2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology

Recommending Inexperienced Products via Learning from Consumer Reviews

Feng Wang
Department of Computer Science
Hong Kong Baptist University
Hong Kong, China
fwang@comp.hkbu.edu.hk

Li Chen
Department of Computer Science
Hong Kong Baptist University
Hong Kong, China
lichen@comp.hkbu.edu.hk

***Abstract*—Most products in e-commerce are with high cost (e.g., digital cameras, computers) and hence less likely experienced by users (so they are called "inexperienced products"). The traditional recommender techniques (such as user-based collaborative filtering and content-based methods) are thus not effectively applicable in this environment, because they largely assume that the users have prior experiences with the items. In this paper, we have particularly incorporated product reviews to solve the recommendation problem. We first studied how to utilize the reviewer-level weighted feature preferences (as learnt from their written product reviews) to generate recommendations to the current buyer, followed by exploring the impact of *Latent Class Regression Models* (LCRM) based cluster-level feature preferences (that represent the common preferences of a group of reviewers). Motivated by their respective advantages, a hybrid method that combines both reviewer-level and cluster-level preferences is introduced and experimentally compared to the other methods. The results reveal that the hybrid method is superior to the other variations in terms of recommendation accuracy, especially when the current buyer states *incomplete* feature preferences.**

***Keywords*-Recommender system; inexperienced products; product reviews; weighted feature preferences; Latent Class Regression Model;**

I. Introduction

In e-commerce, due to the explosive growth of product information, a buyer often feels overwhelmed when making purchase decision. To assist her/his decision process, the recommender system can be an effective support since it aims at eliminating the information overload and returning personalized recommendations that may satisfy the user's needs. However, the recommender technologies developed so far, such as the user-based collaborative filtering technique [1] and the content-based method [2], have been broadly applied to recommend low-risk and frequently experienced products (such as music and movies) for which the current user's ratings or purchase records can be obtained. For the high-risk and inexperienced products (e.g., digital cameras and laptops), given that the user does not have much prior usage and/or purchase experiences, these technologies are limited in terms of generating proper recommendations. Therefore, the challenging issue that remains in the inexperienced product domain is how to effectively recommend products to the buyer, especially considering that her/his inherent

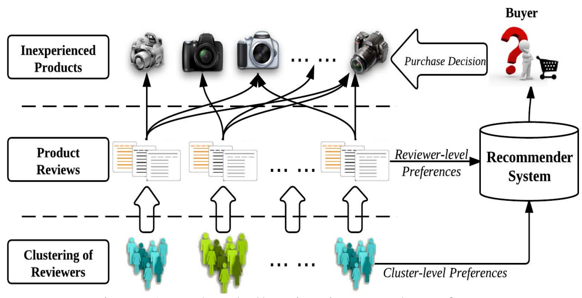

Figure 1. The challenging issue and our focus.

product preferences can be likely spread over multiple facets of the product (which are called "features" such as the camera's "battery life", "image quality", "ease of use").

On the other hand, according to the Adaptive Decision Theory [3] and our prior work on users' preference elicitation [4], [5], though the buyer's preferences over product features can be elicited (for example, the critiquing agent was developed to enable the user to explicitly state her/his feature criteria [5]), these elicited preferences were experimentally found being incomplete and uncertain when the user was searching for high-risk and unfamiliar products [4]. It hence suggests that the methods being developed for ranking high-cost products purely according to the user's stated feature preferences still cannot return accurate recommendations [6]. Thus, in this paper, we have in-depth explored the value of product reviews (which are in the form of free texts that other consumers posted to products) to address the recommending issue. Our concrete objective was to establish the similarity between the current buyer and the reviewer based on their multi-feature preferences, so that products as praised by the buyer's like-minded reviewers could be possibly taken as recommendation candidates (see Figure 1).

As a matter of fact, though recently some works have started to take into account product reviews for recommendation, they are principally limited at the following aspects: 1) most of them have targeted at deriving virtual, one-dimensional ratings from reviews via the sentimental classification techniques [7], but reviewers' multi-dimensional

978-0-7695-4880-7/12 $26.00 © 2012 IEEE
DOI 10.1109/WI-IAT.2012.209

opinions on products' features have been rarely considered. In other words, few attentions have been paid to infer the reviewer's preferences on product features from their written reviews. Another weakness of this branch of work is that the derived single ratings are not sufficient to identify preference similarity between reviewers, because in inexperience product domains, each reviewer just commented one or few products[1]. 2) In some related works, the reviews only acted as a type of supplementary info for complementing the product space. The main recommendation strategy was still the preference-based product ranking [8]. Compared to them, our contribution is that we can recover reviewers' *multi-feature preferences*, based on which we can not only enhance the similarity measure between reviewers, but also locate recommendable products that the buyer will likely truly prefer.

Therefore, in this paper, we have emphasized the usage of *feature-level review analysis results* and investigated the impact of inferring reviewers' *weighted feature preferences* to achieve the above-mentioned objective. Here, the "weight" indeed reflects the degree of importance that the reviewer places on individual feature. Technically speaking, we have explored two ways to infer the reviewer's weighted feature preferences (see Figure 1): 1) one is adopting the probabilistic regression model to learn individual reviewer's weights on features according to the review(s) that s/he wrote (so called *reviewer-level weighted preferences*); 2) another is through Latent Class Regression Model (LCRM) to identify the clusters of reviewers and *the cluster-level weighted preferences*. We have accordingly proposed different approaches to utilize these derived preference data for generating recommendations. Further driven by the respective properties and advantages of the two types of preferences, we propose a hybrid mechanism that combines the reviewer-level and cluster-level preferences together. This method was empirically proven with outperforming recommendation accuracy especially when the current buyer's stated preferences are less complete.

II. Related Work

The works most relevant to ours can be classified into two branches: one is *multi-criteria based recommender*, which is with the primary goal of addressing the single-rating induced limitations; and another is *review-based recommender* because it explicitly incorporated reviews into the recommendation. In the following, we introduce their state-of-the-art and discuss the limitations.

The traditional recommender approaches, such as collaborative filtering (CF) and content-based ones [1], [2], only consider users' single ratings on items, which however cannot reveal why users gave such ratings. Therefore, more works in recent years have attempted to reveal users' ratings on multi-facets of an item and developed the so called *multi-criteria based recommender*. For instance, in [9], classic collaborative filtering was extended by utilizing user-stated multi-criteria ratings for calculating user-user similarity. In [10], they aimed to identify the dependency structure between the overall rating and multi-criteria ratings, and utilized flexible mixture model to predict the rating for un-known items. [11] developed multi-linear singular value decomposition (MSVD) method to derive the latent relation among a user, her/his multi-criteria and an item. [12] adopted the additive utility analysis to estimate the utility of an item by integrating the marginal utilities of a user's multiple criteria on the item's attributes. In a more recent work [13], they first utilized an aggregation function to learn the significance levels of multiple criteria, and then grouped users with common significant criteria, based on which the rating of an un-known item was predicted by considering all users who are with similar significant criteria to the active user. It can hence be seen that the common objective of these methods was to estimate whether a user would be interested in an un-known item based on her/his multi-criteria ratings on a set of known items. However, these methods are unsuitable when none or few ratings can be obtained from individual user (i.e., for inexperienced products). In our work, though we have also strengthened multi-criteria, we mainly aim to infer reviewers' multi-criteria preferences from their written reviews, and then exploit such info to generate recommendation to the current buyer.

As mentioned before, another related branch of work has taken product reviews into account to offer recommendations, but the main focus was simply on enhancing traditional CF methods via deriving one-dimensional virtual ratings from reviews' sentiment classification results [7]. Few works have in-depth explored the effect of multi-dimensional feature-level sentiments on enhancing the recommender. In [14], they proposed a multi-relational matrix factorization (MRMF) method, which is an extension to low-norm matrix factorization, to model the correlation among users, movies and the opinions regarding specific features. In [15], they extracted emotions in addition to sentiments from movie reviews and plot summaries, and extended Latent Dirichlet Allocation (LDA) model in order to capture how likely a user prefers a specific movie by considering both the user's feature-level sentiments and emotions. However, these methods were mainly oriented to recommending low-risk, experienced product when users were able to provide a certain amount of ratings on known items. For high-risk products, the work done by [8] adopted the feature-level sentimental results to enrich the cameras' description, based on which the product ranking was conducted. Unfortunately, they neither evaluated the recommendation's accuracy, nor explored other possibilities, like recovering reviewers' feature preferences from the sentimental results.

[1]In our data analysis, above 69% reviewers only commented one product, and the remaining reviewers commented on average 1.9 products.

III. Problem Statement and Methods

Given the limitations of related works, we have been engaged in addressing two research questions: 1) how to recover reviewers' weighted feature preferences from their written reviews? 2) How to leverage such reviewers' preferences into computing the recommendation list so that the list can likely contain the current buyer's target choice?

Table I
NOTATIONS USED IN THIS PAPER

Notation	Description
$REV = \{rev_1, \ldots, rev_M\}$	A set of M reviewers
$\mathcal{P} = \{p_1, \ldots, p_N\}$	A set of N products
$\mathcal{S} \subseteq REV \times \mathcal{P}$	A set of reviewer-product pairs, where $(rev_i, p_j) \in S$ indicates that a reviewer rev_i wrote review to a product p_j
$\mathcal{F} = \{f_1, \ldots, f_n\}$	The n features extracted from reviews
r_{ij}	The review written by reviewer rev_i on product p_j
R_{ij}	The overall rating reviewer rev_i gave to product p_j
$\boldsymbol{X}_{ij} = [x_{ij1}, \ldots, x_{ijn}]$	The set of opinion values each of which is associated to the corresponding feature f_i in $\mathcal{F}$ as extracted from a review r_{ij}
$\boldsymbol{W}_{rev_i} = [w_{i1}, \ldots, w_{in}]$	The reviewer rev_i's weighted preferences, where each w_{il} is the weight on feature $f_l \in \mathcal{F}$, which could be None if the reviewer did not express any opinions on that feature
$\mathcal{C} = \{c_1, \ldots, c_K\}$	The K clusters of reviewers

Here, we assume every reviewer (including the buyer) inherently has a weighted preference model, which is formally denoted as: $Pref_u = \{< f_i, w_{ui} > | 1 \leq i \leq n\}$, where w_{ui} indicates the importance degree that the user u places on feature f_i. The preference structure is theoretically grounded on the Multi-Attribute Utility Theory (MAUT) because it can explicitly consider trade-offs among attributes (i.e., features in our term) via the *weights* [16]. In this paper, we have investigated two approaches to infer reviewers' weighted feature preferences: 1) rebuilding individual reviewer's weighted feature preferences from her/his written reviews, and 2) inferring cluster-level preferences that represent the common criteria of a group of reviewers (see Figure 2 with the two work flows). Table I summarizes the notations used throughout the paper.

A. Mining Feature-Opinion Pairs from Product Reviews

As shown in Figure 2, the first step is mining feature-opinion pairs from product reviews. In the past decade, more efforts have been devoted to conduct document-level (or called sentiment classification) and feature-level opinion mining (or called sentimental analysis) in the areas of natural language processing and data mining [17], [18].

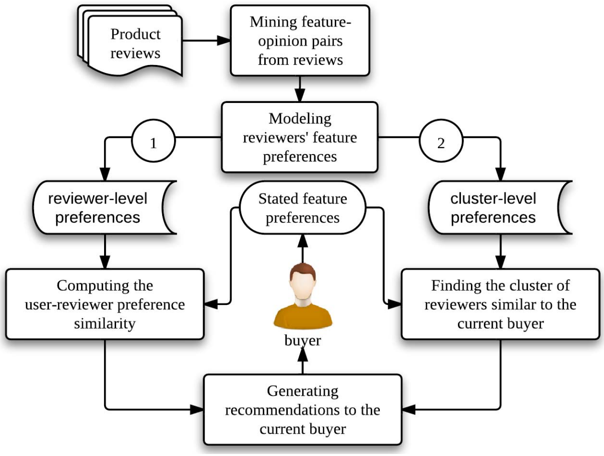

Figure 2. Inferring two types of preferences from product reviews.

Particularly, the feature-level opinion mining can return a set of $< feature, opinion >$ pairs from a review, where *opinion* indicates positive, neutral, or negative sentiment that a reviewer expressed on the *feature*. Therefore, our work can be regarded as the extension to their work, with the particular emphasis on refining the sentiment analysis results and further exploiting them to derive a reviewer's preferences on product features.

To identify feature-opinion pairs, we first used a Part-of-Speech (POS) tagger [19] to extract the frequent nouns and noun phrases from reviews, which are the prospective feature candidates. Moreover, considering that reviewers often use different words or phrases to indicate the same product feature (e.g., "picture", "image" and "appearance"), we manually defined a set of seed words and then grouped synonymous ones by computing their lexical similarity to the seed words. The lexical similarity is concretely determined via WordNet [20].

We then extracted opinion(s) that is associated to each feature in a review sentence. Most of existing works depended on the co-occurrence of product features and opinion bearing words for this purpose [17]. However, these methods cannot identify opinions that are not so "close to" the feature. Therefore, we used a syntactic dependency parser [2], because it can return the syntactic dependency relations between words in a sentence. For example, after parsing the sentence "*it takes great photos and was easy to learn how to use*", "great" is identified with dependency relation AMOD with "photos" (meaning that "great" is an adjectival modifier of the noun word "photos"), and *"easy"* has COMP dependency relation with *"learn"* (indicating that "easy" is an open clausal complement of "learn"). In another example "*the photos are great*", "great" has NSUBJ relation with "photos" (indicating that "photos" is the subjective of

[2]http://nlp.stanford.edu/software/lex-parser.shtml

"great"). Thus, we took all the words with such relations with the product feature words as opinions.

After identifying the *feature-opinion pairs* from a review sentence, the next task was to assess the opinion's sentiment strength (also called polarity). For this purpose, we applied SentiWordNet [21] because it provides us with a triple of polarity scores: positivity, negativity and objectivity, respectively denoted as $\{Pos(s), Neg(s), Obj(s)\}$, and each ranging from 0.0 to 1.0; for each opinion word, $Pos(s) + Neg(s) + Obj(s) = 1$. The triple scores are then merged into a single sentiment value for the opinion word s: $O_s = Neg(s)*R_{min}+Pos(s)*R_{max}+Obj(s)*\frac{R_{min}+R_{max}}{2}$ where R_{min} and R_{max} represent the minimal and maximal rating scales respectively (we set them as $R_{min} = 1$, $R_{max} = 5$ so that O_s ranges from 1 to 5). If there are negation words (e.g., not, don't, no, didn't) in a sentence, the polarity of related opinions is reversed.

We then aggregated all opinion words' sentiment values in relation to a feature in a review. Instead of using the simple arithmetic mean which is common in related methods [22], we performed a weighted average by which each opinion word's sentiment value also behaves as the weight, so that the extremely positive or negative polarizations are less susceptible to shift. For instance, if two opinion words, "good" and "great" are associated to a feature, the feature's final opinion value is $(\frac{4\times4+5\times5}{4+5} = 4.55)$ where 4 and 5 are the sentiment values of the two words ("good" and "great") respectively.

B. Modeling Reviewer-Level Weighted Feature Preferences

Then, with the pairs of *features* and *opinions* as extracted from every reviewer's written review(s), we first developed the method for inferring *reviewer-level weighted feature preference*. Basically, the idea behind our approach is that the overall rating that each reviewer gave to a product can be considered as the weighted sum of her/his opinions on different features, based on which we could learn the reviewer's weights placed on these features.

Formally, to derive reviewer rev_i's weighted preferences $\boldsymbol{W}_{rev_i}$ on product features $\mathcal{F}$, we applied probabilistic regression model for learning the weights [23]. Specifically, given features' opinion values $\boldsymbol{X}_{ij} \in \mathbb{R}^n$ in respect of a product p_j commented by the reviewer rev_i, her/his overall rating R_{ij} can be drawn from a Gaussian distribution around $\boldsymbol{W}_{rev_i}^T\boldsymbol{X}_{ij}$:

$$Pro(R_{ij}|\boldsymbol{W}_{rev_i}, \boldsymbol{X}_{ij}, \sigma^2) = \mathcal{N}(R_{ij}|\boldsymbol{W}_{rev_i}^T\boldsymbol{X}_{ij}, \sigma^2) \quad (1)$$

where $\boldsymbol{W}_{rev_i}$ can be formally represented by a Multivariate Gaussian Distribution, $\boldsymbol{W}_{rev_i} \sim \mathcal{N}(\mu, \Sigma)$, with μ as the mean and Σ as the covariance matrix. We additionally incorporated the occurrence frequency of a feature in a review (denoted as μ_0) into the model, being the prior of μ. It can be essentially used to define the distributions of μ and Σ based on Kullback-Leibler (KL) divergence:

$$Pro(\mu, \Sigma) = \exp[-\psi \cdot KL(Q(\mu, \Sigma)||Q(\mu_0, I))] \quad (2)$$

where $KL(\cdot, \cdot)$ is the KL divergence, $Q(\mu, \Sigma)$ denotes a multivariate gaussian distribution, and ψ is a tradeoff parameter ($\psi = 100$ in our experiment).

The probability that an overall rating R_{ij} accompanying a review r_{ij} posted to a product p_j can be hence as:

$$\begin{aligned} Pro(R_{ij}|\Psi, r_{ij}) = \int &(Pro(R_{ij}|\boldsymbol{W}_{rev_i}, \boldsymbol{X}_{ij}, \sigma^2) \\ &\cdot Pro(\boldsymbol{W}_{rev_i}|\mu, \Sigma) \cdot Pro(\mu, \Sigma))d\boldsymbol{W}_{rev_i} \end{aligned} \quad (3)$$

Because the reviewer's overall rating is known, $\Psi=\{\boldsymbol{W}_{rev_1}, \ldots, \boldsymbol{W}_{rev_M}, \mu, \Sigma, \sigma^2\}$ contains the model parameters that can be estimated by performing the maximum log-likelihood (ML) method. Through identifying the optimal Ψ^* for maximizing the log-likelihood $\Psi^* = \arg\max_\Psi \sum_{(rev_i,p_j)\in S} \log Pro(R_{ij}|\Psi, r_{ij})$, we obtained the optimal values for $\boldsymbol{W}_{rev_i}$ $(1 \leq i \leq M)$, which are the weighted preferences of reviewer rev_i on features $\mathcal{F}$.

C. Modeling Cluster-Level Weighted Feature Preferences

As another, alternative way to derive reviewers' preferences, we adopted the Latent Class Regression Model (LCRM) to identify the clusters of reviewers and the cluster-level weighted feature preferences. The system can then map the current buyer to the most relevant cluster in order to locate product candidates for the recommendation. According to [24], LCRM has the theoretical advantage over traditional clustering analysis (like k-Means) for performing market segmentation, because it could more precisely divide the users into clusters based on their membership probabilities (that is, a user is assigned to a cluster only when this assignment has the highest probability). However, this model has been rarely investigated in the research field of recommender systems, in terms of its potential effect on enhancing the similarity measure among users (e.g., reviewers in our case). Thus, in this paper, we have particularly fused the clusters as being resulted from LCRM into the review-based recommendation process.

Specifically, assume that the population of reviewers can be divided into K clusters $\mathcal{C} = \{c_1, c_2, \ldots, c_K\}$, according to LCRM, if a reviewer rev_i belongs to the cluster c_k, the conditional probability of her/his overall rating R_{ij} on a product should be defined as:

$$Pro(R_{ij}|\boldsymbol{X}_{ij}, c_k) = \mathcal{N}(R_{ij}|\boldsymbol{W}_{c_k}^T\boldsymbol{X}_{ij}, \sigma^2) \quad (4)$$

where $\boldsymbol{X}_{ij}$ is the set of opinion values being associated to the set of features (see the definition in Table 1), and $\boldsymbol{W}_{c_k} \in \mathbb{R}^n$ is the cluster's weighted feature preferences.

Because the overall rating is known, the above formula can be based to calculate the probability that a reviewer belongs to a cluster. Formally, a reviewer rev_i is placed in a cluster c_k if $q_k(rev_i) > q_h(rev_i) \, \forall c_k \neq c_h \in \mathcal{C}$ where

$$q_k(rev_i) = \prod_{(rev_i,p_j)\in S} \frac{\pi_k \cdot Pro(R_{ij}|\boldsymbol{X}_{ij}, c_k)}{\sum_{c_h\in\mathcal{C}} \pi_h \cdot Pro(R_{ij}|\boldsymbol{X}_{ij}, c_h)} \quad (5)$$

In the above formula, $q_k(rev_i)$ is the posterior probability that a reviewer rev_i belongs to a cluster c_k, and π_k is the prior probability. The full mixture likelihood can be accordingly defined as:

$$\mathcal{L}(\psi|\mathcal{S}) = \prod_{(rev_i,p_j)\in\mathcal{S}} \sum_{k=1}^{K} q_k(rev_i)\cdot Pro(R_{ij}|\boldsymbol{X}_{ij},c_k) \tag{6}$$

The parameter set $\psi = \{\pi_1,\ldots,\pi_K,\boldsymbol{W}_{c_1},\ldots,\boldsymbol{W}_{c_K}\}$ is estimated by the Expectation-Maximization (EM) algorithm, which seeks to identify the maximized log-likelihood by iteratively applying the two steps: 1) Expectation step (E step) which updates the posterior probability that a reviewer belongs to a certain cluster and derives the prior cluster probability as $\hat{\pi}_k = \frac{\sum_{i=1}^{M} q_k(rev_i)}{M}$; 2) Maximization step (M step) which aims to find the optimal parameter values of $\hat{\psi}$ for maximizing Eq. 6: $\hat{\psi} = \max_{\psi}\mathcal{L}(\psi|\mathcal{S})$.

E- and M-steps are repeated until Eq. 6 converges. As a result, all reviewers are automatically classified into K disjoint clusters and each cluster is returned with the weighted feature preferences $\boldsymbol{W}_{c_k}$ that represent the common preferences of reviewers within that cluster. In the experiment, we found that this approach is not sensitive to the initial assignment of each reviewer's membership, since we got similar recommendation results with different initialization.

D. Generating Recommendation

1) Recommending based on Reviewer-Level Preferences: As mentioned before, the current buyer's feature preferences can be elicited through some existing preference elicitation methods [5], [25], which can be formally represented as $\boldsymbol{W}_u$=$\{w_{f_i}|i\in\{1,2,\ldots,n\}\}$. The value of w_j is set to zero if the corresponding feature's weight is not explicitly stated by the buyer. To utilize the reviewer-level preference data being obtained from Section III-B, we have tried various approaches. In the following, we describe three typical methods, among which the first one was originated from [8].

1. Preference-based Product Ranking (PPR-Rec)

This method was taken as the baseline in this paper to be compared with other methods. It primarily used reviewers' feature opinions to determine the product's feature space. Concretely, there is a feature score computed for each feature of a product, by aggregating the feature's opinions extracted from the product's reviews:

$$FeatureScore_{f_l}(p_j) = \frac{\sum_{(rev_i,p_j)\in\mathcal{S}} x_{ijl}}{m} \tag{7}$$

where x_{ijl} denotes the feature f_l's opinion value in review r_{ij} to product p_j, and m denotes the number of reviews being associated to the product p_j. Thus, the matching score of each product to the buyer's preferences is computed as:

$$ProductScore(u,p_j) = \sum_{w_{f_l}(u)\in\boldsymbol{W}_u} w_{f_l}(u)\times FeatureScore_{f_l}(p_j) \tag{8}$$

in which $w_{f_l}(u)$ denotes the buyer u's weight on feature f_l. Then, the top-N products with higher scores are recommended to the buyer (in our experiment, we tested the algorithm's performance when $N = 10, 20, 30$).

2. k-NN based Recommending (k-NN-Rec)

In this method, given the buyer's currently stated preferences $\boldsymbol{W}_u$, we aim at first identifying a set of reviewers $\mathcal{K}$ who have similar feature preferences to the buyer. The similarity is formally computed as:

$$sim(\boldsymbol{W}_u,\boldsymbol{W}_{rev_i}) = \frac{1}{1+\sqrt{\sum_{w_{f_l}\in\boldsymbol{W}_u}(w_{f_l}(u)-w_{f_l}(rev_i))^2}} \tag{9}$$

A prediction score is then computed for each product p_j:

$$PredictionScore(u,p_j) = \frac{\sum_{rev_i\in\mathcal{K}} sim(\boldsymbol{W}_u,\boldsymbol{W}_{rev_i})\times R_{ij}}{\sum_{rev_i\in\mathcal{K}} sim(\boldsymbol{W}_u,\boldsymbol{W}_{rev_i})} \tag{10}$$

where R_{ij} is the overall rating that reviewer rev_i gave to product p_j. Still, top-N products with higher scores are recommended to the buyer.

3. k-Means based Recommending (k-Means-Rec)

In this approach, we considered using the traditional clustering technique like the k-Means to divide the reviewers into K disjoint clusters $\{c_1,\ldots,c_K\}$. We then retrieve the cluster that is most relevant to the buyer. Concretely, during conducting k-Means clustering, a reviewer will be moved from one cluster to another if this process could minimize its squared distance to the cluster's "centroid" (the centroid is denoted as $\boldsymbol{W}_{c_k\_centroid}$).

The buyer's preferences are based to compute her/his distance to all clusters' centroids, and the cluster with the shortest distance is matched to the buyer. The distance between the buyer u_i and the centroid of c_k is defined as: $1/sim(\boldsymbol{W}_u,\boldsymbol{W}_{c_k\_centroid})$, in which the similarity $sim(\boldsymbol{W}_u,\boldsymbol{W}_{c_k\_centroid})$ is computed by replacing $\boldsymbol{W}_{rev_i}$ with $\boldsymbol{W}_{c_k\_centroid}$ in Eq. 9. Afterwards, the products as reviewed by this cluster of reviewers are taken as recommendation candidates. Each product p_j is further computed with a prediction score:

$$PredictionScore(u,p_j) = \frac{\sum_{rev_i\in c_l\wedge(rev_i,p_j)\in\mathcal{S}} sim(\boldsymbol{W}_u,\boldsymbol{W}_{rev_i})\times R_{ij}}{\sum_{rev_i\in c_l\wedge(rev_i,p_j)\in\mathcal{S}} sim(\boldsymbol{W}_u,\boldsymbol{W}_{rev_i})} \tag{11}$$

where c_l denotes the cluster of reviewers that is most relevant to the buyer, R_{ij} is the overall rating that a reviewer rev_i gave to the product p_j, and $sim(\boldsymbol{W}_u,\boldsymbol{W}_{rev_i})$ is the preference similarity between the buyer u and the reviewer rev_i, which is computed via Eq. 9. Top-N products with higher prediction scores are finally returned to the buyer as the recommendations.

2) Recommending based on Cluster-Level Preferences: On the other hand, with the cluster-level preferences generated from Section III-C, the similarity between the buyer and a cluster of reviewers, in this case, is formally computed by replacing $\boldsymbol{W}_{rev_i}$ with $\boldsymbol{W}_{c_k}$ in Eq. 9, as follows:

$$sim(\boldsymbol{W}_u,\boldsymbol{W}_{c_k}) = \frac{1}{1+\sqrt{\sum_{w_{f_l}\in\boldsymbol{W}_u}(w_{f_l}(u)-w_{f_l}(c_k))^2}} \tag{12}$$

where $w_{f_l}(c_k)$ denotes the weight on feature f_l regarding cluster c_k. In the following, we concretely describe two variations of utilizing the cluster-level preferences: one is called *hard matching based*, because it retrieves a single cluster among all to relate it to the buyer; another is called *fuzzy matching based*, because multiple clusters are taken into account, but assigned different weights to represent their various similarity degrees to the buyer.

1. LCRM-based Hard Cluster Matching (LCRM-Hard-Rec)

In this method, all clusters are first ranked according to their similarity scores as computed via Eq. 12, and the top one with the highest similarity is chosen. The products reviewed by this cluster of reviewers are then taken as recommendation candidates. Each product p_j is computed with a prediction score:

$$PredictionScore(u,p_j)=\frac{\sum_{rev_i\in c_l(p_j)} R_{ij}}{|c_l(p_j)|} \tag{13}$$

where $c_l(p_j)$ denotes the set of reviewers within the chosen cluster c_l who wrote reviews to product p_j, and R_{ij} is the overall rating on product p_j given by reviewer rev_i. The products which obtain higher prediction scores will be finally offered to the buyer as the recommendation.

It can be seen that because the LCRM-based algorithm cannot derive reviewer-level weighted preferences, the similarity between the buyer and individual reviewer is not taken into account in the above formula. It also implies that if two buyers are matched to the same cluster, the recommended products to them will be the same. Given this disadvantage, we have further proposed the fuzzy matching method.

2. LCRM-based Fuzzy Cluster Matching (LCRM-Fuzzy-Rec)

In this approach, instead of matching the buyer to a single cluster, s/he can be mapped to multiple clusters if the user-cluster similarity exceeds a threshold (which is empirically set as 0.8). The similarity between the buyer and the cluster (via Eq. 12) is then treated as a weight when calculating the prediction score of each product:

$$PredictionScore(u,p_j)=\frac{\sum_{c_l\in C} sim(\boldsymbol{W}_u,\boldsymbol{W}_{c_l})\times(\frac{\sum_{rev_i\in c_l(p_j)} R_{ij}}{|c_l(p_j)|})}{\sum_{c_l\in C} sim(\boldsymbol{W}_u,\boldsymbol{W}_{c_l})} \tag{14}$$

where $\mathcal{C}$ denotes the set of satisfying clusters.

Therefore, compared to the hard matching approach, not only more products (as reviewed by multiple clusters' reviewers) are considered as recommendation candidates, but also the inherent weakness of hard matching method is avoided given that different buyers should have different similarity degrees to the clusters (if their stated preferences were different).

E. Hybrid of Reviewer-Level and Cluster-Level Preferences based Recommending (Hybrid-Rec)

To take advantage of both reviewer-level and cluster-level preferences, we have further developed a hybrid method for which the latent class regression model is adopted to cluster reviewers, and reviewer-level preferences (as learnt via the probabilistic regression model; see Section III-B) are based to calculate the user-reviewer similarity when predicting the score for a product. The potential benefit of this combination is thus that, not only it could be more accurate to identify the common preferences among reviewers at the group level, but also to reflect the preference heterogeneity between reviewers when matching them individually to the current buyer.

In more detail, we first apply the LCRM approach (see Section III-C) to generate the clusters of reviewers. Then, when the cluster that is best mapping to the buyer is identified (through LCRM-Hard-Rec), we adopt the formula (similar to Eq. 11) to compute the weighted prediction score of a product. Here, the weight is the similarity degree between the buyer and every reviewer (within the chosen cluster), as computed by Eq. 9, for which the reviewer-level preferences are produced with the method in Section III-B. The top-N products with higher prediction scores are then recommended to the buyer.

IV. EXPERIMENT

Therefore, in the experiment, six review-based recommending approaches were compared: three are based on reviewer-level preferences, two based on LCRM and cluster-level preferences, and the hybrid method. In addition, we implemented a non-review baseline which is without the fusion of reviews (shorted as *Non-Review*), with the purpose to verify the actual effect of incorporating feature-level review analysis results on enhancing the recommendation accuracy. Concretely, Non-Review is simply based on the product's static features (i.e., technical specifications). Given the buyer's stated preferences $\boldsymbol{W}_u$, the matching score $ProductScore(u,p_j)$ of each product p_j is computed via:

$$ProductScore(u,p_j)=\sum_{w_{f_l}(u)\in \boldsymbol{W}_u} w_{f_l}(u)\times s_{f_l}(p_j) \tag{15}$$

where p_j is the product, and $s_{f_l}(p_j)$ is the value of each static feature f_l which is normalized in the range of $[0,1]$. The top-N products with higher scores are included in the recommendation list.

A. Experiment Setup and Evaluation Metrics

After filtering out invalid reviews which are too short or with meaningless characters, we gathered 7485 digital camera reviews from www.buzzillions.com. These reviews cover 186 digital cameras, and every review on average encompasses 4.7 distinct features (st.d. = 1.91). Assuming that every buyer has a "target choice" (which is the target product s/he is prepared to buy), the experimental goal was to evaluate whether this target choice could be located in the recommendation list by the tested algorithm. For this goal, we adopted the *leave-one-out* evaluation scheme [26]. That is, during each round, we excluded one reviewer from the dataset, who behaved as a simulated buyer. However, as not all of reviewers' commented products can be taken as their target choices, we performed testing only on reviewers who gave full marks to a product (indicating that this product is her/his best choice). Accordingly, 1705 reviewers were selected as the "buyers". At a time, one of them was tested, and her/his stated feature preferences are the ones inferred from his/her review(s). In order to additionally simulate the *incomplete preferences* situation, we randomly chose subsets of the reviewer's full feature preferences ($\boldsymbol{W}_{rev_i}$) to represent the buyer's different preference completeness degrees: one assessment was conducted with the *relative completeness* degree (i.e., with 40%, 60%, 80% and 100% of $\boldsymbol{W}_{rev_i}$ to be the buyer's preferences $\boldsymbol{W}_u$); and another was with the *absolute completeness* degree (i.e., over absolute 4, 6, 8, and 10 features, for which 10 indicates the full feature set).

The recommendation accuracy was measured by two metrics: 1) *H@N* (HitRatio@top-N recommendations) which refers to the percent of successes that a buyer's target choice appears in the top-N recommendation list: $H@N = \frac{\text{\#The number of successes within the top-N}}{\text{\#The total number of testings}}$; 2) *Percentile* (shorted as *Per*) which gives the percent of products which are ranked below the target choice among all alternatives [27]: $Per = \frac{\sum_{t=1}^{T} \frac{N - Rank_{target\_choice}}{|\mathcal{P}|}}{T}$ in which, $|\mathcal{P}|$ is the total number of products, and T is the number of testings.

B. Results Analysis

Table II first shows the comparison results, at *relative* preference completeness variations (i.e., from 40% to 100%). First of all, it can be seen that though PPR-Rec does not show obvious advantage against Non-Review, the other two review-based methods (k-NN-Rec and k-Means-Rec) clearly outperform the two baselines (PPR-Rec and Non-Review) in terms of both hit ratio and percentile. It hence suggests that PPR-Rec, that simply utilizes feature-opinion pairs to enrich the product's feature space for the product ranking, cannot fully fulfill the value of reviews to enhance recommendation. In comparison, k-NN-Rec (for which $|\mathcal{K}| = 2000$ through the experimental trials) and k-Means-Rec (for which the number of clusters was set as 6) are more accurate, given that they are mainly targeted to identify a group of reviewers who possess feature-level preference similarity to the current buyer. The comparison between them further shows that k-Means-Rec is more accurate than k-NN-Rec. This result implies that the pre-cluster of reviewers according to their feature preferences can be more likely to identify the like-minded reviewers to the buyer, relative to the on-site retrieval of neighborhood purely based on the buyer's stated preferences (i.e., k-NN-Rec).

As for LCRM-based methods (for which the best number of clusters is also 6), the fuzzy method (LCRM-Fuzzy-Rec) outperforms all other methods (including k-NN-Rec and k-Means-Rec) particularly when the buyer's preferences are less complete (i.e., 40%, 60% and 80%) under H@20. However, both hard matching (LCRM-Hard-Rec) and fuzzy matching (LCRM-Fuzzy-Rec) methods do not show better performance, especially against k-Means-Rec, in respect of other measures. The result might be caused by their lacking of the similarity matching between the buyer and single reviewer. In fact, when the reviewer-level preferences were integrated with the LCRM-based cluster-level preferences (i.e., Hybrid-Rec), it achieves the highest accuracy at varied preference completeness degrees among all approaches.

The above findings hence highlight the impact of LCRM based clustering, relative to k-Means based one, on identifying like-minded reviewers. The results also reveal the importance of incorporating individual reviewers' weighted feature preferences, so as to adjust her/his actual contribution when optimizing the recommendation accuracy. Table III additionally gives the comparison results at varied *absolute* preference completeness degrees, from which it can be seen that the five review-based recommendation methods are still more accurate than the baseline approaches (i.e., Non-Review and PPR-Rec). In addition, the clustering-based methods such as k-Means-Rec and LCRM-Fuzzy-Rec perform more effective than non-clustering involved, and Hybrid-Rec is shown most accurate among all when the user's preferences were less complete.

V. CONCLUSIONS

In conclusion, this paper presented a novel review-based recommendation framework for aiding buyers' decision making in inexperienced product domains. Particularly, it aims at recovering reviewers' *weighted feature preferences* from their written textual reviews and exploiting such preference data to generate recommendation. To achieve this goal, we first investigated the respective roles of reviewer-level preferences (as learnt from probabilistic regression model) and cluster-level preferences (as inferred through the Latent Class Regression Model (LCRM)), respectively. The two types of preferences were concretely fused into the recommendation process via different methods (i.e., k-NN based and k-Means based for fusing reviewer-level preferences, and hard & fuzzy cluster matching methods for fusing cluster-level preferences). A hybrid mechanism that combines them together was further developed and compared to ones that incorporated them separately. These review-based methods were also compared to related works. The experimental results show the outperforming accuracy of review-based methods, especially the hybrid method (Hybrid-Rec), in the condition that the buyer's stated preferences were less complete. This finding suggests that Hybrid-Rec can be more effective to support new buyers. The results also imply that the system could be more intelligent to adjust its recommendation strategy in condition of the buyer's preference completeness degree. In the future, we will endeavor to research this adaptive improvement.

REFERENCES

[1] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in

Table II
COMPARISON OF ALGORITHMS' RECOMMENDATION ACCURACY WITH DIFFERENT **RELATIVE** PREFERENCE COMPLETENESS DEGREES

	40% complete				60% complete				80% complete				100% complete			
Method	$H@10$	$H@20$	$H@30$	Per	$H@10$	$H@20$	$H@30$	Per	$H@10$	$H@20$	$H@30$	Per	$H@10$	$H@20$	$H@30$	Per
Non-Review	0.053	0.063	0.157	0.484	0.059	0.065	0.157	0.486	0.071	0.101	0.162	0.492	0.084	0.120	0.145	0.486
PPR-Rec	0.059	0.059	0.170	0.502	0.048	0.058	0.177	0.694	0.054	0.064	0.178	0.513	0.043	0.043	0.175	0.518
k-NN-Rec	0.146	0.198	0.324	0.692	0.196	0.201	0.334	0.690	0.186	0.213	0.345	0.700	0.206	0.261	0.345	0.703
k-Means-Rec	0.188	0.193	0.335	0.592	0.234	0.234	0.385	0.643	0.270	0.281	0.416	0.690	0.300	0.312	0.469	0.710
LCRM-Hard-Rec	0.110	0.175	0.225	0.534	0.129	0.192	0.248	0.553	0.142	0.210	0.268	0.551	0.143	0.191	0.244	0.546
LCRM-Fuzzy-Rec	0.130	0.300	0.364	0.634	0.138	0.300	0.367	0.633	0.137	0.298	0.365	0.631	0.154	0.288	0.373	0.630
Hybrid-Rec	**0.229**	0.245	**0.409**	**0.697**	**0.261**	0.269	**0.456**	**0.660**	0.247	0.254	**0.428**	**0.717**	0.242	0.253	0.423	0.706

Table III
COMPARISON OF ALGORITHMS' RECOMMENDATION ACCURACY WITH DIFFERENT **ABSOLUTE** PREFERENCE COMPLETENESS DEGREES

	4 features				6 features				8 features				10 features			
Method	$H@10$	$H@20$	$H@30$	Per	$H@10$	$H@20$	$H@30$	Per	$H@10$	$H@20$	$H@30$	Per	$H@10$	$H@20$	$H@30$	Per
Non-Review	0.053	0.127	0.160	0.497	0.062	0.096	0.125	0.500	0.037	0.051	0.065	0.512	0.039	0.062	0.104	0.420
PPR-Rec	0.052	0.117	0.169	0.518	0.020	0.049	0.110	0.516	0.020	0.049	0.130	0.516	0.029	0.029	0.058	0.549
k-NN-Rec	0.177	0.257	0.348	0.546	0.163	0.243	0.327	0.550	0.165	0.330	0.422	0.509	0.103	0.285	0.326	0.427
k-Means-Rec	0.225	0.320	0.367	0.602	0.281	0.369	0.425	0.622	0.224	0.292	0.380	0.630	0.268	0.268	0.292	0.660
LCRM-Hard-Rec	0.189	0.274	0.354	0.616	0.231	0.350	0.363	0.678	0.117	0.312	0.374	0.719	0.160	0.361	0.371	0.620
LCRM-Fuzzy-Rec	0.201	0.261	0.360	0.610	0.246	0.250	0.318	0.606	0.106	0.223	0.363	0.621	0.100	0.300	0.362	0.630
Hybrid-Rec	**0.226**	0.296	**0.437**	**0.654**	0.251	**0.389**	**0.439**	**0.684**	0.110	0.329	0.407	0.676	0.160	**0.370**	**0.403**	0.657

Proc. 14th UAI, vol. 461, no. 8. San Francisco, CA, 1998, pp. 43–52.

[2] M. Balabanović and Y. Shoham, "Fab: content-based, collaborative recommendation," in *Commun. ACM*, vol. 40, March 1997, pp. 66–72.

[3] J. Payne, J. Bettman, and E. Johnson, *The Adaptive Decision Maker*. Cambridge University Press, May 1993.

[4] L. Chen and P. Pu, "Evaluating critiquing-based recommencler agents," in *Proc. AAAI '06*, vol. 1, 2006, pp. 157–162.

[5] ——, "Hybrid critiquing-based recommender systems," in *Proc. IUI '07*, 2007, pp. 22–31.

[6] C. Schmitt, D. Dengler, and M. Bauer, "Multivariate preference models and decision making with the maut machine," in *User Modeling 2003*, ser. Lecture Notes in Computer Science, 2003, vol. 2702, pp. 148–148.

[7] W. Zhang, G. Ding, L. Chen, C. Li, , and C. Zhang, "Generating virtual ratings from chinese reviews to fuse into collaborating filtering algorithms," in *ACM TIST*, 2012.

[8] S. Aciar, D. Zhang, S. Simoff, and J. Debenham, "Informed recommender: Basing recommendations on consumer product reviews," in *IEEE Intel. Syst.*, vol. 22, 2007, pp. 39–47.

[9] G. Adomavicius and Y. Kwon, "New recommendation techniques for multicriteria rating systems," in *IEEE Intelligent Systems*, vol. 22, May 2007, pp. 48–55.

[10] R. D. N. Sahoo, R. Krishnan and J. P. Callan, "Collaborative filtering with multi-component rating for recommender systems," in *Proc. 6th WITS*, 2006.

[11] Q. Li, C. Wang, and G. Geng, "Improving personalized services in mobile commerce by a novel multicriteria rating approach," in *Proc. 17th WWW*. ACM, 2008, pp. 1235–1236.

[12] K. Lakiotaki, N. F. Matsatsinis, and A. Tsoukias, "Multicriteria user modeling in recommender systems," in *IEEE Intelligent Systems*, vol. 26, March 2011, pp. 64–76.

[13] L. Liu, N. Mehandjiev, and D. L. Xu, "Multi-criteria service recommendation based on user criteria preferences," in *Proc. RecSys '11*. ACM, 2011, pp. 77–84.

[14] N. Jakob, S. H. Weber, M. C. Müller, and I. Gurevych, "Beyond the stars: exploiting free-text user reviews to improve the accuracy of movie recommendations," in *Proc. TSA '09*. ACM, 2009, pp. 57–64.

[15] Y. Moshfeghi, B. Piwowarski, and J. M. Jose, "Handling data sparsity in collaborative filtering using emotion and semantic based features," in *Proc. SIGIR '11*, 2011, pp. 625–634.

[16] R. L. Keeney, H. Raiffa, and D. W. Rajala, "Decisions with multiple objectives: Preferences and value trade-offs," in *IEEE Trans. on SMC*, vol. 9, no. 7, July 1979, p. 403.

[17] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proc. KDD '04*. ACM, 2004, pp. 168–177.

[18] L. Qi and L. Chen, "Comparison of model-based learning methods for feature-level opinion mining," in *Proc. WI '11*, vol. 1, 2011, pp. 265–273.

[19] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proc. HLT-NAACL*, 2003.

[20] *WordNet An Electronic Lexical Database*. The MIT Press, May 1998.

[21] A. Esuli and F. Sebastiani, "Sentiwordnet: A publicly available lexical resource for opinion mining," in *Proc. LREC'06*, 2006, pp. 417–422.

[22] M. Taboada, C. Anthony, and K. Voll, "Methods for creating semantic orientation dictionaries," in *Proc. 5th LREC*, 2006, pp. 427–432.

[23] J. Neter, M. H. Kutner, C. J. Nachtsheim, and W. Wasserman, *Applied Linear Statistical Models*. Chicago: Irwin, 1996.

[24] M. Wedel and W. A. Kamakura, "Market segmentation - conceptual and methodological foundations," in *Target Meas Anal Mark*, vol. 9, no. 1, 2000, pp. 92–93.

[25] L. Chen and P. Pu, "Survey of preference elicitation methods." Technical Report No. IC/200467, EPFL, July 2004.

[26] B. Smyth, L. McGinty, J. Reilly, and K. McCarthy, "Compound critiques for conversational recommender systems," in *Proc. WI '04*, 2004, pp. 145–151.

[27] A. Yates, J. Joseph, A. M. Popescu, A. D. Cohn, and N. Sillick, "Shopsmart: product recommendations through technical specifications and user reviews," in *Proc. CIKM '08*, 2008, pp. 1501–1502.