Knowledge-Based Systems 50 (2013) 44–59

Contents lists available at SciVerse ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Preference-based clustering reviews for augmenting e-commerce recommendation

Li Chen *, Feng Wang

Department of Computer Science, Hong Kong Baptist University, Hong Kong, China

ARTICLE INFO

Article history:
Received 12 December 2012
Received in revised form 9 May 2013
Accepted 9 May 2013
Available online 30 May 2013

Keywords:
Recommender system
Product reviews
Opinion mining
Multi-attribute utility theory
Preference learning
Latent class regression model
Clustering
E-commerce

ABSTRACT

In the area of e-commerce, there exists a special, implicit community being composed of *product reviewers*. A reviewer normally provides two types of info: one is the overall rating on the product(s) that s/he experienced, and another is the textual review that contains her/his detailed opinions on the product(s). However, for the high-risk products (such as digital cameras, computers, and cars), a reviewer usually commented one or few products due to her/his infrequent usage experiences. It hence raises a question of how to identify the preference similarity among reviewers. In this paper, we propose a novel clustering method based on *Latent Class Regression model (LCRM)*, which is essentially able to consider both the overall ratings and feature-level opinion values (as extracted from textual reviews) to identify reviewers' preference homogeneity. Particularly, we extend the model to infer individual reviewers' *weighted feature preferences* within the same iterative process. As a result, both the cluster-level and reviewer-level preferences are derived. We further test the impact of these derived preferences on augmenting recommendation for the active buyer. That is, given the reviewers' feature preferences, we aim to establish the connection between the active buyer and the cluster of reviewers by revealing their preferences' inter-relevance. In the experiment, we tested the proposed recommender algorithm with two real-world datasets. More notably, we compared it with multiple related approaches, including the non-review based method and non-LCRM based variations. The experiment demonstrates the superior performance of our approach in terms of increasing the system's recommendation accuracy.

1. Introduction

Due to the explosive growth of information appearing in current Web, recommender systems have been widely developed in recent years for eliminating the information overload and providing personalized item recommendation to the user. So far, most of recommender systems, such as user-based collaborative filtering techniques [19], content-based methods [35], and matrix factorization approaches [27], have been built under the assumption that a sufficient amount of user ratings on known-items can be easily obtained (based on which the system can infer the user's preferences and identify user-user similarity). However, in the e-commerce environment especially with the so called *high-risk products* (also called *high-cost* or *high-involvement* products, such as digital cameras, computers, and cars), because a user does not buy the high-risk product very often, it is normal that s/he is not able to rate many products. For the same reason, the current buyer is often a new user because s/he would not afford to buy the same kind of high-risk product before. These phenomena can be supported by the statistics reported in [23,53]: a great portion of users (e.g., >68% in Amazon dataset and >90% in resellerratings.com dataset) only gave feedback to one product. It is hence infeasible to purely adopt the classical recommending methods to benefit users in the high-risk product domains.

To solve the "new user" problem, related works have attempted to elicit the buyer's preferences on site by asking her/his to explicitly state the preferences over the product's features (e.g., the laptop's processor speed, memory capacity, screen size, etc.). The preference model is theoretically based on the Multi-Attribute Utility Theory (MAUT) [25], according to which all products can be ranked by their matching utilities with the user's stated preferences. However, though it is possible to obtain the buyer's needs via interactive preference elicitation techniques (such as the critiquing agent [7] that will be described in Section 2.1), the elicited preferences are still less complete and accurate, given the fact that the buyer cannot state her/his full preferences when s/he is confronted with the costly, unfamiliar products. This phenomenon was principally formulated in the field of decision theory as a type of adaptive, constructive decision behavior [34].

* Corresponding author. Address: 224 Waterloo Road, Kowloon Tong, Kowloon, Hong Kong, China. Tel.: +852 34117090; fax: +852 34117892.

E-mail addresses: lichen@comp.hkbu.edu.hk (L. Chen), fwang@comp.hkbu.edu.hk (F. Wang).

0950-7051/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.knosys.2013.05.006

Therefore, with the objective of developing more effective recommender system in e-commerce, in this paper, we have exerted to exploit the deep value buried in *product reviews* as contributed by other users to benefit the current new buyer. Particularly, we are interested in deriving the reviewers' feature preferences from the textual reviews they posted. Given that the review-item matrix is sparse, we have attempted to cluster reviewers into preference-based communities and simultaneously adjust individual reviewers' preferences. Such derived data can then be potentially helpful to predict the current buyer's missing preferences and enable the system to return more accurate recommendation. The main contributions of our work are indeed that: (1) we propose a *preference-driven approach* for learning reviewers and extracting clusters; (2) we build the relevance between the current buyer and reviewers based on *their feature-level preference similarity*; and (3) we develop the *review-based recommendation method* to address the "new user" issue.

Specifically, we aim to construct a reviewer's *weighted feature preferences* from her/his written review(s): $Pref_u = \{\langle f_i, w_{ui}\rangle | 1 \leqslant i \leqslant n\}$, where w_{ui} denotes the *weight* that reflects the importance degree that the user u places on feature f_i. An example is given below to illustrate the problem.

EXAMPLE. Reviewer *A* wrote a review to camera *C*, and her overall rating to this product is 5 (in the range of 1–5).

> *"It can produce a great image in low light environment. You can usually use it in AUTO mode and expect a good result. If you don't mid a little bit heavier and bigger camera compared with most of compact cameras, this is the one you should get it. Only con I can think of is its little bit short battery life. Better to consider to buy an additional battery."*

The question is then how the system could automatically derive the reviewer's weight preferences on the features that she mentioned in the above review (e.g., *which feature(s) is more important to her?*). It might be intuitive to count the feature's occurring frequency, so that if a feature appears more frequently, it is regarded more important than others [2,28]. However, this method cannot distinguish features which are with equal occurrences. Moreover, in the cases like the above example, the less frequent feature "image" is actually more important than the feature "battery life" because its opinion is consistent with the reviewer's overall rating on the product (both are positive) while the battery life's opinion is negative. It hence suggests that the user's overall rating along with her/his opinions on different features should be all considered so as to potentially more accurately reveal her/his weights on those features.

In this paper, we have first applied the Probabilistic Regression Model (PRM) to identify the relationship between the overall rating and features' opinion values for every reviewer (see Section 4). We have accordingly proposed PRM based k-NN & k-Means recommending methods and experimentally proved that the PRM-based methods perform more accurate than the non-review based method (that is without the incorporation of any reviews) and the non-preference based method (that does not stress deriving reviewers' weight preferences). However, the PRM-based methods might be still limited in the situation with sparse reviews (i.e., one or few reviews provided by each reviewer). We have thus endeavored to additionally improve the stability of derived reviewers' weight preferences by involving the clustering process. The basic idea is that, with all reviewers' info (i.e., the overall ratings and features' opinion values), we first conduct unsupervised clustering to group these reviewers, which is targeted to build the *cluster-level preferences* to represent a cluster of reviewers' common preferences. At the same time, we employ the cluster-level preferences to adjust individual reviewer's weight preferences (i.e., *reviewer-level preferences*). During the next iterative cycle, the reviewer-level preferences are further used to polish the clustering results. Such iteration will end when both types of preferences are stabilized and not changed. To accomplish these tasks, we have particularly extended *Latent Class Regression Model (LCRM)*. LCRM has been widely applied in the marketing area to perform the market segmentation (i.e., dividing prospective buyers into subsets who share preference homogeneity) [52]. Its main property is that it can take into account the whole structure of the targeted domain to divide a number of entities into latent classes, and enable each class to contain entities which inherently possess similar regression characteristics (in our case, the regression defines the relationship between the overall rating and features' opinion values). To suit our needs, we have modified the original form of LCRM so that both cluster-level regression model (i.e., with the cluster-level preferences as the outcome) and reviewer-level regression (i.e., with the reviewer-level preferences as the outcome) can be simultaneously generated and optimized. This proposed method is called LCRM*, which is new relative to our previous work [49,50]. Moreover, we have evaluated the algorithm's accuracy and compared it to other related ones on two real-world datasets.

In the following, we will first summarize related works, by classifying them into two categories: the recommender systems developed for high-risk product domains, and the review-based recommender systems (Section 2). After discussing their respective limitations, we start to describe our system's workflow and approaches (Section 3). The methods based on the Probabilistic Regression Model (PRM) will be in detail introduced in Section 4, and the methods based on the extended Latent Class Regression Model (LCRM) will be presented in Section 5. The experiment setup, evaluation metrics and results analysis will then follow (Section 6). At the end, we conclude the major findings (Section 7).

2. Related work

2.1. Recommender systems for high-risk products

As mentioned before, for high-risk products, because it is unusual to obtain a number of ratings on many products from a single user, researchers have mainly put focus on developing effective preference elicitation techniques for solving the "new user" problem.

Related works on preference elicitation. Preference elicitation is a process engaging users in some kind of "dialog" with the system [5]. The traditional methods typically involved users in a tedious and time-consuming procedure. For example, in [25], every attribute's utility function is assessed through the mid-value splitting technique. That is, given a range of attribute value $[x_a, x_b]$, the user is first asked to specify a mid-value point x_c for which the pairs (x_a, x_c) and (x_c, x_b) are differentially value-equivalent. Therefore, if $U(x_a) = 0$, $U(x_b) = 1$ (i.e., the point's utility), it infers that $U(x_c) = 0.5$. The utilities on other points can be similarly estimated (for example, finding mid-value points respectively for $[x_a, x_c]$ and $[x_c, x_b]$). Then, the pairs of products which have indifferent preferences are used to derive the trade-offs (i.e., relative weights) among attributes. Analytic hierarchy process (AHP) is an alternative elicitation method [41]: through a series of pairwise comparisons between products, it obtains the weights of decision criteria (i.e., the attributes) and the value function of each attribute. However, as most of users cannot clearly answer these questions upfront especially in complex decision environments [34,48], in recent years, some researchers