Research Article

Implicit Feedback Recommendation Method Based on User-Generated Content

Bing Fang , Enpeng Hu, Junyang Shen, Jingwen Zhang, and Yang Chen

Department of Information Management, School of Management, Shanghai University, Shanghai 200444, China

Correspondence should be addressed to Bing Fang; melodyfang@shu.edu.cn

Received 10 August 2021; Revised 30 September 2021; Accepted 15 October 2021; Published 28 October 2021

Academic Editor: Pengwei Wang

Copyright © 2021 Bing Fang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Studying recommendation method has long been a fundamental area in personalized marketing science. The rating data sparsity problem is the biggest challenge of recommendations. In addition, existing recommendation methods can only identify user preferences rather than customer needs. To solve these two bottleneck problems, we propose a novel implicit feedback recommendation method using user-generated content (UGC). We identify product feature and customer needs from UGC using Convolutional Neural Network (CNN) model and textual semantic analysis techniques, measure user-product fit degree introducing attention mechanism and antonym mechanism, and predict user rating based on user-product fit degree and user history rating data. Using data from a large-scale review sites, we demonstrate the effectiveness of our proposed method. Our study makes several research contributions. First, we propose a novel recommendation method with strong robustness against sparse rating data. Second, we propose a novel recommendation method based on the customer need-product feature fit. Third, we propose a novel approach to measure the fit degree of customer needs-product feature, which can effectively improve the performance of recommendation method. Our study also indicates the following findings: (1) UGC can be used to predict user ratings with no user rating records. This finding has important implications to solve the sparsity problem of recommendations thoroughly. (2) The customer need-based recommendation method has better performance than existing user preference-based recommendation methods. This finding sheds light on the necessity of mining customer need for recommendation methods. (3) UGC can be used to mine customer need and product features. This finding indicates that UGC also can be used in the other studies requiring information about customer need and product feature. (4) Comparing the opinions of user review should not be solely on the basis of semantic similarity. This finding sheds light on the limitation of existing opinion mining studies.

1. Introduction

In the past decade, with the rapid development of online retailing, recommender systems have deeply affected the daily life of people. When people search for a particular product, they will be recommended several products according to their preferences. When they read books or watch movies, the corresponding commodities will be recommended to them. These all show that much of our daily life is invisibly guided by the recommender system. Recommender systems also bring huge benefits to online retailers. For instance, 30% sales of Amazon are increased by the application of recommender system [1]. Researchers find that a minor improvement in the quality of recommender systems can bring millions of dollars in revenue every year to every online retailer [2].

Given the enormous prospects in the promoting product sales, studying the recommendation method to match products and target user has long been a fundamental area in personalized marketing science. So far, recommender system technology still faces great challenges. According to a survey conducted by Tencent, 86% of users have used recommender systems, but more than half of them believe that only a small part of the products recommended can meet their own needs [3]. It reveals that the existing recommender methods fail to satisfy needs of customers, leaving huge room for improvement.

Among them, the rating data sparsity problem is the biggest challenge faced by all existing recommendation methods. The existing mainstream recommendation methods include content-based recommendation methods,

collaborative filtering methods, hybrid recommendation algorithms, and rule-based recommendation methods [2, 4–6]. They are all overreliant on user rating records. With the decrease of rating records, the accuracy of recommendation methods will drop sharply, which brings the rating data sparsity problem. In recent years, major advances have been made in overcoming the sparsity problem. For example, to improve the performance of matrix factorization recommendation method, which is one of the most popular modern recommendation methods, R. Du et al. [7] add user attribute information, Liu et al. [8] add product content information, Yulong Gu [9] adds contextual information, He et al. [10] and Rong-Ping Shen et al. [11] add user feedback information, and Li and Guo [12] add user local characteristics. These studies have alleviated the sparsity problem to a certain extent, but they are still unable to predict user ratings without user rating record.

To completely solve the sparsity problem, implicit feedback recommendation has gradually become one of the most fascinating recommendation research areas. Existing implicit feedback recommendation methods recommend products mainly using user purchase history [13]. For example, some of them utilize user video browsing history or purchase history to recommend videos or products [14]. In fact, both user rating and user purchase history can only be used to identify user preferences, but they do not contain other detailed information about customer need. The reason why users buy products is that the products can satisfy their needs. Therefore, existing recommendation methods can only identify user preferences rather than customer needs, which will inevitably affect their recommendation performance.

To solve the problems mentioned above, we propose a novel implicit feedback recommendation method using user-generated content (UGC). UGC is the content generated by users to express their views on people, events, and things. It can not only fully express user real ideas on people, events, and things, but also express their subjective feelings [15]. UGC has become one of the most important data sources for big data business analysis [16]. We propose method predicts user ratings based on customer need identified from UGC and can effectively predict user ratings without any user rating record or user purchase history. To demonstrate the superiority of our proposed method, we compare it with several benchmark methods including Convolutional Matrix Factorization (ConvMF) [17], Neural Graph Collaborative Filtering (NGCF) [18], Deep Factorization-Machine based Neural Network (DeepFM) [19], Probabilistic Matrix Factorization (PMF) [20], and User-based Collaborative Filtering (CF) [21].

The remainder of the paper is organized as follows. In Section 2, we review relevant previous research and discuss the differences between our proposed method and existing methods. In Section 3, we propose a novel personalized implicit feedback recommendation method using user-generated content in detail. To demonstrate the superiority of our proposed method, in Section 4, we evaluate its effectiveness on real data using representative existing methods as benchmarks. Finally, in Section 6, we summarize

the findings of this study, discuss them, and conclude with the future work.

2. Literature Review

2.1. Research on Recommendation Algorithm. The core technology recommendation system is recommendation algorithm. Existing recommendation algorithms mainly include content-based recommendation algorithms, collaborative filtering recommendation algorithms, hybrid recommendation algorithms, and recommendation algorithms based on association rules.

The content-based recommendation algorithm is to analyze the product content to establish the similarity relationship between the products and then recommend similar products with high user ratings [22]. It recommends items based on the product features extracted from product content information. For example, Koren et al. [23] use the information extracted from movie description, such as movie category, actor, and director, to compare the similarity of movies. Deldjoo et al. [24] extract video features from video content using analysis techniques. Shu et al. [5] learn implicit features in product description text using convolutional neural network. Yu et al. [25] extract image features from image content using image analysis technology. Yong Wu et al. [26] extract the label that can describe the product from the text information of the product. In summary, existing content-based recommendation research extracts product features from product content information (such as the description of the product by merchants), and the product content information provided by merchants does not fully reflect the product, which will inevitably affect the matching accuracy of the product and the target user.

The collaborative filtering recommendation algorithm analyzes user preferences through user rating records to match products with target users. This kind of recommendation algorithm only needs user rating record data to achieve matching, so it has become the most widely used recommendation algorithm. Collaborative filtering recommendation algorithms can be divided into two categories: memory-based collaborative filtering recommendation and model-based collaborative filtering recommendation. Based on rating records, the memory-based collaborative filtering recommendation algorithm analyzes the similarity of user preferences or the similarity of products through rating records and then recommends high-scoring products purchased by users with similar preferences or high-scoring products similar to those purchased by users [4, 5, 27–30]. This type of recommendation algorithm is very sensitive to rating data. Once the rating data is sparse, its performance will drop sharply, and it is unable to obtain the user's preference for the specific features of the product or to match the recommendation with the target customer. The model-based collaborative filtering recommendation algorithm, which can also be named as matrix factorization (MF) recommendation algorithm, trains the relationship between products and users, users and users (or between products and products), through user history rating data, and can still accurately match products and target customers when the

rating data is sparse [23, 31–33]. It is noted that MF recommendations are still based on user rating history records. Without rating records, it cannot work at all.

Hybrid recommendation algorithms avoid or make up for the weaknesses of their respective recommendation algorithms by combining content-based recommendation algorithms and collaborative filtering recommendation algorithms. It recommends items based on both the product features extracted from product content information and user history rating records. For example, Toon De Pessemier et al. [6] propose a hybrid algorithm based on content recommendation, collaborative filtering recommendation, and knowledge-based recommendation. Cai Biao et al. [34] propose an improved dual-parameter hybrid recommendation algorithm, which applies particle swarm optimization (PSO) to the parameter optimization of the hybrid recommendation algorithm. Li et al. [35] propose a hybrid recommendation algorithm based on content and user collaborative filtering to solve the problems of data sparsity and cold start. Although hybrid recommendation can solve the problem in collaborative filtering recommendation to a certain extent, especially sparsity and cold start, it still cannot work without user history rating records.

In addition, there exists another kind of recommendation methods: implicit feedback recommendation method. They are mainly based on association rules and learn the association rules between products and users based on the user's purchase history [2, 36–38]. Association rules are very widely used pattern recognition algorithms, which are used in shopping analysis and network analysis. Implicit feedback recommendations do not require rating records, but they need user shopping purchase history, which will also bring sparsity problem.

2.2. Researches on Customer Needs Mining Based on User-Generated Content. Customer personalized needs belong to the category of the user personalized behavior. Using user-generated content data (UGC) to analyze user behavior has become very hot in recent years, which is widely used in online marketing [39], public opinion analysis [40], and social media operations [41]. UGC can help companies understand customer needs more fully and deeply, so as to (1) improve product design [42]; (2) manage and innovate products [43, 44]; (3) analyze user preferences for product features [45, 46]; and (4) analyze products competitiveness [47]. These studies have fully proved that UGC is an important source of extracting customer needs. However, these studies mainly focus on mining the needs of user groups for product characteristics; they rarely involve the mining of individual needs of customers. To achieve accurate personalized recommendation, it is necessary to further study how to mine individual personalized needs from UGC.

There also exist researches using UGC for the performance improvement of recommendation methods. Utilizing the text mining techniques, they propose hybrid recommendation methods that combine user opinion mined from UGC with traditional recommendation methods. These research works are based on both UGC and rating records.

For example, using text sentiment classification technique, mine user opinion information from UGC to improve matrix factorization recommendations [48–50], collaborative filtering recommendations [51], hybrid recommendations [52], sequential recommendations [53], and cross-domain recommendations [54]. These researches have demonstrated that UGC can be used to improve the performance of current recommendation methods. However, they still rely on user history rating records or user purchase history. In addition, they only focus on the sentiment analysis of UGC, failing to perform in-depth customer needs mining from UGC.

In summary, current recommendation methods are mainly divided into content-based recommendations, collaborative filtering recommendations, hybrid recommendations, and rule-based recommendations (implicit feedback recommendations). These four categories of recommendation methods have their own merits, but they all face the same challenge that when the data is sparse, the performance of the recommendation algorithm will drop sharply. Researchers alleviate the sparsity problem by considering information mined from UGC, although they rely on user rating history records and cannot work at all without these records. On the other hand, existing recommendation methods can only identify user preferences rather than customer needs, which will inevitably affect their recommendation performance. In order to solve the challenges of the abovementioned related research, we propose a novel implicit feedback recommendation method using UGC.

3. A Proposed Personalized Recommendation Method Using User-Generated Content

As shown in Figure 1, the proposed method consists of six stages

- (i) *Stage 1.* Text preprocessing: We perform text preprocessing such as word segmentation, depunctuation, and stop word removal on the extracted raw UGC.
- (ii) *Stage 2.* Identifying informative sentences: We train word embedding with a CBOW model to map words onto a numerical vector space that can be calculated and then train the convolutional neural network model to identify the informative sentences that can present customer needs.
- (iii) *Stage 3.* Identifying topic of informative sentences: We extract key words from informative sentences with K-means clustering algorithm to construct a key word vocabulary and mark the topic of each informative sentence based on the constructed key word vocabulary.
- (iv) *Stage 4.* Constructing product feature vector and customer needs vector: We calculate the sentence vector of the informative sentences, group all the informative sentences with the same topic and calculate central vector of each sentence group, use

FIGURE 1: Implicit feedback recommendation system architecture based on user-generated content.

text sentiment analysis technology to identify customer need sentences from the informative sentences, and construct product feature vector and customer needs vector.

- (v) *Stage 5. Measuring user-product fit degree:* We measure user-product fit degree according to the following three steps: (1) We measure the extent of need the user has for product feature. (2) We measure customer need-product feature fit degree by introducing attention mechanism and antonym mechanism. (3) We measure user-product fit degree.

- (vi) *Stage 6. Predicting user ratings:* We predict users' rating of each product based on user-product fit degree and user history rating data.

3.1. Preprocessing User-Generated Content. We preprocess all user-generated content texts through three steps: Chinese word segmentation, punctuation, and stop words removal.

Chinese word segmentation is a necessary step of Chinese text preprocessing. Unlike English, there are no word boundaries in Chinese sentences; therefore when Chinese natural language processing is performed, word segmentation is usually required first. We use the Chinese word

segmentation tool, Jieba library, to operate the word segmentation.

After the word segmentation result is obtained, we automatically delete Chinese and English punctuation marks and only keep the text. Finally, we use a common Chinese stop word database to remove stop words that are irrelevant to semantic analysis (such as “the” and “and” in English) in the text. One example of text preprocessing is shown in Figure 2.

It can be seen from Figure 2 that, after the text preprocessing, excessive and invalid words are removed from the raw UGC data.

3.2. Identifying Informative Sentences from User-Generated Content. UGC usually contains substantial amounts of content that does not represent product feature or customer needs [55], such as “this is my first time to this store” and “especially come to this store to pull weeds,” which can hurt the performance of recommendation method. Informative sentence of product feature may include the following: “This store is located in the city center,” “The waiter in this store is very enthusiastic,” and so on. Informative sentence of customer need may include “this store is super convenient for parking,” “this dish is very spicy and super delicious,” and so on. Therefore, before extracting customer need information, it is necessary to identify the informative sentences from UGC.

We use a Convolutional Neural Network (CNN) model to identify informative sentences. The approach can be divided into the following two steps.

- Step 1: We use the Word2vec model to train the preprocessed UGC into word embedding that can be calculated. The input of Word2vec model is the preprocessed corpus text, and the output is a word embedding (each word embedding is 200-dimensional).
- Step 2: We train a classification model to identify informative sentences using convolutional neural network (CNN) model. The judgment of informative sentences can be regarded as a binary classification problem; EC_i is used to represent whether the first i comment sentence is informative:

$$EC_i = \begin{cases} 1, & \text{the first } i \text{ comment sentence is informative} \\ 0, & \text{the first } i \text{ comment sentence is noninformative} \end{cases} \quad (1)$$

CNN is one of the representative algorithms of deep learning, which has been widely used in classification tasks. It is a feedforward neural network that includes convolution calculation and deep structure.

We first manually label a small set of sentences into informative/noninformative to construct training data: the informative sentences are labeled as 1, and the noninformative sentences are labeled as 0. We then train and apply a CNN classification model. Using the trained

FIGURE 2: One example of text preprocessing.

classification model, we filter out noninformative sentences from the rest of the raw UGC. Here, the input of the classification model is the sentence embedding of each sentence, and the output of the model is the classification result of each sentence. Table 1 shows some examples of classification results of informative sentences.

3.3. Identifying Topic of Informative Sentences. Since informative sentences may express different categories of product features, we need to identify the category of product features each sentence expresses. The approach can be divided into the following two steps, as shown in Figure 3.

Step 1: We extract key words from informative sentences to construct a key word vocabulary. The K -means algorithm is used to cluster the word embedding into several text clusters. We randomly select a word from each cluster result as the representative word, and experts judge whether the representative word expresses the product feature and further identify the topic of the cluster. For instance, a cluster result contains words such as “sour, sweet, bitter, and fragrant.” We randomly select “sour” as the representative word. Experts identify “sour” can express the product feature and mark the topic of “sour” as “taste.” Since the semantics of words in the same cluster result are similar, we further mark the whole cluster result as a key cluster result and mark its topic as “flavor”. We use the symbol g_i to represent a key cluster, where i represents the i -th category. A key cluster result is

composed of multiple word embedding belonging to this category:

$$g_i = \{x_i^{(1)} \dots, x_i^{(k)}, \dots\}. \quad (2)$$

Here, $x_i^{(k)}$ represents that the k -th vocabulary belongs to the i -th key cluster result. The m -th vocabulary belongs to the i -th key cluster result. Finally, we put all the words in the key cluster results into a vocabulary, which is called the key word vocabulary and denoted as G .

$$G = \{g_1, \dots, g_k\}. \quad (3)$$

Step 2: We mark the topic of each informative sentence as the topic of the key word which most appears in the sentence. For instance, a sentence is “this dish is really fresh.” Since it contains the keyword “fresh,” and the topic of “fresh” is “flavor,” the sentence will be marked as the topic of “flavor.” By applying these above steps, we mark the topic of each informative sentence, which is recorded as $s_{i,j}^{(m)}$. Here, $s_{i,j}^{(m)}$ represents that the subject of the m -th sentence in the i -th user/product review is j .

Figure 4 shows one example of the process of topic identification of informative sentences.

3.4. Constructing Product Feature Vector and Customer Needs Vector. Our proposed recommendation method predicts user ratings based on user-product fit degree and user

TABLE 1: Examples of classification results of informative sentences.

Classification labels	Preprocessed user-generated content
1	PS service pretty bad add sauce long time ask dish name be entirely ignorant not know a thing often wrong dish
0	Squid stone pot mix rice classmate
1	Surroundings small quite warm and sweet

FIGURE 3: Identify the category of product features each sentence expresses.

FIGURE 4: One example of the process of topic identification of informative sentences.

history rating data. We measure user-product fit degree according to customer needs and product feature. Therefore, we need to construct product feature vector and customer needs vector from informative sentences. As shown in Figure 5, we construct product feature vector as the following two steps.

Step 1: We calculate the sentence vector of the informative sentences of product feature according to the word embedding, as shown in the following formula:

$$\overrightarrow{\text{sen}} = \frac{\sum_{i=1}^n \vec{x}_i}{n}. \quad (4)$$

Here, $\overrightarrow{\text{sen}}$ represents the sentence vector, $\vec{x}_i$ represents the word embedding of the i -th word in the sentence, and n represents the number of words contained in the sentence. The examples of sentences and their sentence vectors are shown in Table 2.

Step 2: We construct feature vector of each product. Since for each product, there exist various categories of features, we need to show each category of features when constructing product feature vector. For each product, we group all the informative sentences with the same topic; each sentence group is denoted as T_i^j , as shown in the following formula:

FIGURE 5: The process of constructing product feature vector.

$$T_i^j = \{\overrightarrow{\text{sen}}_1, \dots, \overrightarrow{\text{sen}}_k, \dots, \overrightarrow{\text{sen}}_n\}. \quad (5)$$

Here, i represents the i -th product; j represents the j -th topic; $\overrightarrow{\text{sen}}_k$ represents the k -th sentence vector. It is worth noting that each topic represents each category of features of products.

Using K -means algorithm, we calculate central vector of each sentence group to construct product feature vector. Each product feature vector is denoted as $\vec{v}_i$, as shown in the following formula:

$$\vec{v}_i = \{C_i^1, \dots, C_i^k, \dots, C_i^m\}. \quad (6)$$

Here, i represents the i -th product; C_i^k represents the central vector of k -th sentence group.

We construct customer needs vector as the following three steps, as shown in Figure 6.

TABLE 2: Examples of sentences and their sentence vectors.

Sentence	Sentence vector
The environment is OK, the sofa seat is comfortable	$[5.30886855e-02, -1.93466116e-02, \dots, 3.88030037e-03, -7.75748491e-04]$
The serving speed is very fast	$[-0.14112341, -0.00157748, \dots, 0.1034796, -0.1947528]$

FIGURE 6: The process of constructing customer needs vector.

Step 1: We identify the informative sentences of customer needs from the informative sentences of product features.

Customer needs can be mined from the informative sentences of product features. It is worth noting that only positive sentences can express customer needs. For instance, “the package is very cheap and cost-effective” is a positive evaluation, which expresses the customer need is a cost-effective product. On the contrary, the users’ negative evaluation sentences can only express the features the customer dislikes, but it cannot express the features customers need. For example, “This dish is unpalatable” is a negative evaluation. It expresses that the user does not like the taste of the restaurant, but it cannot directly express what flavor of the dish the user wants to eat. Therefore, we need to further identify the informative sentences of customer needs from the informative sentences of product features.

We put the informative sentences of product features into a sentiment classifier. The sentiment classifier is a naive Bayes classifier, which calculates the probability of a positive sentiment under the condition that a sentence has n words. Table 3 shows some examples of sentiment classification results.

We identify the sentences with the classification label of positive sentiment as the informative sentences of customer needs.

The remaining steps are similar to the steps of feature vector construction: we calculate the sentence vector of the informative sentences of customer needs and further construct need vector of each customer. Each customer need vector is denoted as $\vec{u}_j$.

3.5. Measuring User-Product Fit Degree. To predict user ratings, we measure user-product fit degree. We measure user-product fit degree as the following three steps.

Step 1: We measure the extent of need the user has for product feature. In fact, even if two users have the same need, they usually have the different extents of need for

product feature. For example, both user A and user B need good service and delicious food, but user A places more weight on good service when he chooses a restaurant, while user B places more weight on delicious food. We measure the extent of need one user has for product feature according to the following formula:

$$w_i^k = \frac{\text{count}_i^k}{\sum_{k=1}^m \text{count}_i^k}. \quad (7)$$

Here, w_i^k represents the extent of need user i has for k -th feature, count_i^k represents the counts of informative words expressing the k -th feature in user i comments, and $\sum_{k=1}^m \text{count}_i^k$ represents the counts of all the informative words in user i comments.

Step 2: We measure customer needs-product feature fit degree. We measure customer needs-product feature fit degree not just according to the semantic similarity between product feature vector and customer needs vector. We find that the semantic similarity between product feature vector and customer needs vector does not surely have a positive relation with the fit degree of customer needs-product feature. For example, there exist two sentences: “the service of the store is too slow,” which expresses the feature of a store; “I really like the service here, it’s fast and good,” which expresses the customer need. The semantic similarity between these two sentences is quite high, while the fit degree of customer needs-product feature that the two sentences express is extremely low.

To solve this problem, we introduce attention mechanism to measure the fit degree of customer needs-product feature more accurately. Attention mechanism can capture and highlight the components of sentence vectors which express customer needs or product feature. Each product feature vector $\vec{v}_i$ and customer need vector $\vec{u}_j$ are composed of sentence vector groups. Using attention mechanism, we calculate attention weight of elements in each sentence

TABLE 3: Examples of sentiment classification results.

Text	Positive sentiment probability
The service is really nothing to say, great!	0.8956884152946737
The small hot pot is very affordable, everything is quite fresh, and the quantity is OK.	0.9720190091868546
Getting worse	0.18266397817735047
In view of the disappointment for the first time, I will not visit this store again this month.	0.04318843318986032

vector, which measures the contribution of each element to expressing product feature or customer need, and recalculate the sentence vector with considering attention weight of each element. Figure 7 shows the process of the sentence vector recalculation.

Here, $\text{Sim}_{i,j}$ represents the similarity between i -th element and j -th elements in one sentence vector, and x_i represents the value of i -th element:

$$\text{Sim}_{i,j} = \frac{x_i \cdot x_j}{\|x_i\| \cdot \|x_j\|}, \quad (8)$$

a_i represents the attention weight of i -th element, and L represents the length of sentence vector:

$$a_i = \text{Softmax}(\text{Sim}_{i,j}) = \frac{e^{\text{Sim}_{i,j}}}{\sum_{j=1}^L e^{\text{Sim}_{i,j}}}. \quad (9)$$

$\vec{u}^{\text{Att}}$ and $\vec{v}^{\text{Att}}$ represent, respectively, the recalculated sentence vector of product feature and sentence vector of customer need, as shown in formulas (10a) and (10b):

$$\vec{u}^{\text{Att}} = \sum_{i=1}^L a_i \cdot x_i, \quad (10a)$$

$$\vec{v}^{\text{Att}} = \sum_{i=1}^L a_i \cdot x_i. \quad (10b)$$

Besides, two words with the opposite meanings usually have the similar word embedding, which will also influence the accuracy of measurement of the fit degree. We introduce the antonym mechanism to solve this problem. Using antonym dictionaries, we distinguish whether customer need and product feature express the contrary meaning, which is denoted as $A_{i,j,k}$:

$$A_{i,j,k} = \begin{cases} 1 & \text{customer need and product feature do not express the contrary meaning} \\ -1 & \text{customer need and product feature express the contrary meaning} \end{cases} \quad (11)$$

Finally, we measure customer need-product feature fit degree:

$$f_{i,j,k} = \frac{\vec{u}_{i,k}^{\text{Att}} \cdot \vec{v}_{j,k}^{\text{Att}}}{\sqrt{\vec{u}_{i,k}^{\text{Att}}^2 \cdot \vec{v}_{j,k}^{\text{Att}}^2}} * A_{i,j,k}. \quad (12)$$

Here, $f_{i,j,k}$ represents the fit degree of k -th feature between customer i need and product j ; $\vec{u}_{i,k}^{\text{Att}}$ represents the customer need vector of user i for k -th feature; $\vec{v}_{j,k}^{\text{Att}}$ represents the product feature vector of product j on k -th feature.

Step 3: We measure the user-product fit degree:

$$F_{i,j} = \frac{\sum_{k=1}^m f_{i,j,k} * w_i^k}{m}. \quad (13)$$

Here, $F_{i,j}$ represents the fit degree between customer i and product j ; w_i^k represents the extent of need user i has for k -th feature; m represents the counts of features of product j .

3.6. Predicting User Ratings. Our proposed recommendation method predicts user ratings based on user-product

fit degree and user history rating data. Besides user-product fit degree, we also consider user history rating data when predicting user ratings. The scoring standard of individuals is diverse. Some users may rate average score while they are not very satisfied with the product. On the contrary, some users also rate average score while they are fairly satisfied with the product. We make use of user history rating data to measure the scoring standard of each user.

We estimate the distribution of user rating according to user history rating data, as shown in formula (14):

$$\text{ratio}_i(x) = \frac{\text{count}_i(x)}{\sum_{x=1}^5 \text{count}_i(x)}, x \in \{1, 2, 3, 4, 5\}. \quad (14)$$

Here, $\text{ratio}_i(x)$ represents the ratio of the rating x user i gives to the all history data of user i ; $\text{count}_i(x)$ represents the counts of the rating x in the history rating data of user i .

We predict rating of user i to product j according to Algorithm 1:

It is noted that if there are not user history rating data, we will predict user ratings just based on user-product fit degree. As described earlier, user-product fit degree is measured just based on UGC. This means we can predict user ratings without user history rating records. In this way, our proposed method can solve the sparsity problem in recommendation methods.

FIGURE 7: Process of the sentence vector recalculation.

4. Empirical Evaluations

We demonstrate the effectiveness of our method with data from a large-scale Chinese review site.

4.1. Dataset. We collect data from the public dataset, published by jinhuairst on github (collected by Professor Yongfeng Zhang from conference papers of WWW 2013, SIGIR 2013, SIGIR 2014 conference papers). The dataset includes 4.4 million review/rating data of 540,000 users on 240,000 restaurants. The data description is shown in Table 4.

To solve the problem of missing data existing in the raw data, we further filter data by the following two standards: (1) The count of reviews/ratings on one product is more than 50; (2) the count of reviews/ratings of one user is more than 30. The filtered dataset for conducting experiment includes 403,527 review/rating data of 9,807 users on 16,504 restaurants. Four graduate students manually label about 10,000 sentences with the classification label of 1 as the informative sentences and 0 as the noninformative sentences. Then we divide the dataset into training set and test set according to the ratio of 75% and 25%.

4.2. Evaluation Procedure. We demonstrate the superiority of our proposed method under three scenarios: (1) we compare the performance of our proposed method with the benchmark methods using the filtered dataset; (2) we gradually randomly delete the number of ratings of the training dataset to build several sparse datasets and compare the performance of our proposed method with the benchmark methods; (3) we compare the performance of our

proposed method with two designed recommendation methods using the sparse datasets.

The benchmark recommendation methods include ConvMF [17], NGCF [18], DeepFM [19], PMF [20], and CF [21]. ConvMF is a recommendation method based on textual reviews, which combines CNN and Probability Matrix Factorization (PMF). It is a very effective recommendation method using UGC. NGCF captures user-item interactions to learn vector representations, showing good recommendation performance. DeepFM is a recommendation method combining factorization machines and deep learning for recommendation, with no need of feature engineering besides raw features [19]. PMF is a matrix decomposition recommendation method, which can handle large and sparse datasets. CF is most widely used recommendation method.

As described earlier, in order to improve the performance of our proposed recommendation method, we propose a novel approach to measure the fit degree of customer needs-product feature and a novel approach to measure the extent of need the user has for product feature. To evaluate the effect of these two approaches on the performance improvement of our proposed recommendation method, we also design two recommendation methods, which are denoted as NUD (Non-User-Need) recommendation method and NP (Non-Preference) recommendation method. The NUD recommendation method differs from our proposed method in one way: it measures the fit degree of customer needs-product feature just according to the semantic similarity between product feature vector and customer comment vector. It is worth noting that customer comment vector is not equal to customer need vector: customer comment vector contains both the positive information and negative information of user comments, while customer need vector only contains positive information of user comments. The NP (Non-Preference) recommendation method differs from our proposed method in one way: it predicts user ratings without considering the extent of need the user has for product feature. Table 5 shows methods compared in the evaluations.

4.3. Evaluation Results and Analyses. Following the evaluation procedure, we conduct evaluations to compare the proposed method and the benchmarks. The performance of each method is evaluated using RMSE. RMSE is a standard metric for assessing methods that predict user ratings [56]. According to Alejandro et al. [57], RMSE is the root mean squared error between the observed value and the true value. Lower values of RMSE equate to better performance of methods. The formula of RMSE is as follows:

$$\text{RMSE}(R) = \sqrt{\frac{1}{m} * \frac{1}{n} \sum_{j=1}^n \sum_{i=1}^m (R_{i,j} - \bar{R}_{i,j})^2}. \quad (15)$$

Here, $R_{i,j}$ represents the predict rating of user i to product j , and $\bar{R}_{i,j}$ represents the true rating of user i to product j .

```

(i) Input: history rating data of user  $i$ ,  $F_{i,j}$ 
(ii) Output:  $R_{i,j}$ 
(2)  if  $F_{i,j} \leq \text{ratio}_i(1)$ :
(3)    then  $R_{i,j} = 1$ 
(4)  else if  $F_{i,j} \leq \text{ratio}_i(1) + \text{ratio}_i(2)$ :
(5)    then  $R_{i,j} = 2$ 
(6)  else if  $F_{i,j} \leq \text{ratio}_i(1) + \text{ratio}_i(2) + \text{ratio}_i(3)$ :
(7)    then  $R_{i,j} = 3$ 
(8)  else if  $\text{Matching} \leq \text{ratio}_i(1) + \text{ratio}_i(2) + \text{ratio}_i(3) + \text{ratio}_i(4)$ :
(9)    then  $R_{i,j} = 4$ 
(10) else:
(11)   then  $R_{i,j} = 5$ 

```

ALGORITHM 1: Predicting $R_{i,j}$.

TABLE 4: Data description.

Data field	Data description
User_Id	User id
Rest_Id	Restaurant id
Rating	Rating for the restaurant, values are between 1 and 5
Rating_env	Rating for the restaurant environment, values are between 1 and 5
Rating_dish	Rating for the dishes of restaurant, values are between 1 and 5
Rating_service	Rating for the service of restaurant, values are between 1 and 5
Timestamp	Time when user posts reviews
Review	Users textual review on each restaurant

In Table 6, we show RMSE of the proposed method (PM) and those of the benchmark methods using the filtered dataset.

Several observations warrant attention. First, the best performance results come from our proposed method: compared with NGCF, DeepFM, ConvMF, PMF, and CF, the RMSE value of PM reduces by 0.7%, 6.5%, 7.7%, 7.8%, and 10.1%, respectively. Second, the performance of NGCF and DeepFM is significantly better than other benchmark methods, and the performance of NGCF is slightly better than the latter one. Third, the performance of ConvMF and PMF recommendation method is better than the collaborative filtering method. Last, the performance of ConvMF method incorporating CNN is slightly better than the PMF method.

As described in evaluation procedure, to evaluate the performance of our proposed method, we gradually randomly delete the number of ratings of the training dataset to build several sparse datasets. These sparse datasets contain 100%, 80%, 60%, 40%, and 0% of rating data, respectively. In Table 7 and Figure 8, we show RMSE of the PM and those of the benchmark methods using the sparsity datasets.

Several observations warrant attention. First, the PM have good performance even when there is no rating data in the dataset, while all the other benchmark methods are unable to predict any ratings in this scenario. Second, the best performance results come from PM no matter which sparsity datasets we use, and the RMSE values of PM show the increasing least as the reduction of the number of

ratings in the sparsity datasets, which justify the robustness of PM against the sparsity problem. Third, the second best performance results come from NGCF and DeepFM, no matter which sparsity datasets we use. This suggests that NGCF and DeepFM can solve the sparsity problem to some extent, although they still cannot work without rating records. Fourth, the RMSE values of ConvMF are nearly about those of PMF when using the sparse datasets containing more than 60% of rating data. However, when using the sparse datasets containing less than 60% of rating data, the RMSE values of ConvMF increase sharply, while those of PMF increase slowly. It shows the poor robustness of ConvMF against the sparsity problem. Last, the RMSE value of CF increases slowly with the reduction of the number of ratings in the sparsity datasets, while the RMSE values are persistently high no matter which sparsity datasets we use.

In Table 8 and Figures 9 and 10, we show RMSE of the PM and those of two design recommendation methods (NUD method and NP method) using the sparsity datasets.

Several observations warrant attention. First, the best performance results still come from our proposed method. Second, the RMSE values of NP are lower than those of NUD no matter which sparsity datasets we use. These justify that our approach of measuring the fit degree of customer needs-product feature can effectively improve the performance of recommendation, and the extent of need the user has for product feature should be taken into considerations when predicting user ratings.

TABLE 5: Methods compared in the evaluations.

Method	Method description
PM	Our proposed recommendation method
NUD	Recommendation method measuring the fit degree of customer needs-product feature just according to the semantic similarity between product feature vector and customer comment vector
NP	Recommendation method predicting user ratings without considering the extent of need the user has for product feature

TABLE 6: RMSE values of the proposed and benchmark methods using the filtered dataset.

	PM	NGCF	DeepFM	ConvMF	PMF	CF
RMSE	0.7667	0.7719	0.8197	0.8311	0.8316	0.8526
PM reduce	--	0.7%	6.5%	7.7%	7.8%	10.1%

TABLE 7: RMSE values of proposed and benchmark methods using the sparse datasets.

The proportion of rating data (%)	PM	NGCF	DeepFM	ConvMF	PMF	CF
100	0.7667	0.7719	0.8197	0.8311	0.8316	0.8526
80	0.7690	0.7910	0.8400	0.84094	0.8442	0.8709
60	0.7719	0.8072	0.8787	0.8687	0.8690	0.8896
40	0.7778	0.8683	0.9072	0.9604	0.9269	0.9134
0	0.9944	--	--	--	--	--

TABLE 8: RMSE values of the proposed and design methods using the sparse datasets.

The proportion of rating data (%)	PM	NUD	NP
100	0.7667	0.7733	0.7706
80	0.7690	0.7746	0.7729
60	0.7719	0.7787	0.7765
40	0.7778	0.7844	0.7823

FIGURE 8: Comparative RMSE analyses of the proposed and benchmark methods using the sparse datasets.

FIGURE 9: Comparative RMSE analyses of PM and NUD method using the sparse datasets.

FIGURE 10: Comparative RMSE analyses of PM and NP method using the sparse dataset.

5. Conclusion and Future Work

Sparsity problem is always the major challenge of recommendation system. Existing recommendation systems predict user ratings mainly according to user preferences. In fact, user buy products because of their need for products, but user preference cannot cover the details of customer need, which will greatly restrict the performance of recommendation system. In order to solve these two bottleneck problems, we propose a novel implicit feedback recommendation method using UGC. We identify product feature and customer needs from UGC using CNN model and textual semantic analysis techniques, measure user-product

fit degree introducing attention mechanism and antonym mechanism, and predict user rating based on user-product fit degree and user history rating data.

Our study makes several research contributions. First, we propose a novel recommendation method with strong robustness against sparse rating data. It can effectively predict user rating even when there is no rating data in the dataset, while all the other benchmark methods are unable to predict any ratings in this scenario. Second, we propose a novel recommendation method based on the customer need-product feature fit. The performance of the proposed method is better than benchmark recommendation methods based on user preference, no matter the rating dataset is sparse or not. Last, we propose a novel approach to measure the fit degree of customer needs-product feature, which can effectively improve the performance of recommendation method.

Two main conclusions are found in our study: (1) UGC can be used to solve the sparsity problem of recommendations thoroughly. Almost all recommendation methods have been inseparable from user rating records or shopping cart records. Rating sparsity has always been the bottleneck problem of recommendation system. Our finding suggests UGC can provide an alternative to rating records for user rating prediction and thereby solve the sparsity problem. (2) The customer need-based recommendation method has better performance than existing user preference-based recommendation methods. This finding sheds light on the necessity of mining customer need for recommendation methods. Existing recommendation methods predict user ratings according to user preference, which will inevitably limit their prediction accuracy.

Our study also indicates two conclusions related to UGC analysis: (1) UGC can be used to mine customer need and product features. UGC records user real experience after using products, and it contains a wealth of detailed information about customer need and product feature. This

finding indicates that UGC can be used not only in recommendations, but also in the other studies requiring information about customer need and product feature. (2) The semantic similarity between reviews does not surely imply they have the same opinion. Comparing the opinions of user review should be not solely on the basis of semantic similarity. This finding sheds light on the limitation of existing opinion mining studies.

Our study also has several implications for practice. First, using our proposed recommendation method, companies can effectively recommend their product to target customers and significantly increase their conversion rate of recommendations to get more profit. Second, our proposed method can also help companies to deeply understand customer needs and make marketing strategy to meet the individual need of customers.

Our study can be extended in several directions. First, our proposed method requires human involvement when identifying informative sentences from UGC. Future research should further study novel approaches to recognize informative sentences merely through machine learning technologies. Second, the opinion mining of sentences cannot rely solely on the sentence vector. In this study, we introduce attention mechanism to solve this problem. Future research will further investigate novel approaches to mine opinion of sentences more accurately. Third, there are spam reviews widespread in UGC, which will affect the recommendation performance. Future research will study how to identify and filter spam reviews in UGC to further improve the performance of recommendation methods using UGC.

Data Availability

Data used in this study is collected from the public dataset, published by jinhuakst on github (collected by Professor Yongfeng Zhang from conference papers of WWW 2013, SIGIR 2013, SIGIR 2014 conference papers), which can be downloaded through https://github.com/SophonPlus/ChineseNlpCorpus/blob/master/datasets/yf_dianping/intro.ipynb.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported by Humanity and Social Science Youth Foundation of Ministry of Education of China (Grant no. 18YJC860006).

References

- [1] K. Hosanagar, D. Fleder, D. Lee, and A. Buja, "Will the global village fracture into tribes? recommender systems and their effects on consumer fragmentation," *Management Science*, vol. 60, no. 4, pp. 805–823, 2014.
- [2] A. Ghoshal, S. Menon, and S. Sarkar, "Recommendations using information from multiple association rules: a probabilistic approach," *Information Systems Research*, vol. 26, no. 3, pp. 473–636, 2015.
- [3] X. Wang, "Smart cool survey | "guess you like" intelligent recommendation is accurate? Only 24% of users said it was reliable [EB/OL]," 2015, <https://tech.qq.com/original/archives/a058.html>.
- [4] F. Xue, X. He, X. Wang, J. Xu, K. Liu, and R. Hong, "Deep item-based collaborative filtering for top-N recommendation," *ACM Transactions on Information Systems*, vol. 37, no. 3, pp. 1–25, 2019.
- [5] J. Shu, X. Shen, H. Liu, B. Yi, and Z. Zhang, "A content-based recommendation algorithm for learning resources," *Multimedia Systems*, vol. 24, no. 2, pp. 163–173, 2018.
- [6] T. D. Pessemier, J. Dhondt, and L. Martens, "Hybrid group recommendations for a travel service," *Multimedia Tools and Applications*, vol. 76, no. 2, pp. 2787–2811, 2017.
- [7] R. Du, J. Lu, and H. Cai, "Double regularization matrix factorization recommendation algorithm," *IEEE Access*, vol. 7, pp. 139668–139677, 2019.
- [8] H. Liu, C. Ling, L. Yang, and P. Zhao, "Supervised convolutional matrix factorization for document recommendation," *Computational Intelligence and Applications*, vol. 17, no. 4, pp. 185–189, 2018.
- [9] Y. Gu, J. Song, W. Liu, L. Zou, and Y. Yao, "CAMF: context aware matrix factorization for social recommendation," *Web Intelligence*, vol. 16, no. 1, pp. 53–71, 2018.
- [10] Y. He, C. Wang, and C. Jiang, "Correlated matrix factorization for recommendation with implicit feedback," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 3, pp. 451–464, 2019.
- [11] R.-P. Shen, H.-R. Zhang, H. Yu, and F. Min, "Sentiment based matrix factorization with reliability for recommendation," *Expert Systems with Applications*, vol. 135, pp. 249–258, 2019.
- [12] Y. Li and Y. Guo, "Cultural distance-aware service recommendation approach in mobile edge computing," *Scientific Programming*, vol. 2018, Article ID 2181974, 8 pages, 2018.
- [13] T. Gu, H. Chen, C. Bin, L. Chang, and W. Chen, "Neighborhood attentional memory networks for recommendation systems," *Scientific Programming*, vol. 2021, Article ID 8880331, 10 pages, 2021.
- [14] J. Chen, C. Wang, S. Zhou, Q. Shi, F. Yan, and C. Chen, "SamWalker: social recommendation with informative sampling strategy," in *Proceedings of the WWW'19: The World Wide Web Conference*, pp. 228–239, New York, NY, USA, May 2019.
- [15] B. Fang, Q. Ye, D. Kucukusta, and R. Law, "Analysis of the perceived value of online tourism reviews: influence of readability and reviewer characteristics," *Tourism Management*, vol. 52, pp. 498–506, 2016.
- [16] E. Kauffmann, J. Peral, D. Gil, A. Ferrández, R. Sellers, and H. Mora, "Managing marketing decision-making with sentiment analysis: an evaluation of the main product features using text data mining," *Sustainability*, vol. 11, no. 15, pp. 4235–4253, 2019.
- [17] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional matrix factorization for document context-aware recommendation," in *Proceedings of the 10th ACM Conference on Recommender Systems*, pp. 233–240, ACM, Boston, MA, USA, September 2016.
- [18] X. Wang, X. He, M. Wang, and F. F, "Neural graph collaborative filtering," in *Proceedings of the SIGIR*, pp. 21–25, Paris, France, July 2019.
- [19] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: a factorization-machine based neural network for CTR

- prediction,” in *Proceedings of the IJCAI*, pp. 1725–1731, Melbourne, Australia, August 2017.
- [20] A. Mnih and R. Salakhutdinov, “Probabilistic matrix factorization,” *Advances in Neural Information Processing Systems*, Massachusetts Institute of Technology Press, San Francisco, CA, USA, 2007.
 - [21] Z. Zhao and M. Shang, “User-based collaborative-filtering recommendation algorithms on hadoop,” in *Proceedings of the 2010 Third International Conference on Knowledge Discovery and Data Mining*, pp. 478–481, NW Washington, DC, USA, January 2010.
 - [22] R. Meymandpour and J. G. Davis, “A semantic similarity measure for linked data: an information content-based approach,” *Knowledge-Based Systems*, vol. 109, pp. 276–293, 2016.
 - [23] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
 - [24] Y. Deldjoo, M. Elahi, P. Cremonesi, F. Garzotto, P. Piazzolla, and M. Quadrona, “Content-based video recommendation system based on stylistic visual features,” *Journal on Data Semantics*, vol. 5, no. 2, pp. 99–113, 2016.
 - [25] L. Yu, F. Han, S. Huang, and Y. Luo, “A content-based goods image recommendation system,” *Multimedia Tools and Applications*, vol. 77, no. 4, pp. 4155–4169, 2018.
 - [26] Y. Wu, S. Xi, Y. Yao, F. Xu, H. Tong, and J. Lu, “Guiding supervised topic modeling for content based tag recommendation,” *Neurocomputing*, vol. 314, pp. 479–489, 2018.
 - [27] Y. Pang, Y. Jin, Y. Zhang, and T. Zhu, “Collaborative filtering recommendation for MOOC application,” *Computer Applications in Engineering Education*, vol. 25, no. 1, pp. 120–128, 2017.
 - [28] W. Fu, J. Liu, and Y. Lai, “Collaborative filtering recommendation algorithm intelligent community,” *Discrete and Continuous Dynamical Systems-S*, vol. 12, no. 4&5, pp. 811–822, 2019.
 - [29] T. Wang and M. Wang, “Distributed collaborative filtering recommendation algorithm based on DHT,” *Cluster Computing*, vol. 22, no. S2, pp. 2931–2941, 2019.
 - [30] Z.-P. Zhang, Y. Kudo, T. Murai, and Y.-G. Ren, “Addressing complete new item cold-start recommendation: a niche item-based collaborative filtering via interrelationship mining,” *Applied Sciences*, vol. 9, no. 9, pp. 1894–1910, 2019.
 - [31] M. Fu, H. Qu, Z. Yi, L. Lu, and Y. Liu, “A novel deep learning-based collaborative filtering model for recommendation system,” *IEEE Transactions on Cybernetics*, vol. 49, no. 3, pp. 1084–1096, 2019.
 - [32] K. Li, L. Chen, and L. Tian, “Matrix factorization for video recommendation based on instantaneous user interest,” in *Proceedings of the 2020 4th International Conference on Electronic Information Technology and Computer Engineering*, pp. 596–601, New York, NY, USA, November 2020.
 - [33] M. Nguyen, J. Yu, B. Quan, and S. Yongchareon, “Attentional matrix factorization with document-context awareness and implicit API relationship for service recommendation,” in *Proceedings of the Australasian Computer Science Week Multiconference*, pp. 1–10, New Zealand, February 2020.
 - [34] B. Cai, X. Zhu, and Y. Qin, “Parameters optimization of hybrid strategy recommendation based on particle swarm algorithm,” *Expert Systems with Applications*, vol. 168, Article ID 114388, 2021.
 - [35] L. Li, Z. Zhang, and S. Zhang, “Hybrid algorithm based on content and collaborative filtering in recommendation system optimization and simulation,” *Scientific Programming*, vol. 2021, Article ID 7427409, 11 pages, 2021.
 - [36] S.-h. Liao and H.-k. Chang, “A rough set-based association rule approach for a recommendation system for online consumers,” *Information Processing & Management*, vol. 52, no. 6, pp. 1142–1160, 2016.
 - [37] C. Li, W. Liang, Z. Wu, and J. Cao, “An efficient distributed-computing framework for association-rule-based recommendation,” in *Proceedings of the 2018 IEEE International Conference on Web Services (ICWS)*, pp. 339–342, Seattle, WA, USA, June 2018.
 - [38] B. Liu, Y. Ma, C. K. Wong, and P. S. Yu, “Scoring the data using association rules,” *Applied Intelligence*, vol. 18, no. 2, pp. 119–135, 2003.
 - [39] X. Zhang, Y. Yu, H. Li, and Z. Lin, “Sentimental interplay between structured and unstructured user-generated contents,” *Online Information Review*, vol. 40, no. 1, pp. 119–145, 2016.
 - [40] S. M. Mohammad, P. Sobhani, and S. Kiritchenko, “Stance and sentiment in tweets,” *ACM Transactions on Internet Technology*, vol. 17, no. 3, pp. 1–23, 2017.
 - [41] B. W. Wirtz, V. Göttel, P. F. Langer, and M.-J. Thomas, “Antecedents and consequences of public administration’s social media website attractiveness,” *International Review of Administrative Sciences*, vol. 86, no. 1, pp. 36–61, 2018.
 - [42] K. Young, J. Chen, and S. Raghunathan, “User-generated content and competing firms’ product design,” *Management Science*, vol. 64, no. 10, pp. 4471–4965, 2018.
 - [43] C. Shan and Y. Du, “A web service clustering method based on semantic similarity and multidimensional scaling analysis,” *Scientific Programming*, vol. 2021, Article ID 6661035, 12 pages, 2021.
 - [44] Y. Deng, “Intelligent innovative knowledge management integration method based on user generated content,” *Cluster Computing*, vol. 22, pp. 4793–4803, 2019.
 - [45] W. Anning, Z. Qiang, Z. Shuangyao, L. Xiaonong, and Z. Peng, “A review-driven customer preference measurement model for product improvement: sentiment-based importance - performance analysis,” *Information Systems and E-Business Management*, vol. 18, 2020.
 - [46] K. Berezina, A. Bilgihan, C. Cobanoglu, and F. Okumus, “Understanding satisfied and dissatisfied hotel customers: text mining of online hotel reviews,” *Journal of Hospitality Marketing & Management*, vol. 25, no. 1, pp. 1–24, 2016.
 - [47] O. Netzer, R. Feldman, J. Goldenberg, and M. Fresko, “Mine your own business: market-structure surveillance through text mining,” *Marketing Science*, vol. 31, no. 3, pp. 369–547, 2012.
 - [48] G. Zhao, X. Lei, X. Qian, and T. Mei, “Exploring users’ internal influence from reviews for social recommendation,” *Proceedings of the IEEE Transactions on Multimedia*, vol. 22, no. 3, pp. 771–781, 2019.
 - [49] G. Zhao, P. Lou, X. Qian, and X. Hou, “Personalized location recommendation by fusing sentimental and spatial context,” *Proceedings of the Knowledge-Based Systems*, vol. 196, Article ID 105849, 2020.
 - [50] G. Zhao, Z. Liu, Y. Chao, and X. Qian, “CAPER: context-aware personalized emoji recommendation,” *Proceedings of the IEEE Transactions on Knowledge and Data Engineering*, vol. 99, pp. 3160–3172, 2020.
 - [51] B. Ramzan, I. Sarwar Bajwa, N. Jamil et al., “An intelligent data analysis for recommendation systems using machine learning,” *Scientific Programming*, vol. 2019, Article ID 5941096, 20 pages, 2019.

- [52] S. Kumar, K. De, and P. Roy, "Movie recommendation system using sentiment analysis from microblogging data," *Proceedings of the IEEE Transactions on Computational Social Systems*, vol. 7, pp. 915–923, 2020.
- [53] L. Zheng, N. Guo, W. Chen, J. Yu, and D. Jiang, "Sentiment-guided sequential recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Xi'an, China, July 2020.
- [54] M. Gan and H. Zhang, "DeepFusion: fusing user-generated content and item raw content towards personalized product recommendation," *Complexity*, vol. 2020, Article ID 4780191, 12 pages, 2020.
- [55] A. Timoshenko and J. R. Hauser, "Identifying customer needs from user-generated content," *Marketing Science*, vol. 38, no. 1, pp. 1–20, 2019.
- [56] H. Steck, "Evaluation of recommendations: rating-prediction and ranking," in *Proceedings of the RecSys'13: 7th ACM conference on Recommender systems*, pp. 213–220, New York, NY, USA, October 2013.
- [57] A. Bellogin, P. Castells, and I. Cantador, "Precision-oriented evaluation of recommender systems: an algorithmic comparison," in *Proceedings of the RecSys'11: Proceedings of the fifth ACM conference on Recommender systems*, pp. 333–336, New York, NY, USA, October 2011.