Personalised ranking online reviews based on user individual preferences

Wei Song, Shiwei Zhang, Lizhen Liu* and Hanshi Wang

Information and Engineering College, Capital Normal University, No. 56, West Third Ring Road, Haidian District, Beijing, 100048, China
Email: wsong@cnu.edu.cn
Email: 2141002047@cnu.edu.cn
Email: lzli@cnu.edu.cn
Email: hswang@cnu.edu.cn
*Corresponding author

Abstract: With the development of e-commerce sites, online reviews have become important data resources for e-customers. Nowadays, there have been many literatures on the category of reviews category or ranking for public. However, they only satisfy common preferences, and ignore personalised preferences of individual users. In view of this phenomenon, this paper is trying to put forward a ranking method for individual preferences. It begins with collecting the rules of user preferences by showing reviews to them to let them mark the reviews they like. Then it combines the common rules with user personalised rules to get the range of features. Finally, after calculating the optimal solution of features, the paper strives to structure a ranking model to rank reviews with the set of optimal solution.

Keywords: attribute word; user preference rule; hill climbing algorithm; ranking.

Reference to this paper should be made as follows: Song, W., Zhang, S., Liu, L. and Wang, H. (2018) ‘Personalised ranking online reviews based on user individual preferences’, Int. J. Reasoning-based Intelligent Systems, Vol. 10, No. 1, pp.32–42.

Biographical notes: Wei Song received his PhD in Computer Application Technology in Harbin Institute of Technology in March 2013. Currently, he is a Lecturer in the Capital Normal University. His research interests are information retrieval, natural language processing and social computing. He has published a number of top conference papers and journal papers.

Shiwei Zhang is a graduate student in Capital Normal University. Her research interests are natural language processing and data mining. She has a paper which was accepted by a journal of Taiwan, but not yet published.

Lizhen Liu is a Supervisor of graduate students. She is the Head of Intelligent Science and Technology and research on data mining and intelligent information retrieval, text sentiment analysis, natural language processing and knowledge engineering, etc. She has been engaged in this field, knowledge and information systems, the famous international journals (KJS), Journal of Computational Biology SCI/EI, as well as the Domestic and Foreign Retrieval Journals and conference and has published more than 100 papers.

Hanshi Wang was a Supervisor of graduate students in the Capital Normal University. His research interests are natural language processing and data mining. He has published a number of top conference papers and journal papers.

1 Introduction

Since an increasingly number of consumers are inclined to give scores and evaluations to explain how they feel when they use certain products or experience services after making a deal on e-commerce websites, the evaluations of goods online have become one of the most important factors considered when people are shopping online. However, with rapid development of online shopping, the phenomenon of overload information becomes serious (Zhiyu, 2013). Online product reviews are diverse and numerous, and thus make consumers more confused rather than clearer about whether they should buy a product or not. Meanwhile, it is difficult for consumers to pick up useful information from such a great deal of it. Many B2C e-commerce websites like Amazon provide an effective evaluation mechanism for online reviews of products, which ranks goods by the percentage of useful evaluations among all evaluations. The more votes, the better ranking. Nevertheless, it is hard to
show online reviews of high quality for consumers accurately and timely if we use such an evaluation mechanism which relies largely on artificial judgments. It takes a long time for the online reviews that were published recently to accumulate the votes, which in turn, will lead to the submergence of these high-quality evaluations. And that's why we need to build up a mechanism to evaluate effective evaluation functions. We need to deal with a large amount of data, and pick out useful evaluations, in order to help consumers make wise decisions.

Nowadays, most evaluation works are regarded as classification which, as the features of evaluations require, extracts useful information, builds up classification models by machine learning method, and reviews the value of evaluations. Based on it, some research is refined to a deeper level and ranks the evaluations from high to low. For example, Kim et al. (2006) considers the task of automatically assessing the helpfulness of reviews. Experiments using SVM regression on a variety of features over http:www.Amazon.com product reviews have shown promising results, with rank correlations of up to 0.66. Also, they found that the most useful features include the length of the reviews, the unigrams, as well as the product rating. Some others study the classifications by extracting features which can improve accuracy of evaluations. For instance, according to the feature of language, Krishnamoorthy (2015) analyses the utility prediction of the semantic feature for evaluations. Besides, most literatures which conduct analysis of evaluation validity and sort evaluations are aimed at the universal acceptance of consumers instead of personal interests. However, every consumer has his/her own preference, and this is what this paper is dealing with and focusing on.

First of all, this paper divides features into inherent characteristics and characteristics of areas. Secondly, rules are collected through providing reviews for users to let them label which one they like. Meanwhile, rules are divided into three layers: common rules, domain rules and user preference rules.

The paper builds rule sets to get the range of each feature, and then ranks reviews through the optimal value of features.

The rest of the paper is organized as follows. Section 2 shows the related work. In Section 3, the method of extracting product attributes is introduced. In Section 4, the labels given by users are made full use of, in order to get both the common rules about the features comparison and the rules of user preferences in the distribution of each feature. Then, the rule sets are employed to rank reviews. In Section 5, experiments and the verification are conducted. At last, the conclusion will be drawn in Section 6.

2 Related work

The rocketing increase of diverse online evaluations and the amount of data endows the research on the utility of evaluations with significant meaning. For some research both at home and abroad, the analysis of utility evaluation is transformed as a task of classification or rank. It extracts multifaceted features related to evaluations, and builds up a classification forecasting and ranking model of utility evaluation through the method of machine learning. For example, Kim et al. (2006) used regression method of SVM to measure the utility of evaluation automatically from five aspects. That is, structure, morphology, syntax, semantic and metadata. According to the experiment, the key points of judging the utility of evaluations are the length of the evaluation, unigram and the grade of products. Liu et al. (2007) noticed that vote accumulation could lead to evaluation bias. Consequently, Liu did the research by using human-annotated data, and described the quality of evaluation quality from three aspects-information content, subjectivity and readability. Ngo-Ye and Sinha (2014) used the text regression model with the combination of bag-of-word model with RMF to predict evaluation quality. Liu et al. (2013) conceived this issue from the perspective of a product designer, assumed that the average utility grade which was evaluated by designers was a gold standard, and proposed four categories of features reflecting designers’ concerns in judging the usefulness and helpfulness of reviews. Yu et al. (2010) found that writing style, as reflected in part-of-speech tags, was effective in predicting the quality of movie reviews when using support vector regression. Chen and Tseng (2011) put forward that high-quality reviews were those with in-depth comments on products’ features, and thus were subjective. Cambria et al. (2013) shed light on new avenues of sentiment analysis and opinion mining by weighing on the notion of concept level analysis in comparison to topic level analysis. SumView (Wang et al., 2013a) refers to a newly-explored semi-supervised web application which is capable of review crawling along with automatic extraction of products’ features. Users can query features according to their personal interest in a process using a sentence selection along with the proposed feature-based weighted nonnegative matrix factorisation (NNMF) algorithm. Wang et al. (2013b) proposed a novel algorithm to identify experts in online knowledge communities. Ngo-Ye and Sinha (2012) found the dimension reduction techniques could enhance the performance of text regression performance in judging and predicting the utility of reviews. Liu and Park (2015) presented a helpfulness prediction model for websites selling travel-related products. They employed a combination of reviewer and review characteristics to predict the utility and helpfulness. Also, in the process, the identity, reputation, expertise and valence of reviewers are all used to build a text regression model in the prediction Danescu-Niculescu-Mizil et al. (2009) confirmed that the perceived helpfulness of reviews depends not only on its content, but also on the relation of its score to other scores. Chen and Tseng (2011) employed an effective information quality framework to extract representative features of reviews, and high-quality reviews were found to be subjective and were able to provide in-depth comments on a number of features of certain products.
Previous studies only satisfy the common preference, but the niche preferences of certain users are not taken into account. So this paper puts forward a new method to rank online reviews for independent individual’s preference about the product attributes. The paper rank reviews according the user individual preference for every consumer.

These papers tend to make classification and sort goods for the majority of customers and on an average level, but they failed to take into account the differences between each user’s preferences. Hence, these methods applied in many previous studies and research cannot satisfy everyone. Being aware of this fact and with the aim of recommending reviews possessing user-concerned attribute and welcome by all users, this paper puts forward an idea to rank for every customer. And to be specific, this main body of paper includes four parts:

1. It extracts the inherent features and domain features, and vectors reviews based on features.
2. It tries to obtain rules from three aspects-common, user preference and domain, while making use of ontology to structure rule set.
3. According to the users marked reviews, the Bayesian models are built, and the climbing hill algorithm is used to converge each feature.
4. The final model for ranking reviews is structured.

3 Establish the feature vector of reviews

3.1 Classification of features

Features are categorised into two sorts: inherent features and domain features. Inherent features are the inherent attributes of reviews; it will not change with the alteration of different fields. Since user preferences can reflect the different users’ habit of reading, the inherent characteristics of reviews are added to users’ choice in dealing with reviews. This paper regards and uses the review’s length, votes, timeliness, readability and star as the inherent features:

- Review’s length: The number of words in a review and the helpfulness of a review are positive correlation (Mudambi and Schuff, 2010) The description of a product will be detailed if there is a long review. So it can both meet customers’ demands for information and reduce the uncertainty of purchase decision to a large extent.
- Review’s votes: The affirmative vote of a review indicates the admissive degree of the public to the review. The more affirmative votes, the more approvers to this review.
- Review’s timeliness: It is the (time) difference between the present time and the releasing time of a review: 
  \[ \text{review time} = \log(\text{actual time} - \text{review data}) \quad (1) \]
  The smaller the value of review age is, the more effective the review can illustrate the real state of a product. For example, there is a phone release in 2013, whose internal parts may change with the passage of time. So, the new reviews appear to be more attractive than the earlier one.
- Review’s readability: It refers to the fluency of a review. When the expression of a review is obscure and the order is reversed, the review will not be catchy. And this impacts the users’ patience to read the reviews. Besides, the higher fluency, the more likely the user will accept the review. This paper uses two-gram to calculate the fluency of a sentence, and the method to calculate the fluency is (2):

  \[ \text{flu}(r) = \frac{\sum \ln \left( p(w_i | w_{i-2}w_{i-1}) \right)}{n} \quad (2) \]
  The ‘n’ is the number of words in a sentence, while \( \text{flu}(r) \) is the result of fluency calculation.
- Review’s star: When customers publish reviews, they will give a rate to the product, which is deemed as a reflection of the reviewers’ attitudes. Specifically, the rating is divided into five levels of stars, that is, 1, 2, 3, 4, 5. While one star shows the disappointment of customers, five stars demonstrate the satisfaction to a product. In this paper, we select the reviews for customers according to the users’ preferences.

On the other hand, domain feature, which refers to the descriptive words on the attribute of a product, is aimed at excavating information from the contents of reviews. The more words on the features of a product in the content of a review, the higher the relevance of the content and the product is. When users read reviews, what they hope most is to find the product attribute they care about and make shopping decisions according to the description. The main thing in this paper is to analyse whether the descriptive words on the attribute of a product are factors that users concern with. So, the words on the product attribute are the paper’s domain feature.

3.2 Extracting descriptive words on the product attribute

Due to individual variations, consumers have different hobbies and concentrate on different things. At the same time, glancing over what they concentrate on by themselves wastes much time and energy, leaving customers unsatisfied. That is why extracting product vocabularies from product evaluations are of great importance. Take an evaluation from Amazon for example. “The phone’s voice quality is sharp and clear, but the battery only lasts one day. In addition, the large screen and reasonable button design makes it fashionable. But because of the high price, I’m not sure I will recommend it to others”. In this evaluation, we can know exactly that voice quality, battery, screen, button design and price are property words.

- Review’s description on the product attribute: The description of a product is the key information that customers care about. This paper regards and uses the review’s description to the product attribute. For example, there is a phone release in 2013, whose internal parts may change with the passage of time. So, the new reviews appear to be more attractive than the earlier one.
In recent years, some researchers have been researching about the extraction method of property words in English product evaluations, and most of them use automatic training method. Hu and Liu (2004) got property words of high frequency by associating rules at first. Then, the recall rate and accuracy rate were improved with pruning. At last, low frequency property words were found as supplement, and the final list of property words was obtained. Qiu et al. (2011) proposed a method which was based on a few seeds and the final list of property words was obtained. Qiu et al. (2011) proposed a method which was based on a few seeds and the final list of property words was obtained. Qiu et al. (2011) proposed a method which was based on a few seeds and the final list of property words was obtained. Qiu et al. (2011) proposed a method which was based on a few seeds and the final list of property words was obtained. Qiu et al. (2011) proposed a method which was based on a few seeds and the final list of property words was obtained. Qiu et al. (2011) proposed a method which was based on a few seeds and the final list of property words was obtained. Qiu et al. (2011) proposed a method which was based on a few seeds and the final list of property words was obtained. Qiu et al. (2011) proposed a method which was based on a few seeds and the final list of property words was obtained. Qiu et al. (2011) proposed a method which was based on a few seeds and the final list of property words was obtained.

Table 1 The rule list about extracting attribute words of product

<table>
<thead>
<tr>
<th>Data type</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amod</td>
<td>Attribute word, descriptor</td>
</tr>
<tr>
<td>NN</td>
<td>Attribute word, attribute word</td>
</tr>
<tr>
<td>Nsubj</td>
<td>Attribute word, descriptor</td>
</tr>
<tr>
<td>Dobj</td>
<td>Attribute word, verb</td>
</tr>
<tr>
<td>Advmod</td>
<td>Descriptor, modifier</td>
</tr>
<tr>
<td>Dobj + advmod</td>
<td>Attribute word + verb + descriptor, or SVD</td>
</tr>
<tr>
<td>Nsubj + advmod</td>
<td>Attribute word, verb, descriptor</td>
</tr>
<tr>
<td>Nsubj + acomp</td>
<td>Attribute word, verb, descriptor</td>
</tr>
<tr>
<td>Nsubj + dobj</td>
<td>Attribute word, verb, descriptor</td>
</tr>
</tbody>
</table>

We use the template to extend the attribute set with incremental iterative method. However, the attribute we extracted cannot reflect the product attribute from all aspects. So, we filter non-attribute words according to Hui and Guang (2014).

Although we got the attribute words, there were cases where two different words represented the same meaning. Consequently, there is a need to calculate the semantic similarity of attribute words. If the result is above the threshold we set, it means that the two words represent the same meaning.

With respect to similarity calculation, it is based on the WordNet which uses word sense similarity computing to calculate the similarity of two different words. Wei and Endong (2004) introduced a method which is based on the WordNet, extracting synonyms and adopting the VSM to calculate the similarity of words:

\[ \text{Similarity}(w_1, w_2) = \sum_{i=1}^{n} \max_{j=1}^{n} \left( \text{similarity}(SW_1, SW_2) \right) \]

\[ = \frac{\sum_{i=1}^{n} \max_{j=1}^{n} \left( \text{similarity}(SW_2, SW_1) \right)}{|SW_1| + |SW_2|} \]

| SW1 | The number of sentences that contain \( w_1 \). |
| SW2 | The number of sentences that contain \( w_2 \). |

3.3 Vectorising reviews

In this section, we mainly vectorise the reviews using inherent features and domain features. The inherent features in reviews’ vector are denoted as \( f_1, f_2, f_3, f_4, f_5 \). The domain features are denoted as \( f_6, \ldots, f_n \).

- We set a threshold value \( \tau_1 \) for the review’ length at the beginning. If the length is bigger than \( \tau_1, f_1 \) is labelled as 1, or else \( f_1 \) is labelled as 0. It shows as (4):

\[ l_{en} = \frac{\sum_{i=1}^{n} (\text{length}_i - \text{ave}(\text{length}))^2}{n} \]

The ‘len\(_{en}\)’ refers to the relative length, \( \text{length}_i \) is the length of review \( i \). \( \text{ave}(\text{length}) \) is the average length of all reviews.

\[ \text{ave}(\text{length}) = \frac{\sum_{i=1}^{n} \text{length}_i}{n} \]

\[ f_1 = \begin{cases} 1 & \text{if } \text{len} > \tau_1 \\ 0 & \text{else} \end{cases} \]

- The timeliness of the reviews is denoted as \( f_2 \). Likewise, we set a threshold value \( \tau_2 \) too. If the result of the review age is bigger than \( \tau_2 \), it is an indication that the review is new. It shows as (7):

\[ f_2 = \begin{cases} 1 & \text{if review age} > \tau_2 \\ 0 & \text{else} \end{cases} \]

- The review’s readability shows the fluency of review. The formula is shown as (8):

\[ f_3 = \begin{cases} 1 & \text{if flu(r)} > \tau_3 \\ 0 & \text{else} \end{cases} \]

The ‘\( \tau_3 \)’ is the threshold value for fluency.

- The votes of the review are reflected in \( f_4 \). If the review has votes, \( f_4 \) will be labelled as 1, or else as 0.

- The stars of the reviews suggest the rating. The feature of rating is denoted as \( f_5 \). If the star is five or four, \( f_5 \) will be labelled 1; if the star is one, two or three, \( f_5 \) will be labelled 0.

Then, it comes to the domain feature. If a review contains adjunct words, we say it includes the feature. And the location of domain feature is labelled as 1. However, sometimes the feature appears in the review with no adjunct words. In this case, the location is labelled as 0. If the review does not contain any feature, the location will be labelled as 0 too.

\( r \) can be represented as
Inherent features domain features
\( f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, \ldots, f_n \),
it is a row of multidimensional matrices.

4 Structure the rule set and ranking model

In this paper, we put forward to adopt three layers of rules to rank reviews. First of all, we defined the rules as follows:

- **Common rules**: rules extracted from the interests of different people in different domains according to different reviews. It contains both the rule of domain feature and the rule of inherent feature.

- **Domain rules**: rules extracted from the interests of different people in the same domain according to different reviews. The domain rules only include the rules of product attribute endowed by the feature of descriptive words.

- **User rules**: rules extracted from the interests that one people have in different domains according to different reviews.

**Figure 1** The relationship among the common rule, the domain rule and the user preference rule

4.1 Collecting rules

The collection of rules is another focus in this paper. We randomly draw several groups (one group has two reviews), and use the form of human-computer interaction to show reviews to the users and let them mark which one she (he) likes more than others. After that, we get a number of rules which highlight comparison of features, so that we can judge the users’ preferences. For example, two reviews about Kindle from Amazon are showed as below, with each one has its own respective features. The example well demonstrates the way to make comparisons between different features.

Through extracting descriptive words on attribute, it can be seen that, there are three features in the first review. That is, ‘light, Bookerly font and storage’. The features in the second review are ‘turn and backlit display’. Since ‘backlit display’ and ‘light’ indicate the same feature, the two reviews both describe the display. If the user likes the first review more than the second, we can get the rule of the user preference about domain features:

light + Bookerly font + storage > backlit display + ‘turn’

Because the ‘light’ is equal to ‘backlit display’, we eliminate the two from the left and right side respectively. The result is:

Bookerly font + storage > ‘turn’ the page.

The reviews have been represented by vectors, so the rule can be expressed as \( f_i + f_j > f_k \). We put the rule into the user rule set. Now we know the programmatic way to collect rules, then what is the specific method to get the rule from the vectored reviews which not only have the domain feature but also have the inherent feature? To demonstrate this, here is an example.

\[
\begin{align*}
&\text{If } r_i \neq r_j \\
&f_i = (1, 0, 1, 0, 0, 1, 0, 1, 0, 1, \ldots) \\
&f_j = (0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, \ldots) \\
&\text{The result of calculating the ‘xor’ between } f_i \text{ and } f_j \text{ is } f_{ij} \\
&= (1, 0, 0, 1, 0, 0, 1, 0, 0, 1, \ldots) \\
&\text{The ‘xor’ is used again between } f_{ij} \text{ with } f_{io} \text{ with } f_{jo} \\
&\text{The location of } f \text{ that the feature value is 0 are } 1, \ldots; 4, 7, \ldots; 1, 4, 7 \ldots \text{ are corresponding to the } f_1, \ldots; f_4, f_7, \ldots \\
&\text{Add the } f_i \text{ that the value is 0 in } f_{ij} \text{ and } f_{io}, \text{ that is } \\
&\quad f_1 + \cdots; \\
&\text{Add the } f_j \text{ that the value is 0 in } f_{ij} \text{ and } f_{io}, \text{ that is, } \\
&\quad f_4 + f_7 + \cdots; \\
&\text{According to the user preference to get the rule: } \\
&\quad \text{if user like } f_{io}, \text{ then } f_1 + \cdots > f_4 + f_7 + \cdots; \\
&\quad \text{else } f_1 + \cdots < f_4 + f_7 + \cdots;
\end{align*}
\]

The ‘xor’ of two reviews is calculated mainly to get the location that the result is one. And use the ‘xor’ between the locations we get by ‘xor’ with the two reviews. The location that the result is one is what we need.
Every user will get several groups of reviews to label when logs in the shopping website. Then we will get user preference rule. After the multiple individual being labelled to the data, we draw out the common rules. We can choose the reviews for users to label based on the common rules if the common rule is large enough. Because the common rules and the domain rules are the mutual rules for different user preferences, we can extract several sets of reviews containing obvious characteristics and differences for users to label again. The user rules we get will not be able to work unless the domain rule is lack of rules, which means the users did not buy the product in the same domain. We extract the reviews which contain those features to let users mark to get rules. And the process is shown in Figure 3.

4.2 Extension of rules

We get the user preference rule; however, there is a condition that two or a plurality of rules can deduce a new rule. In the example of \( f_i > f_j \); \( f_j > f_k \); \( f_i > f_k \) or \( f_i > f_j + f_k \), we put the new rule and the primary rules into the rule set. And we need to introduce a new concept called support degree. It is defined as the probability to support the existence of the rule. After the user labelling the preference, we count the number of forward rules as well as the sum number of forward and backward rules. Since the forward rules and the backward rules are relative, we give the minimum support degree among condition rules to the inference rule. The computational formulas are shown as follows:

\[
Sup(\text{rule} - f) = \frac{n_{rule-f}}{N_{total}}
\]

\[
N_{total} = n_{rule-f} + n_{rule-b}
\]

\[
Sup(\text{infer} - \text{rule}) = \min\{sup(\text{conditional rule})\}
\]

4.3 Ensuring the range of features

In this section, our aim is to get the range of features’ distribution in accordance with the user rule and the support degree of the rule. And the distribution of features is \((0, 1)\). Firstly, we use the reviews that users marked. If we get the rule between two features, we divide the \((0, 1)\) into two parts, that is, \((0, 0.5)\), \((0.51, 1)\), and correspond the features to the corresponding scope. If we get the rule among three features, we divide the \((0, 1)\) into \((0, 0.33)\), \((0.34, 0.67)\),

Figure 2  Example of Amazon reviews (see online version for colours)
(0.68, 1). Likewise, when we find out the rule among several features, we divide (0, 1) into the number of features that have contact relationship.

If there is a common rule, we use the support degree about the feature to calculate its range, and count the number of both its forward and backward rules. In doing this, 0.5 is set as the starting point. The upper limit value in the range is using 0.5 to add the average of forward rule’s support degree about the feature, while the lower limit is using 0.5 to subtract the average of backward rule’s support degree about the feature. For example, the support degree of forward rule for \( f_i \) is \( \sum_{i=1}^{n} \sup(\text{rule} - f_i) \). The support degree of backward for \( f_i \) is \( \sum_{i=1}^{m} \sup(\text{rule} - f_i) \), then the range about \( f_i \) is:

\[
\begin{align*}
0.5 \times \frac{\sum_{i=1}^{n} \sup(\text{rule} - f_i)}{n} & \times \text{step}, \\
0.5 \times \frac{\sum_{i=1}^{m} \sup(\text{rule} - f_i)}{m} & \times \text{step}.
\end{align*}
\]

The step is adjusted automatically by programs to ensure that the range does not run out of (0, 1).

### 4.4 Obtaining the optimal solution of parameters

We have obtained the range of features, then how to get the optimal solution from the range? According to the heuristic rule, we randomly select a parameter, and keep the parameter fixed. This is done in the constraint condition to make the parameter reach its optimum. At the beginning, we got five groups reviews that were labelled, then we use the marked reviews as the standard to converge \( \theta \) which is represented as \( \{\theta_1, \theta_2, \ldots, \theta_n\} \). Each group has two reviews for users to mark. When they label the review he or she likes as one, the other will be labelled as 0 consequently. So the result of marking or labelling by users is, to some degree, classification. The standard of convergence is:

\[
p(C_i = 1 | X_1, \theta) > p(C_i = 1 | X_2, \theta)
\]

(12)\]

The \( X_i \) is the vector of review 1. The \( X_i \) is the vector of review 2. \( C \) refers to the labels given by customers. And \( p(C_i = 1 | X_1, \theta) \) shows the probability of customers’ interests in the condition that the review is \( X_i \) and \( \theta \) is used. The fact the users prefer the review 1 to review 2 is shown in (12).

Formula (13) can be deduced easily from (12):

\[
\begin{align*}
p(X_1 | C_1 = 1, \theta) p(C_1 = 1) > p(X_2 | C_2 = 1, \theta) p(C_2 = 1) \\
p(X_1 | C_1 = 1, \theta) p(C_1 = 1) - p(X_2 | C_2 = 1, \theta) p(C_2 = 1) > 0
\end{align*}
\]

(13)

(14)

Because the feature is 1 or 0 in reviews, so distribution \( \theta \) of each feature conforms to the Bernoulli distribution:

\[
p(C | X; \theta) = h_0(x)^C h_0(x)^{1-C}
\]

(15)

The features are mutually independent, so:

\[
p(x_1, x_2, \ldots, x_n | C_1 = 1, \theta) = \prod_{i=1}^{n} p(x_i | C_1 = 1, \theta)
\]

(17)

\[
\prod_{i=1}^{n} p(x_i | C_1 = 1, \theta) = \prod_{i=1}^{n} \theta_{i, x_i - 0}
\]

(18)

After getting the range of features, the hill climbing algorithm is applied to get the optimal solution of \( \theta \). To begin with, the range of a certain feature is divided into two parts equally, and an initial value is given respectively and randomly from the two parts to the common feature’s \( \theta \) which was denoted as \( \theta_1 \). Next, we give the other features’ \( \theta \) a random value. Then, all features’ values are put randomly into the formula (13)–(18) to observe which parts can cause a good effect. After that, we divide the parts into two parts equally, too, and select a value randomly for \( \theta_1 \) to get the best range. This method is used to converge the range of \( \theta_1 \) until we get the optimal solution of \( \theta_1 \); at the same time, the other features’ value will not change until we get the optimal solution of \( \theta_1 \) about the certain features. After finding out the optimal solution of \( \theta_1 \), we use the same way to get the optimal solution of \( \theta_2 \). The parameter values of other features are deduced by analogy and are dealt with in the same way. In the end, a set of optimal solutions about features which belong to the reviews that users marked is found out.

### 4.5 Ranking reviews

The collection of all reviews of a product can be represented as: \( R = \{r_1, r_2, \ldots\} \), \( R_i \). The \( x_i \) is the value of whether \( f_i \) appears in the \( r_i \). We have got the \( \theta ' \) of each feature, \( \theta = \{\theta_1, \theta_2, \ldots, \theta_n\} \). If the \( \theta ' \) corresponds to the features in the marked reviews, the \( \theta ' \) is the value that we got from the Section 4.5. And if the \( \theta ' \) corresponds to the features that are not in the marked reviews, we combine the range of features which we obtained from the common rule’s support with the common rule to find out solutions.

So, the score of the review is:

\[
\text{Score}_i = \sum_{j=1}^{k} w_j \theta_j * x_j + \sum_{j=q}^{n-k} w_j \theta_j * x_j
\]

(19)

The \( \theta ' \) is the number of features in the user rule. The \( w_j \) is the weight of features in the user rule, and the \( \theta_j \) is the weight of features that appear in the common rule instead of the user rule. The ‘score’ is the result of calculation for reviews. We use the ‘score’ that the reviews get to rank reviews and show it for users.

### 5 Experiments

#### 5.1 Dataset

In the experimental, we use 1,256 Kindle reviews, 5,791 clothing reviews, 2,537 reviews of shoes, 3,189 reviews of phones, and 584 reviews of books. We obtained all the data
In this section, two experiments are conducted. One is to ensure that the method is general: customer, and observes the accuracy rate in two domains to rank reviews for each platform. The experiment combines user preferences with the reviews in different domains to rank reviews for each platform. In doing this, we can get the domain rule. Then, based on the domain rule and the user rule, we ranked the reviews and sent the result and initial data in Amazon to 300 people to get their feedback. In addition, we randomly selected three volunteers so that we could get three ranking results. We showed volunteers four ranking results including the initial review as comparison without telling them which came from him or her. Then we let them make comparison and got their feedback. We did this experiment for ten times, with three people as a group were chosen each time and no reduplicative subjects involved in repeated experiments.

The other experiment is based on the first one. Its aim is to find five people to collect their preference rules in turn, and combine the previous users’ preferences with preference rules of their own to rank the reviews. This experiment includes two rounds. The first round is from user 1 to user 5, and we regard the user 1 as the user 6 to start the second round. After that, we show two different results of ranking to the same user to observe his or her feedback. This is effective in measuring the ranking effect with the increase of user preference information. In the experiment, we also compare the ranking result with the previous one and the original data, which is deemed as the control group, so as to let the user express his feeling about the reviews through marking.

The feedback is divided into 4 levels: Not satisfied at all; general; satisfied; quite satisfied. Each user can choose a level to show his or her feelings to the ranking of reviews. If the user’s feedback is ‘satisfied’ or ‘quite satisfied’, we say that the ranking review for the user is accurate.

However, there is a question about how to select reviews for user to mark by the way of active learning. More importantly, the means of selecting appropriate review group is of great significance. Consequently, two methods are employed to select reviews:

- Batch choice: select five groups reviews with different features randomly and one-off for users to mark. This process can contain a few rounds.
- Incremental choice: select one pair or a few of reviews for users to mark, and according to the result of marking, select reviews further that have pertinence.

We can use:

- a to get rules based on the big data, however, in our experiment, we do not have the big data, so we use (2) to get more accurate information more strictly.

5.2 Result

First of all, we collect the domain features about kindle, which include ads, background, backlight, battery, buttons, charge, clarity, contrast, cover, adapter, display, download, dpi, font, keyboard, library, lightness, page turn, pixel, price, quality, readability, screen, service, settings, size, version = 1, weight = 1, and Wi-Fi. So, the $f_{0-53}$ are domain features. Secondly, we get the information of the inherent features preferred to and cared about by the three people.

Also, we use Incremental choice to select five groups of reviews from all the reviews of kindle to the volunteers, and get the range of features according to the common rule. The range of each feature is shown as Table 2.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>price</td>
<td>1-5</td>
</tr>
<tr>
<td>quality</td>
<td>1-5</td>
</tr>
<tr>
<td>readability</td>
<td>1-5</td>
</tr>
<tr>
<td>screen</td>
<td>1-5</td>
</tr>
<tr>
<td>service</td>
<td>1-5</td>
</tr>
<tr>
<td>settings</td>
<td>1-5</td>
</tr>
<tr>
<td>size</td>
<td>1-5</td>
</tr>
<tr>
<td>version</td>
<td>1-5</td>
</tr>
<tr>
<td>weight</td>
<td>1-5</td>
</tr>
<tr>
<td>Wi-Fi</td>
<td>1-5</td>
</tr>
</tbody>
</table>

As we can see in Table 4, The rules revealed by users’ marked reviews vary, so the distributions of each feature are different, leading to the variation in the scores of reviews. The reason why the result of A can satisfy the user A and B instead of C is that, the user preference of A, B, C we get from the marked reviews are different from each other. The ranking reviews for A is done according to the rule of A, and the user B’s preference is similar to A. As a consequence, the features in reviews are in conformity with the preferences of A and B but not those of C. The user may feel nothing special to the other two ranking reviews if the person and the other two users’ preferences do not have big difference with public, such as the user A’s attitude to the result of C and the control group D. The user may not satisfied with the other two reviews at all if the user’s preference is different from the other two user’s preferences. And this is the case with the user C’s attitude to the results of A and B. The user B feels nothing special to the result of C, but is not satisfied with D at all, demonstrating that the preference of user A is similar to B, but there remains big differences between A and C.
Table 2  The range and the optimal value of features according to the common rule

<table>
<thead>
<tr>
<th>Features</th>
<th>Range values</th>
<th>Parameter values</th>
<th>Features</th>
<th>Range values</th>
<th>Parameter values</th>
<th>Features</th>
<th>Range values</th>
<th>Parameter values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Review's length</td>
<td>(0.43, 0.89)</td>
<td>0.67</td>
<td>Library</td>
<td>(0.23, 0.53)</td>
<td>0.28</td>
<td>Buttons</td>
<td>(0.43, 0.85)</td>
<td>0.77</td>
</tr>
<tr>
<td>Review’s timeliness</td>
<td>(0.45, 0.93)</td>
<td>0.75</td>
<td>Page turn</td>
<td>(0.44, 0.70)</td>
<td>0.68</td>
<td>Backlight</td>
<td>(0.40, 0.77)</td>
<td>0.76</td>
</tr>
<tr>
<td>Review’s readability</td>
<td>(0.5, 0.97)</td>
<td>0.95</td>
<td>Keyboard</td>
<td>(0.36, 0.53)</td>
<td>0.37</td>
<td>Service</td>
<td>(0.44, 0.64)</td>
<td>0.60</td>
</tr>
<tr>
<td>Review’s vote</td>
<td>(0.21, 0.52)</td>
<td>0.24</td>
<td>Cover</td>
<td>(0.32, 0.66)</td>
<td>0.48</td>
<td>Clarity</td>
<td>(0.43, 0.55)</td>
<td>0.39</td>
</tr>
<tr>
<td>Review’s star</td>
<td>(0.35, 0.96)</td>
<td>0.62</td>
<td>Display</td>
<td>(0.43, 0.67)</td>
<td>0.66</td>
<td>Ads</td>
<td>(0.42, 0.69)</td>
<td>0.46</td>
</tr>
<tr>
<td>Screen</td>
<td>(0.11, 0.56)</td>
<td>0.17</td>
<td>Adapter</td>
<td>(0.49, 0.51)</td>
<td>0.5</td>
<td>Lightness</td>
<td>(0.42, 0.73)</td>
<td>0.71</td>
</tr>
<tr>
<td>Size</td>
<td>(0.24, 0.74)</td>
<td>0.33</td>
<td>Price</td>
<td>(0.45, 0.72)</td>
<td>0.55</td>
<td>Download</td>
<td>(0.40, 0.64)</td>
<td>0.40</td>
</tr>
<tr>
<td>Version</td>
<td>(0.15, 0.54)</td>
<td>0.41</td>
<td>Background</td>
<td>(0.40, 0.76)</td>
<td>0.69</td>
<td>Pixel</td>
<td>(0.47, 0.51)</td>
<td>0.48</td>
</tr>
<tr>
<td>Weight</td>
<td>(0.33, 0.65)</td>
<td>0.51</td>
<td>Contrast</td>
<td>(0.45, 0.55)</td>
<td>0.52</td>
<td>Readability</td>
<td>(0.48, 0.98)</td>
<td>0.91</td>
</tr>
<tr>
<td>Battery</td>
<td>(0.32, 0.82)</td>
<td>0.80</td>
<td>Quality</td>
<td>(0.47, 0.85)</td>
<td>0.83</td>
<td>Wi-fi</td>
<td>(0.44, 0.53)</td>
<td>0.52</td>
</tr>
<tr>
<td>Font</td>
<td>(0.37, 0.77)</td>
<td>0.59</td>
<td>Settings</td>
<td>(0.44, 0.53)</td>
<td>0.49</td>
<td>Dpi</td>
<td>(0.05, 0.50)</td>
<td>0.1</td>
</tr>
</tbody>
</table>

| Charge           | (0.48, 0.72) | 0.72             |                  |              |                  |                  |              |                  |

Table 3  The range and the optimal value of features according to the user preference rule

<table>
<thead>
<tr>
<th>User A’s preference</th>
<th>User A</th>
<th>User B</th>
<th>User C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range values</td>
<td>Parameter values</td>
<td>Range values</td>
<td>Parameter values</td>
</tr>
<tr>
<td>Screen</td>
<td>(0.68, 1)</td>
<td>0.90</td>
<td>Screen</td>
</tr>
<tr>
<td>Font</td>
<td>(0.34, 0.67)</td>
<td>0.40</td>
<td>Weight</td>
</tr>
<tr>
<td>Page turn</td>
<td>(0.34, 0.67)</td>
<td>0.63</td>
<td>Download</td>
</tr>
<tr>
<td>Download</td>
<td>(0.5, 0.48)</td>
<td>0.24</td>
<td>Size</td>
</tr>
<tr>
<td>Size</td>
<td>(0.33, 0.33)</td>
<td>0.24</td>
<td>Contrast</td>
</tr>
<tr>
<td>Cover</td>
<td>(0.51, 0.59)</td>
<td>0.59</td>
<td>Cover</td>
</tr>
<tr>
<td>Contrast</td>
<td>(0.34, 0.67)</td>
<td>0.67</td>
<td>Charge</td>
</tr>
<tr>
<td>Price</td>
<td>(0.68, 1)</td>
<td>0.36</td>
<td>Length</td>
</tr>
<tr>
<td>Charge</td>
<td>(0.68, 1)</td>
<td>0.85</td>
<td>Star</td>
</tr>
<tr>
<td>Backlight</td>
<td>(0.51, 1)</td>
<td>0.66</td>
<td>Battery</td>
</tr>
<tr>
<td>Battery</td>
<td>(0.68, 1)</td>
<td>0.83</td>
<td>Clarity</td>
</tr>
<tr>
<td>Library</td>
<td>(0.33, 0.30)</td>
<td>0.30</td>
<td>Ads</td>
</tr>
<tr>
<td>Service</td>
<td>(0.68, 1)</td>
<td>0.69</td>
<td>Pixel</td>
</tr>
<tr>
<td>Length</td>
<td>(0.5, 0.44)</td>
<td>0.44</td>
<td></td>
</tr>
</tbody>
</table>

Table 4  The feedback of user A, B, C for the ranking result and controls

<table>
<thead>
<tr>
<th>Feedback</th>
<th>User A</th>
<th>User B</th>
<th>User C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not satisfied at all</td>
<td>General</td>
<td>Satisfied</td>
<td>Quite satisfied</td>
</tr>
<tr>
<td>Result</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ranking of A</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Ranking of B</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Ranking of C</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>D¹ (control group)</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

Note: ¹is the control group, which is the original ranking by voting to the reviews on the plat of Amazon.
In the second experiment, we show the results of the two rounds to users. We let five users to mark which review is the most satisfying one to them. And their feedback is as follows in Table 5.

It can be seen from Table 5 that five users give their scores to the three groups of reviews respectively. From Table 6, it can be found that the scores of round 2 are all higher than the scores of round 1. The reason is that the ranking results of round 2 combine the information of round 1 with the previous users’ information, which contributes to the plenty and enrichment of the users’ preferences.

Table 5 Scores to the ranking reviews and the control group

<table>
<thead>
<tr>
<th>Result users</th>
<th>User 1</th>
<th>User 2</th>
<th>User 3</th>
<th>User 4</th>
<th>User 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ranking of round 1</td>
<td>0.7</td>
<td>0.85</td>
<td>0.6</td>
<td>0.8</td>
<td>0.7</td>
</tr>
<tr>
<td>Ranking of round 2</td>
<td>0.8</td>
<td>0.93</td>
<td>0.87</td>
<td>1</td>
<td>0.9</td>
</tr>
<tr>
<td>D(control group)</td>
<td>0.2</td>
<td>0</td>
<td>0.4</td>
<td>0</td>
<td>0.5</td>
</tr>
</tbody>
</table>

With more and more evidence, the ranking result will be better than ever. We use Table 6 to express the five users’ favour about the ranking result intuitively.

Table 6 Five users’ feedback about the two rounds ranking reviews

<table>
<thead>
<tr>
<th>User’s satisfactory about ranking result</th>
<th>User 1</th>
<th>User 2</th>
<th>User 3</th>
<th>User 4</th>
<th>User 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ranking of round 1</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Ranking of round 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D(control group)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

As can be seen from Tables 5 and 6, five users prefer the ranking of round 2 to others. Since we get more user preference information by collecting user marked reviews twice, it enables us to obtain more user preference rules and use them to ratiocinate. The more sufficient and accurate the user preference rules are, the more we know about the user. Through using more accurate information, we can refine the range of features and make the result of ranking to conform to users’ preferences. Compared with the result of the first ranking, the second ranking reviews are more accurate and more cater to the user’s tastes.

Simultaneously, we collect the feedback of 300 people. And the results demonstrate that recognition of ranking algorithm is as high as 91.4%, that is, there are 91.4% people think the ranked reviews are useful for them to make purchase decisions. Similarly, the ranking result of other domains illustrates that the method can be applied to diverse domains as well.

6 Conclusions and future work

This paper puts forward a ranking method based on the users’ preferences. In the process of doing this, we find five inherent features that people pay more attention to, and meanwhile, use the domain features and the inherent features as the features to vectorise reviews. All of the works are done in accordance with the feature vectors of reviews that the users marked to get the user preference rules, and the rule set including the common rule, domain rule and user preference rule with support degree is well structured. Also, the support degree of rules and the users’ preferences are taken good advantages of to get the range of feature tendency. Besides, the distribution of features is converged thoroughly. Finally, we rank the reviews with the features distribution to calculate the scores of reviews. And this paper ends both with the finding that the results of the experiment well prove that the method used in ranking reviews can satisfy every user, and with the realisation of its goal to rank reviews for each user with consideration of his (her) preferences instead of the preferences of the general public that ignore some personalised preferences of users.

In the future, we will add some other features to adjust the ranking method, such as the syntactic structure, syntactic analysis and so on. The paper only puts forward a novel idea by using the words as the domain features to rank reviews for users. Next, we will attempt to add other words, such as descriptors, emoticons symbols, punctuations and so on to the method of ranking. Meanwhile, since the value of features is 0 or 1 in this paper, which is discrete, we can consider making some alternations on the value of features by giving continuous value to the feature. In follow-up studies, we will continue to study and research on the personalised ranking and try to discover more to improve the experience of online shopping.

Acknowledgements

This work was supported in part by National Science Foundation of China under Grants No. 61303105 and 61402304, the Humanity and Social Science general project of Ministry of Education under Grants No.14YJAZH046, the Beijing Natural Science Foundation under Grants No. 4154065, the Beijing Educational Committee Science and Technology Development Planned under Grants No. KM201610028015, Science and Technology Innovation Platform, Teaching teacher, and Connotation Development of Colleges and Universities.

References


