Optimization and Explanation of Recommenders to Increase the Causal Effect of Recommendations

March 2020

Masahiro Sato

Optimization and Explanation of Recommenders to Increase the Causal Effect of Recommendations

Graduate School of Systems and Information Engineering

University of Tsukuba

March 2020

Masahiro Sato

Acknowledgement

Firstly, I would like to express my sincere gratitude to my advisor Associate Professor Hajime Nobuhara for the continuous support of my Ph.D. study. I learned a lot from him, including how to present research effectively and how to operate a research group efficiently. I truly thank his valuable time for advising me during his especially busy year with double duty for the university and the government.

Besides my advisor, I would like to thank the rest of my thesis committee: Professor Hiroki Koga, Assistant Professor Takeshi Shibuya, Senior Researcher Toshihiro Kamishima, and Assistant Professor Shin Kawai, for their insightful comments and questions. I am grateful to Assistant Professor Shin Kawai also for his various support for my college life.

I would like to acknowledge Professor Yoshinori Hijikata for coaching me before I entered the Ph.D. course. Valuable discussion with him inspired me to various research directions.

Most researches in this thesis were conducted while I worked at the research team of information recommendation in Fuji Xerox. I would like to thank my current and previous bosses: Tomoko Ohkuma, Takeshi Onishi, Hiroshi Masuichi, and Takashi Sonoda, for providing a great work environment for research. I am also truly thankful for all the colleagues of the team so far: Hidetaka Izumo, Koki Nagatani, Qian Zhang, Budrul Ahsan, Janmajay Singh, Sho Takemori, Raghava Krishnan, and Ryo Shimura. Collaboration with them was essential to complete my researches.

Finally, I would like to send special thanks to my wife, Sakiko, for her patience and encouragement. Life with her gives me refreshments and new perspectives.

Abstract

Recommender systems have been utilized in various online services. Recommendations help users of these services to find interesting items. Recommender systems are also beneficial for businesses since they can improve their objectives, such as sales volume, and users' engagement. For the success of a business, a crucial goal of the recommender systems is to increase positive user actions, such as clicks and purchases. The increase in user actions leads to more sales and engagement.

User actions might be attributed to recommendations or other causes. Understanding and increasing the causal effect of recommendations is essential for better designing of recommenders. However, most previously conducted studies focused on the accuracy of recommendations, not on their causal effect. Particularly, there are three issues for increasing the causal effect of recommendations. The first is the difference in user behaviors with and without recommendations. Most of the conventional recommendations' studies disregarded the influence of recommendation when they train and evaluate models. The second is that a user can purchase recommended items even without recommendations. Although the recommendations are very accurate concerning purchase prediction, sales may not increase if such items are recommended. The third issue is that users are reluctant to take action. A user may be interested in a recommended restaurant but may not visit it at the cost of time and money.

In this study, the aforementioned three issues are addressed to increase the causal effect of recommendations. To solve the first issue, recommendation influences are explicitly modeled and incorporated into purchase prediction. This enables accurate prediction of user behaviors for both cases with and without recommendations. It also facilitates the understanding of how recommenda-

tion influences relate to user and item characteristics. To solve the second issue, recommendations are optimized toward uplift. Uplift is defined as an increase in user actions affected purely by recommendations. There are two approaches for uplift optimization. The first approach is to recommend items by the difference between predicted purchase probability with a recommendation and that without. The second approach is to directly optimize a model toward uplift by deriving positive and negative training samples specific to uplift. To solve the third issue, persuasive explanations are provided for recommendations. Although previous explanation methods conveyed how an item matches a user's preference, they failed to convey the motivation for taking actions. Contexts such as time, location, companion, and purposes, affect users' actions. Showing context for item usage can induce users' actions, and a new explanation style using context is proposed.

To increase the causal effect of recommendations, this thesis provides the following three foundations: modeling recommendation influence, uplift optimization, and context style explanation. These foundations give a new perspective for future research in recommender systems.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Research Background

1.1.1 Recommendation System

Recommendation systems[1] are used in various online services [56]. E-commerce sites recommend products to buy, music streaming services recommend music to listen, and news portals recommend news to read. Recommendations help users of these services to find interesting items (i.e., products, music, and news) from numerous candidates. Nowadays, the recommendation has become a standard feature of mainstream online services and users are getting accustomed to being recommended.

Recommendation systems are also beneficial for businesses not only to users. Recommendations facilitate business objectives, such as clicks, sales, and user engagement [38]. Clicks of Google news are increased by better recommendation algorithms [28]. Recommendations boost sales in various domains such as digital versatile disk (DVD) [77], books [128], grocery [32], and games [55]. Recommendations also raise user engagement, which in turn decrease customer churn, thereby saving more than one billion dollars per year in Netflix [38].

Owing to the progress of ubiquitous computing and the accumulation of big data, the application of recommendation systems has attracted much attention. They can recommend locations to visit, people to communicate with, and other various daily activities. Since recommendation systems are presently supporting numerous human lives, the importance of recommendation studies is fast growing.

Recommendation systems infer users' preferences by leveraging various data, such as users' demographics, item attributes, and users' interactions with items. There are many algorithms employed for recommender systems. A simple and non-personalized one is recommending popular items to all users. In general, algorithms for generating personalized recommendations are mainly categorized into collaborative filtering and content-based filtering [4]. Collaborative filtering leverages the interaction logs of other users to recommend items that are preferred by users similar to a target user. On the other hand, content-based filtering recommends items with contents similar to

[1] In this thesis, *recommendation system* and *recommender system* are used interchangeably.

those preferred by a target user. Modern recommendation algorithms apply machine learning for collaborative filtering and content-based filtering [70, 29, 151]. Parameters for expressing user or item characteristics are trained to predict explicit feedback such as ratings and implicit feedback such as purchases. The most common model is a matrix factorization (MF) [70], which expresses a user's preference on an item as an inner product of her or his latent vector and the item's latent vector.

New algorithms are commonly evaluated offline by employing the accuracy of the prediction [43, 44]. For example, purchase[2] logs can be split into training and test data, and models trained with the former data predict purchases in the latter. Items with the highest purchase probabilities are selected (*recommended* virtually in an offline evaluation environment) to each user. An algorithm is considered better than others if its selected items are included in the purchases within the test data more than those by the others. Although the offline evaluation is commonly used, it is known that its recommendation accuracy does not always correlate with the online success of recommenders [36, 115].

In addition to recommended items, an explanation for the recommendation is often provided by recommendation systems [137, 138]. Previous explanation methods can be categorized into four explanation styles: neighbor [47], influence [47, 13], demographic [7], and content [47, 13, 142] styles. For example, neighbor style explanation provides an explanation like "*Users similar to you also purchased this item.*" Explaining the reason for recommendations supports users' decision-making since it helps them to know why an item is recommended. Various user surveys have been conducted to evaluate the explanation of recommendations. It is known that the explanation is generally perceived to be useful and that it increases the trust in recommenders [137, 138]. On the other hand, it does not necessarily increase user actions in a real online environment [135].

Common metrics for measuring the business benefit are click-through rates, conversion, sales volume, sales diversity, and user engagement [56]. Click-through rates and conversion are measured through the actions for the recommended items, while other metrics are measured throughout the actions for whole items. The most important metrics for a company are sales volume and user

[2]Other feedbacks such as clicks, likes, bookmarks, etc. can also be used. In this thesis, the term *purchase* is used to express those positive actions in general.

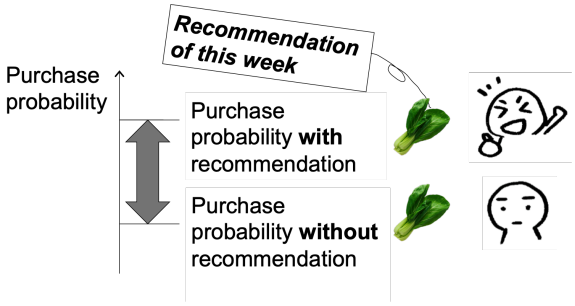


Figure 1.1: Issue 1: The difference between user behaviors with and without recommendations.

engagement. Click-through rates and conversion are considered as proxies for impact on sales volume. User engagement can be defined in many ways, but it is essentially the user's activeness within the service. The increment in user actions improves both sales volume and user engagement; hence, it can be considered as a major goal for business benefit.

1.1.2 Motivation and Objectives

For the success of businesses, recommendation systems aim at increasing user actions, such as clicks and purchases. Therefore, the causal effect of recommendations on clicks and purchases is important. The purpose of the research in this thesis is to increase the causal effect of recommendations. Increasing the causal effect of recommendations can boost revenue for companies which in turn can contribute to the economic growth of the world. Furthermore, it can help users change their lives better. For example, when a user wants to have a healthy dietary habit but she or he is tempted to her/his previous bad eating habits, recommendations can cause a change to the dietary habit. Thus, increasing the causal effect of recommendations contributes to both the economy and individuals.

However, there are three issues encountered when the causal effect of recommendations is increased. The first is that user behaviors with recommendations differ from those without recommendations [58] (Issue 1, see Figure 1.1). Most of the conventional recommendation studies disregard the recommendation influence when they train and evaluate models. Understanding the recommendation influence is essential for initiating actions by recommendations. The second is that a

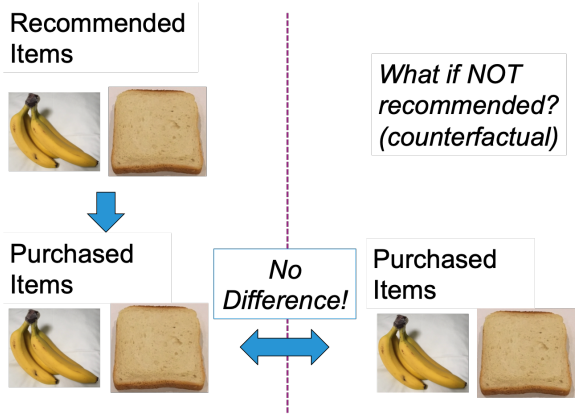


Figure 1.2: Issue 2: Recommended items could have been purchased even without recommendations.

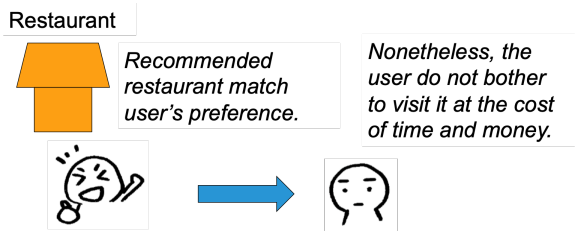


Figure 1.3: Issue 3: A user may not take action even if a recommended item matches the user's preference.

user could have purchased recommended items even without the recommendation [129] (Issue 2, see Figure 1.2). Sales may not increase if such items are recommended, although the recommendation is very accurate concerning purchase prediction. For users, such recommendations do not lead to changes in their behavioral patterns. The third issue is that users are reluctant to take action (Issue 3, see Figure 1.3). A user may be interested in a recommended restaurant but she or he may not spend time and money on visiting it. In particular, this is true for costly actions, for example, buying expensive products, or visiting distant places. Users could also be reluctant to take action to get out of a previous degenerated habit.

In this work, to increase the causal effect of recommendations, the above three issues are addressed as follows.

A solution to Issue 1: Modeling Recommendation Influences. To solve this issue,

recommendation influences are explicitly modeled and incorporated into purchase prediction. This enables accurate prediction of user behaviors for both cases with and without recommendations. It also facilitates the understanding of how recommendation influences relate to the user and item characteristics.

A solution to Issue 2: Uplift Optimization. To address this issue, recommendations are optimized toward uplift. Uplift is defined as an increase in the number of user actions affected by the recommendations. There are two ways to optimize uplift. The first approach is to recommend items by the difference between predicted purchase probabilities with and without recommendations (models designed for Issue 1 can be used for this approach). The second approach is to directly optimize a model toward uplift by deriving positive and negative training samples specific to uplift.

A solution to Issue 3: Persuasive Explanations. To address this issue, persuasive explanations of recommendations are generated. Even though previous explanation methods convey how an item matches a user's preference, they fail to convey the motivation for taking action. Contexts, such as time, location, companion, and purposes, affect users' actions [24]. Since showing context for item usage can induce users' actions, a new explanation style using context is proposed.

1.2 Organizations and Contributions

1.2.1 Organizations of the Thesis

The subsequent chapters of this thesis are organized as follows:

- Chapter 2 describes basic concepts concerning recommendation methods and evaluations. Specifically, a MF model and representative methods for training the model is introduced. Besides, offline evaluation of recommendations is also explained.

- Chapter 3 shows the purchase prediction model that incorporates discount sensitivities of users and items. A recommendation can be regarded as a kind of promotion. Understanding the influence of a price discount, which is another kind of promotion, leads to the understanding of the recommendation influences. First, it is shown that discount effects differ among

users and items through preliminary analysis. Then, a MF model is extended to include discount sensitivities that depend on users and items. The improvement of purchase prediction accuracy by considering the discount sensitivity is demonstrated. Moreover, how discount sensitivities relate to user and item characteristics are shown; the results are discussed in connection with studies in psychology and marketing. Publications related to this chapter are [120, 121], which address Issue 1.

- Chapter 4 presents the purchase prediction model that incorporates the recommendation responsiveness of users and items. By conducting preliminary analysis, it is shown that purchase probability actually differs between cases with and without recommendations and that the difference varies among users. Motivated by this conducted analysis, the recommendation responsiveness is introduced, and the parameters are learned from a combination of purchase logs and recommendation logs. The effectiveness of user- and item-dependent responsiveness is confirmed in terms of both prediction accuracy and recommendation impact[3]. In the cases where recommendation logs are insufficient, the responsiveness needs to be estimated from other sources. How recommendation responsiveness correlates with user and item attributes is investigated and the correlated attributes are utilized to estimate the responsiveness. Publication related to this chapter is [122], which addresses Issues 1 and 2.

- Chapter 5 introduces recommendation influence to exposure modeling [83], which represents a user's action on an item as a two-stage process: first, a user notices the item (exposure), and then the user decides whether or not to purchase it (preference). Recommendations boost awareness differently for various users; while some users are attentive to and trust recommendations, whereas others disregard and distrust them. Furthermore, if a user purchases an item from a recommendation list, she or he would have probably observed other items on the same list. Exposure modeling is extended by considering the aforementioned influences of recommendations on exposure. Experiments using public datasets with recommendation logs demonstrate that considering recommendation influences improves purchase prediction.

[3]The term *recommendation impact* is used for the same meaning with *uplift* in this chapter.

The relationship between the recommendation influences and user demographics is also investigated. Publication related to this chapter is [125], which addresses Issue 1.

- Chapter 6 presents uplift-based evaluation and optimization methods for recommenders. Cases both with and without a recommendation cannot be observed for a specific user-item pair at a particular time instance, making uplift-based evaluation and optimization difficult. To overcome this difficulty, a causal inference framework is applied to estimate the average uplift for the offline evaluation of recommenders. For uplift optimization, the relative priorities of four observable item classes from purchase and recommendation logs are derived. Then the priorities are used to construct both pointwise and pairwise sampling methods for uplift optimization. Experiments with three public datasets demonstrate the effectiveness of proposed optimization methods for improving the uplift. Further, the characteristics of the optimization methods and the resulting recommendations are investigated. Publication related to this chapter is [124], which addresses Issue 2.

- Chapter 7 describes the context style explanation for recommendation systems. The proposed context style explanation method presents contexts (i.e., situations) suitable for consuming the recommended items as explanations. The expected impacts of context style explanations are the following: 1) persuasiveness: recognition of a suitable context for usage motivates users to consume items, and 2) usefulness: envisioning a context can help users to make the right choices since the values of items depend on contexts. The persuasiveness and usefulness of the context-style explanation are investigated by a crowdsourcing-based user study in a restaurant recommendation setting. The context style explanation is compared with the demographic and content style explanations. Besides, the combination of context style and other explanation styles are explored, confirming that hybrid styles improve the persuasiveness and usefulness of the explanation. Further, investigation of the personal preferences for explanation styles reveals how gender and age relate to such preferences. Publications related to this chapter are [119, 123], which address Issue 3.

- Chapter 8 summarizes this thesis.

1.2.2 Contributions of the Thesis

This thesis proposes solutions to three issues encountered when the causal effect of recommendations is increased. More specifically, the major contributions of this thesis are the following:

- A recommendation model that incorporates personal discount sensitivity is proposed and discussed (Chapter 3). As far as we know, this is the first study to unify item preference and discount sensitivity into a single large-scale purchase prediction model.

- A recommendation model that incorporates personal recommendation responsiveness is proposed and investigated (Chapter 4). The individualized difference of recommendation influences is original to this study. Further, the cold-start problem of recommendation logs is newly addressed in this study.

- Extension of exposure modeling to include both direct and indirect recommendation influence on exposure is proposed and investigated (Chapter 5). Considering recommendation influence on exposure modeling is original to this work.

- Uplift-based evaluation and optimization methods are proposed and examined (Chapter 6). This study enables uplift-based evaluations grounded in a causal inference framework. The proposed optimization methods are generic and applicable to most machine-learning-based recommendation models.

- Context style explanation methods are proposed and investigated (Chapter 7). Previous explanation is based on user or item information and this is the first study that utilizes context to explain recommendations. Further, this is the first user study to investigate personal differences in preferences of explanation styles.

Chapter 2

Preliminaries

2.1 Recommendation Methods

2.1.1 Matrix Factorization

Let U and I be defined as a finite set of users and items, respectively. Suppose a user $u \in U$ gives a feedback r_{ui} to an item $i \in I$. For explicit feedback, such as five-scale ratings to movies, r_{ui} takes discrete values from the set $\{1, 2, 3, 4, 5\}$. For implicit feedback, such as purchases of books, r_{ui} takes binary values from the set $\{0, 1\}$, which indicate purchased ($r_{ui} = 1$) or unpurchased ($r_{ui} = 0$). Herein *explicit* means that the user consciously represents an evaluation of an item. On the other hand, *implicit* means that the users' evaluations are inferred from their behaviors; purchased items are probably preferred more than unpurchased ones.

A matrix factorization (MF) [71] decomposes a large matrix of $\{r_{ui} | u \in U, i \in I\}$ with size $|U| \times |I|$ to low-dimensional latent factors.

$$\hat{r}_{ui} = \boldsymbol{\theta_u} \cdot \boldsymbol{\phi_i}, \tag{2.1}$$

where $\hat{r}_{ui} \in \mathbf{R}$ (the real coordinate space) denotes the estimated feedback, and $\boldsymbol{\theta_u} \in \mathbf{R^d}$ and $\boldsymbol{\phi_i} \in \mathbf{R^d}$ denote the d-dimensional latent factors for the user and item, respectively. Typically, $d << |U|, |I|$, hence the representation space of an MF, $(|U| + |I|) \times d$, is much smaller than $|U| \times |I|$. Each dimension of item latent factors can be interpreted as item characteristics, such as seriousness or futurism of a movie, while that of user latent factors as the user's preference on the characteristics. Since r_{ui} is not centered at 0 and has biases, biased MF is often used,

$$\hat{r}_{ui} = \mu + b_u + b_i + \boldsymbol{\theta_u} \cdot \boldsymbol{\phi_i}, \tag{2.2}$$

where $\mu \in \mathbf{R}$ denotes a global bias, $b_u \in \mathbf{R}$ and $b_i \in \mathbf{R}$ denote the biases of the user and item, respectively. For a binary feedback, the sigmoid function, $\sigma(x) = 1/(1 + \exp(-x))$, is often applied to confine the prediction to a proper range $[0, 1]$:

$$\hat{r}_{ui} = \sigma(\hat{x}_{ui}), \tag{2.3}$$

$$\hat{x}_{ui} = \mu + b_u + b_i + \boldsymbol{\theta_u} \cdot \boldsymbol{\phi_i}. \tag{2.4}$$

The MF forms a basis for most modern recommendation models. For example, factorization machines generalize the interactions of the user and item latent factors of MF to the interactions of any features [109]. Neural MF stacks multi-layer perceptron upon interactions of the user and item latent factors [46]. More broadly, the transformation of users and items to the latent factors is similar to word embedding in natural language processing [96, 76, 104, 81], and word2vec can be applied to a recommendation [40, 141, 18].

2.1.2 Model Training

The training methods of recommender models are generally grouped into two categories: pointwise [101, 51, 46] and pairwise [110, 130] methods. The pointwise method optimizes the model to reduce the prediction loss for each feedback, and the pairwise method optimizes the model to reduce the prediction loss for the relative ordering of paired feedbacks. More specifically, for each step of stochastic gradient descent, a parameter Θ is updated as follows.

$$\Theta \leftarrow \Theta - \eta \frac{\partial}{\partial \Theta} \mathcal{L} - \lambda \Theta, \tag{2.5}$$

where η denotes a learning rate, $\mathcal{L}$ denotes a loss function, and λ denotes a regularization coefficient to prevent overfitting. For biased MF, the trainable parameters are $\Theta \in \{\mu, b_u, b_i, \boldsymbol{\theta_u}, \boldsymbol{\phi_i}\}$. The squared error loss ($\mathcal{L}^{se}_{point}$) and the logistic loss ($\mathcal{L}^{ll}_{point}$) are common choices for the pointwise optimization:

$$\mathcal{L}^{se}_{point} = (r_{ui} - \hat{r}_{ui})^2. \tag{2.6}$$

$$\mathcal{L}^{ll}_{point} = -(r_{ui} \log(\hat{r}_{ui}) + (1 - r_{ui}) \log(1 - \hat{r}_{ui})). \tag{2.7}$$

The representative pairwise loss is the Bayesian personalized ranking (BPR) loss [110]. Assuming that user u prefers item i more than item j (e.g., the user purchased i but did not purchase j). Then,

$$\mathcal{L}^{bpr}_{pair} = -\log(\sigma(\hat{r}_{ui} - \hat{r}_{uj})). \tag{2.8}$$

The loss is small when $\hat{r}_{ui} > \hat{r}_{uj}$, whereas it is large when $\hat{r}_{ui} < \hat{r}_{uj}$. Therefore, the BPR loss enables one to learn the correct ordering ($r_{ui} > r_{uj}$).

In real datasets, implicit feedback is largely distorted against negative feedback (e.g., non-purchase) and positive feedback (e.g., purchase) is very sparse. In addition, the fact that the user has not purchased the product does not necessarily mean that the user does not like it. This is because the user may not have noticed it. Consequently, in model training, it is common to assign lower confidences to negative feedbacks either by downweighting them [51, 101] (i.e., multiplying a smaller weight on the loss function) or downsampling them [46, 110][1]. Another method to handle negative feedback is exposure modeling [83], which models the probability of the awareness of unpurchased items.

2.2 Evaluation of Recommendation Systems

Recommendation systems can be evaluated by deploying them in services and observing their performances measured by a specific business metric, such as click-through rates, conversion rates, and sales volumes. Such evaluation is called an online evaluation. However, an online evaluation requires substantial time to deploy and obtain significant results. If the model is weak, then it can greatly damage the business. Therefore, an evaluation before launching the system is necessary.

During the operation of a certain service such as an e-commerce site and a product review site, user feedback logs, such as purchase logs and rating logs, are collected. Recommendation systems can be evaluated by utilizing the collected feedbacks. Such evaluation is called an offline evaluation.

2.2.1 Offline Evaluation

For the offline evaluation of recommendation systems, collected feedbacks are first split into training and test datasets. Then a recommendation model is trained using only the training dataset. If hyperparameters need to be adjusted, the development dataset[2] is sampled from the training dataset

[1] Note that BPR loss is a downweighting method since the number of negative items (denoted by "j" in Equation (2.8)) used in the training is equal to that of positive items (denoted by "i").

[2] The development dataset is also called the validation dataset.

to obtain the hyperparameters that provide the best performance in the development dataset. Finally, the model is evaluated by its performance on the test dataset. The procedure is similar to the common evaluation procedure used for machine learning tasks outside of the research domain of recommender systems.

Recommendation systems are commonly evaluated by their accuracy [43, 44]. There are two common tasks setting: prediction and ranking tasks. In the prediction task, the model is evaluated by how it accurately predicts each feedback. This type of evaluation is used for explicit feedback such as five-scale ratings. Let I_u define a set of items in which user u gives feedback (e.g., ratings). The mean squared error (MSE) is defined as follows:

$$\text{MSE} = \frac{1}{|I_u|} \sum_{i \in I_u} (r_{ui} - \hat{r}_{ui})^2. \tag{2.9}$$

In the ranking task, the model is evaluated by the generated item ranking. This type of evaluation is commonly utilized for implicit feedback, such as purchases. Let $L_u = [i_1, i_2, ..., i_N]$ define a top-N list of recommendations generated by a model for user u. Further, let I_u^+ define a set of items in which the user gives positive feedback (e.g., clicks and purchases). The precision and recall measure the accuracy of the list as follows:

$$\text{Precision@N} = \frac{|I_u^+ \cap L_u|}{N}, \tag{2.10}$$

$$\text{Recall@N} = \frac{|I_u^+ \cap L_u|}{|I_u^+|}. \tag{2.11}$$

Here, "@N" shows that the evaluations are based on top-N lists. The precision measures the ratio of the purchased items in the recommended list. Therefore, it is a proxy of the click-through rates or conversion rates with assumptions that the test dataset reflects the online environment and that recommendations do not change user behaviors. If evaluations need to consider the order in a list, some metrics assign higher weights to items higher in the list. For example, the discounted cumulative gain (DCG) and the normalized discounted cumulative gain (NDCG) are defined as

follows:

$$\text{DCG@N} = \sum_{n=1}^{N} \frac{r_{u,i_n}}{\max(1, \log_2 n)}, \tag{2.12}$$

$$\text{NDCG@N} = \frac{DCG@N}{DCG_{max}@N}. \tag{2.13}$$

DCG downweights the gains of items lower in the list by $\log_2 n$. NDCG is the DCG normalized by the maximum value of DCG (DCG_{max}) that can be obtained by an ideal ranking. Using these metrics, recommendation accuracy can be evaluated offline.

Chapter 3

Modeling Discount Sensitivity

3.1 Introduction

Recommender systems help people find interesting information in the age of "information overload". These systems learn the preference of each user from their past interactions with items and then predict which items will be attractive to them. A vast number of research studies have been dedicated to the advancement of recommendation algorithms and the exploration of their application fields [4, 113].

Recommender systems are beneficial not only for end-users but also for business operators. E-commerce companies increase sales by recommending products [85]. Currently, the use of recommendation engines is prevalent in online shops. While research on recommendations has mainly focused on the prediction of item preferences, users' choices are not always determined by item preferences alone. In the retail business, shopkeepers often offer bargains to attract customers. Consumers are sensitive to these price changes when they make purchasing decisions.

Recently, recommender systems have started to incorporate various psychological aspects of users to fulfill users' needs in more depth [139]. The discount sensitivity of each user is one of these aspects that differs among users.

In this work, a recommendation model with personalized discount sensitivity is proposed by extending the state-of-the-art recommender algorithm. The correlations between discount sensitivity and other personal attributes are also analyzed.

The next section reviews prior work related to this study. Section 3.3 explains the proposed model. Section 3.4 and 3.5 present the experimental procedures and results, respectively. Section 3.6 describes the correlation of discount sensitivity to other features. Finally, Section 3.7 summarizes and concludes this research.

3.2 Related Works

The effect of price promotion has been extensively studied in the field of marketing science [14, 6], and recently, some recommendation studies [147, 64, 53, 61, 80] have taken into account the influence of price. For instance, a hybrid recommender system for supermarkets including discount

information was proposed in [147]. Price has also been personalized using a multi-armed Bandit in [64], depending on three classes of consumers; those who purchase items regardless of promotion, those who purchase items when they are discounted, and those who do not purchase items even if they are discounted. The price range of each item has been incorporated into topic models to learn intrinsic user characteristics concerning prices [53] and the item choice within a category has also been predicted, given the effect of price cuts [61]. Further, the consumer responses to bundled discounts have been modeled, accounting for the correlation of item preferences [80]. This work is different from these studies in that it combines user preference and discount sensitivity in a unified model that learns them simultaneously.

The relationship between personality and recommendation has also attracted interest. Personality can be predicted implicitly [17] and acquired personality can be used to guess item preferences [50, 148]. It has been found that like-logs in social networking services are correlated with personality [17]. Personality similarity has been used to address the difficulty in estimating the preferences of new users [50]. Behaviors in microblog services is an indicator of personality and is useful for inducing brand preferences [148]. Active learning for preference elicitation can leverage personality to acquire ratings efficiently [34]. Personality has been found to be correlated not only to item preference but also to diversity preference [146, 136]. For instance, diversity in movie recommendation was adjusted by personality in [146], and how personality influences the preference of diversity types was investigated in [136]. Discount sensitivity can be regarded as an aspect of personality and its relationship with diversity preference is investigated.

Recommendation model with discount sensitivity can be seen as an example of a multi-criteria recommender system [2], consisting of two criteria: item preference and discount preference. In addition, discount can be considered as one of the contexts and the proposed model can be categorized also in context-aware recommender systems [5]. Relevant contexts depend on domains; in tourism, for example, distance, time available, crowdedness, and knowledge of the surroundings are effective contexts [10].

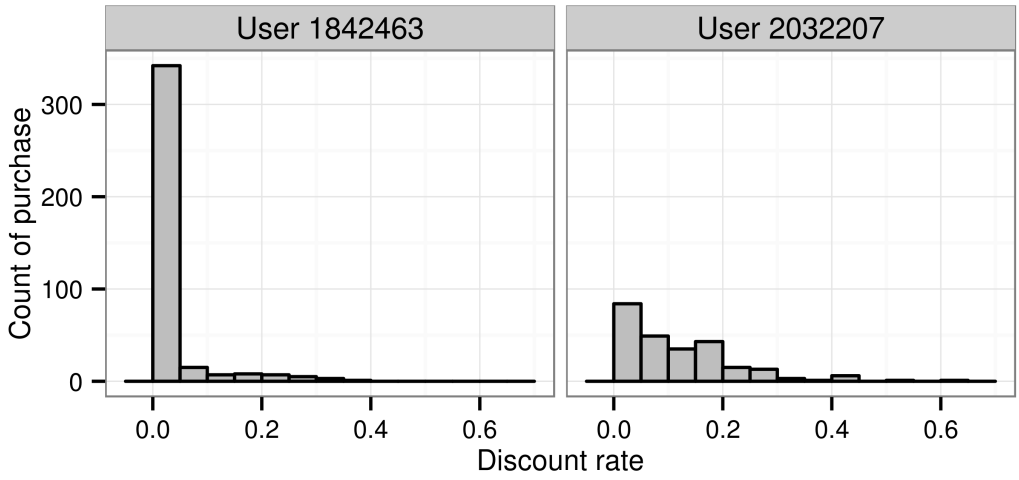


Figure 3.1: Discount rate distributions of purchased items.

3.3 Discount-Sensitive Model

This section describes the extension of the Bayesian personalized ranking (BPR) model [110] with matrix factorization (MF) [71] to incorporate personalized discount effects. Subsection 3.3.1 presents a preliminary analysis of the personal difference in discount sensitivity, which provides motivation to develop a recommendation algorithm that includes it. Subsection 3.3.2 introduces the MF of item preferences in BPR, and Subsection 3.3.3 presents the proposed extensions.

3.3.1 Individual Difference of Discount Sensitivity

The effectiveness of a price promotion depends on the user and the item. Preliminary analysis of the public retail dataset (described in Section 3.4.1) is in accordance with this hypothesis.

The purchase behaviors of two users under various price promotion conditions are compared in Figure 3.1, which shows the distributions of purchase counts for various discount rates. The user in the left panel tends to buy at regular prices and probably has low discount sensitivity. The user in the right panel appears to search for discounted items, and thus should have high discount sensitivity.

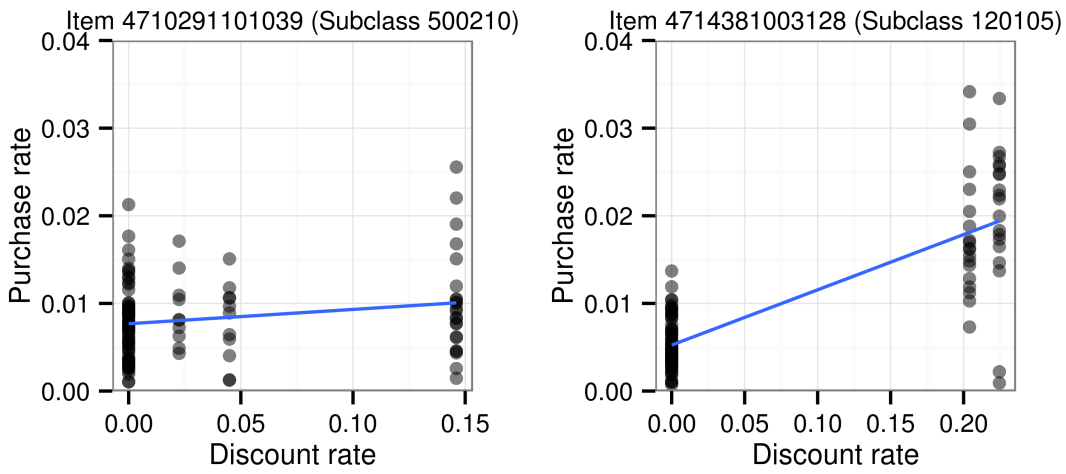


Figure 3.2: Discount rate dependence of purchased rates.

The differences in the discount effect on items were also investigated. Figure 3.2 shows the purchase rates of two items at different discount rates. The purchase rate is defined as the number of purchases divided by the number of visiting users each day. The left and right panels show characteristics of different items in different categories, with blue lines indicating linear regression. Discounts increase sales for both items, but they increase them more for the right item, that is, the discount sensitivity of the right item is much higher than that of the left item.

3.3.2 MF of Item Preference

Collaborative filtering is a recommendation technique to predict item preference of a user from the preferences of similar items and similar users [110]. To overcome the sparsity of feedback data of users for items, MF is commonly used in collaborative filtering [71]. MF decomposes preference into the latent factors of users and items. Adding biases for item preference, the valuation of user u to item i can be expressed as [71]:

$$v_{ui} = \mu + b_i + b_u + \boldsymbol{q}_i^{\mathrm{T}} \boldsymbol{p}_u, \tag{3.1}$$

where $\mu \in \mathbf{R}$ is a bias common to all items and users, $b_i \in \mathbf{R}$ is an item-specific bias, and $b_u \in \mathbf{R}$ is a user-specific bias. Further, $\boldsymbol{q}_i \in \mathbf{R}^n$ is the n-dimensional latent factor of item i and $\boldsymbol{p}_u \in \mathbf{R}^n$ is the n-dimensional latent factor of user u. BPR is a pairwise learning framework [110], that can be adopted for various recommendation tasks [79, 132]. In BPR, probability that user u buys item i and does not buy item j is expressed as a sigmoid function of rating difference between i and j:

$$p(i \in I_u^+ \wedge j \in I \backslash I_u^+) = \frac{1}{1 + \exp(-x_{uij})}, \tag{3.2}$$

where,

$$\begin{aligned} x_{uij} &= v_{ui} - v_{ui} \\ &= b_i - b_j + (\boldsymbol{q}_i - \boldsymbol{q}_j)^{\mathrm{T}} \boldsymbol{p}_u. \end{aligned} \tag{3.3}$$

Here, I_u^+ are items for which a user gives positive feedback (e.g., purchase), and $I \backslash I_u^+$ are items for which the user gives no feedback. Note that μ and b_u are irrelevant in the BPR setting, and hence the scope of the parameters is as follows:

$$\Theta = \{b_i, \boldsymbol{q}_i, \boldsymbol{p}_u | i \in I, u \in U\}. \tag{3.4}$$

Training data set D_s for BPR is composed of user-item triplets:

$$D_s \equiv \{(u, i, j) | i \in I_u^+ \wedge j \in I \backslash I_u^+\}. \tag{3.5}$$

Each triple corresponds to the observation that a user prefers item i over item j. The log-likelihood of this observation is calculated as:

$$\begin{aligned} L &\equiv \ln p(\Theta | D_s) \\ &= \ln p(D_s | \Theta) p(\Theta) - \ln p(D_s). \end{aligned} \tag{3.6}$$

BPR optimizes model parameter Θ by maximizing log likelihood L under training data . Assuming a normal distribution with zero mean and diagonal covariance for priors $p(\Theta)$, the gradient of the log likelihood becomes the following.

$$\frac{\partial L}{\partial \Theta} = \sum_{(u,i,j) \in D_s} \frac{\exp(-x_{uij})}{1 + \exp(-x_{uij})} \cdot \frac{\partial}{\partial \Theta} x_{uij} - \lambda_\Theta \Theta. \tag{3.7}$$

Here λ_Θ is the regularization coefficient for parameter Θ. In this study, stochastic gradient descent was used, as in the original paper [110], and the update rule is:

$$\Theta \leftarrow \Theta + \alpha \left(\frac{\exp(-x_{uij})}{1 + exp(-x_{uij})} \cdot \frac{\partial}{\partial \Theta} x_{uij} - \lambda_\Theta \Theta \right). \tag{3.8}$$

3.3.3 Discount Sensitive Extensions

Let assume that item valuation comes from the preference for the item itself and the preference for discount. The preference for discount can be formalized as the product of the discount rate and discount sensitivity. In order to personalize discount sensitivity, an item-specific bias and a user-specific bias are introduced. Considering the possibility that the combination of a user and an item influences discount sensitivity, latent factors for the user and item are also added. Equation 3.1 is therefore extended to:

$$\begin{aligned} v_{ui} &= \mu + b_i + b_u + \boldsymbol{q}_i^{\mathrm{T}} \boldsymbol{p}_u \\ &\quad + d_i(\mu^d + b_i^d + b_u^d + \boldsymbol{q}_i^{d\mathrm{T}} \boldsymbol{p}_u^d), \end{aligned} \tag{3.9}$$

where $d_i \in \mathbf{R}$ denotes the discount rate of an item i. Terms $\mu^d, b_i^d, b_u^d, \boldsymbol{q}_i^d, \boldsymbol{p}_u^d$ are discount sensitivity terms, which are proportionality coefficients that drive up the valuation in response to the discount. Specifically, $\mu^d \in \mathbf{R}$ is a bias common to all items and users, which shows the general effect of discounts, b_i^d and $b_u^d \in \mathbf{R}$ represent discount sensitivity biases for item i and user u, respectively, and $\boldsymbol{p}_u^d$ and $\boldsymbol{q}_i^d \in \mathbf{R}^d$ respectively correspond to the d-dimensional latent factors of user u and item i.

Including these terms, the rating difference in Equation 3.3 becomes:

$$\begin{aligned} x_{uij} &= b_i - b_j + (\boldsymbol{q}_i - \boldsymbol{q}_j)^{\mathrm{T}} \boldsymbol{p}_u \\ &\quad + d_i(\mu^d + b_i^d + b_u^d) - d_j(\mu^d + b_j^d + b_u^d) \\ &\quad + (d_i \boldsymbol{q}_i^d - d_j \boldsymbol{q}_j^d)^{\mathrm{T}} \boldsymbol{p}_u^d. \end{aligned} \tag{3.10}$$

As a result, the optimized parameter Θ changes, as follows:

$$\Theta = \{b_i, \boldsymbol{q}_i, \boldsymbol{p}_u, \mu^d, b_i^d, b_u^d, \boldsymbol{q}_i^d, \boldsymbol{p}_u^d | i \in I, u \in U\}. \tag{3.11}$$

Training dataset D_s then include the discount rate of each item on the day of shopping $s \in S$.

$$D_s \equiv \{(u, i, j, d_{i,s}, d_{j,s}) | i \in I_u^+ \wedge j \in I \backslash I_u^+ \wedge s \in S\}. \tag{3.12}$$

These training data justify the change of item selection depending on price. For example, user u could have bought item i instead of j when discount $d_{i,s} > d_{j,s}$ and on another day, user j could have bought item j instead of i when discount $d_{i,s} < d_{j,s}$.

The sampling scheme of the training data is also modified. First, user u is chosen randomly and a shopping day s on which the user visited the shop is selected. Next, item i is chosen from the items purchased by the user on that day, and item j from the items not purchased by the user. As explained later in Subsection 3.4.2, items on the shelf might vary each day. As a result, the sampling of j should be confined to items existing on the day.

3.4 Experimental Conditions

This section describes the experimental conditions. First, the dataset used in this study is described. Next, the specifics of training and evaluation are detailed. In Subsection 3.4.3, the tested models and accuracy metrics are specified.

Table 3.1: Statistics of the Ta-Feng dataset and extracted dataset.

Data	#records	#users	#items	#subclasses
Original	817,741	32,266	23,812	2,012
Extracted	132,168	2,373	1,802	373

3.4.1 Dataset

The Ta-Feng dataset [49], which contains the transaction logs of a retail shop, is used for the experiments. This shop sells a wide range of merchandise, from food and grocery items to office supplies and furniture [49]. The transaction logs include user IDs, item IDs, dates, and prices. The record covers four months. The name of the items and subclasses are not published; however, items are categorized into subclasses, and a subclass ID is assigned to each item ID. The unit prices of items on each day were extracted. In most cases, the prices were the same on the same day. If there are multiple prices per day, the median price is the same day price. The discount rates of each item on each day were calculated as:

$$1 - \frac{\text{the day price}}{\max(\text{prices of the item})}. \tag{3.13}$$

A dense subset of the Ta-Feng data is extracted by filtering the data by users that visited the shop 10 times or more and items that sold 100 times or more. This subset comprises 7.4% of the users and 7.6% of the items and includes 16.2% of the records. The basic statistics of the original and extracted data are shown in Table 3.1.

3.4.2 Training and Evaluations

Of the 120 days covered by the dataset, the last 10 days were used for evaluation and the other 110 days for training, considering that the learning precedes the prediction in real scenarios. There was at least one purchase log for all the extracted users in the training subset. In contrast, only 1,850 users had purchase histories in the test subset. The extracted parameters of all users were learned in the training phase, but the evaluation was done for these partial users.

The items on the shelf changed every day. It is assumed that items with at least one purchase

record on a certain day were on the shelf on that day. Of the 1,802 selected items, 1,087 items on average were sold each day. At the evaluation stage, the recommended items of each day were selected from the items on the shelf on that day. Items purchased during training periods were not excluded from the recommended items. In contrast to movie or book consumption, repeat purchase is common in grocery shopping and increasing repeat purchases by recommendations is also beneficial for retailers. Besides, it is not easy to predict repeat purchases because item selection is affected by price discounts and daily availability.

3.4.3 Accuracy Comparison

The conventional MF model was compared with several types of discount sensitive models: MF with non-personalized discount sensitivity (MF-DS(NP)), MF with personalized discount sensitivity (MF-DS(P)), and MF with personalized and user-item-interactional discount sensitivity (MF-DS(PI)).

The area under the curve (AUC), precision and recall were used as evaluation metrics. The metrics were initially calculated for each user on each day and then taken the average for each user over the test period. The statistical significance of the difference in accuracy among different models was verified for user-by-user pairs of metrics using the Wilcoxon signed-rank test. As a representative value, the average of all users was taken for each model.

3.5 Evaluation Results

The proposed model is evaluated for various matrix dimensions (Subsection 3.5.1). Detailed comparisons of each model and the results of the significance tests are shown in Subsection 3.5.2. The data density was adjusted to verify the effectiveness of the proposed models under different densities in Subsection 3.5.3.

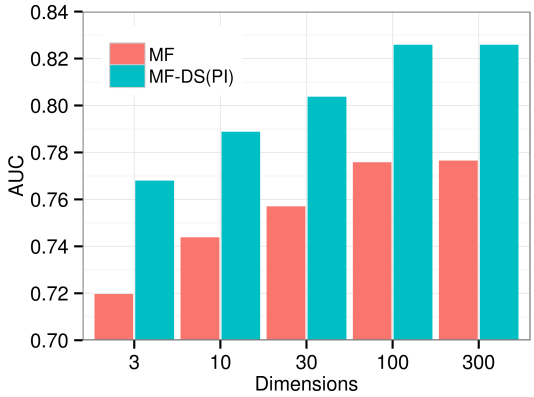


Figure 3.3: AUCs of MF and MF-DS(PI) at different matrix dimensions.

3.5.1 Comparison for Various Matrix Dimensions

The AUC of MF and MF-DS(PI) were evaluated at matrix dimensions ranging from 3 to 300. Figure 3.3 shows results. The discount sensitive models improved the AUC at all dimensions.

3.5.2 Detailed Comparison of Models

Next, a detailed comparison of conventional MF, MF-DS(NP), MF-DS(P), and MF-DS(PI) was conducted. The precision (P) and recall (R) were evaluated when the number of items recommended by the system is 1, 10, and 100. Tables 3.2 and 3.3 show the results at 30 and 100 dimensions, respectively. In most conditions, MF-DS(NP) outperformed MF and MF-DS(P) achieved further improvement. MF-DS(PI) tends to increase accuracy, though not always significantly.

Table 3.2: Accuracy comparison of algorithms at 30 matrix dimensions. Marks * and ** indicate statistically significant differences on the Wilcoxon signed-rank test with $p < 0.1$ and $p < 0.01$, respectively. MF-DS(NP) was compared with MF, MF-DS(P) was compared with MF-DS(NP), and MF-DS(PI) was compared with MF-DS(P).

Dimension 30	AUC	P/R (1 item)	P/R (10 items)	P/R (100 items)
MF	0.7571	0.1467/0.0586	0.0624/0.2323	0.0152/0.4726
MF-DS(NP)	0.8012**	0.1666*/0.0661*	0.0570/0.2171	0.0165**/0.5247**
MF-DS(P)	0.8030**	0.2210**/0.0878**	0.0638**/0.2425**	0.0170**/0.5341**
MF-DS(PI)	0.8038	0.2226/0.0923*	0.0649*/0.2450*	0.0170/0.5387

Table 3.3: Accuracy comparison of algorithms at 100 matrix dimensions. Marks * and ** indicate statistically significant differences on the Wilcoxon signed-rank test with $p < 0.1$ and $p < 0.01$, respectively. MF-DS(NP) was compared with MF, MF-DS(P) was compared with MF-DS(NP), and MF-DS(PI) was compared with MF-DS(P).

Dimension 100	AUC	P/R (1 item)	P/R (10 items)	P/R (100 items)
MF	0.7758	0.1993/0.0812	0.0792/0.2805	0.0172/0.5191
MF-DS(NP)	0.8198**	0.1984/0.0812	0.0646/0.2411	0.0184**/0.5705**
MF-DS(P)	0.8242**	0.2337**/0.0959**	0.0737**/0.2698**	0.0188**/0.5836**
MF-DS(PI)	0.8259*	0.2463**/0.1000*	0.0721/0.2639	0.0191**/0.5875*

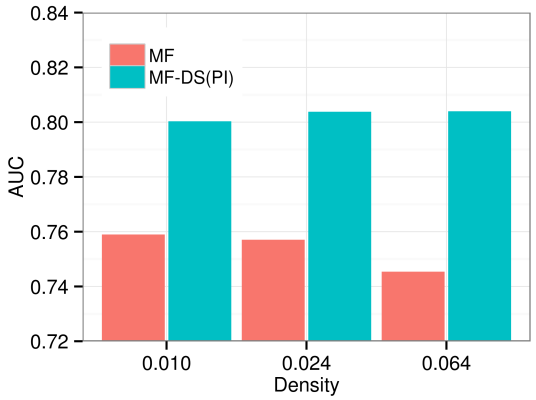


Figure 3.4: AUCs of MF and MF-DS(PI) at different data densities.

3.5.3 Comparison at Various Data Densities

To confirm the universality of the discount sensitive effect, the data density was adjusted. Density is defined by the ratio of purchased item-user pairs to all item-user pairs. Note that the data density of the extracted data in Table 3.1 is 0.024 and experiments in Subsection 3.5.1 and 3.5.2 were conducted at this density. Figure 3.4 shows the AUCs of MF and MF-DS(PI) at different densities. MF-DS(PI) improved the AUC for all densities and tends to be more effective on denser datasets.

3.6 Analysis of Discount Sensitivity

This section investigates the discount sensitivity bias of users and items in MF-DS(P).

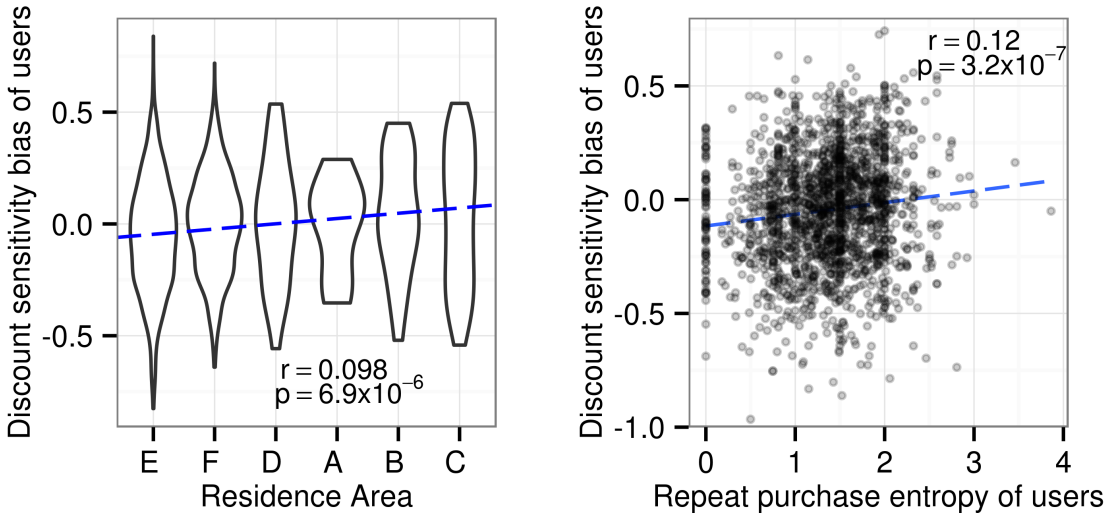


Figure 3.5: Correlations of user discount sensitivity biases with user attributes. The residence area in the left plot is ordered according to distance from the shop.

3.6.1 User Profile and Discount Sensitivity

The Ta-Feng dataset includes customer residence areas and ages. The influence of these factors on discount sensitivity was analyzed.

The left panel of Figure 3.5 shows the distribution of the discount sensitivity biases of users by residential areas. Areas are sorted in order of distance to the shop. The width of the shape expresses the density of the distribution at each vertical value of the discount sensitivity bias. The correlation coefficient r is 0.098 and the statistical significance level p is 6.9×10^{-6}. A weak but significant tendency was found in which users from distant areas responded to discounts more strongly. Distant users might be prone to compensate for their transportation costs with good deals on purchases. Age-dependence was also investigated, but no effect was observed.

Users with a strong tendency to buy particular items (item persistence) might react to discounts differently. The persistence should be closely related to personality. For instance, persistence is most likely correlated positively with neuroticism and negatively with openness and agreeableness. Personal item persistence was extracted from repeat purchases of users within the same category.

In [31], the propensity for diversity, which is the inverse of item persistence, was measured using entropy. The entropy was used as an indicator of the weakness of a user's item persistence.

The per-user subclass-level entropy was first calculated and then taken the average for each user as:

$$H(u) = -\frac{1}{C}\sum_{c\in C}\sum_{i\in I_c} r_{i,u}\log r_{i,u}, \tag{3.14}$$

where I_c denotes the item set in a specific item subclass category, and $r_{i,u}$ represents repeat purchase density, defined as the number of purchases for item i divided by the total purchases of the subclass category. Subclasses purchased less than four times were omitted from the summation. Low entropy in a subclass means that a user has strong persistence in that subclass and tends to buy specific items. High entropy within a subclass means that users tend to purchase various items without worrying about differences among items within a subclass. The average entropy over categories, defined as Equation 3.14, represents whether the user is generally picky or not.

The right panel of Figure 3.5 presents the relation of entropy and discount sensitivity bias of users. Discount sensitivity increases as entropy increases. The correlation coefficient was 0.12 and the statistical significance level p was 3.2×10^{-7}. The result indicates that users without persistence tend to select discounted items, which is reasonable considering that picky users do not like another item regardless of the price offered.

A linear regression model was applied for the discount sensitivity bias of users from the residential area and the entropy. Estimated coefficients are positive (0.024 for residence area and 0.044 for entropy) and significant (the p-values are 4.5×10^5 for residence area and 3.9×10^5 for entropy). The root mean square error (RMSE) of the prediction is 0.2462 for 10-fold cross validation, an improvement of the value of 0.2482, acquired from the mean estimate.

3.6.2 Item Profile and Discount Sensitivity

The correlation between the preference bias of items and discount sensitivity bias of items was examined. The left panel of Figure 3.6 shows that a positive correlation was found between variables with a correlation coefficient of r 0.38. The preference bias of items is similar to item popularity.

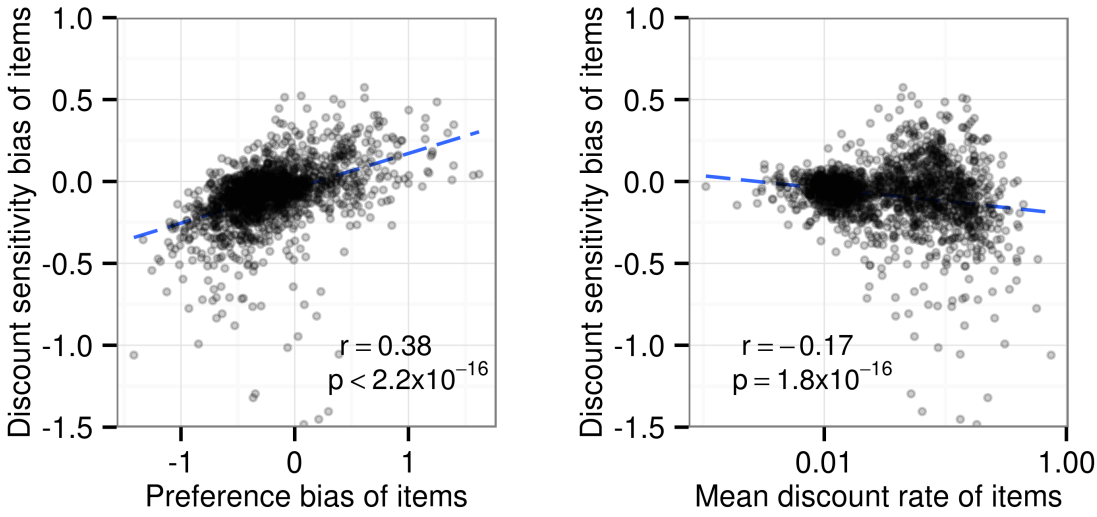


Figure 3.6: Correlations of discount sensitivity bias of items with item attributes.

Therefore, this result suggests that discounts on popular goods are generally more attractive.

It is well known in the field of marketing research that frequent and deep discounts will change consumers' reference price and diminish discount sensitivity [14, 6]. This effect was confirmed by comparing the average discount rates of various items and their discount sensitivity biases. As shown in the right panel of Figure 3.6, the correlation coefficient was -0.17, and a negative correlation was found among the variables.

A linear regression model for item discount sensitivity bias from the popularity and the mean discount was then created. Estimated coefficients are positive (0.186) for the popularity and negative (-0.436) for the mean discount. Both coefficients are significant (the p-values are 2×10^{16} for the popularity and 1.4×10^{9} for the mean discount). The RMSE of the prediction is 0.204 for 10-fold cross validation, an improvement on the value of 0.222, obtained from the mean estimate.

3.7 Conclusion

In this chapter, a recommendation model that incorporates the price discount effect is proposed. Personalized discount sensitivity was introduced into the conventional MF. The proposed model enhances the AUC, precision, and recall in a retail shopping dataset. The results demonstrate that personalized discount sensitivity is a crucially important component in recommender systems in the retail domain.

Individual differences in discount sensitivity associated with the user and item attributes were analyzed. Item discount sensitivities are correlated with item popularity and mean discount rate, and user discount sensitivities are correlated with the distance to the shop and users' item persistence. Persistence is closely related to personality, and these findings may contribute to understanding personality.

In future work, the proposed model could be extended with personality. Combining purchase records and personality information, discount sensitivity can be estimated from personality. Cross-item effects (e.g., how the purchase of a discounted item affects the purchase of another item) have been investigated in marketing science [14, 6]. The fusion of other consumer psychologies and recommendation algorithms is another potential direction for future research.

Chapter 4

Modeling Recommendation Responsiveness

4.1 Introduction

Recommender systems are prevalent in many fields. Electronic commerce websites display items that users might like to buy, and social networking services find people whom users might know and want to connect with. Recommendation research has attracted the interest of both academics and practitioners.

Much effort has been dedicated to algorithms that estimate individual users' preferences. Preferences can be extracted from records of explicit feedbacks such as five-point ratings and implicit feedbacks such as click logs and purchase logs. Traditionally, recommendation research has focused on personal differences in item preferences. On the other hand, it has been indifferent to other personal differences.

Recently, however, new kinds of personal differences have drawn the attention of researchers. For example, the propensity to diversity depends on personality [136, 146]. Novelty-seeking behavior differs among users [66, 150]. Some users accept higher risks to get higher returns from recommendations, while others avoid such risks [152]. Moreover, the change rate in preferences over time is unique to each user [108].

In addition to these personal differences, the responsiveness to recommendations, which is defined as the effect of recommendations to increase a user's rating of an item, might also depend on the user. However, responsiveness has been treated as independent of the users [15, 59, 128] and the individual differences have never been investigated. The responsiveness to recommendations is directly connected to the success of recommendations and requires further investigation in order to design better recommender systems.

Along with the dependence of the responsiveness on users, individual items might trigger different responses in users. This possibility is implied by in-situ experiments in actual stores [75, 32]. Researchers have demonstrated that while certain items in some categories sell easily through recommendations, other items in other categories do not.

In this chapter, a recommender system that incorporates individual differences in responsiveness is proposed. The purchase probability is formulated as a sigmoid function of the sum of a rating

and recommendation response. Recommendation responsiveness is decomposed into common responsiveness, user-specific responsiveness, and item-specific responsiveness. The responsiveness is inferred from the combination of purchase logs and recommendation logs by maximizing the likelihood of the model.

The effectiveness of the proposed model is evaluated in terms of purchase prediction and impact maximization, using a grocery shopping dataset. The accuracy of the predictions made by the proposed model was compared with that of the conventional model that assumes constant responsiveness. The recommendation impact, which is defined as the increase in purchase probability as a result of recommendations, was also compared between the proposed model and the conventional model.

To clarify the characteristics of responsiveness and estimate responsiveness despite inadequate recommendation logs, the correlation between responsiveness and the other attributes of users and items is investigated. The analysis use demographic information about users and features extracted from purchase records. The correlated features were then applied to predict user- and item-specific responsiveness. Furthermore, the recommendation impacts of an individualized responsiveness model is estimated only from the correlated features without using the recommendation log.

The outline for this chapter is as follows. The next section reviews related work. Section 4.3 introduces the proposed model, along with a conventional model and the dataset used for the evaluation. Section 4.4 presents performance comparisons between the proposed model and the conventional model in terms of prediction accuracy and recommendation impact. Section 4.5 describes the correlation of recommendation responsiveness to other attributes and the estimate of responsiveness from correlated attributes. Finally, Section 4.6 summarizes and concludes this chapter.

4.2 Related Work

There are two branches of research that relate closely to this work: meta-personalization beyond item preference, and purchase prediction of recommended items.

4.2.1 Meta-Personalization

Accurately predicting item preferences does not in itself lead to user satisfaction [95]. Consequently, there is a discrepancy between online and offline performances [27]. New perspectives have thus been introduced to recommender systems. For instance, diversity and novelty are vogue topics in recommendation research [19].

As research into diversity and novelty progresses, it is becoming apparent that the desired degree of diversity and novelty differs among users. The propensity to diversity has been measured in terms of the entropy in item selection, and the diversity of recommendations for each user can be adjusted accordingly [31]. Indeed, the preference for diversity is correlated to personality [146], and in particular to "openness to experience" [136]. Recommender system adapted to the novelty-seeking traits of users has also been proposed [66, 150].

Such meta-personalization is not limited to diversity and novelty. Individual differences in risk tolerance have been introduced to recommender systems in order to adjust the tolerance of the variance in rating estimates for each user [152]. Dynamics of preference, or fickleness, also differ among users, and this has been taken into account for recommendations [108].

The proposed modeling of individual users' responsiveness would shed new light on the field of meta personalization.

4.2.2 Recommended Purchase Prediction

A conventional task of recommender systems with implicit feedback is to predict which items users will click or buy [51, 60, 101]. However, such predictions do not always consider the effect of recommendations. Recommendation naturally increases the probability that an item will be clicked or purchased. Recently, the effect of recommendations on purchase predictions has been modeled in several ways. Shani et al. [128] assumed that the increase in purchase probability from recommendations is proportional to the purchase probability without recommendations. Jiang et al. [59] imposed the constraint that consumers buy an item only if the valuation is more than the price of the item. They assumed that recommendation increases the valuation of the item and that

the increase is constant. Bodapati [15] decomposed purchase probability into awareness probability and satisfaction probability, and assumed that recommendations guaranteed awareness of the item.

Whereas the responsiveness to recommendations was considered to be independent of the user in the previous work, this work introduces user-dependent responsiveness to further advance purchase prediction.

4.3 Individualized Responsiveness

4.3.1 Base Model for Purchase Prediction

The probability of binary implicit feedback, such as clicks or purchases, can be formalized in a sigmoid function of a rating of user u on item i ($r_{ui} \in \mathbf{R}$) [60]:

$$p = \sigma(r_{ui}) = \frac{1}{1 + \exp(-r_{ui})}. \tag{4.1}$$

The sigmoid function converts an unbounded real value to a range between zero and one. This is a popular choice for converting a rating to a probability. The matrix factorization is known to perform well in rating prediction [71] and used for the proposed model. The matrix factorization decomposes a rating to the latent factors of the user and the item. Adding bias terms is a common technique because ratings are not zero-centered. Hence rating r_{ui} is expressed as:

$$r_{ui} = b_c + b_u + b_i + \boldsymbol{\theta}_u^{\mathrm{T}} \boldsymbol{\phi}_i, \tag{4.2}$$

where $b_c, b_u, b_i \in \mathbf{R}$ are common, user-specific, and item-specific biases, respectively. Further, $\boldsymbol{\theta}_u, \boldsymbol{\phi}_i \in \mathbf{R}^d$ denote the latent factors of the user and the item, respectively. Recommending an item should increase the probability of purchasing the item. Adding recommendation responsiveness $\gamma \in \mathbf{R}$ to rating r_{ui}, Equation (4.2) becomes

$$r_{ui} = b_c + b_u + b_i + \boldsymbol{\theta}_u^{\mathrm{T}} \boldsymbol{\phi}_i + \delta_{rec}\gamma, \tag{4.3}$$

where $\delta_{rec} \in \{0, 1\}$ is an indicator function of the recommendation. Here, $\delta_{rec} = 1$ when item i is recommended to user u; otherwise, $\delta_{rec} = 0$. Furthermore, γ can be constant or dependent on the user and the item, as discussed below in Subsection 4.3.4. In the proposed model, the parameters Θ to be learned are:

$$\Theta = \{b_c, b_u, b_i, \theta_u, \boldsymbol{\phi}_i, \gamma\}. \tag{4.4}$$

From purchase records and recommendation records, each term is determined such that it minimizes the negative log likelihood (NLL):

$$\begin{aligned} NLL &= -\ln(\prod_{purchase} \sigma(r_{ui} + \delta_{rec}\gamma) \times \ln(\prod_{non-purchase} (1 - \sigma(r_{ui} + \delta_{rec}\gamma)) \\ &= \sum_{purchase} \ln(1 + \exp(-(r_{ui} + \delta_{rec}\gamma))) + \sum_{non-purchase} \ln(1 + \exp(+(r_{ui} + \delta_{rec}\gamma))). \end{aligned} \tag{4.5}$$

Denote by $l_{ui}^{purchase}$ and $l_{ui}^{non-purchase}$ each term in the summation of purchase records and that of non-purchase records, respectively,

$$l_{ui}^{purchase} \equiv \ln(1 + \exp(-(r_{ui} + \delta_{rec}\gamma))), \tag{4.6}$$

$$l_{ui}^{non-purchase} \equiv \ln(1 + \exp(+(r_{ui} + \delta_{rec}\gamma))). \tag{4.7}$$

A stochastic gradient descent (SGD) method was used for iterative learning. For each iteration, SGD randomly picks a user-item pair and updates the parameters in the opposite direction of the gradient. The gradients of $l_{ui}^{purchase}$ and $l_{ui}^{non-purchase}$ are:

$$\frac{\partial}{\partial\Theta} l_{ui}^{purchase} = -\left(\frac{1}{1 + \exp(r_{ui} + \delta_{rec}\gamma))}\right) \frac{\partial}{\partial\Theta}(r_{ui} + \delta_{rec}\gamma), \tag{4.8}$$

and

$$\frac{\partial}{\partial\Theta} l_{ui}^{non-purchase} = -\left(\frac{1}{1 + \exp(-(r_{ui} + \delta_{rec}\gamma)))}\right) \frac{\partial}{\partial\Theta}(r_{ui} + \delta_{rec}\gamma). \tag{4.9}$$

Table 4.1: Summary of the dataset.

Type	#records	#users	#items	#weeks
Purchase	3,743,300	6,937	4,150	39
Recommendation	30,174	6,897	36	10

Table 4.2: Sampling examples of the merged dataset.

User ID	Item ID	Week ID	Purchase?	Recommend?
1	1	1	True	True
1	2	1	True	False
2	1	1	True	False
2	2	1	False	True
1	2	2	True	True
1	3	2	False	False

Parameters are updated as:

$$\Theta \leftarrow \Theta + \zeta_\Theta \left(-\frac{\partial}{\partial \Theta} l_{ui}^{purchase} - \lambda_\Theta \Theta \right), \tag{4.10}$$

$$\Theta \leftarrow \Theta + \zeta_\Theta \left(-\frac{\partial}{\partial \Theta} l_{ui}^{non-purchase} - \lambda_\Theta \Theta \right), \tag{4.11}$$

where $\zeta_\Theta$is the learning rate and λ_Θ is the regularization coefficient of the parameter Θ. Learning the parameters of the model with the SGD always converged in the experiments.

4.3.2 Dataset

Proprietary data from grocery stores were used for the experiments. The dataset included purchase logs and recommendation logs. There was no publically available data with recommendation logs[1], which are crucial for the experiments. Hence, only this dataset was used. The grocery stores mainly deal with foods like vegetables, meat, fish, and various processed foods. The club members of the shop received weekly catalogs of available products and purchased them by mail order. For each week, several "recommended items of the week" were selected by the shop owner. The recommended items were selected from diverse categories of foods in the shop. Flyers with one of the items printed were bundled with the catalog and posted for the club members over ten weeks. The

[1]At the time of the submission of this research.

members targeted for recommendations were chosen randomly each week. The members received at most one flyer per week and a flyer recommended only one item. Table 4.1 summarizes the dataset. From the purchase records, non-purchase records were created, which are user-item pairs comprising users who use the shop on a certain week and the items that they do not purchase despite their availability in that week's catalog. The shop changes the merchandize assortment weekly. The above procedure generated 155 million non-purchase records. Both purchase records and non-purchase records are necessary for evaluating the purchase probability of items.

Purchase records, non-purchase records, and recommendation records were merged each week. It was assumed that the influence of recommendation continued for a week, because the flyers showcased "This week's recommendation" and the merchandise assortment changed each week. Table 4.2 shows examples of the merged dataset. Recommended items differ depending on the week and the user. For example, Item 1 might be available on Week 1 but not on Week 2. Moreover, the same user can repeatedly buy the same item; in this example, User 1 buys Item 2 on both Week 1 and Week 2.

4.3.3 Preliminary Experiment

First, the components of the rating, $b_c, b_u, b_i, \theta_u, \boldsymbol{\phi}_i$, were learned from data without recommendations (Recommend? = False), so as to minimize the NLL. 10% of the data were reserved for validation and hyperparameters such as the learning rate and regularization coefficient were tuned with the validation data. The matrix dimensions were explored from 10 to 1000, and the improvement of the NLL saturated at 300 dimension. Hence, the matrix dimension was set to 300.

Next, the relationship between the predicted purchase probability without recommendations and the observed purchase probability with recommendations was investigated. For all user-item pairs in the recommendation logs (Recommend? = True), the purchase probability without including the recommendation responsiveness γ was calculated. Then, user-item pairs were clustered according to the similarity of the probability. The estimated probabilities for each cluster were averaged. Finally, the observed purchase probability p^{cluster} was calculated using the recommendations for

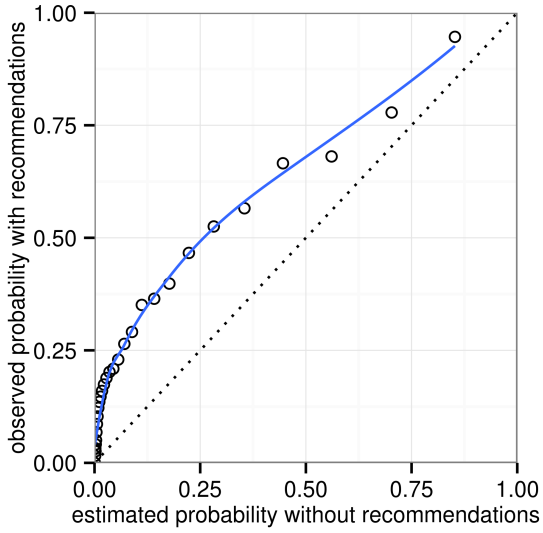


Figure 4.1: Purchase probabilities with and without recommendations.

each cluster, which is defined as:

$$p^{\text{cluster}} = \frac{\text{the number of purchases in a cluster}}{\text{the size of the cluster}}. \tag{4.12}$$

Figure 4.1 shows the results. The x-axis and the y-axis represent the estimated purchase probability without recommendations and the observed probability with recommendations, respectively. If recommendations do not influence purchase probability, both the probabilities should be the same, i.e., $y = x$, as represented by the dotted line in Figure 4.1. The solid line represents the moving average. Here, the solid line is above the dotted line, meaning that recommendations boost the purchase probability. While the probability without recommendations is merely an estimate, it was confirmed that the prediction is fairly accurate (the average NLL, defined later in Equation (4.14), was 0.032 for data without recommendations). In addition, it can be assumed that the prediction error is unbiased, and averaging within the clusters should decrease the error.

The increase in purchase probability among users of different ages is compared. While personality information regarding the users was unavailable in the experiments, it is known that some

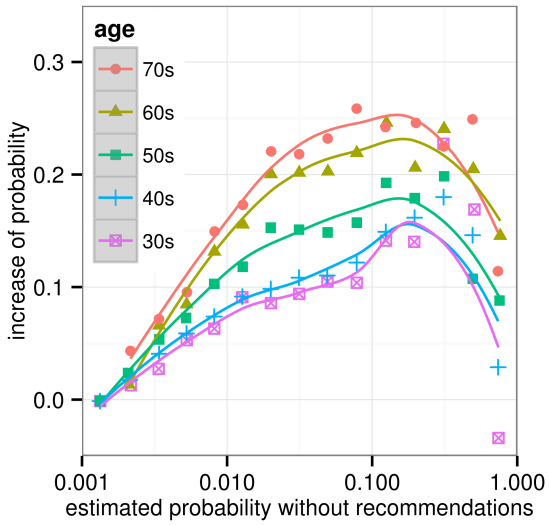


Figure 4.2: Increase in purchase probability from recommendations for various ages. The x-axis is a log scale.

personality traits are correlated with age. For example, age has positive correlations with agreeableness and conscientiousness, and negative correlations with neuroticism, extraversion, and openness [93]. Conscientious people might notice recommendations more often than others, and agreeable people might accept recommendations relatively easily. Hence, it can be expected that the effect of recommendations might depend on age. The clusters of user-item pairs were split according to age, by grouping users in their 30s, 40s, 50s, 60s, and 70s. Figure 4.2 illustrates the difference in the probability increase. Indeed, the increase becomes more significant with advancing age. This result implies that the responsiveness to recommendations depends on the type of user. This supplies the motivation for personalizing responsiveness.

4.3.4 Individualized Responsiveness

The observations in Subsection 4.3.3 indicate that the responsiveness to recommendations differs for each user and each item. Whether a user accepts a recommendation might depend on his or her personality, e.g., the user's agreeableness. In addition, some items might induce impulse shopping,

whereas others might entail more deliberation. The recommendation responsiveness γ is split into a common term γ_c, a user-specific term γ_u, and an item-specific term γ_i:

$$\gamma = \gamma_c + \gamma_u + \gamma_i, \tag{4.13}$$

These terms can be obtained through SGD using the purchase logs and the non-purchase logs with recommendations (Recommend? = True in Table 4.1). This formulation should explain the observed differences in Subsection 4.3.3.

4.4 Comparative Evaluation

In this section, the effect of individualizing recommendation responsiveness is evaluated. The accuracy of purchase prediction and the impact of recommendations were measured. The effectiveness of the model was examined by comparing it with a conventional model, in which responsiveness is constant for all users and items.

4.4.1 Accuracy Comparison

The accuracy of purchase prediction in terms of NLL and precision was compared. The NLL for each user-item pair in the testing data was calculated and then the average was taken.

$$l^{ave} = \frac{\sum_{purchase} l_{ui}^{purchase} + \sum_{non-purchase} l_{ui}^{non-purchase}}{\text{the number of the test data}}. \tag{4.14}$$

The precision was calculated for user-item pairs from the top n% in purchase probability:

$$Precision = \frac{\text{the number of purchase withing top n\%}}{\text{the number of u-i pairs withing top n\%}}. \tag{4.15}$$

In the dataset, 27.1% of all the recommendations were purchased; the baseline for the precision obtained by random recommendation was thus 0.271. The evaluation condition was n = 27.1% because precision and recall are the same at this threshold, and this facilitates the comparison.

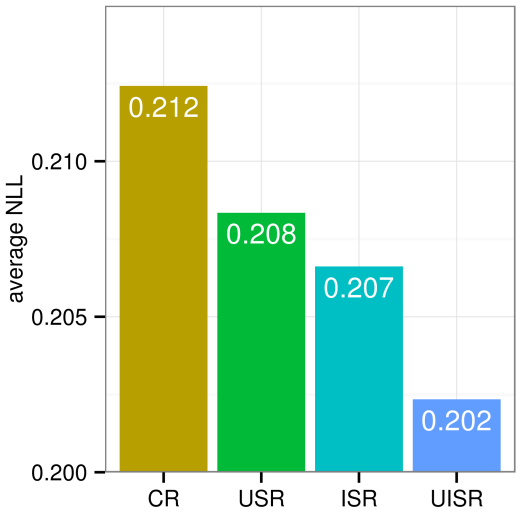


Figure 4.3: Comparison of the average NLL.

The experiment compared four models: constant responsiveness ($\gamma = \gamma_c$, CR), user-specific responsiveness ($\gamma = \gamma_c + \gamma_u$, USR), item-specific responsiveness ($\gamma = \gamma_c + \gamma_i$, ISR), and user- and item-specific responsiveness ($\gamma = \gamma_c + \gamma_u + \gamma_i$, UISR). After pre-training of the components of the rating, $b_c, b_u, b_i, \theta_u, \boldsymbol{\phi}_i$, from the data without recommendations (Recommend? = False), γ for each model was trained using data with recommendations (Recommend? = True). Ten-fold cross validation was performed on each model, and the obtained results were averaged. Figure 4.3 shows a comparison of the mean NLL (l^{ave}), and Figure 4.4 shows a comparison of the precision. UISR outperformed CR with both metrics. Both user- and item-specific terms improved the accuracy and combining them further improved it. The significance of the results were also confirmed. The paired Wilcoxon signed rank test was performed for CR vs. USR/ISR and USR/ISR vs. UISR in terms of both NLL and precision. All of the differences were significant with $p = 0.014$ for CR vs. USR in precision and $p < 0.007$ for the other comparisons. These results demonstrate the effectiveness of modeling user- and item-specific responsiveness for accurate purchase predictions.

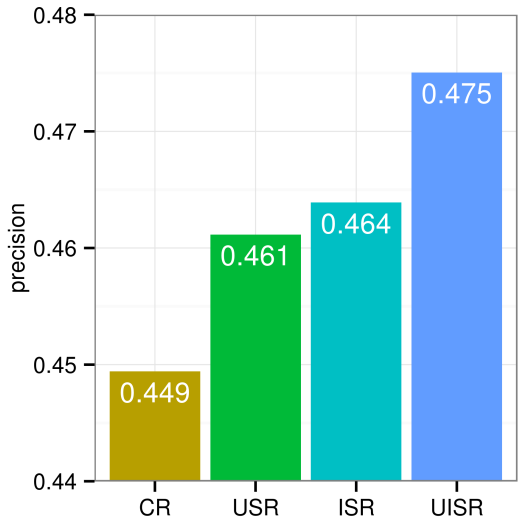


Figure 4.4: Comparison of precision.

4.4.2 Impact Maximization

Next, the recommendation impact, which is defined as the increase in purchase probability through recommendations, is evaluated. Although it is not common in the field of recommendation researches, it is considered to be an important measure of evaluation. Traditional recommender systems are designed to predict whether a user will purchase an item, regardless of whether it is recommended. They then recommend the item with the highest purchase probability. These systems adopt the tacit assumption that there is a positive correlation between the increase in purchase probability from recommendations and the purchase probability without recommendations:

$$p(\delta_{rec} = 1) - p(\delta_{rec} = 0) \propto p(\delta_{rec} = 0). \tag{4.16}$$

However, this assumption is not necessarily true. Consider an extreme example where an item is recommended to a user who has already decided to buy the item in spirit; the purchase probability without a recommendation is almost 100% in this case, and there is no space for a recommendation to further increase this probability. This corresponds to $x \approx 1$ in Figure 4.1. On the other

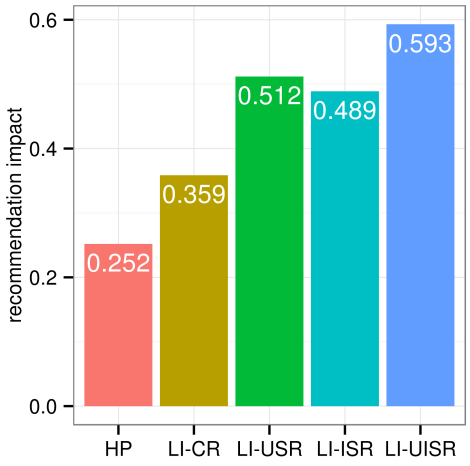


Figure 4.5: Comparison of recommendation impacts.

hand, recommending an item that a user has no intention of buying will not affect the purchase probability either. This corresponds to $x \approx 0$ in Figure 4.1. As can be seen in Figures 4.1 and 4.2, the increase in purchase probability from recommendations is a convex function of the purchase probability without recommendations. Based on the above observations, the convexity would be observed in any recommendation domains, but the peak position may depend on the domain. It is not an optimal strategy to recommend products that are most likely to be purchased without a recommendation.

Recommender systems can be designed for various objectives [118]. End-users might want to maximize utility surplus, which is defined as item utility minus price [42], and maximizing profit is a major concern for retailers [9]. Maximizing recommendation impact can be seen as another form of maximizing the utility surplus or the profit. However, the definition of recommendation impact aims to evaluate the net influence of recommendations.

In order to calculate the recommendation impact, the purchase probability is needed both with and without recommendations. Although their exact values cannot be known, the proposed model can estimate them. Their difference yields the impact of each recommendation. Summing this impact is equivalent to the expected value of the increase in sales volume through recommenda-

tions. Hence, maximizing impact leads directly to profit maximization when commercial goods are recommended for purchase. The experiment compared recommendation impacts obtained with two strategies: 1) the strategy used by traditional systems that recommend items that have the highest purchase probability without recommendations (HP); and 2) recommending items that will result in the largest increase in probability through recommendations (LI). The latter strategy used one of four models introduced in Subsection 4.4.2: the CR, USR, ISR, and UISR models (LI-CR, LI-USR, LI-ISR, and LI-UISR, respectively). There are recommendation logs for 6,897 users and 36 items, and there are 248,292 possible user-item pairs. The best m pairs (the highest probability for Strategy 1 and the largest increase for Strategy 2) were selected from the possible combination, and then the average impact was calculated. The experiment set $m = 3,017$, which is the average number of recommendations per week in the dataset. The UISR model was used for estimating the impacts because it is the most accurate.

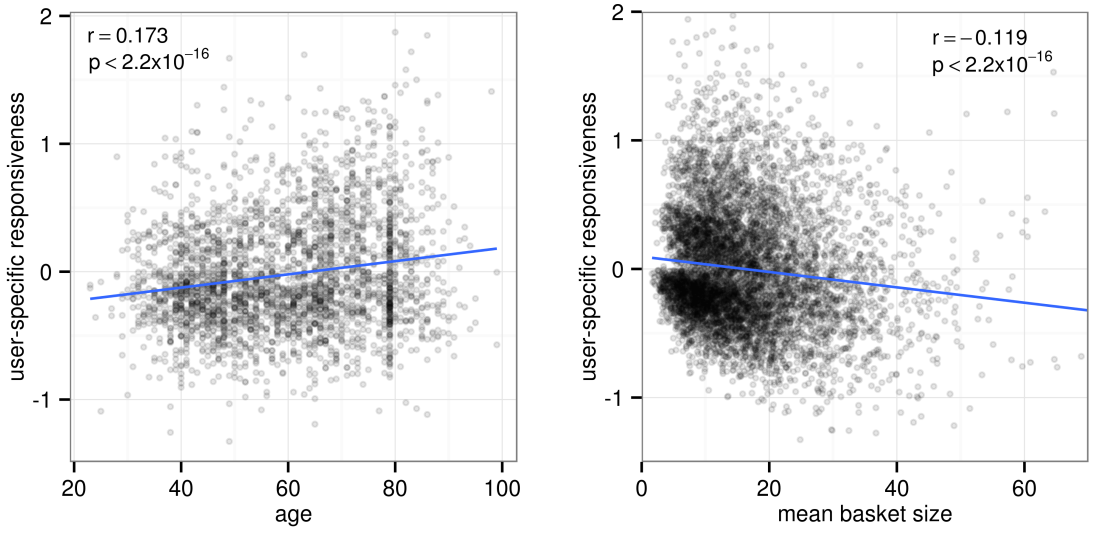
The results are presented in Figure 4.5. LI-CR outperformed HP, proving that maximizing the increase in probability is a superior strategy. Furthermore, LI-UISR had more of an impact than LI-CR. Both LI-USR and LI-ISR surpassed LI-CR, meaning that both user- and item-specific responsiveness contribute to improvement. This result demonstrates the importance of individualized responsiveness for maximizing recommendation impact.

4.5 Responsiveness Estimation

Correlations between responsiveness and other attributes of users and items were investigated to characterize responsiveness and enable estimation despite insufficient recommendation logs. This investigation is described in Subsection 4.5.1. The prediction accuracy of responsiveness from the correlated attributes is evaluated in Subsection 4.5.2.

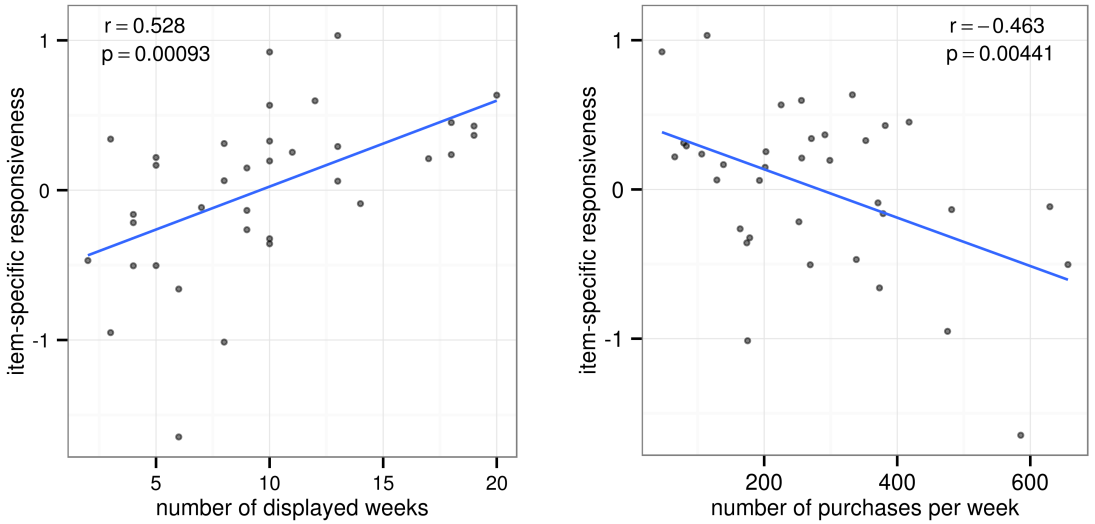
4.5.1 Correlation Analysis

Understanding the correlation between recommendation responsiveness and user and item attributes will lead to clarifying the origin of individual differences in recommendation responsiveness. The



(a) Correlation to age (η_{age}). (b) Correlation to the mean basket size (η_{fam}).

Figure 4.6: Correlation between user-specific responsiveness (γ_u) and user characteristics.



(a) The number of weeks items were displayed (η_{fam}). (b) The number of purchases per week (η_{pop}).

Figure 4.7: Correlation between item-specific responsiveness (γ_i) and item characteristics.

demographic information of users and features derived from purchase records are analyzed.

The demographic features available were age and family size. Among these features, only age was correlated significantly with user-specific recommendation responsiveness, as shown in Figure 4.6 (a). The line shows a linear regression. A positive correlation was found, meaning that elderly people are more easily persuaded to buy an item. It is known that age is positively correlated with agreeableness and conscientiousness [93], and this result might originate from the positive correlations of user-specific responsiveness to agreeableness and conscientiousness.

Some users buy many items at once, while others buy only a few items. The mean basket size of each user is defined as the average number of items purchased at one time. The mean basket size was negatively correlated with recommendation responsiveness, as shown in Figure 4.6 (b). This result suggests that bulk buyers tend to be indifferent to recommendations.

Regarding item-specific responsiveness, the relationship with the number of weeks an item was displayed (η_{fam}) and that with the number of purchases per week (η_{pop}) were examined. It was found that item-specific responsiveness increases the more time an item is displayed (Figure 4.7 (a)) and decreases with the number of weekly purchases (Figure 4.7 (b)). The number of weeks an item is displayed is related to its familiarity to the user, and the number of weekly purchases tracks the popularity of the item. Hence these results suggest that familiar yet unpopular items are good candidates for recommendations.

4.5.2 Estimating Individual Responsiveness

Predicting user- and item-specific recommendation responsiveness is important if the recommendation logs are insufficient. Retailers often keep purchase logs, but they rarely keep recommendation logs. Even when recommendation logs are properly recorded, one can not know the responsiveness when firstly making recommendations to a certain user or when recommending a particular item for the first time. The situation above resembles a situation, in which purchase logs of new users or new items are insufficient for extracting preferences. This problem is known as a cold-start problem in recommender systems [4, 126]. In this case, purchase logs are abundant, but recommendation logs are inadequate. This is a new form of the cold-start problem with the proposed model. To over-

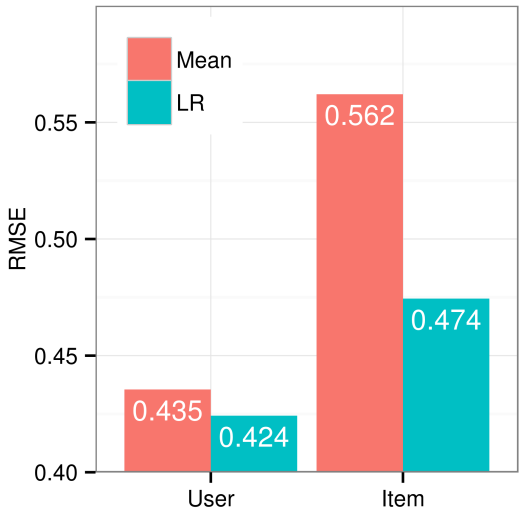


Figure 4.8: Predictive performance of user- and item-specific responsiveness: comparing mean estimates with linear regression estimates.

come the cold-start problem, the responsiveness is estimated from other sources. A linear regression model was built to predict individual responsiveness. Owing to the correlation analysis conducted in Subsection 4.5.1, effective predictors are already known. Thus, user-specific responsiveness can be predicted merely from the age and the mean basket size:

$$\gamma_u = a_1 \cdot \eta_{age} + a_2 \cdot \eta_{bas} + a_3, \tag{4.17}$$

and item-specific responsiveness can be predicted from the familiarity and the popularity:

$$\gamma_i = b_1 \cdot \eta_{fam} + b_2 \cdot \eta_{pop} + b_3, \tag{4.18}$$

The coefficients obtained were, $a_1 = 0.0052$, $a_2 = -0.0080$, $b_1 = 0.0047$, and $b_2 = -0.00055$. It is confirmed that all of the coefficients are statistically significant ($p < 0.01$).

Predictive performance was evaluated by 12 fold cross-validation. The reason why 12 times of cross validation was chosen instead of 10 times of cross validation is that there are 36 items

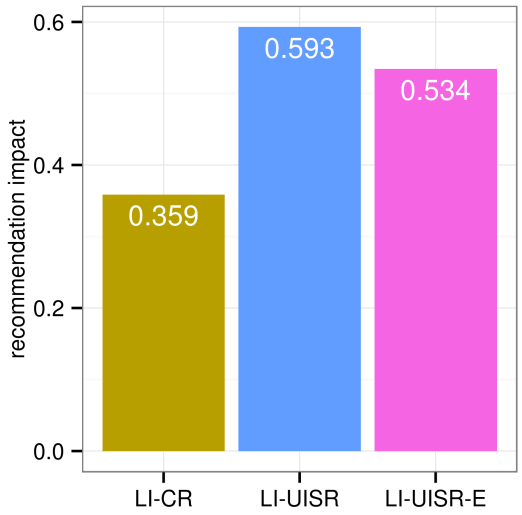


Figure 4.9: Comparison of recommendation impacts among the constant responsiveness model (LI-CR), the user- and item-specific responsiveness model estimated from recommendation logs (LI-UISR), and the model estimated from the correlated attributes (LI-UISR-E).

with the item specific responsiveness, and 36 is divisible by 12. Figure 4.8 shows the root mean square errors (RMSEs) from predicting user-specific responsiveness ("User" in the figure) and item-specific responsiveness ("Item" in the figure). The accuracy of the linear regression model (LR) was compared with that of the mean estimate (Mean). The linear regression outperformed the mean estimate when predicting both user- and item-specific responsiveness. Indeed, item-specific responsiveness improved relatively more than user-specific responsiveness. However, both results were statistically significant (the p-value from the paired Wilcoxon signed-rank tests was 2.6×10^{-6} for user-specific responsiveness and 0.047 for item-specific responsiveness). The individual responsiveness can be estimated at some level merely from the demographic information and purchase logs.

Finally, the effect of the recommendations obtained from the estimated responsiveness was evaluated. Using the user-and item-specific responses estimated from Equations (4.17) and (4.18) for the UISR model (UISR-E), and recommended an item with the largest increase in purchase probability for each user (Strategy 2 in Subsection 4.4.2, LI). Figure 4.9 compares the effects of LI-CR,

LI-UISR, and LI-UISR-E. Note that the results for the LI-CR and LI-UISR models are the same as the results in Figure 4.5. They are again provided in order to facilitate the comparison. LI-UISR-E exceeded LI-CR with the statistical significance ($p < 2.2 \times 10^{-16}$ by the Wilcoxon signed-rank test). LI-UISR was superior to LI-UISR-E, and learning responsiveness directly from recommendation logs is desirable, where available. However, LI-UISR-E closely aligned with LI-UISR. This result shows the potential applicability of the proposed model despite insufficient recommendation logs.

4.6 Conclusions

This chapter proposes a purchase prediction model that incorporates individual differences in recommendation responsiveness. The proposed model improved the accuracy of purchase prediction and the impact of recommendations. These results confirmed the importance of modeling individualized responsiveness. The analysis found a correlation between user-specific responsiveness and both age and the mean basket size. The analysis also found correlations between item-specific responsiveness and both familiarity and popularity. The estimated responsiveness from the correlated attributes outperformed the mean estimates. It was further confirmed that the recommendation impact of the user- and item-specific responsiveness model estimated from the correlated attributes exceeds the impact of the constant-responsiveness model. These findings demonstrate the applicability of the proposed model, even when there are insufficient recommendation logs. This work offers a new research direction in personalizing recommender systems based on recommendation responsiveness.

In future work, the proposed impact-maximization approach can be compared with other approaches, such as diversity- and novelty-seeking approaches. This comparison would help uncover the best recommendation tactics. Whereas a sigmoid function was applied to convert ratings into purchase probabilities, other methods are available, such as Poisson distribution [39]. This research was based on the analysis of purchases and recommendations in grocery shopping. Therefore, investigating the effectiveness of individualized responsiveness in other domains remains for future work. Finally, recommendation responsiveness might relate closely to personality, and this relationship

can be explored to better understand why users are affected by recommendations.

Chapter 5

Exposure Modeling with Recommendation Influence

5.1 Introduction

Recommender systems learn a user's preferences for items from their feedback. There are two types of feedback: explicit feedback such as a 5-scale rating or a thumb-up/down, and implicit feedback such as a click or a purchase. While explicit feedback requires additional actions from users and is scarce, implicit feedback comes from the natural use of services and is abundant. Implicit feedback is more commonly used for real-world recommender systems.

There are two reasons for unpurchased items in the implicit feedback data set: the users do not like them or are not aware of them. Owing to the difficulty in distinguishing between dislike and unawareness, early approaches regarded all unpurchased items as negatives with lower confidences either by downweighting them [51, 101] or downsampling them [110, 46].

Recently, Liang et al. [83] proposed an exposure modeling that represents a user's action on an item as a two-stage process; first a user notices the item (exposure), and then the user decides whether to purchase it (preference). By separating exposure and preference in this way, potential feedback can be better interpreted. The exposure modeling has been extended by incorporating social influences [145, 21] and temporal dynamics [84, 144] to improve recommendation.

However, the recommendation itself influences the awareness of items. Recommendations attract attention to the recommended items differently for different users. Some users might be attentive to and trust recommendations and others might neglect or distrust recommendations. Furthermore, if a user purchases an item from a recommendation list, the user would have likely noticed other items on the same list.

This chapter extends the exposure model by considering the effects of the recommendations. There are two kinds of influence on an item: (1) direct influence by the item's recommendation and (2) indirect influence from other recommended items. The effectiveness of the proposed method is verified by purchase prediction experiments using public datasets with recommendation logs. Furthermore, the recommendation influence is analyzed and its correlation with user demographic is revealed.

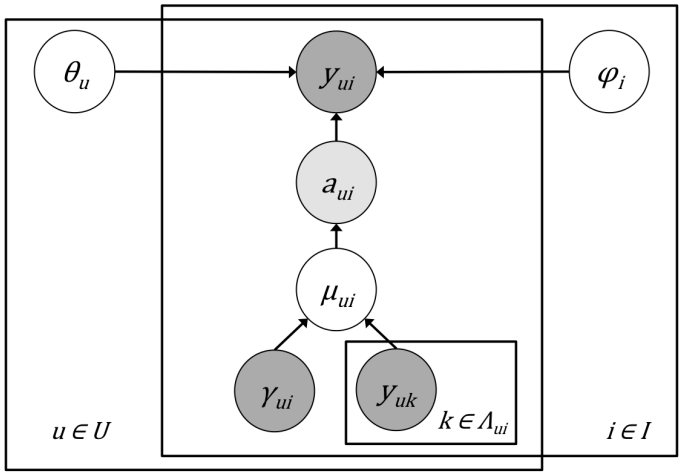


Figure 5.1: Graphical model of RecExpoMF.

Table 5.1: Notation.

Symbol	Description
y_{ui}	Implicit feedback (e.g., purchase or not) of user u to item i
$\boldsymbol{\theta_u}$	Latent vector of user u
$\boldsymbol{\phi_i}$	Latent vector of item i
a_{ui}	Indicator of whether user u has been exposed to item i
μ_{ui}	Prior probability of exposure a_{ui}
γ_{ui}	Indicator of whether item i is recommended to user u
Λ_{ui}	Set of other items on the same recommendation list as item i

5.2 Exposure Modeling with Recommendation Influence

This section shows the exposure matrix factorization with recommendation influence (RecExpoMF). The variables and the graphical model of the proposed method are shown in Table 5.1 and Figure 5.1, respectively.

5.2.1 Exposure Modeling

The proposed method is based on exposure matrix factorization (ExpoMF) [83] described in this subsection. For every combination of user u from user set U and item i from item set I, consider

two sets of binary variables: implicit feedback $y_{ui} \in \{0, 1\}$ and exposure $a_{ui} \in \{0, 1\}$. The variable y_{ui} indicates whether or not user u purchased item i, and a_{ui} indicates whether or not user u is aware of item i. The exposure variable is assumed to follow Bernoulli distribution,

$$a_{ui} \sim \text{Bernoulli}(\mu_{ui}), \tag{5.1}$$

where $\mu_{ui} \in \mathbf{R}$ is the prior probability of exposure. The variable μ_{ui} is expressed by a sigmoid function, $\mu_{ui} = \sigma(x(u, i))$. User and item awareness biases are commonly used for exposure prior [83, 84], $\mu_{ui} = \sigma(x_0(u, i))$ was used as the base expression of the prior, where $x_0(u, i) = b_c^A + b_u^A + b_i^A$, and $b_c^A, b_u^A, b_i^A \in \mathbf{R}$ denote common, user, and item biases, respectively.

When exposed (i.e., $a_{ui} = 1$), the purchase probability is determined by users' preference for items, which is represented by matrix factorization (MF),

$$p(y_{ui} = 1 | a_{ui} = 1) = \text{P}(y_{ui}; \boldsymbol{\theta}_{\boldsymbol{u}}^{\mathsf{T}} \boldsymbol{\phi}_{\boldsymbol{i}}), \tag{5.2}$$

where $\text{P}(y_{ui}; \boldsymbol{\theta}_{\boldsymbol{u}}^{\mathsf{T}} \boldsymbol{\phi}_{\boldsymbol{i}})$ is the Poisson distribution parameterized by $\boldsymbol{\theta}_{\boldsymbol{u}}^{\mathsf{T}} \boldsymbol{\phi}_{\boldsymbol{i}}$, and $\boldsymbol{\theta}_{\boldsymbol{u}} \in \mathbf{R}^{\mathbf{d}}$ and $\boldsymbol{\phi}_{\boldsymbol{i}} \in \mathbf{R}^{\mathbf{d}}$ represent a d-dimensional latent vector of user u's preference and that of item i's attribute, respectively. This study uses Poisson distribution, which works well for binary data [39], and is also used in [84, 87]. Note that Gaussian distribution is used in the original work [83]. A user can not buy an item if the user is not aware of the item; hence,

$$p(y_{ui} = 1 | a_{ui} = 0) = 0. \tag{5.3}$$

In the exposure model, exposure and preference are modeled as independent events, thus a purchase probability is expressed as follows:

$$p(y_{ui} = 1) = p(a_{ui} = 1)\text{P}(y_{ui}; \boldsymbol{\theta}_{\boldsymbol{u}}^{\mathsf{T}} \boldsymbol{\phi}_{\boldsymbol{i}}). \tag{5.4}$$

The reason for inaction comes from either unawareness (i.e., $a_{ui} = 0$) or dislike (i.e., $\boldsymbol{\theta}_{\boldsymbol{u}}^{\mathsf{T}} \boldsymbol{\phi}_{\boldsymbol{i}}$ is low).

5.2.2 Recommendation Influence on Exposure

This subsection models two kinds of recommendation influences on exposure to an item: direct influence ($\tau_{\mathrm{DI}} \in \mathbf{R}$) by the item's recommendation and indirect influence ($\tau_{\mathrm{II}} \in \mathbf{R}$) from other recommended items. They are Incorporated into exposure prior, $\mu_{ui} = \sigma(x_0(u,i) + \tau_{\mathrm{DI}}(u,i) + \tau_{\mathrm{II}}(u,i))$.

Direct Influence on Exposure

It can be expected that recommendations generally increase the exposure probability of the item. Further, the influence of recommendations could differ among users. Some users might be more attentive to recommendations and others might not pay attention to recommendations. Previous research has revealed that recommendation influence is also item-dependent [1, 78]. Hence, the direct influence is expressed as:

$$\tau_{\mathrm{DI}}(u,i) = \gamma_{ui}(b_c^R + b_u^R + b_i^R), \tag{5.5}$$

where $\gamma_{ui} \in \{0,1\}$ is an indicator of recommendation; $\gamma_{ui} = 1$ if item i is recommended to user u. The terms $b_c^R, b_u^R, b_i^R \in \mathbf{R}$ denote common, user, and item biases, respectively, added by the recommendation. Note that recommendation influence might also depend on the displayed order in recommendation lists. The proposed model can be extended to incorporate the order dependence if such information is available[1].

Indirect Influence from Other Recommended Items

Recommendations are often provided as a list of items. If a user does not notice the list, exposures would not increase for any items on the list. Conversely, if a user purchases one of the items on the list, the user is highly likely to have noticed other items on the list. This indirect influence is expressed by,

$$\tau_{\mathrm{II}}(u,i) = b^L(\max_{k \in \Lambda_{ui}} \{y_{uk}\}), \tag{5.6}$$

[1]There were no public datasets with the order information.

where $b^L$is a bias to exposure by indirect influence from other items on the list. Λ_{ui} is a set of items on the same recommendation list as i, but excluding the item i. The variable y_{uk} is a binary variable representing purchase and $\max\{y_{uk}\}$ becomes 1 if a user purchases any other items on the same list.

5.2.3 Inference

The expectation-maximization (EM) algorithm is used to infer model parameters $\{\boldsymbol{\theta_u}, \boldsymbol{\phi_i}, b_c^A, b_u^A, b_i^A, b_c^R, b_u^R, b_i^R, b^L\}$, following the previous research [83, 87, 84].

The E-step computes expectations of the exposure probability of unpurchased items.

$$E[a_{ui}|\boldsymbol{\theta_u}, \boldsymbol{\phi_i}, \mu_{ui}, y_{ui} = 0] = \frac{\mu_{ui}\mathrm{P}(y_{ui}; \boldsymbol{\theta_u^\mathsf{T}\phi_i})}{\mu_{ui}\mathrm{P}(y_{ui}; \boldsymbol{\theta_u^\mathsf{T}\phi_i}) + (1 - \mu_{ui})}. \tag{5.7}$$

For purchased items, the exposure is deterministic, $E[a_{ui}|y_{ui} = 1] = 1$.

The M-step optimizes the model parameters conditioned on the current estimates of exposure probability. The loss function for training is

$$\log Bernoulli(E[a_{ui}]|\mu_{ui}) + E[a_{ui}] \log f(x = y_{ui}|\boldsymbol{\theta_u^\mathsf{T}\phi_i}). \tag{5.8}$$

$E[a_{ui}]$ serves as a prediction target for μ_{ui} and a training weight for $\mathrm{P}(y_{ui}; \boldsymbol{\theta_u^\mathsf{T}\phi_i})$ [84]. Stochastic gradient descent (SGD) is used to update parameters. EM + SGD optimization has been proved to work efficiently in previous work [87, 84].

5.3 Related Works

5.3.1 Implicit Feedback and Exposure Modeling

In implicit feedback datasets, inaction is a mixture of dislike and unawareness; it cannot be treated as completely negative. To handle implicit feedback, weighted matrix factorization (WMF) [101, 51] downweights training from unpurchased items to decrease the confidence for inaction. Downsam-

pling unpurchased items [46] can similarly decrease the confidence. Pairwise learning like Bayesian personalized ranking (BPR) [110] trains a model through comparison between purchased and unpurchased items, which also downsamples unpurchased items. Recently, an exposure model [83] (ExpoMF) has been proposed to better model the reasoning behind inaction. A user's awareness of items can be affected by several factors. People may know items through their friends, and Wang et al. [145] and Chen et al. [21] introduced such social influences on the exposure model. Further, attention to items could change over time due to changes in the item's popularity or the past consumption of items by users. Such temporal dynamics of exposure have been modeled by historical count-based features [84] and a hidden Markov model [144]. This work introduces recommendation influence on exposure. Incorporating social and temporal influences are orthogonal to the proposed methods; combining them would further increase performance.

5.3.2 Models with Recommendation Influence

Users might behave differently with and without recommendations. Increase in purchase probability by recommendations was modeled in [122] (let call it RecResp). Bonner and Vasile [16] proposed the CausE algorithm, which trains two prediction models: one with recommendations and the other without, by regularizing the parameters of the two models to be close to each other. These works do not model exposure. Bodapati [15] proposed a two-step model of user purchases: awareness and satisfaction (let call it AwareSatis), which resembles exposure modeling. AwareSatis assumed that recommendations force users to be aware of the recommended items, which might be too strong an assumption. This work treats recommendation influence on exposure as user- and item-dependent trainable parameters. Furthermore, the three works mentioned above only consider the direct influence of recommendations; indirect influence from other recommended items is unique to the proposed method.

5.4 Experiments

The experiments are designed to address the following research questions:

- **RQ1** Does including direct and indirect recommendation influences improve the performance of the exposure model?
- **RQ2** Who is affected by the recommendation? (more/less)

5.4.1 Experimental Protocol

Datasets

Two public datasets[2] are used: Dunnhumby[3], and Xing[4]. The statistics of datasets after preprocessing are shown in Table 5.2. The purchase and recommendation logs are separated in discrete time intervals because recommended items change over time[5]. Details of each data set are described below.

Dunnhumby. This dataset includes purchase and promotion logs at a retailer. Stores promote items by displaying them in special places. Such items were handled as recommendations and these displays as recommendation lists (average list size is 24.9). The promotions change every week, so the logs are separated by the week. The dataset includes logs from many stores, and promotions are different for each store. If a user visited multiple shops in the same week, one shop was randomly chosen and its logs were used. The datasets were filtered according to the following criteria: shops that have at least one visitor for each week, items recommended for at least one week on average among shops, items that have purchase logs for at least 46 weeks, and users who purchased for at least five weeks. The dataset also includes user demographics which is used to analyze the user-dependent influence of recommendations.

Xing. This dataset includes user interaction with an online job-seeking site. The dataset was provided for a recommender systems competition: the *RecSys Challenge 2017*. The positive user interactions of click, bookmark, and apply, were regarded as purchases. The dataset includes impression logs of the items which are shown to users by the Xing platform. These impressions were

[2]The experiment requires the information of recommendation lists and user interaction logs for both recommended and not-recommended items. Only these two datasets satisfy the requirements.

[3]https://www.dunnhumby.com/careers/engineering/sourcefiles

[4]http://www.recsyschallenge.com/2017/

[5]It was assumed that when recommended, recommendation influence is only within the defined discrete time interval.

Table 5.2: Statistics of datasets after proprocessing.

Dataset	#User	#Item	#Time	#Purchase	#Recommend
Dunnhumby	1,683	2,091	93	428,229	11,968,126
Xing	4,886	6,886	26	28,031	158,783

considered as recommendations, and impressions with the same timestamp as items in the same recommendation list (average list size is 5.8). The dataset was discretized by day. The dataset was filtered according to the following conditions: users and items in Germany, items recommended on at least one day, items that have logs for both first two days and the last two days[6], and users visiting the site on at least three days.

Evaluation Protocols

The experiment evaluated precision (Prec) and normalized discounted cumulative gain (NDCG). The evaluation metrics were calculated for each discrete time, and then they were averaged over the evaluation periods. The datasets were split chronologically for training and evaluation. The lengths of the evaluation periods are 8 and 3 for the Dunnhumby and Xing datasets, respectively. For a dataset with t_d discrete times indexed by 1 to t_d, with the evaluation periods being of length t_e, each phase of validation and testing was conducted as follows:

- *validation phase*: train the model by periods from 1 to $(t_d - 2t_e)$, and evaluate by periods from $(t_d - 2t_e + 1)$ to $(t_d - t_e)$.
- *test phase*: train the model by periods from $(t_e + 1)$ to $(t_d - t_e)$, and evaluate by periods from $(t_d - t_e + 1)$ to t_d.

Compared Methods

The following methods are compared.

- *BPR* [110]: MF trained with a pairwise loss.

[6]This is to confine items available through whole period, since job posts become unavailable when the recruitment is done.

- *WMF* [51, 101]: MF trained with downweighting.
- *RecResp* [122]: MF with recommendation responsiveness.
- *CausE* [16]: Two MFs with and without recommendations.
- *ExpoMF* [83]: Exposure MF, which the proposed method based on.
- *RecExpoMF*: The proposed method with both direct and indirect recommendation influences on exposure.
- *RecExpoMF-D*: RecExpoMF with only direct influence.
- *RecExpoMF-F*: Exposure is forced by recommendations (i.e., $a_{ui} = 1$ if $\gamma_{ui} = 1$) as with AwareSatis [15].

Parameter Settings

All the compared methods are MF-based, and the factor dimensions were set to 100. Poisson distribution was used for all the methods except BPR. SGD with batch size 10000 was used. Regularization coefficients, learning rates, and training iterations were tuned in the validation phase.

5.4.2 Results and Analyses

Performance Comparison (RQ1)

Comparison of prediction accuracy is shown in 5.3 and 5.4. RecExpoMF outperforms ExpoMF, showing the importance of incorporating recommendation influence for exposure modeling. RecExpoMF is mostly better than RecExpoMF-D, which supports the effectiveness of indirect influence. Further, RecExpoMF-D is better than RecExpoMF-F, which validates the proposed modeling of recommendation influences as trainable parameters, unlike forcing exposure as in AwareSatis. The proposed RecExpoMF also outperforms RecResp and CausE, which are recent methods that consider recommendation influence.

Table 5.3: Performance comparison of recommendations in the Dunnhumby dataset. The best result for each metric is highlighted in bold.

	Prec@3	Prec@10	Prec@100	NDCG@3	NDCG@10	NDCG@100
BPR	0.2298	0.1427	0.0385	0.2658	0.2541	0.3524
WMF	0.2720	0.1675	0.0401	0.3106	0.2883	0.3767
RecResp	0.2597	0.1652	0.0404	0.2937	0.2788	0.3712
CausE	0.2715	0.1688	0.0394	0.3124	0.2887	0.3717
ExpoMF	0.2731	0.1686	0.0404	0.3119	0.2920	0.3808
RecExpoMF-F	0.2530	0.1518	0.0387	0.2922	0.2676	0.3572
RecExpoMF-D	0.2720	0.1663	**0.0409**	0.3139	0.2906	0.3840
RecExpoMF	**0.2772**	**0.1699**	**0.0409**	**0.3170**	**0.2943**	**0.3860**

Table 5.4: Performance comparison of recommendations in the Xing dataset. The best result for each metric is highlighted in bold.

	Prec@3	Prec@10	Prec@100	NDCG@3	NDCG@10	NDCG@100
BPR	0.1712	0.0741	0.0104	0.3384	0.3873	0.4284
WMF	0.1658	0.0719	0.0099	0.3332	0.3815	0.4205
RecResp	0.2117	0.0951	0.0111	0.4046	0.4700	0.4971
CausE	0.1784	0.0743	0.0098	0.3561	0.3994	0.4337
ExpoMF	0.1766	0.0708	0.0094	0.3488	0.3854	0.4181
RecExpoMF-F	0.1946	0.0911	0.0109	0.3911	0.4636	0.4894
RecExpoMF-D	0.2000	0.0968	**0.0113**	0.3865	0.4677	0.4931
RecExpoMF	**0.2135**	**0.0987**	0.0112	**0.4147**	**0.4878**	**0.5080**

Table 5.5: Correlation between user demographics (household composition, age group, and income range) and recommendation influence ($b_c^R + b_u^R$) in Dunnhumby.

Household comp.	$(b_c^R + b_u^R)$
One Adult Kids	0.224
Two Adults Kids	0.333
Two Adults No Kids	0.389
Single Female	0.392
Single Male	0.379

Age group	$(b_c^R + b_u^R)$
19-24	0.263
25-34	0.210
35-44	0.429
45-54	0.389
55-64	0.432

Income range	$(b_c^R + b_u^R)$
0-25K	0.375
25-49K	0.420
50-99K	0.398
100-149K	0.286
150K+	0.129

Analysis of Recommendation Influence (RQ2)

The Dunnhumby dataset includes user demographics (household composition, age group, and income range). It was investigated how the trained parameter of user-dependent recommendation influence relates to them (Table 5.5). The recommendation influences ($b_c^R + b_u^R$) are lower for single-parent households (*One Adult Kids*) when compared with two-parent households (*Two Adults Kids*), maybe because they are busy taking care of their children. Regarding age, the parameter is larger for older users (over 35) than for younger users (below 35). This may be because of the difference in personalities [93]; some studies have found that the elderly tend to be more conscientious. Further, high-income classes have lower recommendation influences.

5.5 Conclusions

This chapter extends the exposure model by incorporating the direct and indirect influence of recommendation. Compared with the method proposed in the previous section, this method provides a deeper interpretation of recommendation influence on user actions by decomposing exposure and preference. Experimental results show the effectiveness of the proposed method in terms of prediction accuracy. Relations between recommendation influence and user demographics are further analyzed.

Chapter 6

Uplift-based Evaluation and Optimization

6.1 Introduction

One of the major goals of recommender systems is to encourage positive user interactions, such as clicks and purchases. Because increases in user interactions directly benefit businesses, recommender systems have been utilized in various areas of industry.

Recommendations are typically evaluated in terms of purchases[1] of recommended items. However, these items may have been purchased even without recommendations. For a certain e-commerce site, more than 75% of the recommended items that were clicked would have been clicked even without the recommendations [129]. The true success of recommendations should be represented by the increase in user actions caused by recommendations. Such an increase affected purely by recommendations is called an uplift. The development of a recommender should focus more on the uplift than the accurate prediction of user purchases.

However, evaluating and optimizing the uplift is difficult because of its unobservable nature. An item is either recommended or not for a specific user at a given time instance, so the uplift cannot be directly measured for a given recommendation. This means that there is no ground truth for training and evaluating a model.

Previous studies targeting uplift construct purchase prediction models incorporating recommendation effects [15, 122]. The items recommended are ones that have the largest differences between the predicted purchase probabilities for cases with and without recommendations. Another approach builds two prediction models: one for predictions with recommendations and the other for predictions with no recommendations [16]. All of these methods are based on purchase prediction models optimized for prediction accuracy, even though they target uplift. Improvement of uplift performance is expected by optimizing models directly for the uplift.

This study proposes new evaluation methods and optimization methods for uplift-based recommendation. First, it is shown that common accuracy-based evaluation metrics such as precision do not align with the uplift. Then, evaluation protocols to estimate the average uplift for recommendations is derived, based on a potential outcome framework in causal inference [116, 52, 91]. Further-

[1]The term *purchase* is used to refer to positive interactions in general.

more, optimization methods for recommenders to improve the uplift are proposed. These methods are applied to the matrix factorization model [51, 101, 110], which is the most common model for recommenders. To verify the effectiveness of the proposed optimization methods, the uplift performance of the proposed methods are compared with baselines, including recent recommenders [122, 16] that target the uplift. The characteristics of the proposed uplift-based optimizations and the recommendation outputs are further investigated.

The contributions of this chapter are summarized as follows.

- It proposes offline evaluation metrics for the recommendation uplift (Section 6.2).
- It presents both pointwise and pairwise optimization methods for uplift-based recommendation (Section 6.3).
- It demonstrates the effectiveness of the proposed optimization methods through comparisons with baselines (Subsection 6.5.2).
- It clarifies the characteristics of the optimization (Subsection 6.5.3) and the recommendation outputs (Subsection 6.5.4).

6.2 Uplift-based Evaluation

Recommenders are typically evaluated in terms of recommendation accuracy. A recommender is considered to be better than others if a larger number of its recommended items are purchased. Let refer to this evaluation approach as accuracy-based evaluation. Precision, which is a commonly utilized accuracy metric for recommenders, is defined as the number of purchases divided by the number of recommendations. However, items may have been bought even without recommendations if the user was already aware of and had a preference for those items. Thus, this section aims to evaluate recommenders in terms of the uplift they achieve.

6.2.1 Discrepancy between Accuracy and Uplift

This subsection demonstrates that accuracy metrics such as precision are unsuitable for the goal of increasing user purchases. To describe two cases with and without a recommendation, let adopt the concept of *potential outcome* from causal inference [116, 52, 91]. Let $Y^T \in \{0, 1\}$ be the potential outcome with a recommendation (treatment condition) and $Y^C \in \{0, 1\}$ be the potential outcome without a recommendation (control condition)[2]. $Y^T = 1$ and $Y^C = 1$ indicate that an item[3] will be purchased when recommended and not recommended, respectively. The uplift τ of an item for a given user[4] is defined as $Y^T - Y^C$. Considering the two possible actions of a user in the two given scenarios, there are four item classes for the user:

- **True Uplift (TU)**. $Y^T = 1$ and $Y^C = 0$, hence $\tau = 1$. The item will be purchased if recommended, but will not be purchased if not recommended.
- **False Uplift (FU)**. $Y^T = Y^C = 1$, hence $\tau = 0$. The item will be purchased regardless of whether it is recommended.
- **True Drop (TD)**. $Y^T = 0$ and $Y^C = 1$, hence $\tau = -1$. The item will be purchased if it is not recommended, but will not be purchased if it is recommended.
- **False Drop (FD)**. $Y^T = Y^C = 0$, hence $\tau = 0$. The item will not be purchased regardless of whether it is recommended.

To intuitively illustrate the difference between the uplift and accuracy in an offline evaluation setting, consider four lists of ten recommendations, as shown in Figure 6.1. Let assume that there is an offline dataset, which includes both purchase logs and recommendation logs for a currently deployed recommender. Note that TU items are only purchased if recommended, and TD items are only purchased if not recommended. Purchases of other FU and FD items do not depend on recommendations. The total uplift that would have been obtained if all the ten items were

[2]Control condition means that no recommendation is provided for a specific user-item pair at a given time, not the absence of a recommender.

[3]Items can be product- or category-level ones, depending on the interest of a business.

[4]Recommendations generally change over time. In this study, it is assumed that the influence of a recommendation is within some discrete time interval when recommended.

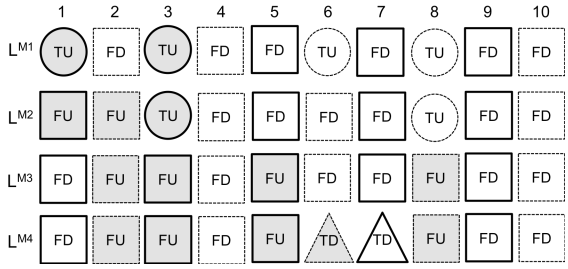


Figure 6.1: A hypothetical example to illustrate the discrepancy between the accuracy and uplift. Four different recommendation lists, L^{M1}, L^{M2}, L^{M3}, and L^{M4} are generated by different recommendation models, M_1, M_2, M_3, and M_4, respectively. The items with solid borders are actually recommended in the offline dataset, and those with dotted borders are not recommended. Shaded items are purchased by the user. The recommendation of circular items (TU) increases purchases, whereas the recommendation of triangular items (TD) decreases purchases. The recommendations of rectangular items (FU or FD) does not affect sales. An evaluation of these lists is presented in Table 6.1.

recommended in the past is shown in Table 6.1. Precision under two settings: one with all items on the list, and the other with only the items recommended in the logs, are also listed. The former is a common setting for the offline evaluations of recommenders [43]. The latter is a setting employed in the previous work [82, 37] to estimate the online performance of a recommender. The former precision value and total uplift exhibit opposite trends for these samples. This means that the best model for achieving a higher uplift cannot be selected based on this former precision. Excluding items without recommendations does not resolve this issue. The latter precision value, calculated using only the recommended items (items with solid boundaries in Figure 6.1), exhibits the same value for all lists and is still unable to select the best model.

As demonstrated by the above illustration, accuracy-based evaluation is not suitable for evaluating the uplift caused by recommenders. Hence an evaluation metric designed for uplift-based evaluation is needed. However, it is not possible to directly calculate the total uplift, because either Y^T or Y^C for a user-item pair is observed at a given time. To overcome this difficulty, a causal inference framework to estimate the average treatment effect is applied.

Table 6.1: Total uplift and evaluation metrics for four recommendation lists in Figure 6.1. The total uplift of a list is indicated by the number of TU items subtracted from the number of TD items. The proposed uplift metric is described in Subsection 6.2.3.

	L^{M1}	L^{M2}	L^{M3}	L^{M4}
Total Uplift (unobservable ground truth)	4	2	0	-2
Precision (all items)	0.2	0.3	0.4	0.5
Precision (items recommended in the log)	0.4	0.4	0.4	0.4
Proposed uplift metric (Subsection 6.2.3)	0.4	0.2	0.0	-0.2

6.2.2 Causal Inference Framework

This subsection introduces the causal inference framework [116, 52, 91], which is applied to the uplift-based evaluation of recommenders in the next subsection. The treatment effect τ for a *subject* is defined as the difference between the potential outcomes with and without treatment: $\tau \equiv Y^T - Y^C$. Note that τ is not directly measurable, because each subject is either treated or not, and either Y^T or Y^C is observed. However, it is possible to estimate the average treatment effect (ATE), which is expressed as $E[\tau] = E[Y^T] - E[Y^C]$.

Let $Z \in \{0, 1\}$ be the binary indicator for the treatment, with $Z = 1$ and $Z = 0$ indicating that the subject does and does not receive treatment, respectively. The covariates associated with the subject are denoted by X, e.g., demographic and past records of the subject before treatment assignment. Let assume that the random variable X takes values in $\mathbf{R^d}$ for some d. Consider N subjects, indexed by n. That is, consider an independent and identically distributed (i.i.d.) sequence of random variables $\left\{(X_n, Y_n^T, Y_n^C, Z_n)\right\}_{n=1}^N$. Denote by S_T and S_C the sets of subjects who do and do not receive treatment, respectively. Naively, the ATE can be estimated as the difference between the average outcomes of the two sets;

$$\hat{\tau} = \frac{1}{|S_T|} \sum_{n \in S_T} Y_n^T - \frac{1}{|S_C|} \sum_{n \in S_C} Y_n^C. \tag{6.1}$$

If treatment is randomly assigned to subjects independent of the potential outcomes, i.e., $(Y^T, Y^C) \perp Z$, then $\hat{\tau}$ converges to the ATE almost surely when $N \to \infty$ (see the proof of [100, Theorem 9.2]).

Because the independence condition $(Y^T, Y^C) \perp Z$ is a strong assumption, instead consider con-

ditional independence $(Y^T, Y^C) \perp Z|X$, which means that the covariates X contain all *confounders* of (Y^T, Y^C) and Z [91]. Under the conditional independence, the inverse propensity scoring (IPS) estimator,

$$\hat{\tau}_{IPS} = \frac{1}{N} \sum_{n \in S_T} \frac{Y_n^T}{e(X_n)} - \frac{1}{N} \sum_{n \in S_C} \frac{Y_n^C}{1 - e(X_n)}, \tag{6.2}$$

is known to be an unbiased estimator of the ATE. Here, $e(X_n) = p(Z_n = 1|X_n)$ is the probability of treatment assignment conditioned on the covariates X, which is called the propensity score [114]. However, the IPS is prone to suffer from the high variance of estimates, because a small propensity score leads to a large weight on an outcome for a certain subject. To remedy this, self-normalized inverse propensity scoring (SNIPS) has been proposed [133]. This adjusts the estimates by the sum of the inverse propensity scores:

$$\hat{\tau}_{SNIPS} = \frac{\sum_{n \in S_T} \frac{Y_n^T}{e(X_n)}}{\sum_{n \in S_T} \frac{1}{e(X_n)}} - \frac{\sum_{n \in S_C} \frac{Y_n^C}{1-e(X_n)}}{\sum_{n \in S_C} \frac{1}{1-e(X_n)}}. \tag{6.3}$$

Under the independence condition $(Y^T, Y^C) \perp Z|X$, the estimator $\hat{\tau}_{SNIPS}$ converges to the ATE almost surely when $N \to \infty$.

6.2.3 Uplift Estimates for Recommenders

This subsection designs evaluation protocols for the uplift caused by recommendation, based on the causal inference framework described in the previous subsection. The goal is to evaluate the uplift performance of a new recommender model M. Let assume that there is an offline dataset comprising purchase and recommendation logs under a currently deployed model D. For the uplift evaluation of recommenders, a treatment Z is a recommendation by D, and $Y_{ui}^T = 1$ means that a user u purchases an item i when it is recommended. Let R be a binary variable such that $R = 1$ if M recommends the item. The objective is to evaluate $E[\tau] = E[Y^T - Y^C|R = 1]$. Let define $p^T = E[Y^T|R = 1]$ and $p^C = E[Y^C|R = 1]$, purchase probabilities of items selected by M with and without an actual recommendation by D, respectively. The uplift can then be interpreted as the increase in purchase probability caused by the recommendation: $p^T - p^C$.

Let L_u^M be a recommendation list for user u, generated by the new model M that needs to be evaluated. In the recommendation logs, there is a list, L_u^D of actually recommended items for the user by the deployed model D. Let assume that some items in L_u^M are included in L_u^D, and some are not. Let write $L_u^{M\cap D}$ for items in both L_u^M and L_u^D, and $L_u^{M\setminus D}$ for items in L_u^M but not in L_u^D. $L_u^{M\cap D}$ and $L_u^{M\setminus D}$ can be regarded as the treatment set S_T and control set S_C, respectively. Therefore, Equation (6.1) becomes,

$$\hat{\tau}_{L_u^M} = \frac{1}{|L_u^{M\cap D}|} \sum_{i \in L_u^{M\cap D}} Y_{ui}^T - \frac{1}{|L_u^{M\setminus D}|} \sum_{i \in L_u^{M\setminus D}} Y_{ui}^C. \tag{6.4}$$

The left and right terms are the purchase probabilities of items in L_u^M if recommended and if not recommended, respectively.

Recommendation lists of Figure 6.1 are evaluated using this metric. The results are shown in the bottom row of Table 6.1. This metric aligns well with the total uplift, indicating that the proposed metric is appropriate for evaluating recommenders in terms of the uplift.

One can also derive the SNIPS estimate of the uplift from Equation (6.3):

$$(\hat{\tau}_{L_u^M})_{SNIPS} = \frac{\sum_{i \in L_u^{M\cap D}} \frac{Y_{ui}^T}{e(X_{ui})}}{\sum_{i \in L_u^{M\cap D}} \frac{1}{e(X_{ui})}} - \frac{\sum_{i \in L_u^{M\setminus D}} \frac{Y_{ui}^C}{1-e(X_{ui})}}{\sum_{i \in L_u^{M\setminus D}} \frac{1}{1-e(X_{ui})}}. \tag{6.5}$$

For recommenders, X_{ui} can be past records of purchase and recommendation, user demographics, and item contents.

As an evaluation metric of the model M, the average is taken over all users U for both estimators:

$$\bar{\tau} \equiv \frac{1}{|U|} \sum_{u \in U} \hat{\tau}_{L_u^M}, \text{ and } \bar{\tau}_{SNIPS} \equiv \frac{1}{|U|} \sum_{u \in U} (\hat{\tau}_{L_u^M})_{SNIPS}. \tag{6.6}$$

In this study, these metrics are employed for the offline evaluation of uplift performance. Let refer to $\bar{\tau}$ as *Uplift*@N and $\bar{\tau}_{SNIPS}$ as *Uplift*$_{SNIPS}$@N, where $N = |L_u^M|$ is the size of the recommendation. Using the protocol described in this subsection, the uplift performance of a new model M is

evaluated offline using the purchase and recommendation logs under a currently deployed model D.

If the purchase probability without recommendation is negligible, e.g., in case of ad clicks, the right terms of Equations (6.4) and (6.5) disappear. The equations then become similar to the previous counterfactual offline evaluation [37, 82]. The proposed evaluation is an extension which considers the possibility of purchase without recommendation.

The uplift estimate by Equation (6.4) depends on the assumption that potential outcomes of items in L_u^M do not relate to logged recommendations by D. The uplift estimate by Equation (6.5) depends on the assumption that covariates X used for estimating the propensity include enough information to resolve dependency between (Y^T, Y^C) and Z. Though it is difficult to guarantee these assumptions, in practice, one can be confident in the evaluation if the results of model comparison are consistent for both $\mathit{Uplift}@N$ and $\mathit{Uplift}_{SNIPS}@N$.

6.3 Uplift-based Optimization

Of the four item classes TU, FU, TD, and TD, defined in Subsection 6.2.1, only TU items can lead to uplift when recommended. However, identification of these four classes requires observation of both Y^T and Y^C, which is not feasible by nature. This implies that there is no observable *ground truth* against which to train models. This section proposes uplift optimization methods to overcome the above problem.

6.3.1 Classification of the Observations

In Subsection 6.2.1, items are categorized into four hidden classes based on the combinations of potential outcomes. Now let categorize items into observable classes from purchase and recommendation logs, while aligning them with the hidden classes. In the observed dataset, for a given user and time instance, an item is either recommended (R) or not (NR); and either purchased (P) or not (NP). This provides the following observable classes (also summarized in Table 6.2):

- An item is recommended and purchased (R-P). Possible hidden classes of the observed item are TU or FU.

- An item is recommended and NOT purchased (R-NP). Possible hidden classes of the observed item are FD or TD.
- An item is NOT recommended and purchased (NR-P). Possible hidden classes of the observed item are FU or TD.
- An item is NOT recommended and NOT purchased (NR-NP). Possible hidden classes of the observed item are TU or FD.

Table 6.2: Observable records and possible hidden item classes. An item is either recommended (R) or not (NR), and either purchased (P) or not (NP).

	P	NP
R	TU or FU	FD or TD
NR	FU or TD	TU or FD

Let define C_{class} as the set of items in $class \in$ {R-P, R-NP, NR-P, NR-NP}[5] for a particular user, $u \in U$. Let also define I_u^+ and I_u^- as the set of positive and negative items for that user. In traditional accuracy-based optimizations [101, 51, 110], $I_u^+ \sim C_{R-P} \cup C_{NR-P}$ (purchased items) and $I_u^- \sim C_{R-NP} \cup C_{NR-NP}$ (non-purchased items). This sampling method is not optimal for uplift and the positive and negative samples need to be redefined. Since TU items result in an uplift, classes that include TU items are considered as positive. Thus, $(C_{R-P} \cup C_{NR-NP})$ should be a reasonable choice for positive item sampling. Following the same reasoning, since C_{R-NP} and C_{NR-P} do not include TU items, $I_u^- \sim (C_{R-NP} \cup C_{NR-P})$.

However, using these positive samples has some problems. Most purchase logs are extremely sparse (NP is large) and most recommenders limit the recommendations to a small number (NR is large). This means that the cardinality of C_{NR-NP} is much larger than that of the other classes and is close to the total number of items. Owing to a consumer's limited purchasing power, the number of TU items should be much smaller than the total number of items. Hence, the probability

[5]In Figure 6.1, C_{R-P} items in L^{M1} are {1,3}; in L^{M2} are {1,3}; in L^{M3} are {3,5}; and in L^{M4} are {3,5}. C_{NR-P} items in L^{M2} are {2}; in L^{M3} are {2,8}; and in L^{M4} are {2,6,8}. C_{R-P} items are either TU or FU and C_{NR-P} items are either FU or TD. Hidden classes of C_{R-NP} and C_{NR-NP} items can be checked similarly.

of the items in C_{NR-NP} belonging to TU should be low:

$$\begin{aligned} P(i \in TU | i \in C_{NR-NP}) &\equiv |TU \cap C_{NR-NP}|/|C_{NR-NP}| \\ &\approx |TU \cap C_{NR-NP}|/|I| < |TU|/|I| \ll 1. \end{aligned} \tag{6.7}$$

On the contrary, considering the fact that recommenders generally improve sales substantially [12], the possibility of the items in C_{R-P} belonging to TU should not be relatively low. Hence,

$$P(i \in TU | i \in C_{R-P}) > P(i \in TU | i \in C_{NR-NP}). \tag{6.8}$$

Because of the above, C_{NR-NP} cannot be considered to be completely positive. Thus, let introduce a parameter α, which is the probability of items from set C_{NR-NP} being sampled as positive. This parameter is further discussed in the following subsection.

6.3.2 Proposed Sampling Method

The optimization methods of recommender models are generally grouped into two categories: pointwise [101, 51, 46] and pairwise [110, 130] methods. This subsection proposes pointwise (ULO_{point}) and pairwise (ULO_{pair}) optimization methods for uplift.

Following the discussion in the previous subsection, items in C_{R-P} are relatively better than the items in the other classes, and thus positive labels are assigned to them. On the contrary, the items in C_{NR-P} and C_{R-NP} are relatively worse and assigned negative labels. The items in C_{NR-NP} are positive with probability α, and negative with probability $1 - \alpha$.

Furthermore, stratified sampling is conducted because the number of items in each observed class is different. A parameter γ_P represents the ratio of sampling from the purchased items. This kind of downsampling for unpurchased items is a common technique for implicit feedback data [46], which is equivalent to downweighting unpurchased items [51, 101]. Similarly, γ_R is the ratio of sampling from the recommended items. For example, the ratio of the items sampled from C_{R-P} is $\gamma_P\gamma_R$ and that from C_{NR-NP} is $(1 - \gamma_P)(1 - \gamma_R)$. For pairwise optimization, the positive and negative samples are selected simultaneously. Positive samples are chosen from $C_{R-P} \cup C_{NR-NP}$

with probability α, and from C_{R-P} with probability $1-\alpha$. The negative samples are selected from the other classes. Candidate classes are sampled with the same probability; that is, if items are selected from $C_{R-P} \cup C_{NR-NP}$, the half are sampled from C_{R-P} and the other half from C_{NR-NP}.

Algorithms 1 and 2 describe the details of each algorithm. r_{ui} is the label for the u-i pair, $\mathcal{L}$ is the loss function, η is the learning rate, and λ is the regularization coefficient. Stochastic gradient descent is used for training. Parameters Θ related to each point or pair are updated at each iteration. As for loss function, the logistic loss [60] is used for the pointwise optimization,

$$\mathcal{L}^{ll}_{point} = -(r_{ui} \log(\sigma(\hat{x}_{ui})) + (1 - r_{ui}) \log(1 - \sigma(\hat{x}_{ui}))). \tag{6.9}$$

The predicted value $\hat{x}_{ui}$ is converted into the label prediction using the sigmoid function, $\sigma(x) = 1/(1 + \exp(-x))$. The Bayesian personalized ranking (BPR) loss [110] is used for the pairwise optimization:

$$\mathcal{L}^{bpr}_{pair} = -\log(\sigma(\hat{x}_{ui} - \hat{x}_{uj})), \tag{6.10}$$

where i is the positive sample and j is the negative sample. In both types of learning, the L_2 regularization term $\Omega = ||\Theta||_2^2$ is added to prevent the overfitting of the parameter Θ. Matrix Factorization (MF) [71] is applied for the expression of $\hat{x}_{ui}$.

$$\hat{x}_{ui} = \mu + b_u + b_i + \theta_u \cdot \phi_i, \tag{6.11}$$

where $\mu \in \mathbf{R}$ is a global bias, $b_u \in \mathbf{R}$ and $b_i \in \mathbf{R}$ are the biases of the user and item, respectively, and $\theta_u \in \mathbf{R^d}$ and $\phi_i \in \mathbf{R^d}$ are the d-dimensional latent factors for the user and item, respectively. In the case of pairwise learning, μ and b_u are dropped by subtraction. The above procedure builds uplift-optimized regularized MF (ULRMF) and uplift-optimized BPR (ULBPR), which are MFs trained by algorithms 1 and 2, respectively.

As for time complexity, the proposed algorithms perform a random sampling of items from prepared sets of observable classes, which is $\mathcal{O}(1)$. The bottleneck is for parameter updates, which is $\mathcal{O}(d)$ for MF with d factor dimensions. This is common to conventional accuracy-based optimiza-

Algorithm 1: Pointwise uplift optimization (ULO_{point}).

Input: α, γ_P, γ_R, η, λ
Output: Θ

Random initialization of Θ
while *not converged* **do**
  draw u from U
  draw i from I, with stratification by γ_P and γ_R
  if $i \in C_{R-P}$ **then**
    set $r_{ui} = 1$
  else if $i \in C_{NR-NP}$ **then**
    if $random(0, 1) \leq \alpha$ **then**
      set $r_{ui} = 1$
    else
      set $r_{ui} = 0$
  else
    set $r_{ui} = 0$
  $\Theta \leftarrow \Theta - \eta \frac{\partial}{\partial \Theta}(\mathcal{L} + \lambda ||\Theta||_2^2)$
return Θ

tions. Further, Subsection 6.5.3 shows empirically that the proposed uplift-based methods converge faster than accuracy-based ones in terms of iterations required.

6.4 Related Work

6.4.1 Causal Inference for Recommenders

Causal inference [116, 52, 91] estimates outcomes through the counterfactual reasoning. It has previously been used to evaluate recommendations in [82, 37, 23, 54, 41], which used IPS, SNIPS, and their extensions. These work evaluated purchases under recommendations, which is equivalent to the use of only the left terms in Equation (6.4) and (6.5). The proposed approach is different, in that it considers the possibility of items being purchased even without recommendation, and evaluate the uplift as the difference between potential outcomes with and without recommendations.

Causal inference is also used to handle the missing-not-at-random (MNAR) nature [92, 131] of user feedback. IPS estimators were used to adjust the item selection bias of explicit feedback [127] and implicit feedback [149]. Another approach to MNAR is exposure modeling [83], which

Algorithm 2: Pairwise uplift optimization (ULO_{pair}).

Input: α, η, λ
Output: Θ
Random initialization of Θ
while *not converged* **do**
  draw u from U
  if $random(0,1) \leq \alpha$ **then**
    draw i from $C_{R-P} \cup C_{NR-NP}$
    draw j from $C_{NR-P} \cup C_{R-NP}$
  else
    draw i from C_{R-P}
    draw j from $C_{NR-P} \cup C_{R-NP} \cup C_{NR-NP}$
  $\Theta \leftarrow \Theta - \eta \frac{\partial}{\partial \Theta}(\mathcal{L} + \lambda ||\Theta||_2^2)$
return Θ

decomposes missing feedback to either a user's unawareness of or dislike for an item. User exposures have been modeled with social influence [145, 21] and temporal dynamics [144], but not with recommendation influence.

6.4.2 Recommendation Targeting Uplift

Most recommendation methods have focused on the accurate prediction of user behavior, and there have only been a few methods targeting uplift. Bodapati [15] proposed a two-stage model of user purchases, comprising awareness and satisfaction stages for items. In this model, recommendations make users aware of the items (let call it *AwareSatis*). Recent work [122] has incorporated user- and item-dependent responsiveness to recommendations into a purchase prediction model (let call it *RecResp*). Very recently, Bonner and Vasile [16] proposed the *CausE* algorithm, which trains two prediction models: one with treatment and the other without. They jointly trained two models as a multi-task objective problem, by regularizing the parameters of the two models to be close to each other. There have also been other methods [140, 120, 143] that incorporated price discount information to improve the purchase prediction accuracy. Price discounts can be regarded as a type of treatment, which could be personalized by recommender systems, although these studies do not target uplift.

Table 6.3: Classification of recommenders targeting uplift.

Approach	Method
Two-Model	*CausE* [16]
Treatment Variable	*AwareSatis* [15], *RecResp* [122]
Label Transformation	*ULRMF* (proposed one), *ULBPR* (proposed one)
Tree-based	-

A closely related field is uplift modeling [30, 107], which is a technique to select the target users of a promotion. Methods of uplift modeling can be classified into four approaches: two-model, treatment variable, label transformation, and tree-based methods. The two-model approach [45] creates two prediction models: one to predict outcomes if treated and the other to predict outcomes if not treated. The treatment variable approach [88] incorporates additional variables for predictions under treatment. Then, the difference between the predicted values with $Z = 1$ and $Z = 0$ is used for uplift prediction. The label transformation approach [57, 65] converts the labels to train the model if not treated. Finally, in the tree-based approach [106, 117], the splitting criteria for a decision tree are modified for uplift.

Recommendation methods targeting uplift are classified in terms of these four approaches in the uplift modeling literature (Table 6.3). *CausE* [16] is basically a two-model approach, enhanced by a regularizer between the two models. *AwareSatis* [15] and *RecResp* [122] are treatment variable approaches. The proposed methods can be classified as label transformation approaches, although the proposed handling of NR-NP as an intermediate between positive and negative (using parameter α) is an original approach to overcome the class imbalance in typical datasets for recommenders.

It has been argued that recommendation should pursue objectives beyond accuracy [95], and various objectives such as diversity, novelty, and serendipity have been studied [63]. Among them, the most relevant is serendipitous recommendation [73, 72, 48], which aims to recommend items relevant, novel, and unexpected to users. Several methods targeting serendipity have been proposed, including serendipity-oriented pointwise [154] and pairwise [89] optimization methods. Serendipity focuses on user perception, while uplift focuses more on user behavior.

6.5 Experiments

Experiments are designed to address the following research questions:

- **RQ1** How do the proposed uplift-based recommenders perform compared with other existing methods?
- **RQ2** What are the properties of uplift-based optimization?
- **RQ3** How do recommended items differ for traditional and uplift-based recommender methods?

6.5.1 Experimental Settings

Datasets and Preprocessing

Experiments are conducted with three publicly available datasets[6]: Dunnhumby[7], Tafeng [49], and Xing[8]. The statistics of datasets after filtering are presented in Table 6.4. The purchase and recommendation logs are separated in discrete time intervals (by day or by week) because recommended items change over time. The details for each dataset are explained below.

Dunnhumby. This dataset includes purchase and promotion logs at a retailer. It provides product category information and these product categories are considered as items. Items featured in the weekly mailer, which is information included in the promotion logs, are regarded as recommendations. Promotions change each week, and so purchase and recommendation logs are separated by week. The dataset includes logs from many stores, and promotions are different for each store. If a user visited a shop when an item was promoted, then the user is regarded to have received a recommendation for the item. The dataset is filtered according to the following conditions: shops that have at least one visitor for each week, items recommended for at least one week on average among the shops, items that existed for at least half the period (47 weeks), and users visiting more than one store in at least five weeks.

[6]Other public datasets are either missing recommendation logs or recording user interactions only for recommended items.

[7]https://www.dunnhumby.com/careers/engineering/sourcefiles

[8]http://www.recsyschallenge.com/2017/

Table 6.4: Statistics of datasets after filtering.

Dataset	#User	#Item	#Time	#Purchase	#Recommend
Dunnhumby	1,760	905	93	968,296	12,479,247
Tafeng	7,520	725	120	362,316	10,988,079
Xing	13,605	15,867	26	105,375	722,882

Tafeng. This dataset contains purchase logs with price information from a Chinese grocery store. This includes the category id for each product, and each category id is considered as a separate item. If the discount ratios of any products in a certain category is over 0.1, then the item is considered to be recommended[9]. The dataset is discretized by days. The dataset is filtered according to the following conditions: items recommended on at least one day, items that existed for at least half of the periods (60 days), and users visiting the shop on at least five days.

Xing. This dataset contains interactions of users at an online job-seeking site. The positive user interactions of click, bookmark, and apply, are regarded as *purchases*. This includes the impression logs of items that are shown to users by the Xing platform. These impressions are considered as recommendations. The dataset is discretized by days. The dataset is filtered according to the following conditions: items recommended on at least one day, items that existed for at least half of the time period (13 days), and users visiting the site on at least three days.

Evaluation Protocols

The uplift performance of each method is evaluated using the proposed $\mathit{Uplift}@N$ and $\mathit{Uplift}_{SNIPS}@N$ for N=10, 30, and 100[10]. $\mathit{Precision}@30$ was also measured, as a reference. Training and evaluation was conducted on each discrete time period. For each training step, a time period from among the training periods is firstly sampled, and then users from among the active users who purchased at least one item during the time period are drown. For evaluation, the metric for each discrete time is calculated, and then they are averaged over the evaluation periods. Chronological splitting of the datasets are conducted for training and evaluation, to prevent the leakage of future information

[9]The discount ratio is calculated as 1- (day price)/(normal price). The median price on the same day is regarded as the day's price. The normal price of a product is defined as the median of the days' prices on all days.

[10]N is set to be typical numbers of recommendations. The average numbers of recommendations users receive at each time are 189.3, 141.7, and 12.1 for Dunnhumby, Tafeng, and Xing, respectively.

for training. The length of evaluation periods are 8, 14, and 3 for the Dunnhumby, Tafeng, Xing datasets. For a dataset with t_d discrete time periods indexed by 1 to t_d, with the evaluation periods being of length t_e, each phase of validation and testing was conducted as follows:

- *validation phase*: train the model by periods from 1 to $(t_d - 2t_e)$, and evaluate by periods from $(t_d - 2t_e + 1)$ to $(t_d - t_e)$.
- *test phase*: train the model by periods from $(t_e + 1)$ to $(t_d - t_e)$, and evaluate by periods from $(t_d - t_e + 1)$ to t_d.

Evaluation of $Uplift_{SNIPS}@N$ requires estimates of propensity $e(X)$. For the Xing dataset, in which recommendations of currently deployed model D are personalized, the propensities are estimated using logistic regression with features representing matches of titles, disciplines, career levels, industries, countries, and regions, between the users and items. The features used were the same as in the baseline model[11] provided by Xing for the *RecSys Challenge 2017* competition. Here, covariates X are these features created from user and item information. The recommendations of D are not personalized in the Dunnhumby and Tafeng datasets. For the Tafeng dataset, the propensities are estimated by the ratio of recommended times for each item in the training periods. That is, past recommendation logs are used as covariates X. For the Dunnhumby dataset, in which time period is much longer (roughly 22 months for Dunnhumby and 4 months for Tafeng), the propensities are estimated by a logistic regression that uses the numbers of purchases and recommendations during previous four weeks as features.

Compared Methods

The following methods are compared.

- *RMF* [101, 51][12]: The regularized MF trained with accuracy-based pointwise optimization.
- *BPR* [110]: The MF trained with accuracy-based BPR loss.
- *RecResp* [122]: The MF with user- and item-specific bias terms for recommendations.

[11] https://github.com/recsyschallenge/2017/tree/master/baseline
[12] While original work downweight unpurchased items, this work downsamples them by γ_P.

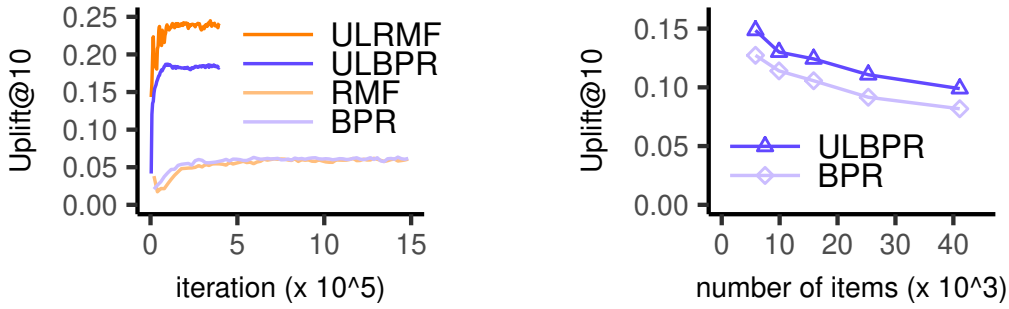
- *CausE* [16]: The joint training of two MFs with and without recommendations.
- *CausE-Prod* [16]: The variant of CausE, which has common user factors for two MFs.
- *ULRMF* (proposed): The MF trained with proposed ULO_{point}.
- *ULBPR* (proposed): The MF trained with proposed ULO_{pair}.

RMF and BPR are trained by conventional accuracy-based optimization, i.e., $C_{R-P} \cup C_{NR-P}$ as positive samples. RecResp and CausE are recent recommendation methods targeting uplift. For these methods, the uplift is predicted using the difference between purchase probabilities with and without recommendations, and they are used for top-N recommendation as described in [122]. They once train models for accurate purchase prediction ($C_{R-P} \cup C_{NR-P}$ as positive samples), and then target uplift using the accuracy-optimized models. Only the proposed methods, ULRMF and ULBPR, are optimized directly for uplift by the unique sampling strategy described in Subsection 6.3.2.

Implementation and Parameter Settings

All the compared methods are latent factor models, and the factor dimensions are set to 100. Adam [68] was employed with batch size 1000, and the initial learning rate was set to 0.0001. For pointwise learning, there are two stratifications of data sampling: one is between purchased and unpurchased items (by γ_P), and the other is between recommended and not recommended items (by γ_R). γ_P is set to 0.2, an optimal ratio for various datasets in [46], for RMF and ULRMF. This stratification is not applied to RecResp and CausE, because it distorts the purchase probability and prohibits the uplift prediction. γ_R is set to 0.5 for RecResp, CausE, and ULRMF.

The regularization coefficient $\lambda \in \{10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}\}$ and other model-specific hyperparameters were tuned in the validation phase to maximize *Uplift@10*. The model-specific hyperparameters and their exploration ranges are as follows: regularization coefficient between the treatment and control latent factors $\lambda_{bet} \in \{10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}\}$ and its distance metric $\in \{L^1, L^2, cosine\}$ for CausE, the probability that NR-NP is regarded as positive $\alpha \in \{1.0, 0.8, 0.6, 0.4, 0.2, 0.0\}$ for ULRMF and ULBPR.



(a) Learning curve convergence.

(b) Different filtering criteria.

Figure 6.2: Scalability of the proposed methods. Used datasets are Dunnhumby for (a) and Xing for (b).

6.5.2 Performance Comparison (RQ1)

The uplift performances between the proposed methods and baselines are compared (Table 6.5). The key observations are the followings:

- The proposed ULRMF or ULBPR methods achieve the best in *Uplift*@N and $Uplift_{SNIPS}$@N for most cases.
- The accuracy-based methods (RMF and BPR) perform the best in *Precision*; however, for the most part, they perform worse in the uplift metrics than other methods.
- The methods targeting uplift (RecResp, CausE, and the proposed methods) tend to outperform RMF and BPR. This implies that the proposed uplift metrics can measure the uplift improvement as expected.

6.5.3 Uplift-based Optimization Properties (RQ2)

The learning curves are investigated in the Dunnhumby dataset (Figure 6.2 (a)). *Uplift*@*10* increases with training iterations. The learning curve of ULBPR tends to be steadier than that of ULRMF. ULRMF and ULBPR converge faster than RMF and BPR, which shows the scalability of the proposed methods in terms of computation time.

Table 6.5: Performance comparison in the three datasets. The best result of each metric is highlighted in bold. * indicates that the method outperforms the others at a significance level of $p < 0.01$ by paired t-tests. Comparisons are only with other families of methods, namely, CausE-Prod is not compared with CausE or ULRMF with ULBPR.

	Uplift			*Uplift*$_{SNIPS}$			*Precision*
	N=10	N=30	N=100	N=10	N=30	N=100	N=30
RMF	0.0644	0.0496	0.0356	0.0393	0.0247	0.0174	**0.1598***
BPR	0.0729	0.0505	0.0353	0.0431	0.0259	0.0168	0.1545
RecResp	0.1594	0.1043	0.0471	0.1009	0.0578	0.0260	0.1056
CausE	0.1621	0.1165	0.0575	0.0942	0.0481	0.0223	0.0862
CausE-Prod	0.1889	0.1042	0.0471	0.1298	0.0539	0.0236	0.0801
ULRMF	**0.2477***	**0.1897***	**0.1227***	**0.1726***	**0.0816***	0.0234	0.0400
ULBPR	0.1881	0.1481	0.1068	0.1481	0.0815	**0.0345***	0.0416

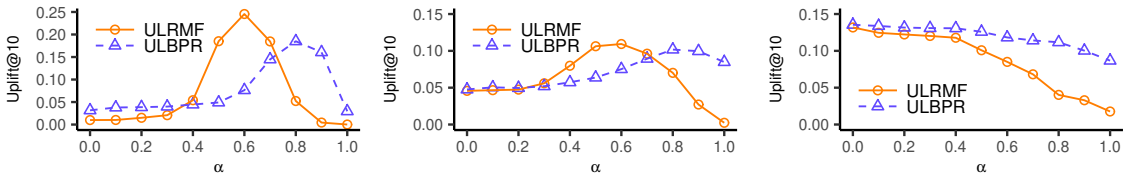
(a) Dunnhumby dataset.

	Uplift			*Uplift*$_{SNIPS}$			*Precision*
	N=10	N=30	N=100	N=10	N=30	N=100	N=30
RMF	0.0732	0.0566	0.0374	0.0706	0.0526	0.0314	0.0565
BPR	0.0713	0.0534	0.0360	0.0685	0.0522	0.0328	**0.0582***
RecResp	0.0595	0.0484	0.0286	0.0532	0.0726	0.0325	0.0560
CausE	0.1157	0.0745	0.0384	0.1011	0.0696	0.0306	0.0403
CausE-Prod	**0.1230***	0.0609	0.0273	**0.1077***	0.0419	0.0173	0.0341
ULRMF	0.1145	**0.1109***	**0.0919***	0.0986	**0.0826***	**0.0467***	0.0129
ULBPR	0.1026	0.0986	0.0777	0.0916	0.0796	0.0376	0.0188

(b) Tafeng dataset.

	Uplift			*Uplift*$_{SNIPS}$			*Precision*
	N=10	N=30	N=100	N=10	N=30	N=100	N=30
RMF	0.1037	0.1118	0.1108	0.1038	0.1121	0.1110	0.0189
BPR	0.1056	0.1168	0.1157	0.1057	0.1168	0.1157	**0.0239***
RecResp	0.0839	0.1017	0.1149	0.0838	0.1015	0.1148	0.0060
CausE	0.1163	0.1243	0.1280	0.1163	0.1243	0.1281	0.0099
CausE-Prod	0.1159	0.1230	0.1296	0.1158	0.1228	0.1297	0.0088
ULRMF	0.1227	0.1266	**0.1298**	0.1228	0.1268	**0.1299**	0.0104
ULBPR	**0.1242***	**0.1283**	0.1282	**0.1244***	**0.1285**	0.1284	0.0113

(c) Xing dataset.



(a) Dunnhumby dataset. (b) Tafeng dataset. (c) Xing dataset.

Figure 6.3: Dependence on the probability of regarding NR-NP as positive (α). The regularization coefficient λ is set to 10^{-2}.

Table 6.6: Ratios of the observable classes for the recommended items in each method.

	Dunnhumby				Xing			
	R-P	NR-P	R-NP	NR-NP	R-P	NR-P	R-NP	NR-NP
RMF	0.151	0.096	0.384	0.369	0.027	0.017	0.115	0.841
BPR	0.144	0.092	0.380	0.384	0.031	0.026	0.122	0.821
ULRMF	0.085	0.013	0.251	0.651	0.022	0.007	0.069	0.902
ULBPR	0.069	0.005	0.289	0.637	0.023	0.008	0.073	0.896

In the experiments, items were filtered by the time periods existed in purchase logs. The filtering criteria were modified from 7 to 19 days by 3-day interval for the Xing dataset, in which the numbers of items varied from 41,099 to 5,828. As shown in Figure 6.2 (b), ULBPR outperforms BPR in all these conditions. Experiments were also conducted with items in product-level instead of category-level for the Dunnhumby dataset, in which the number of items is 4,287. In this condition, *Uplift@10* are 0.0826 and 0.0484 for ULBPR and BPR, respectively. These results indicate that the proposed uplift-based optimization can improve uplift for datasets in wide range of data densities.

ULRMF and ULBPR have a model-specific hyperparameter α, which is the probability of regarding NR-NP as positive. Figure 6.3 shows the dependence on α. The optimal α is less than 1, which supports the claim of treating NR-NP as an intermediate between positive and negative in Subsection 6.3.1.

The proposed optimization methods handle R-P as positive and NR-P as negative, while the accuracy-oriented methods treat both as positive. To see the effect of this difference, the distribution of the recommended items in the observable four classes were investigated (Table 6.6). ULRMF

Table 6.7: Ten items recommended most often by RMF and ULRMF for the Dunnhumby dataset. Numbers in parentheses are popularity ranks from purchase logs. Names of some items are shortened from the original ones.

RMF	ULRMF
FLUID MILK WHITE ONLY(1)	SHELF STABLE MICROWAVE(831)
SOFT DRINKS PK CAN(4)	REFRIGERATED PASTA SAUCE(848)
SHREDDED CHEESE(5)	DRY & SPRAY STARCH(805)
MAINSTREAM WHITE BREAD(3)	JARRED FRUIT(889)
POTATO CHIPS(7)	TEA UNSWEETENED(833)
SFT DRNK 2LITER BTL(6)	NUTS OTHER(829)
BEERALEMALT LIQUORS(11)	INFANT FORMULA TODDLER(863)
100% PURE JUICE ORANGE(8)	DECOR BULBS(687)
TOILET TISSUE(10)	FLUID MILK WHITE ONLY(1)
TORTILLA/NACHO CHIPS(15)	BEEF STEW(638)

and ULBPR successfully reduce the recommendations of the NR-P class, in which items can be purchased without recommendations. The R-NP ratio also decreases, thereby avoiding recommendations that result in no outcome. Further, the sum of R-P and R-NP ratios, which is equal to the ratio of items included in the recommendation logs, is not higher for ULRMF and ULBPR compared to RMF and BPR. This indicates that the proposed optimization methods do not orient a model M for mimicking the recommendation policy of the currently deployed model D.

6.5.4 Trends of the Recommended Items (RQ3)

To intuitively understand the difference in the recommendation outputs between the accuracy-based optimization and uplift-based optimization, Table 6.7 shows the often-recommended items by RMF and ULRMF in the Dunnhumby dataset. While RMF tends to recommend popular items, ULRMF recommends items without an emphasis on popular ones[13]. Often-recommended items by ULRMF include those that might induce impulse purchases such as pasta sauce and heat-and-serve meals.

[13] Average popularity ranks of RMF and ULRMF are 149.6 and 671.4, respectively. Average Jaccard index between recommendation outputs of RMF and ULRMF is 0.0599.

6.6 Conclusions

This study proposed new evaluation and optimization methods for uplift-based recommendation. The illustrative examples demonstrated that accuracy metrics such as precision cannot be utilized to assess recommenders in terms of uplift. Based on a causal inference framework, an offline evaluation protocol to estimate the expected uplift of items in a recommendation list is proposed. Then, the relative priorities of four observation classes are derived from purchase and recommendation logs and their priorities are utilized to construct pointwise and pairwise sampling methods. The experiments using three public datasets confirmed that the proposed optimization methods outperform conventional accuracy-based methods and recent methods targeting uplift. The characteristics of uplift-based optimization and its output recommendations were also investigated.

Because the proposed uplift-based optimizations are generic methods, they can be applied to various recommender models. Recently, recommender models using neural networks have outperformed conventional models [46, 130]. Applying the proposed uplift-based optimizations to neural network models would further enhance the uplift, which is left for future work.

Chapter 7

Context Style Explanation

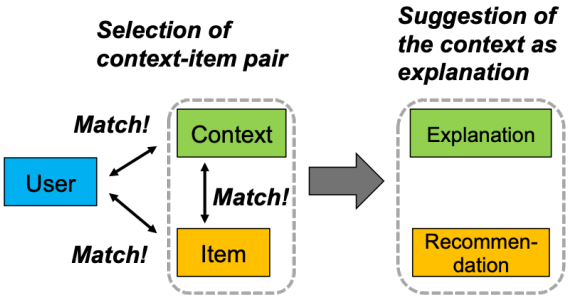


Figure 7.1: Proposed context style explanation.

7.1 Introduction

Recommender systems help users select items from a large number of candidates. Such systems estimate user's preference for items (such as books, movies, and restaurants) from past histories of user's actions (such as purchases, views, and visits), and then present an item that fits the user's preference. Users can then select their favorite items from the recommended items.

Explaining the reason for recommendations further supports user decision-making, as it helps a user understand why the item is recommended. Such an understanding leads to a better decision regarding whether to choose the recommended item. The explanation also invokes the user's interest in the recommended item.

Several explanation styles have been proposed [103, 137, 138, 153]. For example, the neighbor style explanation provides ratings from similar users. The influence style explanation shows items related to those recommended from the user's purchase history. The demographic style explanation describes the user's age and gender. The content style explanation displays item features, such as keywords for books and user-generated tags for movies. These four styles are based on information related to users or items. This is because these factors affect the user's decision-making process [26, 105].

Contextual factors such as time, location, companion, and purpose are also essential elements that affect user's decision-making [24]. Context-aware recommender systems [3, 5] have been developed to model user's choices under various contexts, improving the prediction performance for

items preferred by users.

However, contexts have never been used for the explanation of recommendations. Contexts are entities different from users and items, and previous explanation methods have used information of either users or items. Considering that the user's decisions depend on contexts, explaining recommendations using context will help users.

This study proposes a new explanation style using context (Figure 7.1). The context style explanation indicates the appropriate context for the recommendation. For example, *"This restaurant is recommended to you because it is suitable for dates with your girlfriend/boyfriend"*, where "dates with your girlfriend/boyfriend" is the context presented as an explanation. The selected context should also be related to the user; the user might be more interested in the explanation if she or he is familiar with the context. To generate appropriate context style explanations and item recommendations, context-item pairs are selected for each user by fulfilling three affinities: a) a user-item match, b) an item-context match, and c) a user-context match.

There are two possible effects of context on the explanation.

- **Persuasiveness**: the exhibited context makes users recognize a future consumption situation for the recommended item. Then the recognition of suitable context for usage motivates users to consume the item. of suitable context for usage motivates users to consume items.
- **Usefulness**: users select items based on contexts. Therefore, suggested usage contexts should help user's decision-making.

In this study, the above effects of the context style explanations are investigated from three viewpoints: 1) comparison with other explanation styles, 2) hybridization of context style and other styles, and 3) dependence on gender and age. To investigate these aspects, a restaurant recommender system is implemented with context style explanations, conducting a user study via crowdsourcing.

The contributions of this chapter are summarized as follows.

- This work proposes a novel context style explanation for recommender systems.

- This work verifies the persuasiveness and usefulness of the proposed explanation methods by conducting a user study.
- This work reveals that the personal preference of explanation styles depends on gender and age.

The remainder of this chapter is organized as follows. Related work is presented in the next section. Subsequently, the context style explanation method is described. After that, the experimental details are explained, followed by results and discussion. conclusions are summarized in the last section.

7.2 Related Works

This work is mostly related to two research fields: 1) explanation of recommendation and 2) use of context in recommendation.

7.2.1 Explanation of Recommendation

Explaining recommendations is important for users to understand the reasoning behind them. Explanations have various effects on users [103, 137, 138, 153]. They can help gain users' trust [105] and increase the acceptance of recommendations [47]. They also help users evaluate items accurately [13] and change user's evaluation of items [26].

Various explanation styles have been proposed and evaluated through user studies. Herlocker et al. [47] compared various explanations with different styles and different visualizations, including histograms of user neighbor ratings (i.e., neighbor style); similarity to other items in the user's profile (i.e., influence style); and the user's favorite actor or actress (i.e., content style). Demographic information has been used for explanations in the tourism domain (i.e., demographic style) [7]. Bilgic et al. [13] demonstrated that explanations using keywords (i.e., content style) or showing items influencing recommendations (i.e., influence style) help users evaluate items effectively. Various kinds of contents and ways to visualize them have been explored for content style explana-

tions. User-annotated tags are used for explanations [142] and are displayed in a tagcloud interface [25]. Musto et al. [98] showed that fusing linked open data and choosing specific properties improves explanations. Organizational explanations show the pros and cons of items extracted from user reviews, according to user's priorities [97]. Chen et al. [22] further elaborated the organizational explanation by grouping similar trade-off items. Recent research has endeavored to generate personalized natural language explanations of items [20, 86, 33, 90], which can be regarded as advanced content style explanations. Chang et al. [20] generated explanations via the collaboration of crowdworkers and intelligent systems. State-of-the-art neural network models are also used for that purpose [86, 33, 90].

Although there is a vast amount of research on explaining recommendations, most rely on the four types of information shown in Table 7.1: neighbor, influence, demographic, and content. In addition to these four types, context significantly influences user's decision-making [24]. Zheng [156] and Papadimitriou [103] alluded to the possibility of using contexts for explanations. However, contexts have not yet been used for explaining recommendations. This work is the first study of context style explanations.

Furthermore, several explanation styles can be hybridized [103, 134]. Symeonidis et al. [134] combined content and influence style explanations. The visualization of complex hybrid explanations has also been investigated in [74]. This study investigates the hybrids of context style with other explanation styles.

The preferences of explanation styles might depend on users, which has not been well explored in the previous research. Recently, McInerney et al. used a bandit algorithm to personalize explanation styles [94]. This work also explores the personal preferences of explanation styles. While McInerney et al. focused on model performance, this work clarifies how the preferences differ by gender and age. In addition, this work's investigation of the personal preference includes the proposed context style explanation and hybrids of explanation styles, which were not included in the previous study.

Table 7.1: Overview of the conventional explanation styles and the proposed context style explanation.

Explanation Style	Displayed Information	Example
Neighbor	Ratings or the fact of purchases of similar users	Users similar to you also visit this restaurant.
Influence	Items related to recommended ones from users' past consumption	Recommend for those who also visited *Restaurant C*.
Demographic	Gender, age, profession, etc. of users	Recommend for female students in their 20s.
Content	Content of recommended items, represented by extracted keywords or annotated tags	Recommend for those who like hamburgs.
Context (ours)	Context when users would consume recommended items	Recommend for use in a matchmaking party.

7.2.2 Use of Context in Recommendation

Since users evaluate items differently depending on the context, recommender systems need to recognize the influence of the context [5]. Context-aware item recommendation is a task that involves recommending items suitable for a user in a specific context. The traditional approaches to context-aware recommendations are contextual pre-filtering and post-filtering, in which ratings or items are filtered by relevance to the context either in the initial stage or in the final stage of the recommendation process [3, 102]. Direct modeling of user-item-context relation is called contextual modeling and tends to outperform pre-filtering or post-filtering. Since multi-dimensions of user-item-context can be expressed as a tensor, the direct approach involves utilizing a tensor factorization [67]. However, an exact tensor factorization with Tucker decomposition requires a vast amount of computational resources. Approximation for pairwise interactions can achieve comparable or even better performance [112, 111, 11].

Even if a user has already chosen items, there is room to choose contexts for consuming the items. The notion of recommending contexts to users has recently been investigated [11, 155, 156]. Context recommendation is a task that involves recommending contexts suitable for users and items. Baltrunas et al. [11] collected a dataset of the best usage context for each piece of music and predicted the context using variants of the nearest neighbor technique. Zheng [155] compared several multi-label classification techniques for the same task to recommend contexts conditioned on users and items. Zheng [156] also recommended the context to the user according to the user's

Table 7.2: Comparison of task settings.

Task	Input	Output
Context-aware item recommendation	User + Context	Item
Context recommendation	User (+ Item)	Context
Context style explanation (ours)	User	Context + Item

preference for the context.

To generate context style explanations, context-item pairs are selected for each user. The differences between the task for context style explanations and the conventional tasks are summarized in Table 7.2. Context-aware recommendation generates lists of recommended items for a specified user and context. Contexts are pre-selected, either explicitly (e.g., users input purpose of travel into a hotel booking site) or implicitly (e.g., current place and activity can be estimated from wearable sensors). There are two kinds of tasks for context recommendation: recommending context for specified user-item pair [11, 155], and recommending context for a user [156]. In the former task, items to be consumed in recommended contexts are fixed. In the latter task, items are irrelevant to context recommendations. In the case of the proposed context style explanation, pairs of items and contexts are provided for each user. Both contexts and items are undetermined. Therefore, the context style explanation is different in terms of the recommendation task setting. This difference is addressed by modifying negative sampling in model training. The main focus of this study is to evaluate the impacts of the context style explanation, and the performance improvement of the above task remains for future researches.

7.3 Context Style Explanation

The generation of context style explanations involves two steps: (1) selection of context-item pairs for users, and (2) suggestion of the context of a context-item pair as the item's explanation.

7.3.1 Selection of Context-Item Pairs

Our context style explanation suggests contexts that the user might encounter in the future. This means both the context and the item are unknown in the proposed task, while the context is predetermined in context-aware item recommendation. In this case, the recommender needs to select appropriate pairs of contexts and items for the users. This requires three affinities: a) a user-item matching, b) an item-context matching, and c) a user-context matching. For a restaurant recommendation, the recommended restaurant should match the user's preferences, just as with non-contextualized recommender systems (user-item match). Moreover, the recommended restaurant should match the suggested context (item-context match). If the context of eating with children is suggested in an explanation, then the recommended restaurant should be suitable for that situation. Additionally, the recommended context should be the one anticipated by the user (user-context match). If the user does not have children and lacks many opportunities to eat out with children, a suggestion of eating out with children would likely be inappropriate.

These above three affinities can be learned via the latent representation of pairwise interactions among user, item, and context features [11, 62, 111]. This work uses field-aware factorization machines (FFMs) [62] for their efficiency and performance. The FFM splits features to "fields," and incorporates interaction effects among the features of different fields. The FFM is formulated as,

$$\hat{y} = \sum_{j_1=1}^{n} \sum_{j_2=j_1+1}^{n} (\boldsymbol{w}_{j_1}^{f_2} \cdot \boldsymbol{w}_{j_2}^{f_1}) x_{j_1} x_{j_2}, \tag{7.1}$$

where $\hat{y} \in \mathbf{R}$ is a prediction by the FFM; $\boldsymbol{w}_j^f \in \mathbf{R}^{\mathbf{d}}$ is a d-dimensional latent vector of a feature j that interacts with a field f; and $x_j \in \mathbf{R}$ is the value of the feature j.

To model interaction among users, items, and contexts, the proposed method prepares a user field, an item field, and a context field. Example features in the user field are gender and age. The latent factors of a female user u in her 30s are expressed as,

$$\boldsymbol{w}_u^{\text{Item}} = \boldsymbol{w}_{female}^{\text{Item}} + \boldsymbol{w}_{30s}^{\text{Item}}, \tag{7.2}$$

$$\boldsymbol{w}_u^{\text{Context}} = \boldsymbol{w}_{female}^{\text{Context}} + \boldsymbol{w}_{30s}^{\text{Context}}. \tag{7.3}$$

Features in the item field can be restaurant genres or places. The latent factors of $IndianFood$ restaurant i located in $PlaceA$ are composed as,

$$\boldsymbol{w}_i^{\text{User}} = \boldsymbol{w}_{IndianFood}^{\text{User}} + \boldsymbol{w}_{PlaceA}^{\text{User}}, \tag{7.4}$$

$$\boldsymbol{w}_i^{\text{Context}} = \boldsymbol{w}_{IndianFood}^{\text{Context}} + \boldsymbol{w}_{PlaceA}^{\text{Context}}. \tag{7.5}$$

Similarly for context c of $BusinessEntertaining$,

$$\boldsymbol{w}_c^{\text{User}} = \boldsymbol{w}_{BusinessEntertaining}^{\text{User}}, \tag{7.6}$$

$$\boldsymbol{w}_c^{\text{Item}} = \boldsymbol{w}_{BusinessEntertaining}^{\text{Item}}. \tag{7.7}$$

Then, Equation (7.1) is expressed as follows,

$$\hat{y} = \boldsymbol{w}_u^{\text{Item}} \cdot \boldsymbol{w}_i^{\text{User}} + \boldsymbol{w}_i^{\text{Context}} \cdot \boldsymbol{w}_c^{\text{Item}} + \boldsymbol{w}_u^{\text{Context}} \cdot \boldsymbol{w}_c^{\text{User}}. \tag{7.8}$$

Each term in Equation (7.8) represents a) a user-item match ($\boldsymbol{w}_u^{\text{Item}} \cdot \boldsymbol{w}_i^{\text{User}}$), b) an item-context match ($\boldsymbol{w}_i^{\text{Context}} \cdot \boldsymbol{w}_c^{\text{Item}}$), and c) a user-context match ($\boldsymbol{w}_u^{\text{Context}} \cdot \boldsymbol{w}_c^{\text{User}}$). Latent factors can be learned using the user's past interactions. More specifically, if a user u consumed an item i under a context c, a triplet (u, i, c) is assigned a positive label. Triplets that have not appeared in past consumption logs are assigned negative labels. The FFM is trained using these positive and negative samples.

Note that the definition of negative samples depends on the task settings in Table 7.2. In the conventional context-aware item recommendation, negative samples are defined for each pair of user-context existing in the logs. Items not consumed under a user-context pair are negative samples. In this case, negative samples are defined for each user. Item-context pairs that have not appeared in the user's logs are negative samples. While the former negative samples include

contexts only experienced by the user, the latter negative samples include contexts not experienced by the user. This enables to learn the affinity between users and contexts.

After the training, the best context-item pairs are selected by the score of Equation (7.8).

7.3.2 Suggestion of a Context as Explanation

Selected context-item pairs are used to produce recommendations and explanations. If there is a context-item pair with context c and item i, then item i is presented to a user as a recommendation and context c is used for an explanation. An explanation is generated using human-crafted templates, for example, *"item i is recommended to you because it is suitable for context c."* In case there are several contexts suitable for the recommended item, then the multiple contexts can be displayed together. This study only uses one best context for an explanation for experimental simplicity.

7.4 Experiment

First, users' restaurant visit logs were collected with context via crowdsourcing. Second, a context-item pair selector was trained using the acquired logs and prepared recommendations and explanations. Finally, the same users were asked to evaluate explanation styles.

7.4.1 Collecting Dataset

Restaurant visit logs were collected using a Japanese crowdsourcing platform. For each restaurant, there are three entries: name of a visited restaurant, the URL to the restaurant within a restaurant information site, and the usage scene of the visit (i.e., context). Each crowdworker is asked to input a maximum of 20 restaurants. To limit the area of the visit log, the experiment recruited cloud workers living in specific urban areas[1]. The questionnaire asked for original contexts of users' visits instead of asking for evaluation under a provided context because users behave differently under supposed contexts and real contexts [8, 99]. Usage scenes were selected from 15 options,

[1]The urban areas are Tokyo and Kanagawa, the Japanese capital and a neighboring prefecture of Tokyo, respectively.

Table 7.3: Candidates of 15 usage scenes (contexts) and counts selected by crowdworkers. The crowdworkers chose one context for each visit. The usage scenes were shown to the crowdworkers in the same order as this list. If crowdworkers thought that more than one scene can be associated with the visit, then they were advised to select the uppermost scene on the list.

Usage scene	Count
Matchmaking party	20
Girls' lunch or night out	184
Business entertaining	39
Banquet or drinking party in a large group	15
With children or grandchildren	163
With parents, sisters, or brothers	212
With a husband or wife	414
Dating with opposite gender	275
With close friends (only eating)	284
With close friends (with drink)	325
With colleagues or acquaintances (only eating)	191
With colleagues or acquaintances (with drink)	108
In solitude	386
Take-away	73
None of the above	16

Table 7.4: Statistics of collected dataset via crowdsourcing.

data	numbers
total visits	2,884
unique users	155
unique items (restaurants)	2,730
unique contexts (usage scenes)	15
genres of restaurants	210
nearest stations of restaurants	473

as described in Table 7.3. As for usage scenarios, the explanations of usage scenarios on several restaurant information sites were referred to make it easy for users to understand. If crowdworkers thought that more than one scene can be associated with the visit, then they were advised to select the uppermost scene on the list. The numbers of times each context were chosen by crowdworkers are also shown in Table 7.3. Most contexts obtained substantial votes and *"none of the above"* received only a small portion of the votes. This supports the validity of the context candidate design. If the context candidates did not include appropriate contexts for users, the votes for *"None of the above"* would have been high.

2,884 visit logs from 155 crowdworkers were obtained after removing logs with improper URLs and crowdworkers who provided improper URLs more than half the time. Careless crowdworkers who input improper URLs were removed. There are 2,730 unique URLs in the remaining visit logs. The statistics of the collected dataset are summarized in Table 7.4. The genders and approximate ages of the crowdworkers were provided from the crowdsourcing platform. Among the 155 crowdworkers, 108 were female and 47 were male. Further, 44 crowdworkers were in their 20s, 39 in their 30s, 26 in their 40s, 13 in their 50s, 2 in their 60s, and the ages of 31 were unknown. The average number of visits per restaurant was 1.056 and the sparsity was 99.32%. The obtained URLs were crawled and the restaurant's content information, including genres and nearest stations[2], was collected. Note that each restaurant is assigned multiple genres (2.3 genres on average). There are 210 unique genres and 473 unique stations.

7.4.2 Training the Recommender and Preparing Explanations

The context-item selector was trained using the collected dataset. The libffm library[3] was used for the FFM. The features of the user field are user ID, gender, and age. The features of the item field are genre and nearest station. Using the demographic features of users and the content features of items alleviated the issue of data sparsity. The features of the context field included context ID, which is assigned to 15 usage scenes.

The training of the recommender proceeded as follows. First, the dataset was randomly split into 80% training and 20% validation data. Next, hyper-parameters of the FFM were then optimized to maximize the AUC (areas under the curve) of the validation data. The following hyper-parameters were chosen: learning rate 0.05, regularization coefficient 0.0005, and dimensions of factor 100. The obtained AUC with these hyper-parameters was 0.865. Finally, the model was trained using the entire dataset to select context-item pairs.

After training the model, seven best context-item pairs were selected for each user, according to the score of Equation (7.8). Then, the order of the selected context-item pairs was shuffled,

[2]Restaurants are located in urban areas where public transportation is well developed.
[3]https://github.com/guestwalk/libffm.

Table 7.5: Samples of seven explanation styles. Phrases emphasized in italics are tailored to fit users, recommended items, and supposed contexts.

Style	Sample
Non-specific	Recommend based on your visit logs
Demographic	Recommend for *"women in their 30s"*
Content	Recommend for those who often visit *"Italian restaurants"*
Context	Recommend for use *"with husband or wife"*
Demographic + Context	Recommend for use *"in business entertaining"* of *"men in their 50s"*
Content + Context	Recommend for use *"in solitude"* for those who often visit *"noodle shops"*
Demographic + Content + Context	Recommend for use *"with close friends (with drink)"* of *"women in their 20s"* who often visit *"cafes"*

to ensure that recommendation quality did not correlate to the presentation order. Restaurants visited in the past were removed from the list, whereas contexts experienced in the past were not omitted. The same restaurant was recommended only once per user.

Seven explanation styles were prepared as described in Table 7.5. The non-specific explanation did not include any specific information regarding demographics, contents, and contexts. This explanation was the same for all users and all recommended items. For the context style explanation, the context of context-item pair was directly assigned for the explanation. For the demographic style explanation, user age and gender were used for the explanation. Recommended items related to the user age and gender, because the recommender incorporates them as user features. The content style explanation used a genre common among the recommended restaurants and those that the user visited in the past. The hybrid explanation styles are generated via combinations of the above steps.

Table 7.6: Demographics of participants (crowdworkers) in the evaluation of the explanation styles. The participants were recruited from the respondents of the initial data collection; this is necessary for personalized recommendations and explanations to the participants.

Age	Female	Male	Total
20s	10	9	19
30s	20	4	24
40s	14	2	16
50s	4	2	6
60s	1	0	1
unknown	15	4	19
Total	64	21	85

7.4.3 Evaluating Explanation Styles

The crowdsourcing recruited the 155 crowdworkers who had appropriately submitted restaurant visit logs, and 85 participated in the user study. The participants' demographics are shown in Table 7.6. The experiment presented seven restaurant recommendations with seven different explanation styles to each user. Each recommendation was generated by the same FFM model.

The order of the explanation styles was randomly shuffled among users in order to cancel any biases related to the display order. The questionnaire asked the following four evaluation questions using a 7-point Likert scale for each pair of restaurant recommendations and explanations.

- Persuasiveness 1 (P1): The explanation is convincing.
- Persuasiveness 2 (P2): The explanation triggers interest.
- Usefulness 1 (U1): The explanation is useful for choice.
- Usefulness 2 (U2): The explanation is easy to understand.

In addition to these evaluation questions, the questionnaire asked whether the participants visited the recommended restaurants in the past and whether they knew of them in advance. There were also free entry fields to express any other comments.

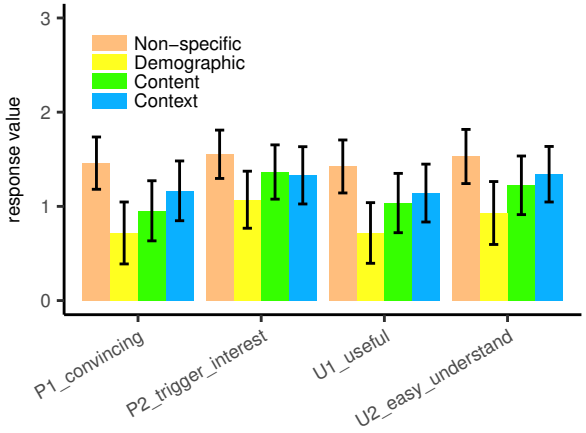


Figure 7.2: Responses to four evaluation questions for four single explanation styles. Error bars represent 95% confidence intervals of average response values.

7.5 Results and Discussion

7.5.1 Quantitative Analysis

Among the restaurants recommended to the participants, 21% were visited in the past and 20% were known in advance. Note that recommended restaurants were those not visited by each user in the collected dataset. This indicates that the item recommendation was fairly accurate and that the recommender system works fine.

This subsection investigates the persuasiveness and usefulness of context style explanations with the following three aspects: 1) comparison with other explanation styles, 2) hybridization of context style and other styles, and 3) dependence on gender and age.

Comparison with other explanation styles

The experiment compared four single explanation styles: non-specific, demographic, content, and context styles. Responses to the four questions are shown in Figure 7.2. Responses ranged from strongly disagree (-3) to strongly agree (+3). The average response for the context style explanation was higher than that for the demographic style ($p = 0.008$, 0.10, 0.047, and 0.036 for P1, P2, U1,

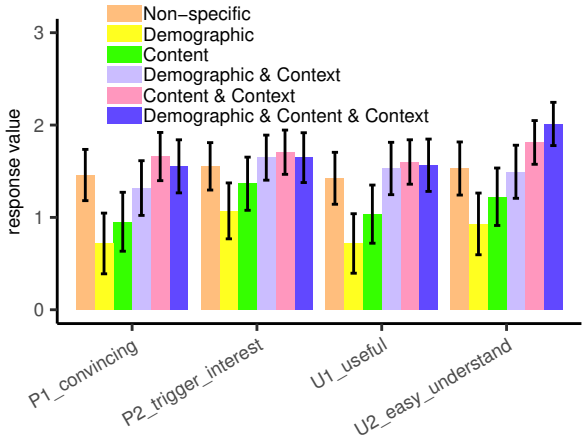
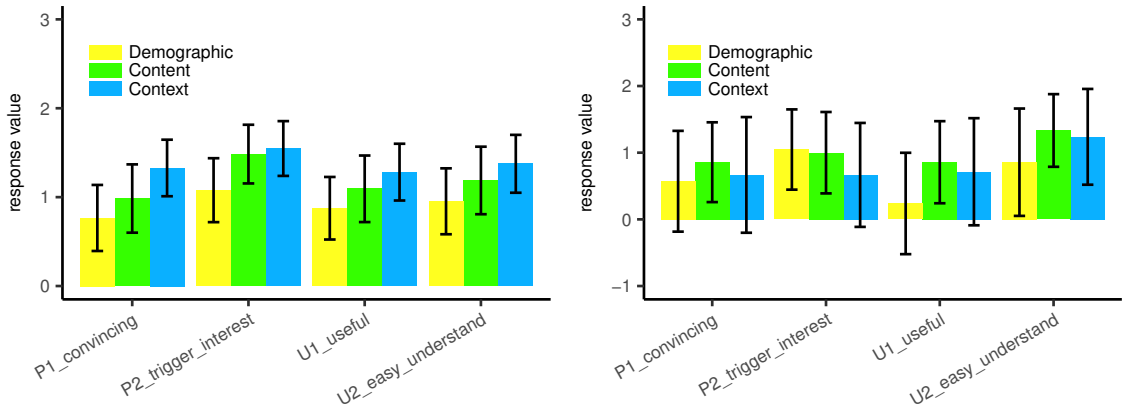


Figure 7.3: Comparison between single and hybrid explanation styles. Error bars represent 95% confidence intervals of average response values.

and U2, respectively, via the Wilcoxon signed-rank test). The average response for the context style explanation also tended to be higher than that for the content style, though not statistically significant. The non-specific explanations tended to perform better than other single styles; the possible reasons are discussed in the later section.

Hybridization of context style and other styles.

The context styles were combined with other styles. Figure 7.3 shows the comparison between single and hybrid explanation styles. The combination of the demographic and context styles outperformed the demographic-only style ($p < 0.01$ for all questions), and the combination of the content and context styles outperformed the content-only style ($p = 0.061$ for P2, and $p < 0.01$ for others). The combination of content and context styles also tended to outperform non-specific style (though not statistically significant). The triple combination of demographic, content, and context styles did show better performance compared to the dual combination of demographic and context styles.



(a) Female users. (b) Male users.

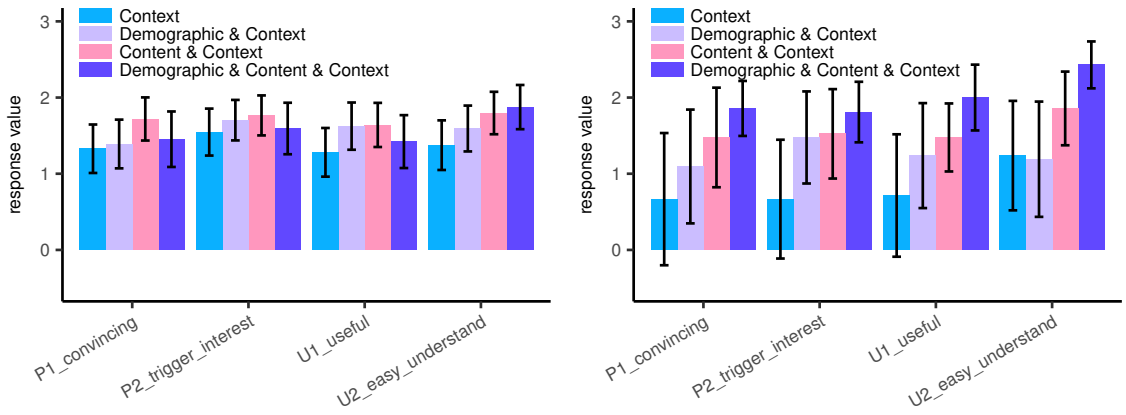
Figure 7.4: Responses for single explanation styles by gender.

Dependence on gender and age.

The preference of explanation styles might depend on users. The experiment investigated the difference in the response for each explanation style in terms of gender and age.

First, the difference by gender is described. Figures 7.4 and 7.5 show the comparisons of single explanation styles and hybrid explanation styles, respectively, for female and male users. While female users tended to prefer the context style over the content style, male users tended to prefer the content style over the context style (Figure 7.4). In terms of hybrids (Figure 7.5), male users preferred the triple combination more than the dual combinations (not significant for P1 and P2, $p = 0.062$ for U1 and $p = 0.020$ for U2 with comparison of Demographic & Content & Context vs. Content & Context).

Next, the difference by age is investigated. Figures 7.6 and 7.7 show the comparisons of single explanation styles and hybrid explanation styles, respectively, for young to elder users. While young users preferred the content style over the context style, middle-age users tended to prefer the context style over the content style (Figure 7.6). In terms of hybrids (Figure 7.7), there was no clear difference by age.



(a) Female users.

(b) Male users.

Figure 7.5: Comparison between single and hybrid explanation styles by gender.

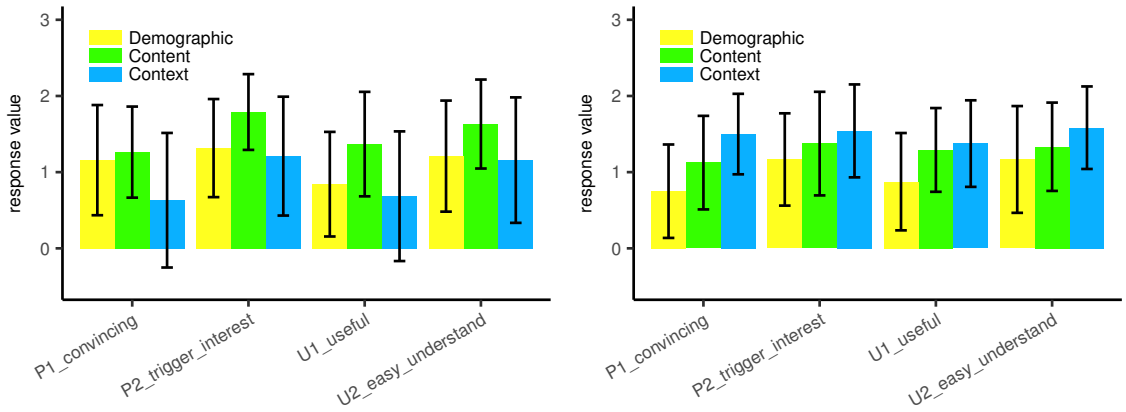
7.5.2 Qualitative Analysis

Sixty-four participants input at least one comment and 293 comments were obtained in total. To further understand the user perception of the context style explanation, these comments were investigated.

User comments indicated two reasons of persuasiveness:

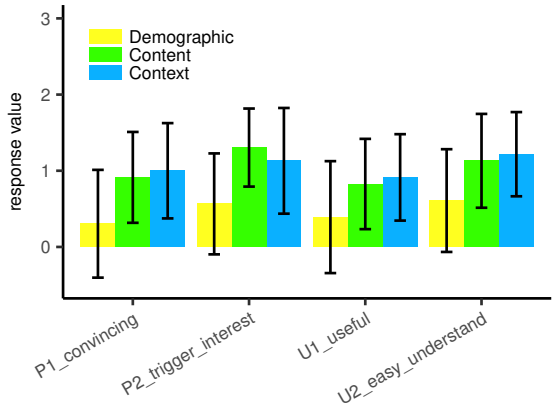
1. Relevance of the proposed context to users:
   - *"Under my current environment, it's a very interesting recommendation, so I became to feel like going."*
   - *"I think I want to go because this situation is probable for me."*
2. Recognition of appropriate context for usage:
   - *"I think I am going to use this when I organize a drinking party."*
   - *"I have been interested in Japanese rice wine bars, but few of my close friends like it. Visiting here with my colleagues sounds nice."*

Users also mentioned the usefulness of the context for decision-making.



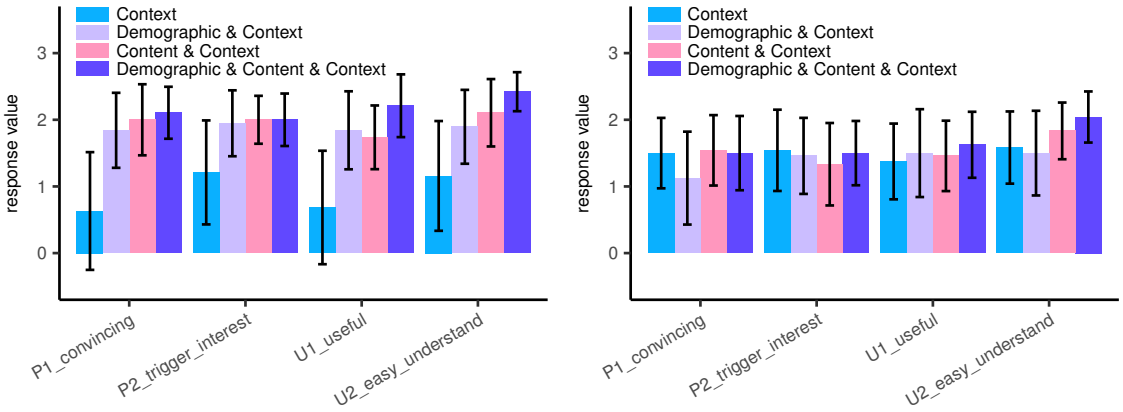
(a) Young users (20s).

(b) Middle-age users (30s).



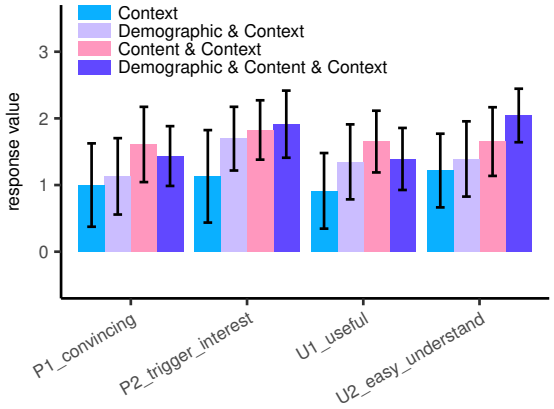
(c) Elder users (40s, 50s, 60s).

Figure 7.6: Responses for single explanation styles by age.



(a) Young users (20s).

(b) Middle-age users (30s).



(c) Elder users (40s, 50s, 60s).

Figure 7.7: Comparison between single and hybrid explanation styles by age.

- *"I'm afraid of making a wrong choice for a girls' night out, so this explanation is useful."*

These findings from the qualitative analysis support the importance of context for the explanation.

On the other hand, there were three kinds of negative responses to the context style explanations.

1. Mismatch of context and restaurants:
    - *"This restaurant is a standing bar, which is not suitable for dating."*
    - *"I don't think it's a good idea to use a buffet restaurant for a banquet."*
2. Context is irrelevant for someone's choice:
    - *"I choose restaurants by whether I like the menu or not."*
    - *"I eat out only with my close friends, so information about situation is useless."*
3. Needs for finer granularity of context:
    - *"You mention just dating, but is it referring to ordinary dating or anniversary dating?"*

7.5.3 Discussion

The non-specific explanation tended to perform better than the context style, and the hybrid of content and context styles tended to perform better than the non-specific explanation (Figures 7.2 and 7.3). Relatively high appraisals of the non-specific explanation might be a result of familiarity with the explanation style. Some users commented as follows: *"There is a comfort in this type of explanation,"* and *"This writing style suits me the best."* The used crowdsourcing platform provides task recommendation for users with explanations of this style: *"Recommendation is based on your past task."* Another reason might be the occasional mismatch of the presented context, as seen in the example comments indicating the mismatch shown in the previous section. User evaluation tends to be affected more by negative experiences (i.e., mismatches) than by positive experiences (i.e., good matches). Similar observations were reported in an experiment of personalizing engaging messages [69]. Improving the accuracy of context-item pair selections is future work. As described

in Subsection 7.2.2, the selection of context-item pairs is an unexplored new task setting. Thus, there should be much room for improvement.

The trio of demographics, contents, and contexts did not produce a significant improvement over the duo of contents and contexts. Users may have felt excessive complexity. Determining adequate amounts of information for an explanation would be an interesting challenge.

As for gender dependence in the preferences of explanation styles, female users tended to prefer the context style and male users tended to prefer the content style (Figure 7.4). This might be a result of the difference in personality traits; it is known that women are more tender-minded than men [35]. The used contexts in this study included accompanying persons, which is important information for tender-minded people. Regarding age dependence in the preferences of explanation styles, users in their 20s tended to prefer the content style and users in their 30s tended to prefer the context style (Figure 7.6). Tender-mindedness correlates positively with ages [93]; this might be the reason for the age dependence. However, the preference for the context style does not increase for users over the age of 40 years compared to users in their 30s.

This work was conducted in a restaurant recommendation domain. Context is important for recommendations in various domains such as movie, travel, and music [5]. Hence, context style explanations would be applied to various domains, though relevant contexts should be unique to those different domains. Future research should investigate those other domains. Further, the experiments compared and hybridized context style explanations with demographic and content styles. Future studies should experiment with other conventional explanation styles (e.g., neighbor and influence styles). Besides, this work evaluated persuasiveness and usefulness to verify the hypothesis of the effects of the context-style. Evaluation of other factors, such as user trust and decision efficiency, should be conducted in future work.

7.6 Conclusions

This chapter proposed the context style explanation for recommenders. The crowdsourcing-based user study was conducted to measure persuasiveness and usefulness. The context style explanation

was better than the demographic style. The context style also tended to show better performance than the content style, but the difference was not statistically significant. It was further confirmed that the hybrids of the context style and other explanation styles improve persuasiveness and usefulness. Findings from user comments support the importance of contexts for explanations. Furthermore, the personal preferences of explanation styles in terms of gender and age were revealed. While female or middle-age users tended to prefer the context style over the content style, male or young users tended to prefer the content style.

Chapter 8

Conclusions

This thesis aimed to increase the causal effect of recommendations by resolving three issues: 1) incomplete modeling, 2) causality of recommendation, and 3) users' reluctance. These issues were addressed by applying the following three methods: 1) modeling recommendation influences (Chapters 3, 4, and 5), 2) uplift optimization (Chapters 4 and 6), and 3) persuasive explanation (Chapter 7).

- Solution 1: modeling recommendation influences. This thesis models recommendation influences in various ways. A recommendation model that incorporated personal discount sensitivity was proposed and studied (Chapter 3). As far as we know, this is the first study that unified item preference and discount sensitivity into a single purchase prediction model. A recommendation model that incorporated personal recommendation responsiveness was proposed and investigated (Chapter 4). The individualized difference of recommendation influences is original to this study. Further, the cold-start problem of recommendation logs is newly addressed in this study. An extension of exposure modeling to include both direct and indirect recommendation influences on exposure was proposed and discussed (Chapter 5).

- Solution 2: uplift optimization. There were two approaches to uplift optimization. The first approach was to design a purchase prediction model that incorporated recommendation influences and then estimate uplift by the difference between predicted purchase probabilities with and without recommendations. Chapter 4 employed this approach for uplift optimization. The second approach was to directly optimize a model toward uplift by deriving positive and negative training samples for uplift. Chapter 6 proposed unique optimization methods based on this approach. The proposed optimization methods are generic and applicable to most machine-learning-based recommendation models.

- Solution 3: persuasive explanation. Users consume items under a specific context. Therefore, it is expected that envisioning the context of usage may motivate users to take action. This thesis proposed to explain recommendations using contexts and verified the assumption that the proposed context style explanation is persuasive (Chapter 7). Previous explanation styles were based on user or item information and this is the first study to apply context to explain

recommendations.

This thesis provides foundations for increasing the causal effect of recommendations. The proposed uplift-based evaluation can be extended in various ways, for example, weighting items upper in the list like NDCG or considering continuous outcomes, such as the prices of purchased items. Uplift-based optimization can be applied to most recommendation models including recent neural recommenders. To motivate users to take action, currently deployed explanation of recommenders can be replaced or combined with the new context style explanation. Besides, the context style explanation also introduces a new task setting for recommendation algorithms, that is, selecting context-item pairs for each user that leaves much room for improvement.

Bibliography

[1] Adamopoulos, P., and Tuzhilin, A. The business value of recommendations: A privacy-preserving econometric analysis. In *36th International Conference on Information Systems: ICIS 2015* (2015), Association for Information Systems.

[2] Adomavicius, G., and Kwon, Y. Multi-criteria recommender systems. In *Recommender Systems Handbook*. Springer, 2015, pp. 847–880.

[3] Adomavicius, G., Sankaranarayanan, R., Sen, S., and Tuzhilin, A. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst. 23*, 1 (Jan. 2005), 103–145.

[4] Adomavicius, G., and Tuzhilin, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge & Data Engineering*, 6 (2005), 734–749.

[5] Adomavicius, G., and Tuzhilin, A. *Context-aware recommender systems*. Springer US, 1 2015, pp. 191–226.

[6] Ailawadi, K. L., Beauchamp, J. P., Donthu, N., Gauri, D. K., and Shankar, V. Communication and promotion decisions in retailing: a review and directions for future research. *Journal of retailing 85*, 1 (2009), 42–55.

[7] Ardissono, L., Goy, A., Petrone, G., Segnan, M., and Torasso, P. Intrigue: personalized recommendation of tourist attractions for desktop and hand held devices. *Applied artificial intelligence 17*, 8-9 (2003), 687–714.

[8] Asoh, H., Motomura, Y., and Ono, C. An analysis of differences between preferences in real and supposed contexts. *Knowledge Creation Diffusion Utilization* (2010).

[9] Azaria, A., Hassidim, A., Kraus, S., Eshkol, A., Weintraub, O., and Netanely, I. Movie recommender system for profit maximization. In *Proceedings of the 7th ACM Conference on Recommender Systems* (New York, NY, USA, 2013), RecSys '13, ACM, pp. 121–128.

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

[11] Baltrunas, L., Ludwig, B., and Ricci, F. Matrix factorization techniques for context aware recommendation. In *Proceedings of the Fifth ACM Conference on Recommender Systems* (New York, NY, USA, 2011), RecSys '11, ACM, pp. 301–304.

[12] Belluf, T., Xavier, L., and Giglio, R. Case study on the business value impact of personalized recommendations on a large online retailer. In *Proceedings of the Sixth ACM Conference on Recommender Systems* (New York, NY, USA, 2012), RecSys '12, ACM, pp. 277–280.

[13] Bilgic, M., and Mooney, R. J. Explaining recommendations: Satisfaction vs. promotion. In *Beyond Personalization Workshop, IUI* (2005), vol. 5, p. 153.

[14] Blattberg, R. C., Briesch, R., and Fox, E. J. How promotions work. *Marketing science 14*, 3_supplement (1995), G122–G132.

[15] Bodapati, A. V. Recommendation systems with purchase data. *Journal of marketing research 45*, 1 (2008), 77–93.

[16] Bonner, S., and Vasile, F. Causal embeddings for recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems* (2018), RecSys '18, ACM, pp. 104–112.

[17] Cantador, I., Fernández-Tobías, I., and Bellogín, A. Relating personality types with user preferences in multiple entertainment domains. In *CEUR workshop proceedings* (2013), Shlomo Berkovsky.

[18] Caselles-Dupré, H., Lesaint, F., and Royo-Letelier, J. Word2vec applied to recommendation: Hyperparameters matter. In *Proceedings of the 12th ACM Conference on Recommender Systems* (New York, NY, USA, 2018), RecSys '18, ACM, pp. 352–356.

[19] Castells, P., Hurley, N. J., and Vargas, S. Novelty and diversity in recommender systems. In *Recommender Systems Handbook*. Springer, 2015, pp. 881–918.

[20] Chang, S., Harper, F. M., and Terveen, L. G. Crowd-based personalized natural language explanations for recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems* (New York, NY, USA, 2016), RecSys '16, ACM, pp. 175–182.

[21] Chen, J., Feng, Y., Ester, M., Zhou, S., Chen, C., and Wang, C. Modeling users' exposure with social knowledge influence and consumption influence for recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (2018), CIKM '18, ACM, pp. 953–962.

[22] Chen, L., and Wang, F. Explaining recommendations based on feature sentiments in product reviews. In *Proceedings of the 22Nd International Conference on Intelligent User Interfaces* (New York, NY, USA, 2017), IUI '17, ACM, pp. 17–28.

[23] Chen, M., Beutel, A., Covington, P., Jain, S., Belletti, F., and Chi, E. H. Top-k off-policy correction for a reinforce recommender system. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* (2019), WSDM '19, ACM, pp. 456–464.

[24] Chen, W., Hoyle, C., and Wassenaar, H. J. A choice modeling approach for usage context-based design. In *Decision-Based Design*. Springer, 2013, pp. 255–285.

[25] Chen, W., Hsu, W., and Lee, M. L. Tagcloud-based explanation with feedback for recommender systems. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2013), SIGIR '13, ACM, pp. 945–948.

[26] Cosley, D., Lam, S. K., Albert, I., Konstan, J. A., and Riedl, J. Is seeing believing?: How recommender system interfaces affect users' opinions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2003), CHI '03, ACM, pp. 585–592.

[27] Cremonesi, P., Garzotto, F., Negro, S., Papadopoulos, A. V., and Turrin, R. Looking for "good" recommendations: A comparative evaluation of recommender systems. In *Proceedings of the 13th IFIP TC 13 International Conference on Human-computer Interaction - Volume Part III* (Berlin, Heidelberg, 2011), INTERACT'11, Springer-Verlag, pp. 152–168.

[28] Das, A. S., Datar, M., Garg, A., and Rajaram, S. Google news personalization: Scalable online collaborative filtering. In *Proceedings of the 16th International Conference on World Wide Web* (New York, NY, USA, 2007), WWW '07, ACM, pp. 271–280.

[29] de Gemmis, M., Lops, P., Musto, C., Narducci, F., and Semeraro, G. *Semantics-Aware Content-Based Recommender Systems*. Springer US, Boston, MA, 2015, pp. 119–159.

[30] Devriendt, F., Moldovan, D., and Verbeke, W. A literature survey and experimental evaluation of the state-of-the-art in uplift modeling: A stepping stone toward the development of prescriptive analytics. *Big data 6*, 1 (2018), 13–41.

[31] Di Noia, T., Ostuni, V. C., Rosati, J., Tomeo, P., and Di Sciascio, E. An analysis of users' propensity toward diversity in recommendations. In *Proceedings of the 8th ACM Conference on Recommender Systems* (New York, NY, USA, 2014), RecSys '14, ACM, pp. 285–288.

[32] Dias, M. B., Locher, D., Li, M., El-Deredy, W., and Lisboa, P. J. The value of personalised recommender systems to e-business: A case study. In *Proceedings of the 2008*

ACM Conference on Recommender Systems (New York, NY, USA, 2008), RecSys '08, ACM, pp. 291–294.

[33] Dong, L., Huang, S., Wei, F., Lapata, M., Zhou, M., and Xu, K. Learning to generate product reviews from attributes. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers* (2017), vol. 1, pp. 623–632.

[34] Elahi, M., Braunhofer, M., Ricci, F., and Tkalcic, M. Personality-based active learning for collaborative filtering recommender systems. In *Congress of the Italian Association for Artificial Intelligence* (2013), Springer, pp. 360–371.

[35] Feingold, A. Gender differences in personality: A meta-analysis. *Psychological bulletin 116*, 3 (1994), 429.

[36] Garcin, F., Faltings, B., Donatsch, O., Alazzawi, A., Bruttin, C., and Huber, A. Offline and online evaluation of news recommender systems at swissinfo.ch. In *Proceedings of the 8th ACM Conference on Recommender Systems* (New York, NY, USA, 2014), RecSys '14, ACM, pp. 169–176.

[37] Gilotte, A., Calauzènes, C., Nedelec, T., Abraham, A., and Dollé, S. Offline a/b testing for recommender systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (New York, NY, USA, 2018), WSDM '18, ACM, pp. 198–206.

[38] Gomez-Uribe, C. A., and Hunt, N. The netflix recommender system: Algorithms, business value, and innovation. *ACM Trans. Manage. Inf. Syst. 6*, 4 (Dec. 2015), 13:1–13:19.

[39] Gopalan, P., Hofman, J. M., and Blei, D. M. Scalable recommendation with hierarchical poisson factorization. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence* (2015), UAI'15, AUAI Press, pp. 326–335.

[40] Grbovic, M., Radosavljevic, V., Djuric, N., Bhamidipati, N., Savla, J., Bhagwan, V., and Sharp, D. E-commerce in your inbox: Product recommendations at scale. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2015), KDD '15, ACM, pp. 1809–1818.

[41] Gruson, A., Chandar, P., Charbuillet, C., McInerney, J., Hansen, S., Tardieu, D., and Carterette, B. Offline evaluation to make decisions about playlistrecommendation algorithms. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* (2019), WSDM '19, ACM, pp. 420–428.

[42] Gu, W., Dong, S., and Zeng, Z. Increasing recommended effectiveness with markov chains and purchase intervals. *Neural Comput. Appl. 25*, 5 (Oct. 2014), 1153–1162.

[43] Gunawardana, A., and Shani, G. A survey of accuracy evaluation metrics of recommendation tasks. *Journal of Machine Learning Research 10*, Dec (2009), 2935–2962.

[44] Gunawardana, A., and Shani, G. Evaluating recommender systems. In *Recommender systems handbook*. Springer, 2015, pp. 265–308.

[45] Hansotia, B., and Rukstales, B. Incremental value modeling. *Journal of Interactive Marketing 16*, 3 (2002), 35–46.

[46] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T.-S. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web* (Republic and Canton of Geneva, Switzerland, 2017), WWW '17, International World Wide Web Conferences Steering Committee, pp. 173–182.

[47] Herlocker, J. L., Konstan, J. A., and Riedl, J. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work* (New York, NY, USA, 2000), CSCW '00, ACM, pp. 241–250.

[48] Hijikata, Y., Shimizu, T., and Nishida, S. Discovery-oriented collaborative filtering for improving user satisfaction. In *Proceedings of the 14th International Conference on Intelligent User Interfaces* (New York, NY, USA, 2009), IUI '09, ACM, pp. 67–76.

[49] Hsu, C.-N., Chung, H.-H., and Huang, H.-S. Mining skewed and sparse transaction data for personalized shopping recommendation. *Mach. Learn. 57*, 1-2 (Oct. 2004), 35–59.

[50] Hu, R., and Pu, P. Enhancing collaborative filtering systems with personality information. In *Proceedings of the Fifth ACM Conference on Recommender Systems* (New York, NY, USA, 2011), RecSys '11, ACM, pp. 197–204.

[51] Hu, Y., Koren, Y., and Volinsky, C. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on* (2008), IEEE, pp. 263–272.

[52] Imbens, G. W., and Rubin, D. B. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction.* Cambridge University Press, 2015.

[53] Iwata, T., and Sawada, H. Topic model for analyzing purchase data with price information. *Data Min. Knowl. Discov. 26*, 3 (May 2013), 559–573.

[54] Jagerman, R., Markov, I., and de Rijke, M. When people change their mind: Off-policy evaluation in non-stationary recommendation environments. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* (2019), WSDM '19, ACM, pp. 447–455.

[55] Jannach, D., and Hegelich, K. A case study on the effectiveness of recommendations in the mobile internet. In *Proceedings of the Third ACM Conference on Recommender Systems* (New York, NY, USA, 2009), RecSys '09, ACM, pp. 205–208.

[56] Jannach, D., and Jugovac, M. Measuring the business value of recommender systems, 2019.

[57] Jaskowski, M., and Jaroszewicz, S. Uplift modeling for clinical trial data. In *ICML Workshop on Clinical Data Analysis* (2012).

[58] Jeunen, O., Rohde, D., and Vasile, F. On the value of bandit feedback for offline recommender system evaluation. *arXiv preprint arXiv:1907.12384* (2019).

[59] Jiang, Y., Shang, J., Liu, Y., and May, J. Redesigning promotion strategy for e-commerce competitiveness through pricing and recommendation. *International Journal of Production Economics 167* (2015), 257–270.

[60] Johnson, C. C. Logistic matrix factorization for implicit feedback data. *Advances in Neural Information Processing Systems 27* (2014).

[61] Johnson, J., Tellis, G. J., Ip, E. H., and Gonçalves, P. To whom, when, and how much to discount. *A Constrained* (2013).

[62] Juan, Y., Zhuang, Y., Chin, W.-S., and Lin, C.-J. Field-aware factorization machines for ctr prediction. In *Proceedings of the 10th ACM Conference on Recommender Systems* (New York, NY, USA, 2016), RecSys '16, ACM, pp. 43–50.

[63] Kaminskas, M., and Bridge, D. Diversity, serendipity, novelty, and coverage: A survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Trans. Interact. Intell. Syst. 7*, 1 (Dec. 2016).

[64] Kamishima, T., and Akaho, S. Personalized pricing recommender system: Multi-stage epsilon-greedy approach. In *Proceedings of the 2Nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems* (New York, NY, USA, 2011), HetRec '11, ACM, pp. 57–64.

[65] Kane, K., Lo, V. S., and Zheng, J. Mining for the truly responsive customers and prospects using true-lift modeling: Comparison of new and existing methods. *Journal of Marketing Analytics 2*, 4 (2014), 218–238.

[66] Kapoor, K., Kumar, V., Terveen, L., Konstan, J. A., and Schrater, P. "i like to explore sometimes": Adapting to dynamic user novelty preferences. In *Proceedings of the 9th ACM Conference on Recommender Systems* (New York, NY, USA, 2015), RecSys '15, ACM, pp. 19–26.

[67] Karatzoglou, A., Amatriain, X., Baltrunas, L., and Oliver, N. Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the Fourth ACM Conference on Recommender Systems* (New York, NY, USA, 2010), RecSys '10, ACM, pp. 79–86.

[68] Kingma, D. P., and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[69] Kocielnik, R., and Hsieh, G. Send me a different message: Utilizing cognitive space to create engaging message triggers. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing* (New York, NY, USA, 2017), CSCW '17, ACM, pp. 2193–2207.

[70] Koren, Y., and Bell, R. *Advances in Collaborative Filtering*. Springer US, Boston, MA, 2015, pp. 77–118.

[71] Koren, Y., Bell, R., and Volinsky, C. Matrix factorization techniques for recommender systems. *Computer*, 8 (2009), 30–37.

[72] Kotkov, D., Konstan, J. A., Zhao, Q., and Veijalainen, J. Investigating serendipity in recommender systems based on real user feedback. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing* (New York, NY, USA, 2018), SAC '18, ACM, pp. 1341–1350.

[73] Kotkov, D., Wang, S., and Veijalainen, J. A survey of serendipity in recommender systems. *Know.-Based Syst. 111*, C (Nov. 2016), 180–192.

[74] Kouki, P., Schaffer, J., Pujara, J., O'Donovan, J., and Getoor, L. User preferences for hybrid explanations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems* (New York, NY, USA, 2017), RecSys '17, ACM, pp. 84–88.

[75] Lawrence, R. D., Almasi, G. S., Kotlyar, V., Viveros, M. S., and Duri, S. S. Personalization of supermarket product recommendations. *Data Min. Knowl. Discov. 5*, 1-2 (Jan. 2001), 11–32.

[76] Le, Q., and Mikolov, T. Distributed representations of sentences and documents. In *International conference on machine learning* (2014), pp. 1188–1196.

[77] Lee, D., and Hosanagar, K. Impact of recommender systems on sales volume and diversity.

[78] Lee, D., and Hosanagar, K. When do recommender systems work the best?: The moderating effects of product attributes and consumer reviews on recommender performance. In *Proceedings of the 25th International Conference on World Wide Web* (2016), WWW '16, International World Wide Web Conferences Steering Committee, pp. 85–97.

[79] Lerche, L., and Jannach, D. Using graded implicit feedback for bayesian personalized ranking. In *Proceedings of the 8th ACM Conference on Recommender Systems* (New York, NY, USA, 2014), RecSys '14, ACM, pp. 353–356.

[80] Letham, B., Sun, W., and Sheopuri, A. Latent variable copula inference for bundle pricing from retail transaction data. In *International Conference on Machine Learning* (2014), pp. 217–225.

[81] Levy, O., and Goldberg, Y. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems* (2014), pp. 2177–2185.

[82] Li, L., Chu, W., Langford, J., and Wang, X. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the Fourth ACM*

International Conference on Web Search and Data Mining (New York, NY, USA, 2011), WSDM '11, ACM, pp. 297–306.

[83] Liang, D., Charlin, L., McInerney, J., and Blei, D. M. Modeling user exposure in recommendation. In *Proceedings of the 25th International Conference on World Wide Web* (Republic and Canton of Geneva, Switzerland, 2016), WWW '16, International World Wide Web Conferences Steering Committee, pp. 951–961.

[84] Lichman, M., and Smyth, P. Prediction of sparse user-item consumption rates with zero-inflated poisson regression. In *Proceedings of the 2018 World Wide Web Conference* (2018), WWW '18, International World Wide Web Conferences Steering Committee, pp. 719–728.

[85] Linden, G., Smith, B., and York, J. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 1 (2003), 76–80.

[86] Lipton, Z. C., Vikram, S., and McAuley, J. Generative concatenative nets jointly learn to write and classify reviews. *arXiv preprint arXiv:1511.03683* (2015).

[87] Liu, L.-P., and Blei, D. M. Zero-inflated exponential family embeddings. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70* (2017), ICML'17, JMLR.org, pp. 2140–2148.

[88] Lo, V. S. Y. The true lift model: A novel data mining approach to response modeling in database marketing. *SIGKDD Explor. Newsl. 4*, 2 (Dec. 2002), 78–86.

[89] Lu, Q., Chen, T., Zhang, W., Yang, D., and Yu, Y. Serendipitous personalized ranking for top-n recommendation. In *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology* (2012), vol. 1, IEEE, pp. 258–265.

[90] Lu, Y., Dong, R., and Smyth, B. Why i like it: Multi-task learning for recommendation and explanation. In *Proceedings of the 12th ACM Conference on Recommender Systems* (New York, NY, USA, 2018), RecSys '18, ACM, pp. 4–12.

[91] Lunceford, J. K., and Davidian, M. Stratification and weighting via the propensity score in estimation of causal treatment effects: a comparative study. *Statistics in medicine 23*, 19 (2004), 2937–2960.

[92] Marlin, B. M., and Zemel, R. S. Collaborative prediction and ranking with non-random missing data. In *Proceedings of the Third ACM Conference on Recommender Systems* (New York, NY, USA, 2009), RecSys '09, ACM, pp. 5–12.

[93] McCrae, R. R., Costa, P. T., de Lima, M. P., Simões, A., Ostendorf, F., Angleitner, A., Marušić, I., Bratko, D., Caprara, G. V., Barbaranelli, C., et al. Age differences in personality across the adult life span: parallels in five cultures. *Developmental psychology 35*, 2 (1999), 466.

[94] McInerney, J., Lacker, B., Hansen, S., Higley, K., Bouchard, H., Gruson, A., and Mehrotra, R. Explore, exploit, and explain: Personalizing explainable recommendations with bandits. In *Proceedings of the 12th ACM Conference on Recommender Systems* (New York, NY, USA, 2018), RecSys '18, ACM, pp. 31–39.

[95] McNee, S. M., Riedl, J., and Konstan, J. A. Being accurate is not enough: How accuracy metrics have hurt recommender systems. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems* (2006), CHI EA '06, pp. 1097–1101.

[96] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (2013), pp. 3111–3119.

[97] Muhammad, K. I., Lawlor, A., and Smyth, B. A live-user study of opinionated explanations for recommender systems. In *Proceedings of the 21st International Conference on Intelligent User Interfaces* (New York, NY, USA, 2016), IUI '16, ACM, pp. 256–260.

[98] Musto, C., Narducci, F., Lops, P., De Gemmis, M., and Semeraro, G. Explod: A framework for explaining recommendations based on the linked open data cloud. In *Proceed-*

ings of the 10th ACM Conference on Recommender Systems (New York, NY, USA, 2016), RecSys '16, ACM, pp. 151–154.

[99] Ono, C., Takishima, Y., Motomura, Y., and Asoh, H. Context-aware preference model based on a study of difference between real and supposed situation data. In *Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization: Formerly UM and AH* (Berlin, Heidelberg, 2009), UMAP '09, Springer-Verlag, pp. 102–113.

[100] Owen, A. B. Monte carlo theory, methods and examples. *Monte Carlo Theory, Methods and Examples. Art Owen* (2013).

[101] Pan, R., Zhou, Y., Cao, B., Liu, N. N., Lukose, R., Scholz, M., and Yang, Q. One-class collaborative filtering. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on* (2008), IEEE, pp. 502–511.

[102] Panniello, U., Tuzhilin, A., Gorgoglione, M., Palmisano, C., and Pedone, A. Experimental comparison of pre- vs. post-filtering approaches in context-aware recommender systems. In *Proceedings of the Third ACM Conference on Recommender Systems* (New York, NY, USA, 2009), RecSys '09, ACM, pp. 265–268.

[103] Papadimitriou, A., Symeonidis, P., and Manolopoulos, Y. A generalized taxonomy of explanations styles for traditional and social recommender systems. *Data Min. Knowl. Discov. 24*, 3 (May 2012), 555–583.

[104] Pennington, J., Socher, R., and Manning, C. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (2014), pp. 1532–1543.

[105] Pu, P., and Chen, L. Trust building with explanation interfaces. In *Proceedings of the 11th International Conference on Intelligent User Interfaces* (New York, NY, USA, 2006), IUI '06, ACM, pp. 93–100.

[106] Radcliffe, N. J., and Surry, P. D. Differential response analysis: Modeling true response by isolating the effect of a single action. *Credit Scoring and Credit Control VI. Edinburgh, Scotland* (1999).

[107] Radcliffe, N. J., and Surry, P. D. Real-world uplift modelling with significance-based uplift trees. *White Paper TR-2011-1, Stochastic Solutions* (2011).

[108] Rafailidis, D., and Nanopoulos, A. Modeling the dynamics of user preferences in coupled tensor factorization. In *Proceedings of the 8th ACM Conference on Recommender Systems* (New York, NY, USA, 2014), RecSys '14, ACM, pp. 321–324.

[109] Rendle, S. Factorization machines. In *2010 IEEE International Conference on Data Mining* (2010), IEEE, pp. 995–1000.

[110] Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence* (Arlington, Virginia, United States, 2009), UAI '09, AUAI Press, pp. 452–461.

[111] Rendle, S., Gantner, Z., Freudenthaler, C., and Schmidt-Thieme, L. Fast context-aware recommendations with factorization machines. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2011), SIGIR '11, ACM, pp. 635–644.

[112] Rendle, S., and Schmidt-Thieme, L. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining* (New York, NY, USA, 2010), WSDM '10, ACM, pp. 81–90.

[113] Ricci, F., Rokach, L., and Shapira, B. Recommender systems: introduction and challenges. In *Recommender systems handbook*. Springer, 2015, pp. 1–34.

[114] Rosenbaum, P. R., and Rubin, D. B. The central role of the propensity score in observational studies for causal effects. *Biometrika 70*, 1 (1983), 41–55.

[115] Rossetti, M., Stella, F., and Zanker, M. Contrasting offline and online results when evaluating recommendation algorithms. In *Proceedings of the 10th ACM Conference on Recommender Systems* (New York, NY, USA, 2016), RecSys '16, ACM, pp. 31–34.

[116] Rubin, D. B. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology 66*, 5 (1974), 688.

[117] Rzepakowski, P., and Jaroszewicz, S. Decision trees for uplift modeling. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on* (2010), IEEE, pp. 441–450.

[118] Said, A., Tikk, D., Stumpf, K., Shi, Y., Larson, M., and Cremonesi, P. Recommender systems evaluation: A 3d benchmark. In *RUE@ RecSys* (2012), pp. 21–23.

[119] Sato, M., Ahsan, B., Nagatani, K., Sonoda, T., Zhang, Q., and Ohkuma, T. Explaining recommendations using contexts. In *23rd International Conference on Intelligent User Interfaces* (New York, NY, USA, 2018), IUI '18, ACM, pp. 659–664.

[120] Sato, M., Izumo, H., and Sonoda, T. Discount sensitive recommender system for retail business. In *Proceedings of the 3rd Workshop on Emotions and Personality in Personalized Systems 2015* (New York, NY, USA, 2015), EMPIRE '15, ACM, pp. 33–40.

[121] Sato, M., Izumo, H., and Sonoda, T. Model of personal discount sensitivity in recommender systems. *IxD&A 28* (2016), 110–123.

[122] Sato, M., Izumo, H., and Sonoda, T. Modeling individual users' responsiveness to maximize recommendation impact. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization* (New York, NY, USA, 2016), UMAP '16, ACM, pp. 259–267.

[123] Sato, M., Nagatani, K., Sonoda, T., Zhang, Q., and Ohkuma, T. Context style explanation for recommender systems. *Journal of Information Processing 27* (2019), 720–729.

[124] Sato, M., Singh, J., Takemori, S., Sonoda, T., Zhang, Q., and Ohkuma, T. Uplift-based evaluation and optimization of recommenders. In *Proceedings of the 13th ACM Conference on Recommender Systems* (New York, NY, USA, 2019), RecSys '19, ACM.

[125] Sato, M., Singh, J., Takemori, S., Sonoda, T., Zhang, Q., and Ohkuma, T. Modeling user exposure with recommendation influence. In *Proceedings of the 35th ACM/SIGAPP Symposium On Applied Computing* (New York, NY, USA, 2020), SAC '20, ACM.

[126] Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2002), SIGIR '02, ACM, pp. 253–260.

[127] Schnabel, T., Swaminathan, A., Singh, A., Chandak, N., and Joachims, T. Recommendations as treatments: Debiasing learning and evaluation. In *International Conference on Machine Learning* (2016), pp. 1670–1679.

[128] Shani, G., Heckerman, D., and Brafman, R. I. An mdp-based recommender system. *J. Mach. Learn. Res. 6* (Dec. 2005), 1265–1295.

[129] Sharma, A., Hofman, J. M., and Watts, D. J. Estimating the causal impact of recommendation systems from observational data. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation* (New York, NY, USA, 2015), EC '15, ACM, pp. 453–470.

[130] Song, B., Yang, X., Cao, Y., and Xu, C. Neural collaborative ranking. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (New York, NY, USA, 2018), CIKM '18, ACM, pp. 1353–1362.

[131] Steck, H. Training and testing of recommender systems on data missing not at random. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2010), KDD '10, ACM, pp. 713–722.

[132] Su, R., Yin, L., Chen, K., and Yu, Y. Set-oriented personalized ranking for diversified top-n recommendation. In *Proceedings of the 7th ACM Conference on Recommender Systems* (New York, NY, USA, 2013), RecSys '13, ACM, pp. 415–418.

[133] Swaminathan, A., and Joachims, T. The self-normalized estimator for counterfactual learning. In *NIPS* (2015), pp. 3231–3239.

[134] Symeonidis, P., Nanopoulos, A., and Manolopoulos, Y. Moviexplain: A recommender system with explanations. In *Proceedings of the Third ACM Conference on Recommender Systems* (New York, NY, USA, 2009), RecSys '09, ACM, pp. 317–320.

[135] ter Hoeve, M., Heruer, M., Odijk, D., Schuth, A., and de Rijke, M. Do news consumers want explanations for personalized news rankings? In *FATREC Workshop on Responsible Recommendation Proceedings* (2017).

[136] Tintarev, N., Dennis, M., and Masthoff, J. Adapting recommendation diversity to openness to experience: A study of human behaviour. In *International Conference on User Modeling, Adaptation, and Personalization* (2013), Springer, pp. 190–202.

[137] Tintarev, N., and Masthoff, J. A survey of explanations in recommender systems. In *Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop* (Washington, DC, USA, 2007), ICDEW '07, IEEE Computer Society, pp. 801–810.

[138] Tintarev, N., and Masthoff, J. Explaining recommendations: Design and evaluation. In *Recommender systems handbook*. Springer, 2015, pp. 353–382.

[139] Tkalcic, M., and Chen, L. Personality and recommender systems. In *Recommender systems handbook*. Springer, 2015, pp. 715–739.

[140] Umberto, P. Developing a price-sensitive recommender system to improve accuracy and business performance of ecommerce applications. *International Journal of Electronic Commerce Studies 6*, 1 (2015), 1–18.

[141] Vasile, F., Smirnova, E., and Conneau, A. Meta-prod2vec: Product embeddings using side-information for recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems* (New York, NY, USA, 2016), RecSys '16, ACM, pp. 225–232.

[142] Vig, J., Sen, S., and Riedl, J. Tagsplanations: Explaining recommendations using tags. In *Proceedings of the 14th International Conference on Intelligent User Interfaces* (New York, NY, USA, 2009), IUI '09, ACM, pp. 47–56.

[143] Wan, M., Wang, D., Goldman, M., Taddy, M., Rao, J., Liu, J., Lymberopoulos, D., and McAuley, J. Modeling consumer preferences and price sensitivities from large-scale grocery shopping transaction logs. In *Proceedings of the 26th International Conference on World Wide Web* (Republic and Canton of Geneva, Switzerland, 2017), WWW '17, International World Wide Web Conferences Steering Committee, pp. 1103–1112.

[144] Wang, M., Gong, M., Zheng, X., and Zhang, K. Modeling dynamic missingness of implicit feedback for recommendation. In *Advances in neural information processing systems* (2018).

[145] Wang, M., Zheng, X., Yang, Y., and Zhang, K. Collaborative filtering with social exposure: A modular approach to social recommendation. In *Thirty-Second AAAI Conference on Artificial Intelligence* (2018).

[146] Wu, W., Chen, L., and He, L. Using personality to adjust diversity in recommender systems. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media* (New York, NY, USA, 2013), HT '13, ACM, pp. 225–229.

[147] Wu, Y.-J., and Teng, W.-G. An enhanced recommendation scheme for online grocery shopping. In *2011 IEEE 15th International Symposium on Consumer Electronics (ISCE)* (2011), IEEE, pp. 410–415.

[148] Yang, C., Pan, S., Mahmud, J., Yang, H., and Srinivasan, P. Using personal traits for brand preference prediction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (2015), pp. 86–96.

[149] Yang, L., Cui, Y., Xuan, Y., Wang, C., Belongie, S., and Estrin, D. Unbiased offline recommender evaluation for missing-not-at-random implicit feedback. In *Proceedings of the 12th ACM Conference on Recommender Systems* (New York, NY, USA, 2018), RecSys '18, ACM, pp. 279–287.

[150] Zhang, F., Zheng, K., Yuan, N. J., Xie, X., Chen, E., and Zhou, X. A novelty-seeking based dining recommender system. In *Proceedings of the 24th International Conference on World Wide Web* (Republic and Canton of Geneva, Switzerland, 2015), WWW '15, International World Wide Web Conferences Steering Committee, pp. 1362–1372.

[151] Zhang, S., Yao, L., Sun, A., and Tay, Y. Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv. 52*, 1 (Feb. 2019), 5:1–5:38.

[152] Zhang, W., Wang, J., Chen, B., and Zhao, X. To personalize or not: A risk management perspective. In *Proceedings of the 7th ACM Conference on Recommender Systems* (New York, NY, USA, 2013), RecSys '13, ACM, pp. 229–236.

[153] Zhang, Y., and Chen, X. Explainable recommendation: A survey and new perspectives. *arXiv preprint arXiv:1804.11192* (2018).

[154] Zheng, Q., Chan, C.-K., and Ip, H. H. An unexpectedness-augmented utility model for making serendipitous recommendation. In *Industrial conference on data mining* (2015), Springer, pp. 216–230.

[155] Zheng, Y., Mobasher, B., and Burke, R. Context recommendation using multi-label classification. In *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - Volume 02* (Washington, DC, USA, 2014), WI-IAT '14, IEEE Computer Society, pp. 288–295.

[156] Zheng, Y., Mobasher, B., and Burke, R. User-oriented context suggestion. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization* (New York, NY, USA, 2016), UMAP '16, ACM, pp. 249–258.