinese, GBK, UTF-8, UTF-7, and UNICODE. The newest version of ICTCLAS2016 realizes the segmenting speed theoretically up to approximately 500KB/s by a single machine, and the precision reaches up to 98.45%. Moreover, it supports its application and development in all environments. This is shown in Table 2.

Table 2. ICTCLAS word segmentation results sample

Word segmentation	The screen is awesome, very clear, can download the software very much, the reaction is very good, the camera function. Personal feeling is good. The sound is clear, put the volume or not so awesome. Battery usage too quick, boot slow, easy to report the wrong exit when playing the game. Small keyboard usage is very small, even if the use of keys. Disk button is too small, the error rate of personal feeling to make 99.5%. Mobile phone built-in ROM is too small.
Result of word segmentation	The screen /n is awesome /d /n, /wd /d /a, /wd very clear, /v can download the /v software /ude1 /n, Pretty much /a /ude1 /vi, /wd reaction /d also /d pretty good /a /ude1 /d, /wd camera /vn, function /q /n /n people feel /n is also a good /vshi /a /ude1 /d. /wj sound /n clear /a, /wd /f /v /n or /c /d is not the volume of /vshi /d /n is awesome. /wj battery /n enable, with /v degree /qv too /d fast /a, /wd boot /vi slow /a, /wd play /v games /ng /n easy /a, /n wrong /vd exit /v. /wj small /a keyboard /ude1 /n use /vn degree /qv very /a small /wd, /d on the /v to use the /v keyboard /ude1 /n /d button /n too small /a /y, /wd /v by /p rate /q /v people /n feel /n make 99.5%/m /v. /wj mobile /n /f /v inside ROM/x /d too small/a

4.2. Experiment of Personalized Recommendation

4.2.1. Personalized Recommendation Review Data

The review data used here is from online hotel ratings, which are English data captured by TripAdvisor（http://www.tripadvisor.com）during March 14, 2016, to May 15, 2016, totalling 235,793 pieces. These reviews were made by users after their own experiences with the commodity, including many emotional vocabularies which can be complimentary or derogatory. Through initial artificial filtering, partial invalid reviews are deleted. The dataset used as experimental testing data is quite reasonable. English text data and Chinese text data are treated in different ways. We need to do the following: shift uppercase letters to lowercase ones, eliminate punctuation, stop words and uncommon phrases (frequency <5), and use porter stemmer to extract root.

Algorithm 1 Feature extraction algorithm

Input: Review data set $\{d_1, d_2, ..., d_{|D|}\}$, Feature seed word $\{T_1, T_2, ..., T_k\}$, Dictionary V, Selection threshold p, Iteration number I

Output: According to the features of goods isolated reviews

1. Separate reviews from sentences, $X = \{x_1, x_2, ..., x_M\}$;
2. In X, each sentence is matched with a set of seeds, and Record matching rate is $Count(i)$;
3. Mark a feature label for each sentence $a_i = \arg\max_i Count(i)$; if it has the same value, it will have more than one label for the sentence;
4. Calculate the x^2 value of each word in V;
5. Sort the words in each feature according to the x^2 value, *top-p* the words into the features of the seeds of the word list T_i;
6. If the seed feature word list is no longer changed, or the number of cycles is I, go to step 7, otherwise go to step 2;

7. Output sentences with marked feature words.

During feature extraction, we firstly define artificially a group of seed feature words; next, we use the bootstrapping algorithm to add relative features. In the experiment, we choose threshold p and p=5, iteration I=10 times. Table 3 lists out initially artificial defined feature words.

Table 3. Manually defined seed feature words

Features	Seed feature word
Value	value, price, quality, worth
Room	room, suite, view, bed
Location	location, traffic, minute, restaurant
Cleanliness	clean, dirty, maintain, smell
Check in/Front Desk	stuff, check, help, reservation
Service	service, food, breakfast, buffet
Business Service	business, center, computer, internet

To avoid calculating missing values of vector and achieve better calculation results, each review must contain a description of the seven kinds of features mentioned above. In this case, only 184 hotels and 780 pieces of reviews can meet this condition. From the whole quantity of reviews, the big number of reviews is adequate to manifest the average rating level of the user group for hotels. Hence, the experiment combines all reviews of the same one hotel to form a holistic review (h-review), to estimate the average rating value of each hotel. After this, there remains about 1900 hotels and 108,891 pieces of review. Table 4 lists the general statistical information of the experimental data.

Table 4. Corpus statistics

Number of hotels	1900
Review number	108,891
Average number of sentences per review	8.45+0.2
Average number of words per feature	9,58+5.2

After word cutting, hotels' source data contains the following metadata. After BootStrapping, we fetch out a few core fields which fit model calculation, with each review formalized to a Vector. Review, Aspect, Vector respectively contains fields as below. It is shown in Table 5.

Table 5. Data format

Data format	Include field
Review	Author, Content, Date, Number of Reader, Number of Helpful Judgment, Overall rating, Value aspect rating, Rooms aspect rating, Location aspect rating, Cleanliness aspect rating, Check in/front desk aspect rating, Service aspect rating and Business Service aspect rating
Aspect	Author, Content, Date, Ratings (Overall, Value, Rooms, Location, Cleanliness, Check in/front desk, Service and Business Service) and Aspect Segments (Overall, Value, Rooms, Location, Cleanliness, Check in/front desk, Service and Business Service)
Vector	Hotel_ID,Overall_Rating,Value_Rating,Room_Rating,Location_Rating,Cleanliness_Rating,Check_in/front_desk_Rating, Service_Rating, Business_Service_Rating

Because the Review and Vector vectors are too complex, here only the Aspect vector is listed as an example. It is shown in Figure 3.

```
<Author>JaneMichigan009
<Content>High Tech Room We had a 20 hour layover in Amsterdam and Citizen M was perfect. The
staff was wonderful. The room was like nothing we had ever stayed in before. The rooms are tiny
- like in New York City but it seems spacious and very cleaverly put together. Enjoyed every
minute of our stay. It is a short walk to airport and train station. FABULOUS!
<Date>Jan 3, 2009
<Rating>5      5      4      5      5      4      4      5
<Aspects>
3      29(minute):1    51(enjoy):1     270(fabulous):1
0
10     41(city):1      60(hour):1      74(perfect):1    111(spacious):1 338(tiny):1     361
(amsterdam):1   424(york):1     3836(tech):1    3917(layover):1 3944(citizen):1
5      7(walk):1       82(station):1   95(airport):1    144(short):1    182(train):1
0
2      0(staff):1      48(wonderful):1
0
0
```

Figure 3. Aspect vector instance

Each hotel has a unique number ID. In the source data, all true score data are marked for 0-5 points. -1 points is empty data, and there is no score in the web page. Finally, 263 features were selected, and 137653 effective users were selected.

4.2.2. Experimental Results of Individualized Recommendation

After extraction of product features, annotate the semantic contents of review corpora: product feature word, opinion word, and degree word, and then obtain the frequency of semantic contents and length of review text. To make the experimental procedure more explicit, we firstly define the following nouns and then describe the process of the algorithm.

1. User set: $U_m = \{u_1, u_2, ..., u_m\}$

2. To be recommended (hotel): $I_N = \{i_1, i_2, ..., i_N\}$

3. Product feature set: $F = \{f_1, f_2, ..., f_n\}$

4. m user on the n merchandise (hotel) score: $R_{m\times n} = \{R_{ij}\}$

Algorithm 2 Personalized recommendation algorithm

Input: Review data: $\{d_1, d_2, ..., d_{|D|}\}$, Nearest neighbor number k , Recommended number p ;

Output: p Hotel recommended by the target user u ;

1. Get the user set, the product set, and the overall score;
2. Extract the features of the goods, to obtain the overall evaluation of each user to the evaluation and the evaluation of the overall evaluation of the number of signs;
3. Generate the User-feature matrix, using the TF-IDF algorithm [14] to calculate the preference matrix C ;
4. Based on matrix C , using the Pearson correlation coefficient to calculate the similarity of users, select the $top\text{-}k$ nearest neighbors of the target user;
5. Generate the User-Item matrix, and generate the total score matrix R ;
6. Predict the target user of the product that has not yet been reviewed, and the $top\text{-}p$ product will be recommended to the target user.

The experiment is made to examine the accuracy of predictive rating values of commodities waiting for the recommendation. Since experimental data differ from common numeric type data used by the traditional recommendation system, the selection of evaluation indicators is largely limited. Here, we use the mean absolute error (MAE) to evaluate. MAE measures the error between the predicted rating by the proposed model and user real rating, so it is quite suitable to confirm the effectiveness of the proposed algorithm. In the experiment, we note that the choice of similar user number k affects the result of MAE. Here, we define k =5, 7, 10 separately to validate the result. Table 6 lists the comparison of partial prediction results and the user's true score.

Table 6. The recommended results of the predictive score and the true score

Hotel_ID	User	True score	k =5	k =7	k =10
618542	Blaize	5	4	4.5	4.5
	circuit	5	5	5	5
74806	Boeing	4	4	4	3.5
	Blaize	4	4	4	3.5
	Scott	3	4	4	4
256659	Harvsman	4	3.5	3.5	3.5
	Boeing	4	4	4	4
81444	Amy	4	4.5	4.5	4.5

As can be seen from Table 6, when k =5, the average accuracy of the predictive score is higher. From Table 7, we see that when k =5, the method achieves the best result. This is because in practical data, large user groups and commodities can inevitably lead to the problem of data sparsity. Only the users with the highest similar interests are chosen. It has the ideal prediction result. Therefore, when data is sparse, the threshold for choosing the nearest neighbor should not be too high.

Table 7. MAE in different K values

Nearest neighbor K	5	7	10
MAE	0.55	0.56	0.58

4.3. Online Recommendation based on Multi-Objective Decision Making

4.3.1. Multi-Objective Recommended Review Data

The data used in this paper is a review of the online mobile phone and is from the http://www.it168.com website. It is China's largest individual and enterprise IT products to buy and has an interactive website. The data is relatively objective, and the reviews are more professional. We downloaded from the site the four equivalent prices of mobile phone reviews, including the Nokia-N96, Nokia-N97, Nokia-C6, and MOTO-XT. For each cell phone, we randomly selected 100 reviews. The experimental selection of such data is motivated by the following considerations: the four phones have a large number of online reviews, are at the same price, and have a consistent configuration. The average score of these four phones is similar. It is difficult for potential consumers to make a purchase decision.

4.3.2. Multi-Objective Recommended Experimental Results

Table 8 lists six frequent features extracted from the mobile phone data. The second column lists the original frequencies that appear in online reviews. The third column lists the approximate words of these features, in a very common phenomenon involving Chinese synonyms. We have screened out the features with synonyms from the first column. The final column lists the final frequency of the commodity features.

Table 8. Frequent features of mobile phones

Features	Original frequency	Homoionym	Final total frequency
keyboard	150	Key,Button	218
Battery	207		208
screen	117	Shell	178
Effect	83	Sound effects / sound effects, sound quality / tone / volume	166
function	138		137
Price	112	Price, value	128
Appearance	80	Style, appearance, modelling, Color	115
Memory	96		96
system	93		93
Software	89		89
Displayer	74		74
Speed	71		71
Design	68		68
Photograph	39	Take pictures	66
video	63		63
Camera	62		62

Table 9 lists the fuzzy decision matrix and feature weight in this experiment, which indicates the importance of the user's attention to the product features in the reviews, using the input data of Fuzzy PROMETHEE. Table 10 is listed as seven models of the phone's fuzzy positive output flow, fuzzy negative output flow, and net output stream. By calculating the expected value of the net output stream, it is easy to obtain the ranking results of the goods. MOTO-XT＞Nokia-N97＞Nokia-N96＞Nokia-C6.

Table 9. Fuzzy decision matrix and feature weight

Features	Weight	Fuzzy decision matrix			
		Nokia-N97	Nokia-N96	Nokia-C6	MOTO-XT
keyboard	0.119	(0.311,0.501,0.682)	(0.191,0.379,0.575)	(0.292,0.486,0.682)	(0.194,0.379,0.575)
Battery	0.113	(0.124,0.310,0.509)	(0.094,0.258,0.458)	(0.137,0.318,0.518)	(0.178,0.371,0.570)
screen	0.098	(0.385,0.583,0.766)	(0.304,0.5,0.695)	(0.284,0.483,0.679)	(0.439,0.639,0.813)
Effect	0.091	(0.295,0.488,0.683)	(0.407,0.605,0.793)	(0.262,0.444,0.635)	(0.504,0.704,0.864)
function	0.075	(0.361,0.558,0.751)	(0.388,0.587,0.777)	(0.359,0.559,0.749)	(0.394,0.593,0.783)
Price	0.07	(0.133,0.311,0.510)	(0.133,0.314,0.514)	(0.212,0.401,0.601)	(0.229,0.422,0.619)
Appearance	0.63	(0.460,0.660,0.828))	(0.334,0.530,0.719)	(0.352,0.550,0.745)	(0.458,0.658,0.846)
Memory	0.052	(0.369,0.561,0.746)	(0.482,0.682,0.869)	(0.198,0.384,0.584)	)(0.140,0.340,0.540)
system	0.051	(0.409,0.609,0.793)	(0.273,0.465,0.657)	(0.317,0.515,0.714)	(0.388,0.588,0.775)
Software	0.049	(0.207,0.396,0.591)	(0.403,0.603,0.800)	(0.302,0.500,0.692)	(0.331,0.531,0.724)
Displayer	0.04	(0.289,0.485,0.677)	(0.253,0.447,0.647)	(0.231,0.417,0.616)	(0.445,0.645,0.82)
Speed	0.039	(0.324,0.512,0.698)	(0.373,0.568,0.758)	(0.38,0.58,0.76)	(0.547,0.747,0.913)
Design	0.037	(0.364,0.558,0.754)	(0.303,0.498,0.688))	(0.293,0.487,0.685)	(0.308,0.508,0.704)
Photograph	0.036	(0.312,0.512,0.799)	(0.3,0.5,0.695)	(0.338,0.533,0.725)	(0.203,0.386,0.586)
video	0.034	(0.305,0.505,0.7)	(0.462,0.662,0.846)	(0.154,0.328,0.522)	(0.348,0.548,0.744)
Camera	0.034	(0.31,0.509,0.709)	(0.315,0.515,0.703)	(0.338,0.538,0.728)	(0.189,0.389,0.584)

Table 10. The calculation results of Fuzzy PROMETHEE

Goods	$\overline{\phi}^{+}(a_i)$	$\overline{\phi}^{+}(a_i)$	$\overline{\phi}^{+}(a_i)$	$E(\phi(a))$	Sort
Nokia-N97	(0,0.663,2.981)	(0,0.584,2.995)	(0,0.079,-0.014)	0.037	2
Nokia-N96	(0,0.527,3.003)	(0,0.766,2.914)	(0,-0.239,0.089)	-0.097	3
Nokia-C6	(0,0.37,2.912)	(0,0.839,3.003)	(0,-0.469,-0.091)	-0.257	4
MOTO-XT	(0,1.070,2.959)	(0,0.441,2.942)	(0,0.629, 0.017)	0.315	1

The method proposed in this paper has a practical effect on the online reviews of users. From the experiments, we see that the final sorting results are consistent with the average ranking of mobile phones on the IT168 website. Moreover, the fuzzy decision matrix of fuzzy PROMETHEE calculation and the index weight can also provide the key commodity feature recognition, which can provide the depth information of the merchandise. For example, the four mobile phones' top five hot features are "keyboard," "battery," "display," "effect," and "function". That is, users are most concerned about these five features of mobile phones. The MOTO-XT is significantly better than the other three phones in terms of battery, sound, screen, and appearance of these features.

5. Conclusions

This paper presented a personalized recommendation model based on the online review. Two kinds of empirical tests are carried out using real data. Based on opinion mining for the personalized recommendation, the experimental results show that the proposed model can effectively predict the user's expectations and achieve good results. The multi-objective decision-making method is introduced into an online review mining experiment, an auxiliary experiment for online opinion mining, hoping to improve the recommendation effect to some extent. The results of the experiments are consistent with the real product ranking and description, and it also proves the effectiveness of multi-objective decision making in opinion mining.

References

1. H. Huang, "Visual Analysis of Online Social Network", *Journal of the Chinese Academy of Sciences*, vol.2, pp.229-237, 2015
2. Y. M. Lin, X. L. Wang, T. Zhu, "Review of the Research on the Quality Control and Detection of User Reviews", *Journal of Software*,vol.3, pp. 506-527,2014
3. Z. Y. Li, "Study on The Utility Order Model of Online Commodity Review", *Modern Library and Information Technology*, vol. 4, pp. 62-68,2013
4. X. Q. Tan, S. He, "A Review of The Research on Music Personalized Recommendation System", *Modern Library and Information Technology*, vol.9, pp.22-32, 2014
5. X. B. Tang, Z. Zhang, "based on Mixed Graphs Online Social Network Personalized Recommendation System", *Information Theory and Practice*, vol.2, pp. 91-95, 2013
6. W. Wang, X. D. Huang, "Study on The Affective Recommendation System", *Information and Control*, vol.2, pp. 218-228,2013
7. W. Wang, H. W. Wang, Y. Meng, "Collaborative Filtering Recommendation Algorithm Research: Considering Online Review of Affective Tendency", *System Engineering Theory and Practice*, vol.12, pp. 3238-3249, 2014
8. H. S. Xia, "Online Reviews in The Preference Recognition Method of Commodity Attribute", *User information magazine*, vol.9, pp. 197-201,2012
9. W. Xiong, Y. J. Guo, "The Text Mining of Hotel Customer Online Reviews", *Journal of Beijing International Studies University*, vol.11, pp. 38-47,2013
10. M. Yang, W. Qi, X. B. Yan, "Analysis of Online Product Reviews the Utility", *Journal of Management Sciences*,vol.5, pp. 65-75, 2012
11. J. F. Zhong, "Based on The Emotion Lexis of Online Product Reviews Personalized Recommendation Method", *Zhengzhou University Journal (NATURAL SCIENCE EDITION)*, vol.2, pp. 48-51, 2011
12. J. F. Zhong, "Fuzzy Intelligent Product Recommendation System based on Online Consumer Reviews", *System Engineering*, vol.11, no.1, pp. 116-120, 2013
13. L. Zhang, Z. X. Chen, "Analysis and Prediction of The Personality of Social Network Users", *Journal of Computer Science*, vol.8, pp.1877-1894, 2014
14. S. J. Zhao, "Based on Semantic and TF-IDF Project Similarity Calculation Method", *Computer Age*, vol.5, pp. 1-6, 2015
15. Y. J. Zhang, Y. L. Du, "Group Recommendation System and Its Application", *Journal of Computer Science*, vol.4, pp.745-764, 2016

Ningbin Zhang received her M.S degree in Engineering from Xidian University. She is a lecturer at the School of Information Engineering of Xi'an University. Her research interests include database technique and electronic commerce.