Uplift-based Evaluation and Optimization of Recommenders

Masahiro Sato
sato.masahiro@fujixerox.co.jp
Fuji Xerox

Janmajay Singh
janmajay.singh@fujixerox.co.jp
Fuji Xerox

Sho Takemori
takemori.sho@fujixerox.co.jp
Fuji Xerox

Takashi Sonoda
takashi.sonoda@fujixerox.co.jp
Fuji Xerox

Qian Zhang
qian.zhang@fujixerox.co.jp
Fuji Xerox

Tomoko Ohkuma
ohkuma.tomoko@fujixerox.co.jp
Fuji Xerox

ABSTRACT

Recommender systems aim to increase user actions such as clicks and purchases. Typical evaluations of recommenders regard the purchase of a recommended item as a success. However, the item may have been purchased even without the recommendation. An uplift is defined as an increase in user actions caused by recommendations. Situations with and without a recommendation cannot both be observed for a specific user-item pair at a given time instance, making uplift-based evaluation and optimization challenging. This paper proposes new evaluation metrics and optimization methods for the uplift in a recommender system. We apply a causal inference framework to estimate the average uplift for the offline evaluation of recommenders. Our evaluation protocol leverages both purchase and recommendation logs under a currently deployed recommender system, to simulate the cases both with and without recommendations. This enables the offline evaluation of the uplift for newly generated recommendation lists. For optimization, we need to define positive and negative samples that are specific to an uplift-based approach. For this purpose, we deduce four classes of items by observing purchase and recommendation logs. We derive the relative priorities among these four classes in terms of the uplift and use them to construct both pointwise and pairwise sampling methods for uplift optimization. Through dedicated experiments with three public datasets, we demonstrate the effectiveness of our optimization methods in improving the uplift.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → **Learning from implicit feedback**.

KEYWORDS

Recommendation Effect; Causal Inference; Counterfactual Analysis

ACM Reference Format:
Masahiro Sato, Janmajay Singh, Sho Takemori, Takashi Sonoda, Qian Zhang, and Tomoko Ohkuma. 2019. Uplift-based Evaluation and Optimization of Recommenders. In *Thirteenth ACM Conference on Recommender Systems (RecSys '19), September 16–20, 2019, Copenhagen, Denmark.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3298689.3347018

RecSys '19, September 16–20, 2019, Copenhagen, Denmark

ACM ISBN 978-1-4503-6243-6/19/09.
https://doi.org/10.1145/3298689.3347018

1 INTRODUCTION

One of the major goals of recommender systems is to induce positive user interactions, such as clicks and purchases. Because increases in user interactions directly benefit businesses, recommender systems have been utilized in various domains 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 [42]. We argue that the true success of recommendations is 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, owing to 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 [2, 40]. 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 [3]. All of these methods are based on purchase prediction models optimized for prediction accuracy, even though they target uplift. We expect an improvement in uplift performance by optimizing models directly for the uplift.

In this study, we propose new evaluation methods and optimization methods for uplift-based recommendation. First, we show that common accuracy-based evaluation metrics such as precision do not align with the uplift. Then, we derive evaluation protocols to estimate the average uplift for recommendations, based on a potential outcome framework in causal inference [15, 28, 37]. Furthermore, we propose optimization methods for recommenders, to improve the uplift. We apply these methods to a matrix factorization model [14, 32, 35], which is the most common model for recommenders. To verify the effectiveness of the proposed optimization methods, we compare the uplift performance of our methods with baselines, including recent recommenders [3, 40] that target the uplift. We further investigate the characteristics of our uplift-based optimizations and the recommendation outputs.

[1] We use the term *purchase* to refer to positive interactions in general.

The contributions of this paper are summarized as follows.

- We propose offline evaluation metrics for the recommendation uplift (Section 2).
- We present both pointwise and pairwise optimization methods for uplift-based recommendation (Section 3).
- We demonstrate the effectiveness of our optimization methods through comparisons with baselines (Subsection 5.2).
- We clarify the characteristics of the optimization (Subsection 5.3) and the recommendation outputs (Subsection 5.4).

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. We 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, we aim to evaluate recommenders in terms of the uplift they achieve.

2.1 Discrepancy between Accuracy and Uplift

In this subsection, we demonstrate that accuracy metrics such as precision are unsuitable for the goal of increasing user purchases. To describe two cases with and without a recommendation, we adopt the concept of *potential outcome* from causal inference [15, 28, 37]. 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 $Y^T - Y^C$. 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, we consider four lists of ten recommendations, as shown in Fig. 1. We assume that we have 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 recommended in the past is shown in Table 1. We also list precision under two settings: one with all items on the list, and the other with only the items recommended in the logs. The former is a common setting for the offline evaluations of recommenders [9]. The latter is a setting employed in the previous work [7, 25] 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 Fig. 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. We need to employ an evaluation metric designed for uplift-based evaluation. However, we cannot directly calculate the total uplift, because we only observe either Y^T or Y^C for a user-item pair at a given time. To overcome this difficulty, we apply a causal inference framework to estimate the average treatment effect.

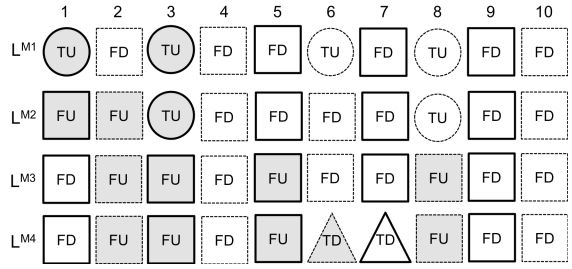


Figure 1: A hypothetical example to illustrate the discrepancy between the accuracy and uplift. Four different recommendation lists, L^{M1}, L^{M2}, L^{M3}, and L^{M4} are generated by different recommendation models, M_1, M_2, M_3, and M_4, respectively. The items with solid borders are actually recommended 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 1.

Table 1: Total uplift and evaluation metrics for four recommendation lists in Fig. 1. The total uplift of a list is indicated by the number of TU items subtracted from the number of TD items. Our uplift metric is described in Subsection 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
Our uplift metric (Subsection 2.3)	0.4	0.2	0.0	-0.2

[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, we assume that the influence of a recommendation is within some discrete time interval when recommended.

2.2 Causal Inference Framework

In this subsection, we introduce the causal inference framework [15, 28, 37], which we apply 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, we can estimate the average treatment effect (ATE), which is expressed as $\mathbb{E}[\tau] = \mathbb{E}[Y^T] - \mathbb{E}[Y^C]$.

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. Consider N subjects, indexed by n. We use S_T and S_C to denote 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;

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

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 [31, Theorem 9.2]).

Because the independence condition $(Y^T, Y^C) \perp Z$ is a strong assumption, we instead consider conditional independence $(Y^T, Y^C) \perp Z|X$, which means that the covariates X contain all *confounders* of (Y^T, Y^C) and Z [28]. Under the conditional independence, the inverse propensity scoring (IPS) estimator,

$$\hat{\tau}_{IPS} = \frac{1}{N} \sum_{n \in S_T} \frac{Y_n^T}{e(X_n)} - \frac{1}{N} \sum_{n \in S_C} \frac{Y_n^C}{1 - e(X_n)}, \tag{2}$$

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 [36]. However, the IPS is prone to suffer from 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 [45]. This adjusts the estimates by the sum of the inverse propensity scores:

$$\hat{\tau}_{SNIPS} = \frac{\sum_{n \in S_T} \frac{Y_n^T}{e(X_n)}}{\sum_{n \in S_T} \frac{1}{e(X_n)}} - \frac{\sum_{n \in S_C} \frac{Y_n^C}{1-e(X_n)}}{\sum_{n \in S_C} \frac{1}{1-e(X_n)}}. \tag{3}$$

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

2.3 Uplift Estimates for Recommenders

In this subsection, we design 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. We assume that we have 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. We want to evaluate $\mathbb{E}[\tau] = \mathbb{E}[Y^T - Y^C|R = 1]$. Let define $p^T = \mathbb{E}[Y^T|R = 1]$ and $p^C = \mathbb{E}[Y^C|R = 1]$, 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 we want to evaluate. In the recommendation logs, we have a list, L_u^D of actually recommended items for the user by the deployed model D. We assume that some items in L_u^M are included in L_u^D, and some are not. We 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, Eq. (1) becomes,

$$\hat{\tau}_{L_u^M} = \frac{1}{|L_u^{M \cap D}|} \sum_{i \in L_u^{M \cap D}} Y_{ui}^T - \frac{1}{|L_u^{M \setminus D}|} \sum_{i \in L_u^{M \setminus D}} Y_{ui}^C. \tag{4}$$

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

We evaluated recommendation lists of Fig. 1 using this metric. The results are shown in the bottom row of Table 1. This metric aligns well with the total uplift, indicating that the proposed metric is appropriate for evaluating recommenders in terms of the uplift.

We can also derive the SNIPS estimate of the uplift from Eq. (3):

$$(\hat{\tau}_{L_u^M})_{SNIPS} = \frac{\sum_{i \in L_u^{M \cap D}} \frac{Y_{ui}^T}{e(X_{ui})}}{\sum_{i \in L_u^{M \cap D}} \frac{1}{e(X_{ui})}} - \frac{\sum_{i \in L_u^{M \setminus D}} \frac{Y_{ui}^C}{1-e(X_{ui})}}{\sum_{i \in L_u^{M \setminus D}} \frac{1}{1-e(X_{ui})}}. \tag{5}$$

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, we take the average over all users U for both estimators:

$$\bar{\tau} \equiv \frac{1}{|U|} \sum_{u \in U} \hat{\tau}_{L_u^M}, \text{ and } \bar{\tau}_{SNIPS} \equiv \frac{1}{|U|} \sum_{u \in U} (\hat{\tau}_{L_u^M})_{SNIPS}. \tag{6}$$

In this study, we employ these metrics for the offline evaluation of uplift performance. We 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 Eq. (4) and (5) disappear. The equations then become similar to the previous counterfactual offline evaluation [7, 25]. Our evaluation is an extension which considers the possibility of purchase without recommendation.

The uplift estimate by Eq. (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 Eq. (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, we can be confident in the evaluation if the results of model comparison are consistent for both *Uplift@N* and $Uplift_{SNIPS}$@N.

3 UPLIFT-BASED OPTIMIZATION

Of the four item classes TU, FU, TD, and TD, defined in Subsection 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 we do not have an observable *ground truth* against which to train models. In this section, we propose uplift optimization methods to overcome the above problem.

3.1 Classification of the Observations

In Subsection 2.1, we categorized items into four hidden classes based on the combinations of potential outcomes. We now 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 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 2: Observable records and possible hidden item classes.

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

We define C_{class} as the set of items in *class* ∈ {R-P, R-NP, NR-P, NR-NP}[5] for a particular user, $u \in U$. We also define I_u^+ and I_u^- as the set of positive and negative items for that user. In traditional accuracy-based optimizations [14, 32, 35], $I_u^+ \sim C_{R-P} \cup C_{NR-P}$ (purchased items) and $I_u^- \sim C_{R-NP} \cup C_{NR-NP}$ (non-purchased items). We argue that this sampling method is not optimal for uplift and redefine the positive and negative samples. Since TU items result in an uplift, we consider classes that include TU items 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, we assume that the number of TU items is much smaller than the total number of items. Hence, the probability of the items in C_{NR-NP} belonging to TU should be low:

$$P(i \in TU | i \in C_{NR-NP}) \equiv |TU \cap C_{NR-NP}|/|C_{NR-NP}| \approx |TU \cap C_{NR-NP}|/|I| < |TU|/|I| \ll 1. \quad (7)$$

On the contrary, considering the fact that recommenders generally improve sales substantially [1], we assume that the possibility of the items in C_{R-P} belonging to TU is not relatively low. Hence,

$$P(i \in TU | i \in C_{R-P}) > P(i \in TU | i \in C_{NR-NP}). \quad (8)$$

Because of the above, we cannot consider C_{NR-NP} to be completely positive. Thus, we propose a parameter α, which is the probability of items from set C_{NR-NP} being sampled as positive. We discuss this further in the following subsection.

3.2 Proposed Sampling Method

The optimization methods of recommender models are generally grouped into two categories: pointwise [11, 14, 32] and pairwise [35, 43] methods. In this subsection, we propose 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 we assign positive labels 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, we conduct stratified sampling because the number of items in each observed class is different. We introduce a parameter γ_P, which represents the ratio of sampling from the purchased items. This kind of downsampling for unpurchased items is a common technique for implicit feedback data [11], which is equivalent to downweighting unpurchased items [14, 32]. 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 the pairwise optimization, we select the positive and negative samples simultaneously. We choose positive samples 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. We sample a candidate class with the same probability; that is, if we sample items from $C_{R-P} \cup C_{NR-NP}$, we sample half 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. We use a stochastic gradient descent for training. Parameters Θ related to each point or pair are updated at each iteration. As for loss function, we use the logistic loss [18] for the pointwise optimization,

$$\mathcal{L}_{point}^{ll} = -(r_{ui} \log(\sigma(\hat{x}_{ui})) + (1 - r_{ui}) \log(1 - \sigma(\hat{x}_{ui}))). \quad (9)$$

The predicted value $\hat{x}_{ui}$ is converted into the label prediction using the sigmoid function, $\sigma(x) = 1/(1+\exp(-x))$. We use the Bayesian personalized ranking (BPR) loss [35] for the pairwise optimization:

$$\mathcal{L}_{pair}^{bpr} = -\log(\sigma(\hat{x}_{ui} - \hat{x}_{uj})), \quad (10)$$

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. We use Matrix Factorization (MF) [22] for $\hat{x}_{ui}$. Our modified versions of MF are called

[5] In Fig. 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.

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

Input: $\alpha, \gamma_P, \gamma_R, \eta, \lambda$
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

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

UpLift-optimized Regularized MF (ULRMF) and UpLift-optimized BPR (ULBPR), which are trained by algorithms 1 and 2 respectively.

As for time complexity, we note that our algorithms perform random sampling of items from prepared sets of observable classes, which is $O(1)$. The bottleneck is for parameter updates, which is $O(d)$ for MF with d factor dimensions. This is common to conventional accuracy-based optimizations. Further, in Subsection 5.3, we show empirically that our uplift-based methods converge faster than accuracy-based ones in terms of iterations required.

4 RELATED WORK

4.1 Causal Inference for Recommenders

Causal inference [15, 28, 37] estimates outcomes through the counterfactual reasoning. It has previously been used to evaluate recommendations in [5, 7, 8, 16, 25], 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 Eq. (4) and (5). Our approach is different, in that we consider 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 [29, 44] of user feedback. IPS estimators were used to adjust the item selection bias of explicit feedback [41] and implicit feedback [50]. Another approach to MNAR is exposure modeling [26], which decomposes missing feedback to either a user's unawareness of or dislike for an item. User exposures have been modeled with social influence [4, 49] and temporal dynamics [48], but not with recommendation influence.

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 [2] 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 (we call it *AwareSatis*). Recent work [40] has incorporated user- and item-dependent responsiveness to recommendations into a purchase prediction model (we call it *RecResp*). Very recently, Bonner and Vasile [3] 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 [39, 46, 47] 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.

A closely related field is uplift modeling [6, 34], 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 [10] creates two prediction models: one to predict outcomes if treated and the other to predict outcomes if not treated. The treatment variable approach [27] incorporates additional variables for predictions under treatment. The label transformation approach [17, 20] converts the labels to train the model if not treated. Finally, in the tree-based approach [33, 38], the splitting criteria for a decision tree are modified for uplift.

We classify recommendation methods targeting uplift in terms of these four approaches in the uplift modeling literature. *CausE* [3] is basically a two-model approach, enhanced by a regularizer between the two models. *AwareSatis* [2] and *RecResp* [40] are treatment variable approaches. Our methods can be classified as label transformation approaches, although our handling of NR-NP as intermediate between positive and negative (using parameter α) is an original approach to overcome class imbalance in typical datasets for recommenders.

It has been argued that recommendation should pursue objectives beyond accuracy [30], and various objectives such as diversity, novelty, and serendipity have been studied [19]. Among them, the most relevant is serendipitous recommendation [12, 23, 24], which aims to recommend items relevant, novel, and unexpected to users. Serendipity focuses on user perception, while uplift focuses more on user behavior.

5 EXPERIMENTS

We experiment to address the following research questions:

- **RQ1** How do our 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?

5.1 Experimental Settings

5.1.1 Datasets and Preprocessing. We experimented with three publicly available datasets[6]: Dunnhumby[7], Tafeng [13], and Xing[8]. The statistics of datasets after filtering are presented in Table 3. The purchase and recommendation logs are separated in discrete time intervals (by day or by week), because recommended items change over time. We explain the details for each dataset below.

Dunnhumby. This dataset includes purchase and promotion logs at a retailer. It provides product category information and we consider these product categories as items. We handle items featured in the weekly mailer, which is information included in the promotion logs, as recommendations. Promotions change each week, and so we separate purchase and recommendation logs 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 we regard the user as having received a recommendation for the item. We filtered the dataset 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.

Tafeng. This dataset contains purchase logs with price information from a Chinese grocery store. This includes the category id for each product, and we consider each category id as a separate item. If the discount ratios of any products in a certain category is over 0.1, then we consider the item as recommended[9]. The dataset is discretized by days. We filtered the dataset 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. We regard the positive user interactions of click, bookmark, and apply, as *purchases*. This includes the impression logs of items which are shown to users by the Xing platform. We consider these impressions as recommendations. The dataset is discretized by days. We filtered the dataset 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.

5.1.2 Evaluation Protocols. We evaluated the uplift performance of each method using the proposed *Uplift*@N and *Uplift*$_{SNIPS}$@N for N=10, 30, and 100[10]. *Precision@30* was also measured, as a reference. Training and evaluation was conducted on each discrete time period. For each training step, we first sampled a time period from among the training periods, and then drew users from among the active users who purchased at least one item during the time period. For evaluation, we calculated the metric for each discrete time, and then averaged them over the evaluation periods. We conducted chronological splitting of the datasets for training and evaluation, to prevent the leakage of future information 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, we estimated the propensities using a 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, we estimated the propensities by the ratio of recommended times for each item in the training periods. That is, we used past recommendation logs as covariates X. For the Dunnhumby dataset, in which time period is much longer (roughly 22 months for Dunnhumby and 4 months for Tafeng), we estimated the propensities by a logistic regression that uses the numbers of purchases and recommendations during previous four weeks as features.

Table 3: 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

5.1.3 Compared Methods. The following methods are compared.

- *RMF* [14, 32][12]: The regularized MF trained with accuracy-based pointwise optimization.
- *BPR* [35]: The MF trained with accuracy-based BPR loss.
- *RecResp* [40]: The MF with user- and item-specific bias terms for recommendations.
- *CausE* [3]: The joint training of two MFs with and without recommendations.
- *CausE-Prod* [3]: The variant of CausE, which has common user factors for two MFs.

[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/
[9]We calculated the discount ratio as 1- (day price)/(normal price). We regard the median price on the same day as the day's price. We define the normal price of a product as the median of the days' prices on all days.
[10]We set N 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.
[11]https://github.com/recsyschallenge/2017/tree/master/baseline
[12]While original work downweight unpurchased items, we downsample them by γ_P.

- *ULRMF* (ours): The MF trained with proposed ULO_{point}.
- *ULBPR* (ours): 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, we predict the uplift using the difference between purchase probabilities with and without recommendations, and use them for top-N recommendation as described in [40]. 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 our methods, ULRMF and ULBPR, are optimized directly for uplift by the unique sampling strategy described in Subsection 3.2.

5.1.4 Implementation and Parameter Settings. All the compared methods are latent factor models, and we set the factor dimensions to 100. Adam [21] 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 [11], for RMF and ULRMF. We do not apply this stratification 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.

5.2 Performance Comparison (RQ1)

We compared the uplift performances between our methods and baselines (Table 4). We make the following key observations:

- Our 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 ours) tend to outperform RMF and BPR. This implies that our uplift metrics can measure the uplift improvement as expected.

5.3 Uplift-based Optimization Properties (RQ2)

We investigated the learning curves in the Dunnhumby dataset[13] (Fig. 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 scalability of our methods in terms of computation time.

In our experiments, items were filtered by the time periods that the items existed for in purchase logs. We modified the filtering

[13]Similar trends were observed in the Tafeng and Xing datasets, which are not presented here due to the space limitation.

Table 4: 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. We compare only with other families of methods, namely, we do not compare CausE-Prod 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

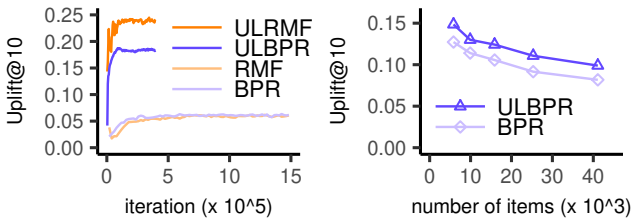
(a) Dunnhumby dataset.

	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.

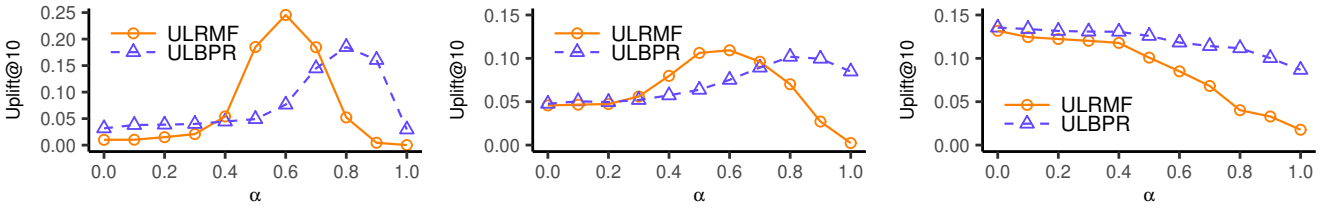
	Uplift			$Uplift_{SNIPS}$			*Precision*
	N=10	N=30	N=100	N=10	N=30	N=100	N=30
RMF	0.1037	0.1118	0.1108	0.1038	0.1121	0.1110	0.0189
BPR	0.1056	0.1168	0.1157	0.1057	0.1168	0.1157	**0.0239***
RecResp	0.0839	0.1017	0.1149	0.0838	0.1015	0.1148	0.0060
CausE	0.1163	0.1243	0.1280	0.1163	0.1243	0.1281	0.0099
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) Learning curve convergence. **(b) Different filtering criteria.**

Figure 2: Scalability of our methods. Used datasets are Dunnhumby for (a) and Xing for (b).



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

Figure 3: Dependence on the probability of regarding NR-NP as positive (α). The regularization coefficient λ is set to 10^{-2}.

Table 5: 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

criteria from 7 to 19 days by a 3-day interval for the Xing dataset, in which the numbers of items varied from 41,099 to 5,828. As shown in Fig. 2 (b), ULBPR outperforms BPR in all these conditions. We also experimented 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 our uplift-based optimization can improve uplift for datasets in a wide range of data densities.

ULRMF and ULBPR have a model-specific hyperparameter α, which is the probability of regarding NR-NP as positive. Fig. 3 shows the dependence on α. The optimal α is less than 1, which supports our claim of treating NR-NP as an intermediate between positive and negative in Subsection 3.1.

Our 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, we investigated the distribution of the recommended items in the observable four classes (Table 5). ULRMF 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 our optimization methods do not orient a model M for mimicking the recommendation policy of the currently deployed model D.

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 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[14]. Often-recommended items by ULRMF include those that might induce impulse purchases such as pasta sauce and heat-and-serve meals.

Table 6: 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)

6 CONCLUSIONS

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

In the future, we plan to compare our uplift-based offline evaluations with online A/B experiments. Because our uplift-based optimizations are generic methods, they are applicable to various recommender models. Recently, recommender models using neural networks have outperformed conventional models [11, 43]. We expect that applying our uplift-based optimizations to neural network models would further enhance the uplift, which is also a subject of our future work.

[14] 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.

REFERENCES

[1] Thiago Belluf, Leopoldo Xavier, and Ricardo Giglio. 2012. 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 (RecSys '12)*. 277–280.

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

[3] Stephen Bonner and Flavian Vasile. 2018. Causal Embeddings for Recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18)*. 104–112.

[4] Jiawei Chen, Yan Feng, Martin Ester, Sheng Zhou, Chun Chen, and Can Wang. 2018. 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 (CIKM '18)*. 953–962.

[5] Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H. Chi. 2019. Top-K Off-Policy Correction for a REINFORCE Recommender System. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19)*. ACM, 456–464.

[6] Floris Devriendt, Darie Moldovan, and Wouter Verbeke. 2018. 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.

[7] Alexandre Gilotte, Clément Calauzènes, Thomas Nedelec, Alexandre Abraham, and Simon Dollé. 2018. Offline A/B Testing for Recommender Systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM '18)*. 198–206.

[8] Alois Gruson, Praveen Chandar, Christophe Charbuillet, James McInerney, Samantha Hansen, Damien Tardieu, and Ben Carterette. 2019. Offline Evaluation to Make Decisions About PlaylistRecommendation Algorithms. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19)*. ACM, 420–428.

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

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

[11] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*. 173–182.

[12] Yoshinori Hijikata, Takuya Shimizu, and Shogo Nishida. 2009. Discovery-oriented Collaborative Filtering for Improving User Satisfaction. In *Proceedings of the 14th International Conference on Intelligent User Interfaces (IUI '09)*. 67–76.

[13] Chun-Nan Hsu, Hao-Hsiang Chung, and Han-Shen Huang. 2004. Mining Skewed and Sparse Transaction Data for Personalized Shopping Recommendation. *Mach. Learn.* 57, 1-2 (Oct. 2004), 35–59.

[14] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE, 263–272.

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

[16] Rolf Jagerman, Ilya Markov, and Maarten de Rijke. 2019. 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 (WSDM '19)*. ACM, 447–455.

[17] Maciej Jaskowski and Szymon Jaroszewicz. 2012. Uplift modeling for clinical trial data. In *ICML Workshop on Clinical Data Analysis*.

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

[19] Marius Kaminskas and Derek Bridge. 2016. 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).

[20] Kathleen Kane, Victor SY Lo, and Jane Zheng. 2014. 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.

[21] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[22] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

[23] Denis Kotkov, Joseph A. Konstan, Qian Zhao, and Jari Veijalainen. 2018. Investigating Serendipity in Recommender Systems Based on Real User Feedback. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing (SAC '18)*. 1341–1350.

[24] Denis Kotkov, Shuaiqiang Wang, and Jari Veijalainen. 2016. A Survey of Serendipity in Recommender Systems. *Know.-Based Syst.* 111, C (Nov. 2016), 180–192.

[25] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. 2011. 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 (WSDM '11)*. 297–306.

[26] Dawen Liang, Laurent Charlin, James McInerney, and David M. Blei. 2016. Modeling User Exposure in Recommendation. In *Proceedings of the 25th International Conference on World Wide Web (WWW '16)*. 951–961.

[27] Victor S. Y. Lo. 2002. The True Lift Model: A Novel Data Mining Approach to Response Modeling in Database Marketing. *SIGKDD Explor. Newsl.* 4, 2 (Dec. 2002), 78–86.

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

[29] Benjamin M. Marlin and Richard S. Zemel. 2009. Collaborative Prediction and Ranking with Non-random Missing Data. In *Proceedings of the Third ACM Conference on Recommender Systems (RecSys '09)*. 5–12.

[30] Sean M. McNee, John Riedl, and Joseph A. Konstan. 2006. Being Accurate is Not Enough: How Accuracy Metrics Have Hurt Recommender Systems. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems (CHI EA '06)*. 1097–1101.

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

[32] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. 2008. One-class collaborative filtering. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE, 502–511.

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

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

[35] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI '09)*. 452–461.

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

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

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

[39] Masahiro Sato, Hidetaka Izumo, and Takashi Sonoda. 2015. Discount Sensitive Recommender System for Retail Business. In *Proceedings of the 3rd Workshop on Emotions and Personality in Personalized Systems 2015 (EMPIRE '15)*. 33–40.

[40] Masahiro Sato, Hidetaka Izumo, and Takashi Sonoda. 2016. Modeling Individual Users' Responsiveness to Maximize Recommendation Impact. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization (UMAP '16)*. 259–267.

[41] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as Treatments: Debiasing Learning and Evaluation. In *International Conference on Machine Learning*. 1670–1679.

[42] Amit Sharma, Jake M. Hofman, and Duncan J. Watts. 2015. Estimating the Causal Impact of Recommendation Systems from Observational Data. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation (EC '15)*. 453–470.

[43] Bo Song, Xin Yang, Yi Cao, and Congfu Xu. 2018. Neural Collaborative Ranking. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*. 1353–1362.

[44] Harald Steck. 2010. 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 (KDD '10)*. 713–722.

[45] Adith Swaminathan and Thorsten Joachims. 2015. The Self-Normalized Estimator for Counterfactual Learning. In *NIPS*. 3231–3239.

[46] Panniello Umberto. 2015. 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.

[47] Mengting Wan, Di Wang, Matt Goldman, Matt Taddy, Justin Rao, Jie Liu, Dimitrios Lymberopoulos, and Julian McAuley. 2017. Modeling Consumer Preferences and Price Sensitivities from Large-Scale Grocery Shopping Transaction Logs. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*. 1103–1112.

[48] Minghui Wang, Mingming Gong, Xiaolin Zheng, and Kun Zhang. 2018. Modeling Dynamic Missingness of Implicit Feedback for Recommendation. In *NeurIPS*.

[49] Menghan Wang, Xiaolin Zheng, Yang Yang, and Kun Zhang. 2018. Collaborative filtering with social exposure: A modular approach to social recommendation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[50] Longqi Yang, Yin Cui, Yuan Xuan, Chenyang Wang, Serge Belongie, and Deborah Estrin. 2018. Unbiased Offline Recommender Evaluation for Missing-not-at-random Implicit Feedback. In *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18)*. 279–287.