Inferring users’ multi-attribute preferences from the reviews for augmenting recommender systems in e-commerce

Feng Wang

Follow this and additional works at: https://repository.hkbu.edu.hk/etd_oa

Recommended Citation
Wang, Feng, "Inferring users' multi-attribute preferences from the reviews for augmenting recommender systems in e-commerce" (2016). Open Access Theses and Dissertations. 336.
https://repository.hkbu.edu.hk/etd_oa/336

This Thesis is brought to you for free and open access by the Electronic Theses and Dissertations at HKBU Institutional Repository. It has been accepted for inclusion in Open Access Theses and Dissertations by an authorized administrator of HKBU Institutional Repository. For more information, please contact repository@hkbu.edu.hk.
DATE: September 15, 2016

STUDENT'S NAME: WANG Feng

THESIS TITLE: Inferring Users' Multi-attribute Preferences from their Reviews for Augmenting Recommender Systems in E-commerce

This is to certify that the above student's thesis has been examined by the following panel members and has received full approval for acceptance in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Chairman: Prof. Liao Lizhi
Professor, Department of Mathematics, HKBU
(Designated by Dean of Faculty of Science)

Internal Members: Dr. Cheung Kwok Wai
Associate Professor, Department of Computer Science, HKBU
(Designated by Head of Department of Computer Science)

Prof. Xu Jianliang
Professor, Department of Computer Science, HKBU

External Members: Prof. Cai Yi
Professor and Associate Head
Department of Software Technology
South China University of Technology
China

Dr. Zhang Hui
Associate Professor
Computer Science & Technology Programme, Division of Science and Technology
United International College
Guangdong Province, China

In-attendance: Dr. Chen Li
Assistant Professor, Department of Computer Science, HKBU

Issued by Graduate School, HKBU
Inferring Users’ Multi-Attribute Preferences from Their Reviews for Augmenting Recommender Systems in E-commerce

WANG Feng

A thesis submitted in partial fulfilment of the requirements for the degree of
Doctor of Philosophy

Principal Supervisor: Dr. Li Chen

Hong Kong Baptist University

September 2016
DECLARATION

I declare that this thesis represents my own work, except where due acknowledgement is made, and that it has not been previously included in a thesis, dissertation or report submitted to this University or to any other institution for a degree, diploma or other qualifications.

Signature:  

Date:    September, 2016
ABSTRACT

By now, people are accustomed to getting some personalized recommendations when they are finding movies to watch, music to listen, and so on. All of these recommendations come from recommender systems, and can aid the process of the decision making to avoid the problem of “information overload”. Over the years, there has been much work done both in industry and academia on developing new approaches for recommender systems. However, there are still some hurdles in adapting recommender systems to a broader range of real-life applications. 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. The traditional recommender techniques (such as user-based collaborative filtering and content-based methods) can thus not be effectively applicable in this environment, because they largely assume that the users have prior experiences with products. Thus, the “data sparsity” and “new users” are two typical challenging issues that the classical recommender systems cannot well address in high-risk product domains. In some recommender systems, a new user will be asked to indicate his/her preferences on some aspects in order to address the so called cold-start problem via collecting some preferences. Such collected preferences are usually not complete due to unfamiliarity with the product domain, which are called partial preferences.

In this thesis, we propose to leverage some auxiliary data of online reviewers’ opinions, so as to enrich the partial preferences. With the objective of developing more effective rec-
omnender systems for high-risk products in e-commerce, in our work, we have exerted to derive reviewers’ preferences from the textual reviews they posted. Then, these recovered preferences are leveraged to estimate and supplement a new buyer’s preference with which the product recommendation is produced. Firstly, we propose a novel clustering method based on Latent Class Regression model (LCRM), which is able to consider both the overall ratings and feature-level opinion values (as extracted from textual reviews) to infer individual reviewers’ weight feature preferences, that represent the weights the user places on different product features. Secondly, we propose a method to estimate reviewers’ value preferences (i.e., the user’s preferences on the product’s attribute values) by matching their review opinions to the corresponding attributes’ static specifications. Thirdly, we investigate how to combine weight preferences and value preferences to model user preferences based on Multi-Attribute Utility Theory (MAUT) with the purpose of providing higher quality product recommendations. Particularly, it was shown from our experimental studies that the incorporation of review information can significantly enhance the recommendation accuracy, relative to those without considering reviews. As the practical implication, our proposed solutions can be usefully plugged into an online system to be adopted in real-ecommerce sites.

**Keywords:** Recommender System, Product Reviews, High-Risk Products, Partial Preferences, E-Commerce
ACKNOWLEDGMENTS

I’m deeply grateful to my supervisor, Dr. Li Chen for the guidance, patience and support she has provided throughout my time as her student. I consider myself very fortunate for having been able to work with a very considerate and encouraging supervisor like her. My thesis would not be completed without her supervision. I would also like to thank Hong Kong Baptist University for providing a wonderful educational environment and funding which allowed me to undertake this research. More thanks go to all my friends for keeping me positive and helping me get through these years. Finally, I would like to give special thanks go to my parents for their support and encouragement during the years of study.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>i</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>iii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>viii</td>
</tr>
<tr>
<td>List of Tables</td>
<td>ix</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Research Background</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Problem Definitions and Notations</td>
<td>5</td>
</tr>
<tr>
<td>1.3 Major System Components</td>
<td>8</td>
</tr>
<tr>
<td>1.4 Objective and Main Contributions</td>
<td>9</td>
</tr>
<tr>
<td>1.5 Overview of the Dissertation</td>
<td>10</td>
</tr>
<tr>
<td>2 State of Art</td>
<td>12</td>
</tr>
<tr>
<td>2.1 Recommender Systems in High-Risk Product Domains</td>
<td>12</td>
</tr>
<tr>
<td>2.1.1 Preference Elicitation Techniques</td>
<td>12</td>
</tr>
<tr>
<td>2.1.2 Decision Support in High-risk Product Domains</td>
<td>14</td>
</tr>
<tr>
<td>2.1.3 Limitations</td>
<td>15</td>
</tr>
</tbody>
</table>
4 Estimating Multi-Attribute Value Preferences from User Reviews

4.1 Introduction

4.2 Methodology

4.2.1 Problem Statement

4.2.2 Multi-attribute Value Preferences Estimation

4.2.3 Recommendation Generation

4.3 Experiment

4.3.1 Experimental Setup and Dataset

4.3.2 Evaluation Metric

4.3.3 Compared Baseline Methods

4.3.4 Results Analysis

4.4 Summary

5 Deriving New Users’ Multi-Attribute Preference Model

5.1 Introduction

5.2 Methodology

5.2.1 Aspect-based Ratings and Weights Estimation

5.2.2 New Users’ Multi-Attribute Preferences Estimation

5.2.3 Recommendation Generation

5.3 Experiment

5.3.1 Experimental Setup and Dataset

5.3.2 Compared Methods and Evaluation Metrics

5.3.3 Results Analysis
5.4 Summary .............................................................. 94

6 Conclusion .......................................................... 95
6.1 Contributions ................................................... 96
6.2 Limitations ......................................................... 97
6.3 Future Work ......................................................... 98
6.4 Summary .......................................................... 100

Bibliography ......................................................... 100

A Reprint Reuse Permissions of Published Articles ............. 113
A.1 Preference-based clustering reviews for augmenting e-commerce recommenda-
......................... tion ......................................................... 113
A.2 Recommendation for New Users with Partial Preferences by Integrating
Product Reviews with Static Specifications ......................... 119
A.3 Recommending Inexperienced Products via Learning from Consumer Re-
views ......................................................... 123

Curriculum Vitae .................................................. 125
List of Figures

1.1 A digital camera review example from Amazon. .......................... 4
1.2 The system-user interaction model of our review-based recommender systems in high-risk product domains. .......................... 9
2.1 Limitation of related recommender systems in high-risk product domains. ......................................................... 15
3.1 Work flow of system with two major branches of proposed approaches which result in five different recommending methods. ............... 30
3.2 Comparison among PRM-k-NN, PRM-k-Means, LCRM*-k-NN, and LCRM*-k-Means, w.r.t. H@10. .................. 57
3.3 Comparison between LCRM*-k-Means and LCRM*, w.r.t. H@10. ...... 57
4.1 Illustration of multi-attribute value preferences completion procedure. ...... 67
5.1 The inter-connected three tasks related to aspect-based opinion mining. ......................................................... 74
5.2 The graphical plate notation for our CARW model. ......................................................... 77
5.3 The preference statement interface in the digital camera recommender system. ......................................................... 89
List of Tables

1.1 Notations used in this thesis ........................................ 7
2.1 Summary of typical works based on feature-level review analysis .... 22
3.1 Summary of developed methods’ properties .......................... 47
3.2 Description of two datasets (digital camera and laptop) used in Chapter 3 48
3.3 List of tested algorithms in the experiment .......................... 52
3.4 Comparison of algorithms w.r.t. Hit Ratio and MRR with varied preference sizes in digital camera dataset (the maximal preference size is 10) .... 54
3.5 Comparison of algorithms w.r.t. Hit Ratio and MRR with varied preference sizes in laptop dataset (the maximal preference size is 8) .... 55
4.1 The attributes and associated value levels for digital cameras ......... 63
4.2 The representation of the structured product reviews ............... 65
4.3 The recommendation performance of CompleteRank and baselines. Note that for PartialRank, HybridRank and CompleteRank, we randomly take 2, 4, 6 attributes for five times to simulate partial preferences, and the average performance of those five runs are reported ....................... 70
5.1 Descriptions of two datasets used in two recommender systems .... 86
5.2 Seed words for digital camera reviews used to guide CARW model learning

5.3 Seed words for laptop reviews used to guide CARW model learning

5.4 Comparison of algorithms w.r.t. Hit Ratio and MRR with varied preference sizes for digital camera recommender system (the maximal preference size is 6)

5.5 Comparison of algorithms w.r.t. Hit Ratio and MRR with varied preference sizes for laptop recommender system (the maximal preference size is 7)
Chapter 1

Introduction

1.1 Research Background

In reality, with the increasing widespread use of the Internet, there has been tremendous growth in electronic commerce (e-commerce) conducted over the Internet. However, due to the explosive growth in the number of items/products (i.e., songs, movies, and digital cameras etc.) on the current Web, people often feel overwhelmed when making decisions about which item is suitable for them. Hence, the recommender systems (RS) have been widely developed in recent years to help people make decisions without exploring all of the available items. Such systems help people to deal with information overload by autonomously gathering information and proactively tailoring it to individual interests, e.g., what product to buy (Amazon), what song to listen (Last.fm), what hotel to stay (TripAdvisor), and so on.

Research works in recommender systems have recently attracted attention in both academia and industry. However, there are still some hurdles in adapting recommender systems to a broader range of real-life applications. So far, most of recommender systems, such as the user/item-based collaborative filtering approaches (Herlocker et al., 2002),
the content-based approaches (Pazzani and Billsus [2007]), and matrix factorization approaches (Koren et al., 2009), 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. In fact, these methods have been broadly applied to recommend low-cost, frequently experienced products such as books, movies and songs, since users are able to provide enough ratings to such type of items. 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 statistically supported by the analysis conducted in (Jindal and Liu, 2008, Xie et al., 2012): a great portion of users (e.g., > 68% in Amazon.com dataset and > 90% in Resellerratings.com dataset) only gave feedback to one product. Thus, the “data sparsity” and “new user” are two typical challenging issues that the classical recommender systems cannot well address in high-risk product domains.

To solve “data sparsity” and “new user” problems, related works have attempted to elicit the buyer’s preferences on site by asking her/him to explicitly state the preferences for the product’s attributes (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) (Keeney et al., 1979), according to which all products can be ranked by their matching utilities with the user’s stated preferences for the product’s attributes. However, though it is possible to obtain the buyer’s needs via interactive preference elicitation techniques (such as the critiquing agent introduced by Chen and Pu (2009)), the elicited
preferences are still less complete and accurate, given the fact that the buyer cannot state her/his complete 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 behaviour (Payne et al., 1993). Therefore, purely relying on their self-stated preferences to produce recommendations might not fit their true interests.

In recent years, more researchers have recognized that the social content (i.e., other consumers’ generated data) essentially plays important role in influencing users’ choices (Lackermair et al., 2013, Hu et al., 2008). Among various types of social content, the product reviews are more prevalent as they can reflect other users’ opinions on the product given their post-purchase evaluation experiences (Lackermair et al., 2013). Empirical findings from marketing also support our observation that “online product reviews have become a major information source for consumers and marketers regarding product quality” (Hu et al., 2008). Recently, some related works have been proposed to fuse product reviews into recommender systems, which can be named as review-based recommender systems. For example, Zhang et al. (2013a) proposed to derive virtual ratings according to reviewers’ sentiment classification results (i.e., the review is overall towards positive, neutral or negative), so as to compensate the rating sparsity limitation of collaborative filtering systems. However, these works neglect the opinions expressed on various product features that can be extracted from reviews. To overcome this limitation, some of related works have in-depth explored the effect of multi-dimensional feature-level sentiments (i.e., opinions on various features as contained in reviews) on enhancing recommenders. For example, in (Jakob et al., 2009), they proposed a multi-relational matrix factorization (MRMF) method, which is an extension to low-norm matrix factorization, to model the correlations
among users, movies and opinions regarding specific features. In (Moshfeghi et al., 2011), 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 products when users are able to provide a certain amount of ratings on known items. For high-risk products, the work done by (Aciar et al., 2007) adopted the feature-level sentiment results to enrich 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 sentiment results.

![Figure 1.1: A digital camera review example from Amazon.](image)

Take a real digital camera review from Amazon as example (see Figure 1.1), besides the overall rating, the author (i.e., reviewer) also posted textual reviews to express his detailed opinions on the camera’s features (e.g., “image quality”, “battery life”, and “ease of use”). It hence sounds feasible to infer the reviewer’s preference for each indicated attribute (or called aspect) from her/his review. Thus, taking into account of valuable data
that though are hidden in reviews can be usefully utilized to construct a user’s preferences and hence improve the product recommendations. Our goal is to recover these reviewers’ multi-attribute preferences, which can then be potentially helpful to predict the current buyer’s missing preferences and enable the system to return more accurate recommendations.

In summary, this thesis studies the solutions of developing effective methods for recovering reviewers’ preferences given their reviews, which can then be utilized to construct a buyer’s preferences to enhance product recommendations.

1.2 Problem Definitions and Notations

Given that most products in e-commerce are described by multiple attributes (for example, a camera is described by digital zoom, battery, face detection quality, screen size, etc.), it is essential to build a multi-attribute preference model for each user so as for the recommender system to locate best matching products. This type of recommender system has commonly be named as preference-based or knowledge-based system (Felfernig et al., 2006). The challenging question then comes to how to construct users’ preferences in an unobtrusive and accurate way.

In order to simplify the complexity of user preference structures and elicitation process, we have modeled user preferences based on the Multi-Attribute Utility Theory (MAUT) (Keeney et al., 1979). The fundamental assumption in this utility theory is that a decision maker always chooses the alternative for which its expected utility is maximal, based on which we could predict a choice that the decision maker will make within a range of alternatives. Thus, in order to assign a utility to each product, we may adopt MAUT as it is a structured methodology inherently in accordance with compensatory decision strategy for processing
products with multiple attributes. Grounded on MAUT, the numerical order of multiple products’ utilities can preserve the decision maker’s preference order among them. If we assume preferential and additive independence, the product utility can be defined as a weighted additive form:

\[ U_i(x) = w_i^T v_i(x) = \sum_{j} w_{ij} \times v_{ij}(x_j) \]  

(1.1)

where \( U_i(x) \) is the utility of a product with attribute values \( x = \{x_j\}_{j=1}^A \) for user \( u_i \), and \( A \) is the number of product attributes. As illustrated in the above equation, the MAUT based user’s preference structure consists of two components: weight preferences \( w_i = \{w_{i1}, w_{i2}, \ldots, w_{iA}\} \) and value preferences \( v_i = \{v_{i1}, v_{i2}, \ldots, v_{iA}\} \).

**Weight preferences.** Weight is an important component of users’ preference structure, as it represents the relative importance of each attribute to the user. It enables users to solve preference conflict explicitly by trading off values among different attributes. Formally, \( w_{ij} \) denotes an attribute \( a_j \)’s relative importance (i.e., weight) to user \( u_i \).

**Value preferences.** In addition to weight, there should be a value function \( v_{ij}(\cdot) \) defined for each attribute \( a_j \), which indicates the user’s preference structure provided that \( x' \sim x'' \iff v(x') = v(x'') \) and \( x' \succ x'' \iff v(x') > v(x'') \).

In many e-commerce websites, users are encouraged to write reviews in form of free texts to products that they have experienced. These textual comments normally emphasize why they like or dislike a product with opinions expressed on different attributes of the product. Thus, in this thesis, we will be engaged in obtaining users’ accurate preferences by particularly considering valuable information hidden in product reviews. Formally, in this thesis, we assume that we have a set of \( U \) users, which can be denoted as \( U = \{u_1, \ldots, u_U\} \), and a set of \( P \) products (such as laptops or digital cameras), which can be
Table 1.1: Notations used in this thesis

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{U} = {u_1, \ldots, u_U}$</td>
<td>the set of users (reviewers), and $U$ is the number of users.</td>
</tr>
<tr>
<td>$\mathcal{P} = {p_1, \ldots, p_P}$</td>
<td>the set of products, and $P$ is the number of products.</td>
</tr>
<tr>
<td>$\mathcal{R} = {r_{ij}</td>
<td>u_i \in \mathcal{U} \text{ and } p_j \in \mathcal{P}}$</td>
</tr>
<tr>
<td>$r_{ij}$</td>
<td>the review written by user $u_i$ for product $p_j$.</td>
</tr>
<tr>
<td>$y_{ij} \in \mathbb{R}^+$</td>
<td>the overall rating associated with review $r_{ij}$.</td>
</tr>
<tr>
<td>$\mathcal{A} = {a_1, \ldots, a_A}$</td>
<td>the set of product attributes, and $A$ is the number of attributes.</td>
</tr>
<tr>
<td>$x_i = {x_{ik}}_{k=1}^A$</td>
<td>the set of attribute values for product $p_j$, and $x_{ik}$ is the value of attribute $a_k$ of product $p_j$.</td>
</tr>
<tr>
<td>$w_u = {w_{uk}}_{k=1}^A$</td>
<td>the set of user’s stated weight preferences.</td>
</tr>
<tr>
<td>$\phi_u = {\phi_{uk}}_{k=1}^A$</td>
<td>the set of user’s stated value preferences, e.g., $weight &lt; 200$.</td>
</tr>
</tbody>
</table>

denoted as $\mathcal{P} = \{p_1, \ldots, p_P\}$. Then, we let $\mathcal{R} = \{r_{ij} | u_i \in \mathcal{U} \text{ and } p_j \in \mathcal{P}\}$ be a set of reviews that have been posted for certain products. Typically, when writing a review $r_{ij}$, the user $u_i$ also assigns an overall rating $y_{ij} \in \mathbb{R}^+$ (say from 1 to 5) to express the overall quality of the reviewed product $p_j$. The major notations used throughout the thesis can be found in Table 1.1. It is worth to note that different notations are used for distinguishing “reviewers” and “the current buyer”, i.e., $u_i (1 \leq i \leq U)$ for the $i$-th reviewer and $\hat{u}$ for the current buyer. The research problems that we have been engaged in solving are as follows:

- **How to estimate the reviewers’ weight preferences given their reviews?** 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. However, this method
cannot distinguish features which are with equal occurrences. The user’s overall rating along with her/his opinions on different features should all be considered so as to potentially more accurately reveal her/his weights on those features.

- **How to estimate the reviewers’ value preferences given their reviews?** Though we know that the camera’s weight is very important to a user, it is still vague about what value the user would accept. Intuitively, the opinions expressed in reviews could reveal the reviewers’ preferences on product attributes.

- **After obtaining reviewers’ weight and value preferences given their reviews, how to use them to enrich the current buyer’s partial preferences, so as for the system to generate appropriate recommendations to the buyer?**

### 1.3 Major System Components

Our review-based recommender systems for high-risk products can be described by a generic model as illustrated in Figure 1.2.

A new buyer first interacts with the recommender system by specifying a set of preferences on certain attributes (i.e., initial preference). Based on the stated initial preferences, the system identifies a set of like-minded reviewers with similar preferences. Then, the current buyer’s missing preferences will be predicted based on the preferences of the like-minded reviewers. The last step is to recommend products to the current buyer based on the predicted preferences.
1.4 Objective and Main Contributions

The ultimate goal of our work is therefore to build an efficient high-risk product recommender system that utilizes online reviews to 1) overcome the limitations (i.e., data sparsity and new user) of the exiting recommenders, and 2) offer higher quality of product recommendations. The main contributions of this thesis are summarized as follows:

1. We envisioned product reviews as valuable resource of other users’ preference information to enrich the current buyer’s preferences.
2. We studied how to leverage reviewers’ aspect-based opinions, by mapping them to attributes’ values (i.e., static specifications), for aiding the product recommendation.

3. We built the relevance between the current buyer and reviewers based on their preference similarity, which is then used to locate a set of like-mined reviewers given the buyer’s initial preferences.

4. We developed different review-based recommendation methods to address the “new user” issue by fusing of recovered reviewers’ preferences.

5. We conducted extensive experiments on real-world data, which shows the outperforming accuracy of our approaches against several baseline methods for digital camera and laptop recommendations.

1.5 Overview of the Dissertation

This chapter introduced the background and motivation of our research. Two problems we have been engaged in addressing: 1) how to model the reviewers’ preference given their reviews, and 2) how to utilize the reviewers’ preference to improve the high-risk product recommendations. The organization of the rest content is listed as follows:

Chapter 2 discusses the related work on recommender systems in high-risk product domains. It then introduces the state-of-the-art on review-based recommender systems and their limitations.

Chapter 3 proposes a novel clustering method based on Latent Class Regression model (LCRM), which is able to consider both the overall ratings and feature-level opinion values
CHAPTER 1. INTRODUCTION

(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. The experiment demonstrates the superior performance of our approach in terms of increasing the system’s recommendation accuracy.

Chapter 4 proposes a method to leverage some auxiliary data of reviewers’ feature-level opinions, so as to predict the buyer’s missing value preferences. Experiment on a real user-study data and a crawled Amazon review data shows that our solution achieves better recommendation performance than several baseline methods.

Chapter 5 describes a unified framework for addressing three inter-connected opinion mining tasks: 1) identifying the aspects mentioned in reviews of a product, 2) estimating the rating of each aspect based on the sentiments expressed in reviews, and 3) estimating the aspect-based weights that reflect how much reviewers care about different aspects of a product. Then, we design a preference enrichment approach via incorporating the reviewers’ aspect-based weights and opinions on products’ attributes. The completed preferences of a new user are then used to match the product profiles, by which the products with highest matching scores are recommended to the user. Experimental results show that our solution can provide more accurate personalized recommendations than baseline methods.

Chapter 6 concludes the main contributions of this thesis, and discusses the limitations of our work with the aim to further improve our recommender technologies.
Chapter 2

State of Art

2.1 Recommender Systems in High-Risk Product Domains

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.

2.1.1 Preference Elicitation Techniques

Preference elicitation is a process engaging users in some kind of “dialog” with the system (Chen and Pu, 2004). The traditional methods typically involved users in a tedious and time-consuming procedure. For example, in Keeney and Raiffa (1976), 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 firstly 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 (Saaty, 1977): 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 (Payne et al., 1993; Viappiani et al., 2006).

In recent years, some researchers have emphasized the incremental preference elicitation techniques to obtain users’ product preferences. Specifically, they first acquired the user’s initial preferences (that might be likely uncertain and incomplete) and then stimulated the user to participate in the conversation with the system so as to discover her/his hidden needs. The representative work is the so called critiquing-based recommender system (Chen and Pu, 2012) which has emerged in the form of both natural language models (Shimazu, 2002; Thompson et al., 2004) and graphical user interfaces (Burke et al., 1997; Reilly et al., 2005; Pu and Chen, 2005). The main interaction component of these systems is that of recommendation-and-critiquing, by which users can provide feedback to the recommended item by choosing to improve some of its feature values and compromising others (e.g., “I would like something lighter and with faster processor speed”). The feedback, in turn, enables the system to more accurately predict what the user truly wants and accordingly recommend some products that may better interest the user in the next conversational cycle. One typical supporting mechanism, so called system-suggested critiquing, pro-actively generates a set of knowledge-based critiques that users may be prepared to accept as ways to improve the current product, which has been adopted in the FindMe system for proposing unit critiques (Burke et al., 1997) and in the Dynamic-Critiquing agent for presenting compound critiques (Reilly et al., 2005). An alternative critiquing
mechanism provides a facility to stimulate users to freely create critiques on their own, which was implemented in the Example Critiquing agent (Pu and Chen, 2005). Previous works proved that the critiquing support allows users to obtain higher decision accuracy and preference certainty, in comparison with non critiquing-based systems.

2.1.2 Decision Support in High-risk Product Domains

In addition to the works mentioned above, some systems have also aimed at supporting users’ decision in high-risk product domains, that include knowledge-based recommenders (Felfernig et al., 2006), case-based reasoning systems (Burke et al., 1997, Bridge et al., 2005), and utility-based recommenders (Huang, 2011, Stolze and Ströbel, 2003). For example, in (Felfernig et al., 2006), the product’s matching degree with the user’s preferences was determined via a knowledge base which comprises the domain restrictions (e.g., “the camera with higher optical zoom should be preferred to the one with lower zoom”). The recommendation problem was then transformed into the constraint satisfaction problem, for which the user’s preferences were formulated as hard and/or soft constraints. In case-based reasoning systems, the recommended products were retrieved via searching for ones that are most similar to what the user preferred before. For instance, in (Burke et al., 1997), the recommendation of restaurants in a new city is based on what the user knows and likes in other cities. The utility-based recommender system was explicitly built on the Multi-Attribute Utility Theory (MAUT) to model the user’s preferences as a weighted additive form under the assumption of additive independence (Huang, 2011). The products with higher utility scores are recommended to the user.
2.1.3 Limitations

However, though it is feasible to obtain the current buyer’s preferences via the aforementioned elicitation techniques, our previous empirical studies pointed out that the elicited user preferences are unlikely certain and complete in complex and unfamiliar conditions (Chen and Pu, 2009, Pu and Chen, 2005). It is thus ineffective to purely base on what the user stated to determine the product’s true matching degree. Moreover, the related decision support systems have primarily used the product’s static specifications to model the current buyer’s preferences, while neglecting the potential usefulness of incorporating other users’ generated contents like product reviews (see Figure 2.1).

Figure 2.1: Limitation of related recommender systems in high-risk product domains.

2.2 Review-based Recommender Systems

As a matter of fact, the growing popularity of social and e-commerce media sites has encouraged users to freely write reviews to describe their assessment of products or services in a human-nature way. These reviews are usually in the form of textual comments that
CHAPTER 2. STATE OF ART

contain users’ opinions of the products based on their usage experiences. From technical perspective, the reviews might compensate for the limitations of classical recommender techniques such as content-based and collaborative filtering-based methods, since they essentially reflect users’ actual opinions on multi-aspect of products and disclose why a user likes or dislikes a product, instead of purely measuring how much the user likes a product. Therefore, more and more works have attempted to propose effective approaches to discovering the values hidden in reviews for practically benefiting recommender systems.

Indeed, because the raw review information is in un-structured, textual form which cannot be directly understood by recommender systems, the most challenging issue is how to interpret reviews and extract useful information from them for serving the recommendation. More specifically, there are several critical issues that researchers have been engaged in addressing:

1. **What kind of review elements could be servable for recommender systems?** The traditional non-review recommender systems mostly neglect that the user’s profile could be built based on reviews s/he posted to products. Actually, not only the textual characteristics of reviews might be utilizable, but also the fine-grained opinions expressed on product features might be employed to understand why and how the author commented on that product. Therefore, according to the type of review information that existing review-based recommender systems have utilized, there are two major branches, respectively text-based approaches that directly harness the textual content of reviews to enrich the recommendation process (Esparza et al., 2010; Raghavan et al., 2012; McAuley and Leskovec, 2013), and opinion-based approaches that attempt to bring the reviews’ opinion mining (also called sentiment analysis) results into play (Zhang et al., 2013a; Aciar et al., 2007; Dong et al., 2016).
For the latter branch, we can further divide the related approaches into the following two sub-categories in terms of actual opinions they have considered:

- **Review-level opinion**: an overall opinion that is estimated based on the whole review to indicate the reviewer’s overall sentiment orientation (i.e., positive or negative) towards the item. It essentially measures the user’s perception of the product from a holistic perspective;

- **Feature-level opinion**: opinions associated with different features of the reviewed item. For example, in the review sentence “*the software that comes with it is amazing*”, “amazing” is a positive opinion associated with the feature “software”. In raw reviews, the feature is normally expressed as noun or noun phrase, which may refer to any distinct objects such as the item itself (e.g., “hotel”), its component (e.g., “bedroom”, “bathroom”), its function (e.g., “service”), or the component’s (or function’s) property (e.g., “size”). Some works map features to an aspect to represent an upper-level abstraction (for example, the hotel’s features “room”, “size”, “cleanness” are mapped to the aspect “room quality”) (Hu and Liu [2004a](#), Popescu and Etzioni [2005a](#)). The feature-level or aspect-level opinions essentially indicate the reviewer’s preferences over multiple facets at a fine-grained level.

2. *How to extract the needed information from reviews?* The advanced opinion mining techniques have been applied to solve this issue. In fact, in the past decade, many efforts have been devoted to analyzing user reviews at the document or feature level. Document-level mining involves the classification of a review document as either *positive* or *negative*, which can return review-level opinion. Typical methods in-
include those that use point-wise mutual information to determine the average semantic orientation (Turney, 2002), and those that apply machine learning algorithms (such as Naive Bayesian Classifier or Support Vector Machine) to learn the classification results (Pang et al., 2002). The feature-level analysis, in contrast, focuses on identifying a reviewer's opinions of multiple features (i.e., feature-level opinion), for which the popular approaches include the statistics based (like the method that captures frequently occurring words as feature candidates through association rule mining (Hu and Liu, 2004a)) and the model-based learning methods (e.g., based on lexicalized Hidden Markov Model (L-HMMs) (Jin et al., 2009) or Conditional Random Fields (CRFs) (Miao et al., 2010, Qi and Chen, 2010)). However, it may not be feasible to adopt the outcomes of these opinion mining algorithms directly for serving recommender system. Some works have therefore adjusted the algorithm to address their specific problems.

3. How to incorporate the extracted review information into the user modeling and recommendation process? This is most critical part of review-based recommender systems. Recommender systems should be aware of the exact advantage that one or more review elements can bring when making use of them. For instance, at the document level, the review-level opinion might be helpful to induce a virtual rating and hence compensate for the lacking of explicit ratings provided by users in collaborative filtering systems (Zhang et al., 2013a). At the feature level, the opinions might be used to measure the product's quality and augment products' ranking (Aciar et al., 2007). They can also be integrated into the matrix factorization (MF) framework to model the relations between users and products regarding certain features (Jakob et al., 2009). Furthermore, the comparative opinions might be utilized to strengthen
the product-to-product relationship (Zhang et al., 2010), and the contextual opinions could be useful to establish the hierarchical relations between contextual uses and products/features (Carter et al., 2011, Levi et al., 2012).

4. **What kind of product domain the system is targeted at?** Typically, there are two types of products: low-value products such as music, movies, and books for which users have much experiences and familiarity, and high-value (also called high-risk) products like laptops, TVs, digital cameras (as popularly sold on e-commerce sites) with which users are less familiar and experienced. The recommender systems have been widely used to serve repeated users in the first type of product domain, and new users in the second one. The way of utilizing reviews should hence be accordingly adjusted and adaptive to the property of concerned product type. Particularly, since users in high-risk product domains have little experience, they are usually uncertain about their requirements at the start (Chen and Pu, 2006). In other words, for high-risk and inexperienced products, users’ preferences may heavily depend on the decision context, and are often unknown, or ill-specified up front (Payne et al., 1993). It hence raises a question of whether reviews might possess potential merit in addressing the phenomenon of “partial preferences” that typically occurs during the elicitation of new users’ preferences.

According to the level of review information that they exploited, we can classify the related methods into two branches: review-level analysis and feature-level analysis.

### 2.2.1 Review-level Analysis for Recommendation

This branch of work has been primarily based on the review’s document-level analysis results. For instance, in (Garcia Esparza et al., 2012), the keywords extracted from reviews
were used to generate documents respectively for the corresponding reviewer and the item. The user document was then treated as a query to search for items that are most relevant in terms of the text similarity between the user document and the item document (through TF/IDF based cosine similarity measure). Another similar work is from (Terzi et al., 2011) that captured the user-user similarity by taking into account their posted reviews’ text similarity (via the latent semantic analysis (LSA)). These methods were proven to provide better recommendation than the traditional methods without considering reviews. However, they mainly emphasized textual descriptions or features that appear in reviews, not explicitly considering the opinions (or called sentiments) users expressed.

On the other hand, the sentimental classification technique (also called document-level opinion mining) has been developed with the goal of returning an overall opinion value (i.e., positive, negative, or neutral) for a review document (Pang et al., 2002, Wang et al., 2013). For example, Pang et al. examined how to apply machine learning tools to address the sentiment classification problem in movie review data (Pang et al., 2002). From the recommender’s perspective, the target is then how to best employ such algorithms’ outputs to handle the recommendation issues (Leung et al., 2006, Poirier et al., 2010, Zhang et al., 2013b). As a typical example, considering the lack of real ratings in some practical scenarios, Zhang et al. converted each review to a virtual score (e.g., -1 for negative, 0 for neutral, and 1 for positive) through the sentiment classification technique, which was subsequently taken as the input to the standard collaborative filtering (CF) algorithm (Zhang et al., 2013b). The conducted experiments on the review-level opinion based approaches indicate that it is advantageous to utilize the overall opinion ratings as derived from reviews to improve the recommendation accuracy, especially when there is short of user-provided star ratings. However, these approaches did not fully exploit the fine-grained opinions.
users expressed on specific features, which may hinder them from building more accurate preference model for users.

2.2.2 Feature-level Analysis for Recommendation

In comparison to review-level opinion which only measures how much a user likes a product, feature-level opinion can further tell why the user likes (or dislikes) the product, so as to better understand her/his preferences. From the aspect of algorithmic procedure, the related approaches can be classified into two major groups: content-based and model-based. Content-based approaches typically utilize the feature-level opinions to estimate the product’s quality or learn reviewers’ preferences, based on which the product ranking is performed. Model-based approaches, on the other hand, exert to train a predictive model with the feature-level opinions and then use the well-trained model to predict the product’s score.

One representative work is from Aciar et al. (2007, 2006), in which, each product is computed with a score by summing up all of its features’ quality scores (i.e., the opinion scores expressed in reviews). To extract fine-grained feature-level opinions from camera reviews, they define an ontology with two components: product quality that refers to the user’s evaluation of product features; and opinion quality that indicates the user’s expertise in the reviewed product. Each product can then be computed with a score by summing up all of its features’ quality scores. The top-$k$ products with higher scores are then returned as recommendations to the target user. Similar to the above work, Yates et al. (2008) attempted to address the “new user” problem by utilizing reviews to characterize products. They developed ShopSmart to recommend electronic products (like digital cameras), for which the opinions mined from product reviews are considered together with techni-
Table 2.1: Summary of typical works based on feature-level review analysis

<table>
<thead>
<tr>
<th>Citation</th>
<th>Review element</th>
<th>Opinion mining method</th>
<th>Recommendation method</th>
<th>Product domain</th>
<th>Served user</th>
<th>User profile</th>
<th>Evaluation metric(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aciar et al. (2007)</td>
<td>Feature-level opinion</td>
<td>Rule-based text classifier</td>
<td>Product ranking based on review features’ quality scores</td>
<td>High-risk (cameras)</td>
<td>New user</td>
<td>Feature weight</td>
<td>NA</td>
</tr>
<tr>
<td>Yates et al. (2008)</td>
<td>Feature-level opinion</td>
<td>Association rule mining based</td>
<td>Product value prediction and ranking based on SVM</td>
<td>High-risk (cameras, TVs, LCD monitors from Amazon)</td>
<td>New user</td>
<td>Feature weight</td>
<td>Percentile</td>
</tr>
<tr>
<td>Dong et al. (2013a)</td>
<td>Feature-level opinion</td>
<td>NLP technique; part-of-speech tagger; opinion patterns</td>
<td>Better score (sentiment improvement) based product ranking</td>
<td>High-risk (GPS devices, laptops, tablets from Amazon)</td>
<td>New user</td>
<td>Query product case</td>
<td>Average Better score</td>
</tr>
<tr>
<td>Dong et al. (2013b)</td>
<td>Feature-level opinion</td>
<td>Same as above</td>
<td>Better score (sentiment improvement) plus similarity based product ranking</td>
<td>High-risk (cameras, phones, printers from Amazon)</td>
<td>New user</td>
<td>Query product case</td>
<td>Ratings benefit and query product similarity</td>
</tr>
<tr>
<td>Jakob et al. (2009)</td>
<td>Aspect-level opinion</td>
<td>Subjectivity clue lexicon and grammatical dependency based</td>
<td>Multi-Relational Matrix Factorization model</td>
<td>Low-risk (movies from IMDb)</td>
<td>Repeated user</td>
<td>Aspect-level opinion scores and overall star ratings</td>
<td>RMSE</td>
</tr>
<tr>
<td>Wang et al. (2012)</td>
<td>Aspect-level opinion</td>
<td>A semi-supervised method called Double Propagation; LDA</td>
<td>Tensor model</td>
<td>Low-risk (movies from IMDb)</td>
<td>Repeated user</td>
<td>Aspect-level opinion scores and overall star ratings</td>
<td>RMSE</td>
</tr>
</tbody>
</table>
CHAPTER 2. STATE OF ART

cal specifications and shoppers’ explicitly stated preferences. Specifically, they learned a product value model based on Support Vector Machine (SVM) regression model to predict the intrinsic value of a product for the average user. The inputs to the model include opinionated features (e.g., image quality, battery life) and technical specifications (e.g., lens, megapixels). Product price is treated as the indicator of the product value and serves as a dependent variable in the training phase. The predicted intrinsic value of a product is formally represented as $V(\hat{x})$. Then, suppose the user’s preferences are represented as a preference vector $\hat{y} = <y_1, \ldots, y_n>$ (which are elicited by asking the user questions like “how much do you care about battery life?”) and a product is represented as a feature vector $\hat{x} = <x_1, \ldots, x_n>$ ($n$ is the number of features), they use the following equation to adjust the feature’s value.

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

where $f_i(x_i, y_i)$ stands for the adjusted value of feature $i$. During recommendation, the top-$k$ products with higher positive value changes (see the formula below) are retrieved as recommendation.

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

where $V(\hat{F})$ ($\hat{F} = <f_1, \ldots, f_n>$) gives the intrinsic value of adjusted features. Hence, the change indicates how much the product is suitable for the target user. To validate the system’s effectiveness, a dataset containing three product catalogs (i.e., digital camera, flat screen TV, and LCD monitor) was collected from Amazon. The user’s preferences were simulated by analyzing her/his reviews in another separate dataset. The compared methods include a preference based ranking method that uses the dot product of the preference vector and the intrinsic value vector as the function to rank products, the product value model
without considering any preference information, and the traditional collaboration filtering method. It was shown that the proposed approach is better than these methods in terms of Percentile (i.e., the ranking position of the user’s target choice in the recommendation list).

(Dong et al., 2013a) aimed at augmenting the content-based recommender system with feature-level opinions. A product case is constructed for each product to contain sentiment scores of different features. When generating recommendations, the products with higher sentiment scores are recommended to the target user. The proposed method was evaluated based on a dataset with three product catalogs (GPS devices, laptops, and tablets) from Amazon. The experiment results show that the method achieves better performance with respect to recommendation accuracy, than the similarity-based ranking methods (based on Jaccard metric or Cosine metric). In their follow-up work, (Dong et al., 2013b), the authors attempted to further improve the method by incorporating product similarity computation. They stated that recommendations that enjoy higher relative sentiment improvement but share little similarity with the query product case might fail in satisfying the user’s needs. The experiment dataset includes more product categories such as cameras, phones, and printers from Amazon. They also proposed two new metrics to evaluate the proposed method: 1) ratings benefit, which measures the relative rating improvement attained by the recommendations in comparison with ones provided by Amazon; 2) query product similarity, which measures the average similarity between the recommendations and the query product. The results indicate that the new method outperforms the originally developed approach (as introduced in (Dong et al., 2013a)) in terms of the two metrics.

As another branch, some existing works have been engaged in performing machine learning on feature-level opinions to improve the recommendation quality and efficiency.
For instance, Jakob et al. (2009) employed the Multi-Relational Matrix Factorization (MRMF) model (Lippert et al., 2008), which is an extension to the low norm MF model to be capable of taking multiple relations among different entities into account. The authors particularly used this model to capture the underlying interactions among users, movies and users’ opinions on movie aspects. Concretely, the model unifies five types of relation matrixes: 1) the relation matrix that contains the overall ratings of users for movies; 2) the relation matrix that contains opinion values on movie aspects; 3) the relation matrix that encodes whether a movie belongs to a genre; 4) the relation matrix that encodes times of opinion the user expressed on certain aspect in her/his written reviews; 5) the relation matrix that maps the similarity between users in their roles, since the user might be decomposed into several roles (e.g., one rating role and roles of reviewing different movie aspects). Given that an entity might occur in multiple relations, the matrix related to that entity will be trained under the influence of multiple relations. In this way, the latent factors related to users and items could be more accurate in generating recommendations. The experiment was performed on a movie dataset collected from IMDb. The results show that the methods, which are based on MRMF model but different in terms of aspect identification approach, outperform the baseline MF model in respect of Root-Mean-Square Error (RMSE).

Another typical work employed tensor model to integrate the aspect opinion information with star ratings (Wang et al., 2012). To be specific, they constructed a 3-dimensional tensor where the three dimensions respectively represent users, products, and user assessments of products (that include both star ratings and aspect opinion scores). This work aims at factorizing the tensor model into three factor matrixes for users, items, and ratings.
respectively. Once the model is trained, prediction can be made by multiplying the user’s, the item’s and the rating’s latent factors. The proposed algorithm was tested under different data sparsity levels, which were simulated by randomly removing some ratings from the dataset. The experiment shows that their method can outperform the other competitors in terms of RMSE, even in sparse data conditions.

It can be seen that by means of leveraging fine-grained opinions at feature (or aspect) level, the above-described works are particularly capable of addressing the “data sparsity” issue (Wang et al., 2012). The model-based approaches provided proof that the probabilistic models can be advantageous in detecting the underlying interactions among users, products and users’ multifaceted opinions on products. To attain a full picture of these typical works, we summarize the feature-level opinion based recommender approaches in Table 2.1.

### 2.2.3 Limitations

The limitations of related review-based recommender system are:

1. They were mostly oriented to serve low-risk, experienced product domains, and less addressed the issue of “new buyers” when the users search for inexperienced products.

2. Some work relied on the sentimental classification results to derive a single rating from a review, whereas reviewers’ specific preferences for the product’s features have been rarely taken into account.

3. In some related works, the reviews mainly acted as a type of supplementary info to complement the product space, based on which the ranking of products were con-
ducted by matching products to the buyer’s preferences. However, they did not consider that the buyer’s preferences can be likely incomplete and uncertain, especially in the complex decision environment.

4. The previous works, like Aciar et al. (2007), did not experimentally test their systems’ recommendation accuracy, not to mention the accuracy when buyers are typical “new users” in the inexperienced product domains.

2.3 Summary

In this chapter, we introduce some related work. Section 2.1 reviews the technologies used in systems for recommending high-risk products. Section 2.2 surveys the existing recommender systems which take into account product reviews to improve the recommendation quality.
Chapter 3

Estimating Multi-Feature Weight Preferences from User Reviews

3.1 Introduction

In this chapter, we have exerted to derive reviewers’ multi-feature weight 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 recommendations. The question is then how the system could automatically derive the reviewer’s weight preferences for features\(^1\) that he mentioned in the review (e.g., which feature(s) is more important to her?). An example is given below to illustrate the problem.

**EXAMPLE.** Reviewer A wrote a review to camera C1, and his overall rating to this product is 5 (in the range of 1 to 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 mind a little

\(^{1}\)A product feature is an attribute/component of the product that has been commented on in reviews.
CHAPTER 3. ESTIMATING MULTI-FEATURE WEIGHT PREFERENCES FROM USER REVIEWS

*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.*”

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 (Ahmad and Doja, 2012, Lappas et al. 2012). 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 all be considered so as to potentially more accurately reveal her/his weights on those features.

3.2 Methodology

3.2.1 Problem Statement

With the existing preference elicitation techniques (see Section 2.1.1), the current new buyer $\hat{u}$’s preferences over product features can be elicited and modeled based on MAUT: $w_\hat{u} = \{w_{\hat{u}k}|1 \leq k \leq n\}$, but the fact is that the elicited preferences are likely incomplete (so $i$ can be any number in the range $[1, n]$; for example, the buyer just stated preferences on features $f_2, f_3, f_6$). Thus, in order to generate accurate recommendation to the buyer, our core idea is to identify her/his inherent preference similarity to product reviewers. The research problems that we have been engaged in solving are: 1) recovering reviewers’ multi-feature weight preferences from the review information that they provided; 2)
building the preference relevance between the current buyer and reviewers; 3) making recommendations for the current buyer by fusing like-minded reviewers’ preferences. In the following, we first give our system’s work flow, and then in detail describe the methods that we have developed.

### 3.2.2 System’s Work Flow

The work flow of our system mainly consists of three steps (see Figure 3.1).

![Figure 3.1: Work flow of system with two major branches of proposed approaches which result in five different recommending methods.](image)

1) The first step is conducting **feature-level opinion mining** to identify `<feature, opinion>` pairs extracted for every reviewer.

---

— 30 —
pairs from every review, where opinion indicates positive, neutral, or negative sentiment that a reviewer expressed on a feature. In this step, we have attempted to improve current opinion mining (or called sentiment analysis) methods [Hu and Liu 2004b, Chen et al. 2012], with the aim to fully exploit the values of review features and their associated opinions for deriving reviewers’ weight preferences.

2) Then, our primary focus is on inferring the reviewers’ weight preferences. In this step, we have developed two alternative approaches. Particularly, in addition to building the probabilistic regression model (PRM-based approaches; see Section 3.2.4), we have investigated the effect of the latent class regression model (LCRM) on enhancing the stability of single reviewer’s preferences (called reviewer-level) and producing cluster-level preferences for a group of reviewers simultaneously. Principally, four review elements are integrated into this model: the reviewer’s overall rating on a product; the opinion associated with each feature in the review; the feature’s occurring frequency (as a type of prior knowledge to be modeled); and the product that the reviewer commented.

3) Resulting from the Step 2), there are three types of outcomes: PRM-based reviewer-level preferences, LCRM* based reviewer-level preferences, and LCRM* based cluster-level preferences (see Figure 3.1). We have accordingly implemented different recommendation methods. With the reviewer-level preferences, we have tried 1) k-nearest neighbor algorithm (k-NN) for locating a group of reviewers who have similar weight preferences to the current buyer (shorted as PRM-k-NN and LCRM*-r-k-NN); 2) k-Means algorithm for clustering reviewers based on their respective preferences and then identifying a cluster which is most relevant to the current buyer (shorted as PRM-k-Means and LCRM*-r-k-Means). With the cluster-level preferences as resulted from LCRM*, we have directly used it to perform the clustering. Thus, there are in total five different recommendation
CHAPTER 3. ESTIMATING MULTI-FEATURE WEIGHT PREFERENCES
FROM USER REVIEWS

methods (see Figure 3.1). Every recommendation method can return the top-\(N\) products, with which our evaluation task is to test whether the current buyer’s target choice (i.e., the product that s/he is prepared to buy) can be located in the list of \(N\) recommended products or not.

3.2.3 Pre-Process: Extracting Feature-Opinion Pairs from Product Reviews

Before deriving reviewers’ weight preferences, we need to first analyze their raw textual reviews and convert them into structured \(<feature, opinion_value>\) pairs. Previously, we have compared different learning methods for mining the feature-level opinions from reviews, and identified the respective advantages of model-based and statistical approaches (Chen et al., 2012). Thus, inspired by prior findings, in the current system, we have concretely carried out three steps for identifying the feature-opinion pairs, which include:

1. Extracting features from a review and grouping synonymous features;

2. Locating opinions that are associated with various features in the review;

3. Quantifying the opinion value in the normalized range \([1, 5]\).

Specifically, to identify the prospective feature candidates, we used a Part-of-Speech (POS) tagger included in the Core-NLP package\(^2\) to extract the frequent nouns (and noun phrases). Moreover, considering that reviewers often use different words for the same product feature (e.g., “picture” and “appearance” for “image”), we manually defined a set of seed words and then grouped the synonymous features by computing their lexical similarity to the seed words. The lexical similarity was concretely determined via WordNet (Fellbaum).

\(^2\)http://nlp.stanford.edu/software/corenlp.shtml
CHAPTER 3. ESTIMATING MULTI-FEATURE WEIGHT PREFERENCES FROM USER REVIEWS

Given that unsupervised methods normally take the risk of returning inaccurate results while supervised methods often require demanding human-labeling efforts, our approach was targeted to achieve the ideal balance between accuracy and effort. The previous experiment showed that such procedure can help identify reliable feature expressions in review text and effectively group them (Chen et al., 2012).

We then located all opinions which are associated with 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 (Hu and Liu, 2004b; Popescu and Etzioni, 2005b). However, these methods cannot identify opinions that are not so close to a feature. Therefore, we took advantage of a syntactic dependency parser3 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” (suggesting that “photos” is the subjective of “great”). Thus, any words that are with such dependency relations with a feature are taken as opinion words.

Note that there might be some noisy information contained in the review text such as grammatical and syntactical errors, that is why we chose the publicly recognized POS tagger. Moreover, the syntactic dependency parser we employed is based on probabilistic model, that means it can be inherently able to handle the noisy inputs because it produces the most likely analysis to a sentence after checking all possibilities. Actually, the exper-

3http://nlp.stanford.edu/software/lex-parser.shtml
imement proved that it can well recover the structure of noisy sentences (Kakkonen, 2008). Besides, we conducted a cleaning process to correct the misspelled words in review text via a statistical spell checker[^1] and remove the duplicate, unnecessary punctuation marks in sentences (e.g., “!!!” that appears at the end of some sentences).

Near the end of this step, we need to assess every opinion word’s sentiment strength (also called polarity value). For this task, we applied SentiWordNet (Esuli and Sebastiani, 2006) because it can provide with a triple of polarity scores for each opinion word: positivity, negativity and objectivity, respectively denoted as \( \text{Pos}(w) \), \( \text{Neg}(w) \), and \( \text{Obj}(w) \), for the word \( w \). Each ranges from 0.0 to 1.0, and \( \text{Pos}(w) + \text{Neg}(w) + \text{Obj}(w) = 1 \). The triple scores can then be merged into a single sentiment value: \( O_w = \text{Neg}(w) \times R_{\text{min}} + \text{Pos}(w) \times R_{\text{max}} + \text{Obj}(w) \times \frac{R_{\text{min}} + R_{\text{max}}}{2} \) (where \( R_{\text{min}} \) and \( R_{\text{max}} \) represent the minimal and maximal scales respectively. We set them as \( R_{\text{min}} = 1 \) and \( R_{\text{max}} = 5 \), so that \( O_w \) ranges from 1 to 5). In addition, we considered negation words (such as “not”, “don’t”, “no”, “didn’t”): if the odd number of such words appears in a sentence, the polarity of related opinion will be reversed. Then, in the case that there are multiple opinion words associated with a feature in a review, we calculated a weighted average for which the opinion word’s sentiment value behaves as the weight, so that the extremely positive or negative polarization is less susceptible to shift. For instance, if two opinion words “good” and “great” are both associated with a feature, the feature’s final opinion value is: \( \frac{4 \times 4 + 5 \times 5}{4 + 5} = 4.55 \) where 4 and 5 are the sentiment values of the two words “good” and “great” respectively.

[^1]: http://norvig.com/spell-correct.html
3.2.4 Approach 1: Probabilistic Regression Model (PRM) based Recommendation

After extracting the pairs \(<\text{feature}, \text{opinion value}\>\) from every review, our focus is then on deriving corresponding reviewer’s weighted feature preferences. The first approach we have developed is based on Probabilistic Regression Model (PRM) (Yu et al., 2011) to learn the weights for individual reviewers, i.e., to generate reviewer-level preferences.

Specifically, we treat the relationship between the overall rating that a reviewer assigned to a product and her/his opinion values being associated with the product’s features as a regression problem. More formally, a reviewer’s \((u_i)\) overall rating \(y_{ij}\) on a product \(p_j\) can be considered as a dependent variable, being the function of a set of independent opinion values \(X_{ij}\) in respect of the set of features \(\mathcal{F}\). The regression coefficients can then be interpreted as the weight preferences of the reviewer \(w_i\) because they essentially define the relative contributions of various features to determine the overall rating:

\[
y_{ij} = \hat{y}_{ij} + \varepsilon = w_i^T X_{ij} + \varepsilon
\]  
(3.1)

where \(\varepsilon\) is a noise term.

Since the overall rating \(y_{ij}\) and opinion values \(X_{ij}\) are known, we can derive the weight preferences \(w_i\) via Bayesian treatment because it can incorporate additional information, such as prior knowledge, to improve the model. Concretely, the noise term \(\varepsilon\) is drawn from a Gaussian distribution with zero as the mean:

\[
\varepsilon \sim \mathcal{N}(0, \sigma^2)
\]  
(3.2)

in which \(\sigma^2\) is the variance parameter that controls the model’s precision. The conditional
probability that a reviewer \( u_i \) gives the overall rating \( y_{ij} \) to a product \( p_j \) can hence be defined as follows:

\[
\text{Pr}(y_{ij}|X_{ij}, w_i) = \mathcal{N}(y_{ij} | w_i^T X_{ij}, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(y_{ij} - w_i^T X_{ij})^2}{2\sigma^2}\right)
\]  

(3.3)

According to the Bayes theory, the posterior probability of \( w_i \) can be defined as the product of Equation 3.3 and the prior probability:

\[
\text{Pr}(w_i|\mathcal{R}) \propto \prod_{(u_i, p_j) \in \mathcal{R}} \text{Pr}(y_{ij}|X_{ij}, w_i) \times \text{Pr}(w_i|\mu, \Sigma) \times \text{Pr}(\mu, \Sigma)
\]  

(3.4)

where \( \mathcal{R} \) denotes the set of reviewer-product pairs (see Table 1.1). \( \text{Pr}(w_i|\mu, \Sigma) \) is the prior probability of \( w_i \) which can be drawn from a multivariate Gaussian distribution with \( \mu \) as the mean and \( \Sigma \) as the covariance matrix:

\[
\text{Pr}(w_i|\mu, \Sigma) \sim \mathcal{N}(\mu, \Sigma)
\]  

(3.5)

We further incorporate the feature’s occurrence frequency, as the prior knowledge \( (\mu_0) \), into \( \mathcal{N}(\mu, \Sigma) \). The prior probability of the distribution \( \mathcal{N}(\mu, \Sigma) \) is consequently formulated as:

\[
\text{Pr}(\mu, \Sigma) = \exp\left(-\psi \cdot KL(\mathcal{N}(\mu, \Sigma)|\mathcal{N}(\mu_0, I))\right)
\]  

(3.6)

where \( KL(\cdot | \cdot) \) is the KL-divergence for calculating the difference between two distributions \( \mathcal{N}(\mu, \Sigma) \) and \( \mathcal{N}(\mu_0, I) \), and \( \psi \) is the trade-off parameter (which is default set as 50 in the experiment) to control the strength of \( \mu_0 \) in the model.

We then apply the expectation-maximization (EM) algorithm \( [\text{Dempster et al., 1977}] \) to identify the optimal values for the set of parameters \( \Psi = \{w_1, \ldots, w_U, \mu, \Sigma, \sigma^2\} \) which contain the reviewer’s weight preferences \( w_i \):

\[
\Psi^* = \arg\max_{\Psi} \mathcal{L}(\Psi|\mathcal{R}) = \sum_{(u_i, p_j) \in \mathcal{R}} \log(\text{Pr}(w_i|\mathcal{R}))
\]  

(3.7)
CHAPTER 3. ESTIMATING MULTI-FEATURE WEIGHT PREFERENCES FROM USER REVIEWS

PRM-based Recommendation via k-NN (PRM-k-NN)

Given individual reviewers’ weight preferences, the intuitive way to generate recommendation is based on the k-nearest neighbor (k-NN) method \(^{(Shakhnarovich et al., 2006)}\). That is, we can first identify a group of \(k\) reviewers who have similar feature preferences to the current buyer, and then locate recommendations from products that were highly rated by those similar reviewers. Formally, suppose the elicited preferences from the current buyer are \(w_{\hat{u}}\), her/his similarity to a reviewer \(u_i\) can be computed via:

\[
sim(w_{\hat{u}}, w_i) = \frac{1}{1 + \sqrt{\sum_{w_{f_l} \in w_{\hat{u}}} (w_{f_l}(\hat{u}) - w_{f_l}(u_i))^2}}
\]

(3.8)

where \(w_{f_l}(\hat{u})\) is the current buyer’s weight preference on feature \(f_l\), and \(w_{f_l}(u_i)\) is the \(i\)-th reviewer’s. We then retrieve \(k\) reviewers who are with higher similarity scores to the current buyer (\(k\) is optimally set through the experiment; see Section 3.3), and get a pool of products that were rated by these reviewers. Each product \(p_j\) from the pool is finally calculated with a prediction score to indicate how much it would interest the current buyer \(\hat{u}\):

\[
ProductScore(\hat{u}, p_j) = \frac{\sum_{u_i \in K} \sim(w_{\hat{u}}, w_i) \times y_{ij}}{\sum_{u_i \in K} \sim(w_{\hat{u}}, w_i)}
\]

(3.9)

where \(K\) denotes the group of \(k\) similar reviewers, and \(y_{ij}\) is the rating that reviewer \(u_i\) gave to the product (it is zero if the reviewer did not review it). The top-\(N\) products with higher prediction scores are returned to the buyer as the recommendation. In our experiment, we tested the algorithm’s accuracy when \(N = 10\) or 20.
As an alternative recommending solution, we turn to conduct the clustering process at first for identifying clusters, each of which is composed of multiple reviewers who possess similar preferences among each other. The current buyer is then matched to the most relevant cluster, within which the k-nearest neighbor algorithm is further performed. Actually, the clustering process has been recognized as an effective way to increase the recommender system’s efficiency (Kim and Yang, 2005, Sarwar et al., 2002). For instance, the clustering based collaborative filtering (CF) system has been found obtaining comparable prediction accuracy to the basic CF approach, while achieving significantly higher efficiency (Sarwar et al., 2002). We were thus motivated to perform the clustering within reviewers before locating similar ones to the current buyer. Specifically, through the classical clustering technique such as k-Means, during each round, a reviewer will be moved from one cluster to another if this process can minimize its distance to the cluster’s centroid. The centroid is formally denoted as $w_{c_k, \text{centroid}}$, which is calculated by averaging the preferences of the cluster’s currently contained reviewers. The distance is calculated via $1/sim(w_i, w_{c_k, \text{centroid}})$ where $sim(w_i, w_{c_k, \text{centroid}})$ is defined similar to Equation 3.8.

Upon the clustering process ends, we get $K$ disjoint clusters $\{c_1, \ldots, c_K\}$. The current buyer’s preferences are then used to compute her/his distance to all clusters’ centroids, and the cluster with the shortest distance is regarded most relevant to the buyer. Within this cluster, we conduct PRM-k-NN to retrieve $k$ nearest neighbors and then calculate each candidate product’s prediction score via Equation 3.9. Still, top-$N$ products with higher scores are recommended to the buyer.
CHAPTER 3. ESTIMATING MULTI-FEATURE WEIGHT PREFERENCES
FROM USER REVIEWS

3.2.5 Approach 2: Latent Class Regression Model (LCRM) based
Recommendation

Background of LCRM: given the above-mentioned limitation of PRM-based methods, as
the follow-up work, we have aimed at more accurately estimating a reviewer’s weight pref-
erences by taking into account additional information, such as her/his inherent preference
similarity to other reviewers. For this purpose, we have studied Latent Class Regression
Model (LCRM) (Wedel and Kamakura, 2000). Historically, LCRM originated from the
area of marketing for conducting the market segmentation task, which is targeted to di-
vide the set of prospective consumers into relatively smaller groups for revealing their
preference homogeneity. In the field of machine learning, LCRM has been regarded as
a specific branch of mixture models (McLachlan and Peel, 2000), to mainly handle the
data with regression characteristics. Specifically, LCRM assumes that the whole popu-
lation can be defined by a finite number of distributions (every distribution represents a
cluster of consumers in the case of market segmentation), so its primary goal is to esti-
mate each distribution’s regression model at the cluster level. Therefore, LCRM does not
require the knowledge of single entity’s regression values (e.g., from a single consumer),
but focuses on exploiting the whole population’s structure to generate the clusters directly.
Every entity is assigned to a cluster only when this assignment has the highest membership
probability.

Due to these properties, we believe it might be useful to address the limitation of PRM-
based methods. Because the standard LCRM can be used to only produce the cluster-
level preferences (i.e., the cluster-level regression model that represents a group of entities
which are “reviewers” in our case), we have extended it to LCRM*, in order to derive every
reviewer’s preferences simultaneously. The concrete idea of LCRM* is that the cluster-
level preferences might be leveraged to polish individual reviewers’ weight preferences, so that not only the reviewer’s own information is taken into account in this regard, but her/his inherent similarity to other reviewers can be incorporated together so as to stabilize her/his preferences. The advantage of LCRM* against LCRM is thus that it can potentially uncover the preference heterogeneity among reviewers within a cluster.

**LCRM*: Deriving Both Cluster-Level and Reviewer-Level Feature Preferences**

In the following, we in detail describe how the LCRM is extended to achieve the above-mentioned objectives. Concretely, four types of review elements are unified in this model: the reviewer’s overall rating on a product; the opinion associated with each feature in the review; the feature’s occurring frequency as a type of prior knowledge; and the product that the reviewer commented.

According to the basic LCRM model, we first assume that all reviewers can be divided into \( K \) clusters \( \mathcal{C} = \{ c_1, c_2, \ldots, c_K \} \). The likelihood probability function for the overall rating \( R_{ij} \) (the dependent variable) is defined as:

\[
Pro(y_{ij}|X_{ij}, \Phi) = \sum_{k=1}^{K} \pi_k Pro(y_{ij}|X_{ij}, c_k)
\]  

where \( \Phi \) denotes the set of all parameters, \( \pi_k \) denotes the prior probability of cluster \( c_k \), and \( X_{ij} \) is the vector of opinion values associated with features \( \mathcal{F} \) w.r.t. reviewer \( u_i \). In the above formula, the right component \( Pro(y_{ij}|X_{ij}, c_k) \) gives the conditional probability of the overall rating \( y_{ij} \) in the case that reviewer \( u_i \) belongs to the cluster \( c_k \):

\[
Pro(y_{ij}|X_{ij}, c_k) = Pro(y_{ij}|X_{ij}, w_i) \cdot Pro(w_i|c_k)
\]  

in which \( w_i \in \mathbb{R}^n \) denotes the reviewer \( u_i \)’s weight preferences, and \( Pro(y_{ij}|X_{ij}, w_i) \) gives the likelihood of \( w_i \) given the overall rating \( y_{ij} \) and features’ opinion vector \( X_{ij} \). Here,
we can assume that each reviewer’s preference is drawn from the cluster-level preference
distribution, which can be a Multivariate Gaussian with \( w_{ck} \) (the cluster-level preferences) as the mean and \( \Sigma_k \) as the covariance matrix:

\[
Pro(w_i|c_k) = Pro(w_i|w_{ck}, \Sigma_k) \sim N(w_i|w_{ck}, \Sigma_k)
\] (3.12)

Furthermore, the uncertainty of the distribution of the cluster-level preferences \( N(w_i|w_{ck}, \Sigma_k) \) can be modeled based on the KL-divergence as follows:

\[
Pro(w_{ck}, \Sigma_k) = \exp\left(-\psi \cdot KL(N(w_{ck}, \Sigma_k)|N(\mu_0, I))\right)
\] (3.13)

where \( \mu_0 \) is the set of occurrence frequencies of features in the reviews.

Because the overall rating \( y_{ij} \) is known, the probability that a reviewer belongs to a cluster can hence be estimated. Formally, a reviewer \( u_i \) is placed into a cluster \( c_k \) if \( q_k(u_i) > q_h(u_i) \) \( \forall c_k \neq c_h \), where

\[
q_k(u_i) = \prod_{(u_i, p_j) \in \mathcal{R}} \frac{\pi_{jk} \cdot Pro(y_{ij}|X_{ij}, c_k)}{\sum_{c_h \in \mathcal{C}} \pi_{jh} \cdot Pro(y_{ij}|X_{ij}, c_h)}
\] (3.14)

In addition, it is reasonable to assume that reviewers who commented the same product should be more preference relevant. Thus, with her/his commented product \( p_j \), the distribution \( \pi_j = \{ \pi_{j1}, \ldots, \pi_{jK} \} \) can be modeled as the prior probability that the reviewer \( u_i \) belongs to a certain cluster. The full mixture log-likelihood with all observations \( \mathcal{R} \) (i.e., the collection of reviewer-product pairs) can hence be defined as:

\[
\mathcal{L}(\Phi|\mathcal{R}) = \sum_{(u_i, p_j) \in \mathcal{R}} \log\left(\sum_{k=1}^{K} \pi_k Pro(y_{ij}|X_{ij}, c_k)\right)
\] (3.15)

We further derive the following two formulas (Equations 3.16 and 3.18), respectively for inferring cluster-level preferences and reviewer-level preferences:

\[
\hat{w}_{ck} = (N_k \Sigma_k^{-1} + \psi \cdot I)^{-1} \left( \Sigma_k^{-1} \sum_{z_i=k}^{U} w_i + \psi \cdot I \cdot \mu_0 \right)
\] (3.16)
where

\[
\hat{\Sigma}_k = \left[ \frac{1}{\psi} \sum_{z_i=k}^U (\mathbf{w}_i - \mathbf{w}_{c_k})(\mathbf{w}_i - \mathbf{w}_{c_k})^T + \left( \frac{N_k - \psi}{2\psi} \right)^2 I \right]^{-\frac{1}{2}} - \frac{N_k - \psi}{2\psi} I \right]^T \tag{3.17}
\]

\[
\hat{w}_i = \frac{1}{N(u_i)} \sum_{(u_i, p_j) \in R} \left( \frac{X_{ij}X_{ij}^T}{\sigma^2} + \Sigma_k^{-1} \right) - 1 \left( \frac{(y_{ij} - \mathbf{w}_i^T X_{ij})}{\sigma^2} + \Sigma_k^{-1} \mathbf{w}_{c_k} \right) \tag{3.18}
\]

in which \( N(u_i) \) is the number of reviews posted by reviewer \( u_i \).

The parameters’ set \( \Phi = \{ z_1, \ldots, z_U, \mathbf{w}_{c_1}, \ldots, \mathbf{w}_{c_K}, \Sigma_1, \ldots, \Sigma_K, \mathbf{w}_1, \ldots, \mathbf{w}_U \} \) is then estimated through the expectation-maximization (EM) algorithm, which seeks to identify the maximal log-likelihood by iteratively performing the following two steps.

- **Expectation step (E-step):**

  In this step, with individual reviewer’s preferences \( \mathbf{w}_i \), we aim to update the reviewer’s cluster assignment, the cluster-level preference distribution, and the prior probability of clusters.

  1. The cluster assignment \( z_i (z_i = k \text{ if reviewer } u_i \text{ belongs to cluster } c_k) \) is determined via:

     \[
     z_i = \arg\max_k q_k(u_i) \tag{3.19}
     \]

     where \( q_k(u_i) \) is referred to Equation\(^3.14\). One reviewer is assigned to a cluster only when the highest probability is obtained.

  2. For each cluster, the cluster-level preferences \( \mathbf{w}_{c_k} \) are updated using the Equation\(^3.16\).
3. The prior probability of clusters (i.e., $\pi_j = \{\pi_{j1}, \ldots, \pi_{jK}\}$) can be treated as multinomial distribution and updated through the Laplace smoothing:

$$
\pi_{jk} = \frac{\sum_{(u, p_j) \in R} I_{z_i = k} + \lambda}{N(p_j) + K \times \lambda} \quad (3.20)
$$

in which $N(p_j)$ is the number of reviews posted to product $p_j$, and the scale variable $\lambda \in [0, 1]$ is the smoothing parameter.

- **Maximization step (M-step):** in this step, we aim to update each reviewer’s preferences $w_i$ through Equation 3.18 (with other parameters fixed).

E- and M-steps are repeated until the Equation 3.15 converges. As the result, all reviewers are divided into $K$ disjoint clusters, plus the cluster-level preferences $w_{ck}$ generated for each cluster and the reviewer-level preferences $w_i$ for every reviewer. It is worth mentioning that Equations 13 to 20 are our proposed extension to the basic LCRM. The major algorithm steps of LCRM* are shown in Algorithm 1.

As for the algorithm’s time complexity, the E-step costs $O(max(|R|, n) \times K \times n^2)$ operations, and the M-step costs $O(K \times n^3 + |R|n^2)$ operations, where $K$ is the number of clusters and $n$ is the number of product features. Suppose LCRM* converges after $t$ iterations, the computational complexity of LCRM* is $O(t \times max(|R|, n) \times K \times n^2)$. In comparison, because the probabilistic regression model (PRM) (see Section 3.2.4) requires to compute the determinant of the covariance matrix that takes $O(n^3)$ operations, its complexity is $O(t \times U \times n^3)$ in which $t$ is still the number of iterations of EM steps and $U$ is the number of reviewers.
Algorithm 1 The pseudo-code of LCRM* algorithm

1: **Data:** The observations $R$; each review is with the overall rating $y_{ij}$ and the extracted feature-opinion vector $X_{ij}$
2: **Result:** $K$ disjoint clusters, along with the cluster-level preferences $w_{ck}$ for each cluster, and the reviewer-level preferences $w_i$ for each reviewer in the cluster.
3: initialize reviewers’ cluster assignment randomly
4: initialize cluster-level preferences $w_{ck} \sim \mathcal{N}(0, I)$
5: initialize reviewer-level preferences $w_i \sim \mathcal{N}(0, I)$
6: repeat
7: 1. **Expectation step (E-step)**
8: for each reviewer $u_i$ do
9: assign each reviewer to a cluster through Equation 3.14
10: end for
11: update the cluster-level preferences $w_{ck}$ using Equation 3.16
12: update the covariance matrix $\Sigma_k$ using Equation 3.17
13: update the prior probability of clusters using Equation 3.20
14: 2. **Maximization step (M-step)**
15: for each reviewer $u_i$ do
16: update each reviewer’s preferences $w_i$ using Equation 3.18
17: end for
18: until Equation 3.15 converges

**LCRM*-based Recommendation Generation**

To generate recommendation for the current buyer, we first classify him/her to the most relevant cluster of reviewers. The preference similarity between the buyer $\hat{u}$ and a cluster $c_k$ is computed via:

$$Sim(w_{\hat{u}}, w_{ck}) = \frac{1}{1 + \sqrt{\sum_{f=1}^{n_f} (w_f(\hat{u}) - w_f(c_k))^2}}$$

(3.21)

where $w_{\hat{u}}$ is the buyer $\hat{u}$’s stated weight preferences and $w_{ck}$ denotes the cluster-level preferences of cluster $c_k$. The cluster with the highest similarity value is then chosen. Within it, we further retrieve $k$ reviewers who are most similar to the current buyer based on their
respective reviewer-level preferences. The formula for calculating the similarity between a 
reviewer and the current buyer can be referred to Equation 3.8. Then, a pool that contains 
products rated by these \( k \) reviewers is generated, and a prediction score is computed for 
each product \( p_j \) to indicate its matching degree with the buyer’s potential interests:

\[
ProductScore(\hat{u}, p_j) = \frac{\sum_{u_i \in c_l \cap \mathcal{K} \cap (u_i, p_j) \in \mathcal{R}} \text{sim}(\hat{w}_u, w_i) \times y_{ij}}{\sum_{u_i \in c_l \cap \mathcal{K} \cap (u_i, p_j) \in \mathcal{R}} \text{sim}(\hat{w}_u, w_i)}
\]  

(3.22)

where \( c_l \) denotes the most relevant cluster, \( \mathcal{K} \) is the set of \( k \) nearest reviewers, \( y_{ij} \) is the 
overall rating that a reviewer gave to the product, and \( \text{sim}(w_u, w_i) \) is the preference similarity between the buyer \( \hat{u} \) and the reviewer \( u_i \). Top-\( N \) products with higher prediction 
scores are recommended to the buyer.

The time complexity of this step is \( O(|c_l|^2 \times |\mathcal{P}| \times |\mathcal{K}| + |\mathcal{P}|^2) \), where \( |\mathcal{P}| \) denotes the 
number of all products.

### 3.2.6 Discussion

The main differences between the PRM-based approaches (i.e., PRM-k-NN and PRM-k- 
Means) and LCRM* can be found in Table 3.1. In contrast with PRM-based approaches, 
individual reviewers’ preferences are derived inter-dependently in LCRM*. That is, both 
cluster membership and reviewers’ commonly reviewed products are considered when 
adjusting each reviewer’s preferences. Indeed, this algorithm attempts to explain every 
reviewer’s behavior by involving her/his inherent similarity to some of other reviewers.

On the other hand, in comparison to the heuristic clustering methods such as k-Means, 
LCRM*, as a type of model-based clustering approach, does not require the pre-knowledge 
of reviewer-level preferences and the pre-definition of entity-entity distance metric. Therefore, its efficiency and accuracy might be higher, especially in the condition that the in-
CHAPTER 3. ESTIMATING MULTI-FEATURE WEIGHT PREFERENCES FROM USER REVIEWS

formation provided by a single reviewer is very limited. As for the recommending procedure, LCRM* and PRM-k-Means both include the process of clustering reviewers, which might potentially help increase the algorithm’s performance and prediction power, whereas PRM-k-NN might be limited as it purely relies on the buyer’s stated preferences to locate similar reviewers.

In addition, in order to identify the specific effect of LCRM*-based method on generating reviewer-level preferences, we have developed two variants of LCRM*: k-NN recommendation based on LCRM* (shorted as LCRM* k-NN) and k-Means recommendation based on LCRM* (shorted as LCRM* k-Means). In these two methods, the cluster-level preferences as produced by LCRM* are not utilized. We compared them respectively to their counterparts, i.e., LCRM* k-NN vs. PRM-k-NN, and LCRM* k-Means vs. PRM-k-Means, because the two in each pair just differ in terms of inducing individual reviewers’ preferences (with other steps identical). Moreover, we compared LCRM* k-Means with LCRM* given their exclusive difference regarding the clustering process, so as to identify which clustering approach is more effective. Besides, it is worth mentioning that the original LCRM is not included in the experimental comparison, mainly because it can only produce cluster-level preferences, which however are not useful to carry out the user-user similarity measure and hence limit the recommendation process.

3.3 Experiment

3.3.1 Experimental Setup and Dataset

We prepared two real-world datasets for conducting the experiment: digital camera dataset and laptop dataset, which were crawled from a commercial website www.buzzillions.com. In both datasets, each textual review is accompanied by an overall rating that ranges
from 1 to 5 stars as assigned by the corresponding reviewer. Before the experiment, we cleaned the datasets: 1) removing the reviews which contain less than 4 features (including the ones that are too short or with meaningless characters), and 2) removing the products that contain less than 10 reviews. The cleaning process ensured that each review contains a certain amount of information, and each product consists of sufficient reviews to be analyzed. After this step, the digital camera dataset has 112 cameras along with totally 18,251 reviews, and the laptop dataset has 155 laptops for which there are 6,024 reviews in total. The analysis on both datasets shows that every reviewer posted only one review to a product, which is consistent with the statistics reported in (Jindal and Liu, 2008, Xie et al., 2012). The details of these two datasets are in Table 3.2.

Following the leave-one-out evaluation mechanism (Reilly et al., 2004), we are able to simulate a set of “buyers” from the dataset. That is, a reviewer can be considered as a simulated “buyer”, if s/he satisfies two criteria: 1) s/he only commented one product, and 2) her/his overall rating on the reviewed product is 5 (full star), indicating that s/he

---

### Table 3.1: Summary of developed methods’ properties

<table>
<thead>
<tr>
<th>Deriving reviewer-level preferences</th>
<th>PRM-k-NN</th>
<th>PRM-k-Means</th>
<th>LCRM*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leveraging cluster-level preferences</td>
<td>X</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Considering reviewers’ commonly reviewed products</td>
<td>X</td>
<td>X</td>
<td>✓</td>
</tr>
</tbody>
</table>

---

---

---
strongly likes this product. Therefore, this highly rated product can be taken as the simulated buyer’s target choice, and the reviewer’s feature preferences can be taken as the simulated buyer’s full preferences. It turns out that 4002 reviewers satisfy these criteria in the camera dataset and 1330 reviewers in the laptop dataset. Therefore, at a time, one of these reviewers was excluded from the dataset to be a “buyer”. The aim of the experiment was hence to measure whether the buyer’s target choice can be located in the list of recommended items as returned by the tested algorithm. Moreover, in addition to performing the testing on the buyer’s full preferences (i.e., over 10 features in the camera dataset and over 8 features in the laptop dataset), we randomly took out subsets of his/her preferences to represent the partial preferences that s/he may state in the real situation (e.g., preferences over 4, 6, or 8 features, out of full 10 features, in the camera dataset; and preferences over 4 or 6 features, out of full 8 features, in the laptop dataset). Thus, in total, there are $4002 \times 4 = 16008$ testings performed in the camera dataset, and $1330 \times 3 = 3990$ testings performed in the laptop dataset. The number of features involved in each input preferences

---

**Table 3.2: Description of two datasets (digital camera and laptop) used in Chapter 3**

<table>
<thead>
<tr>
<th></th>
<th>Digital camera</th>
<th>Laptop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total reviewers</td>
<td>18,251</td>
<td>6,024</td>
</tr>
<tr>
<td>Total products</td>
<td>122</td>
<td>155</td>
</tr>
<tr>
<td>Min. reviews per product</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>Average reviews per product</td>
<td>149.6 (st.d. = 171.835)</td>
<td>33 (st.d. = 34.475)</td>
</tr>
<tr>
<td>Max. reviews per product</td>
<td>1,052</td>
<td>222</td>
</tr>
<tr>
<td>Min. features per review</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Average features per review</td>
<td>5.16 (st.d. = 1.266)</td>
<td>5.03 (st.d. = 1.119)</td>
</tr>
<tr>
<td>Max. features per review</td>
<td>10</td>
<td>8</td>
</tr>
</tbody>
</table>
is called the corresponding buyer’s preference size.

### 3.3.2 Evaluation Metrics

To choose appropriate evaluation metrics, we have considered Kendall’s tau, Hit-Ratio and MRR (Mean Reciprocal Rank), because they have all been applied to measure the algorithm’s accuracy in preference-based recommender systems (Deshpande and Karypis, 2004; Gunawardana and Shani, 2009). However, Kendall’s tau turns out being unsuitable for our case because it assumes that each user has multiple target choices, and aims to compute the similarity between the estimated preference ranking over these choices and the true preference ranking, while in our condition each user only has one target choice. Thus, we have finally decided to use Hit-Ratio and MRR as the metrics in our experiment.

- **$H@N$ (Hit Ratio @ top-$N$ recommendations)** mainly measures whether the user’s target choice appears in the set of $N$ recommendations or not (in the experiment, $N$ is set as 10 or 20). It concretely returns the percent of hits among all users:

$$H@N = \frac{\text{#The number of successes within the top-N}}{\text{#The total number of test cases}}$$

(3.23)

- **$MRR$ (Mean Reciprocal Rank)** is a statistic measure for evaluating the ranking position of the target choice in the whole list:

$$MRR = \frac{\sum_{t=1}^{T} (0 + 1_{\text{rank}_t \leq 20})}{T} \frac{1}{\text{rank}_t}$$

(3.24)

in which $T$ is the number of test cases, and $\text{rank}_t$ is the ranking position of the target choice when testing the $t$-th case. $1_{\text{rank}_t}$ is an indicator function that equals to 1 if $\text{rank}_t \leq 20$ (i.e., if the target choice appears in the top 20 products), and 0 otherwise.
3.3.3 Compared Baseline Methods

In addition to the methods described in Sections 3.2.4 & 3.2.5, we also implemented two baselines: one does not consider product reviews (shorted as Non-Review); and another primarily utilizes reviews to perform product ranking, but does not attempt to derive the reviewers’ feature preferences (shorted as Review-Rank) (Aciar et al., 2007).

1. Non-Review based Recommendation (Non-Review)

This is a baseline that does not incorporate reviews’ feature-level opinions into computing recommendation. It is simply based on the product’s static feature values to determine how much it matches to the current buyer’s stated preferences $\hat{w}_u$:

$$\text{ProductScore}(\hat{u}, p_j) = \sum_{w_{f_l}(\hat{u}) \in \hat{w}_u} w_{f_l}(\hat{u}) \times s_{f_l}(p_j)$$  \hspace{1cm} (3.25)

where $p_j$ is the product, $s_{f_l}(p_j)$ is the utility of every feature $f_l$ (which is normalized in the range from 0.0 to 1.0), and $w_{f_l}(\hat{u})$ denotes the current buyer’s weight preference on feature $f_l$. More specifically, the feature’s utility is computed by assessing the feature’s fixed specification (e.g., for the camera’s logical zoom, it is “the higher, the better”, and for price, it is “the cheaper, the better”). The utility is also called value function in (Keeney and Raiffa, 1976). A default utility function was defined for each feature based on the domain knowledge.

As shown in Equation (3.25), the utility of a feature is multiplied with the buyer’s weight preference, and the weighted additive sum score that involve all features is then computed to indicate the product’s satisfying degree. Top-$N$ products with higher scores are displayed to the buyer in the list of recommendation.
CHAPTER 3. ESTIMATING MULTI-FEATURE WEIGHT PREFERENCES FROM USER REVIEWS

2. Review-based Product Ranking (Review-Rank)

This approach extracted features and opinions from reviews, while its major difference from our methods is that it did not target to derive reviewers’ multi-feature preferences from these extracted data. Actually, it is just based the feature’s opinion value to calculate its utility:

\[
FeatureScore_{f_l}(p_j) = \frac{\sum_{(u_i,p_j) \in S} x_{ijl}}{m}
\]

(3.26)

where \(x_{ijl}\) denotes the feature \(f_l\)'s opinion value in review \(r_{ij}\) w.r.t. product \(p_j\), and \(m\) denotes the number of reviews to the product \(p_j\). Therefore, it can be seen that the feature’s score \((FeatureScore_{f_l})\) regarding a product \(p_j\) is computed by averaging all opinions that were associated with this feature in the product’s reviews. Then, the product’s satisfying score is calculated via:

\[
ProductScore(\hat{u}, p_j) = \sum_{w_{f_l}(u) \in w_{\hat{u}}} w_{f_l}(u) \times FeatureScore_{f_l}(p_j)
\]

(3.27)

Still, top-\(N\) products with higher scores are recommended to the buyer.

3.3.4 Results Analysis

In this section, we first show the results from comparing the three major methods: LCRM*, PRM-k-NN, PRM-k-Means\(^5\). We then present the results from testing two variants of LCRM*, i.e., LCRM*\(^r\)-k-NN and LCRM*\(^r\)-k-Means, in comparison with their counterparts. Finally, we identify the performance difference between LCRM* and LCRM*\(^r\)-k-

---

\(^5\) For each method, the parameters’ optimal values were first tuned through the experiment. In the digital camera dataset, the optimal number of clusters is 6 (i.e., \(K = 6\)) for LCRM*, LCRM*\(^r\)-k-Means and PRM-k-Means, and the optimal neighborhood size in all k-NN based methods is 15 (i.e., \(k = 15\)). In the laptop dataset, \(K = 6\) and \(k = 40\) for LCRM* and LCRM*\(^r\)-k-Means, \(K = 8\) and \(k = 15\) for PRM-k-Means, \(k = 40\) for LCRM*\(^r\)-k-NN, and \(k = 20\) for PRM-k-NN.
CHAPTER 3. ESTIMATING MULTI-FEATURE WEIGHT PREFERENCES FROM USER REVIEWS

Means, with the particular focus on their clustering process. A summary of these abbreviations’ descriptions can be referred to Table 3.3.

Table 3.3: List of tested algorithms in the experiment

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Algorithm description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Review</td>
<td>Generating recommendation without considering product reviews</td>
<td>Section 3.3.3</td>
</tr>
<tr>
<td>Review-Rank</td>
<td>Ranking products by taking into account features’ review opinions</td>
<td>Section 3.3.3</td>
</tr>
<tr>
<td>PRM-k-NN</td>
<td>PRM-based deriving reviewer-level preferences and k-NN based generating recommendation</td>
<td>Section 3.2.4</td>
</tr>
<tr>
<td>PRM-k-Means</td>
<td>PRM-based deriving reviewer-level preferences and k-Means based generating recommendation</td>
<td>Section 3.2.4</td>
</tr>
<tr>
<td>LCRM*</td>
<td>Extending LCRM to derive both reviewer-level and cluster-level preferences</td>
<td>Section 3.2.5</td>
</tr>
<tr>
<td>LCRM*-k-NN</td>
<td>LCRM*-based deriving reviewer-level preferences and k-NN based generating recommendation</td>
<td>Section 3.2.6</td>
</tr>
<tr>
<td>LCRM*-k-Means</td>
<td>LCRM*-based deriving reviewer-level preferences and k-Means based generating recommendation</td>
<td>Section 3.2.6</td>
</tr>
</tbody>
</table>

Overall Comparison

Tables 3.4 & 3.5 list the results in terms of both Hit-Ratio and MRR metrics on digital camera dataset and laptop dataset respectively. The superscript annotations in tables indicate the significant level from pair-wise comparisons.

First of all, it can be seen that PRM-k-NN, PRM-k-Means, and LCRM* all perform significantly better than the two baseline methods (i.e., Non-Review and Review-Rank) with respect to both metrics. Specifically, Review-Rank that simply utilizes features’
opinion values to perform product ranking cannot compete with PRM-based methods and LCRM*, as the latter ones target at building the feature preferences based similarity relationship between the buyer and reviewers. Moreover, it shows that the outperforming accuracy of these *preference-based review-incorporated methods* is more obvious when the buyer’s preferences are less complete (e.g., over 4 or 6 features in camera dataset) as shown in Tables 3.4 & 3.5, inferring that they can more accurately predict the buyer’s un-stated preferences by relating her/him to like-minded reviewers.

Furthermore, LCRM* is shown more accurate than PRM-k-NN and PRM-k-Means on both datasets in most conditions. For example, when the buyer’s preference is stated over 6 features in camera dataset (i.e., the preference size is 6), the Hit-Ratio achieved by LCRM* is 0.305 when N is 10, which is up to 45.2% higher than the one by PRM-k-Means, and 43.9% than the one by PRM-k-NN. The MRR value of LCRM* is also significantly higher than the ones of PRM-k-NN and PRM-k-Means. Combining the two metrics’ results, we can infer that LCRM* not only increases the chance of including users’ target choice in the recommendation list, but also ranking the target choice in top positions in the list. Moreover, the comparison between LCRM*/PRM-k-Means and PRM-k-NN reveals the positive impact of integrating the clustering process on identifying inherently more similar reviewers to the current buyer.

**Evaluation on Reviewer-Level Preferences**

As mentioned in Section 3.2.6, in order to distinguish the particular effect of LCRM* on deriving reviewer-level preferences, we tested its two variants, LCRM*-r-k-NN and LCRM*-r-k-Means (referred to Table 3.3). The comparison between LCRM*-r-k-NN and PRM-k-NN indicates that the former is superior to the latter at varied sizes of the buyer’s
Table 3.4: Comparison of algorithms w.r.t. Hit Ratio and MRR with varied preference sizes in digital camera dataset (the maximal preference size is 10)

<table>
<thead>
<tr>
<th>Preference size</th>
<th>Method</th>
<th>Evaluation metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$H@10$</td>
</tr>
<tr>
<td>4 features</td>
<td>1Non-Review</td>
<td>0.109</td>
</tr>
<tr>
<td></td>
<td>2Review-Rank</td>
<td>0.119$^{1}$</td>
</tr>
<tr>
<td></td>
<td>3PRM-k-NN</td>
<td>0.207$^{1,2}$</td>
</tr>
<tr>
<td></td>
<td>4PRM-k-Means</td>
<td>0.201$^{1,2}$</td>
</tr>
<tr>
<td></td>
<td>5LCRM*</td>
<td>0.234$^{1,2,3,4}$</td>
</tr>
<tr>
<td></td>
<td>6LCRM* -k-NN</td>
<td>0.234$^{1,2,3,4}$</td>
</tr>
<tr>
<td></td>
<td>7LCRM* -k-Means</td>
<td>0.234$^{1,2,3,4}$</td>
</tr>
<tr>
<td>6 features</td>
<td>1Non-Review</td>
<td>0.124</td>
</tr>
<tr>
<td></td>
<td>2Review-Rank</td>
<td>0.117</td>
</tr>
<tr>
<td></td>
<td>3PRM-k-NN</td>
<td>0.210$^{1,2}$</td>
</tr>
<tr>
<td></td>
<td>4PRM-k-Means</td>
<td>0.212$^{1,2}$</td>
</tr>
<tr>
<td></td>
<td>5LCRM*</td>
<td>0.305$^{1,2,3,4,6,7}$</td>
</tr>
<tr>
<td></td>
<td>6LCRM* -k-NN</td>
<td>0.251$^{1,2,3,4}$</td>
</tr>
<tr>
<td></td>
<td>7LCRM* -k-Means</td>
<td>0.260$^{1,2,3,4}$</td>
</tr>
<tr>
<td>8 features</td>
<td>1Non-Review</td>
<td>0.124</td>
</tr>
<tr>
<td></td>
<td>2Review-Rank</td>
<td>0.114</td>
</tr>
<tr>
<td></td>
<td>3PRM-k-NN</td>
<td>0.220$^{1,2}$</td>
</tr>
<tr>
<td></td>
<td>4PRM-k-Means</td>
<td>0.226$^{1,2,3}$</td>
</tr>
<tr>
<td></td>
<td>5LCRM*</td>
<td>0.360$^{1,2,3,4,6,7}$</td>
</tr>
<tr>
<td></td>
<td>6LCRM* -k-NN</td>
<td>0.249$^{1,2,3,4}$</td>
</tr>
<tr>
<td></td>
<td>7LCRM* -k-Means</td>
<td>0.281$^{1,2,3,4,6}$</td>
</tr>
<tr>
<td>10 features</td>
<td>1Non-Review</td>
<td>0.110</td>
</tr>
<tr>
<td></td>
<td>2Review-Rank</td>
<td>0.120$^{1}$</td>
</tr>
<tr>
<td></td>
<td>3PRM-k-NN</td>
<td>0.215$^{1,2}$</td>
</tr>
<tr>
<td></td>
<td>4PRM-k-Means</td>
<td>0.230$^{1,2,3}$</td>
</tr>
<tr>
<td></td>
<td>5LCRM*</td>
<td>0.324$^{1,2,3,4,6,7}$</td>
</tr>
<tr>
<td></td>
<td>6LCRM* -k-NN</td>
<td>0.261$^{1,2,3,4}$</td>
</tr>
<tr>
<td></td>
<td>7LCRM* -k-Means</td>
<td>0.264$^{1,2,3,4,6}$</td>
</tr>
</tbody>
</table>

Note: the superscript indicates that the corresponding algorithm’s accuracy is significantly lower ($p < 0.05$).
## CHAPTER 3. ESTIMATING MULTI-FEATURE WEIGHT PREFERENCES FROM USER REVIEWS

Table 3.5: Comparison of algorithms w.r.t. Hit Ratio and MRR with varied preference sizes in laptop dataset (the maximal preference size is 8)

<table>
<thead>
<tr>
<th>Preference size</th>
<th>Method</th>
<th>Evaluation metrics</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>( H_{@10} )</td>
<td>( H_{@20} )</td>
<td>MRR</td>
<td></td>
</tr>
<tr>
<td>4 features</td>
<td>Non-Review</td>
<td>0.033</td>
<td>0.033</td>
<td>0.021</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Review-Rank</td>
<td>0.032</td>
<td>0.032</td>
<td>0.031</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PRM-k-NN</td>
<td>0.176^1,2</td>
<td>0.231^1,2</td>
<td>0.054^1,2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PRM-k-Means</td>
<td>0.176^1,2</td>
<td>0.223^1,2</td>
<td>0.056^1,2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCRM*</td>
<td>0.214^1,2,3,4,6,7</td>
<td>0.298^1,2,3,4,6,7</td>
<td>0.062^1,2,3,4,6,7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCRM*-k-NN</td>
<td>0.201^1,2,3,4</td>
<td>0.246^1,2,3,4</td>
<td>0.058^1,2,3,4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCRM*-k-Means</td>
<td>0.201^1,2,3,4</td>
<td>0.273^1,2,3,4,6</td>
<td>0.061^1,2,3,4</td>
<td></td>
</tr>
<tr>
<td>6 features</td>
<td>Non-Review</td>
<td>0.053</td>
<td>0.042</td>
<td>0.025</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Review-Rank</td>
<td>0.043</td>
<td>0.043</td>
<td>0.032</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PRM-k-NN</td>
<td>0.195^1,2</td>
<td>0.231^1,2</td>
<td>0.061^1,2,4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PRM-k-Means</td>
<td>0.183^1,2</td>
<td>0.237^1,2</td>
<td>0.057^1,2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCRM*</td>
<td>0.231^1,2,3,4,6,7</td>
<td>0.321^1,2,3,4,6,7</td>
<td>0.068^1,2,3,4,6,7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCRM*-k-NN</td>
<td>0.225^1,2,3,4</td>
<td>0.239^1,2</td>
<td>0.060^1,2,4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCRM*-k-Means</td>
<td>0.230^1,2,3,4</td>
<td>0.288^1,2,3,4,6</td>
<td>0.066^1,2,3,4,6</td>
<td></td>
</tr>
<tr>
<td>8 features</td>
<td>Non-Review</td>
<td>0.042</td>
<td>0.042</td>
<td>0.028</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Review-Rank</td>
<td>0.044</td>
<td>0.044</td>
<td>0.030</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PRM-k-NN</td>
<td>0.204^1,2</td>
<td>0.252^1,2</td>
<td>0.063^1,2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PRM-k-Means</td>
<td>0.205^1,2,3</td>
<td>0.268^1,2,3</td>
<td>0.071^1,2,3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCRM*</td>
<td>0.239^1,2,3,4</td>
<td>0.331^1,2,3,4,6,7</td>
<td>0.078^1,2,3,4,6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCRM*-k-NN</td>
<td>0.235^1,2,3,4</td>
<td>0.267^1,2,3</td>
<td>0.072^1,2,3,4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCRM*-k-Means</td>
<td>0.238^1,2,3,4,6</td>
<td>0.285^1,2,3,4,6</td>
<td>0.074^1,2,3,4,6</td>
<td></td>
</tr>
</tbody>
</table>

*Note:* the superscript indicates that the corresponding algorithm’s accuracy is significantly lower \( p < 0.05 \).
preferences, w.r.t. both Hit-Ratio and MRR (see Tables 3.4 & 3.5 and Figure 3.2). For instance, LCRM\textsuperscript{*r}-k-NN achieves significantly higher $H_{@10}$ than PRM-k-NN at preference size 6, e.g., 0.251 vs. 0.210, $t = 5.047$, $p < 0.01$ in camera dataset. The similar finding appears in the comparison between LCRM\textsuperscript{*r}-k-Means and PRM-k-Means. Taking $H_{@10}$ and preference size @ 6 as an example, the Hit-Ratio of LCRM\textsuperscript{*r}-k-Means is 0.260 in camera dataset, which is significantly higher than the one of PRM-k-Means (that is 0.212; $t = 5.802$, $p < 0.01$; similar phenomena are shown in laptop dataset). These observations highlight the positive impact of extension that we made to the original LCRM. In order words, it is demonstrated that LCRM* is more accurate in terms of deriving single reviewer’s feature preferences than PRM based methods.

In addition, from Figure 3.2 we can verify again the effect of integrating clustering process on improving recommendation accuracy, since the clustering based LCRM\textsuperscript{*r}-k-Means performs better than the k-nearest neighbor based recommendation method LCRM\textsuperscript{*r}-k-NN in both datasets (the same appears in the comparison between PRM-k-Means and PRM-k-NN), although some differences do not reach at significant level. We were then driven to further compare LCRM* and LCRM\textsuperscript{*r}-k-Means, with the focus on their exclusive difference in respect of the clustering procedure, so as to identify which clustering method is more effective.

**Evaluation on Cluster-Level Preferences**

LCRM* and LCRM\textsuperscript{*r}-k-Means differ only in the way of clustering reviewers and generating cluster-level preferences (with other steps identical between them): LCRM* is model-based, while LCRM\textsuperscript{*r}-k-Means performs the heuristic k-Means clustering. The comparative results show that LCRM*-based clustering method is more accurate than the k-Means
CHAPTER 3. ESTIMATING MULTI-FEATURE WEIGHT PREFERENCES FROM USER REVIEWS

Figure 3.2: Comparison among PRM-k-NN, PRM-k-Means, LCRM*-k-NN, and LCRM*-k-Means, w.r.t. $H_{@10}$.

Based, given that both Hit-Ratio and MRR returned by LCRM* are significantly better in most cases (see Tables 3.4 & 3.5 and Figure 3.3). For instance, in the camera dataset, LCRM* achieves 0.305 at $N_{@10}$ when the preference size is 6, versus 0.260 by LCRM*-k-Means ($t = 3.665, p < 0.05$). When the buyer’s preferences become more complete, say to preference size 8, the accuracy of LCRM*-k-Means method is slightly increased to 0.281, but still lower than the one of LCRM* (which is 0.360; $t = 2.386, p < 0.05$).

It hence suggests that the LCRM* clustering based approach, which lies on the division of the whole population of reviewers according to their membership probability, is shown more effective than the heuristic k-Means clustering based method.

Figure 3.3: Comparison between LCRM*-k-Means and LCRM*, w.r.t. $H_{@10}$.
CHAPTER 3. ESTIMATING MULTI-FEATURE WEIGHT PREFERENCES FROM USER REVIEWS

3.3.5 Discussion

The experimental results thus validate several hypotheses we brought forward at the beginning: 1) the proposed two branches of approaches, respectively based on Probabilistic Regression Model (PRM) and Latent Class Regression Model (LCRM), are both more effective than the two baselines Non-Review and Review-Rank; 2) as for deriving the reviewer-level preferences, LCRM* performs better than the PRM-based methods (e.g., LCRM*-k-NN vs. PRM-k-NN); 3) as for the clustering of reviewers, first of all, the methods that involve the clustering process are better than the ones without (e.g., LCRM* & LCRM*-k-Means vs. LCRM*-k-NN); secondly, LCRM* acts more positive in terms of clustering reviewers and generating the cluster-level preferences, as per LCRM* vs. LCRM*-k-Means. As a summary, we can infer the following advantageous relation among these compared methods:

\[ LCRM^* > LCRM^{*r}-k-Means > LCRM^{*r}-k-NN > PRM-k-Means > PRM-k-NN > Review-Rank > Non-Review. \]

Therefore, it implies that LCRM* well addresses the data sparsity issue in the high-risk product domains, as it considers the preference homogeneity among reviewers when clustering them and further leverages the clustering outcomes into refining individual reviewers’ preferences. In comparison, PRM purely relies on each reviewer’s self-provided information to derive her/his preferences, which is unavoidably subject to be biased and likely result in over-fitting phenomenon in the situation with sparse reviews. The comparison results regarding the two variants of LCRM*, i.e., LCRM*-k-NN and LCRM*-k-Means, highlight the particular value of LCRM* in unifying the ideal solutions to derive both reviewer-level and cluster-level preferences within the same framework. Specifically, the cluster-level preferences can be exploited to identify the truly like-minded reviewers.
and accelerate the filtering process, while the reviewer-level preferences can be utilized to adjust each reviewer’s contribution when calculating a product’s prediction score. More notably, our main idea of constructing reviewers’ multi-feature preferences from reviews is empirically demonstrated. They can help identify inherently more relevant reviewers to the current buyer and hence more accurately retrieve the buyer’s target choice. The method’s practical usage in real e-commerce sites for saving the system’s preference elicitation effort is thus suggested.

3.4 Summary

In this chapter, we have emphasized deriving reviewers’ weight feature preferences from both textual reviews and overall ratings that they provided. More specifically, we have investigated two regression models. The first is the Probabilistic Regression Model (PRM) based on which we incorporated the opinion values associated with various features into inferring the weight a reviewer placed on each feature. Such preferences were used to perform the k-nearest neighborhood and k-Means recommendation algorithms (i.e., PRM-k-NN and PRM-k-Means). We have further extended Latent Class Regression Model (LCRM) with the aim to derive both cluster-level preferences and reviewer-level preferences simultaneously (so called LCRM*). Concretely, the clustering was performed based on the whole population’s structure and the membership probability, which was then leveraged to refine individual reviewers’ preferences so that the inherent preference similarity between reviewers can be taken into account. We have additionally implemented two variants of LCRM*: LCRM*-r-k-NN and LCRM*-r-k-Means, for the comparison in the experiment. In total, seven methods were tested, including two baselines Non-Review and Review-Rank. The experiment demonstrates the outperforming accuracy of LCRM* from
several aspects: 1) deriving more stable reviewer-level weight preferences; 2) performing more effective clustering of reviewers; and 3) generating more accurate recommendation even when the buyer’s stated preferences were less complete. Our research hence highlights the value of deriving reviewers’ multi-feature weight preferences from product reviews.
Chapter 4

Estimating Multi-Attribute Value Preferences from User Reviews

4.1 Introduction

In the previous chapter, we considered a user’s preference as importance weights he placed on different product features; in this chapter, the preference is defined on multiple attributes’ values (or static specifications), which is hence named as multi-attribute value preferences. In practice, most buyers are in reality not able to state their complete preferences over all attributes due to their cognitive limitation and/or unfamiliarity with the product domain, even when they are involved in a conversational interaction with the system (Viappiani et al., 2006; Chen and Pu, 2012).

The online product reviews generated by the users who previously purchased products usually express some positive or negative opinions on the products configured with some specific attributes. For example, if a reviewer states positive opinions on a specific attribute of a product, we deduce that the reviewer satisfy the attribute of that product. Similarly, if he state negative opinions, we deduce that the dislike the attribute of that product. Thus, it is intuitively understandable that the aspect-level opinions can reflect
the inherent preferences of the reviewers on products’ attributes. It’s worth to note that
attribute refer to the product’s static specifications, while aspects are features discovered
from reviews. The latter is mapped to the former through a pre-defined dictionary. Inspired
from this idea, we aim to recover the reviewer’s preferences for product attributes from re-
views. Therefore, in the following, we propose a novel preference enrichment framework,
which aims to complete a new buyer’s multi-attribute value preferences by incorporating
reviewers’ aspect-level opinions and the aspects’ static specifications. Specifically, by in-
tegrating with the fine-grained opinion mining results of textual reviews, we target to find
like-minded reviewers for a target new buyer and hence enrich the buyer’s preferences on
all attributes.

4.2 Methodology

4.2.1 Problem Statement

In this work, we assumed that each user specifies preferences on a subset of product at-
tributes, e.g. price $< 300. We also have some auxiliary data of online reviews on a set of
products \( P = \{p_1, p_2, \ldots, p_P\} \). Our goal is to enrich the partial preferences of each new
user, and then recommend a personalized ranked list of products to him.

Specifically, given a set of products \( P \), each product \( p_j \) usually consists a set of pre-
defined attributes \( (a_1, a_2, \ldots, a_A) \) associated with specific values (e.g., “price” = $264.95
and “screen size” = 1.5 inches, etc.), the goal is to identify an ideal product to the user
maximizing the degree of satisfaction by meeting his/her value preferences on a set of at-
tributes. Additionally, we observe that the product is often described using different types
of attributes (numerical, original, and category). To support such distinct attribute types,
all of the attribute values are represented as attribute levels, as shown in Table 4.1.
cretely, in our experiment, an attribute’s levels are determined based on the distribution of the values of that attribute. It’s worth noting that the levels should cover the full range of possibilities for existing products.

Table 4.1: The attributes and associated value levels for digital cameras

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Brand</th>
<th>Screen Size</th>
<th>Resolution</th>
<th>Price</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cannon</td>
<td>≤ 1.0”</td>
<td>≤ 16 megapixels</td>
<td>≤ 200$</td>
<td>≤ 250g</td>
</tr>
<tr>
<td></td>
<td>Casio</td>
<td>1.0” ~ 1.5”</td>
<td>16 ~ 18 megapixels</td>
<td>200$ ~ 250$</td>
<td>250g ~ 500g</td>
</tr>
<tr>
<td></td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td></td>
<td>Olympus</td>
<td>≥ 3.0”</td>
<td>≥ 20 megapixels</td>
<td>≥ 800$</td>
<td>≥ 1750g</td>
</tr>
</tbody>
</table>

Mathematically, a product $p_j$ consisting of $A$ attributes is represented by a vector $x_j = [x_{jk}]_{A \times 1} \in \mathbb{R}^{A \times 1}$, in which the value $x_{jk}$ refers attribute $a_k$ of product $p_j$. In what follows, the preferences of users are initially described using constraints/criteria over certain attributes in term of their values. For example, the price should be less than $300 and the resolution is larger than 2300. Then these preferences are represented as a preference set $\phi_u = [\phi_{uk}]_{A \times 1}$, in which each element $\phi_{uk}$ specifies the value preference for attribute $a_k$ (e.g., weight $< 250g$).

4.2.2 Multi-attribute Value Preferences Estimation

Our proposed solution, called preference completion and ranking (henceforth called CompleteRank), mainly contains three steps: (1) mine reviewers’ aspect-level opinions, which are fine-grained preferences compared with that of overall ratings on products, (2) derive reviewers’ preferences over attributes’ static specifications via their aspect-level opinions, which are then used to complete the new buyer’s preferences, and (3) rank products and
recommend ones with higher matching degrees according to the buyer’s enriched preferences.

**Step 1: Aspect-Level Opinion Mining**

Since reviews are written in natural language, we need to first extract the aspects and opinions from a large amount of reviews automatically. This issue was addressed in the previous Chapter [3] that is capable of identifying aspect-level opinions from a review as described (see Section [3.2.3]). Basically, there are three sub-steps:

1. Identify all (aspect, opinion) pairs in a review through the Part-of-Speech tagger[^1] (which is for extracting frequent nouns and noun phrases as aspect candidates), syntactic dependency parser[^2] (which is for identifying opinion words) and WordNet (for grouping synonymous aspects) (Fellbaum, 1998);

2. Quantify the opinion’s sentiment strength (also called *polarity*) by applying SentiWordNet (Esuli and Sebastiani, 2006). Formally, the aspect-level opinion is classified as negative (-1), positive (1), or neutral (0);

3. Map the opinion to the attribute’s static specification in the structured form (*attribute, opinion, specification*), for example, (*weight, positive, 200g*) which indicates that the reviewer expresses positive opinion on the product’s weight that is 200g. We can hence infer the reviewer’s preference on the corresponding attribute (e.g., the interval of weight that reviewer prefers lies in the range containing 200).

The procedures described above results in a structured representation of each product review, as denoted in Table [4.2]. Here, the value 1 or -1 indicates the opinion value of

[^1]: http://nlp.stanford.edu/software/tagger.shtml
CHAPTER 4. ESTIMATING MULTI-ATTRIBUTE VALUE PREFERENCES FROM USER REVIEWS

the corresponding attribute \( \{a_k\}_{k=1}^A \) expressed in review, and the symbol “?” indicates that attribute is not mentioned in the review.

Table 4.2: The representation of the structured product reviews

<table>
<thead>
<tr>
<th>Reviewer</th>
<th>Product</th>
<th>Attributes</th>
<th>Overall Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u = 1 )</td>
<td>( p = 5 )</td>
<td>( a_1 \quad ? \quad \cdots \quad 1 \quad ? )</td>
<td>4</td>
</tr>
<tr>
<td>( \vdots )</td>
<td>( \vdots )</td>
<td>( \vdots \quad \cdots \quad \vdots \quad \vdots )</td>
<td>( \vdots )</td>
</tr>
<tr>
<td>( u = U )</td>
<td>( p = 2 )</td>
<td>( -1 \quad 1 \quad \cdots \quad 1 \quad 1 )</td>
<td>5</td>
</tr>
</tbody>
</table>

Step 2: Preference Completion

As described in the introduction section, the opinions expressed in a review is a good indicator reflecting the preferences on product attributes held by the reviewer. The reviewer \( u \)'s preference \( \tilde{\phi}_{uk} \) on the attribute \( a_k \) can be deduced from the opinions he expressed on attribute \( a_k \) in his review. If a reviewer states positive opinion on a specific attribute of a product, we deduce that the reviewer likes the value (level) of that attribute associated with that product. Likewise, if he states negative opinions, we deduce that he dislikes the specific attribute of that product. In general, we can divide all types of attributes into three classes. The first class of attributes carries the default preferences “the-larger-the-better” that are assumed applicable to all users, such as the preferences for attribute “resolution” of a camera. The second class of attribute carries the default preferences “the-smaller-the-better”, such as the “price” attribute of a product. The third class of attribute is discrete type, such as the “brand” of a product, would not carry default preferences. In this work, we add some default preferences to the reviewer’s preference model \( \tilde{\phi}_u \). For example, if a reviewer \( u \) expresses positive opinion on the price attribute in a review of a digital cam-
era with price $200, we assume that the reviewer would prefer price less than or equal to $200, which is used to enrich his preferences $\tilde{\Phi}_u$. In another case, if a reviewer does not comment the “price” attribute of a product, we can assume that he is satisfied with this attribute when the overall rating is more than 3.

Then, for the current buyer, if his preference on attribute $a$ is not missing, similar reviewers’ opinions regarding this attribute are used to adjust the buyer’s preference (i.e., in order to fuse the reviewers’ preferences); otherwise, they are adopted to predict the buyer’s preference on that attribute (i.e., which interval he most likely prefers). Moreover, through experiments, we observe that the preferences with adjustment achieve better result than the original stated preferences. Concretely, for each new buyer $\hat{u}$, we complete his preferences on un-stated attributes with the help of some like-minded reviewers’ preferences, 

$$
\Phi_{\hat{u}k} = \begin{cases} 
\left( \phi_{\hat{u}k} + \sum_{u \in N_{\hat{u}}} \tilde{s}_{\hat{u}u} \tilde{\Phi}_{uk} \right) / 2, & \text{if } \phi_{\hat{u}k} \text{ is not missing} \\
\sum_{u \in N_{\hat{u}}} \tilde{s}_{\hat{u}u} \tilde{\Phi}_{uk}, & \text{otherwise}
\end{cases}
$$

(4.1)

where $\tilde{s}_{\hat{u}u} = \frac{s_{\hat{u}u}}{\sum_{u \in N_{\hat{u}}} s_{\hat{u}u}}$ is the normalized similarity between buyer $\hat{u}$ and reviewer $u$. The similarity is calculated as

$$
s_{\hat{u}u} = \sum_{k=1}^{A} y_{\hat{u}k} \times \cos(\Phi_{\hat{u}k}, \Phi_{uk})
$$

(4.2)

where $\Phi_{\hat{u}k}$ and $\Phi_{uk}$ are vector representations of attribute $a_k$ which is defined with multiple intervals. For instance, suppose $a_k$ is the camera’s “weight” which is classified into 8 intervals: $[0, 200)$, $[200, 400)$, . . . , and $[1200, 1400)$ and a buyer $\hat{u}$ prefers the “weight” in ranges of $[200, 400)$ and $[800, 1000)$, then we represent the vector $\Phi_{\hat{u}k}$ as $[0, 1, 0, 0, 1, 0, 0, 0]$. Note that $y_{\hat{u}k}$ is the preference indicator, $y_{\hat{u}k} = 1$ if the preference on attribute $a_k$ is stated by the current buyer $\hat{u}$, and $y_{\hat{u}k} = 0$ otherwise. We illustrate the preference completion procedure in Figure 4.1. Note that we use $|N_{\hat{u}}| = 300$ for the size of group of like-minded reviewers in our experiment.
4.2.3 Recommendation Generation

With the enriched user preferences, we can then calculate the matching score between a buyer $\hat{u}$ and a product $p_j$,

$$\text{ProductScore}(\hat{u}, p_j) = \frac{1}{A} \sum_{k=1}^{A} \text{match}_w(\bar{\phi}_{\hat{u}k}, x_{jk})$$  (4.3)

where $\text{match}_w(\bar{\phi}_{\hat{u}k}, x_{jk}) = \langle \bar{\phi}_{\hat{u}k}, x_{jk} \rangle$ is the inner product of the expanded vectors of the buyer $\hat{u}$’s preference $\bar{\phi}_{\hat{u}k}$ and product $p_j$’s value on attribute $a_k$. The obtained matching scores can then be used to rank products. The ones with highest scores will be recommended to the target buyer.

4.3 Experiment

4.3.1 Experimental Setup and Dataset

We have two data sets, one collected from a previous user study (Chen and Pu 2008) and the other from Amazon review data. In our user study data, there are 57 users ($n = 57$) and 64 digital cameras ($P = 64$), where each product has 8 attributes ($A = 8$). Each user explicitly indicated her/his preferences on the product’s attributes. Each user was also asked to check all products and carefully chose one product as her/his favourite product,
CHAPTER 4. ESTIMATING MULTI-ATTRIBUTE VALUE PREFERENCES FROM USER REVIEWS

denoted as \( \text{choice}(u) \) (i.e., the user’s target choice). For each product, we crawled the corresponding reviews from the Amazon website (http://www.amazon.com/), which contain 4904 pieces of reviews from 4904 reviewers (as each reviewer posted only one review among those products). In our experiment, for each of 57 users, we randomly select 2, 4, or 6 of her/his attribute preferences to represent the simulated buyer’s partial preferences (e.g., 2 means that the buyer just stated preferences on 2 attributes).

4.3.2 Evaluation Metric

For each user \( u \), there is a target choice in the product set, i.e., \( \text{choice}(u) \), which is taken as the ground truth in our evaluation. We use hit ratio to evaluate the recommendation accuracy,

\[
H@N = \frac{1}{n} \sum_{u=1}^{n} \delta(\text{position}(\text{choice}(u)) \leq N)
\]  

(4.4)

where \( \text{choice}(u) \) is the target choice of user \( u \), \( \text{position}(\text{choice}(u)) \) denotes the corresponding ranking position, and \( n \) is the number of users. Note that \( \delta(z) = 1 \) if \( z \) is true and \( \delta(z) = 0 \) otherwise. In our experiment, we use \( N = 10 \), since a typical user only checks a few products which are placed in top positions (Chen and Pu, 2011).

4.3.3 Compared Baseline Methods

We compared our proposed solution with the following four baseline methods (most of which are from related literatures).

**Random** We randomly rank the products for each target user. The result is calculated as \( 10/64 = 0.1563 \).

**PopRank** We calculate the popularity of each product among the reviewers. A product is usually considered as preferred by a reviewer if the rating is larger than 3 in 5-star...
numerical ratings (Sindhwani et al., 2009). The popularity of the product $p_j$ among the reviewers $\mathcal{U}$ can then be estimated as:

$$Pop(p_j) = \frac{1}{|\mathcal{U}|} \sum_{i=1}^{U} \delta(y_{ij} > 3)$$

The obtained popularity $0 \leq Pop(p_j) \leq 1$ can be used to rank the products. Note that PopRank is not a personalized method since the popularity is user independent.

**PartialRank** For each user $\hat{u}$ and product $p_j$, we calculate the matching score between the user’s stated (partial) preferences and the product’s profile as follows:

$$M_{\hat{u}p_j} = \frac{1}{A} \sum_{k=1}^{A} y_{\hat{u}k} \times match(\phi_{\hat{u}k}, x_{jk})$$

where $match(\phi_{\hat{u}k}, x_{jk}) = 1$ if the attribute’s static specification $x_{jk}$ satisfies the user preference $\phi_{\hat{u}k}$, and $match(\phi_{\hat{u}k}, x_{jk}) = 0$ otherwise. The obtained matching scores, $0 \leq M_{\hat{u}p_j} \leq 1$ with $j = 1, \ldots, P$, can then be used to rank the products for user $\hat{u}$.

**HybridRank** For each attribute $a_k$ of product $p_j$, we can calculate the average opinion score from the reviewers, i.e. $opinion(p_j, a_k) \in [-1, 1]$, and the product $p_j$’s overall opinion score via the method proposed in (Aciar et al., 2007), which is defined as:

$$O_{\hat{u}p_j} = \frac{1}{A} \sum_{k=1}^{A} y_{\hat{u}k} \times opinion(p_j, a_k)$$

Then, with the preference matching score $M_{\hat{u}p_j}$ and opinion score $O_{\hat{u}p_j}$, a hybrid score is produced for the product $p_j$,

$$H_{\hat{u}p_j} = \frac{1}{2} (M_{\hat{u}p_j} + O_{\hat{u}p_j})$$

The obtained scores, $-1 \leq H_{\hat{u}p_j} \leq 1$ with $j = 1, \ldots, P$, are used to rank the products for user $\hat{u}$.
Table 4.3: The recommendation performance of CompleteRank and baselines. Note that for PartialRank, HybridRank and CompleteRank, we randomly take 2, 4, 6 attributes for five times to simulate *partial preferences*, and the average performance of those five runs are reported.

<table>
<thead>
<tr>
<th>Method</th>
<th>Given 2</th>
<th>Give 4</th>
<th>Given 6</th>
<th>Given 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.1563</td>
<td>0.1563</td>
<td>0.1563</td>
<td>0.1563</td>
</tr>
<tr>
<td>PopRank</td>
<td>0.2456</td>
<td>0.2456</td>
<td>0.2456</td>
<td>0.2456</td>
</tr>
<tr>
<td>PartialRank</td>
<td>0.1825 (±0.0457)</td>
<td>0.2211 (±0.0342)</td>
<td>0.2772 (±0.0288)</td>
<td>0.3158</td>
</tr>
<tr>
<td>HybridRank</td>
<td>0.2386 (±0.0440)</td>
<td>0.2456 (±0.0447)</td>
<td>0.2947 (±0.0192)</td>
<td>0.2982</td>
</tr>
<tr>
<td>CompleteRank</td>
<td><strong>0.2807 (±0.0372)</strong></td>
<td><strong>0.3088 (±0.0437)</strong></td>
<td><strong>0.3158 (±0.0277)</strong></td>
<td><strong>0.3333</strong></td>
</tr>
</tbody>
</table>

### 4.3.4 Results Analysis

The results are shown in Table 4.3, from which we can have the following observations, (1) our proposed solution CompleteRank is much better than all baselines, which clearly shows the effectiveness of our preference enrichment idea, especially for the buyers with partial preferences; (2) PopRank is better than Random, which demonstrates the usefulness of the online review data for augmenting new-user recommendation; (3) PartialRank performs worse than PopRank given 2 and 4 attribute preferences, but better than PopRank when given 6 and 8 attribute preferences, which shows the effect of taking into account users’ preferences (especially when they are nearly complete) on recommending products; and (4) HybridRank performs better than PartialRank in most cases, which shows the usefulness of the combining the product’s static specifications (by matching to users’ preferences) and reviewers’ opinions, though it is still worse than our solution.
4.4 Summary

We proposed a novel solution, CompleteRank, for solving the problem of new-user recommendation with partial multi-attribute value preferences. Specifically, we designed a preference enrichment approach via incorporating the mined reviewers’ aspect-level opinions on products’ static specifications. The completed preferences of a new user are then used to match the product profiles, by which the products with highest matching scores are recommended to the target user. Experimental results show that our solution can provide more accurate personalized recommendations than several baseline methods.
Chapter 5

Deriving New Users’ Multi-Attribute Preference Model

5.1 Introduction

So far, we have proposed two review-based recommender systems to recommend high-risk products for “new buyer”. In the first one, the user’s preferences are modeled as the *multi-feature weight preferences* that represent the weights the user places on different product features. In the second one, the user’s preferences are represented as the *multi-attribute value preferences* that indicate the user’s preferences on the product’s attribute values. In this chapter, we will investigate how to combine *weight preferences* and *value preferences* as the user’s multi-attribute preferences (called MAUT-based preferences as described in Section 1.2) with the purpose of providing higher quality product recommendations.

Based on the previous two methods, we can observe that the estimations of reviewers’ weight and value preferences largely depend on the results of aspect-level opinion mining. Specifically, the following three opinion mining tasks are involved: 1) identifying the aspects mentioned in a review of a product, 2) estimating the rating of each aspect based

---

1In this thesis, the term “aspect” and “feature” are used with same meaning.
on the sentiments expressed in the review, and 3) estimating the aspect-level weights that reflect how much reviewers care about different aspects of a product. The relationship between these three tasks is not independent in that the output of one task is the input of another task. Hence, in the following, we first develop a unified framework to improve the three tasks simultaneously.

Then, with the results of the aspect-level weights and ratings inferred from reviews, we recover the reviewer’s weight and value preferences, which can then be utilized to enhance the new buyer’s MAUT-based preferences. In the recommendation procedure, the utility of each product for a new buyer can be calculated based on the enhanced MAUT-based preferences.

5.2 Methodology

5.2.1 Aspect-based Ratings and Weights Estimation

Indeed, existing approaches to aspect-based opinion mining have some limitations that restrict their use in practice. For example, in Section 3.2.3, we have implemented an frequency-based approach to estimating reviewers’ aspect-level ratings. However, this approach is based on a sentiment lexicon, which contains a static sentiment score for each word without considering the aspect it is related to. For example, although the word “friendly” can be a strong positive opinion word for the “service” aspect, but not for the “value” aspect in hotel reviews. Moreover, for the task of identifying aspects from reviews, many of the related works require a set of labeled entities to be prepared in advanced (Jin et al., 2009, Li et al., 2010, Qi and Chen, 2011). What’s more, we aim not only to identify aspects mentioned in product reviews and reviewers’ opinions about these aspects at a fine granularity, but to derive reviewers’ weights on these aspects (see Figure 5.1). The main
challenge is how to minimise error propagation when performing the three tasks. Error propagation occurs when errors caused by an upstream sub-task propagate to and adversely affect the performance of downstream sub-tasks. In this chapter, we are interested in investigating the relationship among the three tasks: *aspect identification*, *aspect-based rating inference*, and *aspect-based weight estimation*. To the best of our knowledge, most of related works have just focused on one or two of these tasks (Hu and Liu, 2004b, Yu et al., 2011, Lu et al., 2009). In our view, these tasks are essentially inter-connected between each other. The accuracy of aspect identification can influence the performance of aspect-based rating inference. Errors may be accumulated if the tasks are performed separately. Particularly, we address this problem by using shared representations to create dependencies between the tasks and thereby recast them as three components of a joint learning task. This enables knowledge transfer between tasks. Specifically, we propose a unified unsupervised CARW (shorted from *Collectively estimate Aspect Ratings and Weights*) model. The data-sparsity problem is also solved by discovering cluster-level preferences for accommodating reviewers’ preference similarity.

Figure 5.1: The inter-connected three tasks related to aspect-based opinion mining.

Formally, we assume that we have a set of $U$ users, which can be denoted as $\mathcal{U} = \{u_1, \ldots, u_U\}$, and a set of $P$ products (such as laptops or digital cameras), which can be denoted as $\mathcal{P} = \{p_1, \ldots, p_P\}$. Then, we let $\mathcal{R} = \{r_{ij}| u_i \in \mathcal{U} \text{ and } p_j \in \mathcal{P}\}$ be a set of
reviews that have been posted to certain products. Typically, when writing a review $r_{ij}$, the user $u_i$ also assigns an overall rating $y_{ij} \in \mathbb{R}^+$ (say from 1 to 5) to express the overall quality of the reviewed product $p_j$. The set of words occurring in the review $r_{ij}$ can be represented as $t_{ij}$. We also assume that there are $W$ unique words $\mathcal{W} = \{t_1, \ldots, t_W\}$ occurring in all of the reviews.

In this section, before presenting details of our CARW model, we first list some assumptions:

- The text describing a particular aspect is generated by sampling words from a topic model (i.e., a multinomial word distribution) corresponding to the aspect. For example, the words “service”, “staff” and “waiter” are frequently used to describe the aspect “service” in the hotel reviews.

- The rating for an aspect is determined based on the words describing the corresponding aspect. For example, if the review text says “the staff are very friendly and helpful”, we can infer the rating for the aspect “service” as 5 (within the range $[1, 5]$) because the opinion expression “very friendly and helpful” indicates a strong positive sentiment.

- The overall rating is regarded as the weighted combination of aspect ratings where the weight reflects the relative emphasis of each aspect. Following this assumption, the overall rating has a linear relationship with the aspect ratings, and the ratings for different aspects are independent from each other. Although the assumption of independence may not be true in reality, this assumption can help to maintain the model’s simplicity (Yu et al., 2011).

- Each product has a distribution over the aspects representing how often different
aspects are discussed in reviews of that product.

• Each product has a rating distribution over aspects that represents how well the product is evaluated on different aspects by reviewers.

• Each reviewer belongs to a cluster so reviewers in the same cluster share similar aspect-based weights.

Based on the above assumptions, to generate a review text, we first sample the aspects expressed in that review conditioned on the aspect distribution of the corresponding product $p_j$. Following the basic Latent Dirichlet Allocation (LDA) model, this distribution follows a multinomial distribution $\theta_j$ with prior Dirichlet distribution $\text{Dir}(\gamma)$, denoted as $\theta_j \sim \text{Dir}(\gamma)$. The aspect-based ratings expressed in a review are then sampled conditioned on the rating distribution of the corresponding product. For the sake of simplicity, we define the aspect rating distribution of product $p_j$ as a multivariate Gaussian distribution $v_j \sim \mathcal{N}(\vartheta_j, \eta_j^2 I)$. The aspect-based weights $w_i$ of reviewer $u_i$ are sampled conditioned on the cluster s/he belongs to and the weight distribution associated with that cluster. The aspect weight distribution is also defined by following a multivariate distribution $w_i \sim \mathcal{N}(\mu_k, \Sigma_k)$, given that the user $u_i$ belongs to the $k$-th cluster (denoted as $c_i = k$). The overall rating $y_{ij}$ is sampled based on the aspect-based weights $w_i$ of the reviewer and the aspect-level ratings $v_{ij}$ that follow a Gaussian distribution, denoted as $y_{ij} \in \mathcal{N}(w_i^T v_{ij}, \sigma^2)$. We use $z_{ijl} = k$ to indicate that the $l$-th word in review $r_{ij}$ belongs to the $k$-th aspect. Finally, the words appearing in a review are sampled based on the mapped aspects and their ratings. Figure 5.2 shows the graphical model.
Model Inference and Parameters Learning

Formally, for each review \( r_{ij} \) of product \( p_j \) given by reviewer \( u_i \), the log-posterior probability of the latent variables (note that the latent variables include aspect ratings vector \( v_{ij} \), the word’s topic/aspect identification \( z_{ij} \), and reviewer’s cluster membership \( c_i \) ) is conditioned on the model parameters \( \Phi = \{ \pi_{1:U}, w_{1:U}, \theta_{1:P}, \vartheta_{1:P}, \eta_{1:P}, \mu_{1:K}, \Sigma_{1:K}, \phi, \beta \} \) and the hyperparameters \( \{ \tau, \sigma, \gamma \} \):

\[
\mathcal{L}(\Phi; r_{ij}) = \log P(z_{ij}, v_{ij}, c_i | t_{ij}, y_{ij}, \Phi, \tau, \gamma) \\
= \log P(t_{ij} | v_{ij}, z_{ij}, \phi, \beta) + \log P(y_{ij} | v_{ij}, \sigma^2) + \log P(c_i | \pi_i, w_i).
\]

In the above equation, the log-likelihood probability of the observed words \( t_{ij} \) given the aspect assignments \( z_{ij} \) and ratings \( v_{ij} \) is defined as

\[
\log P(t_{ij} | v_{ij}, z_{ij}, \phi, \beta) = \sum_{l=1}^{N \{ z_{ij} \}} (\phi_{z_{ij}w_l} + \beta_{z_{ij}v_iz_{ij}w_l}),
\]

Figure 5.2: The graphical plate notation for our CARW model
where \( N \) is the number of words contained in a review, \( t_l \) and \( z_l \) indicate the \( l \)-th word and the corresponding word’s aspect assignment, respectively, and \( v_{z_l} \) denotes the rating for aspect \( z_l \). Note that \( \phi_{z_l} \) is indexed by aspect \( z_l \), indicating which words are associated with the aspect. Alternatively, \( \beta_{z_l v_{z_l}} \) is indexed by aspect \( z_l \) and the rating for that aspect is \( v_{z_l} \), so that we can learn the opinion score associated with each word for every aspect.

As mentioned above, given the rating for each aspect in a review and the associated reviewer’s weight on the aspect, the observed overall rating is assumed to be drawn from a Gaussian distribution around \( w_i^T v_{ij} \). Formally, the log-likelihood of the observed overall rating \( y_{ij} \) given the aspect weights \( w_i \) and aspect ratings \( v_{ij} \) is defined as

\[
\log P(y_{ij} | w_i, v_{ij}, \sigma^2) = N(y_{ij} | w_i^T v_{ij}, \sigma^2) = -\frac{1}{2} \ln 2\pi - \frac{1}{2} \ln \sigma^2 - \frac{1}{2\sigma^2}(y_{ij} - \sum_{k=1}^{A} w_{ik} \cdot v_{ijk})^2
\]

where \( w_{ik} \) and \( v_k \) denote the weight and the rating of the \( k \)-th aspect, respectively. And \( A \) is the total number of discovered aspects.

The log-likelihood of the probability of aspect ratings \( v_{ij} \) and the words’ aspect assignments \( z_{ij} \) with regard to a review of product \( p_j \) is defined as

\[
\log P(v_{ij}, z_{ij} | \theta_j, \vartheta_j) = \log P(z_{ij} | \theta_j) + \log P(v_{ij} | \vartheta_j, \eta_j),
\]

where the probability of aspect assignment of each word \( P(z | \theta_j) \) follows a multinomial distribution with parameter \( \theta_j \), denoted as \( z_{ij} \sim \text{Multinomial}(\theta_j) \), and the aspect-based ratings \( v_{ij} \) follow a multivariate Gaussian distribution with mean as \( \vartheta_j \) and covariance matrix as \( \eta_j I \), denoted as \( z_{ij} \sim N(\vartheta_j, \eta_j I) \). The mean rating \( \vartheta_j \) reflects how much most of reviewers enjoy the product, and the variance parameter \( \eta_j \) shows whether the reviewers agree with each other in terms of their opinions about that product as well as the aspects.

According to the observations from previous Chapter 3.
According to the assumptions that we mentioned at the beginning of this section, within the framework of latent class regression model (LCRM) as described in Chapter 3, the reviewer’s aspect weight can be drawn from a multivariate Gaussian distribution $\mathcal{N}(\mu_k, \Sigma_k)$ given that the reviewer belongs to a cluster $k$. We expect that this clustering procedure could enhance a reviewer’s weight estimation by considering the inner-similarity among reviewers within the same cluster. Formally, the aspect-weight probability of the reviewer $u_i$ belonging to cluster $k$ (denoted as $c_i = k$) is defined as

$$\log P(c_i | \pi_i, w_i) = \log \frac{\pi_{ik} P(w_i | \mu_k, \Sigma_k)}{\sum_{k=1}^{C} \pi_{ik} P(w_i | \mu_k, \Sigma_k)},$$

(5.5)

where $\pi_{ik}$ is the prior probability of the reviewer $u_i$ belonging to the $k$-th cluster.

We now show how to learn the model’s parameters $\Phi$ and the hidden variables $v, z, c$, with regard to each review and each reviewer so as to maximize the log-posterior probability as defined in Eqn 5.1. In this work, the optimization proceeds by coordinating ascent on hidden variables including $\{v, z, c\}$\(^2\) and model parameters $\Phi$, i.e., by alternately performing the following operations:

1. Update hidden variables with fixed parameters

$$\hat{(v, z, c_i)} = \arg\max_{(v,z,c_i)} \mathcal{L}(\Phi; r_{ij}).$$

(5.6)

For each review, the aspect ratings $v$ and the words’ aspect assignments are updated as

$$\hat{z} = \arg\max_{z} \left[ \sum_{k=1}^{A} \sum_{l=1}^{N} \delta(z_l = k) \log P(t_l | z_l, \hat{\theta}_k, \phi, \beta) + \log P(z | \theta_j) \right]$$

(5.7b)

\(^2\)In the following, for the sake of simplicity, we use notation without index to represent parameters.
\[ \hat{v} = \arg\max_v \left[ \sum_{k=1}^A \sum_{l=1}^N \delta(z_l) \log P(t_l | z_l, v_k, \phi, \beta) + \log P(y_{ij} | w_i, v, \sigma^2) + \log P(v | \vartheta_j) \right], \]

where \( \delta(z_l = k) \) is an indicator function denoting that the \( l \)-th word is relevant to the \( k \)-th aspect.

Specifically, for updating each word’s aspect assignment \( z_l \) using above equation \( 5.7b \), the parameter \( \phi_{z_l t_l} \) that indicates how likely the word \( t_l \) is assigned to aspect \( k \) is calculated as:

\[ \phi_{z_l t_l | z_l = k} = \frac{n^{(w_l)}_{-l,k} + a}{n^{(z_l)}_{-l,k} + W a}, \]

where \( n^{(\cdot)}_{-l,k} \) is the total number of words assigned to the \( k \)-th aspect, which does not include the current one; \( n^{(t_l)}_{-l,k} \) is the total times of word \( t_l \) assigned to the \( k \)-th aspect; and \( a \) is a hyperparameter that determines how this multinomial distribution is smoothed. The parameter \( \beta_{z_l v_l | z_l = k} \) is calculated via:

\[ \beta_{z_l v_l | z_l = x, z_l = k} = \frac{n^{(t_l)}_{-l,x,k} + b}{n^{(z_l)}_{-l,x,k} + W b}, \]

where \( n^{(\cdot)}_{-l,x,k} \) is the total number of words assigned to aspect \( k \) and aspect rating \( x \); \( n^{(t_l)}_{-l,x,k} \) is the total times of word \( t_l \) assigned to aspect \( k \) and aspect rating \( x \); and \( b \) is a hyperparameter for smoothing the multinomial distribution.

For each reviewer, his/her cluster membership is updated according to

\[ \hat{c}_i = \arg\max_{c_i} \left[ \log P(w_i | c_i) + \log P(c_i | \pi_i) \right], \]

— 80 —
and the cluster-level aspect weight prior \((\mu_c, \Sigma_c)\) can be updated according to

\[
\hat{\mu}_k = \frac{1}{|U_c|} \sum_{i=1}^{U} \delta(c_i = c)w_i
\]

\[
\hat{\Sigma}_k = \frac{1}{|U_c|} \sum_{i=1}^{U} \left[(w_i - \hat{\mu}_c)(w_i - \hat{\mu}_c)^T\right],
\]

where \(U_c\) denotes the set of reviewers who belong to cluster \(c\).

2. Update parameters with fixed hidden variables

\[
(\hat{\theta}, \hat{\vartheta}, \hat{\pi}, \hat{w}) = \arg\max_{(\theta, \vartheta, \pi, w)} \sum_{r_{ij} \in R} L(\Phi; r_{ij}),
\]

so as to update the aspect distribution for product \(p_j\):

\[
\hat{\theta}_j = \arg\max_{\theta_j} \sum_{r_{ij} \in R} \log P(\hat{z}|\theta_j),
\]

and update the aspect-based ratings distribution for product \(p_j\) as

\[
\hat{\vartheta}_j = \arg\max_{\vartheta_j} \sum_{r_{ij} \in R} \log P(\hat{v}|\vartheta_j),
\]

update the aspect-based weights for reviewer \(u_i\) as

\[
\hat{w}_i = \arg\max_{w_i} \sum_{r_{ij} \in R} \log P(y_{ij}|\hat{v}, w_i) + \log P(w_i|\hat{c}_i).
\]

Algorithm 2 gives the pseudo-code of this model’s inference process. In addition, because our CARW model depends on topic modeling techniques to discover the aspects, we also use the seed words contained to guide the model learning. The seed words enable us to align the discovered aspects with the product attributes.
5.2.2 New Users’ Multi-Attribute Preferences Estimation

To address the partial preferences problem, in our previous works, we designed some methods to mine reviewers’ aspect-level opinions for new-user recommendation via predicting the missing weight or value preferences of a new buyer. As a new and potential approach to address this challenge, in this section, we attempt to derive user’s preferences over product attributes under the multi-attribute utility theory (MAUT). All products were then ranked according to their matching utilities with the buyer’s multi-attribute preference model. The challenging issue is then how to predict the buyer’s missing preferences on un-stated attributes.

Specifically, by integrating with the aspect-level weights and ratings from reviews, we target to find like-minded reviewers for the target new buyer and hence enrich the buyer’s...
preferences on all attributes. Formally, the derived aspect-level opinion (i.e., rating) can be classified as negative or positive; and map the opinion to the attribute’s static specification in the structured form (attribute, opinion, specification), for example, (price, positive, $200) which indicates that the reviewer expresses positive opinion on the product’s price that is $200. We can hence infer the reviewer’s preference on the corresponding attribute (e.g., the interval of price that reviewer prefers lies in the range containing $200). Thus, it is expected that the incorporation of product reviews can bring true user preferences so as to ideally augment the system’s recommendation accuracy for the new buyer.

In a typical scenario, with the goal of find an ideal product, a buyer ū would first provide preferences on subset of A attributes, denoted as Θū = {Θāk = (wāk, φāk)|1 ≤ k ≤ A}, where wāk is the weight, and φāk is the weight and value preference on attribute a_k. For example, if the buyer ū would like the price with less than $200, and associated weight is 3, we can represent the preference on attribute price as (3, “<$200”).

Multi-Attribute Preference Similarity

For the purpose of completing the buyer’s missing preferences over un-stated attributes, we need to be able to identify a group of like-minded reviewers who have similar multi-attribute preferences (including weight and value preferences) to the buyer ū based on his/her stated partial preferences Θū. To take into account both multi-attribute weight and value preferences, we formally define the similarity between buyer ū and reviewer u as:

$$sim(ū, u) = \lambda \ast sim_{weight}(ū, u) + (1 - \lambda) \ast sim_{value}(ū, u)$$  \hspace{1cm} (5.16)

where \(sim_{weight}(ū, u)\) and \(sim_{value}(ū, u)\) are the similarity measures based on the multi-attribute weight and value preferences respectively, and \(\lambda \in [0, 1]\) is the tradeoff parameter (\(\lambda\) is set as 0.5) between the weight preference similarity and value preference similarity.
CHAPTER 5. DERIVING NEW USERS’ MULTI-ATTRIBUTE PREFERENCE MODEL

Suppose that the stated weight preferences over subset of attributes from the current buyer $\hat{u}$ are $w_{\hat{u}} = \{w_{\hat{u}k} | 1 \leq k \leq A\}$, the weight preference similarity $sim_{weight}(\hat{u}, u)$ is defined as:

$$sim_{weight}(\hat{u}, u) = \frac{1}{1 + \sqrt{\sum_{w_{\hat{u}j} \in w_{\hat{u}}} (w_{\hat{u}j} - w_{uj})^2}}$$ (5.17)

For the value preference similarity $sim_{value}(\hat{u}, u)$, we first obtain the reviewer’s value preference $\phi_{uk}$ on attribute $a_k$ based on his opinion expressed on attribute $a_k$ in review using the same method explained in Section 4.2.2. Formally, given the current buyer $\hat{u}$’s stated value preferences $\phi_{\hat{u}} = \{\phi_{\hat{u}k} | 1 \leq k \leq A\}$ over subset of $A$ attributes, the value preference similarity $sim_{value}(\hat{u}, u)$ between the buyer $\hat{u}$ and reviewer $u$ is calculated as the cosine similarity between their value preferences, which is defined as:

$$sim_{value}(\hat{u}, u) = \sum_{\phi_{\hat{u}k} \in \phi_{\hat{u}}} \cos(\overrightarrow{\phi_{\hat{u}k}}, \overrightarrow{\phi_{uk}})$$ (5.18)

where $\overrightarrow{\phi_{\hat{u}k}}$ and $\overrightarrow{\phi_{uk}}$ are vector representations of buyer $\hat{u}$’s value preference $\phi_{\hat{u}k}$ and reviewer $u$’s value preferences $\phi_{uk}$ on attribute $a_k$, respectively (see Section 4.2.2).

5.2.3 Recommendation Generation

Given the buyer’s currently stated multi-attribute preferences, it should be time to identify a set of reviewers $K$ who have similarity preferences to the buyer based on the similarity measure defined in Eqn. 5.16. These reviewers’ preferences that the new buyer has not stated might help predict the new buyer’s preferences and locate recommendable products that s/he truly likes. Specifically, in the following, we proposed two recommendation approaches that the difference between them is the way to utilize the like-mined reviewers’ preferences in the recommendation procedure. In the first one, the products which are liked by most of the like-minded reviewers will be chosen as recommendations. In the
second one, the partial preferences $\Phi_{\hat{u}}$ of the new buyer $\hat{u}$ is enhanced by the preferences of like-minded reviewers. The products of higher utility values based on the predicted full preferences are chosen as recommendations.

**k-NN based Recommendation (k-NN-Rec)**

We implement this approach because it is a common way to integrate the similarity between user-reviewer into generating recommendations. In this method, with the set of like-mined reviewers $\mathcal{K}$ for the buyer $\hat{u}$, a prediction score is assigned to each product $p_j$ by following the basic collaborative filtering mechanism:

$$\text{ProductScore}(\hat{u}, p_j) = \frac{\sum_{u_i \in \mathcal{K}} \text{sim}(\hat{u}, u_i) \times y_{ij}}{\sum_{u_i \in \mathcal{K}} \text{sim}(\hat{u}, u_i)}$$

(5.19)

where $y_{ij}$ is the overall rating that reviewer $u_i$ gave to product $p_j$, and $|\mathcal{K}| = 500$ through experimental trials. Then, the top-$N$ products with higher scores are recommended to the buyer (in our experiment, we tested the algorithm’s performance when $N = 10, 20$).

**Preference-based Product Recommendation (PPR-Rec)**

In this approach, we also need to identify a set of like-minded reviewers based on preference similarity $\text{sim}(u, \hat{u})$. Then, the current buyer’s stated preferences are enriched by considering the like-minded reviewers’ preferences. Concretely, if the weight on attribute $a_k$ is not stated by the current buyer $\hat{u}$, it can be predicted by

$$\bar{w}_{uk} = \frac{1}{\sum_{u_i \in \mathcal{K}} \text{sim}(\hat{u}, u_i)} \sum_{u_i \in \mathcal{K}} \text{sim}(\hat{u}, u_i) \times w_{ik}$$

(5.20)

The new buyer’s preference $\bar{\phi}_{\hat{u}k}$ for one attribute $a_k$ can be either completed (if he did not specify it) or adjusted (if he already specified it) by taking like-minded reviewers’ value preferences into account via the the Eqn. 4.1. Based on the Multi-Attribute Utility
CHAPTER 5. DERIVING NEW USERS’ MULTI-ATTRIBUTE PREFERENCE MODEL

Theory as defined by Eqn. [1.1] in Section [1.2], the matching score of each product to the buyer \( \hat{u} \) is computed as:

\[
ProductScore(\hat{u}, p_j) = \sum_{k=1}^{A} w_{\hat{u}k} \times v_{\hat{u}k}(x_{jk})
\]

\[
= \sum_{(w_{\hat{u}k}, \phi_{\hat{u}k}) \in \Phi_{\hat{u}}} w_{\hat{u}k} \times \langle \phi_{\hat{u}k}, x_{jk} \rangle
\]

\[
= \sum_{(w_{\hat{u}k}, \phi_{\hat{u}k}) \notin \Phi_{\hat{u}}} \bar{w}_{\hat{u}k} \times \langle \phi_{\hat{u}k}, x_{jk} \rangle
\]

in which the buyer’s value function \( v_{\hat{u}k}(x_{jk}) \) on attribute \( a_k \) can be defined as:

\[
v_{\hat{u}k}(x_{jk}) = \begin{cases} 
\langle \phi_{\hat{u}k}, x_{jk} \rangle & \text{if } (w_{\hat{u}k}, \phi_{\hat{u}k}) \in \Phi_{\hat{u}} \\
\langle \phi_{\hat{u}k}, x_{jk} \rangle & \text{otherwise}
\end{cases}
\]

Still, top-\( N \) products with higher scores are recommended to the buyer.

5.3 Experiment

5.3.1 Experimental Setup and Dataset

Table 5.1: Descriptions of two datasets used in two recommender systems

<table>
<thead>
<tr>
<th></th>
<th>Digital camera</th>
<th>Laptop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total products</td>
<td>346</td>
<td>303</td>
</tr>
<tr>
<td>Total reviews</td>
<td>46,434</td>
<td>16,047</td>
</tr>
<tr>
<td>Avg. reviews per product</td>
<td>134.20</td>
<td>52.96</td>
</tr>
</tbody>
</table>

For the evaluation purpose, we have collected a test set of records from a user study, where users were instructed to state some preferences over attributes and provide some product recommendations. In our user study, we implemented two product recommenders (digital cameras and laptops), where the products’ information and their reviews were collected from Amazon.com. For the sake of extracting aspect-based opinions from reviews,
the products with less than 10 reviews were eliminated from our dataset. Finally, the digital camera recommender system is composed of 346 products described by 6 attributes, and the laptop recommender system is composed of 303 products characterized by 7 attributes. The details of these datasets are shown in Table 5.1. In particular, to align the aspects discovered from CARW model with the product attributes, a set of seed words on the digital camera reviews is shown in Table 5.2, and the seed words for the laptop reviews are shown in Table 5.3.

Table 5.2: Seed words for digital camera reviews used to guide CARW model learning

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Seed words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brand</td>
<td>sony, cannon, brand, camera, nikon</td>
</tr>
<tr>
<td>Price</td>
<td>price, value, cost, money, dollar, pay, payment, sale, deal</td>
</tr>
<tr>
<td>Screen size</td>
<td>screen, lcd, size</td>
</tr>
<tr>
<td>Effective pixels</td>
<td>resolution, pixel, megapixel, ccd</td>
</tr>
<tr>
<td>Optical zoom</td>
<td>zoom, range</td>
</tr>
<tr>
<td>Weight</td>
<td>weight, heavy, slight, body</td>
</tr>
</tbody>
</table>

Table 5.3: Seed words for laptop reviews used to guide CARW model learning

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Seed words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brand</td>
<td>toshiba, asus, quality, hp, acer, sony, brand</td>
</tr>
<tr>
<td>Price</td>
<td>price, value, cost, money, dollar, pay, payment, sale, deal</td>
</tr>
<tr>
<td>Processor speed</td>
<td>cpu, speed, performance, compute, boot, time, core, processor, i5</td>
</tr>
<tr>
<td>RAM</td>
<td>memory, load, ram</td>
</tr>
<tr>
<td>Hard drive</td>
<td>disk, capacity, io, storage, ssd, disc, drive</td>
</tr>
<tr>
<td>Screen size</td>
<td>screen, size, lcd, monitor, inch, touch, touchscreen</td>
</tr>
<tr>
<td>Weight</td>
<td>weight, heavy, slight</td>
</tr>
</tbody>
</table>

In our studies, we recruited participants through internal email list, advertisements in forum and crowdsourcing (i.e., Amazon Mechanical Turk). Finally, 295 real users’
interaction records were collected. Each record includes the real user’s initial preferences on product attributes and the product h/she selected as the target choice. Specifically, in the user study, the implemented recommender system enables users to express their preferences over different attributes via a preference statement interface. For example, as shown in Figure 5.3, it is the preference statement interface used in digital camera recommender system.

The experiment goal was to evaluate whether the user’s target choice could be located in the recommendation list when it is presented to her/him. For this goal, in our experiment, at a time, one user was randomly chosen to behave as a new buyer, and her/his initial preferences were taken as the new buyer’s stated preferences. We further randomly selected subsets of the user’s initial preferences to represent the new buyer’s preference completeness degrees (i.e., preferences over 2, 4 or 6 attributes, out of full 6 camera attributes; and preferences over 2, 4, 7 attributes, out of full 7 laptop attributes).

5.3.2 Compared Methods and Evaluation Metrics

In addition to the methods described in Section 5.2.3, we implemented three compared methods: one without the fusion of reviews, and the other two with the fusion of reviews in different ways.

1) Recommending without the fusion of reviews (Baseline)

It is purely based on the product’s attributes (i.e., static specifications) for ranking. More specifically, given the new user’s stated preferences \( \Phi_{\hat{u}} \), the matching score for each product \( p_j \) with attribute values \( x_{jk} \) is defined as:

\[
ProductScore(\hat{u}, p_j) = \sum_{(w_{\hat{u}k}, \phi_{\hat{u}k}) \in \Phi_{\hat{u}}} w_{\hat{u}k} \times \langle \phi_{\hat{u}k}, x_{jk} \rangle
\]  

(5.23)
Hence, its difference from Eqn. 5.21 is that it mainly relies on the new user’s stated preferences $\Phi_u$ solely to determine each product’s matching score.

2) Recommending with the fusion of reviews

The following two methods are both with their inputs as the predicted complete preferences by fusing reviewers’ preferences, but they vary in the way of utilizing these completed users’ preferences.

- **Review-fused weight preferences based product recommendation (RF-WP-Rec)**

  Given the new buyer’s current weight preferences $w_u$, in this approach, it aims at first identifying a set of like-mined reviewers $|K|$ who have similar weight preferences to the new buyer. The weight similarity between the new buyer and a re-
viewer is formally computed by Eqn. 5.17. Then, a prediction score is assigned to each product $p_j$ by using Eqn. 5.19, where the similarity $sim(\hat{u}, u)$ is replaced with $sim_{\text{weight}}(\hat{u}, u)$. Still, the top-$N$ products with higher scores are recommended to the new user.

- **Review-fused value preferences based product recommendation (RF-VP-Rec)**

Another approach that we implemented is using the same method which was first introduced in Section 4.2.3. Its major difference from our method proposed in this chapter is that it did not consider the new user’s and reviewers’ weight preferences.

**Evaluation Metrics** As each user only has one target choice in our user study, we decide to use Hit-Ratio and MRR (Mean Reciprocal Rank) as the metrics to evaluate our recommendation approaches in the experiment. As Eqn. 4.4 in Section 3.3.2, Hit-Ratio $H@N$ refers to the percent of successes that new users’ target choices appear in the top-$N$ recommendation lists. For MRR, it is a statistic measure (defined as Eqn. 3.24 in Section 3.3.2) for evaluating the ranking position of the target choice in the recommendation list.

**5.3.3 Results Analysis**

Tables 5.4 & 5.5 show the comparison results in terms of $H@10$, $H@20$ and MRR metrics on digital camera and laptop recommender systems respectively. First of all, it can be seen that RF-WP-Rec, RF-VP-Rec, k-NN-Rec and PPR-Rec clearly outperform the baseline in terms of all metrics. It hence suggests that the recovered reviewers’ preferences have potential to be used to predict the missing preferences of new users. Moreover, we can also observe that the outperforming of these methods is more obvious when the new buyer’s
preferences are less complete (e.g., over 2 or 4 attributes for digital cameras), inferring that the un-stated preferences can be predicted by relating the user to like-minded reviewers.

Moreover, we can observe that RF-WP-Rec provides better recommendations than that by RF-VP-Rec, when the preference completeness size is small. For example, in the camera recommender system, the Hit-Ratio achieved by RF-WP-Rec when $N = 10$, which is up to 3.7% higher than that by RF-VP-Rec in the condition of stated preferences on 2 attributes by new buyers. This observation implies that, given the stated preferences by the buyer, his weight preferences by fusing like-mined reviewers can provide more valuable information for recommending products than his enriched value preferences, especially when the buyer’s preferences are less complete. We can also observe that the proposed methods, k-NN-Rec and PPR-Rec, outperform the other compared methods (including RF-WP-Rec and RF-VP-Rec) in all cases. Hence, we can conclude that the MAUT-based preference model is more likely to represent the user’s true preferences than the weight or value preferences individually.

Furthermore, PPR-Rec is more accurate than k-NN-Rec in both systems. For example, when the buyer’s preference is stated over 2 camera attributes (i.e., the preference size is 2), the Hit-Ratio achieved by PPR-Rec is 0.386 when $N$ is 10, which is 64.3% higher than that by k-NN-Rec. The MRR value of PPR-Rec is also higher than that by k-NN-Rec. This phenomenon might be caused by that the k-NN-Rec prefers to recommend popular products which may not match the user’s preferences. Hence, we can say that PPR-Rec not only increases the possibility of including the user’s target choice in the recommendation list, but also ranking the target choice in top positions in the list.
Table 5.4: Comparison of algorithms w.r.t. *Hit Ratio* and *MRR* with varied preference sizes for digital camera recommender system (the maximal preference size is 6)

<table>
<thead>
<tr>
<th>Preference size</th>
<th>Method</th>
<th>Evaluation metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$H@10$</td>
</tr>
<tr>
<td>2 attributes</td>
<td>Baseline</td>
<td>0.035</td>
</tr>
<tr>
<td></td>
<td>RF-WP-Rec</td>
<td>0.122</td>
</tr>
<tr>
<td></td>
<td>RF-VP-Rec</td>
<td>0.101</td>
</tr>
<tr>
<td></td>
<td>k-NN-Rec</td>
<td>0.235</td>
</tr>
<tr>
<td></td>
<td>PPR-Rec</td>
<td><strong>0.386</strong></td>
</tr>
<tr>
<td>4 attributes</td>
<td>Baseline</td>
<td>0.313</td>
</tr>
<tr>
<td></td>
<td>RF-WP-Rec</td>
<td>0.328</td>
</tr>
<tr>
<td></td>
<td>RF-VP-Rec</td>
<td>0.316</td>
</tr>
<tr>
<td></td>
<td>k-NN-Rec</td>
<td>0.458</td>
</tr>
<tr>
<td></td>
<td>PPR-Rec</td>
<td><strong>0.541</strong></td>
</tr>
<tr>
<td>6 attributes</td>
<td>Baseline</td>
<td>0.623</td>
</tr>
<tr>
<td></td>
<td>RF-WP-Rec</td>
<td>0.315</td>
</tr>
<tr>
<td></td>
<td>RF-VP-Rec</td>
<td>0.601</td>
</tr>
<tr>
<td></td>
<td>k-NN-Rec</td>
<td>0.467</td>
</tr>
<tr>
<td></td>
<td>PPR-Rec</td>
<td><strong>0.624</strong></td>
</tr>
</tbody>
</table>
Table 5.5: Comparison of algorithms w.r.t. *Hit Ratio* and *MRR* with varied preference sizes for laptop recommender system (the maximal preference size is 7)

<table>
<thead>
<tr>
<th>Preference size</th>
<th>Method</th>
<th>Evaluation metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$H@10$</td>
</tr>
<tr>
<td>2 attributes</td>
<td>Baseline</td>
<td>0.035</td>
</tr>
<tr>
<td></td>
<td>RF-WP-Rec</td>
<td>0.126</td>
</tr>
<tr>
<td></td>
<td>RF-VP-Rec</td>
<td>0.121</td>
</tr>
<tr>
<td></td>
<td>k-NN-Rec</td>
<td>0.132</td>
</tr>
<tr>
<td></td>
<td>PPR-Rec</td>
<td><strong>0.455</strong></td>
</tr>
<tr>
<td>4 attributes</td>
<td>Baseline</td>
<td>0.367</td>
</tr>
<tr>
<td></td>
<td>RF-WP-Rec</td>
<td>0.228</td>
</tr>
<tr>
<td></td>
<td>RF-VP-Rec</td>
<td>0.210</td>
</tr>
<tr>
<td></td>
<td>k-NN-Rec</td>
<td>0.353</td>
</tr>
<tr>
<td></td>
<td>PPR-Rec</td>
<td><strong>0.488</strong></td>
</tr>
<tr>
<td>7 attributes</td>
<td>Baseline</td>
<td><strong>0.657</strong></td>
</tr>
<tr>
<td></td>
<td>RF-WP-Rec</td>
<td>0.285</td>
</tr>
<tr>
<td></td>
<td>RF-VP-Rec</td>
<td>0.554</td>
</tr>
<tr>
<td></td>
<td>k-NN-Rec</td>
<td>0.501</td>
</tr>
<tr>
<td></td>
<td>PPR-Rec</td>
<td>0.656</td>
</tr>
</tbody>
</table>
5.4 Summary

In this chapter, we first proposed a unified CARW model that can simultaneously 1) identify the aspects mentioned in reviews, 2) infer the aspect-based ratings based on the sentiments expressed on aspects, and 3) estimate the aspect-based weights placed on aspects by a reviewer. The three tasks are addressed in an unsupervised manner, so that the CARW model can be feasibly applied across different domains by minimizing the training effort. Then, with the aspect-based ratings and weights resulted from CARW model, we then proposed a method to predict new users’ un-stated preferences by fusing the like-mined reviewers’ preferences. In the recommendation procedure, we proposed two recommendation approaches, k-NN-Rec and PPR-Rec. The experiment results show the outperforming accuracy of our review-based methods, especially PPR-Rec approach, in the condition that the new user’s preferences are less complete. This finding hence suggests that RRP-Rec can be more effective to support new buyers, given that their preferences will be likely incomplete when facing high-risk products.
Chapter 6

Conclusion

The main motivation of this thesis was to develop review-based recommendation framework for aiding new buyers’ decision making in high-risk product domains. Specifically, the task of recommending high-risk products is generally formalized as solving a *multi-attribute decision problem*, given that users’ preferences are constructed on multiple attributes. In order to simplify the complexity of user preference structures and elicitation processes, we have modeled user preferences based on the Multi-Attribute Utility Theory (MAUT). Facing such a new-user recommendation problem, related systems ask a buyer to indicate some preferences on certain attributes of the product, such as the camera’s brand, price, resolution, etc. The challenging issue is then how to predict the buyer’s missing preferences on un-stated attributes, which is the *partial preferences* problem. To address this problem, we designed and implemented several methods to recover reviewers’ product preferences (including weight preferences, value preferences and both) from their written reviews, and exploited such preference data to enhance the recommendation for new users. We also conducted a series of experiments to measure our approaches in terms of recommendation accuracy.
6.1 Contributions

Estimating Multi-Feature Weight Preferences from User Reviews

We particularly recover the reviewer’s multi-feature weight preferences. Then, these recovered preferences are leveraged to estimate and supplement the new buyer’s preference with which the product recommendation is produced. 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. A hybrid method that combines both reviewer-level and cluster-level preferences was introduced and experimentally compared to related 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.

Estimating Multi-Attribute Value Preferences from User Reviews

We also attempted to estimate reviewers’ attribute value preference by matching their review opinions to the corresponding attributes’ static specifications. The completed preferences of a new user are then used to match the product profiles, by which the products with highest matching scores are recommended to the target user. Experimental results show that our solution can provide more accurate personalized recommendations than several baseline methods.

Deriving New Users’ Multi-Attribute Preference Model

To model user preferences based on Multi-Attribute Utility Theory (MAUT), we need to obtain the buyer’s two types of preferences: 1) weight (i.e., the relative importance) placed
on the attribute; 2) value preference that defines the buyer’s preference over different values of the attributes. Hence, we also established the buyer’s MAUT preference model by fusing the buyer’s partial preferences and like-minded reviewers’ preferences. To improve the estimation accuracy of the reviewers’ preferences, we particularly proposed a unified model for address three inter-connected tasks: 1) aspect identification, 2) aspect-level rating estimation, and 3) aspect-level weight estimation. In this unified model, the three tasks can be improved simultaneously by processing them together. The experiment on data collected from a real-world user study shows that the buyer’s partial preferences can be enriched by fusing the reviewers’ preferences so as to improve the quality of high-risk product recommendations for “new user”.

6.2 Limitations

The limitations of the research in this thesis are discussed as below:

- **The preferential relationship between attributes is assumed to be independent.**
  All of our proposed solutions have been under the assumption that the user’s preference for one attribute is preferential independent of any of the other attributes. With this assumption, we model the user’s preference in an additive form. The advantage is its simplicity, but it can be too restrictive and not realistic.

- **The aspect-level opinions are used only to facilitate product recommendations.**
  To date, all of our studies have focused on the utilization of aspect-level opinions expressed in reviews to enhance the recommendations. However, some other types of review elements would contain valuable information. 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 product 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 products’ similarity against their relative sentiment improvement (Dong et al., 2013a). Therefore, we believe that more combinations could be explored.

6.3 Future Work

In this section, we highlight several directions for future research.

- **To perform a pre-process to improve the estimation of weight preferences.** The weight preferences are estimated based on the relationship between the overall rating and the opinion values expressed in review. So we need to remove reviews whose rating are conflicting with the sentiment value of the review content.

- **To relax the strong assumption that the preferential relationship between attributes is independent.** To be specific, the user’s acceptance of one attribute may depend on the product’s satisfaction on his preference for other attributes. Such relationship is indeed difficult to acquire from questionnaire approach, so we believe product reviews might be of particular merit to help distill it.

- **To combine various types of review information.** We believe that the combination of various review information could be explored. In particular, as feature
opinions naturally reflect a reviewer’s multi-faceted criteria, it should be beneficial to combine them with other elements, such as review emotions to detect users’ emotion-dependent feature preferences, or comparative opinions (which reveal the competing relation between two (or more) products. For example, the review sentence, “camera X has longer battery life than camera Y”, reveals that the reviewer prefer camera X in term of battery life with regard to camera Y) for constructing a more personalized product comparison graph.

- **To produce review-based explanations beyond recommendations.** 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. 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.

- **To validate our proposed methods’ practical benefits to real users.** Actually, instead of experiments that use evaluation metrics such as Hit Ratio or MRR to determine an algorithm’s recommendation 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 products 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.
6.4 Summary

In conclusion, this thesis in depth studied how to leverage product reviews into improving recommendation for the new buyer in high-risk product domains. With the objective of developing more effective recommender systems for high-risk products in e-commerce, in our work, we have exerted to derive reviewers’ preferences from the textual reviews they posted. Then, these recovered preferences are leveraged to estimate and supplement the new buyer’s preference with which the product recommendation is produced. Particularly, it was shown from our experimental studies that the incorporation of review information can significantly enhance the recommendation accuracy, relative to those without considering reviews. As the practical implication, our proposed solutions can be usefully plugged into an online system to be adopted in real-e-commerce sites.


erences and value trade-offs. *Systems, Man and Cybernetics, IEEE Transactions on*
1979;9(7):403.


International Conference on Knowledge Discovery and Data Mining, KDD ’09; New York, NY, USA: ACM; 2009:1195–1204.


ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD’04; New York, NY, USA: ACM; 2004b:168–177.


Appendix A

Reprint Reuse Permissions of Published Articles

This chapter contains the bibliographic data and reuse license of the publications, which have been prepared during the course of this thesis.

A.1 Preference-based clustering reviews for augmenting e-commerce recommendation

Li Chen and Feng Wang

Published in: Knowledge-Based Systems, Volume 50, September 2013, Pages 4459.
DOI: doi:10.1016/j.knosys.2013.05.006
APPENDIX A. REPRINT REUSE PERMISSIONS OF PUBLISHED ARTICLES

INTRODUCTION

1. The publisher for this copyrighted material is Elsevier. By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the Billing and Payment terms and conditions).
established by Copyright Clearance Center, Inc. ("CCC"), at the time that you opened your
Rightslink account and that are available at any time at http://myaccount.copyright.com).

2. Elsevier hereby grants you permission to reproduce the aforementioned material subject to
the terms and conditions indicated.

3. Acknowledgement: If any part of the material to be used (for example, figures) has
appeared in our publication with credit or acknowledgement to another source, permission
must also be sought from that source. If such permission is not obtained then that material
may not be included in your publication/copies. Suitable acknowledgement to the source
must be made, either as a footnote or in a reference list at the end of your publication, as
follows:
"Reprinted from Publication title, Vol /edition number, Author(s), Title of article / title of
chapter, Pages No., Copyright (Year), with permission from Elsevier [OR APPLICABLE
SOCIETY COPYRIGHT OWNER]." Also Lancet special credit - "Reprinted from The
Lancet, Vol. number, Author(s), Title of article, Pages No., Copyright (Year), with
permission from Elsevier."

4. Reproduction of this material is confined to the purpose and/or media for which
permission is hereby given.

5. Altering/Modifying Material: Not Permitted. However figures and illustrations may be
altered/adapted minimally to serve your work. Any other abbreviations, additions, deletions
and/or any other alterations shall be made only with prior written authorization of Elsevier
Ltd. (Please contact Elsevier at permissions@elsevier.com)

6. If the permission fee for the requested use of our material is waived in this instance,
please be advised that your future requests for Elsevier materials may attract a fee.

7. Reservation of Rights: Publisher reserves all rights not specifically granted in the
combination of (i) the license details provided by you and accepted in the course of this
licensing transaction, (ii) these terms and conditions and (iii) CCC's Billing and Payment
terms and conditions.

8. License Contingent Upon Payment: While you may exercise the rights licensed
immediately upon issuance of the license at the end of the licensing process for the
transaction, provided that you have disclosed complete and accurate details of your proposed
use, no license is finally effective unless and until full payment is received from you (either
by publisher or by CCC) as provided in CCC's Billing and Payment terms and conditions. If
full payment is not received on a timely basis, then any license preliminarily granted shall be
deemed automatically revoked and shall be void as if never granted. Further, in the event
that you breach any of these terms and conditions or any of CCC's Billing and Payment
terms and conditions, the license is automatically revoked and shall be void as if never
granted. Use of materials as described in a revoked license, as well as any use of the
materials beyond the scope of an unrevoked license, may constitute copyright infringement
and publisher reserves the right to take any and all action to protect its copyright in the
materials.

9. Warranties: Publisher makes no representations or warranties with respect to the licensed
material.

10. Indemnity: You hereby indemnify and agree to hold harmless publisher and CCC, and
their respective officers, directors, employees and agents, from and against any and all
claims arising out of your use of the licensed material other than as specifically authorized
pursuant to this license.

11. No Transfer of License: This license is personal to you and may not be sublicensed,
assigned, or transferred by you to any other person without publisher's written permission.

12. No Amendment Except in Writing: This license may not be amended except in a writing
signed by both parties (or, in the case of publisher, by CCC on publisher's behalf).

13. Objection to Contrary Terms: Publisher hereby objects to any terms contained in any
purchase order, acknowledgment, check endorsement or other writing prepared by you,
which terms are inconsistent with these terms and conditions or CCC's Billing and Payment
terms and conditions. These terms and conditions, together with CCC's Billing and Payment
terms and conditions (which are incorporated herein), comprise the entire agreement
between you and publisher (and CCC) concerning this licensing transaction. In the event of
any conflict between your obligations established by these terms and conditions and those

APPENDIX A. REPRINT REUSE PERMISSIONS OF PUBLISHED ARTICLES

https://s100.copyright.com/AppDispatchServlet
established by CCC's Billing and Payment terms and conditions, these terms and conditions shall control.

14. Revocation: Elsevier or Copyright Clearance Center may deny the permissions described in this License at their sole discretion, for any reason or no reason, with a full refund payable to you. Notice of such denial will be made using the contact information provided by you. Failure to receive such notice will not alter or invalidate the denial. In no event will Elsevier or Copyright Clearance Center be responsible or liable for any costs, expenses or damage incurred by you as a result of a denial of your permission request, other than a refund of the amount(s) paid by you to Elsevier and/or Copyright Clearance Center for denied permissions.

LIMITED LICENSE

The following terms and conditions apply only to specific license types:

15. **Translation**: This permission is granted for non-exclusive world English rights only unless your license was granted for translation rights. If you licensed translation rights you may only translate this content into the languages you requested. A professional translator must perform all translations and reproduce the content word for word preserving the integrity of the article.

16. **Posting licensed content on any Website**: The following terms and conditions apply as follows: Licensing material from an Elsevier journal: All content posted to the web site must maintain the copyright information line on the bottom of each image; A hyper-text must be included to the Homepage of the journal from which you are licensing at http://www.sciencedirect.com/science/journal/yyyyy or the Elsevier homepage for books at http://www.elsevier.com; Central Storage: This license does not include permission for a scanned version of the material to be stored in a central repository such as that provided by Heron/XanEdu.

Licensing material from an Elsevier book: A hyper-text link must be included to the Elsevier homepage at http://www.elsevier.com. All content posted to the web site must maintain the copyright information line on the bottom of each image.

**Posting licensed content on Electronic reserve**: In addition to the above the following clauses are applicable: The web site must be password-protected and made available only to bona fide students registered on a relevant course. This permission is granted for 1 year only. You may obtain a new license for future website posting.

17. **For journal authors**: the following clauses are applicable in addition to the above:

**Preprints**: A preprint is an author's own write-up of research results and analysis, it has not been peer-reviewed, nor has it had any other value added to it by a publisher (such as formatting, copyright, technical enhancement etc.). Authors can share their preprints anywhere at any time. Preprints should not be added to or enhanced in any way in order to appear more like, or to substitute for, the final versions of articles however authors can update their preprints on arXiv or RePEc with their Accepted Author Manuscript (see below).

If accepted for publication, we encourage authors to link from the preprint to their formal publication via its DOI. Millions of researchers have access to the formal publications on ScienceDirect, and so links will help users to find, access, cite and use the best available version. Please note that Cell Press, The Lancet and some society-owned have different preprint policies. Information on these policies is available on the journal homepage.

**Accepted Author Manuscripts**: An accepted author manuscript is the manuscript of an article that has been accepted for publication and which typically includes author-incorporated changes suggested during submission, peer review and editor-author communications. Authors can share their accepted author manuscript:

- immediately
  - via their non-commercial person homepage or blog
  - by updating a preprint in arXiv or RePEc with the accepted manuscript
  - via their research institute or institutional repository for internal institutional uses or as part of an invitation-only research collaboration work-group

— 116 —
APPENDIX A. REPRINT REUSE PERMISSIONS OF PUBLISHED ARTICLES

- directly by providing copies to their students or to research collaborators for their personal use
- for private scholarly sharing as part of an invitation-only work group on commercial sites with which Elsevier has an agreement
- after the embargo period
- via non-commercial hosting platforms such as their institutional repository
- via commercial sites with which Elsevier has an agreement

In all cases accepted manuscripts should:
- link to the formal publication via its DOI
- bear a CC-BY-NC-ND license - this is easy to do
- if aggregated with other manuscripts, for example in a repository or other site, be shared in alignment with our hosting policy not be added to or enhanced in any way to appear more like, or to substitute for, the published journal article.

Published journal article (JPA): A published journal article (PJA) is the definitive final record of published research that appears or will appear in the journal and embodies all value-adding publishing activities including peer review co-ordination, copy-editing, formatting, (if relevant) pagination and online enrichment.

Policies for sharing publishing journal articles differ for subscription and gold open access articles:

Subscription Articles: If you are an author, please share a link to your article rather than the full-text. Millions of researchers have access to the formal publications on ScienceDirect, and so links will help your users to find, access, cite, and use the best available version.

Theses and dissertations which contain embedded PJAs as part of the formal submission can be posted publicly by the awarding institution with DOI links back to the formal publications on ScienceDirect.

If you are affiliated with a library that subscribes to ScienceDirect you have additional private sharing rights for others' research accessed under that agreement. This includes use for classroom teaching and internal training at the institution (including use in course packs and courseware programs), and inclusion of the article for grant funding purposes.

Gold Open Access Articles: May be shared according to the author-selected end-user license and should contain a CrossMark logo, the end user license, and a DOI link to the formal publication on ScienceDirect.

Please refer to Elsevier's posting policy for further information.

18. For book authors the following clauses are applicable in addition to the above:

Authors are permitted to place a brief summary of their work online only. You are not allowed to download and post the published electronic version of your chapter, nor may you scan the printed edition to create an electronic version.

Posting to a repository: Authors are permitted to post a summary of their chapter only in their institution’s repository.

19. Thesis/Dissertation: If your license is for use in a thesis/dissertation your thesis may be submitted to your institution in either print or electronic form. Should your thesis be published commercially, please reapply for permission. These requirements include permission for the Library and Archives of Canada to supply single copies, on demand, of the complete thesis and include permission for Proquest/UMI to supply single copies, on demand, of the complete thesis. Should your thesis be published commercially, please reapply for permission. Theses and dissertations which contain embedded PJAs as part of the formal submission can be posted publicly by the awarding institution with DOI links back to the formal publications on ScienceDirect.

Elsevier Open Access Terms and Conditions

You can publish open access with Elsevier in hundreds of open access journals or in nearly 2000 established subscription journals that support open access publishing. Permitted third party re-use of these open access articles is defined by the author’s choice of Creative Commons user license. See our open access license policy for more information.

Terms & Conditions applicable to all Open Access articles published with Elsevier:

Any reuse of the article must not represent the author as endorsing the adaptation of the
APPENDIX A. REPRINT REUSE PERMISSIONS OF PUBLISHED ARTICLES

article nor should the article be modified in such a way as to damage the author's honour or reputation. If any changes have been made, such changes must be clearly indicated. The author(s) must be appropriately credited and we ask that you include the end user license and a DOI link to the formal publication on ScienceDirect. If any part of the material to be used (for example, figures) has appeared in our publication with credit or acknowledgement to another source it is the responsibility of the user to ensure their reuse complies with the terms and conditions determined by the rights holder.

Additional Terms & Conditions applicable to each Creative Commons user license:

**CC BY:** The CC-BY license allows users to copy, to create extracts, abstracts and new works from the Article, to alter and revise the Article and to make commercial use of the Article (including reuse and/or resale of the Article by commercial entities), provided the user gives appropriate credit (with a link to the formal publication through the relevant DOI), provides a link to the license, indicates if changes were made and the licensor is not represented as endorsing the use made of the work. The full details of the license are available at [http://creativecommons.org/licenses/by/4.0](http://creativecommons.org/licenses/by/4.0).

**CC BY NC SA:** The CC BY-NC-SA license allows users to copy, to create extracts, abstracts and new works from the Article, to alter and revise the Article, provided this is not done for commercial purposes, and that the user gives appropriate credit (with a link to the formal publication through the relevant DOI), provides a link to the license, indicates if changes were made and the licensor is not represented as endorsing the use made of the work. Further, any new works must be made available on the same conditions. The full details of the license are available at [http://creativecommons.org/licenses/by-nc-sa/4.0](http://creativecommons.org/licenses/by-nc-sa/4.0).

**CC BY NC ND:** The CC BY-NC-ND license allows users to copy and distribute the Article, provided this is not done for commercial purposes and further does not permit distribution of the Article if it is changed or edited in any way, and provided the user gives appropriate credit (with a link to the formal publication through the relevant DOI), provides a link to the license, and that the licensor is not represented as endorsing the use made of the work. The full details of the license are available at [http://creativecommons.org/licenses/by-nc-nd/4.0](http://creativecommons.org/licenses/by-nc-nd/4.0).

Any commercial reuse of Open Access articles published with a CC BY NC SA or CC BY NC ND license requires permission from Elsevier and will be subject to a fee. Commercial reuse includes:

- Associating advertising with the full text of the Article
- Charging fees for document delivery or access
- Article aggregation
- Systematic distribution via e-mail lists or share buttons

Posting or linking by commercial companies for use by customers of those companies.

20. Other Conditions:

v1.8

Questions? [customercare@copyright.com](mailto:customercare@copyright.com) or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.
A.2 Recommendation for New Users with Partial Preferences by Integrating Product Reviews with Static Specifications

Feng Wang, Weike Pan and Li Chen

DOI: 10.1007/978-3-642-38844-6_24
Hyperlink: http://link.springer.com/chapter/10.1007%2F978-3-642-38844-6_24
This Agreement between Feng Wang ("You") and Springer ("Springer") consists of your license details and the terms and conditions provided by Springer and Copyright Clearance Center.

License Number: 3922410268136  
License date: Aug 05, 2016  
Licensed Content Publisher: Springer  
Licensed Content Publication: Springer eBook  
License Content Title: Recommendation for New Users with Partial Preferences by Integrating Product Reviews with Static Specifications  
License Content Author: Feng Wang  
License Content Date: Jan 1, 2013  
Type of Use: Thesis/Dissertation  
Portion: Full text  
Number of copies: 1  
Author of this Springer article: Yes and you are a contributor of the new work  
Order reference number:  
Title of your thesis / dissertation: Inferring Users' Multi-Attribute Preferences from Their Reviews for Augmenting Recommender Systems in e-Commerce  
Expected completion date: Aug 2016  
Estimated size (pages): 120

Requestor Location: Feng Wang  
RRS734, Sir Run Run Shaw Building  
Hong Kong Baptist University  
224 Waterloo Road, Kowloon Tong Kowloon  
Hong Kong, 00000  
Attr: Feng Wang

Billing Type: Invoice  
Billing Address: Feng Wang  
RRS734, Sir Run Run Shaw Building  
Hong Kong Baptist University  
224 Waterloo Road, Kowloon Tong Kowloon  
Hong Kong, Hong Kong 00000  
Attr: Feng Wang

Total: 0.00 USD

Terms and Conditions

Introduction

The publisher for this copyrighted material is Springer. By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the Billing and Payment terms and conditions established by Copyright Clearance Center, Inc. ("CCC"), at the time that you opened your Rightslink account and that are available at any time at http://myaccount.copyright.com).

Limited License

With reference to your request to reuse material on which Springer controls the copyright, permission is granted for the use indicated in your enquiry under the following conditions:

- Licenses are for one-time use only with a maximum distribution equal to the number stated in your request.
- Springer material represents original material which does not carry references to other sources. If the material in question appears with a credit to another source, this permission is not valid and authorization has to be obtained from the original copyright holder.
- This permission

--- 120 ---
APPENDIX A. REPRINT REUSE PERMISSIONS OF PUBLISHED ARTICLES

8/8/2016 RightsLink - Your Account

This License only pertains to use in electronic form provided it is password protected, on intranet, or CD-Rom/DVD or E-book/E-journal. It may not be republished in electronic open access.

• Print: This License excludes use in electronic form.
• Electronic: This License only pertains to use in electronic form provided it is password protected, on intranet, or CD-Rom/DVD or E-book/E-journal. It may not be republished in electronic open access.

For any electronic use not mentioned, please contact Springer at permissions.springer@spi-global.com.

- Although Springer controls the copyright to the material and is entitled to negotiate on rights, this license is only valid subject to courtesy information to the author (address is given in the article/chapter).
- If you are an STM Signatory or your work will be published by an STM Signatory and you are requesting to reuse figures/tables/illustrations or single text extracts, permission is granted according to STM Permissions Guidelines:
http://www.stm-assoc.org/permissions-guidelines/

For any electronic use not mentioned in the Guidelines, please contact Springer at permissions.springer@spi-global.com. If you request to reuse more content than stipulated in the STM Permissions Guidelines, you will be charged a permission fee for the excess content.

Permission is valid upon payment of the fee as indicated in the licensing process. If permission is granted for free charge on this occasion, that does not prejudice any rights we might have to charge for reproduction of our copyrighted material in the future.

- If your request is for reuse in a Thesis, permission is granted free of charge under the following conditions:
This license is valid for one-time use only for the purpose of defending your thesis and with a maximum of 100 extra copies in paper. If the thesis is going to be published, permission needs to be reobtained.
- Includes use in an electronic form, provided it is an author-created version of the thesis on his/her own website and his/her university’s repository, including UMI (according to the definition on the Sherpa website: http://www.sherpa.ac.uk/romeo);
- is subject to courtesy information to the co-author or corresponding author.

Geographic Rights: Scope
Licenses may be exercised anywhere in the world.

Altering/Modifying Material: Not Permitted
Figures, tables, and illustrations may be altered minimally to serve your work. You may not alter or modify text in any manner.

Abbreviations, additions, deletions and/or any other alterations shall be made only with prior written authorization of the author(s).

Reservation of Rights
Springer reserves all rights not specifically granted in the combination of (i) the license details provided by you and accepted in the course of this licensing transaction and (ii) these terms and conditions and (iii) CCC’s Billing and Payment terms and conditions.

License Contingent on Payment
While you may exercise the rights licensed immediately upon issuance of the license at the end of the licensing process for the transaction, provided that you have disclosed complete and accurate details of your proposed use, no license is finally effective unless and until full payment is received from you (either by Springer or by CCC) as provided in CCC’s Billing and Payment terms and conditions. If full payment is not received by the date due, then any license preliminarily granted shall be deemed automatically revoked and shall be void as if never granted. Further, in the event that you breach any of these terms and conditions or any of CCC’s Billing and Payment terms and conditions, the license is automatically revoked and shall be void as if never granted. Use of materials as described in a revoked license, as well as any use of the materials beyond the scope of an unrevoked license, may constitute copyright infringement and Springer reserves the right to take any and all action to protect its copyright in the materials.

Copyright Notice: Disclaimer
You must include the following copyright and permission notice in connection with any reproduction of the licensed material:
“Springer book/journal title, chapter/article title, volume, year of publication, page, name(s) of author(s), (original copyright notice as given in the publication in which the material was originally published) “With permission of Springer”

In case of use of a graph or illustration, the caption of the graph or illustration must be included, as it is indicated in the original publication.

Warranties: None
Springer makes no representations or warranties with respect to the licensed material and adopts on its own behalf the limitations and disclaimers established by CCC on its behalf in its Billing and Payment terms and conditions for this licensing transaction.

Indemnity
You hereby indemnify and agree to hold harmless Springer and CCC, and their respective officers, directors, employees and agents, from and against any and all claims arising out of your use of the licensed material other than as specifically authorized pursuant to this license.

No Transfer of License
This license is personal to you and may not be sublicensed, assigned, or transferred by you without Springer’s written permission.

No Amendment Except in Writing
This license may not be amended except in a writing signed by both parties (or, in the case of Springer, by CCC on Springer's behalf).

Objection to Contrary Terms

Springer hereby objects to any terms contained in any purchase order, acknowledgment, check endorsement or other writing prepared by you, which terms are inconsistent with these terms and conditions or CCC's Billing and Payment terms and conditions. These terms and conditions, together with CCC's Billing and Payment terms and conditions (which are incorporated herein), comprise the entire agreement between you and Springer (and CCC) concerning this licensing transaction. In the event of any conflict between your obligations established by these terms and conditions and those established by CCC's Billing and Payment terms and conditions, these terms and conditions shall control.

Jurisdiction

All disputes that may arise in connection with this present License, or the breach thereof, shall be settled exclusively by arbitration, to be held in the Federal Republic of Germany, in accordance with German law.

Other conditions:

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.
A.3 Recommending Inexperienced Products via Learning from Consumer Reviews

Feng Wang and Li Chen

DOI: 10.1109/WI-IAT.2012.209
Hyperlink: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6511947
APPENDIX A. REPRINT REUSE PERMISSIONS OF PUBLISHED ARTICLES

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author’s approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author’s university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity’s name goes here]’s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Copyright © 2016 Copyright Clearance Center, Inc. All Rights Reserved. Privacy statement. Terms and Conditions. Comments? We would like to hear from you. E-mail us at customerservice@copyright.com.

https://s100.copyright.com/AppDispatchServlet#formTop

— 124 —
CURRICULUM VITAE

Academic qualification of the thesis author, Mr. WANG Feng:

- Received the degree of Bachelor of Software Engineering from Shandong University, June 2009.

September 2016