Management Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Adaptive Data Acquisition for Personalized Recommendations with Optimality Guarantees on Short-Form Video Platforms

Junyu Cao, Yan Leng

To cite this article:
Junyu Cao, Yan Leng (2025) Adaptive Data Acquisition for Personalized Recommendations with Optimality Guarantees on Short-Form Video Platforms . Management Science

Published online in Articles in Advance 25 Aug 2025

. https://doi.org/10.1287/mnsc.2022.01130

https://pubsonline.informs.org/journal/mnsc

MANAGEMENT SCIENCE
Articles in Advance, pp. 1–19
ISSN 0025-1909 (print), ISSN 1526-5501 (online)

Adaptive Data Acquisition for Personalized Recommendations with Optimality Guarantees on Short-Form Video Platforms

Junyu Cao,[a,*] **Yan Leng**[a,*]

[a] McCombs School of Business, The University of Texas at Austin, Austin, Texas 78712
*Corresponding authors
Contact: junyu.cao@mccombs.utexas.edu, https://orcid.org/0000-0001-9235-1411 (JC); yan.leng@mccombs.utexas.edu, https://orcid.org/0000-0002-7084-2700 (YL)

Received: April 13, 2022
Revised: April 3, 2023; March 1, 2024
Accepted: May 24, 2024
Published Online in Articles in Advance: August 25, 2025

https://doi.org/10.1287/mnsc.2022.01130

Abstract. The recent surge in the popularity of short-form video (SFV) on digital platforms has led to massive numbers of videos and ever-evolving topics. As a result, the task of making personalized recommendations has become increasingly challenging. We introduce a new pure exploration problem on SFV platforms: finding a $(K, \varepsilon^H, \varepsilon^L)$-optimal set that includes all recommendations within the ε^L-optimality gap and that excludes those beyond the ε^H-optimality gap relative to the best arm with a capacity limit of K. To solve this problem, we propose an algorithm called adaptive acquisition tree (AAT). AAT jointly accounts for user preference heterogeneity and high-dimensional product characteristics. It adaptively segments users and then, learns a personalized transductive policy that can be used on partially observed or even unobserved card types to accommodate the dynamic trends on SFV platforms. We derive the sample complexity required to identify a $(K, \varepsilon^H, \varepsilon^L)$-optimal set. Our method's efficiency is validated through numerical tests using data from the NetEase platform. Our results reveal that the proposed policy performs significantly better than several state-of-the-art benchmarks across four transductive scenarios for both spotlight recommendations (i.e., best-arm identifications) and $(K, \varepsilon^H, \varepsilon^L)$-optimal set recommendations. Compared with the best benchmarks for the best card and $(K, \varepsilon^H, \varepsilon^L)$-optimal set recommendations, our approach can elevate the average rewards (measured by view time) by 30% (to 100%) and 43% (to 56%), respectively. Given the increasing popularity and uniqueness of SFVs and more broadly, user-generated content, our method offers significant academic and practical merit.

History: Accepted by Omar Besbes, revenue management and market analytics.
Funding: Y. Leng is supported by the U.S. National Science Foundation (NSF) under [Grant IIS 2153468].
Supplemental Material: The online appendix and data files are available at https://doi.org/10.1287/mnsc.2022.01130.

Keywords: pure exploration • information acquisition • transductive learning • short-form video platforms • recommender systems

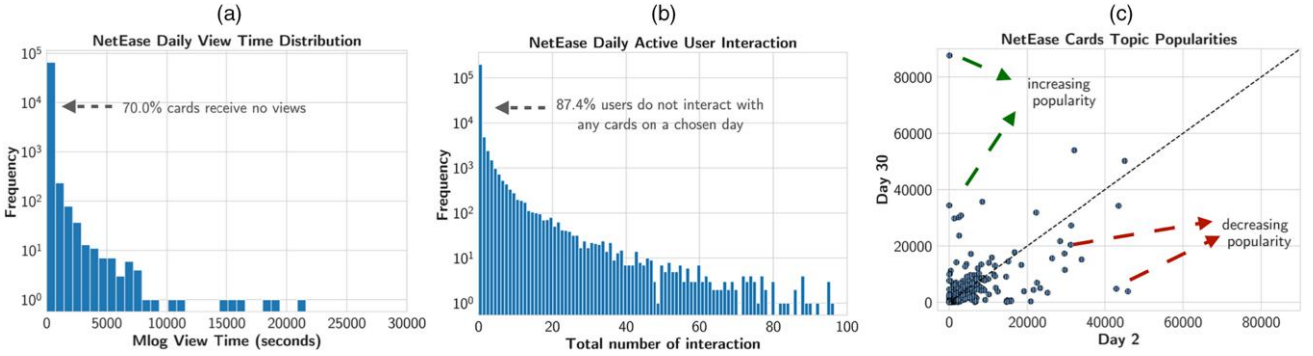
1. Introduction

The recent surge in the popularity of short-form video (SFV) is rapidly reshaping digital content creation, dissemination, and consumption. Users are continuously exposed to an influx of new and evolving videos on platforms like TikTok and NetEase.[1] As of the fall of 2020, TikTok reported an estimated 850 million active users a month in China (Lorenz 2020) and 100 million each month in the United States (Wallaroo 2021). These two platforms provide significant new marketing opportunities as evidenced by the $2.1 billion worth of ads sold in China in 2018 (Cheung 2020). Following the success of TikTok, an increasing number of companies have entered the lucrative arena.

Personalized recommendations, which account for consumer preference heterogeneity, play a vital role on digital platforms. They help reduce user search costs (Liu et al. 2021) and boost user engagement (Kumar and Hosanagar 2019). However, the overwhelming number of SFVs and only sparse user interactions with the breadth of the content pose significant challenges for the learning problem.[2] For platforms that recently launched the SFV service, historical user responses are even more deficient. For example, on NetEase, 70% of music video cards receive no views on a given day[3] (see Figure 1(a)). Also, 87.4% of users do not interact with any cards on SFV platforms on a given day (see Figure 1(b)). Meanwhile, the proliferation of new topics and content leads to a shorter time frame to learn user preferences. Poorly designed recommender systems could cause platforms to lose users and content creators (Besbes et al. 2016, Cao and Sun 2019, Cao et al. 2019, Leng et al. 2020, Bastani et al. 2022b), especially as the competition in the SFV industry intensifies.

These developments have created an urgent need for SFV platforms to acquire information effectively in order to understand user preferences, identify top SFV cards, and train recommendation models. SFV

Figure 1. (Color online) Challenges in Learning User Preferences on the NetEase Platform



Notes. Panels (a) and (b) show histograms of the music video cards' daily view time and users' daily total view time, respectively. Panel (c) reveals the evolving trends on NetEase, showing the number of cards created within each topic category. The x and y axes show the number of cards for each topic generated by content creators during the first seven days and last seven days of November 2019. Topics with decreasing popularity fall below the diagonal line; topics gaining in popularity are situated above it.

platforms can learn the optimal recommendation policy on a random user bucket for a short period before using a pure exploration framework to launch the policy at full scale on the experimentation platform. However, the application of these bandits-based frameworks for making recommendations about SFVs faces unique challenges.

First, content evolves rapidly on SFV platforms as new trends emerge and old trends fade. For instance, SFVs may change as new products enter the market (e.g., movies, games, music, and tech products), as the latest news appears, or during holiday seasons. When machine learning models are trained using existing data, they often struggle to adapt to such changes because of differences in feature distributions between future and existing SFVs. Figure 1(c) demonstrates the dynamic trends of card creation on the NetEase platform. Topics above the dashed diagonal line in Figure 1(c) are gaining popularity, and those below the dashed diagonal line in Figure 1(c) are losing attention. Topics on the horizontal line at $y = 0$ in Figure 1(c) were popular during the first week of November 2019 but saw limited creation in the last week of the month. Conversely, points on the vertical line at $x = 0$ in Figure 1(c) indicate emerging topics. These observations from Figure 1(c) indicate the limitations of non-transductive models,[4] which may falter when new topics emerge as a result of shifts in feature distribution. However, if the topic changes can be anticipated—for example, through news releases, popular trends on other social media platforms, or the holiday season—the platform can use these experimentation platforms to learn optimal recommendation policies.

Second, similar to other user-generated content (UGC) and business-to-consumer two-sided marketplaces, quality control on the content is important because of the opportunity costs and potential harm of off-target recommendations. High-quality content increases the welfare of all platform participants (Shi and Raghu 2020). However, different from B2C platforms,[5] content quality is hard to control on UGC platforms because of the volume of UGC and the absence of regulations and reputation mechanisms for users. Low-quality content on social media platforms can have detrimental effects on users, ranging from minor annoyances to more severe consequences. For example, clickbait and information that does not match users' preferences can lead to frustration, ultimately driving them away from the platform; unethical content, such as hate speech and misinformation, can create a hostile online environment and lead to psychological harm (Matamoros-Fernández and Farkas 2021).

In this paper, we introduce a new pure exploration problem involving data acquisition in recommender systems for identifying an optimal set of cards for recommendations. This set, which we call the $(K, \varepsilon^H, \varepsilon^L)$-optimal set, is a selection of cards that have a specific precision (ε^L) and tolerance level (ε^H) that maximize user engagement. Given the ever-changing nature of platforms' content—stemming from current events, new product releases, and holiday seasons—adaptability is an imperative for platforms. Our approach learns user preferences during an exploration period and subsequently designs personalized card recommendation policies based on expected feature distributions during an exploitation period. Our approach allows for maintaining control over recommendation quality by adjusting the precision and tolerance levels, thereby determining the standards for selecting up to K cards. To achieve this control, we introduce an adaptive data acquisition strategy called an adaptive acquisition tree (AAT). This strategy adaptively gathers samples, splits the feature space, and excludes suboptimal cards, thereby accounting for user and card heterogeneity as well as covariate shifts in a transductive setting. We apply the adaptive acquisition tree to the NetEase platform and assess its performance in four distinct transductive scenarios. Our

findings demonstrate the effectiveness of the proposed algorithm in improving card recommendations. Our main contributions are summarized as follows.

I. We formulate a new data acquisition problem for the recommender systems operating on the SFV platforms. This problem is designed to tackle two challenges: the rapid evolution of trending topics and the severe consequences associated with low-quality content. To address these challenges, we introduce a novel pure exploration problem tailored for the top-arm identification and contextual bandit literature. Our approach seeks to identify all arms within a specific precision level (managed by ε^L) while excluding arms below a particular quality level (controlled by ε^H). Thus, our paper broadens the scope of recommendation and online learning literature, extending it to encompass the emerging internet phenomenon of SFVs while also addressing their unique hurdles.

II. We create an interpretable algorithm, AAT, that is designed to adaptively acquire information to determine the $(K, \varepsilon^H, \varepsilon^L)$-optimal set while maintaining quality control. AAT boasts two main advantages. First, it simultaneously accounts for the heterogeneity in user preferences and card features, improving on existing contextual bandit problems for learning the best or top-K arms for homogenous user preferences (Soare et al. 2014). To the best of our knowledge, our study is the first to account for user heterogeneity amid covariate shifts. Second, AAT is interpretable because it allows decision makers to easily trace how user preferences influence the optimal strategy in the acquisition tree. Our study thus contributes to the interpretable machine learning literature by proposing an explanatory and transparent data acquisition method.

III. To demonstrate the efficacy of AAT, we apply it to data acquisition for recommendations on the NetEase platform. Our findings reveal that the proposed algorithm substantially enhances the performance of spotlight and $(10, \varepsilon^H, \varepsilon^L)$-optimal set recommendations across four transductive settings when compared with several state-of-the-art benchmarks. In all scenarios, we are able to improve the average reward by 30%~100% and 43%~56% compared with the benchmarks for the best card recommendation and best $(10, \varepsilon^H, \varepsilon^L)$-optimal set recommendation, respectively.

The remainder of the paper is organized as follows. Section 2 reviews the relevant literature. The problem formulation and the theoretical results are presented in Section 3. We develop the algorithm and provide the sample complexity proof in Section 4. Section 5 introduces the empirical evaluation using the NetEase platform. Section 6 concludes.

2. Literature Review

This section reviews related research in the context of the multi-armed bandit (MAB) problem and clustering and tree algorithms for market segmentation to account for user heterogeneity.[6]

2.1. Multi-armed Bandits

Our study is situated within MAB research (see Lattimore and Szepesvári 2020 for a comprehensive review), with a focus on best-arm identification problems (Even-Dar et al. 2006, Gabillon et al. 2012, Karnin et al. 2013, Soare et al. 2014, Chen and Li 2015, Kaufmann et al. 2016, Russo 2016, Xu et al. 2018, Jedra and Proutiere 2020, Kazerouni and Wein 2021). The best-arm identification problem, which seeks to use the fewest samples to identify the best arm, is pertinent to management science, computer science, and operation management for personalization in a context of sparse data.

Our problem is also related to top-K arm identification, which seeks to identify the best K arms with high confidence while minimizing the sample size (Kalyanakrishnan and Stone 2010; Bubeck et al. 2013; Chen et al. 2014, 2017a, b; Jiang et al. 2017). Variants of the best-arm identification problem emerge in several studies, including in Abernethy et al. (2016), which aims to identify all arms above a threshold. In addition, Mason et al. (2020) aim to identify arms within a certain optimal gap relative to the best arm, and Ren et al. (2019) study the problem of identifying any K distinct arms among the top ρ fraction.

Pure exploration problems, a subfield in MAB research, usually assume identical historical and future data distributions, which may not hold true in reality because of covariate shift (Storkey and Sugiyama 2007, Gretton et al. 2009, Gelada and Bellemare 2019). Our problem formulation builds on a transductive linear bandit (Fiez et al. 2019), whereby exploitation stage cards can vary from acquisition stage cards as in transductive experiment designs (Yu et al. 2006). We account for user preference heterogeneity, thus departing from works that assume homogenous user preferences, by using a newly developed $(K, \varepsilon^H, \varepsilon^L)$-optimal set identification framework for SFV platforms. In doing so, our problem formulation also generalizes best-arm identification.

2.2. Clustering and Tree Algorithms for Market Segmentation

2.2.1. Clustering Algorithm. Clustering is a proven technique to enhance the performance of downstream tasks by leveraging consumer heterogeneity (Yang et al. 2016, Jagabathula et al. 2018; see also Bastani et al. 2022a for an overview). Interesting works use clustering technique for MABs to improve performance (Gentile et al. 2014, 2017; Nguyen and Lauw 2014). For example, in

the operations management literature, such works address demand forecasting, assortment personalization, and dynamic pricing. In the demand forecasting literature, clustering techniques help to improve demand function estimations by leveraging products with similar features or users with similar characteristics (Baardman et al. 2018, Hu et al. 2019, Cohen et al. 2022). In the assortment literature, Bernstein et al. (2019) consider dynamic personalized assortment optimization by adapting and adjusting customer segments using online transaction data and by estimating user preferences. A growing interest in using clustering is also emerging in the pricing literature. Ferreira et al. (2016) classify the demand of all products into multiple groups using historical information and offline optimization. Cheung et al. (2017) estimate demand functions based on clusters of user behaviors in an offline manner and then, dynamically price another product by learning which clusters its demand belongs to. Extending the learning of cluster structures to an online fashion, Miao et al. (2022) and Keskin et al. (2024) study context-based dynamic pricing, with clustering done on the fly and across products. In particular, Keskin et al. (2024) design a data-driven policy integrating spectral clustering and feature-based pricing. Although most studies focus on either product or user feature data, we integrate both. Moreover, instead of heuristic splitting criteria or unsupervised learning, we employ a tree-based framework that uses supervised learning to weight features according to their predictive power. This nonparametric and interpretable method works well without the need for handcrafted features.

2.2.2. Tree Algorithm. Tree algorithms have been found to have broad application in forecasting and segmentation (such as in linear model trees (Quinlan 1992) and logistic model trees (Chan and Loh 2004, Landwehr et al. 2005)), and they have gained in popularity because of their simplicity and interpretability (Elmachtoub et al. 2017, Mišić 2020). Chen et al. (2019) and Chen and Mišić (2022) propose tree-based discrete choice models. Meanwhile, Ban et al. (2019) employ the residual tree method, which extends the classic scenario tree method of stochastic programming (Shapiro et al. 2014); this method estimates parameters of the demand model using historical sales data of old products. In addition, Aouad et al. (2023) recently have used the tree structure for market segmentation. Similarly, in this paper, we rely on user interactions with the platform to guide segmentation rather than in an unsupervised way (e.g., as in clustering algorithms discussed above). However, in contrast to all of the works mentioned above, which use static prediction for decision making, we extend the work to a dynamic data acquisition context, where arriving consumers and the data acquisition policy are continuously updated.

3. Model

In this section, we formally introduce our model. We study a pure exploration problem with two specific objectives: (1) to identify all arms (up to K) that are within a small optimality gap (ε^L) relative to the best arm and (2) to exclude all arms that fall outside of a larger optimality gap (ε^H) as a means to maintain quality control. We call this set the $(K, \varepsilon^H, \varepsilon^L)$-optimal set and formally define it in Section 3.3. In addition, we propose an implementable sufficient condition and discuss a sequential card removal and optimal card set identification process in Section 3.4.

3.1. Problem Setup

We divide the time horizon into two stages: the *exploration* stage (i.e., information acquisition) and the *exploitation* stage (i.e., implementation). During the exploration stage, users in a chosen bucket in the experimentation platform arrive sequentially, with features drawn independently and identically from distribution $\mathcal{D}(\cdot)$ in the feature space $\mathcal{U} \subseteq \mathbb{R}^{d_u}$, where $d_u \in \mathbb{N}^+$. At the beginning of round $t = 1, \ldots, T$, the platform observes user feature $U_t \in \mathcal{U}$. It then chooses an arm X_t from a set $\mathcal{X}$ that lies in $\mathbb{R}^{d_p}$ with cardinality $|\mathcal{X}|$, where $d_p \in \mathbb{N}^+$. Finally, the platform observes reward R_t.[7] For notational convenience, we use uppercase letters to denote random variables and lowercase letters to indicate particular realizations of a random variable. For easy reference, we include all notations in Online Appendix C. Let $\mathcal{H}_t = \sigma(U_1, X_1, R_1, \ldots, U_{t-1}, X_{t-1}, R_{t-1}, U_t, X_t)$ be the σ-algebra summarizing the information available just before R_t is observed. The statistical relationship between the response variable R_t and the explanatory variables (U_t, X_t) is modeled as

$$R_t(U_t, X_t) = X_t^\top \theta(U_t) + \epsilon_t \quad \text{for } t = 1, \ldots, T,$$

where the noise ϵ_t is $\mathcal{H}_t$ measurable and $\mathbb{E}[\epsilon_t | \mathcal{H}_t] = 0$. The expected reward of recommending card x to user u is $r(u, x) = \mathbb{E}[R(u, x)] = x^\top \theta(u)$. The underlying parameter $\theta(u) \in \mathbb{R}^{d_p}$ is unknown, and it can vary across users. Models assuming that users are homogenous may lead to model misspecification, which can result in a significant loss of revenue. We illustrate this issue in Example D1 in Online Appendix D.

Following a common assumption in the bandit literature on dynamic environments, we assume that $\theta(u)$ remains unchanged between the exploration and exploitation periods.[8] This assumption is reasonable because the exploration period is relatively short. If platform decision makers believe that consumer preferences have changed, they can implement our method again to learn the new consumer preferences.

If the card set $\mathcal{X}$ in the exploration stage is equal to the card set $\mathcal{Z}$ in the exploitation stage, identifying the best card is known as the best-arm identification

problem (Soare et al. 2014, Xu et al. 2018). However, the feature distribution of cards could vary significantly from the exploration stage ($\mathcal{X}$) to the exploitation stage ($\mathcal{Z}$); this variation is known as the covariate shift (Storkey and Sugiyama 2007, Gretton et al. 2009). We use the following examples to illustrate motivations for transductive scenarios on SFV platforms.

a. $\mathcal{X} \subset \mathcal{Z}$. As new products launch on the market (e.g., movie, game, music, or tech product) or a new holiday season approaches, platforms may expect new card types to emerge. As these new features might not be available on the platform yet, there could be significant interest for platforms to understand consumer preferences regarding these new cards. This leads to $\mathcal{X} \subset \mathcal{Z}$ (see Figure 2(a)). Consider the example of a holiday season, during which users typically have a (fixed) preference for timely holiday-related topics. For instance, when Thanksgiving approaches, users start to be more interested in content related to Thanksgiving (e.g., traditional recipes, history, good travel deals). Therefore, before the specific holiday season, platforms can learn consumer preferences on these topics by building an artificial card set $\mathcal{Z}$, with feature distribution learned from the previous year and with adaptation to popular trends of the current year.

b. $\mathcal{Z} \subset \mathcal{X}$. As the features of cards change, some of the old features may no longer be trending. An SFV platform may use a larger measurement set $\mathcal{X}$ to learn policies only on cards with trendy features in $\mathcal{Z}$ (see Figure 2(b)). Because they can provide information about the underlying parameters, the outdated cards in $\mathcal{X}$ may help in learning the optimal set in $\mathcal{Z}$. After the holiday season, as holiday-related topics vanish, platforms can again learn user preferences by removing cards related to the holiday from $\mathcal{X}$; they would do so using the same holiday example described above.

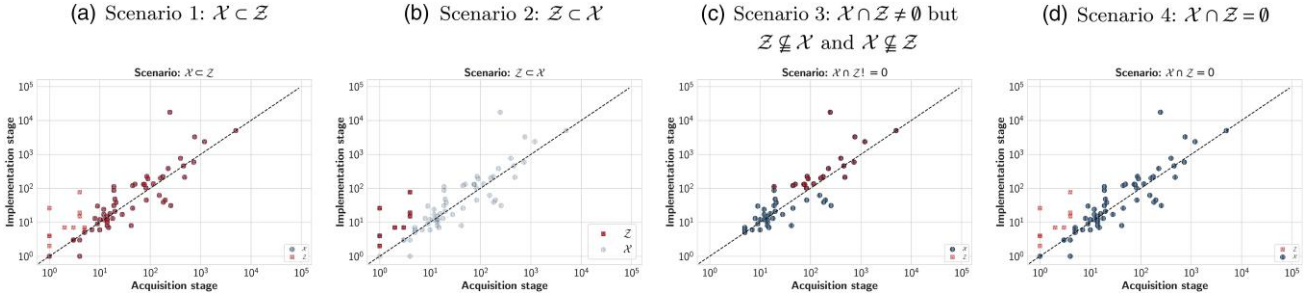
c. $\mathcal{X} \cap \mathcal{Z} = \emptyset$. Figure 2(c) illustrates a scenario in which future card $\mathcal{Z}$ and exploration card $\mathcal{X}$ do not overlap at all. Practical examples for this scenario include learning user preferences in relation to heterogeneous cards with advertisements. This example has important applications for SFV platforms.[9] Platforms may identify some features in a card that can be used for advertising decisions ($\mathcal{Z}$), taking inspiration from cards involving organic, nonadvertising contexts ($\mathcal{X}$), which follow $\mathcal{X} \cap \mathcal{Z} = \emptyset$. Because of the different natures of the cards (advertisement versus nonadvertisement), the cards in the two periods are entirely different.

d. $\mathcal{X} \cap \mathcal{Z} \neq \emptyset$ but $\mathcal{Z} \nsubseteq \mathcal{X}$ and $\mathcal{X} \nsubseteq \mathcal{Z}$. Figure 2(d) illustrates a scenario in which the exploitation card sets and exploration card sets have some overlap. For instance, new card topics may emerge after a new product launches (e.g., movie, game, music, or tech product). Typically, such product launches are expected because firms pitch the product through press releases or commercial advertisements (e.g., trailers) to get media attention. After these events, textual discussions may appear on discussion websites or UGC platforms (e.g., Reddit, Quora, or Diggs). SFV platforms can collect information from these information sources and then, design artificial cards in $\mathcal{Z}$.

These four scenarios on SFV platforms motivate a transductive bandit framework. In the exploration stage, we collect data to learn unknown parameters. As the implementation stage begins, the platform determines a selection strategy for each incoming user based on a mapping from the user feature space $\mathcal{U}$ to a card subset in $\mathcal{Z}$. Specifically, a selection strategy $\pi \in \Pi$ is a function that $\pi : \mathcal{U} \to \mathcal{Z}^K$, where K is the cardinality of the selection; that is, a selection of cards for user u is denoted as $\pi(u) \subseteq \mathcal{Z}^K$. If the content evolves sequentially over time (from $\mathcal{X}$ to $\mathcal{Z}$ to $\mathcal{Z}'$), our method can serve as a building block (i.e., we study one transition), and the platform can implement the method on a user bucket every time that a change is expected to happen using the experimentation platform.

Next, we formally define the $(K, \varepsilon^H, \varepsilon^L)$-optimal set in Definition 1. Parameters ε^H and ε^L describe the tolerance

Figure 2. (Color online) Transductive Scenarios for (a) $\mathcal{X} \subset \mathcal{Z}$, (b) $\mathcal{Z} \subset \mathcal{X}$, (c) $\mathcal{X} \cap \mathcal{Z} = \emptyset$, and (d) $\mathcal{X} \cap \mathcal{Z} \neq \emptyset$ but $\mathcal{Z} \nsubseteq \mathcal{X}$ and $\mathcal{X} \nsubseteq \mathcal{Z}$



Notes. The x and y axes correspond to the number of cards that were created for each topic in the exploration and exploitation stages, respectively. $\mathcal{X}$ and $\mathcal{Z}$ are represented by blue circles and red squares, respectively. The regions above and below the diagonal reference line correspond to topics with increasing and decreasing popularity among creators, respectively. (a) Scenario 1: $\mathcal{X} \subset \mathcal{Z}$. (b) Scenario 2: $\mathcal{Z} \subset \mathcal{X}$. (c) Scenario 3: $\mathcal{X} \cap \mathcal{Z} = \emptyset$. (d) Scenario 4: $\mathcal{X} \cap \mathcal{Z} \neq \emptyset$ but $\mathcal{Z} \nsubseteq \mathcal{X}$ and $\mathcal{X} \nsubseteq \mathcal{Z}$.

and precision levels, respectively. SFV platforms want to detect all "good-enough" cards within the precision level and avoid recommending suboptimal cards outside of the tolerance level. Specifically, ε^H is used to control the quality of the recommendation as motivated in Section 1.

Definition 1 (($K, \varepsilon^H, \varepsilon^L$)-Optimal Set). Fix user $u \in \mathcal{U}$. A set $S(u) \subseteq \mathcal{Z}$ is $(K, \varepsilon^H, \varepsilon^L)$-optimal with respect to set $\mathcal{Z}$, where $\varepsilon^H > \varepsilon^L$, if it satisfies all of the following.

I. For any $z \in S(u)$, $\max_{z' \in \mathcal{Z}} r(u, z') - r(u, z) \le \varepsilon^H$.

II. For any $z \in \mathcal{Z}$, such that $\max_{z' \in \mathcal{Z}} r(u, z') - r(u, z) \le \varepsilon^L$, it holds that $z \in S(u)$.

III. The cardinality constraint $1 \le |S(u)| \le K$.

Intuitively, $S(u)$ is $(K, \varepsilon^H, \varepsilon^L)$ optimal if $S(u)$ contains all cards within the ε^L-optimality gap with respect to the optimal card; meanwhile, all cards in $S(u)$ are within the ε^H-optimality gap with respect to the optimal card. Including only the cards within the ε^H-optimality gap with respect to the optimal card is similar to the minimum quality standard developed by Ronnen (1991), whereas the quality of items needs to be inferred. We illustrate the $(K, \varepsilon^H, \varepsilon^L)$-optimal set shown in Online Appendix D. The $(K, \varepsilon^H, \varepsilon^L)$-optimal set identification problem is a novel problem for the bandit literature, and it unifies multiple cases found in previous studies, including the following: (1) the best-arm identification problem (Soare et al. 2014, Tao et al. 2018, Fiez et al. 2019); (2) the top-K arm identification (Kalyanakrishnan and Stone 2010, Bubeck et al. 2013, Jiang et al. 2017, Chen et al. 2017b); (3) all ϵ-good arms, which aims to find one or more arms from all of the arms that are within ϵ-optimality (Mason et al. 2020); and (4) K of the top ρ fraction of arms, which aims to identify any K-distinct arms within this fraction (Ren et al. 2019). We show in Table D1 in Online Appendix D how the $(K, \varepsilon^H, \varepsilon^L)$-optimal set generalizes these problems when users are homogenous (i.e., using the same preference parameter for all users). Note that the parameters ϵ^H and ϵ^L in some entries in Table D1 in Online Appendix D are, in fact, latent. However, we can adapt our algorithm to overcome the latency of the parameters (see Online Appendix G for details).

Next, we explain our distribution and boundedness assumptions in Assumption 1.

Assumption 1 (Boundedness). *The following conditions hold.*

I. *For $t = 1, \ldots, T$, ϵ_t is 1-sub-Gaussian. Moreover, ϵ_t^2 is $\tilde{\sigma}_\epsilon$-sub-Gaussian with mean σ_ϵ^2.*

II. $\sup_{x \in \mathcal{X} \cup \mathcal{Z}} \|x\|_2 \le \beta_{\mathcal{X}}$; $\sup_{u, u' \in \mathcal{U}} \|u - u'\|_2 \le \beta_{\mathcal{U}}$; $\sup_{u \in \mathcal{U}} \|\theta(u)\|_2 \le \beta_\Theta$; $\sup_{x \in \mathcal{X} \cup \mathcal{Z}, u \in \mathcal{U}} |x^\top \theta(u)| \le 1$.

We introduce *preference divergence* to measure distributional user preference heterogeneity in Assumption 2. This concept is inspired by the notion of leaf node impurity in decision trees in supervised learning, and we use it to quantify the heterogeneity of instances within a leaf node. Let $\mathcal{D}(\mathcal{B})$ denote the conditional distribution of a user feature given that it is drawn from $\mathcal{B}$, where we call $\mathcal{B}$ a bin.

Assumption 2 (Preference Divergence). *Define $\overline{\theta}(\mathcal{B}) = \mathbb{E}_{U \sim \mathcal{D}(\mathcal{B})}[\theta(U)]$. Assume that for each user $u \in \mathcal{U}$, there exists a constant $div(u) \ge 0$, which we call the preference divergence, such that*

$$\|\theta(u) - \overline{\theta}(\mathcal{B})\|_2 \le div(u) \cdot \omega(\mathcal{B}), \quad \forall u \in \mathcal{B},$$

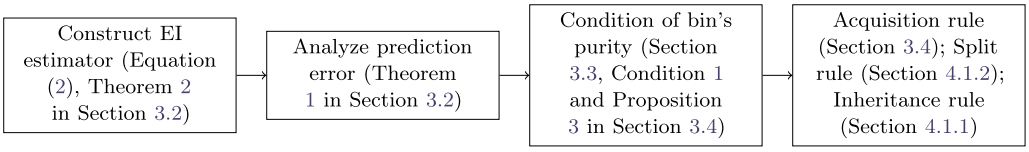
where $\omega(\mathcal{B})$ denotes the maximum two-norm distance of u in $\mathcal{B} \subseteq \mathcal{U}$. Moreover, the upper bound of $div(u)$ exists (i.e., $\ell := \sup_{u \in \mathcal{U}} div(u) > 0$ is a constant).

The metric of *preference divergence* measures the degree of variation, referred to as impurity, in user preferences from the average preference within a specified bin. This metric depends on two elements: (1) user preference, which refers to how the underlying parameter θ varies in bin $\mathcal{B}$, and (2) user type, which describes the distribution of a user feature in bin $\mathcal{B}$. For any given user u, the two-norm distance $\|\theta(u) - \overline{\theta}(\mathcal{B})\|_2$ is determined by the preference distribution of users in the same bin. If a certain user type is common (i.e., closer to the average preference), then the preference divergence $div(\cdot)$ is small. In contrast, if the user type is rare (i.e., far away from the average preference), then the preference divergence is large. Assuming that $div(\cdot)$ is bounded by a positive constant $\ell > 0$ for all users, we can obtain a naive bound $\sup_{u, u' \in \mathcal{B}} \|\theta(u) - \theta(u')\|_2 \le L_\Theta \omega(\mathcal{B})$, where $L_\Theta := \sqrt{2}\ell$. The metric also provides a trivial upper bound, $0 \le x^\top \mathbb{E}_{U \sim \mathcal{D}(\mathcal{B})}[(\theta(U) - \theta(u))(\theta(U) - \theta(u))^\top] x \le (L_\Theta \beta_{\mathcal{X}} \omega(\mathcal{B}))^2$, for all $x \in \mathcal{X} \cup \mathcal{Z}$, where the upper bound converges to zero as $\omega(\mathcal{B})$ shrinks toward zero.

We say a $(K, \varepsilon^H, \varepsilon^L)$-optimal set exists for user u if the cardinality is no larger than K; that is, $1 \le |\{z \in \mathcal{Z} : \max_{z' \in \mathcal{Z}} \theta(u)^\top z' - \theta(u)^\top z \le \varepsilon^L\}| \le K$. The $(K, \varepsilon^H, \varepsilon^L)$-optimal set may not exist if the total number of cards within the ε^L-optimality gap is larger than K. We specifically discuss how to adjust our algorithm to this setting in Section 4.2.

Before formally introducing the framework of one, we provide the road map of its design in Figure 3. We first construct the empirical impurity (EI) estimator, and then, we analyze the prediction error in Section 3.2. Establishing the relationship between prediction errors, sample size, and impurity, we delineate the conditions for determining bin purity in both Section 3.3 and Section 3.4. To elaborate further, we detail the acquisition rule (i.e., the methodology governing data acquisition) in Section 3.4, the inheritance rule (specifying what child bins inherit from parent bins) in Section 4.1.1, and the bin split rule in Section 4.1.2.

Figure 3. Road Map of the Design of Algorithm 1



Note. EI, empirical impurity.

3.2. Learning with User Pooling

In estimating the underlying parameter for a certain type of user, the challenge lies in selecting the data points to be pooled—an important step in tackling the data sparsity problem present in user interaction data (as shown in panels (a) and (b) of Figure 1). A trade-off exists between sample size and parameter impurity (i.e., user preference heterogeneity) within the pooling region. On the one hand, shrinking the pooling region may lead to scarce data, potentially resulting in high estimation variance. On the other hand, expanding the pooling region may subsume more data points, but the increased parameter impurity may also introduce a large estimation bias. This trade-off is demonstrated in Example 1. In this section, we first analyze the prediction error that occurs when users are pooled in bin $\mathcal{B} \subseteq \mathcal{U}$ to estimate $\theta(u)$, where $u \in \mathcal{B}$. We then discuss how to pool the data for estimation.

Example 1 (Data Pooling). Consider users with three binary features: gender (female or male), hometown (Beijing or not), and age (below or above 30 years old). A card is characterized by a three-dimensional feature (x_1, x_2, x_3): whether it is a lip-syncing video, whether the creator is a beauty influencer, and whether it is funny. Assume an even distribution across 2^3 types of users, with 10 data points collected for each. We also assume three user segments defined by their video preferences: (1) females under 30 years old who prefer lip-syncing videos from beauty influencers; (2) males under 30 years old who prefer funny lip-syncing videos; and (3) individuals over 30 years old who dislike lip-syncing videos. As such, hometowns do not influence user preference. Without data pooling, preference parameters for each user type are estimated using just 10 data points, which can lead to significant estimation variance resulting from potential observational errors. However, if we pool all Beijing users, the estimator error increases with the pooling region size (mirroring Example D1 in Online Appendix D).

To facilitate understanding, we first introduce our notations. Let $N_t(\mathcal{B})$ denote the total number of data points (i.e., user feature and reward pairs) in bin $\mathcal{B}$ at time t. Let $\mathcal{T}_t(\mathcal{B})$ represent the set of time steps when the user feature falls into bin $\mathcal{B}$; that is, $s \in \mathcal{T}_t(\mathcal{B})$ if $u_s \in \mathcal{B}$ for $s < t$, where u_s is the feature of user s. We apply ordinary least squares (OLS) regression to estimate parameters for user preferences corresponding to $\mathcal{B}$ by pooling all data points within $\mathcal{B}$. Hence, $\hat{\theta}_t(\mathcal{B})$ is the solution to the following equation:

$$\sum_{s \in \mathcal{T}_t(\mathcal{B})} (r_s - x_s^\top \hat{\theta}_t(\mathcal{B}))x_s = 0,$$

where r_s is the reward realized at time s. The squared-loss function is minimized by

$$\hat{\theta}_t(\mathcal{B}) = V_t(\mathcal{B})^{-1} \sum_{s \in \mathcal{T}_t(\mathcal{B})} r_s x_s \text{ with } V_t(\mathcal{B}) = \sum_{s \in \mathcal{T}_t(\mathcal{B})} x_s x_s^\top.$$

Let $\mathcal{F}_t(\mathcal{B})$ be σ-algebra summarizing the information available regarding the card feature and user feature corresponding to $\mathcal{B}$ up to time t; that is, $\mathcal{F}_t(\mathcal{B}) = \{(x_s, u_s) : s \in \mathcal{T}_t(\mathcal{B})\}$. We define the *empirical impurity* of user preference θ with respect to $\mathcal{F}_t(\mathcal{B})$ in Definition 2. Here, EI quantifies how user preferences vary in user bin $\mathcal{B}$ with respect to the historical card features. Conceptually, this notion measures the extent of user preference heterogeneity.

Definition 2 (Empirical Impurity). Given $\mathcal{F}_t(\mathcal{B})$, define the empirical impurity of user $u \in \mathcal{B}$ at time t with respect to bin $\mathcal{B}$ as

$$I_t(u, \mathcal{B}) := \sqrt{\frac{\sum_{s \in \mathcal{T}_t(\mathcal{B})} (x_s^\top (\theta(u_s) - \theta(u)))^2}{N_t(\mathcal{B})}}.$$

Note that the EI of user preference is a history-dependent measure with respect to both the card feature and the user feature. EI equals zero in any bins with homogenous users.

Our sampling process at particular time steps, which we call the end point of each epoch, is in a fixed design. That is, each time we estimate the underlying parameter, the sample sequence used to obtain the estimator is fixed. The criterion that we use to stop acquiring more samples for each bin and to split the parent bin determines the nature of the fixed design in our algorithm. We derive the prediction error bound for a fixed design in Theorem 1. For notation brevity, we define the matrix as

$$\overline{V}_t(\mathcal{B}) = \frac{1}{N_t(\mathcal{B})} \sum_{s \in \mathcal{T}_t(\mathcal{B})} x_s x_s^\top = \sum_{x \in \mathcal{X}} \lambda_x^t(\mathcal{B}) x x^\top,$$

where $\lambda_x^t(\mathcal{B}) = \sum_{s=1}^{t-1} \mathbf{1}(x_s = x)/N_t(\mathcal{B})$.

Theorem 1 (Prediction Error Bound). *Suppose Assumption 1 holds. For any $u \in \mathcal{B}$ and any $y \in \mathbb{R}^{d_p}$, it holds with probability at least $1-2\delta$ that the prediction error is bounded by*

$$|y^\top(\hat{\theta}_t(\mathcal{B}) - \theta(u))| \le \|y\|_{\overline{V}_t(\mathcal{B})^{-1}} \left(\sqrt{2\frac{\log(1/\delta)}{N_t(\mathcal{B})}} + I_t(u,\mathcal{B}) \right). \tag{1}$$

Complete proofs are included in Online Appendix E. This theorem corroborates the trade-off between the sample size and the level of user preference heterogeneity (impurity) within a user bin; hence, it provides a guideline on how to split the feature space and how to pool data to minimize the prediction error. In Theorem 1, the first term on the right-hand side (RHS) stems from the observational noise, which diminishes when more samples are acquired and vanishes when sample size goes to infinity. The second term corresponds to the EI of u in bin $\mathcal{B}$. In the special scenario where the user preference is homogenous (i.e., the parameter is pure) in $\mathcal{B}$, the second term disappears. Therefore, the first term dominates when $N_t(\mathcal{B})$ is small, which suggests the need to pool neighboring data with sparse samples. The second term dominates when $N_t(\mathcal{B})$ grows larger. That is, if the sample size is large enough and the impurity contributes more to the error, we may pool less data to increase user preference purity, thus reducing the second term in the error bound. This observation motivates us to design an algorithm that adaptively splits user bin $\mathcal{B}$ (to reduce the impurity of θ) and that simultaneously collects data points (to increase $N_t(\mathcal{B})$ and to reduce the noise).

This theorem echoes Example 1 more formally. If we pool users from multiple segments, the second term increases because users with heterogeneous preferences incur a large estimation error. In addition, we provide Example D2 in Online Appendix D to further illustrate the trade-off between bin size and impurity.

In applying the prediction error bound to quantify the confidence region according to Theorem 1, a major issue is that the EI, $I_t(u,\mathcal{B})$, is unknown. Although we can use $L_\Theta \beta_{\mathcal{X}} \omega(\mathcal{B})$ to bound the impurity according to the discussion following Assumption 2, this bound could be loose in some cases. Thus, in the following paragraphs, we propose a method for estimating EI. We first define the EI estimator as

$$\text{(EI estimator) } \hat{I}_t^2(\mathcal{B}) = \frac{1}{N_t(\mathcal{B})} \sum_{s \in \mathcal{T}_t(\mathcal{B})} (r_s - x_s^\top \hat{\theta}_t(\mathcal{B}))^2 - \sigma_\epsilon^2, \tag{2}$$

where σ_ϵ^2 is the variance of the noise. To describe how the EI estimator converges with sample size, we first introduce the following definition of the bound of EI.

Definition 3 (Bound of EI). Define the upper and lower bounds of empirical impurity for bin $\mathcal{B}$, denoted as $\overline{I}_t^2(\mathcal{B})$ and $\underline{I}_t^2(\mathcal{B})$, as follows:

$$\overline{I}_t^2(\mathcal{B}) = \max_{\theta \in \Theta(\mathcal{B})} \frac{1}{N_t(\mathcal{B})} \sum_{s \in \mathcal{T}_t(\mathcal{B})} (x_s^\top(\theta(u_s) - \theta)^2,$$

$$\underline{I}_t^2(\mathcal{B}) = \min_{\theta \in \Theta(\mathcal{B})} \frac{1}{N_t(\mathcal{B})} \sum_{s \in \mathcal{T}_t(\mathcal{B})} (x_s^\top(\theta(u_s) - \theta)^2,$$

where $\Theta(\mathcal{B})$ is the convex hull of $\{\theta(u) : u \in \mathcal{B}\}$.

Define $I_t^2(\overline{u}(\mathcal{B}),\mathcal{B}) = \frac{1}{N_t(\mathcal{B})} \sum_{s \in \mathcal{T}_t(\mathcal{B})} (x_s^\top(\theta(u_s) - \overline{\theta}(\mathcal{B})))^2$, where $\overline{\theta}(\mathcal{B}) = \mathbb{E}_{U \sim \mathcal{D}(\mathcal{B})}[\theta(U)]$. Theorem 2 provides high-probability bounds on the EI estimator.

Theorem 2. *Suppose Assumptions 1 and 2 hold. Fix any $\delta > 0$. Then,*

I. $\hat{I}_t^2(\mathcal{B}) \le \overline{I}_t^2(\mathcal{B}) + C_1\sqrt{\log(1/\delta)/N_t(\mathcal{B})}$ *holds with probability at least $1-2\delta$;*

II. $\hat{I}_t^2(\mathcal{B}) \ge \underline{I}_t^2(\mathcal{B}) - C_2\sqrt{\log(L\beta_\Theta d_p N_t(\mathcal{B})/\delta)/N_t(\mathcal{B})}$ *holds with probability at least $1-3\delta$, where $L = \beta_{\mathcal{X}}\sigma_\epsilon(2\sqrt{\log t} + \sqrt{2\log(2/\delta)})$; and*

III. $I_t^2(\overline{u}(\mathcal{B}),\mathcal{B}) \le \hat{I}_t^2(\mathcal{B}) + C_3\sqrt{\log(1/\delta)/N_t(\mathcal{B})}$ *holds with probability at least $1-6\delta$, where $C_1, C_2, C_3 > 0$.*

Theorem 2 shows that $\hat{I}_t^2(\mathcal{B})$ approximates $I_t^2(\overline{u}(\mathcal{B}),\mathcal{B})$ well and that the estimation error goes to zero in the order of $O(\sqrt{1/N_t(\mathcal{B})})$. When taking $N_t(\mathcal{B})$ to infinity, we can reach the conclusion that $\underline{I}_t^2(\mathcal{B}) \precsim \hat{I}_t^2(\mathcal{B}) \precsim \overline{I}_t^2(\mathcal{B})$ with high probability, where "$\precsim$" represents "asymptotically less than or equal to." Building on this result, we show an upper bound of $I_t^2(u,\mathcal{B})$ specifically for user u.

Proposition 1. *Suppose Assumptions 1 and 2 hold. For all $u \in \mathcal{B}$, it holds with probability at least $1-6\delta$ that*

$$I_t^2(u,\mathcal{B}) \le \hat{I}_t^2(\mathcal{B}) + \varrho_t(u,\mathcal{B}),$$

where $\varrho_t(u,\mathcal{B}) = C_3\sqrt{\log(1/\delta)/N_t(\mathcal{B})} + 4\beta_\Theta\beta_{\mathcal{X}}^2 div(u)\omega(\mathcal{B})$.

Proposition 1 indicates that when $\hat{I}_t^2(\mathcal{B}) + \varrho_t(u,\mathcal{B}) \le c$, it holds that $I_t^2(u,\mathcal{B}) \le c$ (with high probability). Thus, $\hat{I}_t^2(\mathcal{B}) + \varrho_t(u,\mathcal{B})$ provides a conservative estimate on the EI metric. The algorithm can utilize various estimators for the EI, employing optimistic, conservative, or mean estimators. In what follows, we derive the theoretical results based on the EI.

3.3. Optimal Set Identification

Next, we discuss the procedure to identify a $(K, \varepsilon^H, \varepsilon^L)$-optimal set for a targeted card set $\mathcal{Z}$. Let $z^*(u)$ denote the optimal card for user $u \in \mathcal{B}$. For set $Z \subseteq \mathcal{Z}$, define $\mathcal{Y}(Z) = \{z - z' : \forall z, z' \in Z, z \ne z'\}$ as the direction obtained from the differences between each pair of cards in set Z; also, define $\mathcal{Y}^*(Z;u) = \{z^*(u) - z : \forall z \in Z\ z^*(u)\}$ as the direction obtained from the differences between the optimal card and each suboptimal card. Based on Theorem 1 and letting $y = z^*(u) - z' \in \mathcal{Y}^*$

$(\mathcal{Z};u)$, we derive the bound for the prediction error $(z^*(u)-z')^\top\hat{\theta}_t$. If the RHS of Inequality (1) can be bounded by $(z^*(u)-z')^\top\theta(u)$, then it holds that $(z^*(u)-z')^\top\hat{\theta}_t(\mathcal{B}) \geq (z^*(u)-z')^\top\theta(u) - |(z^*(u)-z')^\top(\hat{\theta}_t(\mathcal{B}) - \theta(u))| \geq 0$. This result shows a sufficient condition for identifying the best card for user u. For the bound in Inequality (1) to hold for all $y \in \mathcal{Y}^*(\mathcal{Z};u)$, we need to uniformly bound over $\mathcal{Z}$ and thus, to replace δ with $\delta/|\mathcal{Z}|$. Similarly, if we want this bound to hold for $z-z'$ for all pairs $z,z' \in \mathcal{Z}$, we need to replace δ with $\delta/|\mathcal{Z}|^2$ to get the uniform bound. In doing so and based on the sufficient condition for differentiating the best card from the remaining cards, we want the RHS of Inequality (1) to satisfy that

$$\sqrt{2\frac{\log(|\mathcal{Z}|/\delta)}{N_t(\mathcal{B})}} + I_t(u,\mathcal{B}) \leq \frac{(z^*(u)-z')^\top\theta(u)}{\|z^*(u)-z'\|_{\overline{V}_t(\mathcal{B})^{-1}}}, \quad \forall u \in \mathcal{B},\ \forall z' \in \mathcal{Z}. \tag{3}$$

The objective is to determine an efficient allocation rule. We denote the allocation rule as $\boldsymbol{\lambda} \in \Lambda$, where $\Lambda = \{\boldsymbol{\lambda} \in \mathbb{R}^{|\mathcal{X}|} : \boldsymbol{\lambda} \geq 0, \sum_{x\in\mathcal{X}}\lambda_x = 1\}$ is the domain of probability distributions on $\mathcal{X}$ and where λ_x is the probability of sampling a card with feature x. For example, an allocation rule may suggest sampling so that 60% are lip-syncing cards from noninfluencers without funny content, 20% are cards created by influencers without lip-syncing or funny content, and 20% are funny cards without lip-syncing content from noninfluencers. Given the allocation rule $\boldsymbol{\lambda}$ and because the bound needs to hold for all $z \in \mathcal{Z}$, we define the uniform bound ψ for all $z \in \mathcal{Z}$ as follows:

$$\psi(\boldsymbol{\lambda};u) = \max_{z\in\mathcal{Z}\setminus\{z^*(u)\}} \frac{\|z^*(u)-z\|^2_{(\sum_{x\in\mathcal{X}}\lambda_x xx^\top)^{-1}}}{((z^*(u)-z)^\top\theta(u))^2}. \tag{4}$$

In this case, a sufficient condition is satisfied if $\sqrt{2\frac{\log(|\mathcal{Z}|/\delta)}{N_t(\mathcal{B})}} + I_t(u,\mathcal{B}) \leq 1/\sqrt{\psi(\boldsymbol{\lambda};u)}$ according to Equations (3) and (4). To summarize, Proposition 2 establishes a sufficient condition for identifying an $(M,\varepsilon^H,\varepsilon^L)$-optimal set, where M is the cardinality of set $\mathcal{Z}$.

Proposition 2 ($(M,\varepsilon^H,\varepsilon^L)$-Optimal Set Identification). *Fix $u \in \mathcal{B}$ and allocation rule $\boldsymbol{\lambda}$. Suppose $\mathcal{B}$ contains at least $\lfloor N_t(\mathcal{B})\lambda_x \rfloor$ cards with feature $x \in \mathcal{X}$. If $N_t(\mathcal{B})$ and $I_t(u,\mathcal{B})$ satisfy these two conditions*:

$$\begin{cases} N_t(\mathcal{B}) \geq 16\left(\frac{2\varepsilon^H}{\varepsilon^H-\varepsilon^L}\right)^2 \log(|\mathcal{Z}|^2/\delta)\psi(\boldsymbol{\lambda};u) & (5) \\ I_t(u,\mathcal{B}) \leq \frac{\varepsilon^H-\varepsilon^L}{2\varepsilon^H}\frac{1}{\sqrt{16\psi(\boldsymbol{\lambda};u)}}, & (6) \end{cases}$$

then

I. *it holds with probability at least $1-\delta$ that $(z^*(u)-z)^\top\hat{\theta}_t(\mathcal{B}) \geq 0$ for all $z \in \mathcal{Z}$, and*

II. *set $\mathcal{S}(\mathcal{B}) = \left\{z \in \mathcal{Z} : \max_{z'\in\mathcal{Z}} \hat{\theta}(\mathcal{B})^\top z' - \hat{\theta}(\mathcal{B})^\top z \leq \frac{\varepsilon^H+\varepsilon^L}{2}\right\}$.*

Then, with probability at least $1-\delta$, $\mathcal{S}(\mathcal{B})$ is an $(M,\varepsilon^H,\varepsilon^L)$-optimal set for user u.

According to the conditions in Proposition 2, we minimize $\psi(\boldsymbol{\lambda};u)$ not just to reduce the sample complexity but also, because a smaller value of $\psi(\boldsymbol{\lambda};u)$ would make Conditions (5) and (6) weaker. Therefore, we choose the optimal static allocation rule for u as

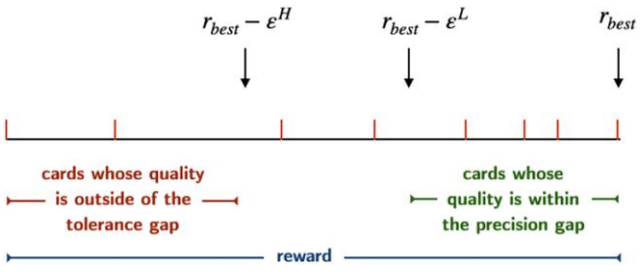
$$\boldsymbol{\lambda}^*(u) := \arg\min_{\boldsymbol{\lambda}\in\Lambda} \psi(\boldsymbol{\lambda};u). \tag{7}$$

However, solving for the optimal allocation rule $\boldsymbol{\lambda}^*(u)$ in Equation (7) requires that we know the true user preference $\theta(u)$, which needs to be estimated. Thus, in the next section, we propose an implementable algorithm that sequentially removes suboptimal arms until a $(K,\varepsilon^H,\varepsilon^L)$-optimal set is identified.

3.4. Sequential Removal and Identification Process

We use the notion of epochs to structure the sequential card removal and optimal card identification process. The criterion for eliminating cards becomes increasingly stringent as each epoch progresses. We discard suboptimal cards having a gap larger than $2^{-(l+1)}+\varepsilon^L$ when epoch l concludes. Any remaining video card with feature z satisfies $((z^*(u)-z)^\top\theta(u))^2 \leq (2^{-(l+1)}+\varepsilon^L)^2$. We label all cards as *active* at the beginning of the time horizon, and when the card is removed from $\mathcal{B}$, we label it as *inactive* for $\mathcal{B}$. Note that the label is bin specific. Conceptually, we start by discarding cards on the left end of the scale (as depicted in Figure 4). These cards are the ones that are relatively easier to distinguish from the best cards. Over time, we gradually shift our focus to the right, concentrating on the cards that are harder to differentiate. For example, consider a bin containing females younger than 30 (as discussed in Example 1). The best card may be the most recent lip-syncing video

Figure 4. (Color online) Illustration of the $(K,\varepsilon^H,\varepsilon^L)$-Optimal Set



Notes. The horizontal scale corresponds to the rewards achieved by cards in the exploration card set $\mathcal{Z}$. The red vertical lines correspond to cards, and the card on the far right is the best card (r_{best}). All cards in the green range are within the ε^L-optimality gap. All cards in the red range are within the ε^H-optimality gap.

on a Christmas song from an influencer who has 1 million followers. We first remove low-reward content for this user segment (e.g., product tutorials and frequently asked questions videos), which is easy to distinguish from the best card. We then progress to the next epoch, removing cards for which the gap between this card and the best card is smaller than that of the cards removed in the first epoch—for example, lip-syncing with funny content.

Define $\hat{\mathcal{Z}}_l(\mathcal{B})$ as the set of active cards at the beginning of epoch l in $\mathcal{B}$, and define the optimal allocation rule during epoch l and the corresponding optimal value as

$$\boldsymbol{\lambda}_l^*(\mathcal{B}) := \arg\min_{\boldsymbol{\lambda}\in\Lambda} \max_{y\in\mathcal{Y}(\hat{\mathcal{Z}}_l(\mathcal{B}))} \|y\|^2_{(\sum_{x\in\mathcal{X}}\lambda_x xx^\top)^{-1}} \text{ and}$$

$$\rho(\mathcal{Y}(\hat{\mathcal{Z}}_l(\mathcal{B}))) := \min_{\boldsymbol{\lambda}\in\Lambda} \max_{y\in\mathcal{Y}(\hat{\mathcal{Z}}_l(\mathcal{B}))} \|y\|^2_{(\sum_{x\in\mathcal{X}}\lambda_x xx^\top)^{-1}}.$$

To compute the optimal allocation rule, we follow an efficient rounding procedure as described in Soare et al. (2014, appendix C) and in Fiez et al. (2019, supplementary section B). This procedure produces $(1+\epsilon)$-approximations as long as N exceeds a required sample size $\nu(\epsilon) \le O(d_p/\epsilon^2)$. Essentially, given $\boldsymbol{\lambda}$ and N, the algorithm returns a sampling sequence $\boldsymbol{x}_N$, thus satisfying $\max_{y\in\mathcal{Y}} \|y\|^2_{(\sum_{i=1}^N x_i x_i^\top)^{-1}} \le (1+\epsilon)\max_{y\in\mathcal{Y}} \|y\|^2_{(\sum_{x\in\mathcal{X}}\lambda_x xx^\top)^{-1}}/N$. We include more details in Online Appendix F.

In the case where all active cards are within the optimality gap $2^{-(l+1)}+\varepsilon^L$, the quantity $(2^{-(l+1)}+\varepsilon^L)^2\rho(\mathcal{Y}(\hat{\mathcal{Z}}_l(\mathcal{B})))$ serves as an upper bound of $\psi(\boldsymbol{\lambda}^*(u);u)$ as defined in Equation (4). The reason is that $((z^*(u)-z)^\top\theta(u))^2$ is bounded by $(2^{-(l+1)}+\varepsilon^L)^2$ and that $\min_{\boldsymbol{\lambda}\in\Lambda}\|z^*(u)-z\|^2_{(\sum_{x\in\mathcal{X}}\lambda_x xx^\top)^{-1}}$ is bounded by $\rho(\mathcal{Y}(\hat{\mathcal{Z}}_l(\mathcal{B})))$. To remove suboptimal cards at the end of each epoch, we define the following condition for epoch completeness for a fixed confidence level $\delta>0$. Let $N_t^l(\mathcal{B})$ denote the number of samples collected for epoch l in $\mathcal{B}$ at time t.

Condition 1 (Epoch Completeness). *For users $u\in\mathcal{B}$ with candidate arm set $\hat{\mathcal{Z}}_l(\mathcal{B})$, epoch l is completed at time t for $\mathcal{B}$ if both of the following conditions are satisfied*:

I. *impurity*: $I_t(u,\mathcal{B}) \le \frac{2^{-(l+2)}}{\sqrt{16\rho(\mathcal{Y}(\hat{\mathcal{Z}}_l(\mathcal{B})))}}$ *or* $\omega(\mathcal{B}) \le \frac{2^{-(l+2)}}{L_\Theta\beta_{\mathcal{X}}\sqrt{16\rho(\mathcal{Y}(\hat{\mathcal{Z}}_l(\mathcal{B})))}}$ *and*

II. *sample size*: $N_t^l(\mathcal{B}) \ge \eta_l(\mathcal{B}) := 16\cdot 2^{2(l+2)}\log(|\hat{\mathcal{Z}}_l(\mathcal{B})|^2/\delta_l)\rho(\mathcal{Y}(\hat{\mathcal{Z}}_l(\mathcal{B})))$, *where* $\delta_l = \frac{\delta}{l^2}$.

Condition 1(I) is imposed on the EI of θ for user u in $\mathcal{B}$. We can use either I_t or ω in the algorithm design. The difference is that I_t is unknown and thus, needs to be estimated by $\hat{I}_t(\mathcal{B})$ according to Equation (2); meanwhile, using metric $\omega(\mathcal{B})$ satisfies a sufficient condition, and so, it is stronger. As we explain in Section 4.1.2, we split the bin by minimizing either the sum of $\hat{I}_t$ or ω in child bins.

Next, we introduce Proposition 3, which gives an implementable condition to identify the optimal set. Compared with Proposition 2, Proposition 3 does not require that we know $\theta(u)$ and thus, provides a practical condition to identify an $(M,\varepsilon^L+2^{-l},\varepsilon^L)$-optimal set.

Proposition 3 (Implementable Condition for $(M,\varepsilon^L+2^{-l},\varepsilon^L)$-Optimal Set). *Suppose at the beginning of epoch l, $\hat{\mathcal{Z}}_l(\mathcal{B})$ is an $(M,\varepsilon^L+2^{-l},\varepsilon^L)$-optimal set for all users $u\in\mathcal{B}$. Assume that at time t, epoch l is completed in $\mathcal{B}$, and let $\hat{\mathcal{Z}}_{l+1}(\mathcal{B}) = \hat{\mathcal{Z}}_l(\mathcal{B})\backslash\{z\in\hat{\mathcal{Z}}_l(\mathcal{B})|\max_{z'\in\hat{\mathcal{Z}}_l(\mathcal{B})}\hat{\theta}_t(\mathcal{B})^\top z' - \hat{\theta}_t(\mathcal{B})^\top z \ge 2^{-(l+2)}+\varepsilon^L\}$. Then, with probability at least $1-\delta_l$, $\hat{\mathcal{Z}}_{l+1}(\mathcal{B})$ is an $(M,\varepsilon^L+2^{-(l+1)},\varepsilon^L)$-optimal set for all users $u\in\mathcal{B}$.*

According to Proposition 3, at the end of epoch l, we remove z if there exists some $z'\in\hat{\mathcal{Z}}_l(\mathcal{B})$, such that $(z'-z)^\top\hat{\theta}_t(\mathcal{B}) \ge 2^{-(l+1)}+\varepsilon^L$. Because $\hat{\mathcal{Z}}_0(\mathcal{U}) = \mathcal{Z}$ is an $(M,\varepsilon^L+2^0,\varepsilon^L)$-optimal set based on Assumption 1 at the start of epoch $l=0$, when we take $\delta_0 = \frac{\delta}{|\mathcal{Z}|}$, Proposition 3 implies that $\hat{\mathcal{Z}}_l(\mathcal{B})$ is an $(M,\varepsilon^L+2^{-l},\varepsilon^L)$-optimal set for all $u\in\mathcal{B}$ with probability at least $1-\sum_{i=0}^l\delta_i$.

4. Algorithm: Adaptive Acquisition Tree

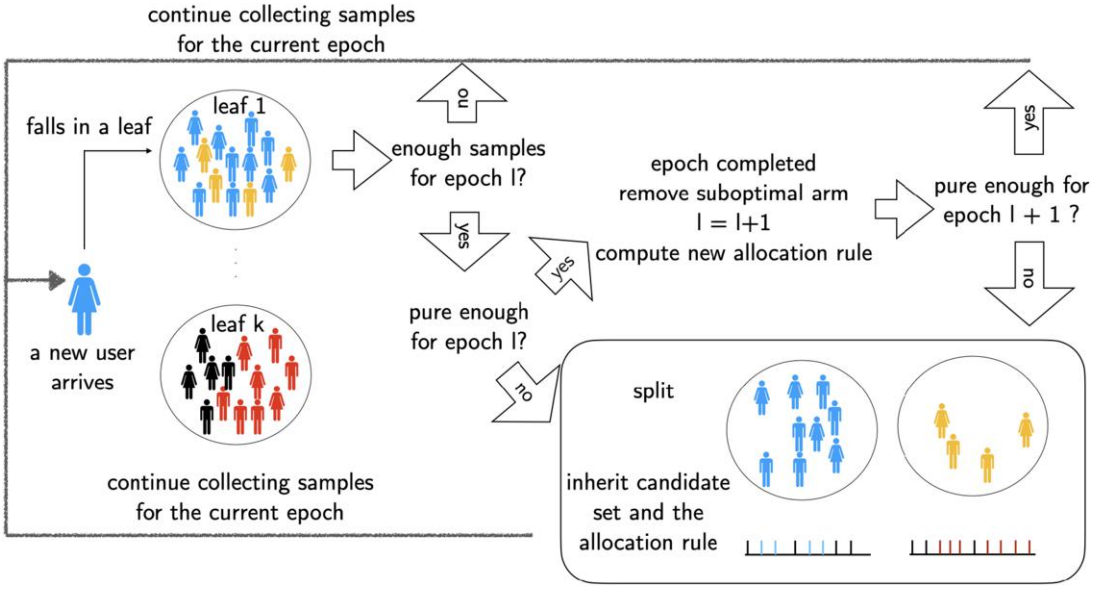
This section introduces Algorithm 1, which adaptively collects data points concurrently with splitting the user feature space. The workflow of this algorithm is presented in Figure 5. We keep track of the epoch label $l(\mathcal{B})$ and the number of samples collected for each bin $\mathcal{B}$. After gathering a sufficient quantity of samples in $\mathcal{B}$ for epoch l, we verify whether the impurity of θ falls below a given threshold (Condition 1(I)). If it does not, we split $\mathcal{B}$ into child bins (see Section 4.1.2); these child bins then take on the active arm set from their parent bin (explained in Section 4.1.1). If the impurity is low enough, epoch l is concluded. We remove suboptimal cards, update the allocation rule according to the process described in Section 3.4, and then, initiate epoch $l+1$. This process continues as we reassess whether impurity at the start of epoch $l+1$ is below the threshold for epoch $l+1$. If not, we partition the bin using the feature importance rule, which we discuss in Section 4.1.2.

4.1. Key Components of Algorithm 1

Algorithm 1 comprises three critical steps: (1) determining which components the child bins inherit from the parent bin, (2) establishing how to split the user bin, and (3) deciding when to halt the exploration process.

4.1.1. Child Bins' Inheritance. As shown in Figure 5, there are two possible scenarios in which a bin is split (labeled in red): (1) when epoch l ends and the impurity

Figure 5. (Color online) Flowchart of Algorithm 1



condition (Condition 1(I)) is not satisfied for epoch $l+1$ and (2) when the sample size condition (Condition 1(II)) is satisfied for epoch l but the impurity condition (Condition 1(I)) is not satisfied. Next, we discuss these two scenarios.

In the first scenario, when epoch l ends, we remove suboptimal cards from the bin and decide whether the bin is then pure enough (i.e., user preferences are similar enough) for epoch $l+1$. If Condition 1(I) is met for epoch $l+1$ (i.e., users' preferences are sufficiently similar), there is no need to split the bin, and epoch $l+1$ starts with newly computed stronger conditions. However, if the impurity condition is not satisfied, the user bin $\mathcal{B}$ is split into child bins $\mathcal{C} \subset \mathcal{B}$ to achieve purer user bins. Suboptimal cards, which have an optimality gap exceeding $2^{-(l+1)} + \varepsilon^L$, have been removed for users in $\mathcal{C}$, implying that the child bins inherit only the active card set from the parent bin. The next stage involves acquiring samples to eliminate cards outside of the $2^{-(l+2)} + \varepsilon^L$ optimality gap for the child bins. Therefore, the allocation rule is updated to obtain cards with smaller gaps, relative to the best card, and child bins collect data until the sample size in Condition 1(II) is satisfied for epoch $l+1$.

Under the second condition, we skip the card removal process because epoch l has not ended, and the child bins continue collecting samples using the same allocation rule. When bin $\mathcal{B}$ is divided, data points within $\mathcal{B}$ fall into different child bins. However, because we remain in the same epoch, the child bin inherits the allocation rule of the parent bin. As the active card set is inherited from a parent bin, the child bins $\mathcal{C}$ merely need to collect the missing samples following the optimal allocation rule. Consider Example 1 for illustration. Assume that the first split after implementing the algorithm is on age, and the next split for the parent node (younger than 30) is gender. We further assume that the cards with product tutorials and with questions and answers are removed and that all other cards remain for the parent node. At this point, the two child nodes for both genders inherit the same active card sets. Meanwhile, assuming that the parent node has already sampled 60% of the lip-syncing videos, the two child nodes continue to collect the missing samples for the remaining 20% influencer-created cards and 20% funny cards.

4.1.2. Splitting Process and Measures of Impurity.

The key question in the splitting process is the division of the parent bin into child bins. Based on Condition 1(I), we use either EI estimator $\hat{I}_t$ (defined in Equation (2)) or ω (maximum distance of the user feature) as the splitting criterion. However, a split based on ω is not guided by the reward information. Using the metric of ω provides an upper bound on the impurity, which may result in an inefficient data acquisition process and thus, necessitate more splits and a larger sample size than would otherwise be needed. Again, we use Example 1 as an illustration. Even though hometown is one of the features that involves the maximum distance, it is not predictive toward the reward. As a result, splitting according to this feature reduces the sample size without decreasing impurity. Alternatively, we can use $\hat{I}_t$ for the measure of impurity. Let $\mathcal{B}^L(i)$ and $\mathcal{B}^R(i)$ be the left boundary and the right boundary, respectively, of the ith dimension of $\mathcal{B}$. We enumerate feature i from one to d_u. When dimension i is selected, bin $\mathcal{B}$ is divided into a left child bin $\mathcal{C}_i^L(\mathcal{B}) =$

$\mathcal{B}(1)\times\cdots\times[\mathcal{B}^L(i),\mathcal{B}^M(i)]\times\cdots\times\mathcal{B}(d_u)$ and a right child bin $\mathcal{C}_i^R(\mathcal{B})=\mathcal{B}(1)\times\cdots\times[\mathcal{B}^M(i),\mathcal{B}^R(i)]\times\cdots\times\mathcal{B}(d_u)$, where $\mathcal{B}(k)=[\mathcal{B}^L(k),\mathcal{B}^R(k)]$ and $\mathcal{B}^M(i)$ represents the split point. Feature space can be either discrete or continuous. According to Theorem 1, the prediction error bound related to the impurity measure is $\|y\|^2_{\bar{V}_t(\mathcal{B})^{-1}} I_t^2(u,\mathcal{B})$. Because the allocation rule does not change before the epoch completes, we minimize the expectation of impurity in child bins to obtain a tight bound: that is,

$$\min_{\mathcal{C}^L(\mathcal{B}),\mathcal{C}^R(\mathcal{B})} \mathbb{E}_{U\sim\mathcal{D}(\mathcal{B})}[I_t^2(U,\mathcal{C}_i^L(\mathcal{B}))\mathbf{1}(U\in\mathcal{C}_i^L(\mathcal{B})) + I_t^2(U,\mathcal{C}_i^R(\mathcal{B}))\mathbf{1}(U\in\mathcal{C}_i^R(\mathcal{B}))]. \quad (8)$$

Let γ_L and γ_R represent the proportions of data points that fall into the left bin and the right bin, respectively; that is, $\gamma_L=\frac{N_t(\mathcal{C}_i^L(\mathcal{B}))}{N_t(\mathcal{C}_i^L(\mathcal{B}))+N_t(\mathcal{C}_i^L(\mathcal{B}))}$, and $\gamma_R=\frac{N_t(\mathcal{C}_i^R(\mathcal{B}))}{N_t(\mathcal{C}_i^L(\mathcal{B}))+N_t(\mathcal{C}_i^L(\mathcal{B}))}$. When using the EI estimator $\hat{I}_t$, the objective in Equation (8) can be approximated by $\gamma_L\hat{I}_t^2(\mathcal{C}_i^L(\mathcal{B}))+\gamma_R\hat{I}_t^2(\mathcal{C}_i^R(\mathcal{B}))$.

We introduce a new notion, *feature importance*, to measure the importance of the feature in the split. Conducting the regression on $\mathcal{C}_i^L(\mathcal{B})$ and $\mathcal{C}_i^R(\mathcal{B})$, we define the feature importance of i in bin $\mathcal{B}$, denoted as $\phi_i(\mathcal{B})$, as follows:

$$\phi_i(\mathcal{B}):=\hat{I}_t^2(\mathcal{B})-(\gamma_L\hat{I}_t^2(\mathcal{C}_i^L(\mathcal{B}))+\gamma_R\hat{I}_t^2(\mathcal{C}_i^R(\mathcal{B}))). \quad (9)$$

Feature importance quantifies the decrease in impurity in θ upon splitting feature i at split point $\mathcal{B}^M(i)$. For each dimension i, we select the split point that maximizes $\phi_i(\mathcal{B})$ and then, select the feature with the largest feature importance in $\mathcal{B}$. In Example 1, the feature importance of the three features, in descending order, is age, gender, and hometown. We would expect $\phi_i(\mathcal{B})$ for hometown for all bins to be zero, which means that splitting this feature results in no improvement in the empirical impurity.

4.1.3. Stopping Criterion.

Proposition 3 indicates that when $l\geq\log(1/(\varepsilon^H-\varepsilon^L))$, arms with a gap larger than ε^H are removed. Therefore, to guarantee that all recommended cards are within the quality gap ε^H, we set the stopping criterion at $l\geq\log(1/(\varepsilon^H-\varepsilon^L))$.

4.2. Algorithm

At this point, we can formally introduce Algorithm 1 to summarize the key procedures described in Section 4.1. At time t, when a user with feature u_t arrives, we first find the bin to which she belongs ($\mathcal{B}(u_t)$). If not enough data points are collected in $\mathcal{B}(u_t)$, we select the card according to the allocation rule in that bin (Algorithm 1, line 8). Otherwise, we determine whether the user preference θ is pure enough for epoch l in $\mathcal{B}$ (Algorithm 1, line 11). Once the impurity is sufficiently small (i.e., users in $\mathcal{B}$ have sufficiently similar preferences), we remove suboptimal cards with an optimality gap larger than $2^{-(l(\mathcal{B})+2)}+\varepsilon^L$ (Algorithm 1, line 12) compared with the best card, and we update both the allocation rule and the sample threshold for the next epoch $l(\mathcal{B})+1$ according to Algorithm Subroutine: Allocation Rule in Online Appendix F. If the impurity is not low enough, we split users in $\mathcal{B}$ into child bins by maximizing the feature importance in Algorithm Subroutine: Adaptive Split in Online Appendix F (Algorithm 1, line 16). The child bins inherit the active card set and the allocation rule from the parent bin (Algorithm 1, line 18). Furthermore, we explain how our algorithm can be adapted to other identification problems (summarized in Table D1 in Online Appendix D) in Online Appendix G.

In Algorithm 1 (AAT), we can use different impurity estimates for determining whether to split the bin. For example, we can either use the EI estimator or the conservative estimator (specified in Theorem 2). To guarantee that the bin is pure enough for all users, we use the conservative estimator to characterize the theoretical performance.

Algorithm 1 (AAT)

1: **input**: arm set $\mathcal{X}$, target set $\mathcal{Z}$, confidence level $\delta_l\in(0,1)$;
2: $\hat{\mathcal{Z}}_1=\mathcal{Z}$, $t=1$; $\mathcal{B}_1(u)=\mathcal{U}$ for all $u\in\mathcal{U}$; $l(\mathcal{U})=0$; $\eta(\mathcal{U})=\eta_0$; ▷ initialization for partition, epoch label, and initial sample threshold
3: **Initial exploration phase**: randomly explore;
4: **while** $l(\mathcal{B}_t(u))\leq\lfloor\log(1/(\varepsilon^H-\varepsilon^L)\rfloor$ or $|\hat{\mathcal{Z}}_{l(\mathcal{B}_t(u))}(\mathcal{B}_t(u))|>K$ for some $\mathcal{B}_t(u)$ **do**
5: **input** u_t; $\mathcal{B}=\mathcal{B}_t(u_t)$; $l=l(\mathcal{B})$; ▷ input the feature of user t
6: $N^l(\mathcal{B})+=1$; ▷ find the bin that u_t belongs to
7: **if** $N^l(\mathcal{B})<\eta_l(\mathcal{B})$ **then** ▷ if not enough samples are collected
8: Select card according to the allocation rule in bin $\mathcal{B}$;
9: **else** ▷ if enough samples are collected for the bin
10: Compute $\hat{\theta}_t(\mathcal{B})$ by OLS;
11: **if** $\hat{I}_t(\mathcal{B})\leq\frac{2^{-(l+2)}}{\sqrt{16\rho(\mathcal{Y}(\hat{\mathcal{Z}}_l))}}$ **then** ▷ if impurity of θ in bin $\mathcal{B}$ is sufficiently small
12: $\hat{\mathcal{Z}}_{l+1}(\mathcal{B})=\hat{\mathcal{Z}}_l(\mathcal{B})\backslash\{z\in\mathcal{Z}|\exists z'\in\hat{\mathcal{Z}}_l(\mathcal{B}):(z'-z)^\top\hat{\theta}_t\geq 2^{-(l+2)}+\varepsilon^L\}$; ▷ remove suboptimal arms, epoch l ends
13: $l(\mathcal{B})+=1$; update allocation rule by Algorithm Subroutine: Allocation Rule in Online Appendix F;
14: **end if**
15: **if** $(\hat{I}_t(\mathcal{B})\geq\frac{2^{-(l+2)}}{\sqrt{16\rho(\mathcal{Y}(\hat{\mathcal{Z}}_l))}})$ and $(N^l(\mathcal{B})\geq\eta_l(\mathcal{B}))$ **then**
16: Apply Algorithm Subroutine: Adaptive Split in Online Appendix F for the feature space split; ▷ split the user feature space
17: **for** each child bin $\mathcal{C}\subseteq\mathcal{B}$ **do** ▷ update the allocation rule

18: Inherit the allocation rule and active arm set from the parent bin;
19: **end for**
20: **end if**
21: **end if**
22: $t+=1$;
23: **end while**

Remark 1. If the total number of cards within the ε^L-optimality gap is larger than K, we can randomly select K cards, similar to Ren et al. (2019). Alternatively, we can gradually shrink the precision gap until there are K cards remaining (see Online Appendix G).

4.3. Sample Complexity

In this section, we establish a theoretical bound on the sample complexity of Algorithm 1. The sample complexity differs from that in traditional best-arm identification problems because the underlying parameters are heterogeneous among users. We show the sample complexity for identifying a $(K, \varepsilon^H, \varepsilon^L)$-optimal set both for a fixed user $u \in \mathcal{U}$ and for all users under certain conditions. Based on the adaptive splitting process, we define the bin containing user u that completes epoch l as $\mathcal{P}_l(u)$; then, $\mathcal{P}_1(u), \mathcal{P}_2(u), \ldots, \mathcal{P}_{e(u)}(u)$ constitutes a splitting path for user u, where $e(u)$ is the final epoch label of user u as shown in Figure H1 in Online Appendix H. Based on our sampling policy, $\mathcal{P}_1(u) \supseteq \mathcal{P}_2(u) \supseteq \cdots \supseteq \mathcal{P}_{e(u)}(u)$.

To describe the sample complexity, let $\Delta^*(z;u) = r(u,z^*(u)) - r(u,z)$ be the reward gap between card z and the optimal card for user u. Define $\Delta^i(u)$ as the ith minimum gap from the optimal card for user u. We further define $\mathcal{S}_l^*(u) = \{z \in \mathcal{Z} : \Delta^*(z;u) \le 2^{-l} + \varepsilon^L\}$ as the set of arms within the $(2^{-l} + \varepsilon^L)$-optimality gap. In epoch l, we expect $\hat{\mathcal{Z}}_l(u) \subseteq \mathcal{S}_l^*(u)$ according to Proposition 3. Let $N^l(\mathcal{B})$ denote the number of samples collected for bin $\mathcal{B}$ during epoch l.

Theorem 3 (Sample Complexity). *Suppose Assumptions 1 and 2 hold. Using an ϵ-efficient rounding procedure, with probability at least $1-\delta$, the following conclusions hold.*

I. *Single user. Define* $l_{\max}(u) = \lfloor \log(1/\min\{\varepsilon^H - \varepsilon^L, \Delta^K(u) - \varepsilon^L\}) \rfloor$. *Suppose $\mathcal{P}_l(u)$ is the bin that contains user u at the completion of epoch l. When using $\hat{I}_t(\mathcal{B}) + \varrho_t(u,\mathcal{B})$ as the impurity estimate (in lines 11 and 15 in Algorithm 1), Algorithm 1 correctly identifies a $(K, \varepsilon^H, \varepsilon^L)$-optimal set for user u and requires a worst-case sample complexity along bin path $\mathcal{P}_1(u), \ldots, \mathcal{P}_{e(u)}(u)$, in which*

$$\sum_{l=1}^{e(u)} N^l(\mathcal{P}_l(u)) \le \sum_{l=1}^{l_{\max}(u)} \max\{h_l(u), \lceil \overline{n}_l(u) \rceil, \nu(\epsilon)\},$$

where $h_l(u) := \lceil 16\rho(\mathcal{Y}(\mathcal{S}_l^*(u)))(1+\epsilon)\log(|\mathcal{Z}|^2/\delta_l)2^{2(l+2)} \rceil$ *with* $\delta_l = \frac{9\delta}{l^2}$, $\overline{n}_l(u)$ *is the solution to* (15), *and* $\nu(\epsilon)$ *is the minimum number of samples that satisfies* $\nu(\epsilon) = O(d_p/\epsilon^2)$.

II. *All users. Define* $\overline{h}_l = \sup_{u\in\mathcal{U}} h_l(u)$ *with* $\delta_l = \frac{\delta}{l^2 m_l}$, *where* $m_l = (L_\Theta \beta_\mathcal{X} \beta_\mathcal{U} 2^{l+5} \sqrt{\sup_{u\in\mathcal{U}} \rho(\mathcal{Y}(S_l^*(u)))})^{d_u}$, $\overline{l} := \sup_{u\in\mathcal{U}} l_{\max}(u)$, *and* $\overline{n}_l = \sup_{u\in\mathcal{U}} \lceil \overline{n}_l(u) \rceil$. *Assume* $\overline{h}_l < \infty$ *and* $\overline{l} < \infty$. *When using metric ω for impurity criteria in Condition 1*(I), *Algorithm 1 (AAT) correctly identifies a $(K, \varepsilon^H, \varepsilon^L)$-optimal set for all users and requires a worst-case sample complexity for which the following holds*:

$$N \le \sum_{l=1}^{\overline{l}} m_l \max\{\overline{h}_l, \overline{n}_l, \nu(\epsilon)\}.$$

Theorem 3(I) characterizes the sample complexity for differentiating an optimal set for a single user, whereas Theorem 3(II) characterizes it for all users. The sample complexity is affected by both the card features and the user characteristics (heterogeneity).

We first discuss how the determinants of the card features affect the sample complexity.

1. Epoch number. $l_{\max}(u)$ depends on the quantity $\Delta^K(u) - \varepsilon^L$, where $\Delta^K(u)$ is customized for user u. Recall that $\Delta^K(u)$ is the optimality gap between the Kth-best arm and the best arm for user u. The depth of the tree expands as the optimality gap ($\Delta^K(u)$) gets closer to the precision gap (ε^L) because the identification problem becomes more difficult. This dependence aligns with the best K-arm identification problem (Kaufmann et al. 2016, Chen et al. 2017b) and the problem of finding all ε-good arms (Mason et al. 2020).

2. The quantity $h_l(u)$ is determined by three terms: $\rho(\mathcal{Y}(\mathcal{S}_l^*(u)))$, $|\mathcal{Z}|$, and δ. First, a larger number of cards in the target set $|\mathcal{Z}|$ makes the identification problem harder. Second, requiring a higher confidence interval (δ) leads to increased sample complexity.

3. The complexity of $\rho(\mathcal{Y}(\mathcal{S}_l^*(u)))$ is affected by the geometry of $\mathcal{X}$ and $\mathcal{Z}$. From Fiez et al. (2019, lemma 1), we can show that

$$\max_{y\in\mathcal{Y}} \|y\|_2^2/\beta_\mathcal{X} \le \rho(\mathcal{Y}) \le d_p/\gamma_\mathcal{Y}^2,$$

where we define the gauge of $\mathcal{Y}$ as $\gamma_\mathcal{Y} = \max\{c > 0 : c\mathcal{Y} \subset \mathrm{conv}(\mathcal{X} \cup -\mathcal{X})\}$ and $\mathrm{conv}(\cdot)$ denotes the convex hull. Consider a special scenario where only two active arms remain in $\mathcal{Z}$ (i.e., when $\mathcal{Y}$ is a singleton that is $y = z_1 - z_2$); $\gamma_\mathcal{Y}$ becomes the gauge norm of y with respect to $\mathrm{conv}(\mathcal{X} \cup -\mathcal{X})$. When $\mathcal{X} = \{y\}$, we observe that $\max_{y\in\mathcal{Y}} \|y\|_2^2/\beta_\mathcal{X} = 1$ and $\gamma_\mathcal{Y} = 1$. In this case, the exploitation card set $\mathcal{X}$ contains the card that can help to identify the top card in the exploration card set $\mathcal{Z}$. However, if y lies in the orthogonal space of $\mathcal{X}$, then differentiating between z_1 and z_2 through cards in $\mathcal{X}$ becomes impossible, and $\gamma_\mathcal{Y} = 0$; as a result, $d_p/\gamma_\mathcal{Y}^2$ is infinite. Please see Example D4 in Online Appendix D for an illustration.

We next discuss how the sample complexity is affected by user types. Intuitively, when users' preferences are more homogenous (i.e., the bin is purer), the

sample complexity is lower for identifying an optimal set for all users because more user data can be pooled. One special case of our problem is to identify the best card for homogenous users in the transductive setting, which is the problem studied in Fiez et al. (2019). In this case, the impurity is zero, thereby automatically satisfying the impurity condition. When the goal is to identify the best card, we have $l_{\max} = \lfloor \log(1/\Delta_{\min}) \rfloor$. Then, according to Theorem 3, the sample complexity is $\sum_{l=1}^{\lfloor \log(1/\Delta_{\min}) \rfloor} \max\{\lceil 16\rho(\mathcal{Y}(\mathcal{S}_l^*(u)))\log(l^2|\mathcal{Z}|^2/\delta)2^{2(l+2)} \rceil, \nu(\epsilon)\}$. The sample complexity coincides with that of Fiez et al. (2019) (up to a constant), matching the lower bound derived in their study up to logarithmic factors. As illustrated in Example 1, when all users share the same preferences, the tree does not need to split, and the minimum sample complexity is achieved. In contrast, when preferences differ significantly among user types, the tree deepens, leading to a higher sample complexity.

5. Empirical Evaluation

We demonstrate the practical applicability and effectiveness of our proposed method through numerical experiments. We use real-world data from the NetEase Cloud Music platform, one of the leading music-streaming services in China. The data set includes music video cards, which are either short videos with background music or sets of pictures and texts accompanied by music.

We conducted our evaluation based on the data generated on NCM on November 2, 2019. A prominent feature of the NCM platform is "cloud village," where users receive card recommendations. Each card contains a frame of the video, a brief video description, creator logo and name, and the number of likes. The characteristics of the card and its creator influence user interactions. The aim of our experiment is to carry out online learning, collecting information during an exploration period and identifying the optimal set during an exploitation period in four transductive scenarios. The ultimate objective is to improve metrics that SFV platforms value, such as future card view time, click-through rate, and number of likes.

5.1. Experimental Procedure

5.1.1. Data and Experiment Setup. Interpretation of user responses on the NCM platform is confounded with the recommendation system (Chan and Loh 2004); users interact only with cards ranked high in the cloud village. Consequently, their behaviors concerning cards not recommended or ranked low in the cloud village are unobservable. To learn user preferences for counterfactuals, we use all data at hand to estimate θ, which we consider the ground-truth parameter in our experiments.[10] We measure user preferences based on the time that users spend viewing cards in their feed. We exclude cards and users with total view time of less than 60 seconds on November 2, 2019 to avoid bias. After this exclusion, our analysis includes 7,511 users during the exploration period and 50,809 users during the exploitation period. We assume that users having similar characteristics have similar tastes for cards.[11] To model user preference heterogeneity, we use K-means clustering to split users into five bins based on the characteristics of age, gender, registered month, follower count, and activity level. Each bin consists of 7.9%, 7.3%, 25.4%, 28.4%, and 31.0% of users in the exploration set, respectively. We perform one-hot encoding on gender, leading to a total of six features for each user. User reward (view time) for a card is computed as $R(u,x) = x^\top \theta(u) + \epsilon$, where the variance of noise ϵ follows the normal distribution. In this paper, we normalize the view time relative to the best card within each cluster so that the view time is in the range of zero to one.

We design our experiment around a hypothetical manager at NCM who aims to make two types of recommendations: (1) a spotlight recommendation, where a prominent position in the interface is assigned to one card with the longest normalized view time (relative to the view time of the best card), and (2) a $(K, \varepsilon^H, \varepsilon^L)$-optimal set recommendation, where no more than K cards are selected such that all cards have a normalized view time within 95% of the view time of the best card and where no cards with 75% of the view time of the best card are included. Therefore, the objective is to find the $(K = 10, \varepsilon^H = 0.25, \varepsilon^L = 0.05)$-optimal set for all users. There are one, four, two, one, and three cards in the precision set of each of the five clusters. In addition, there are 14, 18, 41, 16, and 99 cards in the tolerance set.

We randomly selected 1,200 cards with unique features to evaluate the performance of our method. We study the four transductive scenarios detailed in Section 3.1 with the same exploration set $\mathcal{Z}$ with 200 card types and with different exploration sets $\mathcal{X}$. During the exploration period, users in the exploration set arrive sequentially and receive one recommended card from the measurement set ($\mathcal{X}$) according to the allocation strategy. The manager can observe the reward from this card. In the exploitation period, users in the exploitation set also arrive sequentially and receive the optimal recommended card types based on different sampling strategies. Each card type corresponds to a different feature combination. We evaluate the metrics on all users in the exploitation set. We designed the four transductive settings as follows, and each card is characterized by a 73-dimensional feature:

- $\mathcal{Z} \subset \mathcal{X}$: 1,000 card types in $\mathcal{X}$ such that $\mathcal{Z}$ is a subset of $\mathcal{X}$;
- $\mathcal{X} \subset \mathcal{Z}$: 150 card types in $\mathcal{X}$ such that $\mathcal{X}$ is a subset of $\mathcal{Z}$;

- $\mathcal{X} \cap \mathcal{Z} = \emptyset$: 1,000 card types in $\mathcal{X}$ with no overlap between the two sets; and
- $\mathcal{X} \cap \mathcal{Z} \neq \emptyset$ but $\mathcal{Z} \nsubseteq \mathcal{X}$ and $\mathcal{X} \nsubseteq \mathcal{Z}$: 1,000 card types in $\mathcal{X}$ such that 100 card types of the two sets overlap.

5.1.2. Performance Metric. We evaluate the performance of our method using various metrics. For both the spotlight recommendation and the $(10, \varepsilon^H, \varepsilon^L)$-optimal set recommendations, we evaluate the average reward on the recommended cards using the normalized view time (relative to the best cards). We measure the expected number (i.e., percentage) of times that the best card is identified for the spotlight recommendation. For the $(10, \varepsilon^H, \varepsilon^L)$-optimal set recommendation, we compute the expected number of cards found in the ε^L set. A higher value implies a higher quality of recommendation; in other words, a higher value indicates that a larger number of cards within the ε^L set is found. For instance, in this particular setting, an oracle method can identify an average of 2.09 cards within the ε^L set computed from a weighted average of cards in the precision sets across all bins. We also evaluate the expected number of selected cards that are not in the ε^H set. A *smaller* value indicates better performance; in this case, fewer cards outside of the ε^H set are included. An oracle model obtains zero cards outside of the ε^H set. Furthermore, we evaluate the area under the curve (AUC) for identifying both the best card (in the spotlight recommendation) and the precision set (in the top card recommendation). AUC is a common metric used in binary classification when the positive and negative cases are not balanced. For the $(10, \varepsilon^H, \varepsilon^L)$-optimal set recommendation, we also use an F_1 score specifically for the precision set. The F_1 score is the harmonic mean of precision and recall, two key metrics that help evaluate the accuracy of a system.[12]

5.1.3. Benchmarks. We use two types of benchmarks in this study. First, we compare our method with state-of-the-art methods in the contextual bandit literature to identify the best card. Because we introduce the concept of an optimal set and we account for user heterogeneity, none of the bandit benchmarks can directly optimize our objective. Therefore, we select a set of methods in the vein of pure exploration, which aims for best-arm identification while learning coefficients for card features. This approach naturally enables the top-K recommendation. We use both adaptive and nonadaptive exploration strategies as well as transductive and nontransductive strategies. Next, we discuss each of the benchmark strategies.

- $\mathcal{XY}$ static (Soare et al. 2014) is a static allocation design where the goal is to reduce the uncertainty of the gaps for all arms.
- LinGapE (Xu et al. 2018) adopts a gap-based sampling rule.
- ALBA (Tao et al. 2018) is an elimination-based algorithm that offers improvement over $\mathcal{XY}$ adaptive's performance by using a tighter elimination criterion.
- RAGE (Fiez et al. 2019) extends the $\mathcal{XY}$-adaptive strategy to a more general transductive setting. The method is also based on elimination.

To demonstrate the effectiveness of data acquisition alone, we compare our method with a straightforward prediction approach that integrates both user heterogeneity and card features within the decision tree structure. This approach uses a decision tree-based model with uniform sampling (named a uniform sampling segmentation tree [USST]), where we adapt the tree to perform regression in its leaf nodes. Users are segmented based on their responses to card view time, with each split incorporating both user features and card characteristics to refine the segmentation and response predictions within each node.[13]

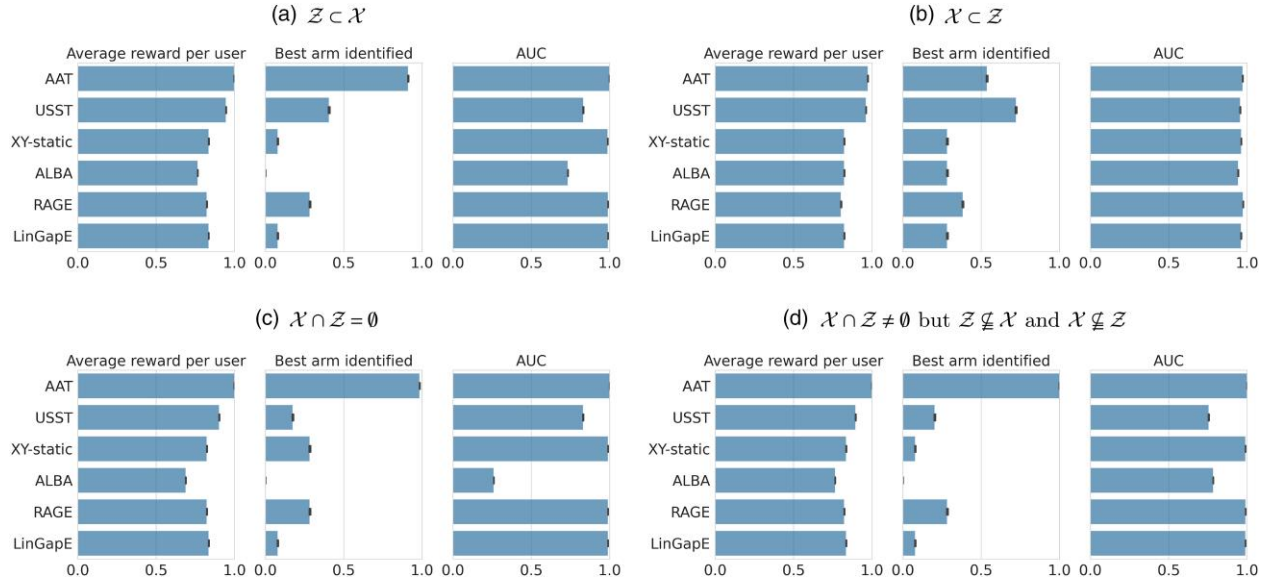
5.2. Results

We present the experimental results for spotlight recommendation (i.e., best-arm identification) in Figure 6 and the $(10, \varepsilon^H, \varepsilon^L)$-optimal set identification in Figure 7 for the four transductive scenarios. For implementing AAT, we directly use the EI estimator (Equation (2)) for measuring the impurity in Algorithm 1 (refer to lines 11 and 15).

As shown in Figure 6, our method achieves spotlight recommendation performance that is approximately 85%–100% higher than the best acquisition benchmark, and 30%–100% higher in accuracy compared to USST.[14] Even though RAGE is designed for transductive settings, it assumes homogenous user segments. It fails to provide any performance guarantees when user heterogeneity is present, and it underperforms in our setting (as demonstrated in Example D1 in Online Appendix D). In terms of the best card for the spotlight recommendation, we see that our method can increase the average reward by about 25%~100% compared with the top data acquisition benchmark and by about 85% ~100% compared with USST for all scenarios except for $\mathcal{X} \cap \mathcal{Z} \neq \emptyset$ but $\mathcal{Z} \nsubseteq \mathcal{X}$ and $\mathcal{X} \nsubseteq \mathcal{Z}$, which is reduced by about 26%.[15] Our improved performance, compared with a USST, suggests the effectiveness of implementing data acquisition on SFV platforms.

Our method shows an increase in average reward by approximately 45%–73% compared with the best data acquisition benchmark and by 45%–61% compared with the USST benchmark. When compared with the best benchmark across both groups, our method demonstrates an improvement of approximately 43%–56%. For the ε^H-gap set, AAT retains between 0 and 0.47 cards—substantially outperforming the top acquisition benchmark, which acquires around 3.8–4.2 cards in intolerable sets. The best-performing benchmark overall is the RAGE (3.66–4.16 cards) except in

Figure 6. (Color online) Performance on the Spotlight Recommendation (Best-Card Identification)



Notes. The y axes correspond to the methods, and the x axes correspond to the performance metrics. (a) $\mathcal{Z} \subset \mathcal{X}$. (b) $\mathcal{X} \subset \mathcal{Z}$. (c) $\mathcal{X} \cap \mathcal{Z} = \emptyset$. (d) $\mathcal{X} \cap \mathcal{Z} \neq \emptyset$ but $\mathcal{Z} \nsubseteq \mathcal{X}$ and $\mathcal{X} \nsubseteq \mathcal{Z}$.

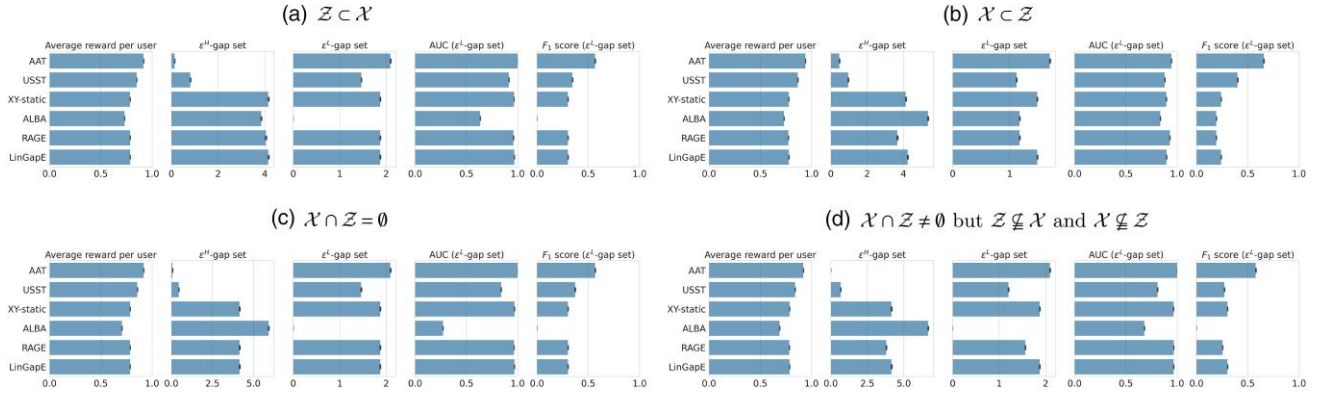
scenario 1, where ALBA performs best with 3.85 cards. Additionally, AAT surpasses USST in the ε^H gap, which selects approximately 0.46–0.96 cards in intolerable sets.

In the ε^L-gap set, our results align closely with those of the top benchmark, achieving an improvement of approximately 12%–15% over the acquisition set and 43%–74% over USST. Note that AAT exclusively selects cards from the predicted tolerance set, potentially resulting in fewer selected cards compared with other methods (which always select 10 cards). Consequently, comparing the number of selected cards in the ε^L gap as a metric may not be entirely fair. To present our results in a way that excludes the number of selected cards as a factor, we also employ AUC and F_1 scores in our performance analysis. Our model improves the AUC of the best acquisition benchmark by approximately 22%–100% and exceeds USST by about 31%–43%. The F_1 scores range from 38% to 55% for the best acquisition benchmark and from 31% to 43% for USST. This result underscores the efficacy of our approach in accurately predicting user behavior, even when the number of selected cards is not a factor in the evaluation.

6. Conclusion

In this paper, we introduce a new pure exploration problem, where the objective is to return all cards (up to K)

Figure 7. (Color online) Performance on the $(K, \varepsilon^H, \varepsilon^L)$-Optimal Set Recommendation



Notes. The y axes correspond to the methods, and the x axes correspond to the performance metrics. (a) $\mathcal{Z} \subset \mathcal{X}$. (b) $\mathcal{X} \subset \mathcal{Z}$. (c) $\mathcal{X} \cap \mathcal{Z} = \emptyset$. (d) $\mathcal{X} \cap \mathcal{Z} \neq \emptyset$ but $\mathcal{Z} \nsubseteq \mathcal{X}$ and $\mathcal{X} \nsubseteq \mathcal{Z}$.

in the $(K, \varepsilon^H, \varepsilon^L)$-optimal set in light of user preference heterogeneity. We further propose an adaptive split and acquisition algorithm (AAT) to identify cards on SFV platforms with a $(K, \varepsilon^H, \varepsilon^L)$-optimal set guarantee. The algorithm accommodates user preference heterogeneity and can group users dynamically. Further, we prove the sample complexity for identifying the $(K, \varepsilon^H, \varepsilon^L)$-optimal set both for a single user and for all users.

AAT's application is not limited to SFV platforms but extends to other digital platforms or online recommender systems, where features vary and user preference heterogeneity is high. AAT can be applied to other social media platforms (e.g., Twitter and Facebook) and search engines (e.g., Microsoft Bing). It can aid marketers who are designing ads and who want to test a small set of advertisements incorporating certain design characteristics; they can make more effective decisions by learning user preferences from a larger set of designs (Schwartz et al. 2017). Moreover, AAT is beneficial in conducting consumer searches, in which a number of results are displayed for each user query (Liu et al. 2021, Derakhshan et al. 2022).

Several future research directions emerge from our study. First, our problem formulation and the algorithm deal with stationary user preferences because the exploration is performed within a short time horizon. Considering nonstationary user preferences would be a natural extension of our work. Second, our two-phase framework can be applied to other managerial problems in marketing analytics and revenue management, such as a generalized linear demand function to characterize user behaviors and applying our method to determine pricing and assortment strategies. Nonparametric pricing has garnered considerable attention (e.g., Avramidis and den Boer 2021, Chen and Gallego 2021, Chen et al. 2023), and AAT could augment existing methods by providing a transductive framework that accounts for both user and choice set features. Third, future research can develop a more robust content moderation tool by improving AAT and building on our optimal set concept. Content moderation is crucial and complex because of the high volume and ambiguity of content on social media platforms. Future research could explore combining AAT with other moderation techniques to create an effective moderation system. We leave these directions for future research.

Acknowledgments

The authors thank Department Editor Omar Besbes, the associate editor, and anonymous referees for their constructive and insightful review comments. Junyu Cao and Yan Leng contributed equally to the manuscript.

Endnotes

[1] TikTok is a video-sharing, social-networking service. The platform is used to make SFVs that include genres like dance, comedy, and education; the videos typically range from 15 seconds to one minute in length. NetEase Cloud Music (NCM) is one of the largest music-streaming companies in China. First launched in 2012, NetEase had around 800 million users in 2019, with a valuation of around $9 billion.

[2] The comparisons of recommendations between SFV platforms and conventional business-to-consumer (B2C) marketplaces are discussed in Online Appendix A.

[3] A music video card (hereafter referred to as a card) contains a video or a set of pictures and texts along with background music (Zhang et al. 2022). We use the terms cards, items, and arms interchangeably.

[4] This problem is *transductive* if the measurement sets in the exploration period (denoted by $\mathcal{X}$) differ from the selection sets in the exploitation period (denoted by $\mathcal{Z}$).

[5] A research stream in this area studies strategies for quality control on the two-sided market, such as minimum quality standards (Ronnen 1991) and exclusive distribution and optimal quality thresholds (see Huang et al. 2007 and the references therein). However, such strategies are not realistic for UGC platforms.

[6] Our research is also relevant for personalized recommendation, which we discuss in Online Appendix B.

[7] Platforms can use different metrics to quantify the reward based on the application of interest. For example, platforms can use view time, likes, and shares for the spotlight or front-page recommendations and can use click-through rate for identifying the optimal set for videos with advertisements. For identifying kid-friendly sets, the reward is negative in relation to irritating or annoying content.

[8] Consumers may have an innate preference for newer and up-to-date cards. This innate preference for newer cards can be captured by using a "recency" variable in the card features to measure the timeliness of a card (e.g., published time).

[9] According to a survey by the marketing intelligence firm Hubspot, 36% of social media marketers planned to invest more in SFVs than any other social media marketing strategy in 2022 (HubSpot 2025).

[10] We note the option of using off-policy evaluation (OPE) to assess the method's effectiveness. Specifically, OPE involves two steps: (1) excluding units for which the observed treatment does not align with the proposed treatment under a given policy and (2) calculating the policy value based on the remaining units (Dudík et al. 2014, Leng and Dimmery 2024). However, in our specific setting, given the vast customer base and expansive item sets, the "treated" scenarios—that is, instances where the observed and proposed treatments align—are remarkably scarce. This scarcity of "treated" scenarios could lead to a highly skewed and potentially unrepresentative evaluation if we were to apply OPE.

[11] The number of bins is chosen by the elbow method. We run a lasso regression with 10-fold crossvalidation to learn user preferences on cards within each cluster. We present the differences between θ in different clusters in Online Appendix I.

[12] Precision measures the number of true-positive results divided by the total number of positive results, including those not identified correctly. Meanwhile, recall measures the number of true-positive results divided by the total number of samples that should have been identified as positive. By using these metrics, we can ensure that our recommendation system is accurate and effective.

[13] We use the same number of samples acquired by AAT to train USST. However, we want to emphasize that USST is not an acquisition algorithm and cannot tell us when to stop for identifying the optimal set with high probability.

[14] The improvement is computed by taking the absolute improvement normalized by the room for improvement (1 − the accuracy of the best benchmark).

[15] The improvement in the average reward is computed as the improvement in the average reward normalized by the room for improvement, which is the difference between the maximum reward (one in our case) and the benchmark.

References

Abernethy JD, Amin K, Zhu R (2016) Threshold bandits, with and without censored feedback. *Adv. Neural Inform. Processing Systems*, vol. 29 (Curran Associates Inc., Red Hook, NY), 4889–4897.

Aouad A, Elmachtoub AN, Ferreira KJ, McNellis R (2023) Market segmentation trees. *Manufacturing Service Oper. Management* 25(2):648–667.

Avramidis AN, den Boer AV (2021) Dynamic pricing with finite price sets: A non-parametric approach. *Math. Methods Oper. Res.* 94(1):1–34.

Baardman L, Levin I, Perakis G, Singhvi D (2018) Leveraging comparables for new product sales forecasting. *Production Oper. Management* 27(12):2340–2343.

Ban G-Y, Gallien J, Mersereau AJ (2019) Dynamic procurement of new products with covariate information: The residual tree method. *Manufacturing Service Oper. Management* 21(4):798–815.

Bastani H, Zhang DJ, Zhang H (2022a) Applied machine learning in operations management. Babich V, Birge JR, Hilary G, eds. *Innovative Technology at the Interface of Finance and Operations*, Springer Series in Supply Chain Management, vol. 11 (Springer, Cham, Switzerland), 189–222.

Bastani H, Harsha P, Perakis G, Singhvi D (2022b) Learning personalized product recommendations with customer disengagement. *Manufacturing Service Oper. Management* 24(4):2010–2028.

Bernstein F, Modaresi S, Sauré D (2019) A dynamic clustering approach to data-driven assortment personalization. *Management Sci.* 65(5):2095–2115.

Besbes O, Gur Y, Zeevi A (2016) Optimization in online content recommendation services: Beyond click-through rates. *Manufacturing Service Oper. Management* 18(1):15–33.

Bubeck S, Wang T, Viswanathan N (2013) Multiple identifications in multi-armed bandits. *Internat. Conf. Machine Learn.* (PMLR, New York), 258–265.

Cao J, Sun W (2019) Dynamic learning of sequential choice bandit problem under marketing fatigue. *Proc. AAAI Conf. Artificial Intelligence* 33(1):3264–3271.

Cao J, Sun W, Shen Z-JM (2019) Doubly adaptive cascading bandits with user abandonment. Preprint, submitted April 8, http://dx.doi.org/10.2139/ssrn.3355211.

Chan K-Y, Loh W-Y (2004) Lotus: An algorithm for building accurate and comprehensible logistic regression trees. *J. Comput. Graph. Statist.* 13(4):826–852.

Chen N, Gallego G (2021) Nonparametric pricing analytics with customer covariates. *Oper. Res.* 69(3):974–984.

Chen L, Li J (2015) On the optimal sample complexity for best arm identification. Preprint, submitted November 12, https://arxiv.org/abs/1511.03774.

Chen Y-C, Mišić VV (2022) Decision forest: A nonparametric approach to modeling irrational choice. *Management Sci.* 68(10):7090–7111.

Chen N, Gallego G, Tang Z (2019) The use of binary choice forests to model and estimate discrete choices. Preprint, submitted August 3, https://arxiv.org/abs/1908.01109.

Chen L, Li J, Qiao M (2017b) Nearly instance optimal sample complexity bounds for top-k arm selection. *Proc. 20th Internat. Conf. Artificial Intelligence Statist.* (PMLR, New York), 101–110.

Chen J, Chen X, Zhang Q, Zhou Y (2017a) Adaptive multiple-arm identification. *Internat. Conf. Machine Learn.* (PMLR, New York), 722–730.

Chen N, Cire AA, Hu M, Lagzi S (2023) Model-free assortment pricing with transaction data. *Management Sci.* 69(10):5830–5847.

Chen S, Lin T, King I, Lyu MR, Chen W (2014) Combinatorial pure exploration of multi-armed bandits. *Adv. Neural Inform. Processing Systems*, vol. 27 (Curran Associates Inc., Red Hook, NY).

Cheung M-C (2020) Why short-form video apps are so popular in China. *eMarketer* (January 2), https://www.emarketer.com/content/why-short-form-video-apps-are-so-popular-in-china.

Cheung WC, Simchi-Levi D, Wang H (2017) Dynamic pricing and demand learning with limited price experimentation. *Oper. Res.* 65(6):1722–1731.

Cohen MC, Zhang R, Jiao K (2022) Data aggregation and demand prediction. *Oper. Res.* 70(5):2597–2618.

Derakhshan M, Golrezaei N, Manshadi V, Mirrokni V (2022) Product ranking on online platforms. *Management Sci.* 68(6):4024–4041.

Dudík M, Erhan D, Langford J, Li L (2014) Doubly robust policy evaluation and optimization. *Statist. Sci.* 29(4):485–511.

Elmachtoub AN, McNellis R, Oh S, Petrik M (2017) A practical method for solving contextual bandit problems using decision trees. Preprint, submitted June 14, https://arxiv.org/abs/1706.04687.

Even-Dar E, Mannor S, Mansour Y (2006) Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *J. Machine Learn. Res.* 7(39):1079–1105.

Ferreira KJ, Hong B, Lee A, Simchi-Levi D (2016) Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing Service Oper. Management* 18(1):69–88.

Fiez T, Jain L, Jamieson K, Ratliff L (2019) Sequential experimental design for transductive linear bandits. *Adv. Neural Inform. Processing Systems*, vol. 33 (Curran Associates Inc., Red Hook, NY).

Gabillon V, Ghavamzadeh M, Lazaric A (2012) Best arm identification: A unified approach to fixed budget and fixed confidence. *Adv. Neural Inform. Processing Systems*, vol. 25 (Curran Associates Inc., Red Hook, NY).

Gelada C, Bellemare MG (2019) Off-policy deep reinforcement learning by bootstrapping the covariate shift. *Proc. AAAI Conf. Artificial Intelligence* 33(1):3647–3655.

Gentile C, Li S, Zappella G (2014) Online clustering of bandits. *Internat. Conf. Machine Learn.* (PMLR, New York), 757–765.

Gentile C, Li S, Kar P, Karatzoglou A, Zappella G, Etrue E (2017) On context-dependent clustering of bandits. *Internat. Conf. Machine Learn.* (PMLR, New York), 1253–1262.

Gretton A, Smola A, Huang J, Schmittfull M, Borgwardt K, Schölkopf B (2009) Covariate shift by kernel mean matching. *Dataset Shift Machine Learn.* (MIT Press, Cambridge, MA), 131–160.

Hu K, Acimovic J, Erize F, Thomas DJ, Van Mieghem JA (2019) Forecasting new product life cycle curves: Practical approach and empirical analysis. *Manufacturing Service Oper. Management* 21(1):66–85.

Huang Z, Zeng DD, Chen H (2007) Analyzing consumer-product graphs: Empirical findings and applications in recommender systems. *Management Sci.* 53(7):1146–1164.

HubSpot (2025) The top video marketing tactics brands are investing in [+which are losing steam]. Accessed August 10, 2025, https://blog.hubspot.com/marketing/top-video-marketing-tactics?

Jagabathula S, Subramanian L, Venkataraman A (2018) A model-based embedding technique for segmenting customers. *Oper. Res.* 66(5):1247–1267.

Jedra Y, Proutiere A (2020) Optimal best-arm identification in linear bandits. *Adv. Neural Inform. Processing Systems*, vol. 33 (Curran Associates Inc., Red Hook, NY), 10007–10017.

Jiang H, Li J, Qiao M (2017) Practical algorithms for best-*K* identification in multi-armed bandits. Preprint, submitted May 19, https://arxiv.org/abs/1705.06894.

Kalyanakrishnan S, Stone P (2010) Efficient selection of multiple bandit arms: Theory and practice. *Proc. 27th Internat. Conf. Internat. Conf. Machine Learn.* (Omnipress, Madison, WI).

Karnin Z, Koren T, Somekh O (2013) Almost optimal exploration in multi-armed bandits. *Internat. Conf. Machine Learn*. (PMLR, New York), 1238–1246.

Kaufmann E, Cappé O, Garivier A (2016) On the complexity of best-arm identification in multi-armed bandit models. *J. Machine Learn. Res.* 17(1):1–42.

Kazerouni A, Wein LM (2021) Best arm identification in generalized linear bandits. *Oper. Res. Lett.* 49(3):365–371.

Keskin NB, Li Y, Sunar N (2024) Data-driven clustering and feature-based retail electricity pricing with smart meters. *Oper. Res.*, ePub ahead of print September 3, https://doi.org/10.1287/opre.2022.0112.

Kumar A, Hosanagar K (2019) Measuring the value of recommendation links on product demand. *Inform. Systems Res.* 30(3):819–838.

Landwehr N, Hall M, Frank E (2005) Logistic model trees. *Machine Learn.* 59(1–2):161–205.

Lattimore T, Szepesvári C (2020) *Bandit Algorithms* (Cambridge University Press, Cambridge, UK).

Leng Y, Dimmery D (2024) Calibration of heterogeneous treatment effects in randomized experiments. *Inform. Systems Res.* 35(4): 1721–1742.

Leng Y, Ruiz R, Liu X (2020) Interpretable recommendations and user-centric explanations with geometric deep learning. Preprint, submitted November 13, https://dx.doi.org/10.2139/ssrn.3696092.

Liu J, Toubia O, Hill S (2021) Content-based model of web search behavior: An application to TV show search. *Management Sci.* 67(10):6378–6398.

Lorenz T (2020) This is why you heard about TikTok so much in 2020. *New York Times* (December 31), https://www.nytimes.com/2020/12/31/style/tiktok-trends-2020.html.

Mason B, Jain L, Tripathy A, Nowak R (2020) Finding all ε-good arms in stochastic bandits. *Adv. Neural Inform. Processing Systems*, vol. 33 (Curran Associates Inc., Red Hook, NY).

Matamoros-Fernández A, Farkas J (2021) Racism, hate speech, and social media: A systematic review and critique. *Television New Media* 22(2):205–224.

Miao S, Chen X, Chao X, Liu J, Zhang Y (2022) Context-based dynamic pricing with online clustering. *Production Oper. Management* 31(9):3559–3575.

Mišić VV (2020) Optimization of tree ensembles. *Oper. Res.* 68(5):1605–1624.

Nguyen TT, Lauw HW (2014) Dynamic clustering of contextual multi-armed bandits. *Proc. 23rd ACM Internat. Conf. Inform. Knowledge Management* (ACM, New York), 1959–1962.

Quinlan JR (1992) Learning with continuous classes. *Proc. 5th Australian Joint Conf. Artificial Intelligence*, vol. 92 (World Scientific, Singapore), 343–348.

Ren W, Liu J, Shroff NB (2019) Exploring k out of top ρ fraction of arms in stochastic bandits. *22nd Internat. Conf. Artificial Intelligence Statist.* (PMLR, New York), 2820–2828.

Ronnen U (1991) Minimum quality standards, fixed costs, and competition. *RAND J. Econom.* 22(4):490–504.

Russo D (2016) Simple Bayesian algorithms for best arm identification. *Conf. Learn. Theory* (PMLR, New York), 1417–1418.

Schwartz EM, Bradlow ET, Fader PS (2017) Customer acquisition via display advertising using multi-armed bandit experiments. *Marketing Sci.* 36(4):500–522.

Shapiro A, Dentcheva D, Ruszczyński A (2014) *Lectures on Stochastic Programming: Modeling and Theory* (SIAM, Philadelphia).

Shi Z, Raghu TS (2020) An economic analysis of product recommendation in the presence of quality and taste-match heterogeneity. *Inform. Systems Res.* 31(2):399–411.

Soare M, Lazaric A, Munos R (2014) Best-arm identification in linear bandits. *Adv. Neural Inform. Processing Systems*, vol. 27 (Curran Associates Inc., Red Hook, NY).

Storkey AJ, Sugiyama M (2007) Mixture regression for covariate shift. *Adv. Neural Inform. Processing Systems*, vol. 19 (Curran Associates Inc., Red Hook, NY), 1337.

Tao C, Blanco S, Zhou Y (2018) Best arm identification in linear bandits with linear dimension dependency. *Internat. Conf. Machine Learn.* (PMLR, New York), 4877–4886.

Wallaroo (2021) TikTok statistics. (May 7), https://wallaroomedia.com/blog/social-media/tiktok-statistics/.

Xu L, Honda J, Sugiyama M (2018) A fully adaptive algorithm for pure exploration in linear bandits. *Internat. Conf. Artificial Intelligence Statist.* (PMLR, New York), 843–851.

Yang J, Liu C, Teng M, Liao M, Xiong H (2016) Buyer targeting optimization: A unified customer segmentation perspective. *2016 IEEE Internat. Conf. Big Data* (IEEE, Piscataway, NJ), 1262–1271.

Yu K, Bi J, Tresp V (2006) Active learning via transductive experimental design. *Proc. 23rd Internat. Conf. Machine Learn.* (PMLR, New York), 1081–1088.

Zhang DJ, Hu M, Liu X, Wu Y, Li Y (2022) NetEase cloud music data. *Manufacturing Service Oper. Management* 24(1):275–284.