KuaiRand: An Unbiased Sequential Recommendation Dataset with Randomly Exposed Videos

Chongming Gao∗
chongming.gao@gmail.com
University of Science and Technology of China

Shijun Li∗
lishijun@mail.ustc.edu.cn
University of Science and Technology of China

Yuan Zhang∗
yuanz.pku@gmail.com
Kuaishou Technology Co., Ltd.
China

Jiawei Chen
sleepyhunt@zju.edu.cn
Zhejiang University

Biao Li
libiao@kuaishou.com
Kuaishou Technology Co., Ltd.

Wenqiang Lei
wenqianglei@gmail.com
Sichuan University

Peng Jiang
jiangpeng@kuaishou.com
Kuaishou Technology Co., Ltd.
China

Xiangnan He
xiangnanhe@gmail.com
University of Science and Technology of China

ABSTRACT

Recommender systems deployed in real-world applications can have inherent exposure bias, which leads to the biased logged data plaguing the researchers. A fundamental way to address this thorny problem is to collect users’ interactions on randomly expose items, i.e., the missing-at-random data. A few works have asked certain users to rate or select randomly recommended items, e.g., Yahoo! [16], Coat [22], and OpenBandit [19]. However, these datasets are either too small in size or lack key information, such as unique user ID or the features of users/items. In this work, we present KuaiRand, an unbiased sequential recommendation dataset containing millions of intervened interactions on randomly exposed videos, collected from the video-sharing mobile App, Kuaishou. Different from existing datasets, KuaiRand records 12 kinds of user feedback signals (e.g., click, like, and view time) on randomly exposed videos inserted in the recommendation feeds in two weeks. To facilitate model learning, we further collect rich features of users and items as well as users’ behavior history. By releasing this dataset, we enable the research of advanced debiasing large-scale recommendation scenarios for the first time. Also, with its distinctive features, KuaiRand can support various other research directions such as interactive recommendation, long sequential behavior modeling, and multi-task learning. The dataset and its news will be available at https://kuairand.com.

CCS CONCEPTS

• Information systems → Recommender systems.

1 INTRODUCTION

Recommender systems have become a powerful tool for tech companies to connect users with appropriate items or services. Although developing and evaluating a recommendation model with real users are effective, online A/B testing is time- and money-consuming and entails the risk of hurting customer engagement, thus impairing the platform’s value proposition [7, 21]. Hence, most of the efforts have been put into developing advanced recommendation models on the offline data generated from the interaction logs.

However, the bottleneck becomes the offline data per se. Since the highly sparse data comes from the collected historical interactions between users and the system, it is unavoidably affected by the exposure bias [3, 13, 14]. The biased data has catastrophic effects on researchers and practitioners of recommender systems. It leads to inconsistent online and offline results of the model [22] and
To tackle the problem at the root, we have to collect the user preferences. The fundamental problem lies in the fact that we have no knowledge about the massive missing interactions in the offline data [6]. To tackle the problem at the root, we have to collect the user preferences regarding those previously NOT exposed items. Gao et al. [6] collected a fully-observed dataset with each user’s preferences on each item known. However, this dataset can inevitably be biased towards popular items due to its extremely high density.

A more practical way to obtain truly unbiased data is to collect users’ feedback on uniformly sampled items, i.e., the missing-at-random (MAR) data [29]. It has been demonstrated that we can conduct an unbiased offline evaluation on the MAR data [20, 29]. Few works have considered this solution and collected user feedback on randomly observed items. The well-known datasets Yahoo!R3 [16] and Coat [22] asked a set of people to explicitly rate a certain number of randomly selected items in the score range from 1 to 5. However, the scales of the two datasets are quite limited — only 54,000 ratings in Yahoo! and 4,640 ratings in Coat for the randomly selected items. Besides, both of them lack some key information such as user/item features and timestamp. The recently introduced dataset OpenBandit [19] collected users’ clicks on items from a slate of three items. Each slate is generated by a multi-armed bandit policy or a random policy. However, it only evolves 80 items, and the user identification (ID) information is removed. These insufficient information problems make existing datasets fail to satisfy the requirement of many recommendation tasks.

### 1.1 Open Problem: Collecting Random Data

The fundamental problem lies in the fact that we have no knowledge about the massive missing interactions in the offline data [6]. To tackle the problem at the root, we have to collect the user preferences regarding those previously NOT exposed items. Gao et al. [6] collected a fully-observed dataset with each user’s preferences on each item known. However, this dataset can inevitably be biased towards popular items due to its extremely high density.

A more practical way to obtain truly unbiased data is to collect users’ feedback on uniformly sampled items, i.e., the missing-at-random (MAR) data [29]. It has been demonstrated that we can conduct an unbiased offline evaluation on the MAR data [20, 29]. Few works have considered this solution and collected user feedback on randomly observed items. The well-known datasets Yahoo!R3 [16] and Coat [22] asked a set of people to explicitly rate a certain number of randomly selected items in the score range from 1 to 5. However, the scales of the two datasets are quite limited — only 54,000 ratings in Yahoo! and 4,640 ratings in Coat for the randomly selected items. Besides, both of them lack some key information such as user/item features and timestamp. The recently introduced dataset OpenBandit [19] collected users’ clicks on items from a slate of three items. Each slate is generated by a multi-armed bandit policy or a random policy. However, it only evolves 80 items, and the user identification (ID) information is removed. These insufficient information problems make existing datasets fail to satisfy the requirement of many recommendation tasks.

### 1.2 Contributions: The KuaiRand Dataset

Our goal is to enable unbiased offline evaluation in real-world recommendation scenarios. To this end, we present KuaiRand, a large-scale sequential recommendation dataset, collected from Kuaishou App, one of the China’s largest video-sharing Apps with more than 300 million daily active users. Different from previous datasets, it intervenes in the recommendation policies by randomly inserting randomly chosen videos in the normal recommendation feed for two weeks without user awareness. Hence, we can collect genuine unbiased user reflections on these randomly exposed videos. We recorded all 12 kinds of feedback signals, such as click, add to favorites, and view time. To facilitate model learning, we further collected users’ historical behaviors as well as rich side information of users/items. The process is illustrated in Fig. 1 and the detailed information can be referred to Table 1.

It is worth mentioning that KuaiRand is the first unbiased sequential recommendation data where we intervene in the original recommendation policy millions of times by inserting random items. The unbiased sequential data enables us to carry out the unbiased offline evaluation [20, 29], thus facilitating the research of debiasing [3] in large-scale recommendation scenarios. Furthermore, with its distinctive features, KuaiRand can naturally facilitate a variety of other recommendation tasks, such as interactive recommendation [4, 27], long sequential behavior modeling [17, 18], and multi-task learning [2, 28].

### Table 1: Comparison of currently available datasets with randomly exposed data.

| Dataset        | Collection policy | # Users | # Items | # Interactions | # User features | # Item features | Timestamp | Feedback |
|----------------|-------------------|---------|---------|----------------|----------------|----------------|-----------|----------|
| Yahoo!R3 [16]  | Normal interaction | 15,400  | 1,000   | 311,704        | 0              | 0              | No        | Rating: [1,5] |
|                | Random policy     | 5,400   | 1,000   | 54,000         | 7              | 0              | No        | Rating: [1,5] |
| Yahoo!R6A [11] | Random policy     | No ID   | 271     | 45,811,883     | 6              | 6              | Yes       | Click    |
| Yahoo!R6B [11] | Random policy     | No ID   | 652     | 27,777,695     | 136            | 0              | Yes       | Click    |
| Coat [22]      | User self-selected | 290     | 300     | 6,960         | 4              | 4              | No        | Rating: [1,5] |
|                | Random policy     | 290     | 300     | 4,640         | 4              | 4              | No        | Rating: [1,5] |
| Open Bandit [19]| Bernoulli TS      | No ID   | 80      | 24,011,308    | 4              | 4              | Yes       | Click    |
|                | Random policy     | No ID   | 80      | 2,691,861     | 4              | 4              | Yes       | Click    |
| KuaiRand-27K   | 15 policies       | 27,285  | 3,016,725| 322,278,385  | 30             | 62             | Yes       | 12 signals |
|                | Random policy     | 27,285  | 7,583   | 1,186,059     | 30             | 62             | Yes       | (e.g., click, like, hate) |
| KuaiRand-1K    | 15 policies       | 1,000   | 4,369,953| 11,713,045   | 30             | 62             | Yes       | (e.g., click, like, hate) |
|                | Random policy     | 1,000   | 7,388   | 43,028        | 30             | 62             | Yes       | (e.g., click, like, hate) |
| KuaiRand-Pure  | 15 policies       | 27,285  | 7,551   | 1,436,609     | 30             | 62             | Yes       | (e.g., click, like, hate) |
|                | Random policy     | 27,285  | 7,583   | 1,186,059     | 30             | 62             | Yes       | (e.g., click, like, hate) |

1. Online A/B tests are effective in tech companies to assess the performances of recommender models [7]. However, it usually consumes much time and money, which makes it impractical for academic researchers to conduct the evaluation online.

Therefore, researchers usually resort to offline evaluation, which includes: (1) computing traditional indicators (such as Precision, Recall, and NDCG [9]) on the test set; (2) conducting off-policy evaluation or counterfactual reasoning [1, 23] to perform an offline evaluation.
A/B test [7]; (3) using user simulation techniques [8, 30] to evaluate the recommender in the online A/B test.

However, computing traditional indicators cannot answer how the recommender performs on the massive missing data, i.e., the user-item pairs which have not occurred in the test set. Though off-policy evaluation and user simulation can solve this pitfall, they suffer the problem of high-variance issue [19] and additionally introduced error [5], respectively.

2.2 Datasets with Randomly Exposed Items

A way to fundamentally solve this problem in offline evaluation is to collect unbiased data, i.e., to elicit user preferences on the randomly exposed items. We briefly introduce classic existing high-quality datasets that contain unbiased randomly-sampled data used for offline evaluation in recommendation.

- Yahoo!R3 [16]. It contains the conventional missing-not-at-random (MNAR) data from 15, 400 users on 1,000 items in total. It also contains a set of missing-complete-at-random (MCAR) data by asking 5, 400 users to give ratings on 10 items that are randomly selected from the 1,000 items. However, it does not contain timestamps or the features of users and items.

- Yahoo!R6 [11]. It has two versions, namely R6A and R6B, both of which have only the random policy to expose hundreds of news millions of times. Besides, all user IDs are removed, which forbids it to perform basic collaborative filtering tasks in recommendation.

- Coat [22]. It collects the ratings of 290 users on 24 self-selected items and 16 randomly-selected items from a total of 300 items. Similar to Yahoo!R3, the scale of it is too small compared with other datasets in real-world recommendation scenarios. Neither does it has the timestamp to support research on sequential recommendation.

- Open Bandit Dataset [19]. It contains interactions collected from two logged bandit policies: a Bernoulli Thompson Sampling policy and a uniform (random) policy. There are approximately 26 million interactions collected by users’ clicks on 80 items. The disadvantage of this data is the lack of user ID, making it hard to support the research on sequential recommendation. In addition, the fact that it has only 80 items limits its reliability in evaluating most recommendation algorithms.

The statistics of these datasets are summarized in Table 1. Compared with them, our proposed KuaRand has the following key advantages: (1) it has the most comprehensive side information including explicit user IDs, interaction timestamps, and rich features for users and items; (2) it has 15 policies with each catered for a special recommendation scenario in Kuaishou App; (3) we introduce 12 feedback signals for each interaction to describe the user’s comprehensive feedback; (4) each user has thousands of historical interactions on average. These distinctive characteristics can enable a lot of potential research directions.

3 DATA DESCRIPTION

Here, we introduce the basic characteristics of Kuaishou App and the process of how we collect the KuaRand dataset.

3.1 Characteristics of Kuaishou App

Kuaishou App is one of the largest short video-sharing platforms in China. It has amassed over 300 million daily-active users with an average spent time of nearly 128.1 minutes per day [24]. Different from other recommendation scenarios such as e-commerce, users in short video recommendation platforms can generate very long interaction sequences every day. There are also many new videos published every day, hence the videos become outdated very quickly.

The videos are mainly recommended to users in different 15 scenarios, e.g., under different user interfaces (UIs) in Kuaishou App or the embedded UI in other Apps such as WeChat Mini Program. Each scenario has distinctive rules and objectives. Therefore, each scenario has its own recommendation policy and there are 15 policies in total. We illustrate the most two commonly used UIs in two scenarios in Fig. 2.

On the left-hand side lies the main UI, where the thumbnails of recommended videos are organized in a two-column cascade style. In this UI, users can scroll up or down until they find the video they want to view, and then view this video by clicking the thumbnail. Therefore, click is a critical signal indicating user interest in this UI style. On the right-hand side, we can see the single-column recommendation cascading UI. It is the most commonly used UI. The video will automatically play whenever switching to this UI. There is no click signal in the two-column UI, but users can choose to stop viewing this video by exiting this UI or scrolling to the last/next video. Therefore, view time in this UI becomes an important signal to reflect user satisfaction. Besides, there are other signals representing users’ behaviors when they interact with this video, including like, enter author profile, follow author. These feedback signals correspond to the buttons on this UI (Fig. 2), so we can easily collect these timely signals.

3.2 Data Collection

Our goal is to collect the sequential recommendation data with randomly exposed videos inserted. We achieve this goal via the following three steps:
3.3 Statistics and Usage

The basic statistics of KuaiRand are summarized in Table 1. We will release three versions (KuaiRand-27K, KuaiRand-1K, and KuaiRand-Pure) for different usages.

For KuaiRand-27K, we visualize the distribution of the users w.r.t. the numbers of interactions Fig. 3. An interaction can either be a normally recommended video or a randomly chosen video. From the figure, we can see that most users in this dataset have viewed thousands of recommended videos in one month. Each user has more than 10 times viewed random videos. For the detailed usage and news of this dataset, please refer to https://kuairand.com.

We hide the exact value for a commercial security reason, i.e., to avoid leaking out the key parameters of the online recommendation strategy.

4 POTENTIAL RESEARCH DIRECTIONS

Through releasing KuaiRand, we offer the opportunity to carry out the debiasing task in the large-scale sequential recommendation for the first time.

- **Debiasing in Recommendation.** Recommender systems suffer from various biases in the data collection stage [3]. Most existing datasets are very sparse and affected by user-selection bias [16] or exposure bias (previous model bias) [13]. It is of critical importance to develop models that can alleviate biases. To evaluate the models, we need reliable unbiased data. KuaiRand is the first dataset that inserts the random items into the normal recommendation feeds with rich side information and all item/user IDs provided. With this authentic unbiased data, we can evaluate and thus improve the recommender policy.

With its distinctive features, KuaiRand can further support the following promising research directions in recommendation.

- **Off-policy Evaluation (OPE).** OPE is proposed to overcome the difficulties in conducting online A/B testing in recommendation. OPE aims to estimate the performance of a counterfactual policy using only log data generated by past policies [19]. It requires that we have the log sequences rather than merely positive samples. KuaiRand satisfies this requirement. It recorded logs under 15 policies catered for different situations. Therefore, we can leverage OPE and conduct offline A/B tests to improve our recommendation policies [7].

- **Interactive Recommendation.** An interactive recommender system (IRS) is usually formulated as a decision-making process to pursue long-term success in recommendation. IRS built on reinforcement learning has shown superiority to the traditional supervised frameworks [4, 26, 27]. KuaiRand contains the sequential logs generated from different policies that can naturally support the research on IRSs with the help of OPE and user simulation techniques [5, 10].

- **Long Sequential Behavior Modeling.** In practice, traditional sequential recommendation models, such as DIN [32] and DIEN [31], can only model a maximum of hundreds of items in a sequence. Recently, some works have demonstrated that modeling users’ long sequential behaviors is very effective in lifting click-through rate (CTR) [17, 18]. KuaiRand has thousands of interactions for each user, which is perfect for the long sequential user modeling task.

- **Multi-Task Learning.** In real-world recommender systems, user satisfaction can be reflected in multiple behaviors, e.g., clicks, writing down positive comments, and staying longer. Usually, these metrics cannot be pursued at the same time in one model. Therefore, multi-task learning or multi-object learning has received growing attention [2, 25, 28]. KuaiRand recorded 12 feedback signals for each interaction, which promotes the research on learning with more than ten tasks. In contrast, most existing datasets in multi-task recommendation [12, 15] only provide less than 5 signals.

5 ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China (U19A2079, 62102382).
