Recommender systems based on user reviews: the state of the art

Article *in* User Modeling and User-Adapted Interaction · June 2015

DOI: 10.1007/s11257-015-9155-5

3 authors:

Li Chen
Hong Kong Baptist University
90 PUBLICATIONS **3,388** CITATIONS

SEE PROFILE

Guanliang Chen
Delft University of Technology
16 PUBLICATIONS **354** CITATIONS

SEE PROFILE

Feng Wang
Hong Kong Baptist University
8 PUBLICATIONS **296** CITATIONS

SEE PROFILE

Noname manuscript No.
(will be inserted by the editor)

Recommender Systems Based on User Reviews: The State of the Art

Li Chen · Guanliang Chen · Feng Wang

Received: date / Accepted: date

Abstract In recent years, a variety of review-based recommender systems have been developed, with the goal of incorporating the valuable information in user-generated textual reviews into the user modeling and recommending process. Advanced text analysis and opinion mining techniques enable the extraction of various types of review elements, such as the discussed topics, the multi-faceted nature of opinions, contextual information, comparative opinions, and reviewers' emotions. In this article, we provide a comprehensive overview of how the review elements have been exploited to improve standard content-based recommending, collaborative filtering, and preference-based product ranking techniques. The review-based recommender system's ability to alleviate the well-known rating sparsity and cold-start problems is emphasized. This survey classifies state-of-the-art studies into two principal branches: *review-based user profile building* and *review-based product profile building*. In the user profile sub-branch, the reviews are not only used to create term-based profiles, but also to infer or enhance ratings. Multi-faceted opinions can further be exploited to derive the weight/value preferences that users place on particular features. In another sub-branch, the product profile can be enriched with feature opinions or comparative opinions to better reflect its assessment quality. The merit of each branch of work is discussed in terms of both algorithm development and the way in which the proposed algorithms are evaluated. In addition, we discuss several future trends based on the survey, which may inspire investigators to pursue additional studies in this area.

L. Chen
Department of Computer Science, Hong Kong Baptist University, Hong Kong
E-mail: lichen@comp.hkbu.edu.hk

G. Chen
Department of Computer Science, Hong Kong Baptist University, Hong Kong
E-mail: anguschan7733@gmail.com

F. Wang
Department of Computer Science, Hong Kong Baptist University, Hong Kong
E-mail: fwang@comp.hkbu.edu.hk

Keywords Recommender systems · User reviews · Text analysis · Opinion mining · User profile building · Product profile building · Content-based recommending · Collaborative filtering · Preference-based product ranking

1 Introduction

Recommender systems (RS) have attracted attention in both academia and industry. Such systems help to manage information overload by autonomously gathering information and proactively tailoring it to individual interests (Adomavicius and Tuzhilin 2005), e.g., what product to buy (*Amazon*), what song to listen to (*Last.fm*), which hotel to stay in (*TripAdvisor*), and so on. Currently, most of the various types of recommender techniques use user-provided ratings to infer user preferences. There are two common memory-based collaborative filtering (CF) approaches (Adomavicius and Tuzhilin 2005; Herlocker et al 2004; Sarwar et al 2001; Schafer et al 2007; Su and Khoshgoftaar 2009): the *user-based method* uses ratings to associate a user with a group of like-minded users and then recommends to the target user a set of items that are enjoyed by her/his neighbors; and the *item-based method* aims to find items that are similar to those that a user has viewed/purchased before. In contrast, model-based CF systems focus on learning the latent factors that represent users' inherent preferences over an item's multiple dimensions (Koren and Bell 2011; Koren et al 2009).

Collaborative filtering techniques perform well when there is sufficient rating information (Su and Khoshgoftaar 2009). However, their effectiveness is limited when the well-known *rating sparsity* problem occurs, due to the poor coverage of recommendation space (Garcia Esparza et al 2010), or the difficulty in letting users express their preferences as scalar ratings on items (Leung et al 2006). To address this problem, content-based recommender approaches have been developed that rely instead on the content representations of items to locate items that have similar content to items the target user liked (Lops et al 2011; Pazzani and Billsus 2007). Some studies have used other types of user-generated information, such as tags (freely chosen/written keywords) (Marinho et al 2011; Zhao et al 2008), and social relationships (like friendship, membership, and trust relationship) (Beilin and Yi 2013; Chen et al 2013; Yang et al 2012), to augment the accuracy of recommendation. However, these methods are still inadequate, especially when the target user has little historical data. They are also of limited usefulness when the overall data sparsity level is high.

Therefore, in this paper, we particularly emphasize *user reviews*, and provide a comprehensive survey of recent attempts to use the valuable information in reviews to solve the rating sparsity issue. The growing popularity of social and e-commerce media sites has encouraged users to naturally write reviews describing their assessment of items. These reviews are usually in the form of textual comments that explain *why* they like or dislike an item based on their usage experiences. The system can capture the multi-faceted nature of a user's opinions from her/his reviews and hence build a fine-grained preference model for the user, which however cannot be obtained from overall ratings. Empirical findings from marketing and consumer behavior studies

have also documented the positive influence of product reviews on the decision processes of new users (Chatterjee 2001; Chevalier and Mayzlin 2006; Kim and Srivastava 2007).

There are increasing efforts to incorporate the rich information embedded in reviews into the process of user modeling and recommendation generation. In particular, information obtained from reviews is likely to benefit recommender systems in the following three ways (Chen and Wang 2013; Garcia Esparza et al 2011; Hariri et al 2011; Jamroonsilp and Prompoon 2013; Levi et al 2012; McAuley and Leskovec 2013; Pero and Horváth 2013; Wang et al 2012; Yates et al 2008; Zhang et al 2013).

- First, they can help to deal with the problem of large data sparsity by providing additional information about user preferences. In the extreme case of no ratings being available, the reviews can be used to infer the ratings that CF systems require (see Section 4.2).
- Second, they can help to solve the cold-start problem for new users. Usually, there are two types of new users: a user with limited experience with the items, who therefore has not provided many ratings; and a user who is totally new to the system. For the first type, Section 4.3 summarizes research that has used reviews to enhance ratings so that a preference model can be constructed for a user with few ratings by aligning the review information (such as review topics or feature opinions) with numerical ratings. For the second type, a user's current preference is often elicited on site when s/he is using the system, so the main focus has been on using review elements to either assist the user to complete the preference (see Section 4.4.3) or enrich the product profile (see Section 5). As an example from product profiles, the comparative opinions extracted from reviews can be helpful for constructing product-to-product comparison relationships and enhancing the ranking quality.
- Third, when the dataset is not sparse (i.e., in a relatively dense data condition), the reviews can still be useful. They have been used to determine rating quality (with the degree of the review's helpfulness) (see Section 4.3.1), to help derive users' context-dependent preferences (with the contextual information extracted from reviews) (see Section 4.3.4), and to learn users' latent preference factors by considering the aspect opinions mentioned in reviews (see Section 4.4.1).

In the following content, we classify state-of-the-art studies into two main categories according to the exact role that reviews have taken (see Figure 1): *review-based user profile building*, which emphasizes exploiting reviews to construct a user's profile (Section 4); and *review-based product profile building*, which focuses on using reviews to build informative product profile (Section 5). In the first category, we further sub-categorize the related studies into several groups according to the type of user profile they emphasize: those based on *term-based profile*, which is built by extracting frequent terms from reviews (Section 4.1); those based on *rating profile*, which use reviews to either infer ratings (when the ratings are not available) (Section 4.2) or enhance existing ratings (Section 4.3); and those based on *feature preference*, which embody users' multi-faceted opinions in reviews (Section 4.4). We show how the standard recommending approaches (see the background in Section 2), such as the content-based approach, memory-based CF (including user-based and item-based methods), model-based CF, and preference-based product ranking, are improved by incorporating the reviews. We also highlight the performance of those

review-based recommenders in different data conditions based on experimental findings. Finally, we discuss the practical implications of these findings (Section 6) and future trends (Section 7).

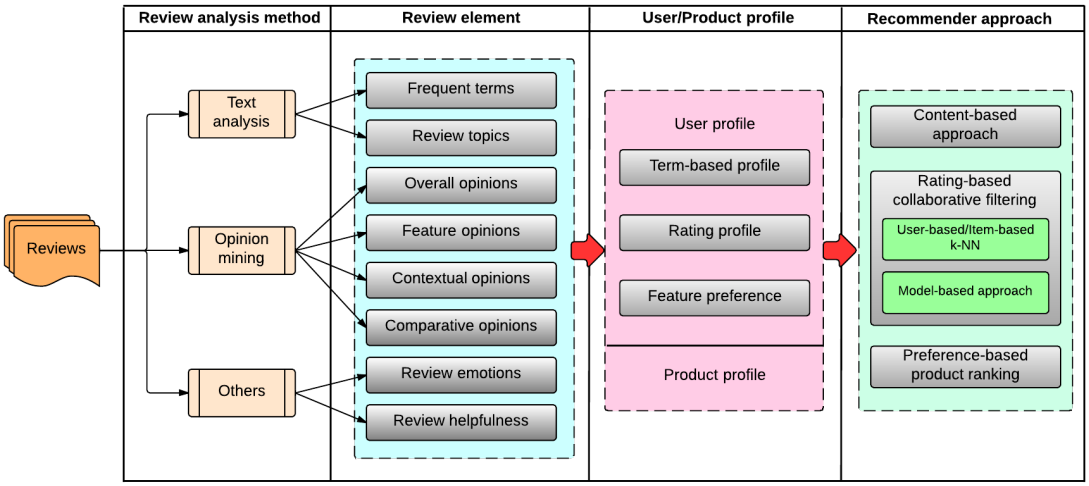

Fig. 1 Research focuses of review-based recommender systems.

2 Standard Recommending Approaches

In this section, we introduce three standard recommending approaches: the content-based approach, rating-based collaborative filtering, and preference-based product ranking, given that review-based recommender systems have been mainly targeted to address the limitations of these approaches for increasing the recommendation accuracy.

2.1 Content-based Approach

One typical recommending method is the content-based approach. It draws on detailed representations of items to build a user profile (Balabanović and Shoham 1997; Garcia Esparza et al 2010; Lops et al 2011; Pazzani and Billsus 2007). Specifically, it first assumes that each item can be defined by a profile in the form of a vector $X = (x_1, x_2, \ldots, x_n)$, where x_i might be a term obtained from either the item's textual description, meta-data, or keywords. A weighting strategy, such as the *Term Frequency/Inverse Document Frequency* (TF-IDF) measure (Salton and Buckley 1988), can be applied to determine each term's representativeness. For instance, one way to compute the TF-IDF weight of word w_i in document d is as follows (note that other formulations for TF-IDF can be found in (Manning et al 2008)):

$$x_{d,i} = f_{d,i} \times \log(N/n_i) \tag{1}$$

where $f_{d,i}$ is the term frequency of w_i in document d, N is the number of documents, and n_i is the number of documents in which w_i appears. Then, the item's profile, given document d, is defined as $X(d) = (x_{d,1}, \ldots, x_{d,n})$. Correspondingly, a user's profile $X(u)$ can be established by aggregating the profile vectors of the items that the user liked or purchased in the past: $X(u) = (x_{u,1}, \ldots, x_{u,n'})$. The content-based approach then tries to recommend items whose profiles are most similar to the user's profile (Lops et al 2011). For example, the *Cosine* similarity measure can be used:

$$sim(X(u), X(d)) = \frac{X(u) \cdot X(d)}{||X(u)||||X(d)||} \tag{2}$$

The content-based approach has been mainly used for recommending items containing textual information, such as documents, web sites, and news (Balabanović and Shoham 1997). It has also appeared in recommender applications for TVs (Smyth and Cotter 2000), e-commerce (Schafer et al 2001), and travel (Chelcea et al 2004). It assumes that the items' static descriptions can be obtained for extracting frequent terms, which, however, may not be the case in reality. Another limitation is that, because the profiles are based on static descriptions, different users are likely to have the same profile if they have visited the same items, even if their preferences among these items are different.

2.2 Rating-based Collaborative Filtering

The rating-based collaborative filtering (CF) system uses the ratings (such as 4 out of 5 star rating) that users have provided for the items. Normally, a user-item rating matrix R is created with size $U \times I$, where the entry $r_{u,i}$ denotes the rating that user u gives to item i. The goal of CF is to predict the unknown ratings in R based on the available ratings. There are two major sub-branches of CF: *memory-based CF* and *model-based CF* (Adomavicius and Tuzhilin 2005).

As mentioned in Section 1, the memory-based CF approach is typically *user-based* or *item-based*. Taking the user-based method as an example, we can run the k-Nearest Neighbors (k-NN) algorithm (Adomavicius and Tuzhilin 2005) to locate k users most similar to the target user, for which a similarity metric such as the *Pearson correlation coefficient* can be applied to identify the similarity between two users u and v:

$$sim(u, v) = \frac{\sum_{i \in I(u,v)} (r_{u,i} - \overline{r}_u)(r_{v,i} - \overline{r}_v)}{\sqrt{\sum_{i \in I(u,v)} (r_{u,i} - \overline{r}_u)^2} \sqrt{\sum_{i \in I(u,v)} (r_{v,i} - \overline{r}_v)^2}} \tag{3}$$

where $I(u, v)$ denotes the items that both u and v have rated, and $\overline{r}_u$ is the average rating of the items rated by user u. The predicted rating for an unknown item i for the target user u is then computed as:

$$\hat{r}_{u,i} = \overline{r}_u + \frac{\sum_{v \in Neighbors(u)} (r_{v,i} - \overline{r}_v) \times sim(u, v)}{\sum_{v \in Neighbors(u)} |sim(u, v)|} \tag{4}$$

where $Neighbors(u)$ denotes the set of k most similar users. The items with the highest predicted ratings are then recommended to the user.

In contrast, model-based CF approaches aim to train a parametric model with the rating matrix, which can then be applied to predict ratings of unknown items or to rank these items. Example models include the clustering model (Chee et al 2001; Ungar et al 1998), Bayesian networks (Horvitz et al 1998; Zigoris and Zhang 2006), the aspect model (Hofmann 2004), and the latent factor model (Koren et al 2009); the latent factor model has become the most popular model in recent years, as it can discover the latent interests underlying the user ratings. The standard form of the latent factor model is the low-rank matrix factorization (MF) (Koren et al 2009). Specifically, item i and user u can be respectively associated with k-dimensional latent factors, i.e., $q_i \in \mathbb{R}^k$ for the item and $p_u \in \mathbb{R}^k$ for the user, which can be considered as k properties of the item and the user's preference for these properties. Then, the user u's rating on item i is predicted as

$$\hat{r}_{u,i} = q_i^T p_u \tag{5}$$

The involved parameters can be optimized by minimizing the following squared error function:

$$\min_{p_*,q_*} \sum_{(u,i)\in R} (r_{u,i} - q_i^T p_u)^2 + \lambda(||q_i||^2||p_u||^2) \tag{6}$$

where λ is the tradeoff parameter for the regularization.

A more commonly used latent factor model is the biased MF (Koren and Bell 2011; Koren et al 2009), which takes into account the user and item biases that are likely to be caused by rating deviations. For example, some users may consistently give higher ratings than other users, and some items may get higher ratings than other items. Formally, the model predicts the rating $r_{u,i}$ for item i for user u as

$$\hat{r}_{u,i} = \mu + b_i + b_u + q_i^T p_u \tag{7}$$

where μ is the global rating mean, and b_i and b_u are, respectively, the vectors of item and user biases that indicate how the ratings deviate from μ. A variety of methods can be used to optimize the parameters implied by Equation 7, such as the alternating least-squares or gradient-based methods (Koren and Bell 2011).

Except for the biased MF, variations such as probabilistic MF and attribute-based MF have also been applied in review-based recommender systems (Raghavan et al 2012; Seroussi et al 2011). Probabilistic MF (Salakhutdinov and Mnih 2008) offers a probabilistic foundation for learning the parameters in MF model. For instance, a probabilistic linear model with the Gaussian distribution can be made to model an item's latent factors, and a user's latent factors can be viewed as the weights. In attribute-based MF (Koren et al 2009), an element is added to Equation 7 for incorporating user attributes, such as gender and age. Every user is concretely described by a set of binary attributes $A(u)$. An attribute-factor matrix is learnt from the available ratings, where each attribute a is associated with a latent factor vector $y_a \in \mathbb{R}^k$. The attributes' latent factors are then accumulated to represent a user's latent factors p_u, such that $p_u = \sum_{a\in A(u)} y_a$.

However, as the CF approaches rely mainly on rating information, a user normally needs to provide a sufficient number of ratings before the system can return accurate recommendation. The performance is unavoidably impaired when the user is new, which is known as the "new user" phenomenon. Another related limitation is the "new item" problem, as a new item must be rated by a certain number of users before the system is able to recommend it.

2.3 Preference-based Product Ranking

In cases where a product can be described by a set of attributes (e.g., price, weight, optical zoom, size of camera), the preference-based product ranking is usually performed (Chen and Pu 2004; Pu and Chen 2005; Smyth 2007). In this approach, a user's preference can be elicited in the form of weight and/or value criterion placed on each of the attributes.

Specifically, a user's preference can be represented as $(\{V_1, \ldots, V_n\}, \{w_1, \ldots, w_n\})$, where V_i is the value function (criterion) a user specifies for attribute a_i, and w_i is the relative importance (i.e., the weight) of a_i. For instance, the value function can be defined as $V_i(a_i) = \frac{p(a_i)-min(a_i)}{max(a_i)-min(a_i)}$ for the *more-is-better* attribute (like the camera's optical zoom), where $p(a_i)$ is a candidate product's value on attribute a_i, and $min(a_i)$ and $max(a_i)$ are, respectively, the minimum and maximum values of a_i in the whole data catalog. According to the multi-attribute utility theory (MAUT) (Keeney and Raiffa 1976), a utility can be computed for each product $< a_1, a_2, ..., a_n >$, as follows:

$$U(< a_1, a_2, \ldots, a_n >) = \sum_{i=1}^{n} w_i \times V_i(a_i) \tag{8}$$

This weighted additive form is a simplified version of MAUT, under the assumption of preferential and additive independence. It is inherently in accordance with the compensatory decision strategy, the weighted additive rule (WADD), that resolves conflicting values explicitly by considering attribute tradeoffs (Payne et al 1993). All of the products can then be ranked according to their utilities, with the top products recommended to the user. In the situation where only a weight preference is obtained, the dot product of the preference vector $< w_1, \ldots, w_n >$ and the product vector $< a_1, \ldots, a_n >$ can be used as the function to rank products.

In case-based recommender systems (Lorenzi and Ricci 2005; McSherry 2003; Smyth 2007), a user's preference is represented as a query product that s/he likes. The system then constructs each product case as a set of attributes, and retrieves those that are similar to the user's query case. For example, the similarity between a candidate product case (c) and a user's query case (t) can be calculated via (Smyth 2007):

$$Similarity(t, c) = \frac{\sum_{i=1}^{n} w_i \times sim_i(t_i, c_i)}{\sum_{i=1}^{n} w_i} \tag{9}$$

where $sim_i(t_i, c_i)$ gives the similarity between the two products regarding the *i-th* attribute (e.g., via Euclidean distance) and w_i is the *i-th* attribute's weight. Hence, the products most similar to the user's query can be recommended.

A non-personalized ranking method, which has often been used as the baseline to be experimentally compared with the above preference-based ranking, is the so-called popularity-based ranking (Musat et al 2013; Poriya et al 2014). In this approach, no user preference info is needed. It scores each candidate product by simply averaging all users' ratings. Alternatively, if ratings are given for the product's attributes, the product score can be calculated as

$$score(i) = \frac{\sum_{a_i \in A} \frac{\sum_{(u,i) \in R(i)} r_{u,a_i}}{N(a_i)}}{n} \tag{10}$$

where r_{u,a_i} denotes the rating that user u gives to attribute a_i of item i, $N(a_i)$ denotes the total number of users who have rated a_i, and n is the total number of attributes. The products with the highest scores are thus recommended.

3 Review Elements

In this section, we summarize the valuable information that can be extracted from reviews and used to enhance the above-mentioned standard recommending approaches. Indeed, although the raw review information is in un-structured, textual form that cannot be easily understood by the system, the advances in the field of topic modeling and opinion mining (also called sentiment analysis) make it possible to interpret reviews and extract useful elements from them (Blei et al 2003; Liu 2012; Pang and Lee 2008; Snyder and Barzilay 2007). We list these review elements and briefly describe how they have been used in review-based recommender systems. Details about the related studies are given in the relevant sections.

Fig. 2 A hotel review example from *TripAdvisor* (note that the blue underline marks *feature opinions*, the red underline marks *comparative opinions*, and the yellow underline marks *contextual information*).

1. *Frequent terms*: Because a review is written in natural language, the most obvious way of analyzing it is to identify frequently used terms. A weighting measure such as TF-IDF (as

mentioned in Section 2.1) can be applied to determine how representative each term is in the review. The extracted terms can then be used to characterize the reviewer with a term-based user profile. In (Garcia Esparza et al 2010, 2011), the built profile is leveraged into the content-based approach to generate recommendations (see Section 4.1).

2. *Review topics*: Topics are the aspects of an item that a writer discusses in a review. For example, in Figure 2 (a real hotel review from *TripAdvisor*[1]), the mentioned topics include the hotel room's quality, food, gym facility, location. There are two approaches to identifying topics in reviews. The first is the frequency-based approach, which first extracts frequently occurring nouns based on a set of seed words, and then groups the nouns into topics manually or according to a pre-defined dictionary (Musat et al 2013). The second approach is to use a topic modeling technique such as Latent Dirichlet Allocation (LDA) (Blei et al 2003), to automatically uncover hidden topics in review documents. The objective of LDA is to cluster words that co-occur in documents to form topics, so that each document d can be represented as a K-dimensional topic distribution θ_d, and each topic k is assigned a word distribution ϕ_k to indicate the probability that a particular word is related to it. The discovered review topics can then be used to enhance real ratings in CF based recommending approaches (McAuley and Leskovec 2013; Seroussi et al 2011) (see Section 4.3.2).
3. *Overall opinions*: A user's sentiment orientation (i.e., positive or negative) towards an item can be inferred from review to represent her/his overall opinion. For instance, for the review in Figure 2, we can infer that the reviewer has an overall positive opinion about this hotel. A simple way to estimate the overall opinion is to aggregate the sentiments of all of the opinion words that are contained in the review (Leung et al 2006; Zhang et al 2013). Alternatively, a machine learning algorithm (such as the naive Bayesian classifier or Support Vector Machine (SVM)) can be adopted to learn the opinion and classify it into a proper sentiment category (Pang et al 2002; Poirier et al 2010b). The inferred overall opinions can then be converted into virtual ratings, which may take the role of real ratings in CF (Poirier et al 2010b; Zhang et al 2013) (see Section 4.2), or be used to enhance real ratings (Pero and Horváth 2013) (see Section 4.3.3).
4. *Feature opinions*: In addition to the overall opinion, fine-grained opinions about specific features of an item can also be extracted from reviews. For example, the review sentence (see Figure 2) "*Rooms are spacious and luxuriously appointed*" expresses the author's positive sentiment towards the feature "room". In a raw review, the feature is normally expressed as a noun or noun phrase, which may refer to a distinct object such as the item itself (e.g., "hotel"), one of its components (e.g., "bedroom" or "bathroom"), its function (e.g., "service"), or a property of the component (or function) (e.g., "size"). Multiple features can further be mapped to an aspect to indicate an upper-level abstraction (Hu and Liu 2004b; Popescu and Etzioni 2005). For example, the hotel's features "room," "size," and "cleanness" can be projected onto the aspect "room quality". The typical approaches to feature extraction include statistics based methods, such as one that captures frequently occurring nouns/phrases as feature candidates through association rule mining (Hu and

[1] http://www.tripadvisor.com/

Liu 2004b), a LDA or SVM based method for identifying aspects directly, or a machine learning method based on a lexicalized Hidden Markov Model (L-HMMs) (Jin et al 2009) or Conditional Random Fields (CRFs) (Miao et al 2010; Qi and Chen 2010). The opinions associated with features (or aspects) are then identified by looking for nearby adjectives, or through opinion pattern mining (Hu and Liu 2004a; Moghaddam and Ester 2010). In Section 4.4.1, we show how the feature opinions can be modeled as latent preference factors of users and used to augment model-based CF (Jakob et al 2009; Wang et al 2012); or exploited to derive users' weight preferences (Chen and Wang 2013; Liu et al 2013) or attribute value preferences (Wang et al 2013), for use in preference-based product ranking (see Sections 4.4.2 and 4.4.3). In addition, they are helpful for building product profiles to increase the ranking quality (Aciar et al 2007; Dong et al 2013b; Yates et al 2008) (see Section 5.1).

5. *Contextual opinions*: The review sentence "*first visit to company's Hong Kong offices*" (see Figure 2) provides the contextual information related to this review. As another example, "*This camera's image quality is not good when I used it to take pictures at night,*" "at night" is the context, "image quality" is the feature, and "not good" is the opinion that is negative. This kind of contextual opinion can reflect the contextual uses (or conditions) of an item or a specific feature, which can be discovered from reviews through keyword matching (Chen and Chen 2014), rule-based reasoning (Li et al 2010), or a LDA-based classifier (Hariri et al 2011; Ramage et al 2009). In recommender systems, they can be combined with star ratings to infer a user's utility of selecting an item in different contexts (Hariri et al 2011), or to model a user's context-related latent factors (Li et al 2010) or context-dependent aspect preferences (Chen and Chen 2014; Levi et al 2012) (see Sections 4.3.4 and 4.4.2).
6. *Comparative opinions*: Another type of opinion that can be extracted from reviews is comparative opinion (Jindal and Liu 2006), such as the sentence "*Bed was comfortable, perhaps not as good as some St. Regis' but clearly better and more luxurious than the Westins heavenly stateside*" (see Figure 2). Comparative opinions indicate whether an item is superior or inferior to another, with regard to some feature. Such opinions can be extracted using a set of special linguistic rules (Ganapathibhotla and Liu 2008). In Section 5.2, it can be seen that comparative opinions can be used to model products' comparative relationships via a graph, and thus improve the products' ranking quality (Jamroonsilp and Prompoon 2013; Li et al 2011; Zhang et al 2010).
7. *Review emotions*: Emotion reflects a reviewer's mood (e.g., sadness, joy, distress, happiness, etc.) when writing the review. It is harder to detect in review sentences than opinion. However, we can construct an emotion classifier to automatically label a text with certain emotion(s) (Shaikh et al 2009). The extracted review emotions can then be used to determine the probability that a user will like an item, as proposed in (Moshfeghi et al 2011) (see Section 4.3.5). Emoticons (e.g., smiley and sad faces), as symbolic representations of emotions, can also be aggregated with opinion words to infer the reviewer's overall rating (Zhang et al 2013) (see Section 4.2).
8. *Review helpfulness*: Beyond review texts, the number of "helpful" votes given by readers to a review can also be useful. For instance, this number can be used to determine the

accompanying rating's quality score (Raghavan et al 2012). The quality-aware ratings can then be input into the CF framework to make better predictions (see Section 4.3.1).

The above list includes the main elements of a review that can be considered. Clearly, rich information is embedded in, or along with, reviews. In the following sections, we discuss in detail current review-based recommender systems. Specifically, we discuss how they exploit review elements to improve the standard recommending approaches' performance.

4 Review-based User Profile Building

This section summarizes related studies that use reviews to build or enhance user profiles. The classification of these approaches is primarily based on the type of user profile each system emphasizes: the *term-based profile*, the *rating profile*, or the *feature preference*. In each sub-branch, we describe how the user profile can be constructed based on reviews and how this shapes the recommendation process.

4.1 Term-based Profile

In this sub-branch of approaches, users are characterized by the textual content of their reviews (Garcia Esparza et al 2010). The term-based user profile $\{t_1, \ldots, t_n\}$ includes keywords extracted from user reviews, and each keyword t_j is assigned a weight $U_{i,j}$ by TF-IDF (see Section 2.1) that indicates its importance to the reviewer U_i; this is called *term-based user index* in (Garcia Esparza et al 2010). Similarly, a *term-based product index* $P_{i,j}$ can also be constructed using terms extracted from reviews posted to the product P_i. During the recommendation process, the target user's index serves as a query that is matched to the products' indices and used for retrieving the most similar products. This method is called an *index-based approach*, and it is essentially an extension of the content-based recommending approach (see the background of this approach in Section 2.1) (Garcia Esparza et al 2011).

Evaluation. The proposed *index-based approach* has been evaluated using a dataset collected from *Blippr*, which consists of four product catalogs: *movies* (with 1,080 items, 542 users, and 15,121 reviews), *applications* (with 268 items, 373 users, and 10,910 reviews), *books* (with 313 items, 120 users, and 3,003 reviews), and *games* (with 277 items, 164 users, and 3,472 reviews) (Garcia Esparza et al 2010). The results demonstrate that the index-based approach that uses reviews to build user/product index outperforms methods that consider tags, across all product catalogs in terms of the following measures: *precision*, which measures the proportion of recommended items that are enjoyed by the target user; *recall*, which measures the proportion of enjoyed items that are recommended within the whole set of enjoyed items; $F1$, which gives the harmonic mean of precision and recall; and *coverage*, which measures the percentage of $\langle user, item \rangle$ pairs for which the system is capable of making predictions (The details of these evaluation metrics can be found in (Shani and Gunawardana 2011)). A second experiment is

conducted to compare the index-based approach with the CF baselines that use ratings (including user-based and item-based k-NN methods) (Garcia Esparza et al 2011). This experiment uses the *Flixster* dataset, which contains 43,179 reviews given by 2,157 users to 763 movies. The results show that although the accuracy of the index-based approach is slightly lower than that of rating-based CF techniques, it is superior in terms of novelty, diversity, and coverage. Here, *novelty* measures how new or different products are recommended to a user, which is concretely determined by the product's *popularity* (Celma and Herrera 2008; Ziegler et al 2005) (that is, the more popular a product is, the less novel it would be to the user). The *popularity* is formally calculated as the number of reviews a product receives divided by the maximal number of reviews across all products. *Diversity* is computed as the average pairwise product dissimilarity in the recommendation list (Smyth and McClave 2001).

Summary of Section 4.1. This sub-branch of systems is summarized in Table 1. The summary suggests that a content-based recommending process with review-based term profiles can be more effective than rating-based CF methods with regard to novelty, diversity, and coverage. Furthermore, Garcia Esparza et al (2011)) claim that as the process of building users' term-based profiles is independent of the process used to create products' profiles, the user profile created from one source might be used to match products in another source, thus allowing for cross-domain recommendations. However, their evaluation did not empirically verify this suggestion. In addition, their method's advantage over the standard content-based approach is not identified, so it is unclear whether review-based term profiles would perform better than traditional profiles built with static descriptions of items.

4.2 Rating Profile - Inferring Ratings from Reviews

A well-known problem in collaborative filtering (CF) is the rating sparsity problem, which commonly occurs in domains where numerical ratings of items are difficult to collect, or user preferences for items are too complex to be expressed as ratings (Leung et al 2006). In this section, we survey approaches that aim to infer a user's overall preference for a product based on the opinions s/he expresses in the review, which can act as a *virtual rating* (Zhang et al 2013) (also called an *inferred rating*, *opinion rating*, or *text-based rating*) for a CF system. In the following, we introduce each type of system in terms of how it infers user ratings from reviews, and how recommendations are generated based on the inferred ratings.

Aggregating words' opinion strengths or sentiments. A previous study published in (Leung et al 2006) first applies Part-of-Speech tagging to extract adjectives and verbs as opinion words. Then, to determine the opinion's sentiment orientation, they assume that similar opinion words may not imply a similar sentiment orientation. For instance, the words "terrible" and "frightening" are synonyms in WordNet (Kamps et al 2004), but "frightening" appears less frequently in negative reviews, suggesting that people are more likely to use this word to describe a certain property of a movie (e.g., a horror movie), rather than to express a negative opinion. Based on this assumption, the sentiment orientation of an opinion word is concretely determined by its relative strength in a sentiment class c (i.e., positive or negative), which is

formally estimated according to the word's occurrence frequency in the reviews that belong to c. In the next step, rating inference, the strengths of all of the opinion words contained in a review are aggregated to determine the overall sentiment implied by the reviewer. The review is then assigned a corresponding overall rating (at 2-point or 3-point scale). Using the inferred overall ratings, the system runs the classical memory-based CF algorithm to generate recommendations.

In contrast, the work by Zhang et al (2013) aggregates the sentiments expressed in emoticons (such as smiley and sad faces) and opinion words to infer a review's overall sentiment. Concretely, each review r is first parsed into clauses using punctuation marks. Then, the sentiment score of clause c is calculated via $CS(c) = \sum S_w$, where $S_w = \frac{L_w^2}{L_{clause}} S_w^{\mathcal{W}} N_w$, in which L_w is the length, i.e., the number of characters, of the sentiment word w, L_{clause} is the length of the clause, $S_w^{\mathcal{W}}$ is the sentiment score of w in the sentiment word vocabulary $\mathcal{W}$, and N_w is a negation check coefficient. The sentiment score of a review r is subsequently defined as $RS(r) = \alpha RS^W(r) + (1-\alpha) RS^E(r)$, where $RS^W(r) = \sum_{c \in r} CS(c)$, $RS^E(r) = \sum_{e \in E_{sent} \cap e \in r} S_e^E$, in which S_e^E is the sentiment score of emoticon e in the emoticon set E_{sent}, and $\alpha \in [0, 1]$ is a parameter to control the relative contributions of the two sentiment components. As a result, each review can be classified as either *positive* or *negative*. During the recommendation process, a user-item rating matrix is constructed with virtual ratings (1 for *positive*, and -1 for *negative*) as derived from the reviews, which serves as the input for the standard user-based and item-based CF algorithms.

Evaluation. An experiment reported in (Leung et al 2006) demonstrates that their approach is capable of inferring users' ratings of movies by comparing them with user-specified real ratings (with the *MovieLens-100k* dataset that contains approximately 30k IMDb reviews provided by 1,065 users to 1,477 movies after filtering out users with less than 10 reviews). However, they do not measure the accuracy of their recommendation algorithm that is based on the inferred ratings.

Zhang et al (2013) use a dataset collected from *Youku* (a popular video-sharing website in China) that contains 120,174 reviews written by 6,450 users about 1,085 videos to prove that CF methods with virtual ratings are better than the non-personalized popularity-based approach with regard to *precision*. Moreover, they indicate that a user-based CF with virtual ratings outperforms an item-based CF with virtual ratings, even in a cold-start setting where users are with the least ratings. Another experiment using an *Amazon* dataset with 318,730 reviews of 1,805 books written by 5,502 users further verifies this finding. This experiment also tests the possibility of combining virtual ratings and user-specified real ratings by simply averaging them, and shows that the combination will likely improve the algorithm's accuracy, although this is not the focus of their work.

Classifying opinions via machine learning. Poirier et al (2010a,b) also attempt to derive overall ratings from reviews and then create a user-item rating matrix for enabling CF, but unlike the above-mentioned approaches, they use machine learning to classify the opinions. Specifically, they first represent each review as a vector of word frequencies. The review vectors in combination with user-specified ratings are then used to train a selective Naive Bayes classifier

on two sentiment classes: *positive* and *negative*. Afterwards, the trained classifier is applied to infer ratings from new reviews.

Evaluation. The ratings inferred by the above opinion classification method have been tested on a dataset from *Flixster* that includes 3,330,000 reviews written by almost 100,000 users about 10,500 movies (Poirier et al 2010b). The results demonstrate that the recommendation accuracy of item-based CF with inferred ratings is comparable to that of the standard item-based method with user-specified ratings, in terms of *Root Mean Square Error* (RMSE), which measures the square root of the mean square error between predicted and actual ratings (Shani and Gunawardana 2011).

Summary of Section 4.2. The above approaches use essentially the same procedure: first, they infer the overall opinions from reviews (via an aggregation approach (Leung et al 2006; Zhang et al 2013), or machine learning (Poirier et al 2010a,b)), then they convert the overall opinions into ratings (usually in the form of binary ratings), and run a memory-based CF technique. The experimental results from (Poirier et al 2010b) indicate that the accuracy of item-based CF, based on their inferred ratings, is comparable to the accuracy of traditional CF method based on real ratings. Zhang et al (2013) further find that a user-based CF with inferred ratings outperforms an item-based CF with those ratings. In summary (see Table 1), these studies validate the possibility of deriving ratings from reviews, thus enabling CF to address the rating sparsity problem.

4.3 Rating Profile - Enhancing Ratings with Reviews

In another sub-branch of research, investigators assume that both reviews and real ratings are available in a particular scenario, in which reviews act as an auxiliary resource to enhance ratings. Some of these approaches stress that using reviews can be helpful for dealing with the cold-start problem, since relying on only a few ratings provided by new users might prevent the recommender from returning satisfactory results (McAuley and Leskovec 2013; Seroussi et al 2011).

4.3.1 Considering Review Helpfulness

In Section 3, we mention that a review's helpfulness can be associated with its accompanying rating to indicate the rating's authenticity (Raghavan et al 2012). Specifically, for a review that receives a certain number of votes from readers (i.e., a vote denotes whether a reader found the review helpful or unhelpful), they compute a *quality score* for its accompanying star rating via $helpfulness = \frac{Number\ of\ helpful\ votes}{Total\ number\ of\ votes}$. If a review receives few votes, the quality score is estimated using a regression model that is trained with features extracted from both reviews that receive sufficient votes and the items' meta-data. Then, the quality scores are taken as weights assigned to the star ratings in a probabilistic matrix factorization (MF) framework for performing the rating prediction.

Table 1 Summary of Typical Works on Review-based User Profile Building (Part A: Term-based Profile & Inferred Rating Profile)

Citation	**Review element**	**User profile**	**Recommending method**	**Evaluation data condition**	**Tested products**	**Baseline**	**Evaluation metric**
Term-based user profile							
Garcia Esparza et al (2010, 2011)	Frequent words weighted by TF-IDF	Term-based user index	Content-based method	Normal	*Blippr* (movies, applications, books, games), *Flixter* (movies)	User-based and item-based CF	Novelty, diversity, coverage, precision, recall, F1
Inferred rating profile							
Leung et al (2006)	Feature opinions	Inferred ratings by aggregating words' opinion strengths	User-based CF	Normal	*MovieLens* (movies)	User-specified ratings	*NA*
Zhang et al (2013)	Sentiment words and emoticons	Inferred ratings by aggregating words' and emoticons' sentiments	User-based and item-based CF	1) Normal; 2) New users	*Youku* (videos), *Amazon* (books)	1) Popularity-based ranking; 2) CF with real ratings	Precision
Poirier et al (2010a,b)	Frequent words	Inferred ratings via opinion classification	Item-based CF	Normal	*Flixster* (movies)	Item-based CF with real ratings	RMSE

Evaluation. The above method that uses review helpfulness to enhance ratings has been tested on a benchmark *Amazon* dataset (Jindal and Liu 2008) with two product catalogs *books* and *audio CDs* (Raghavan et al 2012). By comparing it to the primitive probabilistic MF (Salakhutdinov and Mnih 2008) that does not involve quality scores, the authors show that the quality-aware probabilistic MF can achieve better performance with regard to RMSE.

4.3.2 Considering Review Topics

Weighting ratings with topic profile. The topics (i.e., aspects) that review writers discuss in reviews can also be used to weight ratings. For instance, Musat et al (2013) take the similarity between topics mentioned in a review of a candidate product and the topics appearing in the target user's topic profile as a weight assigned to the review's associated star rating. Specifically, the target user's topic profile is established via a frequency-based method in which the nouns in reviews with the highest opinion counts (which counts the number of attached opinions) are selected and manually grouped into topics (such as hotel location, cleanliness, room view, etc.). The topic profile is formally represented as $Z_i = \{z \mid count(z, R_i) > ts\}$, where $count(z, R_i)$ denotes the opinion count of topic z that appears in the user's written review set R_i, and ts is a threshold, which is set as zero in their experiment. For a review $r_{j,A}$ from the review set R_A of a candidate product A ($j \in 1, \ldots, |R_A|$), its relevance to the target user i is defined by $Z_{i,r_{j,A}}$ which consists of topics appearing in both the user profile Z_i and the review $r_{j,A}$. Then, a weighted average of the product's ratings is computed to indicate its potential interest to the user, for which $Z_{i,r_{j,A}}$ is the rating's weight. In addition, this work defines a parameter $\gamma(i, A)$ that represents the number of reviews with $\left|Z_{i,r_{j,A}}\right| \geq 3$. Only the products with $\gamma(i, A)$ greater than or equal to a minimum confidence threshold θ are calculated with the weighted score. Otherwise, a non-personalized approach is used to calculate the product's score, which simply averages all of the ratings that the product receives.

Evaluation. Musat et al (2013) use a dataset collected from *TripAdvisor* that includes 68,049 reviews of 216 hotels written by 59,067 users. They find that the approach using weighted ratings with topic profiles is better than non-personalized product ranking with respect to *Mean Absolute Error* (MAE), which computes the deviation between predicted ratings and actual ratings (Shani and Gunawardana 2011). It also performs better in terms of Kendall's tau rank correlation coefficient, which measures the fraction of pairs with the same order in both system's and user's rankings (Kendall 1938), for the pairs of items with large rating differences (i.e., much stronger preference for one item over another in the pair).

Associating review topics with latent factors. Instead of using review topics to weight ratings (Musat et al 2013), McAuley and Leskovec (2013) and Seroussi et al (2011) directly incorporate them into the latent factor model for rating prediction (see the background of this model in Section 2.2). In (McAuley and Leskovec 2013), the review topics help to uncover the relationship between users' implicit tastes and products' inherent properties. For example, when deciding whether to recommend a "Harry Potter" book to a user, it may be helpful to know that the book is about wizards, and that the target user has an interest in wizardry. Specifically, they develop a Hidden Factors as Topics (HFT) model for combining the latent factors learned from

item ratings with the latent topics learned from reviews, in which the log likelihood of latent topics acts as the regularizer in the objective function. In more detail, the authors first define the set of all reviews related to an item i as a document d_i, for which the topic distribution θ_i returned by the Latent Dirichlet Allocation (LDA) (Moshfeghi et al 2011) reveals the extent to which each of K topics is discussed across all of the reviews of that item. They then assume that if an item possesses a certain property (i.e., one with a high value $\gamma_{i,k}$ in its associated latent factors γ_i), it should correspond to a particular topic being discussed (i.e., one with a high value $\theta_{i,k}$ in the latent topic model θ_i). Note that the total number of latent factors is the same as the number of latent topics. HFT then links the two: the latent item factor $\gamma_{i,k}$ and the latent topic $\theta_{i,k}$, via the following transformation:

$$\theta_{i,k} = \frac{\exp(\kappa\gamma_{i,k})}{\sum_{k'} \exp(\kappa\gamma_{i,k'})} \tag{11}$$

where κ is a parameter to control the peakiness of this transformation; a large value of κ means that users only discuss the most important topics, whereas a small value of κ means that users discuss all of the topics evenly. In this way, the latent topics contained in reviews can be incorporated into the process of training the latent factor model. The trained model is then used to predict the rating of an item for the target user.

The review topics discovered by LDA can also be used to reflect users' latent attributes, such as their demographics (e.g., gender and age) and vocabulary use (Seroussi et al 2011). In this study, the Matrix Factorization with User Attributes (MFUA) model (also called attribute-based MF; see Section 2.2) is used to incorporate these latent attributes. In particular, to address the new user problem (i.e., users with few ratings), a switching strategy is proposed: if the target user submitted ratings less than n ($n = 2$ in their experiment), they adopt the attribute-based MF model; otherwise, the classical biased MF model is used. Formally, the predicted rating of an item i for the target user u is

$$\hat{r}_{u,i} = \begin{cases} \mu + b_i + \sum_{a=1}^{L} P(a|u)(b_a + y_a^T q_i) & \text{if } |R_u| < n \\ \mu + b_u + b_i + p_u^T q_i & \text{if } otherwise \end{cases} \tag{12}$$

where μ is the global rating mean, b_u, b_i, and b_a are, respectively, user, item and attribute biases, y_a is the a-th column vector of the attribute-factor matrix, p_u and q_i are the latent factors of user u and item i respectively, $P(a|u)$ is the probability that the user u has attribute a, and R_u is the user u's rating set.

Evaluation. Both studies (McAuley and Leskovec 2013; Seroussi et al 2011) prove that the developed latent factor model enhanced with review topics performs better than the standard model. In (McAuley and Leskovec 2013), a large-scale dataset consisting of 42 million reviews of 3 million items provided by 10 million users is used in the experiment; it covers 29 different product catalogs such as *books* and *movies* from *Amazon*, *beers* and *wines* from *Ratebeer* and *Beeradvocate*, and *restaurants* from *citysearch.com* and *Yelp*. The experimental results show that their developed Hidden Factors as Topics (HFT) model can achieve lower *Mean Absolute*

Error (MAE). They also test a variation of the HFT model that associates θ_u (the latent topic vector learned from the set of reviews written by the user u) with the user's latent factors γ_u, and find that the HFT model based on user topics exhibits a similar performance to the one based on item topics, implying that they are not substantially different. Moreover, they demonstrate the special advantage of HFT in terms of serving new users/items, which suggests that, given the same number of ratings, using the reviews can provide additional information about a user or product and can help solve the cold-start problem.

The experiment reported in (Seroussi et al 2011) presents similar findings. Its dataset is *IMDb-1M* (containing 22,116 users who submitted 204,809 posts and 66,816 reviews, after filtering out users who had not submitted any reviews). The implemented attribute-based MF model obtains higher accuracy than either the traditional MF or the non-personalized baseline in which only the global rating mean and the item bias are considered, with respect to *normalized Root Mean Square Error* (NRMSE), especially when the target users are new. The improvement is also larger than the one that is simply based on users' explicitly provided demographic attributes.

4.3.3 Considering Overall Opinions

As mentioned in Section 4.2, virtual ratings are the overall opinions inferred from reviews. In the previous section, we describe how the virtual ratings can be used to generate recommendations when real ratings are not available. In this section, we describe a representative system that uses inferred ratings to enhance user-specified real ratings in a situation where they both exist. Actually, the preliminary trial in (Zhang et al 2013) (in Section 4.2) already suggests that combining inferred ratings with real ratings is likely to return better recommendations, even if they are simply averaged.

Pero and Horváth (2013) investigate three ways of combining overall opinions with real ratings in a biased matrix factorization (MF) model (Koren et al 2009): 1) *opinion pre-filtering*, in which opinions are used to pre-process the training data (i.e., the ratings in the training set are modified such that they are closer to the overall opinions); 2) *opinion post-filtering*, in which ratings and overall opinions are first used independently to train two prediction models, and then a linear combination of the two models is realized to obtain the final rating prediction; and 3) *opinion modeling*, in which overall opinions are used *implicitly* in the training phase. That is, the rating matrix is factorized in the standard way, but a lower weight is assigned to the prediction error if the predicted rating $\hat{r}_{u,i}$ lies between the real rating r_{ui} and the overall opinion o_{ui}, or if it is equal to o_{ui}. To obtain the overall opinions, they apply the sentiment aggregation method. Each review c_{ui} written by user u for item i is treated as a sequence of words, i.e., $c_{ui} = (w_1, \ldots, w_n)$. If w_j is an adjective word or phrase, it is assigned a semantic orientation $s(w_j)$ based on the sentiment lexicon S constructed in (Liu 2010). The overall opinion of the review c_{ui} is then computed as $o_{ui} = \frac{\sum_{w_j \in c_{ui}} s(w_j)}{|\{w_j \in c_{ui} | w_j \in S\}|}$, which is normalized in the range $[-1, +1]$.

Evaluation. The three variations proposed in (Pero and Horváth 2013) have been tested on an *Amazon* dataset containing 5,838,898 ratings and reviews written by 2,146,275 users about

1,231,018 products (covering the catalogs of *movies*, *music*, and *books*). The results show that the *opinion post-filtering* method gives the best performance in terms of prediction accuracy RMSE, compared with the other two and with the standard biased-MF that only takes real ratings or overall opinions as input. In addition to the original, sparse dataset, they also test the algorithms on a smaller, much denser sample dataset, created by filtering out users who provided fewer than 50 or more than 500 reviews; this smaller dataset contains 4,654 users, 287,666 items, and 606,294 ratings/reviews. The results are similar, implying that their proposed approach works well in both sparse and dense data conditions.

4.3.4 Considering Review Contexts

The review contexts are the contextual information embedded in reviews (see Section 3). For example, in the review sentence "*This camera has very good picture quality at night*," "at night" is the context related to the feature "picture quality." In context-aware recommender systems, it is a challenge to acquire users' contexts (Adomavicius and Tuzhilin 2011). In this section, we show that reviews can address this issue by providing contextual information.

Using review contexts to predict an item's utility. Hariri et al (2011) assume that a user may give a hotel the same rating in different contexts, but the utility of selecting the hotel may vary across contexts. Therefore, unlike traditional systems that focus on rating predictions, they use review contexts to make *utility predictions*. First, they treat *trip type* as the key context that is pre-defined with five possible options: *family*, *couples*, *solo travel*, *business*, and *friends' get away*. They then use the labeled Latent Dirichlet Allocation (LDA) model (Ramage et al 2009), which is a supervised classification algorithm for multi-labeled text corpus based on topic modeling, to train a multi-class classifier that can determine the probability of each trip type being related to a review or the user's current query. The classifier is concretely trained on a set of reviews with users' explicitly specified trip types. The detected review context is then represented as a distribution over the five options. For example, if there are two trip types detected from a review for the hotel i: *solo travel* and *business*, the context is represented as $context_u^i = \{P\,(family) = 0, P\,(couples) = 0, P\,(solotravel) = 0.5, P\,(business) = 0.5, P\,(friends' getaway) = 0\}$, where u is the reviewer.

The utility of an item is defined by two factors: the *predicted rating*, calculated through the standard item-based kNN algorithm, and the *context score*, which measures the relevance of an item i to the target user u's current context. For the latter, an item-based collaborative filtering approach is applied to predict the context that u would assign to item i. The predicted context is then compared to the user's current context. The similarity between the two items i and j in terms of their contexts is computed via the *Cosine* metric:

$$contextualSimilarity(i,j) = \frac{\sum_u commonLabels(i,j)}{\sqrt{\sum_u |labels(i)| \times |labels(i)|}} \tag{13}$$

where $commonLabels(i,j)$ denotes the number of common trip types assigned to items i and j by the same set of users, and $labels(i)$ denotes the number of trip types assigned to i by all

users. Then, the predicted context for user u on item i is calculated as:

$$predictedContext(u,i) = \frac{\sum_{k \in Neighbors(i)} context_u^k \times contextualSimilarity(k,i)}{\sum_{k \in Neighbors(i)} |contextualSimilarity(k,i)|} \quad (14)$$

where $Neighbors(i)$ denotes the set of items that have the highest contextual similarity to item i, and $context_u^k$ is the context given by user u to item k ($k \in Neighbors(i)$), which can be detected from the user's review of k through the method described above. Subsequently, a context score is computed for item i to indicate its relevance to user u's current context:

$$contextScore(u,i) = \frac{IC_u \cdot PC_u^i}{||IC_u|| ||PC_u^i||} \quad (15)$$

where PC_u^i is the output of $predictedContext(u,i)$ and IC_u denotes u's current context. Finally, the utility score of item i for user u is calculated as:

$$utility(u,i) = \alpha \times predictedRating(u,i) + (1-\alpha) \times contextScore(u,i) \quad (16)$$

where α is a constant. The top-N items with the highest utility scores are then recommended to the user.

Evaluation. The authors test this approach in a relatively dense dataset, by removing items with less than five ratings from the *TripAdvisor* dataset, which originally contained 12,558 reviews of 8,941 hotels by 1,071 reviewers. Their results show that the proposed context-based utility prediction method outperforms the standard non-context based rating prediction using the item-based kNN algorithm, in terms of *Hit Ratio*, which returns the probability that the user's target choice is included in the top-N recommendation list.

Associating review contexts with latent factors. Another related work (Li et al 2010) uses the contextual information discovered from reviews to enhance the latent factor model. The authors first define four types of context for restaurants: 1) *time* of dining; 2) *occasion* (e.g., birthday or anniversary); 3) *location* (city the meal is eaten in); and 4) *companion* (the accompanying person(s)). The *time* and *occasion* contexts are extracted from reviews through a string matching method. The *companion* context is also obtained from reviews using a hybrid classifier that first identifies related features via rule-based methods and then uses the features to train a Maximum Entropy (MaxEnt) classifier (Ratnaparkhi 1998). (Note that this hybrid classifier gives a better performance than the rule-based methods or MaxEnt alone.) The *location* context is acquired from the user's profile, as it rarely exists in the review text. This work postulates that a user's interest in an item is influenced by two factors: the user's *long-term preference*, which can be learnt from the user's rating history; and the *current context*. To capture the two factors simultaneously, a probabilistic latent relational model (PLRM) is developed by which the rating $y_{i,j,\boldsymbol{c}}$ on item j is predicted for user i under context $\boldsymbol{c} = (c_1, \ldots, c_k)$ (e.g., component c_i = "dinner time") is modeled as a Gaussian distribution with mean $\mu_{i,j,\boldsymbol{c}}$ and

variance $\frac{1}{\lambda^{(y)}}$:

$$y_{i,j,\boldsymbol{c}} \sim \mathcal{N}(\mu_{i,j,\boldsymbol{c}}, \frac{1}{\lambda^{(y)}}) \tag{17}$$

$$\mu_{i,j,\boldsymbol{c}} = \boldsymbol{u}_i^T A \boldsymbol{v}_j + (W_u f_i)^T (W_v f_j) \tag{18}$$

where $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ represent the latent factors of user i and item j respectively (as learnt from the rating data), W_u and W_v represent the feature transformation matrices for users and items respectively, and f_i and f_j respectively denote the observed feature vectors of user i and item j, where each component is a type of feature such as "gender of the user" or "price range of the restaurant". In the above formula, $\boldsymbol{u}_i^T A \boldsymbol{v}_j$ is the estimation based on the user's long-term preference, where A models the interaction between $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$, and the second term $(W_u f_i)^T (W_v f_j)$ is the estimation based on the context and the observed features of users and items. (Note that the context $\boldsymbol{c}$ is integrated into the user's observed features f_i.) The model's parameters are learned through a modified expectation maximization (EM) algorithm.

Evaluation. This context-based latent factor model has been tested on a dataset containing 12,533 restaurants with 756,031 reviews from 82,892 users (Li et al 2010). It is compared with two baseline methods: the application of PLRM to a standard CF setting that only considers rating information (denoted as *Noncontext*); and, based on Boolean model, first filters out items irrelevant to the user's current context, and then predicts the remaining items' ratings using *Noncontext*. The results show that the method achieves better performance for top-N recommendations, but not for rating predictions, because although a user's selection process might be influenced by the context, how a user rates an item after selecting it is not strongly related to the context. For example, a user may in general prefer to eat breakfast in a cafeteria, but his rating of a particular cafeteria is based on the quality of the service, food, price, and environment. This explanation is consistent with the reason given for using review contexts to make utility predictions rather than rating predictions in (Hariri et al 2011).

4.3.5 Considering Review Emotions

The emotional features of reviews (such as sadness, fear, distress, anger, happiness, etc.), which reflect reviewers' moods and attitudes when they were writing the reviews, can also be used to learn reviewers' preferences. Moshfeghi et al (2011) attempt to use the emotions expressed in reviews to predict the probability of a user liking an item. They first represent each reviewed movie with three types of feature space: *movie space*, in which the movie itself is treated as a feature; *semantic space*, which contains three sub-spaces actor, director, and genre; and *emotion space*, which is assessed using movie reviews and plot summaries. To extract the emotional features, they use the OCC model proposed in (Ortony et al 1990) to define emotions, and the emotion classifier constructed in (Shaikh et al 2009) to determine whether a certain type of emotion occurs in a text. Then, given a feature space s and a user u (with her/his past ratings), the probability that a user likes a movie m is represented as $P(+ \mid m, u, s)$, where $+$ denotes whether the user u likes the movie m. This probability is estimated by aggregating

the probabilities over a set of features that are included in the feature space s: $P(+ \mid f, u, s)$, where f is one particular feature (like the movie's actor included in the *semantic space*). To compute $P(+ \mid f, u, s)$ (i.e., the probability that a movie is liked or disliked because of a feature), they extend the Latent Dirichlet Allocation (LDA) model for defining the user as a probability distribution over a set of latent groups and each group in turn as a probability distribution over the movies that are liked by the group's users. $P(+ \mid f, u, s)$ is then computed by marginalizing over these latent groups. Finally, to obtain the final prediction, they adopt a standard machine learning technique, the gradient boosted tree (Friedman 2000), to combine the predictions made about the three types of feature spaces.

Evaluation. This emotion-based method has been tested on two datasets (Moshfeghi et al 2011): a *Movielens-100K* dataset that contains 100,000 ratings for 1,682 movies from 943 users, and a *Movielens-1M* dataset with 1 million ratings for 3,900 movies from 6,040 users. The reviews are extracted from *IMDb*. Two related approaches are compared: 1) the approach based on the original LDA model that only considers *movie space*; and 2) the nonparametric probabilistic principal component analysis (NPCA) method introduced in (Yu et al 2009). The results show that the proposed method combining three types of feature spaces outperforms the other methods, even in sparse data conditions, with regard to *Mean Squared Error* (MSE). Furthermore, *movie* and *actor* feature spaces perform well in predicting the top ranked items (with higher *Mean Average Precision* (MAP)), whereas the models based on *emotion space* and to some extent *genre sub-space* are better at predicting ratings. The model using *semantic space* is superior to the one using *emotion space* in low sparsity situations, indicating that emotions might not be important when the data already contain more direct indicators such as actors, directors, and genres. However, the *emotion space*, especially that constructed from reviews, is shown to be helpful in a high data-sparsity scenario.

Summary of Section 4.3. The state-of-the-art systems surveyed in this section all indicate that fusing reviews with ratings improves the accuracy of recommender systems relative to the standard approaches that consider ratings alone (see summary in Table 2). The reviews are able to enhance ratings in several ways: they determine the rating's quality score (Raghavan et al 2012); enrich the latent factor model with review topics, overall opinions, or review contexts (Li et al 2010; McAuley and Leskovec 2013; Pero and Horváth 2013; Seroussi et al 2011); personalize products' ranking with the user's topic profile (Musat et al 2013); predict the item's utility by involving contextual relevance (Hariri et al 2011); and construct the item's feature space with review emotions (Moshfeghi et al 2011).

More notably, some studies have empirically proven that their approaches help new users (those with few ratings) by using review topics (McAuley and Leskovec 2013; Seroussi et al 2011), or deal with sparse data situations by using overall opinions or review emotions (Moshfeghi et al 2011; Pero and Horváth 2013). Thus, reviews can be useful for complementing sparse ratings by offering additional preference information. Another interesting observation is that most works are based on the latent factor model (Li et al 2010; McAuley and Leskovec 2013; Pero and Horváth 2013; Raghavan et al 2012; Seroussi et al 2011), demonstrating the model's flexibility in terms of incorporating additional review elements.

Table 2 Summary of Typical Works on Review-based User Profile Building (Part B: Enhanced Rating Profile)

Citation	Review element	User profile	Recommending method	Evaluation data condition	Tested products	Baseline	Evaluation metric
Ratings enhanced with review helpfulness							
Raghavan et al (2012)	Review helpfulness (votes by readers)	Rating weighted by quality score	Quality-aware probabilistic MF (PMF)	Normal	*Amazon* (books, audio CDs)	Primitive PMF	RMSE
Ratings enhanced with review topics							
Musat et al (2013)	Frequent topics based on opinion count	Topic profile	Weighted average of product's ratings for ranking	Normal	*TripAdvisor* (hotels)	Non-personalized ranking	MAE, Kendall's tau rank coeffecient
McAuley and Leskovec (2013)	Latent topics by LDA	Latent factors associated with latent topics	Hidden Factors as Topics (HFT)	1) Normal; 2) New users/items	*Amazon* (books, movies), *Ratebeer* and *Beeradvocate* (beers, wines), *citysearch.com* and *Yelp* (restaurants), etc.	Standard latent factor model	MAE
Seroussi et al (2011)	Latent attributes by LDA	Latent factors with attributes	Switching between attribute-based MF and biased MF	1) Normal; 2) New users	*IMDb* (movies)	Standard MF	Normalized RMSE
Ratings enhanced with overall opinions							
Pero and Horváth (2013)	Sentiment words	Real ratings and inferred opinion ratings (by aggregating words' sentiments)	Biased MF	1) Normal; 2) Sparse ratings	*Amazon* (movies, music, books)	Biased MF with real ratings	RMSE
Ratings enhanced with review contexts							
Hariri et al (2011)	Review context *trip type* via multi-class classifier	Ratings and context	Context-aware utility prediction	Normal	*TripAdvisor* (hotels)	Standard item-based CF	Hit Ratio
Li et al (2010)	Review contexts *time* and *occasion* by string matching, and *companion* by rule-based method and Maximum Entropy classifier	Latent factors with ratings and contexts	Probabilistic latent relational model (PLRM)	Normal	restaurants	PLRM with ratings	Top-N recommendation
Ratings enhanced with review emotions							
Moshfeghi et al (2011)	Review emotions via emotion classifier	Ratings and three feature spaces for products (movie space, semantic space, and emotion space)	Extended LDA and gradient boosted tree	1) Normal; 2) Sparse ratings	*MovieLens* (movies)	Original LDA with movie space	MSE

4.4 Feature Preference

The common motivation behind this category of approaches is that a user's preference can be *explicitly* characterized by multiple features, or upper-level aspects, based on her/his opinions expressed in reviews. In comparison to rating profiles which only indicate *how much* a user likes an item, a feature-based preference model can tell *why* a user likes (or dislikes) an item.

4.4.1 Incorporating Aspect Opinions

As mentioned in Section 3, we can obtain feature opinions from reviews, by first extracting features using a statistics or model based method and then identifying associated opinions based on word distance or pattern mining. The extracted features can further be mapped to upper-level aspect representations. Alternatively, we can apply a LDA or SVM based classifier to directly locate the aspect-level opinions (in the form of ⟨aspect, sentiment⟩ pairs).

Associating aspect opinions with latent factors. Some studies model the aspect opinions using latent factors. For instance, Wang et al (2012) establish a 3-dimensional tensor model $\mathcal{R}$ to uncover the complex relationships among users, items, and aspects. The tensor model is an extension of matrix factorization (MF), which preserves the data's multi-dimensional nature and determines the latent factors for each dimension. More formally, in (Wang et al 2012), it is defined with the size $I \times J \times (K+1)$, where I, J and K denote the numbers of users, items, and aspects, respectively. To obtain the aspects' opinion ratings, they adopt a semi-supervised method called double propagation (Qiu et al 2011) to expand opinion words and extract aspect terms. The Latent Dirichlet Allocation (LDA) model is applied to cluster the aspect terms into aspects. The user's opinion rating on one aspect is determined via: $NumberofPositiveOpinionWords/TotalNumberofOpinionWords$.

Then, a decomposition method, called CP Weighted OPTimization (CP-WOPT) (Acar et al 2011), is applied to decompose the high-order tensor into a sum of rank-one tensors: A (with size $I \times R$) for users, B (with size $J \times R$) for items, and C (with size $(K+1) \times R$)) for aspects (note that the first entry in C is the overall rating, and the others are the K aspects' opinion ratings), such that

$$r_{i,j,k} = \sum_{r=1}^{R} a_{i,r} b_{j,r} c_{k,r} \tag{19}$$

for all $i = 1, \ldots, I$, $j = 1, \ldots, J$, and $k = 1, \ldots, K+1$, where $r_{i,j,k}$ is the entry of the tensor model $\mathcal{R}$, and R is $\mathcal{R}$'s rank. They then consider the CP decomposition as a weighted least squares problem and minimize the following objective function.

$$f_{\mathcal{W}}(A, B, C) = \frac{1}{2} \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K+1} \{ w_{i,j,k} \left(r_{i,j,k} - \sum_{r=1}^{R} a_{i,r} b_{j,r} c_{k,r} \right) \}^2 \tag{20}$$

where $\mathcal{W}$ is a non-negative weight tensor (with the same size as $\mathcal{R}$) defined as

$$w_{i,j,k} = \begin{cases} 1 \text{ if } r_{i,j,k} \text{ is known} \\ 0 \text{ if } r_{i,j,k} \text{ is unknown} \end{cases} \quad (21)$$

The gradient descent method is used to find the factor matrices A, B, and C. The rating predicted for an unknown item m_j for user u_i is then calculated as: $\hat{r}_{i,j} = \sum_{r=1}^{R} a_{i,r} b_{j,r} c_{1,r}$, where $c_{1,r}$ denotes the overall rating.

In contrast, Jakob et al (2009) attempt to incorporate more types of information, such as the number of opinions, into the MF-based model. They develop a multi-relational matrix factorization (MRMF) model (Lippert et al 2008) to capture the five types of relations among users, movies, and users' opinions about particular movie aspects: 1) the relation matrix $rates \in \mathbb{R}^{u \times m}$ contains users' star ratings for movies, where u is the number of users and m is the number of movies; 2) the relation matrix $hasOp_N \in \mathbb{R}^{u \times m}$ contains the opinion ratings for movie aspects (the aspect is called "opinion type" in this work), where N is the number of aspects; 3) the relation matrix $has \in \{0,1\}^{m \times g}$ maps movies onto genres, where g is the number of genres; 4) the relation matrix $hasCount_N \in \{0,1\}^{u \times c}$ encodes whether the user expressed c times of an opinion about an aspect in her/his written reviews; and 5) the relation matrix $sim_N \in \mathbb{R}^{u \times u}$ maps the similarity between users in their different roles (i.e., the rating role and N roles of commenting on N different aspects). The involved entities and relations are then treated as feature vectors for running the MRMF algorithm (Lippert et al 2008), by which the matrix related to each entity is trained under the influences of multiple relations. Similar to (Wang et al 2012), they also apply LDA to identify aspects in reviews. A subjectivity clue lexicon (Wilson et al 2005) is used to extract opinion-bearing words, which are then linked to movie aspects by checking grammatical dependency. The opinion rating for one aspect is concretely obtained by aggregating all of the sentiments of opinion words that are linked to that aspect.

Evaluation. The systems of (Wang et al 2012) and (Jakob et al 2009) have both been tested on the *IMDb* movie dataset. In (Wang et al 2012), the dataset consists of 946 users, 1,525 items, and 53,353 associated reviews; in (Jakob et al 2009), the dataset contains 509 users, 2,731 items, and 53,112 associated reviews. Compared with the traditional MF approach that only considers overall ratings (Takács et al 2007) or the classical MRMF model that combines overall ratings and movie genres only (Lippert et al 2008), their proposed models are more accurate at generating recommendations, as measured by the RMSE.

Moreover, Wang et al (2012) show that incorporating aspect opinions can be more effective than incorporating review emotions (Moshfeghi et al 2011) (that is introduced in Section 4.3.5). They also test the algorithm in a sparse dataset by randomly removing some ratings, and find that its accuracy decreases less as the data becomes sparser.

Clustering users with aspect opinions. Ganu et al (2013) propose a different method for incorporating aspect opinions. They use them to perform a clustering-oriented user-based CF. They first build a multi-label text classifier based on the Support Vector Machine (SVM) to classify review sentences into different aspects (called topics in their study) such as a restaurant's food, service, price, ambience, anecdotes, and so forth, and sentiment categories includ-

ing positive, negative, neutral, and conflict. The reviewer's profile for a particular item is then constructed with a set of ⟨aspect, sentiment⟩ pairs. A normalized score for each pair ⟨aspect, sentiment⟩ (e.g., $\langle food, positive \rangle$) is generated by calculating the percentage of sentences that are classified into that pair among all of the sentences in the review. All of the pairs' scores are then used to establish a reviewer-item-aspect matrix, and a soft clustering algorithm called the iterative Information Bottleneck (iIB) (Slonim 2002; Tishby et al 1999) is applied to cluster the reviewers. After the clustering, each reviewer is assigned a probability vector in which each entry indicates the degree to which the user belongs to a cluster. During the recommendation process, the predicted rating of an item R_t for the user U_t is calculated as

$$Pr(U_t, R_t) = \frac{\sum_{i=1}^{m} U_t(c_i) \times Contribution(c_i, R_t)}{\sum_{i=1}^{m} U_t(c_i)} \tag{22}$$

where $U_t(c_i)$ denotes the probability that the user U_t belongs to cluster c_i, and m denotes the total number of clusters (which is fixed as 300 in their experiment). $Contribution(c_i, R_t)$ represents the contribution from cluster c_i, which is computed as follows:

$$Contribution(c_i, R_t) = \frac{\sum_{j=1}^{n} U_j(c_i) \times rating(U_j, R_t)}{\sum_{j=1}^{n} U_j(c_i)} \tag{23}$$

where $rating(U_j, R_t)$ refers to the rating given by user U_j to item R_t.

Evaluation. This clustering-oriented CF is compared with two baselines in (Ganu et al 2013): the standard matrix factorization (MF) based method, and the standard user-based kNN algorithm. These two baselines both take the overall opinion ratings as inputs. To derive the overall opinion ratings from reviews, they first perform a multivariate regression for learning the weights associated with the ⟨aspect, sentiment⟩ pairs. With the trained model, the overall rating can then be inferred. Their experiment uses a dataset collected from *New York Citysearch* that contains 31,814 users, 5,531 restaurants, and 51,162 associated reviews. The results show that their method is more accurate in terms of RMSE. Therefore, this study implies that fine-grained aspect opinions can have a more positive effect than overall opinions on improving rating prediction accuracy.

4.4.2 Deriving Users' Weight Preferences

Another sub-branch of research uses reviews to learn the user's weight preference (i.e., the weights s/he places on different features), rather than directly incorporating aspect opinions into the recommending process. The preference can be formally represented as a vector $\mathbf{u}_i = \{w_{i1}, w_{i2}, \ldots, w_{iL}\}$, where w_{ij} denotes feature f_j's relative importance (i.e., weight) to user u_i, and L is the total number of features. Once such preference is learnt, it can be used to predict an item's satisfaction degree.

(Liu et al 2013) and (Chen and Wang 2013) are two representative studies in this regard. In (Liu et al 2013), the weight w_{ij} is determined by two factors: how much the user cares about

the feature (i.e., *concern*) and her/his *requirement* for it, as follows:

$$w_{ij} = concern(u_i, f_j) \times requirement(u_i, f_j) \tag{24}$$

If user u_i commented on feature f_j very frequently in her/his review(s), but other users commented on it less often, $concern(u_i, f_j)$ increases. For $requirement(u_i, f_j)$, if user u_i frequently rates f_j lower than other users across different items, its value is higher. To obtain ⟨*user, item, feature opinion pair, and polarity*⟩ for the above formula, they develop an adverb-based opinion-feature extraction method that can accommodate the characteristics of Chinese reviews. Specifically, they first extract opinion words from reviews according to domain-independent adverbs, and then find features that are close to the opinion words. This process is repeated until the opinion and feature sets do not expand. To generate recommendations, they compute a satisfaction degree for item r_k based on the derived user preference $\mathbf{u}_i$:

$$Satisfaction(\mathbf{u}_i, r_k) = \frac{w_{i1} \times S_{1k} + w_{i2} \times S_{2k} + \ldots + w_{iL} \times S_{Lk}}{\sum_{j=1}^{L} w_{ij}} \tag{25}$$

where S_{jk} is the average of reviewers' opinions about feature f_j of item r_k. The top-N items with the highest satisfaction degrees are then recommended to the user.

Liu et al (2013) assume that each user provides a certain number of reviews (>= 5 in their experiment), so the user's weight preference is derived purely from her/his own reviews. In comparison, Chen and Wang (2013) focus on situations with sparse data due to scanty reviews supplied by each user. They claim that clustering can help to solve this issue. That is, if the number of reviews produced by a user is not sufficient to infer her/his weight preference, they first derive the *cluster-level preference* that denotes a group of users' common preference and then use it to refine the *reviewer-level preference* established for each user. The refined preference can in turn help to adjust the cluster-level preference, and the process continues until the two types of preference are stable. They extend the latent class regression model (LCRM) (Wedel and Kamakura 2000) to achieve this goal, as it considers inherent preference homogeneity among users. The inputs to the LCRM are the extracted ⟨feature, opinion⟩ pairs from reviews. WordNet (Fellbaum 1998) is used to group synonymous features, and SentiWordNet (Esuli and Sebastiani 2006) is used to quantify the opinion's polarity strength. During the recommendation process, all of the users are first clustered according to their cluster-level preferences, and then the user-based k-NN is applied within the cluster to which the target user belongs.

Evaluation. Both (Liu et al 2013) and (Chen and Wang 2013) show that deriving users' weight preferences from reviews can improve a system's ability to understand users' interests and achieve higher recommendation accuracy relative to the methods that are based on overall ratings. Specifically, Liu et al (2013) compare the proposed method to three baselines: the user-based and item-based CF techniques for which the overall rating is obtained by averaging opinion scores on several main features, and the classical item-based multi-criteria CF (Tang and McCalla 2009). A review dataset collected from a Chinese restaurant website is used for the comparison. After removing users who commented on fewer than five restaurants, the sample dataset consists of 35,027 reviews of 3,094 restaurants by 1,707 users. The comparison shows

that their method has the lowest prediction error in terms of *Mean Absolute Error* (MAE), whereas the item-based CF has the highest prediction error. Moreover, the advantage over the item-based multi-criteria CF suggests that their method that considers weight preference can perform better than one that directly incorporates aspect opinions.

To show that the clustering process can address the review sparsity issue, Chen and Wang (2013) compare their method with an approach that uses a probabilistic regression model (PRM) to infer *reviewer-level preference* purely from a user's own reviews. The assumption behind the PRM is that the overall rating that each reviewer assigns to a product can be considered as the *weighted* sum of her/his opinions about different features. Their experiment uses a dataset of electronic products, as these items usually have only a few reviews provided by each customer. The dataset includes 122 digital cameras with 18,251 associated reviews and 155 laptops with 6,024 reviews. The results show that their clustering-based approach outperforms the PRM-based approach in terms of both *Hit Ratio* and *Mean Reciprocal Rank* (MRR). (MRR measures the ranking position of a user's target choice in the list of recommendations.)

Relating weight preference to context. The weight a user places on a feature can be different under different contexts. For example, "atmosphere" might be a concern to a user having a meal with colleagues, but it would become less important than "food" when he is with his family. Therefore, some studies aim to relate users' weight preferences to their current contexts when recommending items. For example, the core idea of (Levi et al 2012) is to consider the relevance of an item's features to the target user's contexts (such as *trip intent* and *nationality* in the hotel domain), when matching the item to her/his weight preference. They first use the method introduced in (Hu and Liu 2004b) to extract frequent nouns and noun phrases to identify hotel features from reviews. Then, each extracted feature f is assigned three weights. The weights related to contexts *trip intent* and *nationality*, i.e., $W^f_{u_p}$ and $W^f_{u_n}$, are respectively calculated based on the frequency with which feature f occurs in reviews written by users with the same trip intent p or nationality n as the target user. The third weight $W^f_{u_{pref}}$ is the feature f's weight to the target user, obtained by asking the user to explicitly specify a weight for the aspect that the feature is associated with (for example, the feature "train" is associated with the aspect "location"). Then, the final weight W^f_u assigned to feature f is the product of these three weights:

$$W^f_u = W^f_{u_p} \cdot W^f_{u_n} \cdot W^f_{u_{pref}} \tag{26}$$

Finally, a candidate hotel h is assigned a score by adding the average of its reviews' scores and a bias adjustment (that captures the bias of a user with trip intent p and nationality n and the hotel bias h), for which each review's score is calculated as

$$score(v, u) = \sum_{s \in v} \sum_{f \in s} W^f_u \cdot score(f, s) \tag{27}$$

where u is the target user and $score(f, s)$ gives the sentiment of feature f in sentence s that belongs to review v of hotel h. The hotels with the highest scores are then recommended to the target user.

In (Chen and Chen 2014), the relationship between weight preference and context is directly extracted from users' reviews by considering the co-occurrence of aspect opinions and review contexts. Specifically, they detect two types of preference from reviews: *context-independent preference*, which is relatively less sensitive to contextual changes and reflects users' stable requirements over time; and *context-dependent preference*, which refers to aspect-level contextual needs that are common to users who are under the same context.

The *context-independent preference* is learnt purely from overall ratings and aspect opinions. A linear least squares regression model is constructed in which the overall rating that a user assigns to an item is the interaction function of the weighted aspect opinion ratings that are derived from her/his review. They then apply a statistical t-test to select weights that pass the significance level, and regard these aspects' weights as the user's context-independent preference. For deriving the *context-dependent preference*, they first perform a contextual review analysis based on keyword matching and rule-based reasoning for mining contextual opinion tuples. The tuple is formally denoted as $< i, rev_{u,i}, a_k, Con_{i,k} > (1 \leq k \leq K)$, indicating the user u's opinion a_k about the aspect k of item i under contexts $Con_{i,k}$ as 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 occurs and 0 otherwise. An aspect might appear in different opinion tuples related to different contexts. For instance, the aspect "atmosphere" of a restaurant is contained in two tuples $< i, rev_{u,i}, a_{atmosphere} = 1, Con_{i,atmosphere} =$ "$colleague$" $>$ and $< i, rev_{u,i}, a_{atmosphere} = -1, Con_{i,atmosphere} =$ "$family$" $>$, which are associated with two opposite opinions (1 for " positive" and -1 for "negative") in the two different contexts "*colleague*" and "*family.*" They then propose three variations of the contextual weighting method based on different text feature selection strategies: *mutual information*, *information gain*, and *Chi-square statistic*. The common characteristic of these algorithms is that they all consider the aspect-related terms' relevance to context, but they vary in terms of the technique used to infer the terms' weights. After the terms' weights are obtained, they incorporate them into the calculation of the aspect's frequency under a certain context, which is further used to determine the user's contextual weight placed on that aspect through the method proposed in (Levi et al 2012).

During the recommendation process, the two types of preference, *context-independent preference* and *context-dependent preference*, are combined to compute an item's matching score. Similar to (Levi et al 2012), they first calculate a score for each review of the item and then average all of the review scores. The top-N items with the highest matching scores are finally retrieved and recommended to the target user.

Evaluation. The procedure for generating recommendations in the two studies (Chen and Chen 2014; Levi et al 2012) is basically the same. Both are grounded in preference-based product ranking (see Section 2.3), but they differ in terms of the way they attain users' weight preferences. In (Levi et al 2012), the preference is stated by the current user, but adjusted by considering the relevance of the feature to the user's contexts. This approach can be applied to new users who have few or no reviews in the system. In contrast, Chen and Chen (2014) can directly infer the context-dependent weight preference from a user's written reviews if the user

has provided a sufficient number of reviews to enable this inference process. Therefore, these two methods have been tested in different conditions.

Levi et al (2012) conduct a user study (with 150 evaluations) to measure new users' satisfaction with their recommendation system. The list of recommendations tested in the experiment includes both hotels returned by their proposed method, and the most popular hotels (i.e., those with the highest overall ratings) from *TripAdvisor* and *Venere*. Users were asked to answer questions like *"Would you stay in this hotel?"* to indicate their satisfaction. The results show that users' average satisfaction with the recommendations generated by their method is higher than the satisfaction with the most popular items. The algorithm in (Chen and Chen 2014) has been tested on two restaurant datasets, one from *TripAdvisor* containing 121,932 reviews of 15,315 restaurants by 6,203 users, and the second from *Yelp* containing 125,286 reviews of 10,581 restaurants by 3,969 users. They find that, compared with the baseline method that does not consider users' context-dependent preferences, the three variations of their approach are all significantly better with respect to *Hit Ratio* and *Mean Reciprocal Rank* (MRR). Among the three variations, the method based on the Chi-square statistic is the best, followed by the method based on information gain.

4.4.3 Deriving Users' Attribute Value Preferences

For products that can be described by a set of static attributes (for example, a camera's price, weight, optical zoom, etc.), the user's *value preference* for an attributes might also be derived from her/his reviews. According to multi-attribute utility theory (MAUT) (Keeney and Raiffa 1976), the value preference can be formally represented as a value function $V(a_i)$ where a_i is the *i-th* attribute (see Section 2.3). Wang et al (2013) attempt to derive such preference in the form of a tuple $< attribute, opinion, specification >$ as extracted from reviews, where *opinion* is the sentiment the user expresses about the feature(s) that is mapped to the *attribute*, and *specification* is the attribute's specification value. For example, $<$ "*weight*", 1,200g$>$ denotes that the reviewer expressed a *positive* opinion ('1') about the camera's *weight* that is of static value $200g$. Specifically, they first adopt the method proposed in (Chen and Wang 2013) (see its introduction in Section 4.4.2) to identify the reviewer's opinions about features. The feature-level opinions are then linked to the product's attribute specifications. The derived value preferences are mainly used to handle "new users" in their work. Suppose that for a new user u, the system can initially elicit her/his value preference on site by asking her/him to specify the criteria for the attributes. However, considering that the elicited preferences are usually incomplete due to the user's unfamiliarity with the complex product domains (such as high-cost electronic products) (Payne et al 1993; Pu and Chen 2005), they propose using the derived reviewers' value preferences to help complete the new user's value preference as follows:

$$\overrightarrow{\tilde{\phi}_{ua}} = \begin{cases} (\overrightarrow{\phi_{ua}} + \sum_{\tilde{u} \in N_u} \bar{s}_{u\tilde{u}} \overrightarrow{\tilde{\phi}_{\tilde{u}a}})/2 & \text{if } \phi_{ua} \text{ is not missing} \\ \sum_{\tilde{u} \in N_u} \bar{s}_{u\tilde{u}} \overrightarrow{\tilde{\phi}_{\tilde{u}a}} & \text{otherwise} \end{cases} \tag{28}$$

where $\overrightarrow{\phi_{ua}}$ and $\overrightarrow{\tilde{\phi}_{\tilde{u}a}}$ are respectively the vector representations of the new user u's and the reviewer $\tilde{u}$'s value preference for attribute a; $\bar{s}_{u\tilde{u}} = s_{u\tilde{u}}/(\sum_{\tilde{u}\in N_u} s_{u\tilde{u}})$ is the normalized similarity between u and $\tilde{u}$; and N_u is the set of reviewers whose preferences are above a pre-defined threshold of similarity to the new user's currently stated preference. Thus, it can be seen that the new user's preference for one attribute a can be either completed (if s/he did not specify it) or adjusted (if s/he already specified it) by taking similar reviewers' value preferences into account. After estimating the new user's preference over all of the major attributes, a candidate product p's matching score is computed as

$$M_{up} = \frac{1}{k}\sum_{a=1}^{k} match_w\left(\bar{\phi}_{ua}, x_{pa}\right) \tag{29}$$

where $match_w\left(\bar{\phi}_{ua}, x_{pa}\right) = \langle\overrightarrow{\bar{\phi}_{ua}}, \overrightarrow{x_{pa}}\rangle$ (the inner product of the user's preference vector and the product's value vector w.r.t. attribute a), and k is the total number of attributes. The top-N products with the highest matching scores are then recommended to the user.

Evaluation. They test the proposed method on a set of user logs that contain 57 real users' records with their full preferences for all attributes and target choices among a catalog of 64 digital cameras and the products' 4,904 reviews crawled from *Amazon*. A subset of each user's full preferences is randomly retrieved (respectively over 2, 4, and 6 of the full size 8 attributes) to simulate the incomplete preference stated by a "new user." The proposed method (called *CompleteRank*) is compared with *PartialRank*, which considers only the user's stated preference for ranking products. The experimental results show that *CompleteRank* is more accurate than *PartialRank* in terms of *Hit Ratio*.

Summary of Section 4.4. In this section, we have seen that valuable information hidden in reviews can be used to construct user preference at the feature (or aspect) level. There are three related sub-branches of research (see summary in Table 3). The first sub-branch shows that systems incorporating aspect opinions (i.e., ⟨aspect, sentiment⟩ pairs) achieve better prediction accuracy than the ones that only consider overall ratings (Ganu et al 2013; Jakob et al 2009; Wang et al 2012). In addition, Wang et al (2012) show the superiority of aspect opinions over review emotions, and the merit of their developed tensor model for scenarios with sparse rating data.

The second sub-branch derives users' weight preferences (in the form of ⟨feature/aspect, weight⟩) from reviews and uses them to activate a preference-based product ranking. Liu et al (2013) show that ranking based on derived weight preferences can be more effective than rating-based CF. Their experimental findings also show that deriving users' weight preferences achieves better ranking accuracy than considering only aspect opinions. Chen and Wang (2013) address the situation of scanty reviews and propose a clustering-driven preference inference approach. They demonstrate that this approach can result in more accurate recommendations than an approach that relies purely on a user's own reviews to derive her/his weight preference. Levi et al (2012) and Chen and Chen (2014) go further, and relate weight preference to user contexts. Chen and Chen (2014) correlate aspect opinions with review contexts to derive users' context-

Table 3 Summary of Typical Works on Review-based User Profile Building (Part C: Feature Preference)

Citation	Review element	User profile	Recommending method	Evaluation data condition	Tested products	Baseline	Evaluation metric
Aspect opinions							
Wang et al (2012)	Aspect-level opinions	Latent factors based on ratings and aspect opinions	Tensor model	1) Normal; 2) Sparse ratings	*IMDb* (movies)	1) Standard MF; 2) (Moshfeghi et al 2011) based on review emotions	RMSE
Jakob et al (2009)	Aspect-level opinions	Ratings, aspect opinions, and number of opinions	Multi-relational MF (MRMF)	Normal	*IMDb* (movies)	Standard MRMF with ratings and genre info only	RMSE
Ganu et al (2013)	Aspect-level opinions	Ratings and aspect opinions	Soft clustering and user-based CF	Normal	*New York Citysearch* (restaurants)	1) Standard MF; 2) User-based CF with inferred opinion ratings	RMSE
Derived weight preference							
Liu et al (2013)	Feature opinions	Weight preference for features	Preference-based ranking	Normal	restaurants	1) User-based and item-based CF; 2) Item-based multi-criteria CF	MAE
Chen and Wang (2013)	Feature opinions	Weight preference for features	Latent class regression model (LCRM) based clustering and user-based CF	Sparse ratings/reviews	*Amazon* (cameras, laptops)	User-based CF with weight preference derived by probabilistic regression model (PRM)	Hit Ratio and MRR
Levi et al (2012)	Features (frequent nouns and noun phrases)	Weight preference for features in contexts (*nationality* and *trip intent*)	Preference-based ranking	New users	*Venere* and *TripAdvisor* (hotels)	Popularity-based ranking	User satisfaction
Chen and Chen (2014)	Aspect-level opinions under contexts	Context-independent and context-dependent weight preference for aspects	Preference-based ranking	Normal	*TripAdvisor* and *Yelp* (restaurants)	Non-context based ranking	Hit Ratio and MRR
Derived attribute value preference							
Wang et al (2013)	Feature opinions	Value preference for attributes	Preference-based ranking	New users	*Amazon* (cameras)	1) Popularity-based ranking; 2) Partial ranking with users' stated preferences only	Hit Ratio

dependent preferences, and show that this approach can lead to more accurate ranking outcomes. Levi et al (2012) assess how to use reviews to elicit new users' contextual preferences, and show their method's advantage over the popularity-based recommendation via a user evaluation.

The third sub-branch aims to derive another type of preference, attribute value preference in the form of $< attribute, opinion, specification >$, from reviews. A typical work particularly addresses the phenomenon of "incomplete value preference" that commonly occurs among new users and describes how to adopt derived reviewers' value preferences to predict a new user's unstated preference (Wang et al 2013). This approach performs better than the standard algorithm that simply uses incomplete user preferences for product ranking.

5 Review-based Product Profile Building

As indicated in Section 1, previous studies have also considered the role of reviews in enriching product profile, which in turn can augment the preference-based product ranking (see the background of this recommending approach in Section 2.3). For this type of work, it has been assumed that a target user's preference has already been elicited in the form of either weight preferences for attributes, or a query case. The main focus has therefore been on how to use reviews to build product profile for increasing products' ranking accuracy.

5.1 Considering Feature Opinions

This sub-category of studies aims to determine a product's quality using the feature opinions extracted from reviews. They vary in terms of how quality is computed and the extra information elements considered.

Involving feature opinions and reviewer expertise. Aciar et al (2007) develop an ontology-based product profile by translating the review content into a structured form with two components: *product quality*, which refers to the reviewer's evaluation of product features; and *opinion quality*, which indicates the reviewer's expertise with the reviewed product. They first apply text mining tools such as a text-miner software kit and rule induction kit (Weiss et al 2005) to select and classify review sentences into three categories: *good*, *bad*, and *quality* (that refers to the quality of the opinion). They then label each review sentence with one or more concepts (i.e., features). For instance, the words related to the concept "picture" for a camera include "photo," "photograph," "pixel," etc., so if a sentence contains one or more of these words, it will be labeled with the concept "picture". Afterwards, a set of computations is performed to determine a product's overall assessment degree (OA). The authors first compute a feature's quality value as the function of its associated opinion in a review and OQ (which indicates the reviewer's expertise): $FQ_f = r \times OQ$, where r is the aggregated opinion on feature f in the review. Then, the overall valuation of the feature with regard to all reviews of a product is calculated via: $OFQ_f = \frac{\sum Scalingfactor \times FQ_f}{NumberOfOpinions}$, where $Scalingfactor = 1/n$ (where n is the number of all features commented in a review) used to make a minor adjustment. Finally, the overall

assessment of the product is obtained by summing up all of its features' overall quality scores: $OA = \sum OFQ_f \times ImportanceIndex$, where *ImportanceIndex* denotes the feature's relative importance (i.e., the weight) to the target user, which can be either explicitly stated by the user in the current query, or estimated based on the feature's frequency of occurrence in the user's previously posted reviews. The products with the highest OA scores are then recommended.

Involving feature opinions and popularity. In (Dong et al 2013a,b), a case-based recommender is developed, in which user preference is represented by a query case Q (i.e., a product the user inputs as the reference for the query). The product case (to be matched to the query case) is constructed using both feature sentiment and feature popularity (note that popularity refers to the feature's occurring frequency in the product's reviews), which is formally modeled as (Dong et al 2013a)

$$Case(P) = \{[F_i, Sentiment(F_i, P), Pop(F_i, P)] : F_i \in Features(P)\} \tag{30}$$

where $Features(P)$ is the set of all of the valid features extracted from the product P's reviews, $Pop(F_i, P) = \frac{|R_k \in Reviews(P): F_i \in R_k|}{|Reviews(P)|}$ (i.e., the percentage of reviews that contain feature F_i), and $Sentiment(F_i, P)$ gives the product-level (also called case-level) sentiment score of the feature F_j which is computed via

$$Sentiment(F_i, P) = \frac{Pos(F_i, P) - Neg(F_i, P)}{Pos(F_i, P) + Neg(F_i, P) + Neutral(F_i, P)} \tag{31}$$

in which $Pos(F_i, P)$, $Neg(F_i, P)$, and $Neutral(F_i, P)$ respectively give the numbers of positive, negative, and neutral sentiment instances of F_i in the reviews of product P. For feature extraction and opinion identification, they apply shallow natural language processing (NLP) and a statistical method to extract frequent single nouns and bi-gram phrases as product features, and identify the opinions expressed about features through the opinion pattern mining method proposed in (Moghaddam and Ester 2010).

When generating recommendations, they prefer products that are not only highly similar to the query case Q with regard to feature popularity, but also enjoy a higher relative sentiment improvement. Thus,

$$Score(Q, P) = (1 - \omega) \times Sim(Q, P) + \omega \times (\frac{Better(Q, P) + 1}{2}) \tag{32}$$

where ω is a weighting parameter used to control the sentiment's relative contribution. The similarity score $Sim(Q, P)$ is concretely calculated based on the feature popularity:

$$Sim(Q, P) = \frac{\sum_{F_i \in Features(Q) \cap Features(P)} Pop(F_i, Q) \times Pop(F_i, P)}{\sqrt{\sum_{F_i \in Features(Q)} Pop(F_i, Q)^2} \times \sqrt{\sum_{F_i \in Features(P)} Pop(F_i, P)^2}} \tag{33}$$

and the sentiment-based better score $Better(Q, P)$ is obtained via one of the following two variations:

$$B1(Q, P) = \frac{\sum_{\forall F \in Features(P) \cap Features(Q)} better(F, Q, P)}{|Features(Q) \cap Features(P)|} \tag{34}$$

and

$$B2(Q,P) = \frac{\sum_{\forall F \in Features(P) \cup Features(Q)} better(F,Q,P)}{|Features(Q) \cup Features(P)|} \tag{35}$$

in which $better(F,Q,P) = \frac{Sentiment(F,P) - Sentiment(F,Q)}{2}$. It can be seen that $B1(Q,P)$ computes the average *better* score across all of the shared features between Q and P, whereas $B2(Q,P)$ computes the average score across the union of features. Only products that have at least k features ($k = 15$ in their experiment) appearing in the target user's query case Q are selected as recommendation candidates, among which the top-N products which are with the highest $Score(Q,P)$ are recommended to the user.

Involving feature opinions and static specifications. The feature opinions mined from product reviews can also be combined with a product's technical specifications to build a product profile, which is called a "product value model" in (Yates et al 2008). The model indicates the intrinsic value of a product for the average user, which is then personalized to the target user during the recommendation process. Specifically, they first train a Support Vector Machine (SVM) regression model, which uses opinion features (extracted from reviews by association rule mining (Liu et al 2005)) and technical specifications (e.g., the camera's lens, megapixels) as inputs. Product price is treated as an indicator of product value and the dependent variable in the training phase. Then, applying the trained SVM model to a new product can return its predicted intrinsic value, represented as $V(\hat{x})$, where $\hat{x} =< x_1, \ldots, x_n >$, and n is the number of features. Suppose the preference of the target user is $\hat{y} =< y_1, \ldots, y_n >$, where each component represents a feature in the value range [1,10], which is elicited by asking the user questions such as *"How much do you care about feature X?"*). A personalized value model $\hat{F} =< f_1, \ldots, f_n >$ is produced for the target user by adjusting the product's feature vector $\hat{x}$ as follows:

$$f_i(x_i, y_i) = \frac{1}{2} + \frac{y_i}{5} \times (x_i - \frac{1}{2}) \tag{36}$$

The difference between the user's personalized value model $V(\hat{F})$ and the product's value model $V(\hat{x})$ then indicates the product's suitability for the user:

$$ChangeinValue(\hat{x}, \hat{y}) = \frac{V(\hat{F}) - V(\hat{x})}{V(\hat{x})} \tag{37}$$

That is, the higher $ChangeinValue$, the more suitable the product for the user.

Evaluation. Aciar et al (2007) have not empirically tested the performance of their recommending approach, although they claim that their approach can be helpful for overcoming the cold-start problem. Dong et al (2013b) have tested a *Better* score based method, which does not consider the similarity score in Equation 32, with a dataset crawled from *Amazon* that includes 41,000 reviews of about 1,000 electronic products such as *GPS devices*, *laptops*, and *tablets*. They compare this method with two similarity-based approaches, one that uses *Jaccard* metric that prefers products that have a higher percentage of shared features with the query case; and another that uses *Cosine* metric, which is based on the sentiment scores of shared

features. The results show that the *Better* score based method is better with respect to two measures: the average *Better* score of the recommended products, and the ranking improvement of the recommended item against the query case according to the overall ratings. In another experiment (Dong et al 2013a), the method that combines the *Better* score and *Similarity* score (Equation 32) is compared with the recommendation returned by *Amazon*. The results show that their method can achieve the optimal balance between *query product similarity* (the average similarity based on mined features between the set of recommendations and the query case) and *ratings benefit*, when ω is around 0.9 (i.e., with higher contribution from *Better* score) and when *Better* score is computed via $B2(Q, P)$ (i.e., with the union of features).

Yates et al (2008) test their system's effectiveness on a dataset containing three product categories: 55 *digital cameras*, 105 *flat screen TVs*, and 78 *LCD monitors*. They compare the proposed *product value model* based ranking with three baselines: the standard preference-based ranking method, the traditional CF, and a non-personalized approach that is based purely on the product value model without considering the user's preference. The results show that their approach performs better in terms of *Percentile*, which measures the ranking position of the user's target choice in the recommendation list, and that the second best method is standard preference-based ranking.

5.2 Considering Comparative Opinions

Another sub-category of research uses reviewers' comparative opinions to enhance the products' ranking performance. As described in Section 3, the comparative opinion determines whether an item is superior or inferior to another item with regard to particular features. Ganapathibhotla and Liu (2008) develop a method to extract comparative relations from reviews. They express the comparative relation as ⟨*Comparative word, Features, Entitiy1, Entity2, Type*⟩. For example, ⟨*longer, battery life, camera X, camera Y, non-equal-gradable*⟩ is extracted from the comparative opinion sentence "*Camera X has longer battery life than camera Y*," where *non-equal-gradable* means that the two involved entities are not graded as equal in terms of the mentioned feature. Because comparative sentences use different language constructs, the authors define some special linguistic rules and further incorporate language context into the process of inferring which entity the review writer prefers. For example, consider the sentence "*Program X runs more quickly than program Y.*" The word "more" can be identified as an *increasing comparative* according to the linguistic rules and the pair ("run", "quickly") is extracted as a positive context. It is therefore inferred that "program X" is preferred to "program Y" in terms of running speed.

The approaches that use comparative opinions for product ranking aim to establish a weighted and directed graph with respect to each aspect (or feature). For example, in the graph built by Jamroonsilp and Prompoon (2013), each node represents a product. The direct edge from one product P_i to another P_j implies that P_j is preferred to P_i according to reviews that compared them in terms of aspect q; the edge's weight is defined by the number of related positive reviews. In (Zhang et al 2010), the edge indicates that there are review sentences comparing P_j with P_i

regarding feature f, and the edge's weight is the ratio of the number of *positive comparative* sentences (implying P_j is better than P_i) to that of *negative comparative* sentences (implying P_j is worse than P_i). The product node P is also assigned a weight in (Zhang et al 2010), to denote the product's inherent quality. It is formally defined as the ratio of the number of *positive subjective* sentences to the number of *negative subjective* sentences that all mention feature f.

More specifically, Jamroonsilp and Prompoon (2013) focus on the software domain. They pre-define five quality aspects for software products: performance, reliability, usability, security, and maintainability. Each comparative review sentence is then classified into one of the aspects via keyword matching, and the comparative relation is classified as either *increasing comparative* or *decreasing comparative* through the rule-based method described above (Ganapathibhotla and Liu 2008). The identified comparative relation is formally represented as <*software1, software2, quality type, sentiment*>, where *software1* is the directly mentioned software, *software2* is the one compared with *software1*, *quality type* is one of the five quality aspects, and *sentiment* is either 1 (indicating that *software1* is better than *software2*) or -1 (*software1* is worse than *software2*). The score assigned to product i for quality aspect q is then calculated as $r(i)_q = take(i) - give(i)$, where $take(i)$ is a score that takes from the software that points to software i, and $give(i)$ is a score that gives to the software that i points to. More formally, $take(i)$ is computed via $take(i) = \sum_{j \in B(i)} r(j)_q \times E(j,i)$ and $give(i)$ is via $give(i) = take(i) \times \sum_{k \in C(i)} E(i,k)$, where $B(i)$ is the set of products that point to product i and $C(i)$ is the set of products to which product i points; $E(j,i) = W_e(j,i)/W_e$, which is a normalized weight, where $W_e(j,i)$ is the weight of the edge from j to i, and W_e is the total weight of all of the edges. An overall score can then be computed for the product i by combining all of the scores related to the five quality aspects:

$$O(i) = \frac{\sum_{q=1}^{5} W_q \times r(i)_q}{5} \tag{38}$$

where W_q is the aspect's weight, which can be explicitly stated by the target user to indicate its relative importance to her/him. The products with the highest scores can then be recommended.

In contrast, Zhang et al (2010) extend the *PageRank* algorithm (Page et al 1999) with the constructed graph. They propose an algorithm called *pRank* that involves product nodes' weights. By including all of the comparative and subjective sentences, the algorithm can generate not only a feature-specific product ranking, but also an overall ranking.

Another related study uses product-related community-based Question Answering (cQA) pairs as an additional source of information about comparative opinions (Li et al 2011). For example, the phrase "*I'd go with*" can accommodate a preferred product in the answer. Specifically, they build two product comparative relation graphs based on user-generated reviews and cQA pairs, respectively, for each of the four product aspects: design, feature, performance, and ease of use. The two graphs are then fused together through a graph propagation strategy that assigns each product a superiority score:

$$PCS(p_i)^{k+1} = (1-d) + d \sum_{m=1}^{2} \sum_{j=1}^{n} \mu_m \beta_m PCS_m(p_j)^k \times E_m(p_j p_i) \tag{39}$$

where $PCS(p_i)^{k+1}$ is the superiority score w.r.t. one aspect of product p_i, k is the number of iterations, d is the damping factor, $m \in \{1, 2\}$ indicates the comparative relations from the reviews ($m = 1$) or from the cQA pairs ($m = 2$), and $\mu_m \in [0, 1]$ is used to control the relative contributions of the two resources ($\sum_{m=1}^{2} \mu_m = 1$). If there is a direct edge from product p_j to p_i, β_m is set to 1, otherwise it is set to 0. $E_m(p_j p_i)$ is calculated as $\frac{W_{m^{ji}}}{\sum_{l=1}^{L} W_{m^{jl}}}$, where L is the number of outbound links of product p_j, and $W_{m^{ji}}$ is the weight of the edge from p_j to p_i, which is higher if more users think product p_i is better than product p_j. Hence, the product with the highest superiority score is regarded as outperforming the others for the corresponding aspect. However, this study does not discuss how to compute an overall ranking for products if all of the aspects are considered together.

Evaluation. Several studies involve human experts or ordinary users to evaluate their proposed methods. Both (Jamroonsilp and Prompoon 2013) and (Zhang et al 2010) show that the rankings produced by their methods are statistically consistent with the rankings made by domain experts. Jamroonsilp and Prompoon (2013) further demonstrate that their rankings are more consistent than those in (Zhang et al 2010) due to the higher Pearson's correlation coefficient. Concretely, Jamroonsilp and Prompoon (2013) test their approach on a dataset with 105 software reviews collected via a Google custom search API. It covers three types of software: content management systems, PHP web application frameworks, and database management systems. Zhang et al (2010) conduct their experiment on a dataset of 1,350 digital cameras (with 83,005 reviews) and 760 televisions (with 24,495 reviews), collected from *Amazon*.

Li et al (2011) perform a user study (involving 12 ordinary users) to test their approach. Their two datasets include, respectively, 50,893 reviews of six mobile phones and 3,604 reviews of five MP3 players, collected from multiple websites such as *Cnet*, *Amazon*, *Reevoo*, and *Gsmarena*. In addition, 215 cQA pairs are crawled from *Yahoo! Answers*. Results show that more participants in the study prefer the rankings generated by the proposed method for most product aspects, relative to the rankings returned by two popular review websites, which rank products according to the aspect's average rating, and by a variation of their approach that only considers reviews without cQA pairs.

Summary of Section 5. Users' preferences can be elicited when they are using the system, which is especially helpful when the products are inexperienced by users, for example, if they are high-cost electronic products, or if users' preferences are changing. Preference-based product ranking is applied to cope with this type of "new user." Traditionally, ranking is based on products' static specifications, which are matched with a user's stated weight and/or value preference (see Section 2.3). In this section, reviews are used to construct the product's profile to improve the ranking accuracy.

There are two sub-categories of research (see the summary in Table 4). The first uses feature opinions to model products, such as the ontology built in (Aciar et al 2007), the product value model developed in (Yates et al 2008), and the product case constructed in (Dong et al 2013a,b). The sentiment-enriched product profile can be further integrated with other elements, such as reviewer expertise, feature popularity, or static specifications. The proposed ranking approaches are shown to perform better than the standard preference-based ranking (Yates et al 2008) or

Table 4 Summary of Typical Works on Review-based Product Profile Building

Citation	Review element	Product profile	Recommending method	Evaluation data condition	Tested products	Baseline	Evaluation metric
Enriched with feature opinions							
Aciar et al (2007)	Feature opinions and reviewer expertise	Ontology with product quality and opinion quality	Preference-based ranking	New users (with elicited weight preference)	cameras	*NA*	*NA*
Dong et al (2013a,b)	Feature opinions and popularity	Product case with feature sentiment and popularity	Product ranking based on sentiment improvement and popularity similarity	New users (with query case)	*Amazon* (GPS devices, laptops, tablets, etc.)	1) Similarity-based ranking; 2) *Amazon* recommendation	Average *Better* score, ratings benefit, and query product similarity
Yates et al (2008)	Feature opinions	Product value model based on feature opinions and technical specifications	Preference-based ranking	New users (with elicited weight preference)	*Amazon* (cameras, TVs, LCD monitors)	1) Standard preference-based ranking; 2) Non-personalized method; 3) Traditional CF	Percentile
Enriched with comparative opinions							
Jamroon-silp and Prompoon (2013)	Comparative opinions	Weighted and directed comparative graph	Graph based product ranking	New users (with elicited weight preference)	software	Human ranking	Pearson's correlation coefficient
Zhang et al (2010)	Comparative opinions	Weighted and directed comparative graph	pRank (extension of PageRank)	NA	*Amazon* (cameras, televisions)	Human ranking	Overlapping
Li et al (2011)	Comparative opinions from reviews and cQA pairs	Two product comparative relation graphs	Graph propagation strategy	NA	*Cnet*, *Amazon*, *Reevoo*, *Gsmarena*, and *Yahoo! Answers* (mobile phones, MP3 players)	Popularity-based ranking	User satisfaction

similarity-based methods (Dong et al 2013b). In the second sub-category, comparative opinions are used to establish a comparison relationship graph among products. The related studies primarily differ in terms of the algorithm developed to process the graph, which determines each product's superiority score (Jamroonsilp and Prompoon 2013; Li et al 2011) or performs the ranking directly (Zhang et al 2010). The algorithm's outcome is validated by human experts or ordinary users. However, some studies have not tailored the ranking to user preference (Li et al 2011; Zhang et al 2010), which limits their ability to provide personalized recommendations.

6 Practical Implications

From the foregoing survey, we can see that most studies have demonstrated the advantages of their review-based algorithms compared with standard recommending methods. In this section, we summarize the practical implications of these findings for five dimensions: data condition, new users, algorithm improvement, profile building, and product domain.

6.1 Sparse Rating Data

As mentioned in Section 1, a major limitation of collaborative filtering (CF) is a lack of sufficient ratings. Reviews can address this rating sparsity phenomenon in three ways: by creating term-based user profiles for content-based recommendations; by generating virtual ratings that make CF workable when the rating data are extremely sparse; and by enriching the available ratings with additional preference information.

For the first two, systems that incorporate review terms have demonstrated an ability to recommend novel, diverse, and high-coverage items, although the accuracy is slightly compromised compared with the rating-based CF (Garcia Esparza et al 2011). Systems that infer virtual ratings from reviews have shown that inferred ratings can be comparable to user-specified real ratings in terms of serving CF recommendation needs (Leung et al 2006; Poirier et al 2010b; Zhang et al 2013).

To enhance ratings, different types of review elements have been investigated, including review helpfulness, review topics, overall opinions, review contexts, review emotions, and aspect opinions (Hariri et al 2011; Moshfeghi et al 2011; Musat et al 2013; Pero and Horváth 2013; Raghavan et al 2012; Wang et al 2012). Basically, all of these approaches have demonstrated the positive effect of review elements on recommendation accuracy, relative to the standard approaches that take into account ratings alone. More notably, some studies have validated their algorithms' capacity to deal with sparse rating data, for example, a system that combines *overall opinions* with star ratings (Pero and Horváth 2013), a system that accommodates *review emotions* (Moshfeghi et al 2011), and a system that associates *aspect opinions* with latent factors (Wang et al 2012). Approaches grounded in preference-based product ranking have also demonstrated the effect of deriving users' multi-faceted preferences from their reviews (i.e., weight preference for features or aspects, or value preference for attributes) on improving ranking

accuracy, in comparison with the popularity-based ranking system (Chen and Wang 2013; Levi et al 2012; Wang et al 2013) or the classical rating-based CF (Liu et al 2013).

Furthermore, even in data conditions with a relatively lower sparsity level, reviews can still enrich the ratings. For instance, it has been found that review elements such as *review helpfulness* and *review topics* can be helpful for deciding the accompanying rating's quality (Musat et al 2013; Raghavan et al 2012). The *review contexts* can be used not only to predict an item's utility (i.e., the probability that a user will select it) (Hariri et al 2011; Li et al 2010), but also to help to expose users' contextual aspect-based preferences (Chen and Chen 2014; Levi et al 2012).

6.2 New Users

The rating sparsity problem discussed in the previous section refers to a dataset with a low overall rating sparsity level. "New users," on the other hand, represent a special group of users in the system who are either new to the system, so have not yet generated much data; or whose previous experiences are not suitable for the current task, so the system needs to elicit their current preferences on site. Such new user phenomenon causes the well-known cold-start problem, which impedes CF from producing satisfactory results for this group of users. Several of the systems discussed above have addressed this problem by incorporating review elements into the recommendation process.

For new users with few ratings, McAuley and Leskovec (2013) and Seroussi et al (2011) show that models that integrate *review topics* with latent factors return more accurate recommendations to users than the standard matrix factorization (MF) model that only considers ratings. This finding suggests that reviews can provide additional preference information relative to the same number of ratings, which is useful for solving the cold-start problem.

For the other type of new user whose preference is elicited on site, the existing systems principally rely on the preference-based ranking approach for studying the role of reviews. One sub-branch of research aims to help users to complete their preferences with the help of reviews. For instance, Levi et al (2012) use *review contexts* to adjust users' stated weight preferences by considering the contextual relevance of an item features. Wang et al (2013) aim to predict missing preferences that are not stated by new users by using reviewers' value preferences as derived from their reviews. Another category focuses on enriching the product profile with review elements such as *feature opinions* or *comparative opinions* to augment the ranking quality (Aciar et al 2007; Dong et al 2013a; Jamroonsilp and Prompoon 2013; Li et al 2011; Yates et al 2008; Zhang et al 2010).

6.3 Algorithm Improvement

Attempts to improve the standard recommending algorithms (in Section 2) can be divided into two major streams of research.

The first stream attempts to generate information that can be useful for existing algorithms. For instance, a term-based user profile can be built and used by the *content-based recommending*

approach (Garcia Esparza et al 2010, 2011). The virtual ratings inferred from reviews can take the role of real ratings in *user-based* or *item-based CF* (Leung et al 2006; Poirier et al 2010b; Zhang et al 2013). The weight/value preference derived from reviews can enable the *preference-based product ranking* system (Chen and Chen 2014; Levi et al 2012; Liu et al 2013; Wang et al 2013).

The second stream has the goal of revising the standard algorithms to accommodate added review elements. Most of the attempts to enhance ratings belong to this category, among which various extensions to the *latent factor model* have been proposed, including the Hidden Factors as Topics (HFT) model developed to align review topics with ratings (McAuley and Leskovec 2013), the revised probabilistic latent relational model (PLRM) designed to relate review contexts to ratings (Li et al 2010), the tensor model for embedding aspect opinions (Wang et al 2012), and the multi-relational MF (MRMF) for modeling various interactions among users, items, and aspect opinions (Jakob et al 2009). The *item-based k-NN* has also been extended to embody both rating-based and context-based similarities to make utility predictions (Hariri et al 2011). For the *preference-based product ranking*, a user's weight preference has been matched to the product's feature sentiment scores that are derived from reviews, rather than the product's static specifications (Aciar et al 2007; Levi et al 2012; Liu et al 2013). The traditional similarity-based ranking in case-based recommenders has been improved by considering product features' sentiment improvement relative to the query case (Dong et al 2013a,b).

The developed algorithms have been compared with the traditional forms in those studies. For example, the model-based CF algorithms designed to combine review elements with ratings have been shown to be more accurate than the standard models that do not consider the review elements (Jakob et al 2009; Li et al 2010; McAuley and Leskovec 2013; Pero and Horváth 2013; Raghavan et al 2012; Seroussi et al 2011; Wang et al 2012). The user-based and item-based CF methods based on inferred ratings have been shown to generate results comparable to those of the standard CF based on real ratings (Poirier et al 2010b; Zhang et al 2013), and it has further been demonstrated that the inferred ratings are more effective in user-based CF than in item-based CF (Zhang et al 2013). Moreover, it has been shown that context-based utility prediction is better than standard item-based rating prediction (Hariri et al 2011).

In addition, some studies have made comparisons across different review-based algorithms. The latent factor model embedded with aspect opinions obtains better rating predictions than the model incorporated with review emotions (Wang et al 2012). The aspect opinions also provide more benefits to the user-based CF, than the overall opinions (Ganu et al 2013). Deriving weight preference has been found to be more effective in improving product ranking than directly using aspect opinions (Liu et al 2013).

6.4 User Profile and Product Profile

There are three types of user profiles. The first type is a *term-based profile*, which is constructed from frequent keywords that are extracted from reviews. Each term is assigned a weight to indicate its importance to the reviewer. This profile is different from the traditional term profile

that is built with the item's static descriptions. The advantage is that the review terms can be more representative of the user's personal characteristics including her/his subjective preferences and writing style, whereas the terms from static descriptions are the same for people who select the same items. Another advantage is that it enables the content-based recommending technique to be feasibly applied to a broader scope of products if they have user reviews.

The second type of use profile is the common *rating profile* upon which collaborative filtering techniques depend. As indicated before, reviews have been used not only to infer ratings, but also to enhance ratings. In addition, the rich information hidden in reviews can be useful for discovering users' fine-grained *feature preference* which is the third type of user profile. Actually, a rating can only tell how much a user likes an item, whereas feature preference can reveal why s/he likes it. The research summarized above shows that feature preference can be either directly expressed as an ⟨aspect, sentiment⟩ pair (Ganu et al 2013; Jakob et al 2009; Wang et al 2012), or derived in the form of a weight/value preference placed on a feature/aspect or attribute (Chen and Wang 2013; Liu et al 2013; Wang et al 2013). The proven benefits of incorporating this information are not only the improvement of the recommendations for users who have review histories in the system (Chen and Wang 2013; Liu et al 2013), but also allowing the system to more effectively elicit a new user's feature preference (Wang et al 2013). In addition to the three types of user profile, a user's contextual preference can also be determined by correlating the review contexts with her/his ratings or feature preference (Chen and Chen 2014; Hariri et al 2011; Levi et al 2012; Li et al 2010).

Regarding product profile, feature opinions or comparative opinions embedded in reviews have been used to enrich the product profile. The feature opinions are beneficial for building the product ontology (Aciar et al 2007), the product value model (Yates et al 2008), and the product case (Dong et al 2013a,b). The comparative opinions are useful for showing comparative relationships between products (Jamroonsilp and Prompoon 2013; Li et al 2011; Zhang et al 2010). Researchers have used the enriched product profiles to increase the quality of product ranking, so that products with higher sentiment values can be recommended to the user. They demonstrate that the ranking results are consistent with human judgement, implying that the review-based product profile satisfies users' needs of using reviews to assess products' quality.

6.5 Product Domain

To some extent, the product's inherent property determines how the user's preference should be modeled and what kind of recommender algorithm should be applied. After classifying all of the surveyed studies according to the products in their experimental datasets, we find there are three major types of product: *frequently experienced products* (such as movies, music, books, games), *infrequently experienced products* (such as high-cost electronic products including digital cameras, TVs, laptops), and *context sensitive products* (such as hotels and restaurants).

For the first type of product, the user's preference can be modeled as a term-based profile or a rating profile. Therefore, we can either apply the content-based recommending method with the input of review terms (Garcia Esparza et al 2010, 2011), or the rating-based collaborative

filtering technique. For the latter, if users only provide reviews, we can convert the reviews into ratings (Leung et al 2006; Poirier et al 2010b; Zhang et al 2013); if both ratings and reviews are available, the information elements extracted from reviews can be used to enhance the ratings (Jakob et al 2009; McAuley and Leskovec 2013; Moshfeghi et al 2011; Musat et al 2013; Pero and Horváth 2013; Raghavan et al 2012; Seroussi et al 2011; Wang et al 2012).

For the second type, as the products are usually expensive, the user is unlikely to have experience with many. In reality, most users are new buyers. Therefore, recommending methods based on transaction histories such as content-based and collaborative filtering techniques are not suitable. The preference-based product ranking strategy has been widely applied in this situation, for which users' preferences are expressed as weights placed on multiple features (i.e., the weight preference) and/or value requirements for different attributes (i.e., the value preference). The weight/value preferences derived from reviews can be helpful for predicting a new user's preference (Chen and Wang 2013; Wang et al 2013). Moreover, the product profile can be enriched with review elements to better match the user's preference (Aciar et al 2007; Dong et al 2013a,b; Jamroonsilp and Prompoon 2013; Yates et al 2008).

Review contexts have been exploited for the third type of product, *context sensitive products*. Some common systems are based on item-based k-NN or the latent factor model, which associate contextual factors with user ratings (Hariri et al 2011; Li et al 2010). The advantage of incorporating reviews is that they can expose the different kinds of context that affect users' ratings, such as time, occasion, and companion when eating a meal in a restaurant. In addition, they can reveal the influence of the context on users' aspect preferences, which is why Levi et al (2012) and Chen and Chen (2014) have attempted to model users' context-dependent weight preferences based on reviews. Their results have highlighted the algorithms' ability to outperform other systems in returning the top-N recommendations.

7 Future Trends

Our survey of review-based recommender systems shows that although there has been remarkable progress in recent years, further investigation is needed. In this section, we highlight several directions for future research.

Combining different types of review elements. To date, most studies have focused on investigating the role of one type of review element in enhancing the recommendation. Few studies have tried to combine two or more elements to further increase the accuracy of the user/product profile. For instance, one study has shown that review contexts can be combined with aspect opinions to detect users' contextual preferences at the aspect level, which has been demonstrated to perform better than a method that does not consider review contexts, but also better than a method that models contextual preferences at the item level (Chen and Chen 2014). In the studies about product profiles, it has been found that combining a reviewer's expertise with feature opinions can more accurately disclose the opinion's quality (Aciar et al 2007), and combining feature popularity (i.e., occurrence frequency in reviews) with feature opinions can balance two items' similarity against their relative sentiment improvement (Dong

et al 2013a,b). Therefore, we believe that more combinations could be explored. In particular, as feature opinions naturally reflect a reviewer's multi-faceted criteria, it should be beneficial to combine them with different elements, such as review emotions to detect users' emotion-dependent feature preferences, or comparative opinions for constructing a more personalized product comparison graph. Moreover, the review elements could also be combined with products' static descriptions. As a successful trial, Yates et al (2008) have unified opinion features with static specifications to make product value model, which can be used to predict the intrinsic value of a product to the average user, and furthermore be personalized to the target user given her/his weight preference. Combining feature opinions with static attribute values can also be useful for understanding a reviewer's value preferences for particular attributes, as shown in (Wang et al 2013).

Benefiting other types of recommender systems. The connection between review-based recommender systems and other types of recommenders could be strengthened in the future. For example, reviews could potentially benefit a multi-criteria recommender system. The classical multi-criteria recommender system relies on user ratings for multiple aspects of an item (Adomavicius and Kwon 2007) to calculate the user-based or item-based similarity for CF (Tang and McCalla 2009). In our view, the feature opinions embedded in reviews might be used to infer multi-aspect ratings. For this purpose, Faridani (2011) has proven the effectiveness of a supervised learning model, Canonical Correlation Analysis (CCA) (Bach and Jordan 2005), for estimating multi-aspect ratings in a review. Additionally, in a situation in which multi-aspect ratings have already been specified by users, we believe that reviews could enhance these ratings, just as they can enhance overall ratings (see Section 4.3). Indeed, unlike the ratings that users give to a set of aspects that the system pre-defines, reviews can uncover other aspects that users freely mention in the text. Moreover, the words in the review text may more precisely indicate users' personal opinions about the aspects. For example, users may use adverbs as intensifiers to strengthen or soften opinions. Thus, combining the extracted aspect opinions with user-specified aspect ratings could improve the system's understanding of users and hence return better recommendations.

Research on review contexts could augment existing context-aware and mobile recommenders (Adomavicius and Tuzhilin 2011; Baltrunas et al 2012). It would be interesting to verify the benefits that review contexts bring to these systems through well-designed experiments. Similarly, more research could be meaningfully conducted on review emotions to boost the field of emotion-based recommenders (Gonzalez et al 2007; Tkalcic et al 2013). For instance, Martin and Pu (2014) have recently shown that the emotions detected in a review can help predict the review's helpfulness, which might indicate a way to improve the accuracy of the quality score assigned to the accompanying rating (Raghavan et al 2012).

Improving evaluation. According to our survey, the evaluations of existing review-based recommender systems are with the following limitations. 1) The evaluations of the methods grounded in the content-based recommending technique do not compare user profiles built with review terms with traditional profiles built with item descriptions (Garcia Esparza et al 2011). 2) Some studies that exploit reviews for product ranking simply compare their approaches with

the non-personalized popularity-based method, rather than with the standard preference-based ranking that is based on static attribute values (Dong et al 2013a; Jamroonsilp and Prompoon 2013; Levi et al 2012). 3) The methods that use reviews to infer or enhance ratings lack experimental comparisons across the different approaches. For example, we should compare the method based on review topics (McAuley and Leskovec 2013) with the method based on aspect opinions (Wang et al 2012) to identify the different review elements' respective advantages.

Moreover, few existing studies have conducted user evaluation to validate their methods' practical benefits to real users (Jamroonsilp and Prompoon 2013; Levi et al 2012). Actually, instead of experiments that use evaluation metrics such as *RMSE* or *Hit Ratio* to determine an algorithm's rating prediction or ranking accuracy, user evaluation could reveal the system's performance from the perspective of user experiences, and indicate whether a system can efficiently assist users in locating favorite items by measuring users' time consumption and interaction cycles. Moreover, it could measure users' subjective feelings, such as their perception of recommendation quality and their satisfaction with the system. Such evaluation results would be meaningful for practitioners, as they would see a system's value in real life. In designing and implementing user evaluation, the recently developed user-centered evaluation frameworks for recommender systems could be referential (Knijnenburg et al 2012; Pu et al 2011).

Beyond recommendation: producing review-based explanations. In addition to using reviews to improve a recommender algorithm's accuracy, we could also exploit them to explain recommendations. It has been shown that a good explanation can be effective in increasing users' trust in the system, as it can tell users why the items are recommended to them (Al-Taie 2013; Tintarev and Masthoff 2011). Based on reviews, the explanations for recommendations could be improved by focusing on the aspects users like/dislike, which would help them to make a more informed and accurate decision. For instance, we have recently proposed fusing reviews with products' static specifications to explain the pros and cons of recommended items in terms of both feature sentiments and attribute values (Chen and Wang 2014). This work is an extension of our previously developed organization-based explanation interface (Chen and Pu 2010), with the objective of accommodating the influence of reviews on supporting users to make attribute tradeoffs. User evaluation could be conducted on such a review-based explanation interface to verify its effectiveness, trustworthiness, and persuasiveness from users' perspectives (Tintarev and Masthoff 2011).

8 Conclusions

In recent years, due to the appearance of advanced opinion mining and text analysis techniques that transform unstructured textual reviews into structured forms that can be more easily understood by a computer system, much effort has been devoted to using reviews to augment recommender systems. The rich opinion information embedded in reviews, such as the multi-faceted nature of feature opinions, contextual opinions, and comparative opinions, makes this branch of research distinguishable from the branches based on simple texts like posts and tweets from social networking websites (Hannon et al 2010; Wu et al 2010).

In this article, we survey state-of-the-art research on review-based recommender systems. We classify the systems according to the two main types of profile building: *review-based user profile building*, and *review-based product profile building*. For the first category, we discuss how existing studies have used reviews to create term-based user profile, enrich rating profile, and derive feature preference. Various types of review elements, such as review helpfulness, review topics, overall opinions, feature opinions, review contexts, and review emotions, have been used to enhance the standard content-based recommending method and rating-based collaborative filtering method. In the category of product profile building, feature opinions and comparative opinions have been exploited, which can be helpful for increasing the products' ranking accuracy. We further discuss the practical implications of these studies in terms of solving the well-known rating sparsity and new user problems, and their proven ability to improve the currently used algorithms and practical uses in different types of product domains.

We expect this survey to encourage investigators to pursue the hidden values of reviews in future studies. For instance, combining multiple types of review elements might be more effective than considering a single type when modeling a user's preference. The effects of reviews on enhancing multi-criteria recommenders, context-aware recommenders, and emotion-based recommenders could be investigated in more comprehensive studies. More realistic evaluation techniques, such as user evaluation, could validate the practical benefits of the review-based recommending method. Beyond recommendation, reviews could also be exploited to design more effective user interfaces, such as an explanation interface.

9 Acknowledgements

We thank Hong Kong RGC for sponsoring the reported work (under project ECS/HKBU211912). We also thank reviewers for their suggestions and comments.

References

Acar E, Dunlavy DM, Kolda TG, Mørup M (2011) Scalable tensor factorizations for incomplete data. Chemometrics and Intelligent Laboratory Systems 106(1):41–56

Aciar S, Zhang D, Simoff S, Debenham J (2007) Informed recommender: Basing recommendations on consumer product reviews. IEEE Intelligent Systems 22(3):39 –47

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

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

Adomavicius G, Tuzhilin A (2011) Context-aware recommender systems. In: Ricci F, Rokach L, Shapira B, Kantor PB (eds) Recommender Systems Handbook, Springer, pp 217–253

Al-Taie MZ (2013) Explanations in recommender systems: Overview and research approaches. In: Proceedings of the 14th International Arab Conference on Information Technology, Khartoum, Sudan, ACIT'13

Bach F, Jordan MI (2005) A probabilistic interpretation of canonical correlation analysis. Tech. Rep. 688, Department of Statistics, University of California, Berkeley, USA

Balabanović M, Shoham Y (1997) Fab: Content-based, collaborative recommendation. Communications of the ACM 40(3):66–72

Baltrunas L, Ludwig B, Peer S, Ricci F (2012) Context relevance assessment and exploitation in mobile recommender systems. Personal and Ubiquitous Computing 16(5):507–526

Beilin L, Yi S (2013) Survey of personalized recommendation based on society networks analysis. In: Proceedings of the 6th International Conference on Information Management, Innovation Management and Industrial Engineering, Xi'an, China, ICIII' 13, vol 3, pp 337–340

Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. Journal of Machine Learning Research 3:993–1022

Celma O, Herrera P (2008) A new approach to evaluating novel recommendations. In: Proceedings of the 2nd ACM International Conference on Recommender Systems, Lausanne, Switzerland, ACM, RecSys'08, pp 179–186

Chatterjee P (2001) Online reviews: Do consumers use them? Advances in Consumer Research 28:129–133

Chee SHS, Han J, Wang K (2001) Rectree: An efficient collaborative filtering method. In: Kambayashi Y, Winiwarter W, Arikawa M (eds) Proceedings of the 3rd International Conference on Data Warehousing and Knowledge Discovery, Munich, Germany, Springer-Verlag, DaWaK'01, pp 141–151

Chelcea S, Gallais G, Trousse B (2004) A personalized recommender system for travel information. In: Proceedings of the 1st French-speaking Conference on Mobility and Ubiquity Computing, Nice, France, ACM, UbiMob'04, pp 143–150

Chen G, Chen L (2014) Recommendation based on contextual opinions. In: Dimitrova V, Kuflik T, Chin D, Ricci F, Dolog P, Houben GJ (eds) Proceedings of the 22nd Conference on User Modeling, Adaptation, and Personalization, Alborg, Denmark, Springer, UMAP'14, pp 61–73

Chen L, Pu P (2004) Survey of preference elicitation methods. Tech. Rep. IC/200467, Swiss Federal Institute of Technology in Lausanne (EPFL), Lausanne, Switzerland

Chen L, Pu P (2010) Experiments on the preference-based organization interface in recommender systems. ACM Transactions on Computer-Human Interaction 17(1):5:1–5:33

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

Chen L, Wang F (2014) Sentiment-enhanced explanation of product recommendations. In: Proceedings of the 23rd International Conference on World Wide Web Companion, Seoul, Korea, ACM, WWW Companion'14, pp 239–240

Chen L, Zeng W, Yuan Q (2013) A unified framework for recommending items, groups and friends in social media environment via mutual resource fusion. Expert Systems with Applications 40(8):2889–2903

Chevalier JA, Mayzlin D (2006) The effect of word of mouth on sales: Online book reviews. Journal of Marketing Research 43(3):345–354

Dong R, O'Mahony MP, Schaal M, McCarthy K, Smyth B (2013a) Sentimental product recommendation. In: Proceedings of the 7th ACM International Conference on Recommender Systems, Hong Kong, China, ACM, RecSys'13, pp 411–414

Dong R, Schaal M, Oahony M, McCarthy K, Smyth B (2013b) Opinionated product recommendation. In: Delany SJ, Ontanon S (eds) Proceedings of the 21st International Conference on Case-Based Reasoning, Saratoga

Springs, NY, USA, Springer Berlin Heidelberg, ICCBR' 13, pp 44–58

Esuli A, Sebastiani F (2006) Sentiwordnet: A publicly available lexical resource for opinion mining. In: Proceedings of the 5th Conference on Language Resources and Evaluation, Genoa, Italy, LREC'06, vol 6, pp 417–422

Faridani S (2011) Using canonical correlation analysis for generalized sentiment analysis, product recommendation and search. In: Proceedings of the 5th ACM Conference on Recommender Systems, Chicago, Illinois, USA, ACM, RecSys'11, pp 355–358

Fellbaum C (1998) WordNet: An Electronic Lexical Database. MIT Press

Friedman JH (2000) Greedy function approximation: A gradient boosting machine. Annals of Statistics 29:1189–1232

Ganapathibhotla M, Liu B (2008) Mining opinions in comparative sentences. In: Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1, Manchester, UK, Association for Computational Linguistics, COLING'08, pp 241–248

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

Garcia Esparza S, O'Mahony MP, Smyth B (2010) Effective product recommendation using the real-time web. In: Bramer M, Petridis M, Hopgood A (eds) Proceedings of the 30th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, Cambridge, UK, Springer, pp 5–18

Garcia Esparza S, O'Mahony MP, Smyth B (2011) A multi-criteria evaluation of a user-generated content based recommender system. In: Proceeding of the 3rd Workshop on Recommender Systems and the Social Web in RecSys'11, Chicago, Illinois, USA, pp 49–56

Gonzalez G, de la Rosa JL, Montaner M, Delfin S (2007) Embedding emotional context in recommender systems. In: Proceedings of the International Workshop on Web Personalisation, Recommender Systems and Intelligent User Interfaces in ICDE'07, Istanbul, Turkey, IEEE, pp 845–852

Hannon J, Bennett M, Smyth B (2010) Recommending twitter users to follow using content and collaborative filtering approaches. In: Proceedings of the 4th ACM Conference on Recommender Systems, Barcelona, Spain, ACM, RecSys'10, pp 199–206

Hariri N, Mobasher B, Burke R, Zheng Y (2011) Context-aware recommendation based on review mining. In: Proceedings of the 9th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems in IJCAI'11, Barcelona, Spain, pp 30–36

Herlocker JL, Konstan JA, Terveen LG, Riedl JT (2004) Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS) 22(1):5–53

Hofmann T (2004) Latent semantic models for collaborative filtering. ACM Transactions on Information Systems 22(1):89–115

Horvitz E, Breese J, Heckerman D, Hovel D, Rommelse K (1998) The lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, Madison, Wisconsin, USA, Morgan Kaufmann Publishers Inc., pp 256–265

Hu M, Liu B (2004a) Mining and summarizing customer reviews. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, ACM, KDD'04, pp 168–177

Hu M, Liu B (2004b) Mining opinion features in customer reviews. In: Proceedings of the 19th National Conference on Artifical Intelligence, San Jose, California, USA, AAAI Press, AAAI'04, pp 755–760

Jakob N, Weber SH, Müller MC, Gurevych I (2009) Beyond the stars: Exploiting free-text user reviews to improve the accuracy of movie recommendations. In: Proceedings of the 1st International Workshop on Topic-Sentiment Analysis for Mass Opinion in CIKM'09, Hong Kong, China, ACM, TSA'09, pp 57–64

Jamroonsilp S, Prompoon N (2013) Analyzing software reviews for software quality-based ranking. In: Proceedings of the 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, Krabi, Thailand, IEEE, ECTI-CON'13, pp 1–6

Jin W, Ho HH, Srihari RK (2009) Opinionminer: A novel machine learning system for web opinion mining and extraction. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, ACM, KDD'09, pp 1195–1204

Jindal N, Liu B (2006) Mining comparative sentences and relations. In: Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2, Boston, Massachusetts, USA, AAAI Press, AAAI'06, pp 1331–1336

Jindal N, Liu B (2008) Opinion spam and analysis. In: Proceedings of the 1st International Conference on Web Search and Data Mining, Palo Alto, California, USA, ACM, WSDM'08, pp 219–230

Kamps J, Mokken RJ, Marx M, de Rijke M (2004) Using WordNet to measure semantic orientation of adjectives. In: Proceedings of the 4th International Conference on Language Resources and Evaluation, Lisbon, Portugal, European Language Resources Association, LREC'04, vol IV, pp 1115–1118

Keeney R, Raiffa H (1976) Decisions with Multiple Objectives: Preferences and Value Tradeoffs. Cambridge University Press

Kendall MG (1938) A new measure of rank correlation. Biometrika 30(1/2):81–93

Kim Y, Srivastava J (2007) Impact of social influence in e-commerce decision making. In: Proceedings of the 9th International Conference on Electronic Commerce, Minneapolis, MN, USA, ACM, ICEC'07, pp 293–302

Knijnenburg BP, Willemsen MC, Gantner Z, Soncu H, Newell C (2012) Explaining the user experience of recommender systems. User Modeling and User-Adapted Interaction 22(4-5):441–504

Koren Y, Bell R (2011) Advances in collaborative filtering. In: Ricci F, Rokach L, Shapira B, Kantor PB (eds) Recommender Systems Handbook, Springer, pp 145–186

Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. IEEE Computer 42(8):30–37

Leung CWK, Chan SCF, Chung F (2006) Integrating collaborative filtering and sentiment analysis: A rating inference approach. In: Proceedings of the ECAI 2006 Workshop on Recommender Systems, Riva del Garda, Italy, pp 62–66

Levi A, Mokryn O, Diot C, Taft N (2012) Finding a needle in a haystack of reviews: Cold start context-based hotel recommender system. In: Proceedings of the 6th ACM International Conference on Recommender Systems, Dublin, Ireland, ACM, RecSys'12, pp 115–122

Li S, Zha ZJ, Ming Z, Wang M, Chua TS, Guo J, Xu W (2011) Product comparison using comparative relations. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, Beijing, China, ACM, SIGIR'11, pp 1151–1152

Li Y, Nie J, Zhang Y, Wang B, Yan B, Weng F (2010) Contextual recommendation based on text mining. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, Beijing, China, Association for Computational Linguistics, COLING'10, pp 692–700

Lippert C, Weber SH, Huang Y, Tresp V, Schubert M, Kriegel HP (2008) Relation-prediction in multi-relational domains using matrix factorization. In: Proceedings of the Workshop on Structured Input-Structured Output

in NIPS'08, Vancouver, B.C., Canada

Liu B (2010) Sentiment analysis and subjectivity. In: Indurkhya N, Damerau FJ (eds) Handbook of Natural Language Processing, Second Edition, Taylor and Francis Group, pp 627–666

Liu B (2012) Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers

Liu B, Hu M, Cheng J (2005) Opinion observer: Analyzing and comparing opinions on the web. In: Proceedings of the 14th International Conference on World Wide Web, Chiba, Japan, ACM, WWW'05, pp 342–351

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

Lops P, de Gemmis M, Semeraro G (2011) Content-based recommender systems: State of the art and trends. In: Ricci F, Rokach L, Shapira B, Kantor PB (eds) Recommender Systems Handbook, Springer, pp 73–105

Lorenzi F, Ricci F (2005) Case-based recommender systems: A unifying view. In: Mobasher B, Anand SS (eds) Proceedings of the IJCAI 2003 Workshop on Intelligent Techniques for Web Personalization, Acapulco, Mexico, Springer-Verlag, ITWP'03, pp 89–113

Manning CD, Raghavan P, Schütze H (2008) Introduction to Information Retrieval. Cambridge University Press, 1st Edition

Marinho LB, Nanopoulos A, Schmidt-Thieme L, Jäschke R, Hotho A, Stumme G, Symeonidis P (2011) Social tagging recommender systems. In: Ricci F, Rokach L, Shapira B, Kantor PB (eds) Recommender Systems Handbook, Springer, pp 615–644

Martin L, Pu P (2014) Prediction of helpful reviews using emotions extraction. In: Proceedings of the 28th National Conference on Artificial Intelligence, Quebec City, Quebec, Canada, AAAI Press, AAAI'14, pp 1551–1557

McAuley J, Leskovec J (2013) Hidden factors and hidden topics: Understanding rating dimensions with review text. In: Proceedings of the 7th ACM International Conference on Recommender Systems, Hong Kong, China, ACM, RecSys'13, pp 165–172

McSherry D (2003) Similarity and compromise. In: Ashley KD, Bridge DG (eds) Proceedings of the 5th International Conference on Case-Based Reasoning, Trondheim, Norway, Springer, ICCBR'03, pp 291–305

Miao Q, Li Q, Zeng D (2010) Mining fine grained opinions by using probabilistic models and domain knowledge. In: Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01, Toronto, Canada, IEEE Computer Society, WI-IAT'10, pp 358–365

Moghaddam S, Ester M (2010) Opinion digger: An unsupervised opinion miner from unstructured product reviews. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, Toronto, Canada, ACM, CIKM'10, pp 1825–1828

Moshfeghi Y, Piwowarski B, Jose JM (2011) Handling data sparsity in collaborative filtering using emotion and semantic based features. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, Beijing, China, ACM, SIGIR'11, pp 625–634

Musat CC, Liang Y, Faltings B (2013) Recommendation using textual opinions. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, AAAI Press, IJCAI'13, pp 2684–2690

Ortony A, Clore GL, Collins A (1990) The Cognitive Structure of Emotions. Cambridge University Press

Page L, Brin S, Motwani R, Winograd T (1999) The pagerank citation ranking: Bringing order to the web. Tech. Rep. 1999-66, Stanford University, California, USA

Pang B, Lee L (2008) Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2(1-2):1–135

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 - Volume 10, Philadephia, Pennsylvania, USA, Association for Computational Linguistics, EMNLP'02, pp 79–86

Payne JW, Bettman JR, Johnson EJ (1993) The Adaptive Decision Maker. Cambridge University Press

Pazzani MJ, Billsus D (2007) Content-based recommendation systems. In: Brusilovsky P, Kobsa A, Nejdl W (eds) The Adaptive Web, Springer, pp 325–341

Pero Š, Horváth T (2013) Opinion-driven matrix factorization for rating prediction. In: Carberry S, Weibelzahl S, Micarelli A, Semeraro G (eds) Proceedings of the 21st International Conference on User Modeling, Adaptation, and Personalization, Rome, Italy, Springer, UMAP'13, pp 1–13

Poirier D, Fessant F, Tellier I (2010a) Reducing the cold-start problem in content recommendation through opinion classification. In: Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01, Toronto, Canada, IEEE Computer Society, WI-IAT'10, pp 204–207

Poirier D, Tellier I, Fessant F, Schluth J (2010b) Towards text-based recommendations. In: Proceeding of the 9th International Conference on Adaptivity, Personalization and Fusion of Heterogeneous Information, Paris, France, RIAO'10, pp 136–137

Popescu AM, Etzioni O (2005) Extracting product features and opinions from reviews. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, Vancouver, Canada, Association for Computational Linguistics, HLT'05, pp 339–346

Poriya A, Bhagat T, Patel N, Sharma R (2014) Non-personalized recommender systems and user-based collaborative recommender systems. International Journal of Applied Information Systems 6(9):22–27

Pu P, Chen L (2005) Integrating tradeoff support in product search tools for e-commerce sites. In: Proceedings of the 6th ACM Conference on Electronic Commerce, Vancouver, Canada, ACM, EC'05, pp 269–278

Pu P, Chen L, Hu R (2011) A user-centric evaluation framework for recommender systems. In: Proceedings of the 5th ACM Conference on Recommender Systems, Chicago, Illinois, USA, ACM, RecSys'11, pp 157–164

Qi L, Chen L (2010) A linear-chain CRF-based learning approach for web opinion mining. In: Chen L, Triantafillou P, Suel T (eds) Proceedings of the 11th International Conference on Web Information Systems Engineering, Hong Kong, China, Springer-Verlag, WISE'10, pp 128–141

Qiu G, Liu B, Bu J, Chen C (2011) Opinion word expansion and target extraction through double propagation. Computational Linguistics 37(1):9–27

Raghavan S, Gunasekar S, Ghosh J (2012) Review quality aware collaborative filtering. In: Proceedings of the 6th ACM Conference on Recommender systems, Dublin, Ireland, ACM, RecSys'12, pp 123–130

Ramage D, Hall D, Nallapati R, Manning CD (2009) Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Singapore, Association for Computational Linguistics, EMNLP'09, pp 248–256

Ratnaparkhi A (1998) Maximum entropy models for natural language ambiguity resolution. PhD thesis, University of Pennsylvania, Philadelphia, PA, USA

Salakhutdinov R, Mnih A (2008) Probabilistic matrix factorization. In: Proceedings of the 22nd Annual Conference on Advances in Neural Information Processing Systems, Vancouver, Canada, NIPS'08, vol 20, pp

1257–1264

Salton G, Buckley C (1988) Term-weighting approaches in automatic text retrieval. Information Processing & Management 24(5):513–523

Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web, Hong Kong, China, ACM, WWW'01, pp 285–295

Schafer JB, Konstan JA, Riedl J (2001) E-commerce recommendation applications. Data Mining and Knowledge Discovery 5(1-2):115–153

Schafer JB, Frankowski D, Herlocker J, Sen S (2007) Collaborative filtering recommender systems. In: Brusilovsky P, Kobsa A, Nejdl W (eds) The Adaptive Web, Springer, pp 291–324

Seroussi Y, Bohnert F, Zukerman I (2011) Personalised rating prediction for new users using latent factor models. In: Proceedings of the 22nd ACM Conference on Hypertext and Hypermedia, Eindhoven, The Netherlands, ACM, HT'11, pp 47–56

Shaikh M, Prendinger H, Ishizuka M (2009) A linguistic interpretation of the OCC emotion model for affect sensing from text. In: Tao J, Tan T (eds) Affective Information Processing, Springer London, pp 45–73

Shani G, Gunawardana A (2011) Evaluating recommendation systems. In: Ricci F, Rokach L, Shapira B, Kantor PB (eds) Recommender Systems Handbook, Springer US, pp 257–297

Slonim N (2002) The information bottleneck: Theory and applications. PhD thesis, Hebrew University of Jerusalem, Jerusalem, Israel

Smyth B (2007) Case-based recommendation. In: Brusilovsky P, Kobsa A, Nejdl W (eds) The Adaptive Web, Springer, pp 342–376

Smyth B, Cotter P (2000) A personalised TV listings service for the digital TV age. Knowledge-Based Systems 13(2):53–59

Smyth B, McClave P (2001) Similarity vs. diversity. In: Aha D, Watson I (eds) Proceedings of the 4th International Conference on Case-Based Reasoning: Case-Based Reasoning Research and Development, Vancouver, Canada, Springer-Verlag, ICCBR'01, pp 347–361

Snyder B, Barzilay R (2007) Multiple aspect ranking using the good grief algorithm. In: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Rochester, NY, USA, HLT-NAACL'07, pp 300–307

Su X, Khoshgoftaar TM (2009) A survey of collaborative filtering techniques. Advances in Artificial Intelligence 2009(4)

Takács G, Pilászy I, Németh B, Tikk D (2007) Major components of the gravity recommendation system. ACM SIGKDD Explorations Newsletter 9(2):80–83

Tang TY, McCalla G (2009) The pedagogical value of papers: A collaborative-filtering based paper recommender. Journal of Digital Information 10(2)

Tintarev N, Masthoff J (2011) Designing and evaluating explanations for recommender systems. In: Ricci F, Rokach L, Shapira B, Kantor PB (eds) Recommender Systems Handbook, Springer, pp 479–510

Tishby N, Pereira FC, Bialek W (1999) The information bottleneck method. In: Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computing, Allerton, IL, USA, pp 368–377

Tkalcic M, Burnik U, Odié A, Kosir A, Tasic J (2013) Emotion-aware recommender systems - a framework and a case study. In: Markovski S, Gusev M (eds) ICT Innovations 2012, Advances in Intelligent Systems and

Computing, vol 207, Springer Berlin Heidelberg, pp 141–150

Ungar L, Foster D, Andre E, Wars S, Wars FS, Wars DS, Whispers JH (1998) Clustering methods for collaborative filtering. In: Proceedings of the AAAI Workshop on Recommendation Systems, Madison, Wisconsin, USA, AAAI Press, pp 114–129

Wang F, Pan W, Chen L (2013) Recommendation for new users with partial preferences by integrating product reviews with static specifications. In: Carberry S, Weibelzahl S, Micarelli A, Semeraro G (eds) Proceedings of the 21st International Conference on User Modeling, Adaptation, and Personalization, Rome, Italy, Springer, UMAP'13, pp 281–288

Wang Y, Liu Y, Yu X (2012) Collaborative filtering with aspect-based opinion mining: A tensor factorization approach. In: Proceedings of the IEEE International Conference on Data Mining, Brussels, Belgium, IEEE Computer Society, ICDM'12, pp 1152–1157

Wedel M, Kamakura WA (2000) Market Segmentation: Conceptual and Methodological Foundations. Kluwer Academic Publishers, 2nd Edition

Weiss SM, Indurkhya N, Zhang T, Damerau F (2005) Text Mining: Predictive Methods for Analyzing Unstructured Information. Springer

Wilson T, Wiebe J, Hoffmann P (2005) Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, Vancouver, Canada, Association for Computational Linguistics, HLT'05, pp 347–354

Wu W, Zhang B, Ostendorf M (2010) Automatic generation of personalized annotation tags for twitter users. In: Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Los Angeles, California, USA, Association for Computational Linguistics, HLT'10, pp 689–692

Yang X, Steck H, Guo Y, Liu Y (2012) On top-k recommendation using social networks. In: Proceedings of the 6th ACM International Conference on Recommender Systems, Dublin, Ireland, ACM, RecSys'12, pp 67–74

Yates A, Joseph J, Popescu AM, Cohn AD, Sillick N (2008) Shopsmart: Product recommendations through technical specifications and user reviews. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, Napa Valley, California, USA, ACM, CIKM'08, pp 1501–1502

Yu K, Zhu S, Lafferty J, Gong Y (2009) Fast nonparametric matrix factorization for large-scale collaborative filtering. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Boston, Massachusetts, USA, ACM, SIGIR'09, pp 211–218

Zhang K, Narayanan R, Choudhary A (2010) Voice of the customers: Mining online customer reviews for product feature-based ranking. In: Proceedings of the 3rd Workshop on Online Social Networks, Boston, MA, USA, USENIX Association, WOSN'10, pp 11–11

Zhang W, Ding G, Chen L, Li C, Zhang C (2013) Generating virtual ratings from chinese reviews to augment online recommendations. ACM Transactions on Intelligent Systems and Technology (TIST) 4(1):9

Zhao S, Du N, Nauerz A, Zhang X, Yuan Q, Fu R (2008) Improved recommendation based on collaborative tagging behaviors. In: Proceedings of the 13th International Conference on Intelligent User Interfaces, Gran Canaria, Canary Islands, Spain, ACM, IUI'08, pp 413–416

Ziegler CN, McNee SM, Konstan JA, Lausen G (2005) Improving recommendation lists through topic diversification. In: Proceedings of the 14th International Conference on World Wide Web, Chiba, Japan, ACM, WWW'05, pp 22–32

Zigoris P, Zhang Y (2006) Bayesian adaptive user profiling with explicit & implicit feedback. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, Arlington, Virginia, USA, ACM, CIKM'06, pp 397–404

Vitae

1. Dr. Li Chen:
   Department of Computer Science, Hong Kong Baptist University, 224 Waterloo Road, Kowloon Tong, Hong Kong
   Dr. Li Chen is Assistant Professor of Computer Science at Hong Kong Baptist University. Dr. Chen received her bachelor and master degrees in Computer Science from Peking University, China, and Ph.D. degree in Computer Science from Swiss Federal Institute of Technology in Lausanne (EPFL). Her primary interests lie in the areas of user modeling, Web personalization, recommender systems, human-computer interaction, and data mining applications. She has co-authored over 70 technical papers and has co-edited several special issues in ACM transactions.
2. Mr. Guanliang Chen:
   Department of Computer Science, Hong Kong Baptist University, 224 Waterloo Road, Kowloon Tong, Hong Kong
   Mr. Guanliang Chen received his B.E. and M.E. degrees in Software Engineering from South China University of Technology. He had been an exchange research student at Hong Kong Baptist University, under the supervision of Dr. Li Chen, from May 2013 to Jan. 2014. His primary research interests lie in the areas of personalization and recommender systems, social network, data mining, web intelligence, personalized E-learning, and human computer interaction.
3. Mr. Feng Wang:
   Department of Computer Science, Hong Kong Baptist University, 224 Waterloo Road, Kowloon Tong, Hong Kong
   Mr. Feng Wang is a Ph.D. candidate in Department of Computer Science at Hong Kong Baptist University, under the supervision of Dr. Li Chen. He received his B.E. degree in Software Engineering from Shandong University, China, in 2009. His primary research interests lie in the areas of machine learning, natural language processing, and recommender systems.