Leveraging Post-click Feedback for Content Recommendations

Hongyi Wen
Cornell Tech, Cornell University
New York, NY, USA
hw557@cornell.edu

Longqi Yang
Cornell Tech, Cornell University
New York, NY, USA
ylongqi@cs.cornell.edu

Deborah Estrin
Cornell Tech, Cornell University
New York, NY, USA
destrin@cornell.edu

ABSTRACT

Implicit feedback (e.g., clicks) is widely used in content recommendations. However, clicks only reflect user preferences according to their first impressions. They do not capture the extent to which users continue to engage with the content. Our analysis shows that more than half of the clicks on music and short videos are followed by skips from two real-world datasets. In this paper, we leverage post-click feedback, e.g. skips and completions, to improve the training and evaluation of content recommenders. Specifically, we experiment with existing collaborative filtering algorithms and find that they perform poorly against post-click-aware ranking metrics. Based on these insights, we develop a generic probabilistic framework to fuse click and post-click signals. We show how our framework can be applied to improve pointwise and pairwise recommendation models. Our approach is shown to outperform existing methods by 18.3% and 2.5% respectively in terms of Area Under the Curve (AUC) on the short-video and music dataset. We discuss the effectiveness of our approach across content domains and trade-offs in weighting various user feedback signals.

CCS CONCEPTS

• **Information systems** → **Collaborative filtering**; **Relevance assessment**; **Recommender systems**.

KEYWORDS

Collaborative filtering; Implicit feedback; Content recommendation; User modeling

ACM Reference Format:
Hongyi Wen, Longqi Yang, and Deborah Estrin. 2019. Leveraging Post-click Feedback for Content Recommendations. 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/10.1145/3298689.3347037

1 INTRODUCTION

Content recommendation systems have become increasingly important to help users discover information items such as videos, music, news, and podcasts [2, 10, 12, 23, 29]. Existing recommenders are often built on implicit feedback signals (e.g. click [5] and watch [2]) that are readily available in large volume; but implicit feedback is notoriously hard to interpret [7, 13]. Prior literature assumes that a click indicates positive preference, while a lack of that signal is a negative indicator [5, 16, 21]. Such a simplified assumption can result in misinterpretations of user preferences, because implicit feedback is often logged before content consumption, and thus is only a reflection of user's first impressions or impulsive response to clickbait content. Consequently, recommenders built on implicit feedback alone can be significantly biased and vulnerable [7, 15, 34].

In this paper, we propose to leverage post-click signals such as skips and completions to disambiguate clicks and improve content recommendation performance. Skips can be perceived as a form of user feedback that naturally appears after user clicks, e.g., skipping a song or video as opposed to consuming it to completion. Intuitively, "click-complete" is a stronger positive signal than "click" and "click-skip", which implicitly encodes users' negative feedback. Through extensive experiments with large-scale music and short-video datasets, we find that post-click signals capture important preference signals that are not learned by existing systems trained on the click data alone. For example, more than half of clicks are followed by skips in music and short video consumption; skip rates are negatively correlated with item popularity; and recommenders trained only on click data perform poorly on post-click-aware ranking metrics.

Building on the insights above, we develop a generic probabilistic framework to incorporate post-click signals in training and evaluating content-recommenders. In our framework, clicks are disentangled into positive and negative observations, and the confidence levels for different types of feedback are modeled through the variances of Gaussian distributions. We show that such a framework can be readily applied to existing pointwise [5] and pairwise [21] recommendation algorithms. Our main contributions in this paper are as follows:

- We show empirical evidence that although existing recommendation algorithms predict user clicks reliably, they predict post-click preferences rather poorly. This problem has been highlighted in a lab study to measure multi-phrase user preferences in online news consumption [15]. Our experiment is the first to study the performance gap empirically, using large-scale real-world datasets.
- We propose a generic probabilistic framework to fuse click and post-click feedback. We demonstrate its applications in improving pointwise and pairwise collaborative filtering models. Our framework achieves an average of 18.3% performance improvements in terms of Area Under Curve (AUC) for short video and 2.5% for music recommendations. We discuss the effectiveness of our approach across different content domains.

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

ACM ISBN 978-1-4503-6243-6/19/09…$15.00
https://doi.org/10.1145/3298689.3347037

- We demonstrate that our framework enables flexible trade-offs between confidence levels placed on the various post-click signals. This is useful in those scenarios when users' negative feedback are especially undesirable. We discuss how this relates to balancing exploitation and exploration in recommendations.

2 RELATED WORK

There have been few studies of post-click feedback for content recommender systems. Most relevant work addresses web searching and browsing tasks, where clicks are noisy indicators of item relevance. Below, we briefly review the literature on leveraging negative feedback to improve performance on several Information Retrieval (IR) tasks. They share some similarities in terms of modeling implicit feedback from user logs, but would require substantial modifications to be applied in our problem setting. This motivated us to take a step further into post-click behavior in content consumption.

2.1 Mismatch between clicks and preferences

User clicks have long been used in recommendation systems to predict user preferences [5, 21]. However, previous research has pointed out that click data may be confounded by factors like position bias [7] and presentation [7, 32]. These problems are raised in the specific context of web searching and browsing. In a recent laboratory study of online news reading, Lu et al. [15] show that click signals often do not align with users' true preferences towards content. They find that more than half of the clicked articles are disliked by users. Inspired by these findings, we aim to understand how informative post-click signals are in content consumption and how they could be leveraged to improve user preference modeling by mitigating the gaps between clicks and user preferences.

2.2 Dwell time as engagement metric

Previous work used dwell time (the time a user spends browsing a website before she moves away from it) as an alternative evaluation metric to clicks for measuring user satisfaction. It has shown improved performance in many IR tasks, such as searching and document retrieval [8, 11, 31]. A key challenge for dwell time is that the reliability of the signal often relies on heuristic criterion to decide whether an item is relevant or not. For example, clicks on search results with over 30 seconds dwell time are considered to be relevant [11]. Using skips to measure engagement in content consumption is related to the idea of dwell time. Yang et al. [26] implemented a music recommendation system in which various implicit feedback were collected, such as search, archive, and completion. They found that completion was a strong indicator of positive preference. However, similar to dwell time in web browsing, which is impacted by the speed of reading and webpage presentation, skips could also be influenced by confounding factors such as user context. For example, skip patterns while listening to music in the background could be very different from those when actively searching for new songs. Our focus in this work is to examine user skips in real-world content consumption and address the challenges in learning from such noisy data.

2.3 Learning from negative feedback

Wang et al. [25] conducted an overview of algorithms that can learn from negative feedback signals, specifically in the context of search query retrieval. They investigated language models and vector-space models where non-clicked documents were treated as negative relevance feedback. In their learning framework, the relevance score given to each (document, query) pair is:

$$S(\vec{Q}, \vec{D}) = \vec{Q}\vec{D} - \gamma \times \frac{1}{|N|} \sum_{D_n \in N} \vec{D}_n \vec{D},$$

where $\vec{Q}, \vec{D}$ are query and document vectors respectively, N is the set of negative documents related to the query, and γ is the weight of negative feedback. Similar strategies are leveraged for query expansions where not-selected query suggestions are used as negative feedback [33].

However, the proposed approaches cannot be directly applied to content recommendations, since each item can only receive a positive or a negative feedback at interaction time, while in the above setting each query is accompanied by a combination of positive and negative feedback. Instead of re-computing the relevance score for each (Q, D) pair, we propose a probabilistic approach to model the influence manifested by sets of positive and negative feedback across users and items through collaborative filtering. In practice, our approach can be implemented through re-sampling during batch training, without the need to change the objective function for optimizations, making it generalizable to many pointwise and pairwise models.

3 ANALYSIS OF POST-CLICK FEEDBACK

Previous work has raised the concern that clicks are noisy estimators of user preferences in web browsing and searching [7, 15, 31]. In this section, to understand the gap between click and post-click feedback on content platforms, we analyze two large-scale real-world datasets. The datasets were collected for two public challenges: the Spotify Sequential Skip Prediction Challenge [1] and the ByteDance Short Videos Understanding Challenge.[1]

The **Spotify** dataset contains hundreds of millions of listening sessions across Spotify users. Each session consists of at most 20 songs, and a user may skip or complete listening to each song. [2] We randomly sampled a subset of nine million listening sessions for our experiments. Similarly, the **ByteDance** dataset provides details of user interactions with short videos (10 seconds in length), including whether or not each video was completed. We used a subset of the data that contains over 13 million records.

3.1 Properties of post-click feedback

Our analysis shows that post-click feedback is pervasive across domains. We observed that 51% and 56% of the clicks on music and short videos are followed by skips, respectively. In other words, *more than half of all clicks lead to potential user dissatisfaction.* This aligns with findings from a user study on online news consumption in which only 51% of clicked news are preferred by users post-read [15].

[1] https://biendata.com/competition/icmechallenge2019/

[2] A song is skipped if it is consumed less than a duration threshold defined by the challenge.

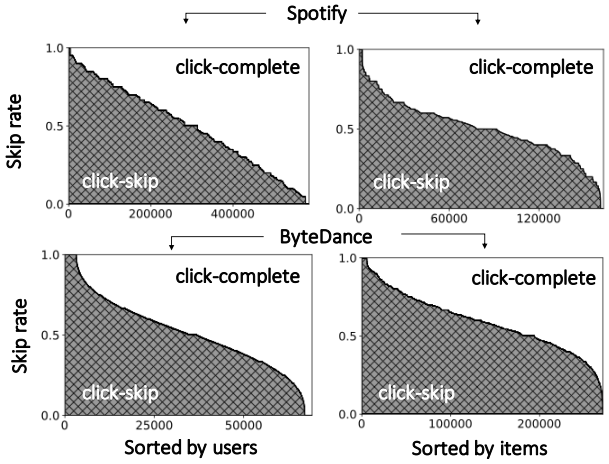

Figure 1: Distributions of skip rates across users and items on Spotify and ByteDance. The first column illustrates the distribution of user-wise skip rate, ranked from high to low. The second column illustrates the distribution of item-wise skip rate. The area filled with dark shades refer to "click-skip" and lighter shades refer to "click-complete." We observed different skip behavior across the two content domains.

We also observed that such skip behavior manifests different patterns for music and short videos. As shown in Fig. 1, the percentage of music skipped by each individual user (i.e., user-wise skip rate) is uniformly randomly distributed. However, for short videos, the skip rates are distributed normally, with most users skipping 50%-55% of the content. In terms of item-wise skip rates, both datasets manifest similar patterns.

We hypothesize that the differences in skip rates may be due to domain-specific consumption behavior. Skips might be dependent on the context of use, as users may have constrained access to the player when listening to songs (e.g., during running). Also, since in general video content is more engaging and requires more user attention, skips may be less noisy as indicators of user preferences for this content.

Another hypothesis concerns time thresholds. The timing of a skip can further determine the underlying user preferences in content consumption, similar to the heuristics for interpreting dwell time in web searches [11]. The skips provided by the Spotify dataset are derived from pre-defined bucketed time thresholds [1], while time may be less of an issue for videos as the duration of these videos is relatively short.

We further investigated the relationship between skip rate and item popularity (Fig. 2). For this analysis, we only included items that received more than 100 user clicks. For both datasets, we found that as an item gets less popular (i.e., receives fewer clicks), its skip rate tends to become higher ($r = 0.15, p < 10^{-64}$).

The above empirical results demonstrate that post-click feedback such as skips potentially capture significant information about user preferences that are unreflected in the click data. Next, we

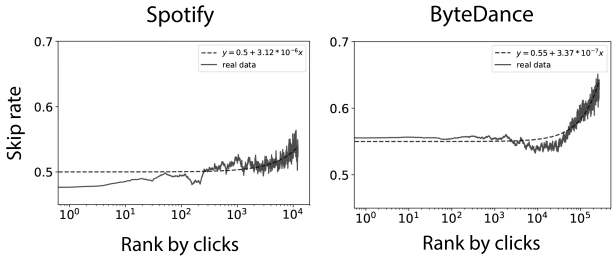

Figure 2: Relationship between skip rate and item popularity as indicated by user clicks. In general, skip rates increase as items become less popular.

conducted a pilot experiment to quantify the gap between click and post-click feedback signals.

3.2 Gap between click and post-click feedback

To measure the gap, we trained several existing recommendation models using click data alone, and then evaluated the performance of these algorithms on the task of ranking "click-complete" items against "click-skip" ones. If the gap between click and post-click feedback is large, we expect to observe poor performance on the post-click-aware ranking task.

3.2.1 Dataset preprocessing. We chronologically split both datasets into training, validation and testing sets. We held out five items from each user for validation and testing sets respectively. We ensured both "click-skip" and "click-complete" signals were included in the validation and test set for each user. The rest of the items from the user were used for training. The statistics of the processed datasets are summarized in Table 1. After preprocessing, we verified that the distribution of skips was unchanged.

3.2.2 Evaluations. We trained two widely-used recommenders, Weighted Regularized Matrix Factorization (**WRMF**) [5] and Bayesian Personalized Ranking (**BPR**) [21], using clicks as implicit feedback on the training set [5]. We then evaluated the performance of the recommenders on two ranking tasks:

- Task 1: A standard offline evaluation task in which a recommender aims to rank held-out clicked items above unobserved items.
- Task 2: A post-click-aware ranking task in which the recommender is evaluated against ranking held-out completed items above skipped items.

For both tasks, we used Area under the ROC curve (AUC) to evaluate the ranking quality of recommenders:

$$\text{AUC} = \frac{1}{|U|}\sum_{u}\frac{1}{|E(u)|}\sum_{(i,j)\in E(u)}\delta(\hat{p}_{ui} > \hat{p}_{uj}),$$

where δ is an indicator function and $\hat{p}$ denotes the user preference scores predicted by the recommender (more details in Section 4). For Task 1, the evaluation pairs for user u are defined as: $E(u) := \{(i,j)|(u,i) \in S_{\text{test}} \wedge (u,j) \notin (S_{\text{test}} \cup S_{\text{train}})\}$. For Task 2, $E(u) := \{(i,j)|(u,i),(u,j) \in S_{\text{test}}, i \in S_{\text{completed}} \wedge j \in S_{\text{skipped}}\}$. AUC for random guessing is 0.5 and for perfect ranking would equal 1.

Dataset	# of users	# of items	# of records	Density	Percentage of skips %
Spotify	229,792	100,586	4,090,895	0.018%	51.05%
ByteDance	37,043	271,259	9,391,103	0.093%	55.13%

Table 1: Summary of datasets used for experiments after pre-processing.

Method	ByteDance		Spotify	
	Task 1	Task 2	Task 1	Task2
WRMF	0.789	0.521	0.925	0.540
BPR	0.791	0.527	0.919	0.541

Table 2: Performance of WRMF and BPR on two personalized ranking tasks measured by AUC. We observed that state-of-the-art recommenders based on clicks perform well on ranking held-out clicked items against unobserved items, but poorly on ranking completed items against skipped items, indicating a significant gap between user clicks and post-click preferences.

3.2.3 Results. We observed a significant performance gap between the two ranking tasks, shown in Table 2. In predicting whether users tend to click on certain items, WRMF and BPR performed reasonably well on both datasets. However, both recommenders performed rather poorly in predicting whether or not users would later complete or skip the items. For the post-click-aware ranking task, AUC drops to 0.52 and 0.54 respectively on ByteDance and Spotify datasets. The results of this pilot experiment demonstrate that existing recommenders trained on click data alone do not effectively learn user preferences embedded in post-click feedback data. Therefore, we can potentially improve recommendation performances by enabling recommenders to incorporate these new data sources. In the next section, we present a generic probabilistic framework to achieve this objective.

4 MODELING USER PREFERENCES WITH POST-CLICK FEEDBACK

Our main insight is that post-click feedback provides an additional layer of granularity for user preferences, and preference indications from click and post-click signals can have different levels of reliability. In this section, we describe the details of our framework and show how it can be applied to improve pointwise and pairwise recommendation models. The notations used in this paper are summarized in Table 3.

4.1 General framework

We considered the problem of building recommendation algorithms using a set of observations on user preferences, denoted as $O = \{o_1, o_2, .., o_n\}$. Each o_k represents an instance of observation derived from user interaction records.

For example, in the pointwise setting, O can refer to interactions between user-item pairs (u, i), such as "click" and "no-click"; in the pairwise setting, O can refer to the interactions among (user, item, item) triplets (u, i, j), where (u, i) and (u, j) indicate positive and negative interactions respectively. The model parameters Θ of a recommender are optimized to maximize $P(O|\Theta)$ using Maximum Likelihood Estimation (MLE), i.e.,

$$\Theta_{MLE} = \arg\max_{\Theta} P(O|\Theta) = \arg\max_{\Theta} \prod_{1}^{n} P(o_k|\Theta) \tag{1}$$

With post-click feedback, we divided O into three disjoint types of observations: positive observations (denoted as O_P and $o_k = 1, \forall o_k \in O_P$), negative observations (denoted as O_N and $o_k = 0, \forall o_k \in O_P$), and no observations (denoted as O_M and $o_k = 0, \forall o_k \in O_M$). In other words, $O = O_P \cup O_N \cup O_M$.

We assume all three types of observations satisfy Gaussian distributions with different variances. This reflects the confidence of an observation being correctly interpreted (e.g., clicks and skips are noisy indicators of user preferences), i.e.,

$$o \sim \begin{cases} \mathcal{N}(\mu_o, \sigma_P^2), & o \in O_P \\ \mathcal{N}(\mu_o, \sigma_N^2), & o \in O_N \\ \mathcal{N}(\mu_o, \sigma_M^2), & o \in O_M \end{cases}$$

Under the above assumptions for observations, the MLE estimator of Θ (eqn. 1) can be written as:

$$\begin{aligned} \Theta_{MLE} &= \arg\max_{\Theta} \prod_{o \in O_P} P(1|\Theta) \prod_{o \in O_N} P(0|\Theta) \prod_{o \in O_M} P(0|\Theta) \\ &= \arg\min_{\Theta} \frac{1}{2\sigma_P^2} \sum_{o \in O_P} (1-\mu_o)^2 + \frac{1}{2\sigma_N^2} \sum_{o \in O_N} (\mu_o)^2 + \frac{1}{2\sigma_M^2} \sum_{o \in O_M} (\mu_o)^2, \end{aligned} \tag{2}$$

which defines a generic probabilistic framework for learning from post-click user feedback (eqn. 2). Specifically, for pointwise observations, O_P consists of (u, i) pairs where user u "click-complete" item i; O_N contains pairs where user u "click-skip" item i; and other pairs in which user u does not interact with item i are included in O_M. Under this pointwise setting, μ_o can be calculated as:

$$\mu_o := \hat{p}_{ui}, \forall u \in \mathcal{U}, i \in \mathcal{I}$$

where $\hat{p}_{ui}$ are preference scores predicted by a recommendation model. Eqn. 2 can then be interpreted as minimizing the euclidean distances between predicted scores and ground truth scores.

For pairwise observations, O_P consists of (u, i, j) triplets where item i is implicitly ranked higher than item j by user u; O_N contains triplets where item i is ranked lower than j; and other triplet combinations are in O_M. With such form of observations, μ_o can be defined as:

$$\mu_o := \begin{cases} \hat{p}_{uij} & o \in O_P \cup O_N \\ 0 & o \in O_M \end{cases}$$

where $\hat{p}_{uij}$ is the probability of item i being ranked above item j for user u. The MLE objective in this setting is to maximize the probability of correctly ranking items. Notice we only define μ_o on

$\mathcal{U}, \mathcal{I}$	set of users and items
O	observations of user preference
O_P, O_N, O_M	positive, negative and missing observations
$\mathcal{I}_u^+, \mathcal{I}_u^-$	set of items user u interacts/not interacts with
$\mathcal{P}_u$	set of items user u indicates positive feedback
$\mathcal{N}_u$	set of items user u indicates negative feedback
p_{ui}	defined preference score between user u and item i
$\hat{p}_{ui}$	model predicted preference score
λ_p, λ_n	model hyperparameters for pointwise models
$\lambda_{p,n}$	model hyperparameters for pairwise models

Table 3: Notations used in the paper.

subsets of O, since for the missing observations we lack enough information about their relative rankings. We describe other details of pointwise and pairwise models in Sections 4.3 and 4.4.

The key difference between our framework and the classical approach to learning from implicit feedback [5] is that we break the ties among clicked or not-clicked items by incorporating post-click feedback signals. In addition, we leverage the variances of Gaussian distributions to control the confidence levels placed on different types of user feedback signals. This allows our models to adapt across diverse recommendation environments.

4.2 Preliminary

In this section, to show how our framework can be realized concretely in pointwise and pairwise recommendation settings, we first define notations to represent the structure of click and post-click data.

For user $u \in \mathcal{U}$ and item $i \in \mathcal{I}$, Hu et al. [5] interpreted user preferences using clicks as following:

$$p_{ui} = \begin{cases} 1, & u \text{ clicks on } i, \\ 0, & u \text{ does not click on } i, \end{cases}$$

For simplicity, we defined $\mathcal{I}_u^+ := \{i | i \in \mathcal{I} \wedge p_{ui} = 1\}$ and $\mathcal{I}_u^- := \mathcal{I} \setminus \mathcal{I}_u^+$ to denote clicked and not-clicked items for user u. To reflect user preferences indicated by post-click feedback, we introduced a set of binary variables, s_{ui} :

$$s_{ui} = \begin{cases} 1, & u \text{ clicks and skips } i, \\ 0, & u \text{ clicks and completes } i, \end{cases}$$

Based on such interpretations, we further disambiguated interacted items into two sets:

$$\mathcal{P}_u := \{i | i \in \mathcal{I}_u^+ \wedge s_{ui} = 0\}, \mathcal{N}_u := \{i | i \in \mathcal{I}_u^+ \wedge s_{ui} = 1\},$$

where $\mathcal{P}_u$ and $\mathcal{N}_u$ denote positive and negative feedback from user u. Note that $\mathcal{P}_u, \mathcal{N}_u, \mathcal{I}_u$ are different from O_P, O_N, O_M defined in the previous subsection: $\mathcal{P}_u, \mathcal{N}_u, \mathcal{I}_u$ are sets of items with which a user indicates her preferences, while O_P, O_N, O_M are abstractions of observations (pointwise or pairwise) across all users and items.

4.3 Pointwise recommendation model

A representative approach under the pointwise category is low-rank Matrix Factorization. The classical matrix factorization framework aims to learn latent user vectors $x \in \mathbb{R}^{|\mathcal{U}| \times l}$ and latent item vectors $y \in \mathbb{R}^{|\mathcal{I}| \times l}$ by factorizing the sparse user-item matrix through singular value decomposition (SVD) [19]. Users' preference scores on items are given by the dot products between user and item vectors:

$$\hat{p}_{ui} = x_u^T y_i. \tag{3}$$

However, the classical MF model can easily overfit implicit feedback signals. To address this problem, Weighted Regularized MF [5, 18] was proposed. The main contributions of WRMF are: (1) regularizing model parameters and (2) reweighting pointwise loss to balance the impact of positive feedback. Specifically, WRMF optimizes the following objective function:

$$\min_{x^* y^*} \sum_{u,i} c_{ui}(p_{ui} - \hat{p}_{ui})^2 + \lambda R(x, y), \tag{4}$$

where $R(x, y) = ||x||_f^2 + ||y||_f^2$ represents the Frobenius norms for user and items matrices, λ is regularization parameter, and c_{ui} is the confidence level for each observation. Specifically, Pan et al. [18] used $c_{ui} = 1$ for $i \in \mathcal{I}_u^+$ and chose a lower constant for $i \in \mathcal{I}_u^-$. Hu et al. estimated c_{ui} for positive items with additional data, such as the number of clicks, and set $c_{ui} = 1$ for the unobserved items. This pointwise approach is a special instance of our general framework (eqn. 2). Specifically, with click data only, the observations O are:

$$O_{P_{\text{click}}} = \{(u, i) | u \in \mathcal{U}, i \in \mathcal{I}_u^+\}, O_{M_{\text{click}}} = \{(u, i) | u \in \mathcal{U}, i \in \mathcal{I}_u^-\}$$

Notice that $O_{N_{\text{click}}} = \emptyset$, as there is no negative interactions with click data alone. The MLE estimation (eqn. 2) can be written as:

$$\arg\min_{x,y} \frac{1}{2\sigma_P^2} \sum_{(u,i) \in O_{P_{\text{click}}}} (1 - \hat{p}_{ui})^2 + \frac{1}{2\sigma_M^2} \sum_{(u,i) \in O_{M_{\text{click}}}} \hat{p}_{ui}^2 \tag{5}$$

With post-click signals, we can extend the definition of O_{click} to incorporate negative feedback, i.e.:

$$O_P = \{(u, i) | u \in \mathcal{U}, i \in \mathcal{P}_u\},$$
$$O_N = \{(u, i) | u \in \mathcal{U}, i \in \mathcal{N}_u\},$$
$$O_M = \{(u, i) | u \in \mathcal{U}, i \in \mathcal{I}_u^-\}$$

Let $\alpha = \frac{1}{2\sigma_P^2}, \beta = \frac{1}{2\sigma_N^2}, \gamma = \frac{1}{2\sigma_M^2}$, our objective function in eqn. 2 can be simplified as (the regularization term is omitted):

$$\arg\min_{x^* y^*} \alpha \sum_{(u,i) \in O_P} (1 - \hat{p}_{ui})^2 + \beta \sum_{(u,i) \in O_N} \hat{p}_{ui}^2 + \gamma \sum_{(u,i) \in O_M} \hat{p}_{ui}^2 \tag{6}$$

where α, β, γ are model parameters and can be tuned through validations [24]. Intuitively, we would want $\alpha > \gamma$ and $\beta > \gamma$ to reflect the assumptions that positive and negative feedback are given higher weights than missing observations. However, how to determine the weights for positive feedback and negative feedback is not clear. This is because in some cases we may put higher weights on negative feedback, as recommending items that users negatively interacted with can lead to detrimental outcomes. In other cases where negative feedback signals are weak and noisy, we may want to prioritize positive feedback. Our model is flexible enough to account for these diverse scenarios. Throughout the rest of this paper, we refer to this model as **WRMF-NR**, which is short for **WRMF** with **N**egative feedback **R**e-weighting.

WRMF-NR is a generalization of the classic WRMF [5, 18]. However, setting $\beta = 0$ reduces eqn. 6 to a naive extension of WRMF to

incorporate post-click feedback. This naive model considers "click-complete" as positive feedback and "non-click" as missing feedback, without directly learning from negative feedback (skips). We refer to this baseline model as **WRMF-BL**.

4.4 Pairwise recommendation model

In addition to training recommendation models using pointwise observations, pairwise data [21] is also commonly used in prior literature. For example, under our framework, Bayesian Personalized Ranking (BPR) can be viewed as leveraging the following sets of pairwise data:

$$O_{P_{\text{click}}} := \{(u,i,j)|i \in \mathcal{I}_u^+ \wedge j \in \mathcal{I}_u^-\}, O_{N_{\text{click}}} = O_{M_{\text{click}}} = \emptyset \quad (7)$$

The underlying assumption is that users prefer clicked items over not-clicked ones. Therefore, BPR is optimized to maximize the differences of preference scores between observed and unobserved items, i.e.:

$$\max_{\theta} \sum_{(u,i,j)\in O_{P_{\text{click}}}} \ln \sigma(\hat{p}_{uij}) - \lambda_\theta ||\theta||_f^2, \quad (8)$$

where θ denotes the model parameters, σ denotes the logistic sigmoid function, and $\hat{p}_{uij}$ denotes the preference estimator for (u,i,j) triplets, defined as $\hat{p}_{uij} = \hat{p}_{ui} - \hat{p}_{uj}$. A standard embedding-based collaborative filtering model can be used to predict $\hat{p}_{ui}$ and $\hat{p}_{uj}$, as in eqn. 3. Notice that the vanilla BPR model is a special case of our general framework by setting $\mu_o = 1 - \sqrt{\ln \sigma^{-1}(\hat{p}_{uij})}$, where $\sigma^{-1}(x) = 1 + e^{-x}$.

To improve the pairwise BPR model with post-click data, we extended the triplets defined in eqn. 7 to the following sets:

$$O_P := \{(u,i,j)|i \in \mathcal{P}_u \wedge j \in \mathcal{N}_u\},$$

$$O_N := \{(u,i,j)|i \in \mathcal{I}_u^- \wedge j \in \mathcal{P}_u\},$$

where O_P assumes that user prefer "click-complete" items over "click-skip" ones, and O_N assumes that user prefer "click-complete" items over "not-clicked" ones. The MLE objective in eqn. 2 can then be written as follows (the regularization term is omitted):

$$\arg\max_{\theta} \alpha \sum_{(u,i,j)\in O_P} \ln \sigma(\hat{p}_{uij}) + \beta \sum_{(u,i,j)\in O_N} \ln \sigma(\hat{p}_{uji}), \quad (9)$$

where α, β are tuning parameters that reflect the variances of O_P and O_N under Gaussian assumption. We refer to this model as **BPR-NR**, which is short for **BPR** with **N**egative feedback **R**e-weighting. Similar to WRMF-BL, if we set $\beta = 0$, BPR-NR reduces to a naive model that learns from positive-unobserved item pairs only. We refer to this naive model as **BPR-BL**.

4.5 Implementations

We used the Adam optimizer [9] with a learning rate of 0.001 to optimize for pointwise and pairwise models. To implement the weights proposed in the MLE objectives, we used stratified sampling techniques to over- or under-sample observations of different types. Specifically, to train WRMF-NR, we defined parameters λ_p and λ_n to control the proportion of positive and negative items being sampled into a mini-batch. Tuning these two parameters is equivalent to changing the weights $\alpha, \beta, and \gamma$ in eqn. 6. Similarly, we set $\lambda_{p,n}$ as the hyperparameter to control the proportion of positive-negative pairs in a mini-batch for BPR-NR. Such a resampling technique makes our framework readily adaptable to other types of recommendation algorithms, e.g., content-based [14, 20] or Nerual Network-based recommenders [2, 3], as these models rely on pointwise or pairwise sampling during training. We used the OpenRec [27] framework to implement and evaluate recommendation models. Our code is available at *https://github.com/whongyi/post-click-contentRecSys*.

5 EXPERIMENTS

We evaluated our framework and competitive baseline methods against the post-click-aware ranking metrics described in Section 3.2 (Task 2). Below is a brief summary of the methods used.

- **WRMF**: Standard WRMF implementation proposed by [5].
- **WRMF-BL**: Baseline extension of WRMF to use post-click feedback as corrections for preference labels. Treats "click-complete" as positive feedback and unobserved items as negative feedback.
- **WRMF-NR**: Our proposed approach to probabilistically sample from positive, negative, and unobserved items.
- **BPR**: Standard BPR implementation proposed in [21] using matrix factorization as preference predictor.
- **BPR-BL**: Baseline extension of BPR to incorporate post-click feedback. Treats "click-complete" items as positive feedback and unobserved items as negative feedback during pairwise sampling.
- **BPR-NR**: Our proposed method to explicitly distinguish between positive, negative and unobserved items. Adjusts the weights of positive- negative triplets and positive-unobserved triplets during pair-wise sampling.

5.1 Hyperparameters

We trained all models up to 10K iterations with a mini-batch size of 1000. We used an embedding size of $l = 100$ for both user and item vectors. For BPR-NR with negative re-weighting, we tuned $\lambda_{p,n} \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$. For WRMF-NR, we selected combinations of λ_p and $\lambda_n \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$ that satisfied $\lambda_p + \lambda_n \leq 1$. We tuned the L_2-regularization parameter $\lambda \in \{0.1, 0.01, 0.001, 0.0001\}$. All hyperparameters were tuned on the validations set and the best parameters were selected for evaluations on the test set.

5.2 Distinguishing positive and negative items

Overall, our purposed approach achieves the best performance on both datasets (Table 4). Compared to WRMF, WRMF-NR has a 17.3% relative improvement on ByteDance and 2.2% on Spotify. Similarly, BPR-NR has 19.4% and 2.7% improvements over BPR on ByteDance and Spotify, respectively. Interestingly, WRMF-BL and BPR-BL both show minor improvements over WRMF and BPR, which demonstrates that by simply leveraging user completions as a criteria for positive feedback, the models learn improved user preference on content to some extent. However, our proposed models are shown to learn content-wise user preferences more effectively by explicitly learning from user post-click feedback through re-weighting their impacts on the objective functions.

Method	ByteDance	Spotify
WRMF	0.521	0.540
WRMF-BL	0.549	0.548
WRMF-NR	**0.610**	**0.552**
WRMF-NR v.s. WRMF	+17.3%****	+2.2%****
WRMF-NR v.s. WRMF-BL	+11.1%****	+0.7%***
BPR	0.527	0.541
BPR-BL	0.556	0.547
BPR-NR	**0.633**	**0.555**
BPR-NR v.s. BPR	+19.4%****	+2.7% ****
BPR-NR v.s. BPR-BL	+13.8%****	+1.5%****

Table 4: Performance of ranking "click-complete" against "click-skip" items using baseline models and our proposed models. Bold numbers indicate the best performance achieved under pointwise and pairwise models, respectively. Asterisks represent the significance level under t-test (*: $p < 0.001$, ****: $p < 10^{-11}$). Our proposed approach, which incorporates post-click feedback, shows improvement over models and baseline extensions that only learn from clicks.**

Performance improvements on ByteDance are more significant than on Spotify. This indicates that the effectiveness of skips behavior as post-click preference indicators could be dependent on specific content domains. We speculate that since videos require more engagement than audio, skips are stronger indicators of negative preference in video watching than in music listening. Further research is needed to comprehensively understand skip behavior across various content domains.

5.3 Effectiveness of post-click feedback

To understand which factors contribute to performance improvements, we examined the rankings of positive and negative items by different algorithms. An ideal recommender would rank positive items higher and negative items lower. Table 5 illustrates the rankings (AUC) produced by baseline models and our proposed models. For BPR and BPR-BL, we observed only a minor ranking difference between positive and negative items. This means BPR and its naive extension hardly learn from negative feedback, as they still rank negative items very high in the recommendation lists. In real-world applications, this means that these recommendations have an almost equal chance of being liked or disliked by users, which could be an explanation for the high skip rates in both datasets. In contrast, our proposed BPR-NR model is able to push the rankings of negative items below positive items by increasing the weights on negative feedback signals. With $\lambda_{p,n} = 0.8$, the AUC for negative items on ByteDance is 0.519, which is close to the random guessing AUC of 0.5. Similar observations are found for WRMF and their extensions.

At the same time, the rankings for positive items go down as we increase $\lambda_{p,n}$ and λ_n. Such a trade-off is expected as weights for positive-unobserved pairs are lowered. In those cases, positive items have a higher chance of showing up in top recommendations than negative items, while some unobserved items may also be presented. This is not necessarily a bad outcome, as such explorations may prevent recommendation systems from accelerating the degeneration of user interests [6] and alleviate filtering bubble problems [4, 17]. Our framework provides flexible controls of the trade-off between exploration and exploitation in recommendations.

6 DISCUSSION AND FUTURE WORK

In this section, we discuss the implications of our findings and directions for future work, particularly on understanding post-click feedback across content domains and other available user feedback beyond clicks and skips.

Data sparsity. User feedback are sparse for training recommendation systems, especially for deep-learning-based systems. Implicit feedback (e.g. clicks) are preferred over explicit feedback (e.g. ratings), due to their large volume and reduced bias. Such an advantage also manifests in post-click feedback, as they can be collected with implicit feedback without imposing extra burdens on users. In other words, post-click feedback do not increase data sparsity and capture finer-grained user preferences.

Skips in various content domains. As we observe from the above analyses, skipping behavior can be context-dependent, and their strengths as negative preference indicators vary across content domains. Our proposed framework learns more effectively from user skips in short videos than in music. Such differences may be associated with the intrinsic properties of video and audio as different media. We note that podcasts [22, 28, 30] may be an attractive domain for further work on skip signals. User engagement levels for podcasts lie somewhere between video and music, since podcasts are presented in audio but require more user attention than music. User skips could also be very informative in understanding user preferences on podcasts.

Negative feedback beyond user skips. In this paper, we specifically investigated user skips as negative post-click feedback for content recommendations. There exists a broader range of user behavior that capture negative user preferences; for example, social network platforms such as Twitter and Instagram enable users to flag items that they are "not interested" in. On e-commerce platforms (e.g., on Amazon), users can identify specific items as "do not use for recommendations." Our proposed approach should be able to incorporate those types of user feedback as well. Future work could systematically characterize different types of feedback captured in user interactions with recommendation systems and propose novel learning algorithms to better understand user preferences beyond clicks.

7 CONCLUSION

In this work, we highlight the mismatches between user clicks and user preferences in content recommendation. We show empirical evidence of how existing collaborative filtering algorithms that focus on optimizing user clicks fail to predict user post-click engagement. To address such misalignment, we propose a framework that incorporates post-click feedback such as skips. Through investigations of real-world music and short video datasets, we demonstrate how our framework can be applied to improve existing pointwise and pairwise recommendation models. We hope this

Model	ByteDance			Spotify		
	AUC_p	AUC_n	$AUC_p - AUC_n$	AUC_p	AUC_n	$AUC_p - AUC_n$
WRMF	0.778	0.760	0.018	0.937	0.930	0.007
WRMF-BL	0.791	0.751	0.040	0.915	0.905	0.011
WRMF-NR($\lambda_p = 0.2, \lambda_n = 0.4$)	0.706	0.666	0.041	0.753	0.733	0.020
WRMF-NR($\lambda_p = 0.2, \lambda_n = 0.6$)	0.759	0.682	0.076	0.855	0.835	0.020
WRMF-NR($\lambda_p = 0.2, \lambda_n = 0.8$)	0.737	0.678	0.059	0.829	0.805	0.024
WRMF-NR($\lambda_p = 0.4, \lambda_n = 0.2$)	0.709	0.634	0.074	0.907	0.894	0.013
WRMF-NR($\lambda_p = 0.4, \lambda_n = 0.4$)	0.783	0.745	0.038	0.870	0.851	0.019
WRMF-NR($\lambda_p = 0.4, \lambda_n = 0.6$)	0.661	0.574	0.086	0.841	0.820	0.021
BPR	0.806	0.785	0.021	0.923	0.915	0.008
BPR-BL	0.804	0.760	0.044	0.915	0.903	0.012
BPR-NR($\lambda_{p,n} = 0.2$)	0.785	0.728	0.057	0.898	0.884	0.014
BPR-NR($\lambda_{p,n} = 0.4$)	0.759	0.682	0.076	0.884	0.867	0.018
BPR-NR($\lambda_{p,n} = 0.6$)	0.717	0.614	0.103	0.861	0.839	0.021
BPR-NR($\lambda_{p,n} = 0.8$)	0.651	0.519	0.131	0.807	0.778	0.029
BPR-NR($\lambda_{p,n} = 1.0$)	0.567	0.429	0.138	0.546	0.512	0.034

Table 5: Ranking of positive items ($\mathcal{P}_u$) and negative items ($\mathcal{N}_u$) against unobserved items($\mathcal{I}_u^-$). Higher AUC_p and lower AUC_n is better. $AUC_p - AUC_n$ indicates the ranking performance differences between positive items and negative items. We observed larger margins on ByteDance compared to Spotify, which explains why the improvements on ByteDance are much more significant.

work will inspire future studies on understanding comprehensive user feedback in recommendation systems across various content, as well as applications in other domains.

ACKNOWLEDGMENTS

This research was supported by the National Science Foundation under grant IIS-1700832 and by Yahoo Research (via the Connected Experiences Laboratory at Cornell Tech). The work was further supported by the small data lab at Cornell Tech, which received funding from NSF, NIH, RWJF, UnitedHealth Group, Google, and Adobe. We thank the anonymous reviewers for their insightful comments and suggestions.

REFERENCES

[1] Brian Brost, Rishabh Mehrotra, and Tristan Jehan. 2019. The Music Streaming Sessions Dataset. In *Proceedings of the 2019 Web Conference*. ACM.

[2] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. ACM, 191–198.

[3] 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*. International World Wide Web Conferences Steering Committee, 173–182.

[4] Cheng-Kang Hsieh, Longqi Yang, Honghao Wei, Mor Naaman, and Deborah Estrin. 2016. Immersive recommendation: News and event recommendations using personal digital traces. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 51–62.

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

[6] Ray Jiang, Silvia Chiappa, Tor Lattimore, Andras Agyorgy, and Pushmeet Kohli. 2019. Degenerate Feedback Loops in Recommender Systems. *arXiv preprint arXiv:1902.10730* (2019).

[7] Thorsten Joachims, Laura A Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2005. Accurately interpreting clickthrough data as implicit feedback. In *Sigir*, Vol. 5. 154–161.

[8] Youngho Kim, Ahmed Hassan, Ryen W White, and Imed Zitouni. 2014. Modeling dwell time to predict click-level satisfaction. In *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, 193–202.

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

[10] Noam Koenigstein, Gideon Dror, and Yehuda Koren. 2011. Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy. In *Proceedings of the fifth ACM conference on Recommender systems*. ACM, 165–172.

[11] Mounia Lalmas, Janette Lehmann, Guy Shaked, Fabrizio Silvestri, and Gabriele Tolomei. 2015. Promoting positive post-click experience for in-stream yahoo gemini users. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1929–1938.

[12] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on Intelligent user interfaces*. ACM, 31–40.

[13] Nathan N Liu, Min Zhao, and Qiang Yang. 2009. Probabilistic latent preference analysis for collaborative filtering. In *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 759–766.

[14] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. 2011. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*. Springer, 73–105.

[15] Hongyu Lu, Min Zhang, and Shaoping Ma. 2018. Between Clicks and Satisfaction: Study on Multi-Phase User Preferences and Satisfaction for Online News Reading. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 435–444.

[16] Andriy Mnih and Ruslan R Salakhutdinov. 2008. Probabilistic matrix factorization. In *Advances in neural information processing systems (NIPS '08)*. 1257–1264.

[17] Tien T Nguyen, Pik-Mai Hui, F Maxwell Harper, Loren Terveen, and Joseph A Konstan. 2014. Exploring the filter bubble: the effect of using recommender systems on content diversity. In *Proceedings of the 23rd international conference on World wide web*. ACM, 677–686.

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

[19] Arkadiusz Paterek. 2007. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop*, Vol. 2007. 5–8.

[20] Michael J Pazzani and Daniel Billsus. 2007. Content-based recommendation systems. In *The adaptive web*. Springer, 325–341.

[21] 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)*.

AUAI Press, 452–461.
[22] Manos Tsagkias, Martha Larson, and Maarten De Rijke. 2010. Predicting podcast preference: An analysis framework and its application. *Journal of the American Society for information Science and Technology* 61, 2 (2010), 374–391.
[23] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep content-based music recommendation. In *Advances in neural information processing systems*. 2643–2651.
[24] Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 448–456.
[25] Xuanhui Wang, Hui Fang, and ChengXiang Zhai. 2008. A study of methods for negative relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 219–226.
[26] Byoungju Yang, Sangkeun Lee, Sungchan Park, and Sang-goo Lee. 2012. Exploiting various implicit feedback for collaborative filtering. In *Proceedings of the 21st International Conference on World Wide Web*. ACM, 639–640.
[27] Longqi Yang, Eugene Bagdasaryan, Joshua Gruenstein, Cheng-Kang Hsieh, and Deborah Estrin. 2018. OpenRec: A Modular Framework for Extensible and Adaptable Recommendation Algorithms. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 664–672.
[28] Longqi Yang, Michael Sobolev, Christina Tsangouri, and Deborah Estrin. 2018. Understanding user interactions with podcast recommendations delivered via voice. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 190–194.
[29] Longqi Yang, Yu Wang, Drew Dunne, Michael Sobolev, Mor Naaman, and Deborah Estrin. 2019. More than Just Words: Modeling Non-textual Characteristics of Podcasts. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. ACM.
[30] Longqi Yang, Yu Wang, Drew Dunne, Michael Sobolev, Mor Naaman, and Deborah Estrin. 2019. More Than Just Words: Modeling Non-Textual Characteristics of Podcasts. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. ACM, 276–284.
[31] Xing Yi, Liangjie Hong, Erheng Zhong, Nanthan Nan Liu, and Suju Rajan. 2014. Beyond clicks: dwell time for personalization. In *Proceedings of the 8th ACM Conference on Recommender systems*. ACM, 113–120.
[32] Yisong Yue, Rajan Patel, and Hein Roehrig. 2010. Beyond position bias: Examining result attractiveness as a source of presentation bias in clickthrough data. In *Proceedings of the 19th international conference on World wide web*. ACM, 1011–1018.
[33] Aston Zhang, Amit Goyal, Weize Kong, Hongbo Deng, Anlei Dong, Yi Chang, Carl A Gunter, and Jiawei Han. 2015. adaqac: Adaptive query auto-completion via implicit negative feedback. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 143–152.
[34] Qian Zhao, F Maxwell Harper, Gediminas Adomavicius, and Joseph A Konstan. 2018. Explicit or implicit feedback? engagement or satisfaction?: a field experiment on machine-learning-based recommender systems. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*. ACM, 1331–1340.