Recommendation Based on Contextual Opinions

Guanliang Chen and Li Chen

Department of Computer Science, Hong Kong Baptist University,
Hong Kong, China
{glchen,lichen}@comp.hkbu.edu.hk

Abstract. Context has been recognized as an important factor in constructing personalized recommender systems. However, most context-aware recommendation techniques mainly aim at exploiting item-level contextual information for modeling users' preferences, while few works attempt to detect more fine-grained aspect-level contextual preferences. Therefore, in this article, we propose a contextual recommendation algorithm based on user-generated reviews, from where users' context-dependent preferences are inferred through different contextual weighting strategies. The context-dependent preferences are further combined with users' context-independent preferences for performing recommendation. The empirical results on two real-life datasets demonstrate that our method is capable of capturing users' contextual preferences and achieving better recommendation accuracy than the related works.

Keywords: Context-aware recommender systems, user-generated reviews, aspect-level context, opinion mining, context-dependent preferences.

1 Introduction

It has been well recognized that context-aware recommender systems are able to outperform traditional recommenders because users' preferences can be depicted more accurately by capitalizing on contextual information [1]. Take one typical approach, *pre-filtering*[2], as an example, when estimating the rating of a user for an item, the recommender considers other users' data acquired in the same contextual situation of the target user given that they might be more valuable for capturing the user's contextual needs. However, the main limitation of existing context-aware techniques is that the preference modeling is purely at the item level. That is, the contextual preference is mainly related to the overall evaluation of an item, rather than to multiple aspects of the item (e.g., "food", "atmosphere", and "service" of the restaurant).

Although recent years some works have attempted to model users' preferences at the aspect level and employ multi-faceted preference profiles for product recommendation [3], movie recommendation [4, 5, 6], hotel recommendation [7], or restaurant recommendation [8], these works neglect the fact that such aspect-level preferences can be likely influenced by context. Consider a restaurant review from *Yelp* that is shown in Example 1.

V. Dimitrova et al. (Eds.): UMAP 2014, LNCS 8538, pp. 61–73, 2014.

Example 1. *I went to this place with my colleagues. The comfortable atmosphere here was perfect for business conversation. We ordered the salad and pizza, which were delicious. After I ate here, I decided to go back with my family because of the excellent food, even though the dining atmosphere here is not suitable for a family-gathering meal.*

In the above review, it can be seen that the aspect "atmosphere" is of more importance when the user is *having meals with colleagues*, while the aspect "food" is more of a concern when the user is *accompanied by family*. Thus, in our view, the aspect-level preferences can be context-sensitive. In other words, people may possess different aspect-level preferences in different contexts. We are hence interested in detecting such aspect-level contextual opinions particularly from user-generated reviews so as to more precisely model their preferences.

In our work, we emphasize two kinds of user preferences: *context-dependent* and *context-independent*. Specifically, the context-dependent preferences refer to the aspect-level contextual needs that are common to users who are under the same context; while the context-independent preferences are relatively less sensitive to contextual changes and reflect more stable requirements for an item's aspects over time. To derive the context-dependent preferences, we propose three variations of contextual weighting methods based on different text feature selection strategies: mutual information, information gain, and Chi-square statistic. They all focus on modeling the context-dependent preferences at the aspect level by analyzing the relation between the aspect frequency and context. The context-independent preferences, on the other hand, are also learned from reviews, but without considering the contextual influence. Our recommendation algorithm takes both kinds of preferences into account, which is empirically demonstrated superior to the state-of-the-art in terms of recommendation accuracy.

The following content is organized as follows. Section 2 briefly summarizes existing researches related to our work. Section 3 gives our research problem and methodology. Section 4 presents the experimental results on two real-life datasets. We draw the conclusion and indicate the future work in Section 5.

2 Related Work

Our work is mainly related to two branches of researches: context-aware recommenders and review-based recommenders.

Related Work on Context-Aware Recommenders. Existing context-aware techniques can be classified into three categories [1]: 1) *contextual pre-filtering*, by which data are first filtered according to contextual relationship before the classical recommendation approach (such as collaborative filtering) is applied [2, 9]; 2) *contextual post-filtering*, which adopts the contextual information to distill the recommendation results after the classical approach was applied [9]; 3) *contextual modeling*, which incorporates the context into the machine learning model (e.g., Tensor Factorization) for recommendation [10]. These works have been proven effective and successful in some applications like movie recommendation.

However, in reality, datasets that contain both ratings and the user-specified contexts rarely exist [11].

Compared to these works, the novelty of our work lies in that we utilize widely available reviews to establish the relation between aspect-level opinions and contextual factors for modeling users' preferences.

Related Work on Review-Based Recommenders. Review-based recommenders mainly rely on advanced opinion mining techniques to infer the reviewers' overall opinion (called virtual rating [12]) or even multi-aspect ratings, which are then leveraged into the standard recommenders [3, 5]. For instance, [8] developed a multi-label text classifier based on Support Vector Machine to reveal users' aspect-level evaluations of restaurants and generate recommendation through regression-based and clustering-based algorithms. In [3], the reviews are used to model users' multi-aspect preferences for computing user-user similarity during recommendation. Rather than using heuristic-based algorithms, some works turn to model-based approaches, such as Multi-Relational Matrix Factorization [5] or Tensor Factorization [4], for capitalizing on multi-aspect ratings as derived from reviews to augment recommendation. However, these works did not consider the contextual information that might also be extracted from reviews to derive the relation between aspects and contexts. To our knowledge, two works have endeavored to fill in this gap. [13] constructed the aspect-context relations via manual efforts and then combined them with user-specified preferences to generate recommendation, but it did not identify the contextual influences on users' aspect-level preferences. [14] created aspect-context relations by relating aspect-level opinions expressed in reviews with user-specified contexts, but it is still limited since the opinions on the same aspect in different contexts were not captured.

Compared to these works, our contribution rests in proposing an automatic review-based aspect-context relation detection method and carrying out in-depth research for revealing the impact of contextual factors on building users' aspect-level preferences.

3 Problem Statement and Methodology

As mentioned before, we mainly aim at addressing two problems: 1) *How to correlate aspect-level opinions with contextual factors and derive users' context-dependent preferences from their reviews?* 2) *How to leverage both context-dependent and context-independent preferences into computing the recommendation list?*

We summarize our solution in Figure 1. We first implement an automatic method to conduct contextual review analysis for mining contextual opinion tuples. Contextual opinion tuples refer to users' aspect-level evaluations of items under certain contexts, formally denoted as $\{\langle i, rev_{u,i}, a_k, Con_{i,k}\rangle \mid 1 \leq k \leq K\}$ (i.e., the user u's opinion a_k on the aspect k of item i under contexts $Con_{i,k}$ expressed in the review $rev_{u,i}$), where K denotes the number of aspects, and $Con_{i,k}$ is a vector whose element value equals 1 when the associated context

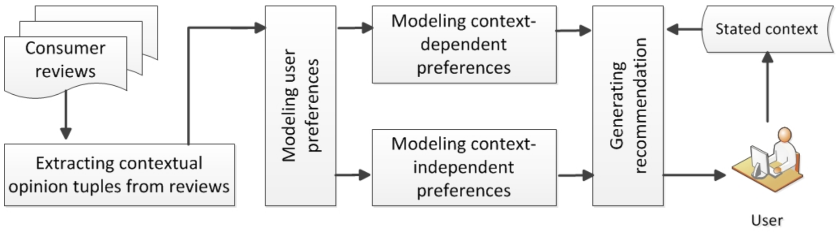

Fig. 1. Contextual preferences' detection and recommendation based on aspect-level review analysis

occurs and 0 otherwise. Then, we delve into detecting two types of user preferences. For context-independent preferences, we adopt the linear least-square regression method and the statistical t-test to attain users' weights (i.e., relative importance) laid on different aspects, and regard these weights as users' context-independent preferences. For context-dependent preferences, we propose three alternative contextual weighting methods to capture users' preference changes in different contexts. The three weighting methods are respectively based on three different text feature selection strategies: mutual information, information gain, and Chi-square statistic. Then, the context-independent and context-dependent preferences are combined via the multiplication approach for generating recommendation to the target user.

3.1 Extracting Contextual Opinion Tuples from Consumer Reviews

As described in Figure 1, the first step focuses on extracting contextual opinion tuples from reviews. Inspired by our previous work on aspect-level opinion mining [3] and related ones on context extraction [15, 11], we propose a synthetic method to perform contextual review analysis for extracting contextual opinion tuples. It mainly consists of four sub-steps:

1) Aspect Identification. In reviews, different terms are often used to refer to the same aspect of item. For example, terms "value", "price", "money" are all related to the aspect *Value* of restaurant. The task of aspect identification is thus to identify the relevant terms for each aspect. To this end, we adopt the bootstrapping method proposed in [16], by which each aspect is first equipped with a set of manually-selected keywords, and the other related terms are searched out through measuring the dependency between the aspect and the candidate terms based on Chi-square statistic [17]. Because the datasets collected for our experiments are about restaurants, we define five major aspects: *Value*, *Food*, *Atmosphere*, *Service*, and *Location*. Notice that only frequently occurring nouns and noun phrases, which are extracted by using a Part-of-Speech (POS) tagger[1], are considered as the prospective term candidates.

2) Opinion Detection. To determine users' opinions associated with each aspect-related term, we regard adjective words as opinion carriers. The adjectives

[1] `http://nlp.stanford.edu/software/tagger.shtml`

in the review are also extracted through the POS tagger and their sentiment polarity is determined with an opinion lexicon [18]. We summarize all of the opinions expressed in one sentence using a distance-based score: $score\,(s, f) = \sum_{op \in s} sent_{op} \;/\; d\,(op, f)$, where f denotes the aspect-related term that appears in sentence s, op denotes an opinion word in sentence s, $sent_{op}$ denotes its sentiment score (1 for positive and -1 for negative), and $d\,(f, op)$ denotes the distance from op to f.

3) Context Extraction. Before uncovering the aspect-context relation from reviews, we first employ a keyword matching method to extract contexts. Suppose that there are three contextual variables in restaurant reviews, including *Time*, *Occasion*, and *Companion*. Each contextual variable can be assigned with different values. For example, the values of *Companion* are "family", "friends", "colleague", "couple", and "solo". Moreover, each contextual value can be defined by a set of manually-selected keywords. For instance, the keywords related to contextual value "colleague" are {colleague, business, coworker, boss, etc.}. Therefore, once any of the keywords appear in a review sentence, the sentence will be tagged with the corresponding contextual value.

4) Aspect-Context Relation Construction. The next question is then how to relate the review's contextual values to its corresponding aspects, for which we propose to automatically construct the aspect-context relation based on the following rules: a) if both aspect-level opinion and context occur in the same sentence, they will be related; b) if a sentence only contains aspect-level opinion without mentioning context, the opinion will be related to contextual values that occur in the previous, nearest sentence. Notice that the user's opinion on the same aspect under different contexts could be different (such as the opinion on aspect "atmosphere" in Example 1). Thus, when constructing contextual opinion tuples, we sum up only the opinions pertinent to the aspect in the same context. In other words, the opinion a_k in tuple $\langle i, rev_{u,i}, a_k, Con_{i,k}\rangle$ is the aggregation of opinion scores of aspect-related terms that are under the same context $Con_{i,k}$. By applying our construction method, an aspect might be assigned with different opinion tuples in different contexts. For instance, the review presented in Example 1 can be extracted with tuples like $\langle i, rev_{u,i}, a_{\text{atmosphere}} = 1, Con_{i,\text{atmosphere}} = \text{“colleague”}\rangle$ and $\langle i, rev_{u,i}, a_{\text{atmosphere}} = -1, Con_{i,\text{atmosphere}} = \text{“family”}\rangle$ [2]. In this way, we expect that the user's preferences could be more precisely depicted.

3.2 Detecting Context-Independent Preferences

The context-independent preferences reflect the individual user's consistent aspect-level requirements for items. To detect such preferences, we adopt the linear least-square regression function with the statistical t-test to analyze the user's history data. To be specific, with aspect-level opinions obtained in Section 3.1, each review written by the user can be represented as a rating vector $\langle a_1, \ldots, a_K\rangle$ on the set of K aspects without considering their relations with

[2] To ease understanding, we use the context's value in the example, but it should be formally represented as a boolean vector.

contextual factors. All the rating vectors (corresponding to the set of reviews written by the user) can then be used to construct the linear least-square regression function, formally denoted as: $r_0 = \sum_{k=1}^{K} w_{u,k} \cdot a_k + \varepsilon$, in which the the overall rating r_0 is determined by the underlying interaction among multi-aspect ratings, ε denotes the error term, and $\langle w_{u,1}, \ldots, w_{u,K} \rangle$ denotes the user's weights laid on different aspects. Then, we apply the t-test to select weights that pass the significance level (e.g., $p < 0.1$) and regard these weights as the user's context-independent preferences.

3.3 Detecting Context-Dependent Preferences

The basic assumption behind our approach is that all the reviews written under the same context should be taken into account to capture the users' aspect-level context-dependent preferences. Therefore, we propose three variations of contextual weighting methods for assigning weights to aspects in different contexts and utilize these contextual weights to represent users' context-dependent preferences.

An intuitive method is to assign weights to aspects by analyzing the relation between the aspect's occurring frequency and the context. That is, the more frequently the aspect-related terms appear in the sentences of a specific context, the more important the aspect is to that context so it should receive higher weight. Hence, we first calculate the occurring frequency of aspect k under context c:

$$freq_{k,c} = \frac{\sum_{rev \in R} \sum_{s \in rev} \Delta_{s,c} \cdot \left(\sum_{f \in s} \Theta_{f,k} \right)}{\sum_{rev \in R} \sum_{s \in rev} \Delta_{s,c} \cdot \left(\sum_{f \in s} 1 \right)} \tag{1}$$

where f, s, and rev respectively represent an aspect-related term, a sentence, and a review, R denotes the set of all reviews, $\Delta_{s,c}$ denotes an indicator function whose value equals 1 if the sentence s is related to context c and 0 otherwise, and $\Theta_{f,k}$ denotes another indicator function whose value equals 1 if the term f is related to aspect k and 0 otherwise. In fact, Equation 1 computes the aspect frequency as the relative number of occurrences of its related terms in sentences related to context c. The aspect frequencies regarding different context values are used to compute the aspect's average frequency $avg_k = \sum_{c \in \mathcal{C}} freq_{k,c} / |\mathcal{C}|$ and standard deviation $stdv_k = \sqrt{\sum_{c \in \mathcal{C}} \left(freq_{k,c} - avg_k \right)^2 / |\mathcal{C}|}$ (where $\mathcal{C}$ denotes the set of context values), and we define $dev_{k,c} = freq_{k,c} - avg_k$. Then, we adopt the strategy proposed in [14] as our basis to compute the weight of aspect k regarding context value c:

$$w_{k,c} = \begin{cases} 1, & \text{if } |dev_{k,c}| < stdv_k \\ Max\left(0.1, 1/\left|\frac{dev_{k,c}}{stdv_k}\right|\right), & \text{if } \frac{dev_{k,c}}{stdv_k} <= -1 \\ Min\left(3, \frac{dev_{k,c}}{stdv_k}\right), & \text{else} \end{cases} \tag{2}$$

The above strategy mainly searches for important aspects based on the frequency identification. However, this method is limited in that it does not consider the importance of the aspect-related term in different contexts. For instance, the term *ambiance* may be important in both contexts: dining *as a couple* and *with colleagues*, but it might be more important to users in the first context than in the second one. To account for this, we propose to extend the above method by taking into account the term's weight. Particularly, as inspired by the research of Text Categorization in terms of how it selects representative features (i.e., words or terms) for categorizing documents, we propose three feature selection methods for identifying the context-dependent weights of aspect-related terms and compare their effectiveness in the experiment. The three methods include *Mutual Information*, *Information Gain*, and *Chi-Square Statistic*, which are detailed as follows.

Mutual Information. In information theory, mutual information is used to measure the mutual dependence between two random variables [17]. For our task, the two random variables can be *aspect-related term* and *context*. Given a term f and a context value c, the mutual information between them is defined as:

$$MI\left(f,c\right) = log\frac{p\left(f \wedge c\right)}{p\left(f\right) \cdot p\left(c\right)} \tag{3}$$

where $p\left(f\right)$ denotes the probability of f appearing in sentences, $p\left(c\right)$ denotes the probability of sentences that are associated with context c, and $p\left(f \wedge c\right)$ denotes the probability that f appears in sentences that are related to context c.

Information Gain. Information gain has been frequently employed in text categorization for measuring the number of bits of information obtained for categorizing documents by knowing the presence or absence of a word in a document [17]. We can hence apply this metric to measure the importance of an aspect-related term to a specific context. We concretely implement it as a binary classification model in which each sentence is classified into two categories, *related to* context c *or not*: $\mathcal{O} = \{c_{presence}, c_{absence}\}$. The information gain is then calculated as:

$$\begin{aligned} IG\left(f,c\right) = & -\sum_{c\in\mathcal{O}} p\left(c\right) \cdot log\, p\left(c\right) \\ & + p\left(f\right)\sum_{c\in\mathcal{O}} p\left(c \mid f\right) log\, p\left(c \mid f\right) + p\left(\bar{f}\right)\sum_{c\in\mathcal{O}} p\left(c \mid \bar{f}\right) log\, p\left(c \mid \bar{f}\right) \end{aligned} \tag{4}$$

where $\bar{f}$ denotes the absence of f in a sentence, and $p\left(c \mid f\right)$ denotes the probability that sentences containing f are related to context c.

Chi-Square Statistic. Based on Chi-square statistic, we can measure the lack of independence between an aspect-related term f and context c by computing the variance between the sample distribution and chi-square distribution [17]. The Chi-square statistic is formally defined as:

$$CHI\left(f,c\right) = \frac{D \times \left(D_1 D_4 - D_2 D_3\right)^2}{\left(D_1 + D_3\right) \times \left(D_2 + D_4\right) \times \left(D_1 + D_2\right) \times \left(D_3 + D_4\right)} \tag{5}$$

where D_1 is the number of times that f occurs in sentences related to context c, D_2 is the number of times that f occurs in sentences not related to c, D_3 is the number of sentences in context c that do not contain f, D_4 is the number of sentences that are neither related to context c nor containing f, and D is the number of times that all terms occur in sentences related to context c.

After obtaining the weights of the aspect-related terms via either of the three above-described methods, we further incorporate them into calculating the user's contextual weights placed on different aspects. Equation 1 is modified as follows:

$$freq_{k,c} = \frac{\sum_{rev\in R}\sum_{s\in rev}\Delta_{s,c}\cdot\left(\sum_{f\in s}\Theta_{f,k}\cdot MI\left(f,c\right)\right)}{\sum_{rev\in R}\sum_{s\in rev}\Delta_{s,c}\cdot\left(\sum_{f\in s}MI\left(f,c\right)\right)} \tag{6}$$

where $MI\left(f,c\right)$ is via Equation 3, which can be replaced with $IG\left(f,c\right)$ (Equation 4) or $CHI\left(f,c\right)$ (Equation 5). The results can then be applied in Equation 2 to determine the aspect's weight in a certain context.

3.4 Generating Recommendation

As stated before, users' behavior can be influenced by both context-independent preferences and context-dependent preferences. We hence combine both to compute a score of review $rev_{v,i}$ (wrote by user v for item i) for target user u:

$$score\left(u, rev_{v,i}, T\right) = \sum_{\langle i, rev_{v,i}, a_k, Con_{i,k}\rangle\in S\left(rev_{v,i}\right)}\prod_{c\in T}\left(1+\alpha\cdot w_{k,c}\right)\cdot w_{u,k}\cdot a_k\cdot g\left(Con_u, Con_{i,k}\right) \tag{7}$$

where $w_{k,c}$ is the context-dependent preference for aspect k under context c (derived via either of the three proposed variations of contextual weighting method in Section 3.3), $w_{u,k}$ is the target user's context-independent preference placed on aspect k (Section 3.2), α is a parameter used to control the relative contributions of context-independent and context-dependent preferences in computing the review's score, a_k is aspect k's score contained in contextual opinion tuple $\langle i, rev_{v,i}, a_k, Con_{i,k}\rangle$, $S\left(rev_{v,i}\right)$ is the set of contextual opinion tuples derived from $rev_{v,i}$, T is the set of contexts specified by the target user, Con_u denotes the vector form of T, and the function $g\left(Con_u, Con_{i,k}\right)$ is defined as:

$$g\left(Con_u, Con_{i,k}\right) = \begin{cases} 1, & \text{if } Con_u\cdot Con_{i,k}\neq 0 \\ 0, & \text{else} \end{cases} \tag{8}$$

Equation 8 ensures that only the aspect-level opinions pertinent to the target user's specified contexts are taken into account. The score of item i for user u is then finally calculated by averaging the scores of all of its reviews:

$$score\left(u, i\right) = avg_{rev_{v,i}\in R(i)}\left[score\left(u, rev_{v,i}, T\right)\right] \tag{9}$$

where $R\left(i\right)$ denotes the set of reviews for item i. The top-N items with highest scores are then retrieved and recommended to the target user. In the experiment, we set $N = 5, 10, 15$.

4 Experiment

4.1 Dataset and Evaluation Metrics

To conduct the experiment, we adopt two real-life restaurant datasets: one was crawled from TripAdvisor, and the other was from Yelp as published by the RecSys'13 challenge[3]. Table 1 shows their basic descriptions.

As for evaluation procedure, we adopt the per-user evaluation schema as commonly used in [19, 20]. That is, for each user, we randomly select three ratings which are above 4 (i.e., "like" the item), as well as the accompanying reviews (which are used to simulate the target user's contexts), that s/he provided to items as testing data while the others serve as training data. We then apply two metrics to measure the recommendation accuracy: 1) Hit ratio @ top-N recommendations (H@N), which measures the percentage of successes: H@N $= \sum_{t=1}^{T} \delta_{rank_t \leq N}/T$, where T is the number of testings, $rank_t$ is the ranking position of the user's choice (i.e., the item with high rating) in the t-th testing, and $\delta_{rank_t \leq N}$ is an indicator function that equals 1 if $rank_t \leq N$ (i.e., the recommendation list contains the choice), or 0 otherwise. 2) Mean reciprocal rank (MRR), which evaluates the ranking position of the target user's choice in the recommendation list: $MRR = \sum_{t=1}^{T} \frac{\delta_{rank_t \leq N}}{rank_t}/T$. Notice that, the target user's context for an item is simulated by performing the context extraction to the accompanying review in the testing data, and the parameter α is determined empirically through experimental trials. In addition, all of the reported results are the averages of per-user evaluations and the Student t-test is applied to compute the statistical significance of the difference between the compared methods.

Table 1. Dataset description

Dataset	#reviews	#users	#items	Sparsity	%reviews with contextual opinions
TripAdvisor	121932	6203	15315	99.87%	49.2%
Yelp	125286	3969	10581	99.70%	57.3%

4.2 Compared Methods

For the experiment, the following related methods were implemented to be compared with our proposed approaches MI/IG/CHI Connecter:

- **Context Freer.** This method adopts the regression-based method proposed in [6] to take into account the multi-aspect ratings derived from reviews. In fact, this method implements a simplified version of Equation 7, which does not consider the context-dependent preferences. We select this context-free method as our baseline and denote it as *Freer*.

[3] `http://recsys.acm.org/recsys13/recsys-2013-challenge/`

- **Context Pre-filter.** In accordance with [2], the extracted contextual information can be utilized at the item level, i.e., pre-filtering data according to contexts before applying the recommendation algorithm like *Freer*. That is, only the scores derived from reviews written under the target user's contexts are considered for calculating the item's score in Equation 9. We denote it as *Pre-filter*.
- **Default Connecter.** This method is similar to the one proposed in [14], which mines contexts from reviews and correlates them with users' opinions at the aspect level, but makes no distinction between users' opinions for the same aspect in different contexts. We denote it as *Default*.
- **Discriminative Connecter.** This method is also similar to the one proposed in [14], but relies on the results of contextual review analysis we obtained in Section 3.1 to assign context-dependent weights to aspects. Compared to our approaches, this method does not consider the weights of aspect-related terms. We denote it as *Discriminator*.
- **MI/IG/CHI Connecter.** The three methods proposed by us (see Section 3.3), which are different in terms of the feature selection metric used to calculate the aspect-related term's weight, respectively shorten to *MI* (mutual information), *IG* (information gain), and *CHI* (Chi-square statistic).

4.3 Results and Discussion

The experimental results on two datasets are shown in Table 2. We can have the following observations: 1) *Pre-filter* is better than *Freer*, which verifies that

Table 2. Experiment Results. Results marked with * are statistically significantly better than ($p < 0.001$) the method being compared. Here, the significance values are calculated between *Pre-filter* and *Freer*, *Default* and *Pre-filter*, *Discriminator* and *Default*, *MI/IG/CHI* and *Discriminator*.

Dataset	Method	H@5	H@10	H@15	MRR@5	MRR@10	MRR@15
Trip-Advisor	*Freer*	0.0145	0.0416	0.0760	0.0050	0.0085	0.0112
	Pre-filter	0.0296	0.0664*	0.1061*	0.0115*	0.0163*	0.0194*
	Default	0.0403*	0.0895*	0.1396*	0.0158*	0.0221*	0.0261*
	Discriminator	0.0464*	0.1008	0.1502*	0.0188*	0.0259	0.0297*
	MI	0.0565*	0.1173*	0.1707*	0.0237*	0.0317*	0.0359*
	IG	0.0680*	0.1369*	0.1938*	0.0301*	0.0391*	0.0436*
	CHI	**0.0915***	**0.1717***	**0.2310***	**0.0423***	**0.0528***	**0.0574***
Yelp	*Freer*	0.0205	0.0426	0.0598	0.0091	0.0119	0.0133
	Pre-filter	0.0267*	0.0521	0.0788*	0.0124*	0.0158*	0.0178*
	Default	0.0338*	0.0603*	0.0852*	0.0153*	0.0187*	0.0206*
	Discriminator	0.0487	0.0835*	0.1161*	0.0232	0.0277*	0.0303*
	MI	0.0543*	0.0951*	0.1261*	0.0266*	0.0320*	0.0345*
	IG	0.0729*	0.1195*	0.1608*	0.0361*	0.0422*	0.0454*
	CHI	**0.0985***	**0.1559***	**0.2075***	**0.0513***	**0.0588***	**0.0629***

it is meaningful to extract contexts from reviews and such contextual information does play an important part in enhancing recommendation; 2) *Default* defeats *Pre-filter*, which demonstrates that the contextual opinions can further be used to build more precise user profile, i.e., the aspect-level context-dependent preferences; 3) *Discriminator* is significantly superior to *Default* regarding most measures, which shows that it is meaningful to correlate users' aspect-level opinions with contexts based on review analysis. However, we also notice that the improvement achieved by *Discriminator* over *Default* is limited and some differences are not statistically significant. This is mainly owning to the limited amount of reviews that contain contextual opinions of the same aspect under different contexts (it is 23.01% in Yelp dataset and 17.6% in TripAdvisor dataset); 4) *MI/IG/CHI* are all significantly better than *Discriminator*, which suggests that the aspect-related term's relevance to context should also be considered when modeling the user's context-dependent preferences. Among the three variations, *CHI* achieves the best performance, followed by *IG*, and then *MI*. We believe that the differences can be explained by the way of how to compute the relevance of an aspect-related term to a specific context. The relevance weight computed by either *CHI* (i.e., Equation 5) or *IG* (i.e., Equation 4) takes all of the possible combinations of *presence* and *absence* statuses of the aspect-related term as well as the context into consideration. It hence can measure the weight more accurately over *MI* (i.e., Equation 3). In addition, *MI* tends to favor low-frequent terms, which might result in biases towards the calculation of the terms' relevance.

5 Conclusion and Future Work

In this paper, we presented a novel recommendation strategy that particularly performs contextual review analysis for detecting users' aspect-level context-dependent preferences and further combines them with users' context-independent preferences to generate recommendation. Through the experiment, we have successfully proved that: 1) it is meaningful to correlate users' aspect-level opinions (as expressed in their reviews) with the contextual factors; and 2) aspect-related terms are of important value to discriminate users' aspect-level preferences under different contexts. The experimental results on two datasets empirically show that our approaches significantly outperform the related context-aware recommendation techniques.

In the future, we plan to verify the performance of our method in other product domains, such as hotel recommendation. In addition, we will continue to explore different strategies for fusing together users' context-independent and context-dependent preferences. For instance, the parameter α in Equation 7 can be learned for each user by applying some machine learning techniques.

Acknowledgements. We thank grants ECS/HKBU211912 and NSFC/61272365.

References

[1] Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. In: Recommender Systems Handbook, pp. 217–253. Springer (2011)

[2] Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating contextual information in recommender systems using a multidimensional approach. ACM Transactions on Information Systems (TOIS) 23(1), 103–145 (2005)

[3] Chen, L., Wang, F.: Preference-based clustering reviews for augmenting e-commerce recommendation. Knowledge-Based Systems 50, 44–59 (2013)

[4] Wang, Y., Liu, Y., Yu, X.: Collaborative filtering with aspect-based opinion mining: A tensor factorization approach. In: 2012 IEEE 12th International Conference on Data Mining (ICDM), pp. 1152–1157. IEEE (2012)

[5] Jakob, N., Weber, S.H., Müller, M.C., Gurevych, I.: 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. ACM (2009)

[6] Adomavicius, G., Kwon, Y.: New recommendation techniques for multicriteria rating systems. IEEE Intelligent Systems 22(3), 48–55 (2007)

[7] Liu, L., Mehandjiev, N., Xu, D.L.: Multi-criteria service recommendation based on user criteria preferences. In: Proceedings of the Fifth ACM Conference on Recommender Systems, pp. 77–84. ACM (2011)

[8] Ganu, G., Kakodkar, Y.: Improving the quality of predictions using textual information in online user reviews. Information Systems 38(1), 1–15 (2013)

[9] Panniello, U., Tuzhilin, A., Gorgoglione, M., Palmisano, C., Pedone, A.: Experimental comparison of pre-vs. post-filtering approaches in context-aware recommender systems. In: Proceedings of the Third ACM Conference on Recommender Systems, pp. 265–268. ACM (2009)

[10] Karatzoglou, A., Amatriain, X.: Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. In: Proceedings of the Fourth ACM Conference on Recommender Systems, pp. 79–86. ACM (2010)

[11] Li, Y., Nie, J., Zhang, Y.: Contextual recommendation based on text mining. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, pp. 692–700. Association for Computational Linguistics (2010)

[12] Zhang, W., Ding, G., Chen, L., Li, C., Zhang, C.: Generating virtual ratings from chinese reviews to augment online recommendations. ACM Transactions on Intelligent Systems and Technology (TIST) 4(1) (2013)

[13] Carter, S., Chen, F., Muralidharan, A.S., Pickens, J.: Dig: A task-based approach to product search. In: Proceedings of the 16th International Conference on Intelligent User Interfaces, pp. 303–306. ACM (2011)

[14] Levi, A., Mokryn, O., Diot, C., Taft, N.: Finding a needle in a haystack of reviews: Cold start context-based hotel recommender system. In: Proceedings of the Sixth ACM Conference on Recommender Systems, pp. 115–122. ACM (2012)

[15] Hariri, N., Mobasher, B., Burke, R., Zheng, Y.: Context-aware recommendation based on review mining. In: Proceedings of the 9th International Workshop on Intelligent Techniques for Web Personalization and Recommender Systems (ITWP), International Joint Conferences on Artificial Intelligence (IJCAI), pp. 30–36 (2011)

[16] Wang, H., Lu, Y., Zhai, C.: Latent aspect rating analysis on review text data: a rating regression approach. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 783–792. ACM (2010)

[17] Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: Proceedings of the 14th International Conference on Machine Learning, vol. 97, pp. 412–420 (1997)
[18] Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 347–354. Association for Computational Linguistics (2005)
[19] Shani, G., Gunawardana, A.: Evaluating recommendation systems. In: Recommender Systems Handbook, pp. 257–297. Springer (2011)
[20] Codina, V., Ricci, F., Ceccaroni, L.: Exploiting the semantic similarity of contextual situations for pre-filtering recommendation. In: Carberry, S., Weibelzahl, S., Micarelli, A., Semeraro, G. (eds.) UMAP 2013. LNCS, vol. 7899, pp. 165–177. Springer, Heidelberg (2013)