Electron Commer Res (2017) 17:3–29
DOI 10.1007/s10660-016-9240-9

Finding users preferences from large-scale online reviews for personalized recommendation

Yue Ma[1] · Guoqing Chen[1] · Qiang Wei[1]

Published online: 8 October 2016

Abstract Along with the growth of Internet and electronic commerce, online consumer reviews have become a prevalent and rich source of information for both consumers and merchants. Numerous reviews record massive consumers' opinions on products or services, which offer valuable information about users' preferences for various aspects of different entities. This paper proposes a novel approach to finding the user preferences from free-text online reviews, where a user-preference-based collaborative filtering approach, namely UPCF, is developed to discover important aspects to users, as well as to reflect users' individual needs for different aspects for recommendation. Extensive experiments are conducted on the data from a real-world online review platform, with the results showing that the proposed approach outperforms other approaches in effectively predicting the overall ratings of entities to target users for personalized recommendations. It also demonstrates that the approach has an advantage in dealing with sparse data, and can provide the recommendation results with desirable understandability.

Keywords Online review · Recommendation systems · Collaborative filtering · User preference · Opinion mining

✉ Qiang Wei
weiq@sem.tsinghua.edu.cn

Yue Ma
may.10@sem.tsinghua.edu.cn

Guoqing Chen
chengq@sem.tsinghua.edu.cn

[1] Research Center for Contemporary Management, Key Research Institute of Humanities and Social Sciences at Universities, School of Economics and Management, Tsinghua University, Beijing 100084, China

1 Introduction

User-generated online reviews have evolved into a pervasive part of e-commerce nowadays, as well as an essential focus of business intelligence and big data analytics. Both the online retail websites, like Amazon.com and Taobao.com, and the forum websites, such as Dianping.com and TripAdvisor.com, are collecting tremendous amounts of online reviews. Both consumers and companies benefit greatly from the rich and valuable knowledge contained in the reviews [2]. Search efforts have been devoted to developing business intelligence techniques that incorporate online reviews to make personalized recommendations [3, 4].

The mainstream of traditional recommendation approaches is usually based on the commonality among users [1, 11, 26], i.e., similar users or entities are found by measuring the similarities of the common rating scores of users. However, the insufficiency of relevant data such as sparsity significantly weakens the effectiveness of these approaches due to the fact that there are often a limited number of common ratings among users. For example, as shown in Fig. 1, the user frequency in the largest online restaurant review platform in China (www.dianping.com) turned out to be skewed (in log format), revealing that most users only review very few restaurants. This leads to a difficulty in calculating reliable similarities. Such a sparsity phenomenon brings great challenges to inferring user preferences from commonly rated entities.

Except for the ratings by users, the user reviews can offer much finer-grained information, and have become a rich source to help detect the users' preferences. Most of the reviews contain users' opinions on various aspects of the target products/services (referred to as entities). Here, an aspect, also called feature or attribute in literature, refers to a component or an attribute of a certain entity. A sample review on a restaurant like "The attitude of waiters is very good." reveals a positive opinion on the aspect "service", though it uses the phrase "attitude of waiters" rather than "service" directly. User aspect preference implies the importance of the aspects for the users' decisions on product choosing or evaluating processes. Empirical findings have proven that aggregated aspect preferences of a group of users to a product type,

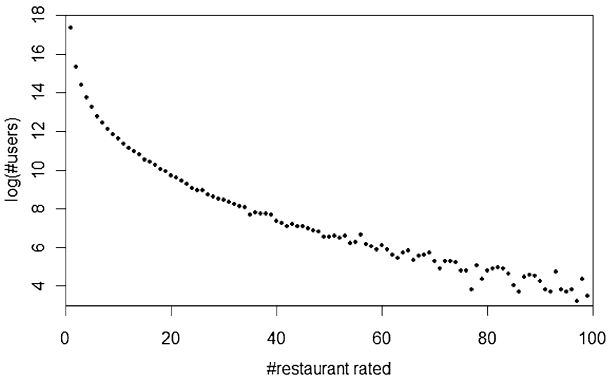

Fig. 1 Frequency of users who rated a certain number of restaurants

such as mobile phones, can be derived from the textual reviews they post, by analyzing the relationship between the user's overall ratings and aspect-level opinions with regression models [6, 38]. Though useful, this kind of models suffers from their limitation in effectively inferring the individual aspect preferences, since one specific user usually posts very small amount of reviews, i.e., the sparsity in reviews, which also weakens the reliability of the models.

In practice, a user's preferences for the aspects of a certain entity, which is hereby referred to as individual aspect preferences, are of great value in developing personalized recommendations. Depending on the individual aspect preferences, a profile can be further constructed for each individual user. Then, the similarities between any two users can be measured by their aspect preference profiles, no matter how many entities they rate commonly, which in fact is of robustness in dealing with the sparsity problem. Furthermore, another benefit of utilizing aspect preferences is the understandability of recommendations. By understandability we mean that the derived aspect preferences of a user reflect more fruitful and granular information about the user, which helps tell why the recommended entities are more appropriate for the user.

Driven by integrating user aspect preferences into recommendation, we propose in this paper a novel approach called user-preference-based collaborative filtering (UPCF) to inferring user aspect preferences from the online reviews, and employing them in improving recommendations. First of all, aspect-level opinions mining should be conducted to transform the free-text reviews to structured aspect opinions. Then, the user preferences can be further captured from two angles: *aspect importance* and *aspect need*. The former is based on the fact that the opinions on those important aspects are more influential to the overall ratings than other aspects, and uses the similarity between the opinions on one aspect and the overall ratings. The latter is measured as the difference between the opinions of a user on an aspect and those of other users, which indicates the differentiated needing level on this aspect with respect to the user. Based on the developed user preferences in light of aspect importance and aspect need, a user-based collaborative filtering approach is devised, in that the user aspect preferences are integrated to calculate the similarities between users.

The rest of the paper is organized as follows. In Sect. 2, related work on recommendation approaches based on reviews and aspect-level opinion mining is discussed. Subsequently, the problem definition and formulation are described in Sect. 3. Section 4 presents the proposed measures of user preferences along with a detailed investigation for a novel approach developed upon these user preferences. The experiments and discussion are presented in Sect. 5. Final conclusion is provided in Sect. 6.

2 Related work

Prior work in the marketing and information systems literature has recognized the importance of finding user preferences for e-commerce activities [6, 38]. A considerable amount of research efforts has been devoted into developing business

intelligence techniques that incorporate user preferences in order to provide personalized recommendation for individual online consumers [10, 13, 18]. Especially, to obtain the fine-grained user opinions and preferences, online reviews are utilized widely in the recent recommendation systems [3, 4]. On the research issues of concern, the related literature could be categorized into two groups, i.e., review-based recommendation, and aspect-level opinion mining, respectively.

2.1 Review-based recommendation

The research of personalized recommendation can be generally classified into collaborative filtering (CF)-based, content-based, and hybrid approaches [1]. CF-based approaches are usually based on commonality among users with historical ratings to infer their preferences on entities [1, 11, 26]. However, the effectiveness of CF approaches is limited since the well-known sparsity problem occurs frequently [10, 17]. To cope with this problem, multiple content-based approaches are proposed to profile users or entities by various types of information, such as entities descriptions [10], tags [20], and social relationships [18, 37], to augment the accuracy of recommendation. On the other hand, hybrid recommender systems aim to combine the advantages of various strategies to make more informative recommendations.

Increasing efforts have been made to incorporate the rich information embedded in user reviews into the process of user profiling and recommendation generation. Several types of review information are exploited to enhance the traditional rating-based recommender systems. For example, the review-level opinion orientations can combine overall opinions with real ratings in a biased matrix factorization model [23]. The helpfulness of reviews can be associated with the ratings to indicate the quality of ratings used in CF [25]. Keywords can be extracted from user reviews to generate indices respectively for the corresponding reviewers and the entities [10]. The target user's index serves as a query to search for the entities with the most similar indices. Also, similarities between the historical reviews of two users are considered to measure the user similarity [30]. The topics obtained by topic models, such as the Latent Dirichlet Allocation (LDA), can also be associated with the latent factors in model-based CF [21, 27] and with the similarity measure in memory-based CF [34].

It is worth mentioning that, though useful, these efforts generally did not take full advantages of the aspect-level information in reviews. The opinions on different aspects can reveal why a user likes (or dislikes) an entity, which depicts the user preferences and is deemed valuable in improving personalized recommendations. Recently, some researchers have used the aspect-level opinion mining results to enhance the recommendation accuracy. For example, a hybrid recommendation model has been extended to uncover the relationships among users, entities and the opinions of specific aspects [13]. In addition, a clustering-oriented user-based CF has been proposed [9], where the aspect opinions are used to measure the user similarities to conduct the user clustering. In these efforts, users' opinions on aspects of entities are incorporated into the recommending process directly.

Although being used to enhance recommendations in existing studies, the direct opinions fail to capture the importance of the aspects in the user's evaluation process, which, however, is a great indicator of the user preferences. Some literature has discussed how to measure such aspect preferences by econometric models [6] or probabilistic regression model (PRM) [38, 39]. Nevertheless, their objective is usually set as a type of entity rather than an individual user. Only a few studies have utilized this kind of information to conduct recommendations. For instance, a linear regression model has been employed to measure the reviewers' aspect weights and use them to generate personalized entity ranking [33]. Apparently, the effectiveness of regression models is limited as far as the sparsity problem prevails, since the amount of a single user's historical reviews is often very small or reviews are usually very short and cover only few aspects. Notably, in line with the spirit of our work, a preference and opinion-based recommendation (PORE) proposed by Liu et al. [17] is worth mentioning. They also measured the preferences with two variables, i.e. *concern* and *requirement*, which are then used for personalized recommendation. But compared with our work, the user preferences for aspects are not comprehensively incorporated. Differently, our work introduces a comprehensive measure for evaluating user aspect preferences, considering not only the aspects' influence to the overall ratings (which reflects how important the aspect is to the user), but also the opinions' differences with those of other users on a specific aspect (which indicates the user's level of need to the aspect). Subsequently, the user preferences on aspects are further employed to measure the user similarities and develop a better CF approach.

2.2 Aspect-level opinion mining

Opinion mining or sentiment analysis comprises an area of natural language processing, computational linguistics and text mining, aiming to determine the sentiment orientations in reviews, sentences or on specific aspects, corresponding to review/document-level, sentence-level and aspect-level opinion mining, respectively [19]. The main task of aspect-level opinion mining is to identify the aspects and analyze the corresponding sentiment polarities, which are typically positive and negative, or the sentiment degrees on the aspects expressed by users.

First, on aspect identification, existing techniques can be divided into supervised and unsupervised approaches. Supervised approaches learn patterns of aspects from a collection of labeled reviews, and then identify aspects in new reviews. The majority of these approaches are based on the structured models, like Hidden Markov Models (HMM), Conditional Random Fields (CRF) [24], and their variations [16, 24]. All of these approaches require sufficient labeled training, which, however, is too time-consuming and labor-intensive to obtain the sufficiently labeled datasets [38]. Several unsupervised approaches have been proposed to tackle this problem. The approach proposed by Hu and Liu [12] is representative. Their approach first extracts nouns and noun phrases, counts the occurrence frequencies of them, and then keeps the frequent ones as aspects. The infrequent aspects could be found by exploiting the relationships between aspects and opinion words. However, the results yielded from this approach always contain many noises and too many

fragmented aspects. Subsequently, several efforts have been made to solve the problem. For example, a phrase dependency parser is used to extract noun phrases from reviews as aspect candidates, followed by employing a language model to filter out the unlikely aspects [35]. Su et al. [29] designed a mutual reinforcement strategy to cluster product aspects and opinion words simultaneously by iteratively fusing both content and sentiment link information. Zhai et al. [40] used semi-supervised learning to cluster features with a small set of labeled examples. In brief, most of these efforts are designed for English contexts, and could hardly be applied in Chinese corpus. Except for the different grammars and habits of the two languages, another reason is the lack of effective Chinese lexicon, like WordNet [8] in English, to obtain lexical similarity [40]. In our work, we use the Chi square statistic [36] to measure distributional similarities between words and a bootstrapping process is also applied to grouping the feature words to aspects, which shows satisfactory output.

The second task is to determine the orientation of opinions expressed on each aspect in a sentence. Supervised learning and lexicon-based approaches are the mainstream. The supervised learning approaches use labeled datasets to train classical classifiers, e.g., Support Vector Machine (SVM), Maximum Entropy (ME) model and Naïve Bayes model etc., to classify the opinions on aspects [22]. The lexicon-based approaches rely on a sentiment lexicon listing the opinion orientation of each word. Some approaches have been proposed to generate a high-quality lexicon [12, 28]. Ding et al. [7] presented a holistic lexicon-based method to improve the approach in [12] by addressing two issues: the opinions of sentiment words would be content-sensitive and the conflicts in the review. In the aspect opinion mining process in our work, the idea in [7] is also adopted to further develop strategies for content-sensitive opinion words and opinion aggregation of each review.

3 Problem definition

Widely available user-generated reviews can be used to model users' preferences more accurately [3, 4]. The users' opinions on the different aspects of an entity, rather than only their overall ratings, deliver more fruitful and granular information. An entity could be a product such as a camera, or a service such as a restaurant. The research problem is to address the following issues: (1) how to infer the users' aspect-level preferences from the opinions in the reviews, (2) how to mine opinions from free-text reviews to infer the users' preferences and (3) how to leverage the user preferences to elaborate recommendations.

Specifically, let $E = \{e_1, e_2, \ldots, e_{|E|}\}$ be a set of entities (e.g., restaurants), $R = \{r_1, r_2, \ldots, r_{|R|}\}$ be a set of reviews about E, and $U = \{u_1, u_2, \ldots, u_{|U|}\}$ be the set of users who wrote these reviews. For each review r_i, it may consist of some sentimental sentences about the corresponding entity's aspects (or features). Let $A = \{a_1, a_2, \ldots, a_{|A|}\}$ be a set of aspects of entities, such as "taste", "environment" and "price" for restaurants, etc. The aspects can be expressed as different feature words, including explicit feature words and implicit feature words [42]. For

example, the aspect "price" could be described by an explicit feature word ''cost'', which is a synonym of ''price'', and could also be implied by the implicit feature word ''expensive''. Let $F_j = \{f_1, f_2, \ldots, f_{|F_j|}\}$ be the set of feature words, where f_l ($l = 1, 2, \ldots, |F_j|$) is an explicit or implicit feature word to describe aspect a_j. In addition, opinions towards aspects are expressed in sentimental sentences. An opinion is represented by words that convey a positive, negative or neutral sentiment to an aspect in a review sentence. For example, in the sentence of "The cost performance is very great", "cost performance" is a feature word of aspect "price" and "great" is the opinion word.

For an opinion on aspect a_j in the reviews, its sentiment orientation can be determined as a numeric score, i.e., denoted as o_j (e.g., $+1$ for positive, -1 for negative and 0 for neutral). Therefore, a pair $\langle a_j, o_j \rangle$ could be detected to show the sentiment orientation for each aspect in a review sentence. As the sentiment orientation o_j is usually measured by a numeric score, similar to the overall rating, it could also be called the rating or the sentiment score to aspect a_j.

Thus, to address the issues of the research problem, an approach called UPCF is proposed. Firstly, a process to identify aspects and corresponding sentiment orientations is conducted. After the aspect-level opinion mining process, a collection of sentiment vectors in the aspect spaces can be obtained, where each vector $\vec{r} = \langle o_1, o_2, \ldots, o_L \rangle$ represents the structured free-text review r in R, and o_j is the sentiment score to the corresponding aspect a_j in the review. Then, the key task is to measure user preferences, i.e., aspect weights, from the structured aspect-level opinion data obtained in the opinion mining. Here, two important measures are introduced to infer the aspect weights for each user from two angles, which will be discussed in detail in Sect. 4. Finally, a recommendation approach is developed as an extended CF approach to recommending the entities to a user based on his/her preferences.

4 A novel approach for user-preference-based collaborative filtering

In this section, the details of the novel UPCF approach are presented. The framework overview of UPCF is illustrated in Fig. 2. The process is composed of three major components: aspect-level opinion mining, user preferences inference, and recommendation generation. The aspect-level opinion mining is the precursor of the user preferences inference to convert the raw data to structured aspect-opinion data. Then, two novel measures are developed, namely aspect importance and aspect need. Based on the measures, the user preferences inference process is constructed, which can effectively infer a user's preferences by considering the influence of the user's opinions given to each aspect over their overall opinions and the user's differentiated need level on each aspect. Finally, with the derived user preferences knowledge, an extended CF process is further devised to generate high-quality recommendation results. As the profiling and inference on user preferences is the crux of the whole approach, we introduce the details of user preference inference in Sect. 4.1. Then the full implementation including all the three components will be discussed in Sect. 4.2.

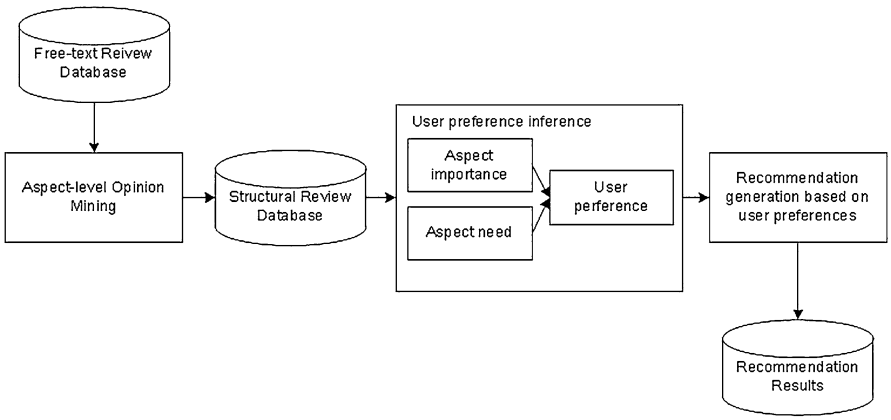

Fig. 2 The framework of UPCF

4.1 User preferences inference

Learning user's preferences to different aspects are helpful in personalized recommendation. One crucial question in the collaborative filtering is how to measure the similarities between users. Traditional approaches focus on the commonality among users or entities and usually only take advantage of overall/summarized rating information, which to a large extent ignores the diversified and granular information on various aspects as well as the differentiated preferences of users on various aspects. Differently, our work focuses on extracting opinions on each aspect from review dataset. The extracted personalized preferences of an individual user can be further measured from the opinions information in the reviews posted by the user. The user preferences on aspects can then be used to scale the similarities between users. Compared with traditional approaches, more personalized information on aspects can be utilized in our approach to improve recommendation performance.

In the decision making process of a particular user, some aspects are more influential than the others, and have greater impacts on the final overall ratings he/she gives. For different users, their perceptions and expectations on a same aspect are not the same. This is why we could easily observe that there are quite a lot of different opinions to one aspect in practice. In this regard, user preferences can be further measured from two angles: the *aspect importance* and the *aspect need*.

The first measure, i.e., *aspect importance*, is to evaluate to what extent that a user perceives the aspect is important to him/her. For instance, a middle-aged businessman might tend to choose the restaurants with considerate service and quiet environment, while a younger person might seek the tasty and cheaper places even though they are crowded and noisy. When they provide feedbacks online, their reviews and ratings may reflect different perceived importance on the aspects, e.g., environment, service, taste, or price. Their ratings highly depend on the important aspects they prefer, e.g., the businessman might rate a restaurant mainly depending on service and environment, while the younger person might care more on taste and

price. That is to say, the user's opinions on the important aspects greatly influence his/her overall opinions on the products/services. The overall rating in a review could be considered as an aggregation of the opinions given to specific aspects in the review, and various aspects have different contributions in the aggregation.

Based on this observation, it can be discovered that, if a user's opinions on an aspect are more consistent/similar to his/her overall ratings on entities than those on other aspects, the user perceives the aspect more important than other aspects.

Previously, regression models are applied to measure the relationship between users' overall ratings and opinions on aspects [6, 14]. These methods assume that a user's overall rating on an entity is the summarization of her/his opinions on different aspects of the entity, so it can be calculated by aggregating the aspect-level opinions. They regard the coefficient assigned to each aspect variable in the aggregation function as the weight that the user gives to that aspect [14]. However, this kind of methods has some limitations in dealing with data sparsity, which is usual in real-world applications. One is that the reviews given by each user for training regression models is quite limited, which significantly weakens the effectiveness and robustness of the regression models. In addition, not all aspects are covered by each review, while most users only mentioned few aspects in online reviews [33]. The large proportion of missing values in datasets hurts the models' performance, even though the absence of opinions on aspects could be treated as neutral. Furthermore, the neutral treatment could easily cause information distortion.

To remedy the above limitations, we propose a novel measure, called *aspect importance*. Given a user u_i and an aspect a_k, the measure *aspect importance* can be defined as follows:

$$aspect{-}importance\ (\mathrm{u_i}, \mathrm{a_k}) = \frac{\sum_{\mathrm{e_j} \in \mathrm{E_i}} \mathrm{O_{ij}} \mathrm{o_{ijk}}}{\sqrt{\sum_{\mathrm{e_j} \in \mathrm{E_i}} \mathrm{O_{ij}^2}} \sqrt{\sum_{\mathrm{e_j} \in \mathrm{E_i}} \mathrm{o_{ijk}^2}}} \tag{1}$$

where o_{ijk} represents the opinions (e.g., sentiment orientation) that user u_i comments on aspect a_k for entity e_j, O_{ij} is the overall rating that user u_i assigns to entity e_j, and E_i is the entity set that user u_i reviewed. If user u_i does not provide opinions to aspect a_k of e_j, then $o_{ijk} = 0$. The *aspect importance* reflects the correlation between user u_i's opinions on aspect a_k and overall rating O_{ij} with respect to all the entities the user reviewed. The value range of *aspect importance* is $[-1, +1]$. When the value is bigger than zero, the overall ratings and the aspect ratings of the user are positively correlated, otherwise negatively correlated.

Clearly, with the above definition, a user's opinions on each aspect he/she has ever rated can be statistically aggregated. Not like traditional regression models, which frequently encounter the difficulty in collecting a sufficient number of reviews with multiple co-occurred aspects, the calculation of *aspect importance* can easily be conducted with a sufficient number of reviews focusing on one aspect each time. This characteristic can help alleviate the sparsity problem in profiling user preferences.

The other measure for evaluating user preferences is called *aspect need*. As above mentioned, the satisfying aspect performances that users expect are not

always the same. For instance, given the same set of products, a critical user may seldom provide positive opinions on quality, but a tolerant user may frequently write "good quality" in the reviews. This phenomenon reveals that different users may have different needs on aspects, i.e., called *aspect need*. Thus, if a user always provides lower ratings on a specific aspect than those of other users, it can be inferred that the user's need for this aspect is higher than those of other users. In other words, if a user's ratings on a specific aspect are always higher than those of others, the user's need for this aspect is lower. The notion of *aspect need* is in line with the idea of *requirement* in [17], while, however, *requirement* in [17] only considers the situation that a user has a higher need for only one certain aspect. In portraying user preferences, the need should be considered from all aspects, which will be investigated in the following discussion.

In consideration of *aspect need*, there are two cases to be explored. One is that user u_i has reviewed aspect a_k several times; the other is that the user has never mentioned aspect a_k at all. In the former case, the *aspect need* can be defined as:

$$aspect{-}need\ (\mathrm{u_i}, \mathrm{a_k}) = \frac{1}{|\mathrm{E_{ik}}|} \sum_{\mathrm{e_j} \in \mathrm{E_{ik}}} \frac{\overline{\mathrm{o_{jk}}} - \mathrm{o_{ijk}} + 1}{\overline{\mathrm{o_{jk}}}}, \text{ where } \overline{\mathrm{o_{jk}}} = \frac{1}{|\mathrm{U_{jk}}|} \sum_{\mathrm{u_i} \in \mathrm{U_{jk}}} \mathrm{o_{ijk}} \quad (2)$$

where E_{ik} represents the set of entities with the reviews written by user u_i covering a_k, U_{jk} denotes the set of users who have reviewed on the aspect a_k of entity e_j, $\overline{o_{jk}}$ is the average rating on aspect a_k of entity e_j. $\overline{o_{jk}} - o_{ijk}$ measures the difference between the rating given by user u_i on aspect a_k of entity e_j and the average rating given by all users. The numerator $\overline{o_{jk}} - o_{ijk} + 1$ guarantees that the measure has a positive value.

When a user has never reviewed on aspect a_k, which means $E_{ik} = \emptyset$, it can be assumed that the user's need for a_k is less than others. To make the value comparable to that in the other case, we consider all the entities whose aspect a_k has been reviewed by users, and a relatively small number 0.1 is used to measure the difference. Let E_k represent the set of entities whose aspect a_k has been commented by users. The *aspect need* in this case can be defined as:

$$aspect{-}need\ (\mathrm{u_i}, \mathrm{a_k}) = \frac{1}{|\mathrm{E_k}|} \sum_{\mathrm{e_j} \in \mathrm{E_k}} \frac{0.1}{\overline{\mathrm{o_{jk}}}}, \text{ where } \overline{\mathrm{o_{jk}}} = \frac{1}{|\mathrm{U_{jk}}|} \sum_{\mathrm{u_i} \in \mathrm{U_{jk}}} \mathrm{o_{ijk}} \quad (3)$$

Then, Eqs. (2) and (3) can be combined as:

$$aspect{-}need\ (\mathrm{u_i}, \mathrm{a_k}) = \begin{cases} \frac{1}{|\mathrm{E_{ik}}|} \sum_{\mathrm{e_j} \in \mathrm{E_{ik}}} \frac{\overline{\mathrm{o_{jk}}} - \mathrm{o_{ijk}} + 1}{\overline{\mathrm{o_{jk}}}} & \text{where } \mathrm{E_{ik}} \neq \emptyset \\ \frac{1}{|\mathrm{E_k}|} \sum_{\mathrm{e_j} \in \mathrm{E_k}} \frac{0.1}{\overline{\mathrm{o_{jk}}}} & \text{others} \end{cases} \quad (4)$$

Therefore, the *aspect need* can be used to measure a certain user's differentiated expectation on a specific aspect with respect to all users.

With the two proposed measures, namely, *aspect importance* and *aspect need*, the user preferences for aspects can be inferred. To obtain the overall preference of

user u_i for aspect a_k, a composite measure *preference*, short for p, is defined. Concretely, the *preference* of user u_i for aspect a_k is defined as follows:

$$p_{ik} = aspect{-}importance\ (\mathrm{u_i}, \mathrm{a_k}) \times aspect - need\ (\mathrm{u_i}, \mathrm{a_k}) \tag{5}$$

Therefore, the *preference* of user u_i for aspect set $A = \{a_1, a_2, \ldots, a_{|A|}\}$ can form a vector $p_i = \langle p_1, p_2, \ldots, p_{|A|} \rangle$, which can be deemed as an aspect-based profiling or portraying for user u_i. Then the similarities between users can be calculated by the aspect preference vectors of users to further generate recommendations.

4.2 Algorithm implementation

In this section, the whole process of UPCF is introduced. As shown in Fig. 2, to infer the user preferences defined in the previous section, a process of aspect-level opinion mining should be conducted first to convert free-text reviews to structured data. After inferring user preferences, a collaborative filtering process based on the user preferences is designed to generate the recommendation results.

The aspect-level opinion mining could be further divided into 4 sub-tasks: preprocessing, aspect identification, feature grouping, and sentiment analysis. The preprocessing is to remove duplicated reviews from raw data, and takes word segmentation and POS tagging to obtain a refined dataset. Then, a boot-strapping strategy is applied to extract feature words and group them into several aspects identified for the entities. And sentiment analysis is conducted to determine the sentiment degree of each aspect in the review and construct a structured review dataset. These sub-tasks are detailed as follows.

(1) *Preprocessing* The raw data of the reviews may often contain the duplicated entries and misspellings, and need to be segmented into words. Hence, preprocessing is considered necessary before data mining. Several steps are taken in this subtask. Firstly, repeated reviews that one user writes about the same entity (e.g., the same restaurant) are removed. Secondly, ICTCLAS[1] is applied to carry out the word segmentation and POS tagging. In this step, some slangs, new Internet buzzwords and names of the common entities (e.g., the same dishes) are added to user dictionary. Finally, the noun, verb, adverb and adjective words are retained for the following subtasks, many of which may likely serve as the candidate features words and opinion words [12, 42].

(2) *Aspect identification* In review websites on products/services, users are usually asked to provide ratings to multiple listed aspects of an entity [14]. For example, a review platform on restaurants may list "taste", "environment", "service" and "price" as the aspects for ratings. On the other hand, there are also some other important aspects that are also discussed widely by users in their reviews, such as "location", "characteristic", "variety", "amount of food", "waiting time", "cleanness", etc. To identify these aspects, the top 100 frequent nouns are extracted based on the result of pre-processing. Then, three experts

[1] http://ictclas.nlpir.org/.

were asked to inspect these words and summarize similar aspects to form a qualified and unanimous aspect list.

(3) *Feature grouping* Based on the derived aspect list, the words describing the same aspect need to be grouped as the feature words of the aspect. Initially, each aspect can be equipped with 3–5 widely-used and unambiguous words as seeds, where 3–5 seed words possess good effect according to empirical test. Other related feature words could be searched out by measuring the dependency between the candidate feature word f and the seed words of aspect a based on the well-known χ^2 statistic [36], defined as follows:

$$\chi^2(f,a) = \frac{N \times (AD - CB)^2}{(A+C) \times (B+D) \times (A+B) \times (C+D)} \tag{6}$$

where A is the frequency of f co-occurring with seed words of a in a short sentence (short sentences are clauses separated by punctuation), B is the frequency of f not co-occurring with any seed words of a in the same short sentence, C is the frequency of short sentences including seed words of a but excluding f, D is the frequency of short sentences excluding both f and any seed words of a, and N is the overall frequency of short sentences. The words with high dependencies to an aspect are grouped to update the corresponding aspect seed word list, which are used to search for other highly dependent feature words in the next loop. The boot-strapping processing is described in Fig. 3 [33].

After the boot-strapping processing with necessary noises checking operations, both the explicit and implicit feature words are grouped for corresponding aspects.

(4) *Sentiment analysis* Finally, a classical and effective sentiment analysis method is applied to identify the orientation of each opinion [7, 19]. First, the adjectives are extracted, which can be treated as users' opinion carriers in reviews and be used to determine their orientations as numeric scores (+1 for positive, −1 for negative and 0 for neutral) with the aid of a Chinese opinion lexicon HowNet and a neutral word list collected by domain experts. Then, the adverbs of degree are aggregated as well to improve sentiment analysis performance, where the intensity of orientations can be expressed by the adverbs of degree. Frequent adverbs occurring near opinion words are assigned with three levels of weight $(w_l, 1, w_h)$, for example (0.8, 1, 1.2), according to their intensity. We multiply the weights with the original scores to obtain the new sentiment scores. Additionally, two general opinion rules [7] are also adopted: (1) *But* rule to determine the orientation of context-dependent adjectives; and (2) *Negation* rule to make the score reversed if a negation word exists. Sometimes users write several sentences about one aspect in a review. For example, restaurant customers may try to describe each dish they have taken. Some of them write a summarization before or after the details, such as "Overall, the taste is satisfying." In this case, these opinions given in the summary sentence are regarded as the sentiment score of the review to the corresponding aspects (e.g., "taste"). If no summarizations given in the review, the average of the sentiment scores related to one aspect is calculated as the aspect score of the review. Hence, a word list, called summary word list, is

Algorithm: Feature Grouping

Input: A collection of reviews $R = \{r_1, r_2, ..., r_{|R|}\}$, a set of aspect seed words lists $\{A_1, A_2, ..., A_{|A|}\}$, candidate feature words V, selection threshold t and iteration step limit M;

Output: expended aspect feature word list.

1. Split all reviews into short sentences, $X = \{x_1, x_2, ..., x_{|X|}\}$;
2. Calculate χ^2 measure of each word in V with each aspect;
3. Get a ranked words list under each aspect with respect to the χ^2 value and join the top t words for each aspect into their corresponding aspect seed list A_q ($q = 1, 2, ..., |A|$);
4. If the aspect feature word list is unchanged or iteration exceeds M, go to Step 5, else go to Step 2;
5. Output expended aspect feature word list $\{A_1, A_2, ..., A_{|A|}\}$.

Fig. 3 The process of feature grouping

also collected to help identify the summary sentence. In addition, users' ranking habits significantly influence the opinions of users [17], because criteria vary with different users when they give ratings to an entity. To eliminate the effect cased be ranking habits, a typical amendment is adopted, i.e., the difference between actual rating and the average rating of the user is used as the final sentiment score [26].

With the process of the above aspect-level opinion mining, a free-text review r is then transformed to a vector $\vec{r} = \langle o_1, o_2, \ldots, o_L \rangle$, where each element is the sentiment score on corresponding aspect in the review. Moreover, for subsequent processing, the sentiment scores are normalized to the range of [1, 6], which is consistent to score range on many real-world review platforms. The structured dataset can be further used for user preferences inference as well as the collaborative filtering.

Next, the calculation of the measures of user preferences can be conducted on the structured data. The major steps of the UPCF are listed in the Fig. 4.

As shown in Fig. 4, the inferred user preferences are used to calculate the users' similarity, which can be further integrated to extend traditional user-based CF. The main idea of user-based collaborative filtering is that users will like the entities recommended by others who share the similar interests with them. Therefore, traditional user-based CF tries to predict the ratings of entities for a particular user based on the entities previously rated by the users who are similar to him. The similarity between user u_i and user u_j is often measured by the Pearson's correlation coefficient, as follows:

Algorithm: UPCF

Input: A collection of reviews, each review is a quad <*u, e, r, O*>, including a user $u \in U = \{u_1, u_2, ..., u_{|U|}\}$, an entity $e \in E = \{e_1, e_2, ..., e_{|E|}\}$, a text review $r \in R = \{r_1, r_2, ..., r_{|R|}\}$, and an overall rating O

Output: the predictive overall ratings of a user u to the entities in E

1. Do pre-processing to R
2. Identify the aspect set $A = \{a_1, a_2, ..., a_{|A|}\}$ and do the feature grouping
3. For each review r:
4. analyze the sentiment to extract a list of pair $\langle a_k, o_k \rangle$, translate r to $\vec{r} = \langle o_1, o_2, ..., o_{|A|} \rangle$, where a_k is an aspect and o_k is the sentiment orientation of r to a_k
5. For each o_{ijk}
6. do Amendment
7. For each entity e_j and each aspect a_k:
8. calculate the $\overline{o_{jk}}$ by Eq.(2)
9. For each user u_i:
10. for each aspect a_k:
11. calculate $Aspect\text{-}importance(u_i, a_k)$ by Eq.(1)
12. calculate $Aspect\text{-}need(u_i, a_k)$ by Eq.(4)
13. calculate $p(u_i, a_k)$ by Eq.(5)
14. For each entity $e_j \in E$:
15. for each user u_i having rated e_j:
16. calculate $sim_p(u, u_i)$
17. calculate predictive overall rating $pred(u, e_j)$

Fig. 4 The algorithm of UPCF

$$sim_o(u_i, u_j) = \frac{\sum_{e_q \in E_i \cap E_j} (O_{iq} - \overline{O_i})(O_{jq} - \overline{O_j})}{\sqrt{\sum_{e_q \in E_i \cap E_j} (O_{iq} - \overline{O_i})^2 \sum_{e_q \in E_i \cap E_j} (O_{jq} - \overline{O_j})^2}} \tag{7}$$

where $\overline{O_i}$ and $\overline{O_j}$ respectively denote the average overall ratings given by user u_i and user u_j to entity $e_q \in E_i \cap E_j$, which is the set of entities they both have rated ever.

Apparently, the similarity in Eq. (7) only considers the overall ratings on the entities they both have rated ever. In a sparse dataset, the common entities between two users are rare, which makes the above calculation of similarity hard to execute. Notably, from the perspective of user preferences, the sparsity problem can be

surmounted to a certain extent, since the number of common aspects shared by users are usually not rare, i.e., the sparsity in user's preference vector can be significantly eliminated. Thus, the user similarities can be effectively calculated based on their preference vectors.

In this way, after deriving the user preference vector from all reviews that a user writes, a Pearson's correlation coefficient between two preference vectors of user u_i and user u_j can be calculated to measure the user similarity, as shown in Eq. (8).

$$sim_p(u_i, u_j) = \frac{\sum_{k=1}^{|A|} (p_{ik} - \overline{p_i})(p_{jk} - \overline{p_j})}{\sqrt{\sum_{k=1}^{|A|} (p_{ik} - \overline{p_i})^2 \sum_{k=1}^{L} (p_{jk} - \overline{p_j})^2}} \tag{8}$$

Then, the most popular aggregation function is also used for rating prediction [26]:

$$pred(u_i, e_k) = \overline{O_i} + \frac{\sum_{u_j \in U_k} sim_p(u_i, u_j) \times (O_{jk} - \overline{O_j})}{\sum_{u_j \in U_k} sim_p(u_i, u_j)} \tag{9}$$

where U_k denotes the set of users who ever write reviews about entity e_k. In this regard, for a certain user u_i, the entity e_k with higher $pred(u_i, e_k)$ will be recommended, since it can be predicted that he/she may rate the entity with a high score, through analyzing his/her preference with respect to other users with similar preferences.

5 Experimental results and discussion

In this section, extensive experiments are conducted to evaluate the effectiveness of the proposed user-preference-based recommendation approach.

5.1 Experimental data and settings

To evaluate the performance of the proposed approach, a pool of data composed of real online restaurant reviews was crawled from a well-known platform in China (http://www.dianping.com/). The reviews of the restaurants located in Beijing were collected, and the total number of reviews is 1,288,673, which were posted by 302,412 users for 3764 restaurants.

On the dianping.com platform, each rating is with a scale from one to five stars. The format of the dataset contains userIDs, restaurantIDs, overall ratings and reviews. Since a text review is required when an overall rating is given by a user, the overall rating and review of every entry in the dataset are not empty. Furthermore, 2 datasets are obtained to represent low and high levels of sparsity by removing the users with less than 5 and 20 reviews, respectively. The level of sparsity can then be calculated as:

$$\text{Sparsity} = \frac{\text{Num. of ratings and reviews}}{\text{Num. of users} \times \text{Num. of items}}.$$

The data description of the two datasets is shown in Table 1.

Root mean square error (RMSE) between predicted and actual ratings is used to measure the performance of different approaches, which can be calculated as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (O_i^{predict} - O_i^{actual})}{n}} \tag{10}$$

where $O_i^{predict}$ is the overall rating score predicted by an approach, and O_i^{actual} is the actual overall rating score, and n is the size of testing data. Clearly, the smaller the RMSE value, the better the performance.

For cross validation purposes, the reviews of each user were divided into five partitions, and four of the partitions were used to identify preferences and one was used to test the performance. In the following result tables, RMSE_i ($i = 1, 2, 3, 4,$ and 5) is the RMSE value when the ith partition of the dataset was used as the test dataset, and the rest as training sets. All experiments were conducted on a computer with a 3.6 GHz Intel Core i7 processor, 16 GB RAM, and 64-bit Windows Server 2008. All algorithms were implemented in Python 2.7 or Java 1.7.

5.2 Performance analysis on the aspect-level opinion mining process

First, an empirical study was conducted to evaluate the performance of the aspect-level opinion mining method described in Sect. 4.2. For this purpose, we randomly selected a set of 100 reviews from the dataset. Two annotators carefully processed each sentence to extract important aspects and mark corresponding opinion orientations. Thus, the annotated reviews were presented in the following format:

Aspect1: Feature1, Opinion1, Sentiment1; Aspect2: Feature2, Opinion2 Sentiment2;…

If the sentiment of a feature was annotated as positive, score 1 was assigned, or if neutral, 0 was assigned, otherwise, if negative, score -1 was assigned. Thereafter, another expert was asked to inspect both of annotators' judgments and to double check the labeled dataset. Then, the finally labelled results were treated as benchmark. The descriptive statistics of test data is as shown in Table 2. With the

Table 1 Descriptive statistics of datasets

Dataset	Data-5	Data-20
Num. of users	60,290	11,978
Num. of entities	3760	3760
Num. of ratings and reviews	919,737	480,767
Average rating	3.850	3.823
Variance of ratings	0.742	0.672
Min num. of entities rated by a user	5	20
Sparsity	0.41 %	1.07 %

Table 2 Descriptive statistics of test data

	Test dataset
Num. of reviews	100
Num. of sentences	438
Frequency of mentions on extracted aspects	369
Frequency of mentions on annotated aspects	420

test data as well as the labelled results, we evaluated the performances, i.e., *precision*, *recall* and *F*1-*measure*, of the aspect-level opinion mining process, i.e., aspect identification and grouping, and sentiment analysis. The calculations of these measures are provided in Eqs. (11–13).

$$precision(a_k) = \frac{\left|S_p(a_k) \cap S_a(a_k)\right|}{\left|S_p(a_k)\right|} \tag{11}$$

$$recall(a_k) = \frac{\left|S_p(a_k) \cap S_a(a_k)\right|}{\left|S_a(a_k)\right|} \tag{12}$$

$$F1(a_k) = \frac{2 \times precision(a_k) \times recall(a_k)}{precision(a_k) + recall(a_k)} \tag{13}$$

where $S_p(a_k)$ denotes the set of sentences covering the opinions to aspect a_k obtained by our approach, $S_a(a_k)$ denotes the set of sentences labelled by annotators as related to a_k. Clearly, high values of *precision*, *recall* and *F*1-*measure* represent good performance.

Table 3 shows the overall evaluation result of each aspect when applying our proposed aspect identification and feature grouping, revealing a satisfactory performance in finding the features words. The aspect "Location" has relatively low precision and recall, because it is difficult to distinguish the "location of the restaurant" and the "location of the seat" in Chinese without further context. Table 4 depicts the evaluation result of overall aspect-level opinion mining, showing the effectiveness of our solution. It can be seen that our aspect-level opinion mining had a quite good performance, which can effectively support the following user preference inference and collaborative filtering processes.

Additionally, to further evaluate the performance sensitivity with respect to weight assignment (w_l, 1, w_h) on the frequent adverbs in sentiment analysis, experiments with five representative weight assignments were conducted. Generally, the weight assignment (1, 1, 1) represents that all adverbs are equally considered, while the other four assignments, i.e., (0.8, 1, 1.2), (0.6, 1, 1.4), (0.4, 1, 1.6) and (0.2, 1, 1.8), representing four different weight assignments on sentiment intensity. The results are detailed in Table 5, showing that the result with weight assignment (0.8, 1, 1.2) was slightly better than those of other assignments. Therefore, in the following experiments, the weight assignment (0.8, 1, 1.2) is employed.

Table 3 Performance of aspect identification and feature grouping

Restaurant aspects	Frequency of mentions on annotated aspects	Precision (%)	Recall (%)	F1-measure (%)
Taste	141	94.31	82.27	87.88
Environment	52	97.83	86.54	91.84
Service	41	100.00	70.73	82.86
Price	48	93.33	87.50	90.32
Location	24	72.73	66.67	69.57
Characteristic	20	84.21	80.00	82.05
Variety	27	100.00	74.07	85.11
Amount of food	23	82.61	82.61	82.61
Waiting time	24	86.96	83.33	85.11
Cleanness	20	100.00	95.00	97.44
Total	420	92.68	81.43	86.69

Table 4 Performance of aspect-level opinion mining

Restaurant aspects	Precision (%)	Recall (%)	F1-measure (%)
Taste	89.43	78.01	83.33
Environment	97.83	86.54	91.84
Service	100.00	70.73	82.86
Price	91.11	85.42	88.17
Location	68.18	62.50	65.22
Characteristic	73.68	70.00	71.79
Variety	100.00	74.07	85.11
Amount of food	82.61	82.61	82.61
Waiting time	78.26	75.00	76.60
Cleanness	100.00	95.00	97.44
Total	89.43	78.57	83.65

5.3 Performance analysis on UPCF and other approaches

In order to demonstrate the performance of the proposed UPCF approach, five mainstream approaches were compared. First, two classical collaborative filtering approaches, i.e., user-based collaborative filtering approach (namely, UCF) and entity-based collaborative filtering approach (namely, ECF), were compared. To keep consistency with the UPCF approach, Pearson correlation was also used to measure the similarities between users and between entities in the two approaches, respectively. In UCF, the top-K most similar users or entities were selected to make the prediction, while in ECF, the K-nearest neighbors were selected to make the prediction, both of which are the general treatments. Parameter K was set as 10,000 and 4000 for datasets Data-5 and Data-20, respectively. These two approaches are well known but do not take user preferences into consideration.

Table 5 RMSE in Data-5 using different kinds of weight assignments

Weight	(1, 1, 1)	**(0.8, 1, 1.2)**	(0.6, 1, 1.4)	(0.4, 1, 1.6)	(0.2, 1, 1.8)
RMSE1	0.77873	**0.77859**	0.77874	0.77872	0.77871
RMSE2	0.77416	**0.77406**	0.77416	0.77415	0.77415
RMSE3	0.76721	**0.76706**	0.76720	0.76720	0.76720
RMSE4	0.76643	**0.76629**	0.76644	0.76643	0.76647
RMSE5	0.76743	**0.76736**	0.76744	0.76750	0.76750
Average	0.77079	**0.77067**	0.77080	0.77080	0.77081

Bold values indicate the best result

Another approach to compare is PORE [17], which also focuses on learning a user's preference with a constructed weight on a certain aspect. Based on the derived weight, a satisfaction degree can be calculated and further be used to generate recommendation. However, PORE showed some deficiencies. First, PORE used the average of scores on four explicit aspects as the overall rating which does not fit the facts in many platforms. Second, only a certain user's aspect weights and the average of users' opinions on these aspects of an entity were taken into consideration in predicting the overall ratings, and the similarities between users and overall ratings of other users were ignored, which help make better predictions. As discussed in previous sections, the UPCF approach carefully deal with these. In experiments, PORE was conducted based on the same structured data preprocessed in UPCF.

In addition, two CF approaches employing reviews to measure the similarities between users were considered in the experimental comparison. One is User-Topic-Interest based collaborative filtering (namely, UTICF) proposed by Wang and Luo [34], where LDA was used to infer the topic probability distributions of the users' reviews, and to aggregate them to uncover user-topic-interest profiles. Though LDA is widely used to deal with free text, it has some shortages: (1) it cannot easily differentiate the feature words and opinion words, which makes it hard to conduct the sentiment analysis to the reviews; (2) Some work has indicated that LDA cannot deal with short text very well, such as reviews [32]; (3) the topics obtained by LDA are various, and there are not corresponding relationships between the product aspects and the topics, which reduces its suitability. The other CF approach considered in comparison is referred to as Opinion-based CF (namely, OCF), which utilizes the similarities between user opinions on common entities, rather than the user preferences on aspects we proposed. Hence, OCF also encounters the challenge of sparsity in user opinions on common entities.

The results of UPCF in comparison with ECF, UCF, PORE, OCF and UTICF on the two datasets are detailed in Tables 6 and 7. It can be discovered that, UPCF outperforms the other 5 approaches. Concretely, except for PORE performing worse (due to the shortages mentioned previously), the approaches considering review information (i.e., OCF, UTICF, UPCF) were generally superior to the approaches considering only overall ratings (i.e., ECF, UCF). Furthermore, UPCF outperformed

Table 6 RMSE on Data-5 with different recommendation approaches

Approaches	ECF	UCF	PORE	OCF	UTICF	UPCF
RMSE1	0.8315	0.7992	0.9597	0.7891	0.7883	**0.7786**
RMSE2	0.8251	0.7938	0.9546	0.7837	0.7831	**0.7741**
RMSE3	0.8151	0.7883	0.9499	0.7764	0.7763	**0.7671**
RMSE4	0.8143	0.7849	0.9469	0.7747	0.7739	**0.7663**
RMSE5	0.8137	0.7850	0.9453	0.7754	0.7738	**0.7674**
Average	0.8199	0.7902	0.9513	0.7798	0.7791	**0.7707**

Bold values indicate the best result

Table 7 RMSE on Data-20 with different recommendation approaches

Approaches	ECF	UCF	PORE	OCF	UTICF	UPCF
RMSE1	0.7435	0.7325	0.9360	0.7181	0.7174	**0.7139**
RMSE2	0.7437	0.7272	0.9334	0.7180	0.7172	**0.7137**
RMSE3	0.7388	0.7108	0.9297	0.7144	0.7143	**0.7100**
RMSE4	0.7372	0.7113	0.9301	0.7130	0.7124	**0.7087**
RMSE5	0.7418	0.7277	0.9304	0.7168	0.7156	**0.7137**
Average	0.7410	0.7219	0.9319	0.7161	0.7154	**0.7120**

Bold values indicate the best result

OCF and UTICF, showing that the user preferences identified by UPCF could help distinguish the differences between users in a better way than other two approaches.

Pairwise t test was also conducted to verify the significance of the RMSE difference between UPCF and other five approaches. Let Δ denote the difference of RMSE between UPCF and any of the other approaches. The hypotheses are: H0: $\Delta = 0$, and H1: $\Delta > 0$. The test results are shown in Table 8. On both datasets, the p values are less than 0.05, meaning that the null hypothesis is rejected, that is to say, the RMSE value of UPCF is significantly smaller than the RMSE value of any one of the other approaches. Finally, it is worth mentioning that the improvements of RMSE we obtained (as shown in Tables 6 and 7) are meaningful, since empirical studies have demonstrated that even a small improvement in a rating prediction error can affect the final performance of the recommender systems, due to the fact that the top entities presented to users may change greatly [15].

Table 8 Pairwise t-test results between different approaches

Pairs	Data-5	Data-20
UPCF versus ECF	1.4078E−09	5.6443E−08
UPCF versus UCF	1.7460E−02	2.9344E−07
UPCF versus PORE	9.2851E−12	1.0643E−11
UPCF versus OCF	4.5404E−06	1.6548E−06
UPCF versus UTICF	1.3946E−04	1.2124E−05

5.4 Performance on hybrid similarity measure

This section discusses the performance when the two similarity measures [i.e., the user-preference-based similarity defined in Eq. (8), and the user-based similarity defined in Eq. (7)] are combined. These two similarity measures are denoted as $sim_p(u_i, u_j)$ and $sim_o(u_i, u_j)$ respectively. Then, a hybrid similarity measure can be defined as:

$$sim_h(u_i, u_j) = \alpha sim_p(u_i, u_j) + (1 - \alpha) sim_o(u_i, u_j) \tag{14}$$

where $\alpha \in [0, 1]$. The prediction can also be carried out by the aggregation function in Eq. (9). Thus, the UPCF with $sim_h(u_i, u_j)$ is called Hybrid-UPCF.

Furthermore, the Hybrid-UPCF was compared with UPCF and UCF. Firstly, several trials were taken to determine the appropriate value of α, and the results of average RMSE values of the five-fold cross validation are shown in Fig. 5. It can be seen that, when α was around 0.6, the best RMSE could be achieved. In addition, the curve is right-skewed, i.e., user-preference-based similarity contributes more than user-based similarity in improving RMSE, which is also consistent with the previous analysis.

The best RMSE values with corresponding α values are listed in Table 9. As shown in Fig. 5 and Table 9, the appropriate combination of the two similarity

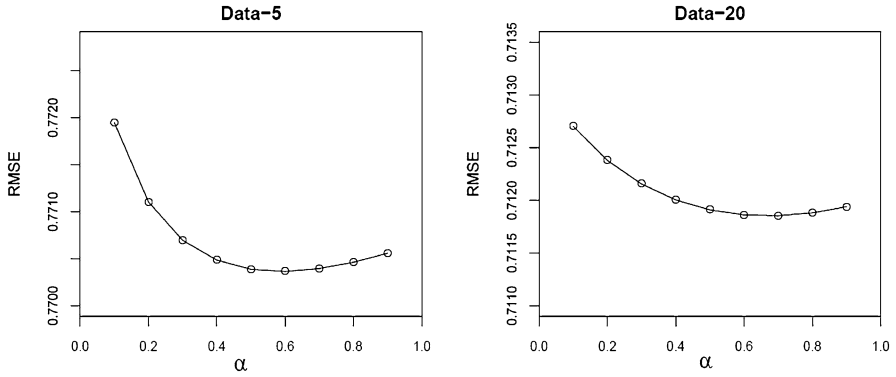

Fig. 5 RMSE of hybrid-UPCF with values of α

Table 9 RMSE with hybrid UPCF and other two approaches

Approaches	Data-5			Data-20		
	UPCF	UCF	Hybrid UPCF	UPCF	UCF	Hybrid UPCF
RMSE1	0.7786	0.7992	**0.7783** ($\alpha = 0.6$)	0.7139	0.7325	**0.7136** ($\alpha = 0.6$)
RMSE2	0.7741	0.7938	**0.7738** ($\alpha = 0.6$)	0.7138	0.7272	**0.7136** ($\alpha = 0.8$)
RMSE3	0.7671	0.7883	**0.7668** ($\alpha = 0.6$)	0.7101	0.7108	**0.7099** ($\alpha = 0.8$)
RMSE4	0.7663	0.7849	**0.7660** ($\alpha = 0.6$)	0.7088	0.7113	**0.7085** ($\alpha = 0.6$)
RMSE5	0.7674	0.7850	**0.7669** ($\alpha = 0.5$)	0.7137	0.7277	**0.7134** ($\alpha = 0.6$)
Average	0.7707	0.7902	**0.7704** ($\alpha = 0.6$)	0.7120	0.7219	**0.7118** ($\alpha = 0.7$)

Bold values indicate the best result

measures could help obtain better performance on RMSE. We also conducted the pairwise t-tests to verify the significance of the RMSE difference between UPCF and Hybrid-UPCF. The p values were 4.6874E−05 and 1.7094E−03 on Data-5 and Data-20 respectively, meaning that the improvements are significant.

5.5 Discussions on sparsity

As shown in Table 5, the data are quite sparse (which is also the case in the real world environment). Previous work pointed out that sparsity remarkably impacts the performance of the collaborative filtering process. To evaluate the performance of UPCF on dealing with sparsity, six subsets with different users' frequency were extracted from the typical sparse Data-5 dataset. Concretely, the users with 5–9 reviews were extracted to form the sparsest subset, the users with 10–14 reviews as the second subset, and so on, as shown in Fig. 6. Clearly, the 6th subset (i.e., the one containing the users with no less than 30 reviews) is deemed as the least sparse subset. The experimental results of UPCF and UCF on the six subsets (Fig. 6a) showed some merits of UPCF. First, with the decrease of sparsity, both UPCF and UCF showed a better and better performance, which is also consistent with the practical intuition. Second, UPCF performed better than UCF on all subsets. Third, the more sparse the data, the better the outperformance of UPCF over UCF. Moreover, Fig. 6b further illustrates that UPCF had more significant advantage on more sparse data over UCF. In a sentence, the proposed UPCF approach could effectively deal with the sparsity problem.

5.6 Discussions on understandability of recommendations

One of the key concerns in CF-based recommendation in real-world applications is the difficulty to interpret the recommendation results, i.e., the understandability of recommendations is usually low [41]. Some researchers have demonstrated that

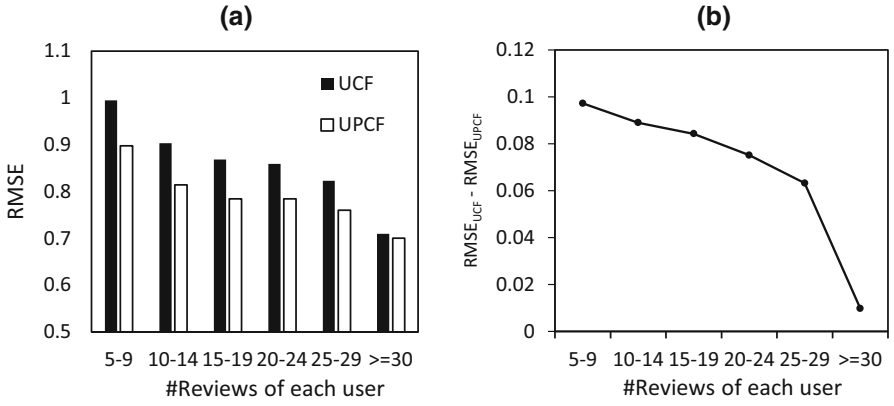

Fig. 6 **a** RMSE on subsets of users of Data-5. **b** $\mathrm{RMSE}_{\mathrm{UCF}} - \mathrm{RMSE}_{\mathrm{UPCF}}$

providing appropriate explanations from various aspects can improve user acceptance of the recommended entities and user trust of the systems [5, 31]. However, it is usually hard to know how a user cognitively forms his/her opinions from many aspects into a single and simple overall rating. In addition, the traditional CF-based recommendation systems (especially those based on matrix factorization techniques) usually only attempt to estimate ratings in a latent factorization space, thus having it even more difficult to make the recommendations explainable, although the implanted approaches may obtain satisfactory rating prediction accuracies [41].

In contrast, the existence of textual user reviews, as exposited in previous sections, provides more fruitful and granular information on aspects to help understand a user's key focuses (i.e., aspect importance) and specific needs (i.e., aspect need). Through the user preferences extracted from a user's historical reviews, we are able to target the aspects he/she concerns, and take advantages of the ratings given by the users with similar aspect preferences to make better personalized recommendations. The preferences can further provide reasonable explanations to these results, hence increasing the understandability of recommendation results.

In this regard, we conducted an additional case study to show that our proposed UPCF can generate personalized and explainable results in accordance with user preferences. For the same reviews dataset, we extracted a particular user whose preference to aspect *environment* is 0.91, and to each of the other nine aspects are 0.01, derived after conducting user preference inference. It means that the user cares more on the *environment* of restaurants and has a high need for this aspect. Thus, the top-5 restaurants recommended by UPCF are listed on the left in Table 10, while the restaurants with top-5 highest average overall ratings (i.e., without considering the user's specific preferences) are listed on the right in Table 10 for comparison. The overall rating is the average value of overall ratings that all users assign to the restaurant. Ratings on *environment* denote the average value of all sentiment scores to the aspect *environment* obtained from the reviews on the restaurant, while the frequency of *environment* denotes the proportion of reviews mentioning the aspect

Table 10 Top-5 UPCF recommendations and top-5 overall rating recommendations

Top-5 recommended by UPCF				Top-5 in dataset			
ID	Overall rating	Rating on *environment*	Frequency of *environment* (%)	ID	Overall rating	Rating on *environment*	Frequency of *environment* (%)
2139	4.75	4.54	62.50	2139	4.75	4.54	62.50
1951	4.48	4.63	55.32	1846	4.53	3.98	28.13
3085	4.49	4.11	23.40	2340	4.52	4.42	19.00
1502	4.41	4.28	55.88	2435	4.52	4.17	44.33
984	4.45	4.52	32.65	366	4.50	3.97	24.07
Average	4.52	**4.42**	**45.95**	Average	**4.56**	4.22	35.61

Bold values indicate the best result

environment in all the reviews on the restaurant. Though the restaurants recommended by UPCF have slightly lower overall ratings, they have higher and more ratings on the aspect *environment*, which are more appreciated for the user. This reveals that the UPCF could provide personalized recommendation according to a particular user's specific preferences. Meanwhile, the extracted user preferences would help us know how a user forms its overall ratings from multiple aspects, i.e., showing better understandability, which is desirable in consumers' decision making.

6 Conclusions

This paper has proposed a user-preference-based CF (UPCF) recommendation approach that incorporates the aspect-level information reflecting user preferences. Two measures for aspect preference evaluation have been introduced, namely *aspect importance* and *aspect need*, to reflect the aspect relationship to overall rating and the opinions' differences to aspects, respectively. UPCF utilizes the aspect preferences to identify user similarities, which is then incorporated into collaborative filtering. Unlike the other approaches only considering the overall ratings on common entities between users, which is perplexed by frequently-observed sparsity problem, the aspect preferences in UPCF are aggregated from all the reviews of a user, facilitating the calculation of the similarities between any pairs of users no matter how many entities they commonly rate, which can alleviate the sparsity problem to a certain extent. The experiments conducted on a collection of real review data from a large Chinese online platform have showed that UPCF significantly outperformed other approaches, achieved better performance in dealing with sparsity, and improved the understandability of the personalized recommendation results.

Furthermore, some managerial implications could also be concluded. The proposed measures to evaluate user aspect preferences have revealed user behavior patterns in a rich and granular manner. The recommendation considering the user preferences can offer the results in which the users are more likely to be interested so as to better support their decision-making. Thus, it can bring more targeted customers to merchants and manufacturers, which is valuable in the cruel market competition. More importantly, the approach can solve the sparsity problem to a certain extent, showing great potential in the real-world e-commerce environment.

Future research centers on two directions. One is to conduct more data experiments with reviews on a large online shopping platform, e.g., Amazon.com, so as to fine-tune UPCF to conform to a wide application context. The other effort is to extend UPCF by incorporating social networking information of users to further improve recommendations.

Acknowledgments The work was partly supported by the National Natural Science Foundation of China (71110107027/71490724/71372044), the MOE Project of Key Research Institute of Humanities and Social Sciences at Universities of China (12JJD630001), and China Retail Research Center of Tsinghua University School of Economics and Management.

References

1. Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering, 17*(6), 734–749.
2. Archak, N., Ghose, A., & Ipeirotis, P. G. (2011). Deriving the pricing power of product features by mining consumer reviews. *Management Science, 57*(8), 1485–1509.
3. Chen, L., Chen, G., & Wang, F. (2015). Recommender systems based on user reviews: The state of the art. *User Modeling and User-Adapted Interaction, 2*(25), 99–154.
4. Chen, L., & Wang, F. (2013). Preference-based clustering reviews for augmenting E-commerce recommendation. *Knowledge-Based Systems, 50*, 44–59.
5. Cramer, H., Evers, V., Ramlal, S., Someren, M., Rutledge, L., Stash, N., et al. (2008). The effects of transparency on trust in and acceptance of a content-based art recommender. *User Modeling and User-Adapted Interaction, 18*(5), 455–496.
6. Decker, R., & Trusov, M. (2010). Estimating aggregate consumer preferences from online product reviews. *International Journal of Research in Marketing, 27*(4), 293–307.
7. Ding, X., Liu, B., & Yu, P. S. (2008). A holistic lexicon-based approach to opinion mining. In *Proceedings of the international conference on web search and data mining* (pp. 231–240), New York.
8. Fellbaum, C. (1998). *WordNet: An electronic lexical database*. Cambridge, MA: MIT.
9. Ganu, G., Kakodkar, Y., & Marian, A. (2013). Improving the quality of predictions using textual information in online user reviews. *Information Systems, 38*(1), 1–15.
10. Garcia Esparza, S., O'Mahony, M. P., & Smyth, B. (2011). Effective product recommendation using the real-time web. In *Proceedings of the 30th SGAI international conference on innovative techniques and applications of artificial intelligence* (pp. 5–18), Cambridge.
11. Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions On Information Systems, 22*(1), 5–53.
12. Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 168–177).
13. Jakob, N., & Weber, S. (2009). Beyond the stars: Exploiting free-text user reviews to improve the accuracy of movie recommendations. In *Proceedings of the 1st international CIKM workshop on topic-sentiment analysis for mass opinion* (pp. 57–64), Hong Kong.
14. Jannach, D., Karakaya, Z., & Gedikli, F. (2012). Accuracy improvements for multi-criteria recommender systems. In *Proceedings of the 13th ACM conference on electronic commerce* (pp. 674–689).
15. Koren, Y. (2008). Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 426–434), Las Vegas.
16. Li, F., Han, C., Huang, M., Zhu, X., Xia, Y. J., Zhang, S., et al. (2010). Structure-aware review mining and summarization. In *Proceedings of the 23rd international conference on computational linguistics*. (pp. 653–661), Beijing.
17. Liu, H., He, J., Wang, T., Song, W., & Du, X. (2013). Combining user preferences and user opinions for accurate recommendation. *Electronic Commerce Research and Applications, 12*(1), 14–23.
18. Liu, B., & Sun, Y. (2013). Survey of personalized recommendation based on society networks analysis. In *Proceedings of 2013 6th international conference on information management, innovation management and industrial engineering (ICIII)* (pp. 337–340).
19. Liu, B., & Zhang, L. (2012). A survey of opinion mining and sentiment analysis. In *Mining text data* (pp. 415–463). Berlin: Springer.
20. Marinho, L. B., Nanopoulos, A., Schmidt-Thieme, L., Jäschke, R., Hotho, A., Stumme, G. et al. (2011). Social tagging recommender systems. In *Recommender systems handbook* (pp. 83–95). Berlin: Springer.
21. McAuley, J., & Leskovec, J. (2013). Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on recommender systems* (pp. 165–172), Hong Kong.
22. Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs Up?: Sentiment classification using Machine learning techniques. In *Proceedings of the ACL-02 conference on empirical methods in natural language processing* (pp. 79–86), Stroudsburg, PA.

23. Pero, Š., & Horváth, T. (2013). *Opinion-driven matrix factorization for rating prediction*. Lecture notes in computer science (Vol. 7899, pp. 1–13). Berlin: Springer.
24. Qi, L., & Chen, L. (2011). Comparison of model-based learning methods for feature-level opinion mining. In *Proceedings of IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology* (pp. 265–273), Washington, DC.
25. Raghavan, S., Gunasekar, S., & Ghosh, J. (2012). Review quality aware collaborative filtering. In *Proceedings of the sixth ACM conference on recommender systems* (pp. 123–130), Dublin.
26. Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). *Collaborative filtering recommender systems* (pp. 291–324). Berlin: Springer.
27. Seroussi, Y., Bohnert, F., & Zukerman, I. (2011). Personalized rating prediction for new users using latent factor models. In *Proceedings of the 22nd ACM conference on hypertext and hypermedia* (pp. 47–56), New York.
28. Stede, M. T. M. T. (2011). Lexicon-based methods for sentiment analysis. *Computational Linguistics, 37*(2), 267–307.
29. Su, Q., Xu, X., Guo, H., Guo, Z., Wu, X., Zhang, X., et al. (2008). Hidden sentiment association in Chinese web opinion mining. In *Proceedings of the 17th international conference on World Wide Web* (pp. 959–968), New York.
30. Terzi, M., Ferrario, M., & Whittle, J. (2011). Free text in user reviews: Their role in recommender systems. In *Proceedings of the 3rd ACM RecSys'10 workshop on recommender systems and the social web* (pp. 45–48), Chicago.
31. Tintarev, N., & Masthoff, J. (2007). A survey of explanations in recommender systems. In *Workshop at the IEEE international conference on data engineering* (pp. 801–810).
32. Titov, I., & McDonald, R.. (2008). Modeling online reviews with multi-grain topic models. In *Proceedings of international conference on World Wide Web*, Beijing.
33. Wang, H., Lu, Y., & Zhai, C. (2010). Latent aspect rating analysis on review text data: A rating regression approach. In *Proceedings of ACM SIGKDD international conference on knowledge discovery & data mining*, Washington, DC.
34. Wang, H., & Luo, N. (2014). Collaborative filtering enhanced by user free-text reviews topic modelling. In *Proceedings of 2014 international conference on information and communications technologies* (pp. 1–5).
35. Wu, Y., Zhang, Q., Huang, X., & Wu, L. (2009). Phrase dependency parsing for opinion mining. In *Proceedings of the 2009 conference on empirical methods in natural language processing* (Vol. 3, pp. 1533–1541).
36. Yang, Y., & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *Proceedings of the fourteenth international conference on machine learning* (pp. 412–420).
37. Yang, X., Steck, H., Guo, Y., & Liu, Y. (2012). On Top-K recommendation using social networks. In *Proceedings of the sixth ACM conference on Recommender systems* (pp. 67–74).
38. Yu, J., Zha, Z., Wang, M., & Chua, T. (2011). Aspect ranking: Identifying important product aspects from online consumer reviews. In *Computational linguistics* (pp. 1496–1505).
39. Zha, Z., Yu, J., & Tang, J. (2014). Product aspect ranking and its applications. *IEEE Transactions on Knowledge and Data Engineering, 26*(5), 1211–1224.
40. Zhai, Z., Liu, B., Xu, H., & Jia, P. (2011). Clustering product features for opinion mining. In *Proceedings of the 4th international conference on web search and data mining* (pp. 347–354), Hong Kong.
41. Zhang, Y. (2015). Incorporating phrase-level sentiment analysis on textual reviews for personalized recommendation. In *Proceedings of the eighth ACM international conference on web search and data mining* (pp. 435–440), New York.
42. Zhang, W., Xu, H., & Wan, W. (2012). Weakness finder: Find product weakness from Chinese reviews by using aspects based sentiment analysis. *Expert Systems with Applications, 39*, 10283–10291.

Yue Ma is a PhD candidate at the School of Economics and Management, Tsinghua University, Beijing China. Her research interests include recommender systems, and business intelligence.

Guoqing Chen received his PhD from the Catholic University of Leuven (K.U. Leuven, Belgium) and now is EMC Chair Professor of Information Systems at the School of Economics and Management, Tsinghua University, Beijing China. His research interests include electronic commerce, business analytics, decision support and soft computing.

Qiang Wei received his PhD from Tsinghua University and now is Associate Professor of Information Systems at the School of Economics and Management, Tsinghua University, Beijing China. His research interests include information systems management, social networks, business intelligence, and online consumer behavior.