Записки научных
семинаров ПОМИ
Том 530, 2023 г.

M. Shirokikh, I. Shenbin, A. Alekseev, A. Volodkevich, A. Vasilev, S. I. Nikolenko

USER RESPONSE MODELING IN RECOMMENDER SYSTEMS: A SURVEY

Abstract. Over the last several decades, recommender systems have become an integral part of both our daily lives and the research frontier at machine learning. In this survey, we explore various approaches to developing simulators for recommendation systems, especially for modeling the user response function. We consider simple probabilistic models, approaches based on generative adversarial networks, and full-scale simulators, and also review the datasets available for the research community.

§1. Introduction

Over the last several decades, recommender systems have become an integral part of our daily lives, especially in domains such as e-commerce, social networks, and content streaming platforms. These systems analyze user behavior and preferences to provide personalized recommendations, enhancing user experience and improving user engagement. However, evaluating the effectiveness of recommendation algorithms and understanding their behavior in a real world setting can be a challenging task. This is where simulators for recommendation systems come into play.

Simulators are programs specifically designed to fit given datasets and provide a realistic way for modeling new *synthetic* users, items, and/or their interactions based on previous interaction history. These simulations allow researchers and developers to study the behavior and performance of recommender algorithms under controlled conditions. By using simulators, researchers can conduct extensive experiments without impacting real user experiences, production systems, or requiring access to sensitive user data. Another key usage scenario of such simulators is offline training for recommendation systems based on reinforcement learning, which is far cheaper than online learning; the latter is usually infeasible in a real world setting anyway.

Key words and phrases: user response function, recommender systems, adversarial learning, synthetic data.

The present survey is devoted to exploring various approaches to developing simulators for recommendation systems, especially for modeling the user response function. We aim to provide a comprehensive overview of existing literature and present an analysis of different approaches employed in simulators for recommendation systems. We note, however, that there are relatively few works in this area that are actually directly relevant, i.e., present novel simulators with user response modeling, so the survey necessarily branches out to adjacent topics as well.

Existing research can be broadly classified into several categories based on the approaches utilized; our survey follows this categorization:

(1) *simple probabilistic models* (Section 2), from hard-coded action distributions to Bayesian networks and Poisson stochastic processes;
(2) approaches based on *generative adversarial networks* (Section 3), a popular class of models where the generator learns to imitate a user or produce a user model, and the discriminator learns to separate real actions or users from fake ones synthesized by the generator;
(3) *full-scale simulators* (Section 4) that are most directly relevant to our topic and usually are accompanied by efficient implementations; often, such simulators are designed with an explicit goal to provide training environments for recommender systems based on reinforcement learning.

In what follows, we consider each of these subgroups in detail, examine the methodologies, strengths, and limitations of each approach, and discuss their applicability in the context of user response function modeling and for the simulator as a whole. Finally, we note several other interesting approaches (Section 5), present a brief overview of available datasets and list some of the most recent papers that are related to this field.

§2. Simple probabilistic models

In this section, we review several probabilistic models that do not employ complex neural networks but show significant promise in generating synthetic datasets for recommender systems. They are usually based on probabilistic modeling, from simple learning of dataset statistics to complex probabilistic graphical models.

2.1. Early approaches. In this survey, we concentrate on works from the last ≈ 5 years of research, but synthetic data had been used for recommender systems far longer. A 1994 IBM Quest synthetic data generator [8]

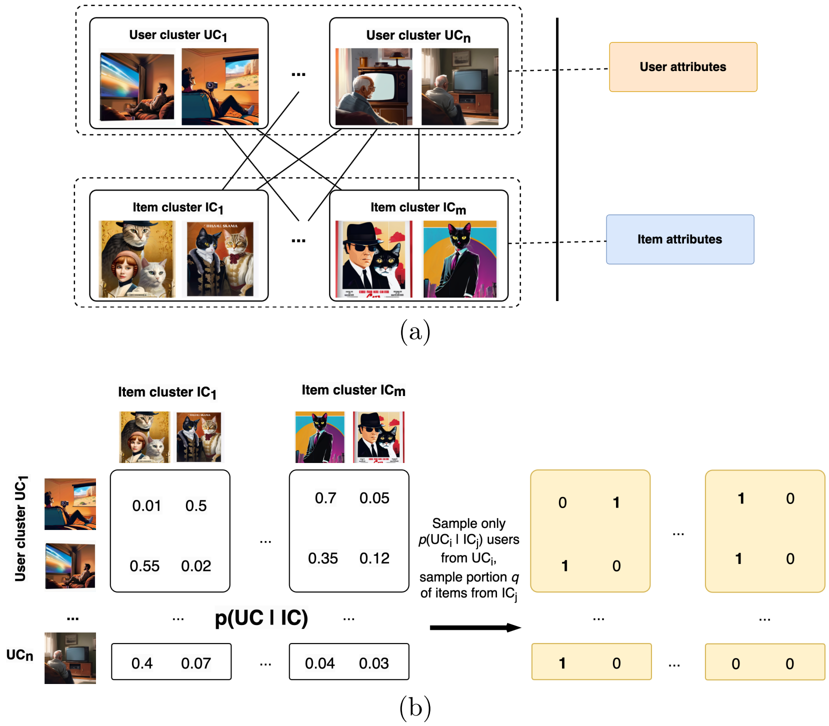


Figure 1. Simple probabilistic model for user and item clusters [118]: (a) structure of synthetic data; (b) illustration of the sampling process.

was intended for evaluating association rule algorithms but was also used to evaluate collaborative filtering models [30]. Early approaches to creating synthetic data based on database schemas have also been applied to evaluate recommender systems [48]. Popescul et al. [99] used a clustering approach with uniform sampling from each cluster while Traupman and Wilensky [117] introduced skewed data according to distributions learned from a real dataset. Marlin et al. [78] produced synthetic data by resampling a real dataset.

Tso and Schmidt-Thieme [118,119] present a simple probabilistic model with user clusters and item clusters; distributions for user and item attributes are drawn from a prior for every cluster, and then ratings are

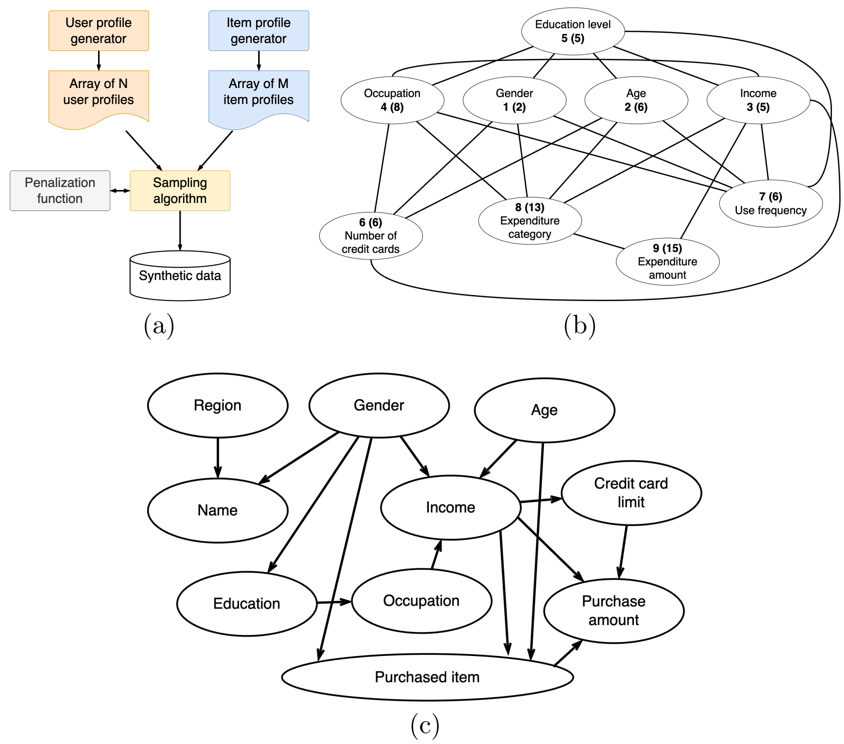
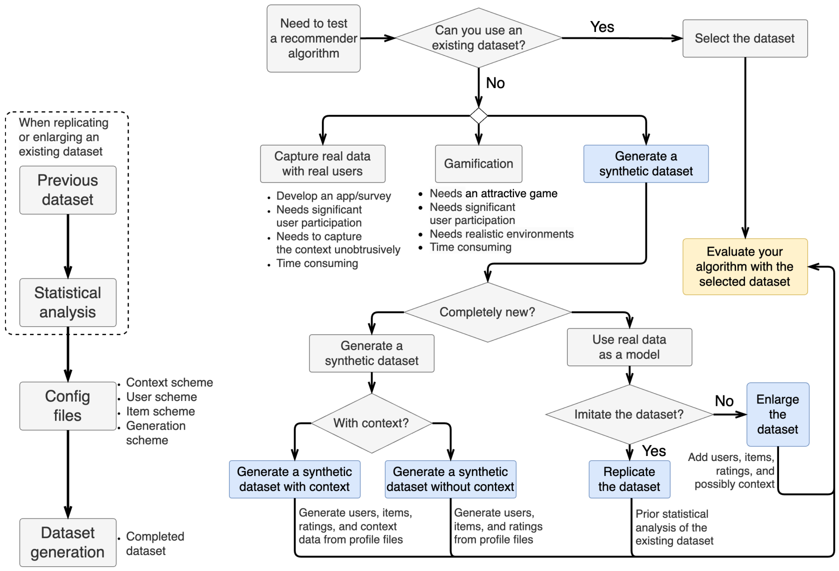


Figure 2. Early probabilistic synthetic data generators: (a) context-aware generation [94]; (b) sample semantic graph for database synthesis [56]; (c) same for [67].

sampled according to the attributes drawn for specific clusters (see Fig. 1 for an illustration). They compare several standard collaborative filtering baselines and find, e.g., which of them are more sensitive to the choice of distribution parameters, an important finding relevant to their generalization ability.

Pasinato et al. [94] propose a synthetic data generator for context-aware recommender systems (CARS); since this approach is based on context attributes related to user-item ratings, a synthetic data generator has to produce contexts as well. Therefore, the system contains user profile generators for user "tastes", product profile generators for item attributes, and



(a) (b)

Figure 3. DataGenCARS [103]: (a) general scheme of operation; (b) testing a recommender algorithm with synthetic data.

a special penalization function that represents the influence of context on ratings (Fig. 2a); see also Section 2.2. This system was similar to other contemporary solutions for generating synthetic datasets based on database schemas and semantic graphs; for example, the works [56, 67] present generators for credit card records based on semantic graphs (see Figs. 2b and 2c respectively for sample such graphs).

2.2. DataGenCARS. DataGenCARS is a synthetic data generator for context-aware recommender systems (CARS) that had given rise to several works on evaluation and development of new recommender systems [28, 29, 53, 103]. It has a full-fledged implementation[1] allowing to define user schemas, context schemas, types of items, user profiles, and various kinds of rating generation. DataGenCARS is also able to learn distributions of

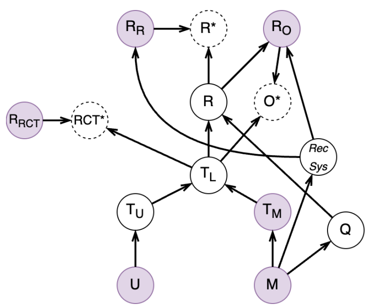
[1]http://webdiis.unizar.es/~maria/?page_id=70

attribute values from a real dataset to complement it with synthetic data. Figure 3a shows the general structure of DataGenCARS, while Figure 3b shows the scheme for using synthetic data in recommender system evaluation that the authors of DataGenCARS propose.

2.3. All You Need Is Ratings. This work by Monti et al. [87] concentrates on generating synthetic rating datasets and poses research questions very similar to ours: "what is the impact of using a synthetic dataset instead of a real one on the results of an offline experiment" and "can a generative approach be exploited to create a synthetic dataset that exhibits properties similar enough to the ones of a real dataset". In their study of collaborative filtering datasets (mostly classical ones such as *MovieLens* and *LastFM*), they argue that global statistical distributions learned from a whole dataset are insufficient since they do not contain individual user preferences, and propose to cluster the users into a fixed number of communities (clusters) that could each have its own distribution of preferences.

As a result, the proposed approach begins with k-means clustering applied to vectors of user preferences (only positive preferences are considered) followed by learning the distributions for (1) number of ratings and (2) items with positive feedback for each cluster individually. The authors compare several collaborative filtering approaches (user k-NN, BPRMF, WRMF) on the generated datasets and find that while the results are generally lower (which is natural since the generated datasets do not reflect real user preferences), the order of CF models ranked by their recommender quality metrics remains the same, a promising property that we would want to hold in new synthetic data generators as well.

2.4. Causal Tags and Ratings. This work by Lyu et al. [74] concentrates on understanding the *causality* behind user ratings. While for "pure" collaborative filtering datasets there are no causes to be inferred, most real datasets have additional information about items that may allow to infer such causes, e.g., "the user likes romantic movies" for movie recommendations or "the user likes everything in pink" for an online store. Causes, however, are never explained in data even when additional item features are present, and it is hard to expect large-scale user surveys with accurate information about such causes, so it is very hard to check whether the causes inferred by a recommender system are real. Hence, the need arises to create synthetic datasets where known causes are already built in and known in advance.



M	Movie id
U	User id
T_M	Tags associated with a movie
T_U	Tags associated with user preferences
T_L	Overlap between T_M and T_U
Q	Quality (intrinsic feature of a movie)
R	Rating
RCT	Observed tags of a movie that indicate user preference, collected via RCT experiment from T_L
O^*	Observed tags of a movie that indicate user preference, collected via observational experiment from T_L
R^*	Observed ratings
R_{RCT}	Missing mechanism associated with RCT
R_O	Missing mechanism associated with O^*
R_R	Missing mechanism associated with R^*

Figure 4. The graphical model (m-graph) for the Causal Tags and Ratings model and corresponding variable descriptions [74].

To do that, Lyu et al. rely on causal graphical models, a formalism similar to directed graphical models but allowing to reason probabilistically about causality [95, 96]; causal models had already been applied to recommender systems [131] but not to synthetic data generation. Lyu et al. introduce the concept of causal tags, particular features that may affect ratings, and introduce into recommender systems missingness graphs (m-graphs), a formalism that adds new nodes to the graphical model to explicitly model various kinds of missing data; this is exactly what is needed for recommender systems since the entire point of such a system is to fill in missing data in the user-item matrix.

Figure 4 shows the resulting m-graph together with the definition of variables. Unfortunately, the authors show the results of only three baselines on generated datasets, and it appears that straightforward matrix factorization is the winner at least half the time, but we believe that using additional features to "anchor" generated ratings, not necessarily with a causal graphical models, is still an interesting idea to explore.

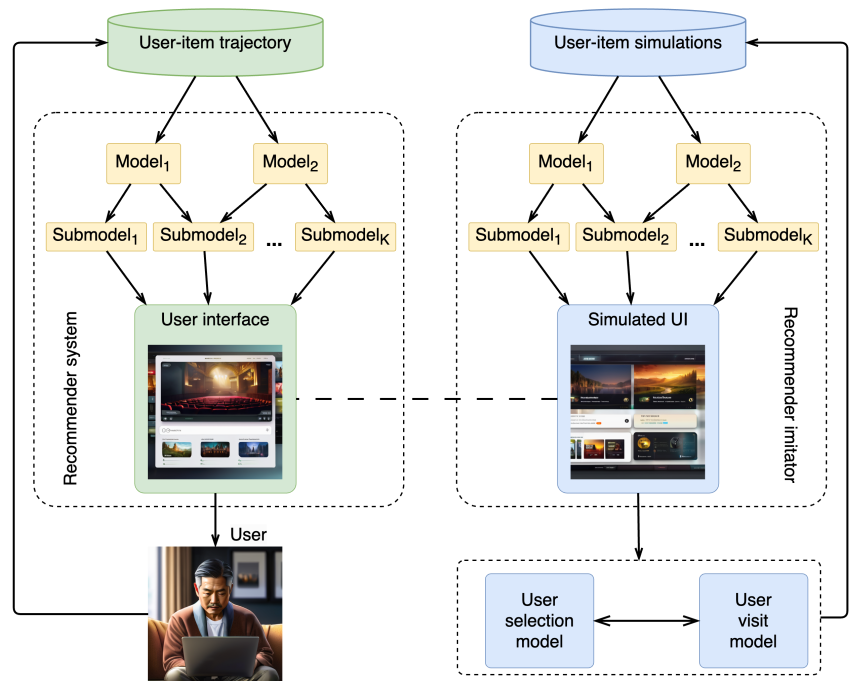
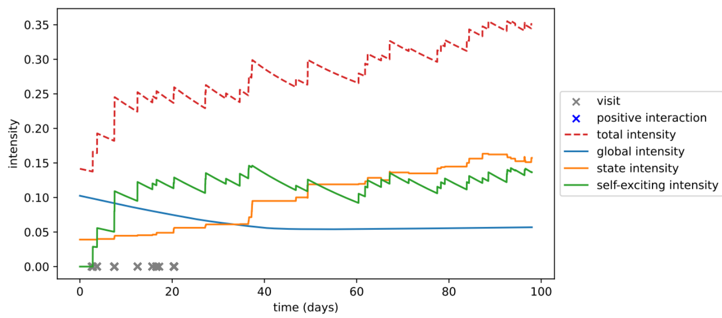


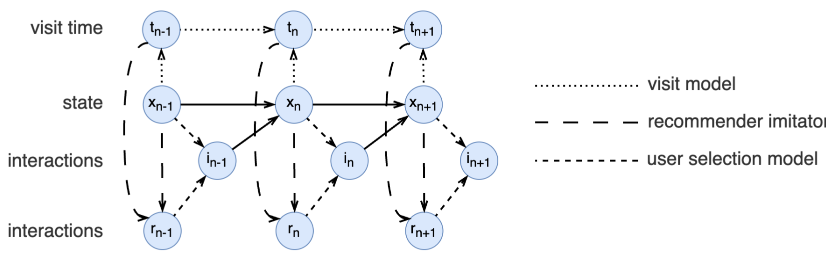
Figure 5. A general scheme of the Accordion simulator [80].

2.5. Accordion. The key idea of the Accordion model [80] is to represent user response time as a Poisson process with an intensity function learned from offline data; intensity λ is the only one parameter of a Poisson process, and in this case it is allowed to vary in time (inhomogeneous Poisson process). Figure 5 shows the general structure of Accordion: the model is learned based on real user trajectories (simulated user-item interactions are compared to real user-item trajectories) and based on the impressions that comprise the output of the recommender system.

Poisson processes have the superposition property: a sum of two Poisson processes with intensities λ_1 and λ_2 is again Poisson with intensity $\lambda_1 + \lambda_2$. Thus, observed intensity can be split into several simple and readily interpretable parts. An example of this effect is shown in Fig. 6a: events from a real user from the *ContentWise* dataset are shown with crosses, and the total intensity is composed of three parts: global (time-related activity



(a)



(b)

Figure 6. The Accordion simulator: (a) superposition of Poisson processes (example from [80]); (b) random variables present in the model.

changes across all users), state (based on user features), and self-exciting intensity (the latter increases after a positive interaction occurring in the history).

The authors of [80] develop novel algorithms for learning deep inhomogeneous Poisson processes. The three components exemplified in Fig. 6a are further illustrated in Figure 6b, which shows the random variables present in the simulator and their interdependencies. Accordion uses a variety of neural architectures (admittedly very simple, mostly dense fully connected networks) to model the three components of the intensity function. As a result, Accordion is reported to be able to simulate long-time dependencies in user histories, capture the effects of interactions on subsequent interactions, and scale up to realistic sized datasets [80].

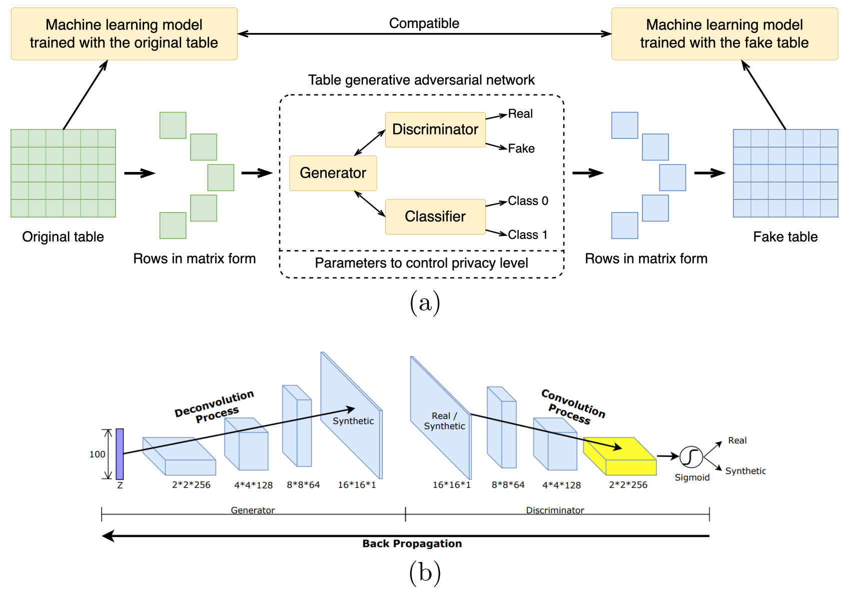
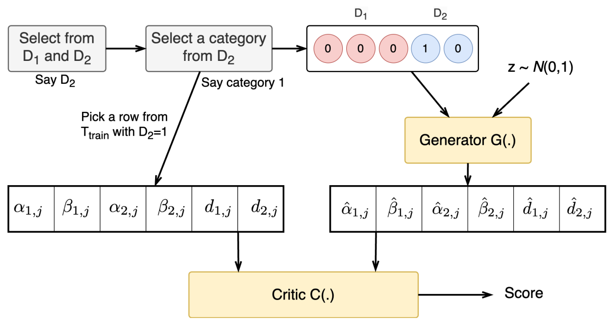


Figure 7. TableGAN [93]: (a) the goal is that machine learning models trained on the fake table should exhibit the same behaviour as trained on the real table; (b) TableGAN architecture: the classifier has the same architecture as the discriminator (illustration from [93]).

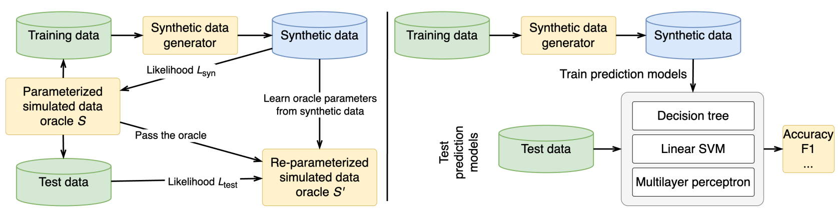
§3. Adversarial approaches in user modeling

3.1. Introduction: discrete generation with GANs. Classical *generative adversarial networks* (GAN) [38,39,81,100] have been traditionally applied to generating continuous objects such as images. Generating discrete variables such as synthetic user behaviour for recommender systems was problematic because gradients cannot flow directly through a generated discrete object in the way needed to train the generator in a standard GAN.

However, several solutions to this problem appeared soon after. Since these solutions can all be potentially helpful for generation of synthetic tabular data for recommender systems, we give a brief overview of the general



(a)



(b)

Figure 8. CTGAN [127]: (a) CTGAN architecture; (b) evaluation frameworks for synthetic data (left) and real data (right).

approaches (see also a recent survey [36] of GANs for spatio-temporal data, including discrete time series):

- medGAN [22] was designed to generate synthetic electronic health records, i.e., high-dimensional discrete variables, with a special emphasis on privacy-related concerns;
- TableGAN [93] also concentrated on synthetic data generation for privacy concerns, but this time with an explicit requirement that machine learning models trained on newly generated tables should show the same results as trained on real data (see Figure 7); for this purpose, TableGAN adds a third network, the classifier, that learns the semantics from the original table in order to increase its semantic integrity;

- CTGAN [127] was able to model the probability distribution of rows in tabular data and generate realistic synthetic data, including a mixture of continuous and discrete columns; the architecture here is relatively standard (Fig. 8a), and the authors also introduce a novel evaluation framework where they test for likelihood fitness (likelihood of a test set) on synthetic data and machine learning efficacy (how good are models trained on synthetic data) for real datasets, as shown in Fig. 8b;
- TVAE, introduced in the same work as CTGAN [127], is not a GAN but a modification of the variational autoencoder for tabular data; in the experiments shown in [127], it performed on par with CTGAN and better than other considered approaches.

These and other works [83] have made it possible to use generative adversarial networks with discrete data. In this part of the survey we consider solutions that apply these or similar techniques to generating synthetic data, especially online user responses, for recommender systems. While GANs have been predominantly employed in image generation tasks, they have also shown great potential in modeling user response; we will see that they can be utilized to generate synthetic user behavior, interactions, and feedback. The generator component of the GAN is responsible for creating realistic user responses (such as item ratings, clicks, or purchases) based on various input factors, including user profiles, item features, and contextual information. The discriminator, on the other hand, aims to distinguish between real and generated user responses, driving the generator to refine its output to become more indistinguishable from real user data.

We note a separate direction of research where adversarial components are introduced into the recommender system itself [35]. Next we note some important works but do not consider it in detail since in this direction, generative adversarial networks are not used to generate synthetic data from scratch but rather to improve the training process on existing datasets. There are two primary ways to use them:

- either to mitigate data noise, both causal and malicious noise, by adding adversarial perturbations to input data for recommender systems [18, 45, 66, 113, 114, 116, 129],
- or to distinguish informative samples in the vast majority of unobserved data for negative sampling in recommender systems based on contrastive learning; here we note IRGAN [120], collaborative

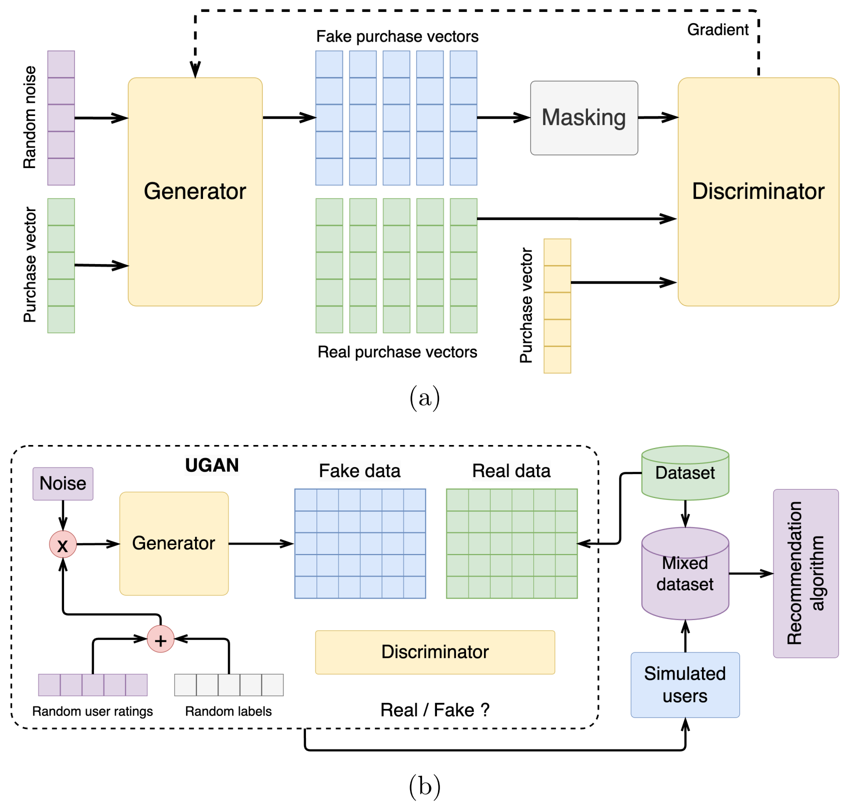
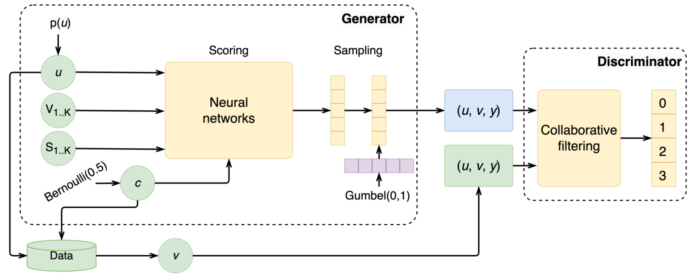


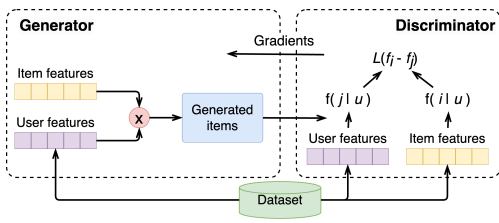
Figure 9. GAN-based generation of user feedback for data augmentation: (a) CFGAN [17, 35]; (b) UGAN [124].

filtering GAN (CoFiGAN) [71], and several other works [15, 31, 92, 101, 122].

3.2. Adversarial Generation of User Feedback to Reduce Data Sparsity. In this section, we consider GAN-based approaches that were designed to generate additional data to either augment existing datasets or alleviate their excessive sparsity. These methods can be thought of as a form of data augmentation for recommender system datasets.



(a)



(b)

Figure 10. GAN-based generation of user feedback for data augmentation: (a) AugCF [123]; (b) APL [35, 109].

One of the first GAN-based models to generate additional data for learning recommender systems was CFGAN [17]. As illustrated in Figure 9a, it generates user preference vectors in an adversarial fashion, with the generator producing user preferences and a discriminator trying to distinguish them from real user preference vectors after masking the generated ones with real purchases. CFGAN also served as the basis for a rating augmentation framework RAGAN explicitly designed to decrease data sparsity characteristic for recommender datasets [16].

UGAN (Unified GAN) [124] uses a generic adversarial architecture to generate user profiles (Fig. 9b); the authors show that recommender system results improve on several standard datasets (such as DouBan and MovieLens) after data augmentation with UGAN-generated user profiles.

AugCF [123] brings GAN-based generation down to the level of individual interactions: for a given user u sampled from some prior distribution $p(u)$, we sample a class $c \in \{0, 1\}$ at random and choose a sample of K items $V_{1..K}$ with their side information $S_{1..K}$, from which the generator

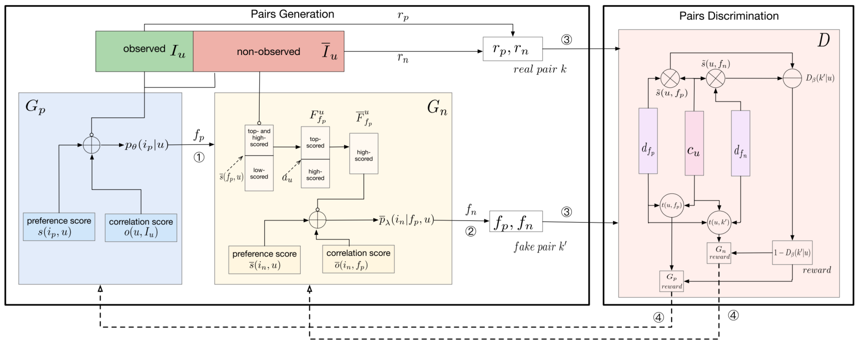


Figure 11. The workflow and high-level architecture of IPGAN (illustration from [40]).

produces the probabilities of each of these items being the user's most preferred item in a given interaction category, samples one (user, item, category) triple via the Gumbel-softmax trick, and the discriminator discerns whether the resulting (u, v, y) triple is realistic (Fig. 10a).

We note that all of the above approaches to extending recommender datasets with GANs claim that with their data augmentations, results of classical and/or state of the art recommender systems on standard datasets improve, sometimes quite significantly.

3.3. Adversarial Generation for Constrastive Learning. Adversarial pairwise learning (APL) for recommender systems [109] is an approach that modifies the standard GAN training pipeline as shown in Fig. 10b: the generator attempts to approximate the real data distribution for each user, and the discriminator learns a pairwise preference ranking function for pairs of items with a contrastive loss function $L(f_i - f_j)$.

Another interesting adversarial model that deals with ranking-based recommendations is IPGAN (Item Pair GAN) [40]. The problem it is trying to solve is generating not only positive candidates but also candidates for hard negative samples that are needed in contrastive training. Figure 11 shows its basic workflow: IPGAN has two generators, G_p for positive examples and G_n for negative examples, and one discriminator D that tries to distinguish fake item pairs from real.

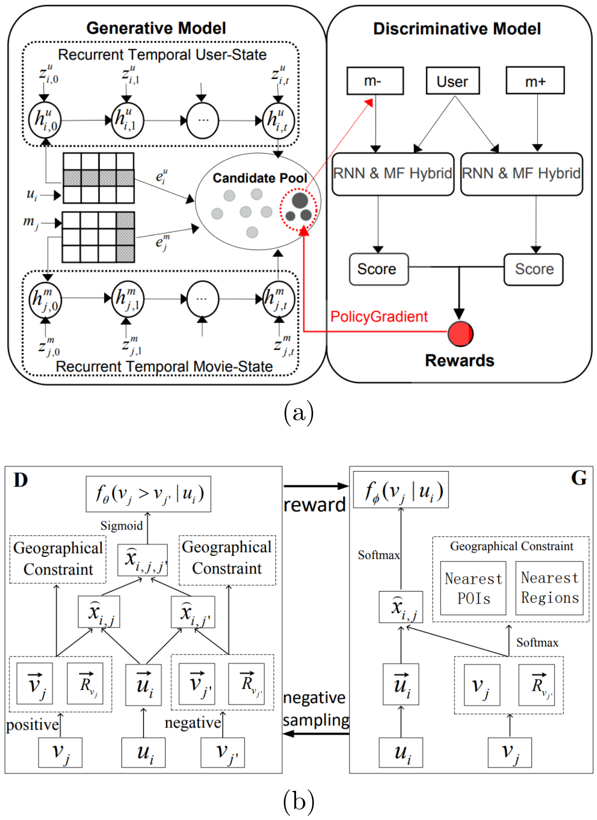


Figure 12. GAN-augmented temporal recommendations: (a) PLASTIC (illustration from [132]); (b) GeoALM (illustration from [72]).

3.4. GAN-Based Generation with Additional Information. We also note several (less directly relevant) works that deal with recommendations with additional information. PLASTIC [132], which stands for Prioritizing Long And Short-Term Information in top-n reCommendation systems, moves to sequential recommendations and joins together matrix factorization and a recurrent neural network as the basic recommenders

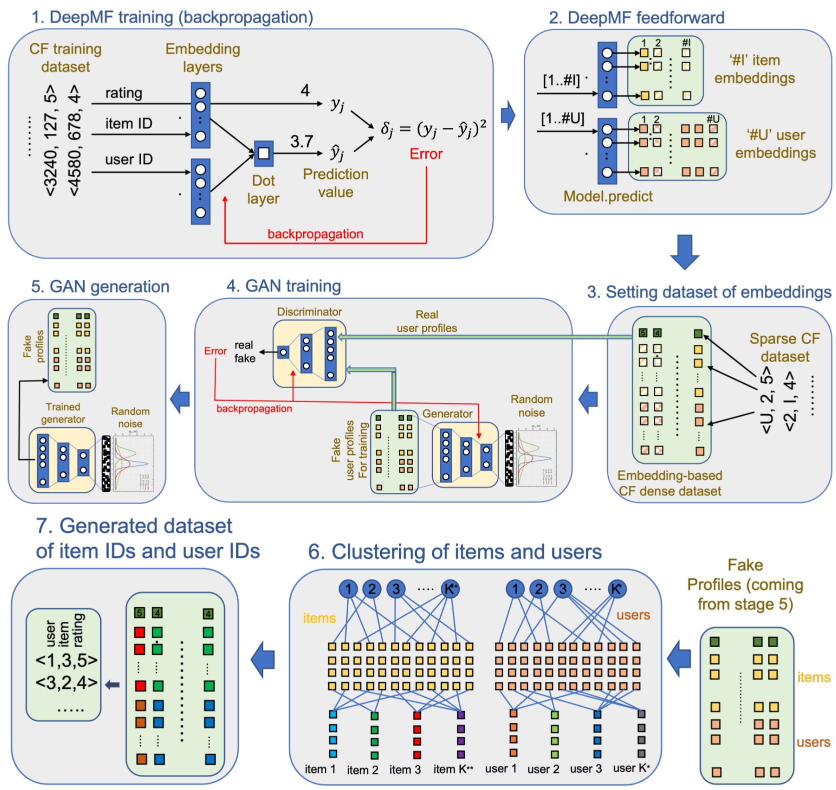


Figure 13. Outline of the training and generation process for GANRS (illustrations and examples from [12]).

via an adversarial mechanism (see Fig. 12a). RecGAN [11] develop a custom GRU-based architecture for both generator and discriminator in order to model the temporal evolution of user preferences. APOIR [133] (which stands for adversarial point-of-interest recommendation) and GeoALM [72] move into the geospatial domain: the generator provides recommendations for points of interest, while the discriminator distinguishes them from real check-in data and provides gradients for the generator's training; Fig. 12b illustrates GeoALM's architecture and training process.

3.5. GANRS. A very recent effort by Bobadilla et al. [12] presents a GAN-based model, called GANRS, which is able to generate entire collaborative filtering datasets with a high degree of control, conditioning by selecting their preferred number of users, items, samples, and stochastic variability. The training process for this model is outlined in Figure 13: (1) a deep matrix factorization model learns user and item embeddings, so that (2) its feedforward process is able to convert sparse one-hot vectors of user and item ids into embeddings, and (3) we can replace the collaborative filtering dataset with a dataset of dense embeddings; (4) then a GAN learns to generate fake user profiles, compared with the discriminator against real samples from the dense dataset, so that (5) the GAN can output good fake dense user representations; (6) the resulting fake profiles are not exactly equal to existing dense vectors, and they are all different, so they are clustered into a predefined number of clusters with k-means; (7) finally, dense samples from the generator are converted to sparse labels corresponding to their cluster ids. Bobadilla et al. report generating high-quality samples based on *MovieLens*, *Netflix*, and *MyAnimeList* real datasets.

§4. Synthetic data generators for state of the art recommender systems

In this section, we consider recently developed synthetic data generators that are designed with an eye towards state of the art neural recommender systems. In many cases, the generator can also serve as an environment for training a reinforcement learning agent, which is a popular framing for modern recommender systems.

4.1. RecSim and RecSim NG. RecSim [50] is a platform developed by *Google Research* for simulation environments for recommender systems that support sequential interaction with users. A general overview of RecSim is shown in Figure 14; the platform consists of:

- a *user model* that samples a user from a prior distribution over user features, including latent features such as personality and tastes, observable features such as demographics, and behavioural features such as the average session length or maximal time budget;
- a *document model* that samples a document from a prior over document features, including latent ones such as the quality and observable features such as the topic or overall popularity;

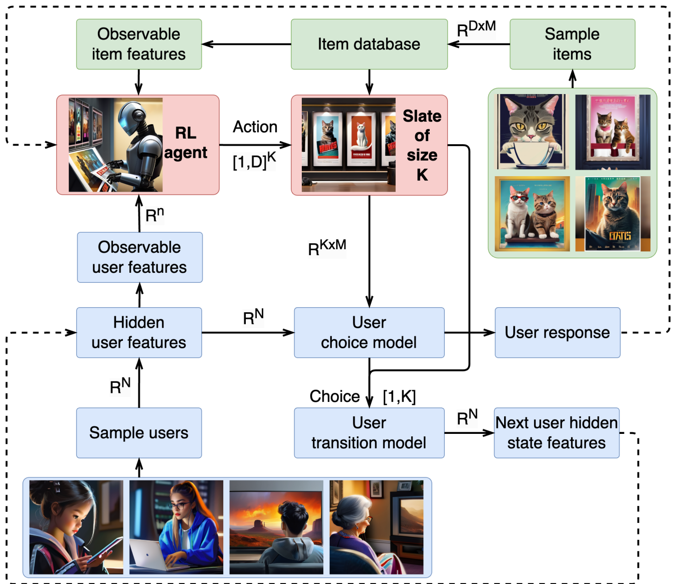


Figure 14. The general overview of RecSim [50]; N is the number of features in the user's hidden state; n, in the user's observed state; M, in the item's hidden state; m, in the item's observed state; D is the total number of items (documents); K is the slate size.

- a *user choice model* that determines the user's response to a document;
- a *user transition model* that configures how the user state changes after a document has been interacted with.

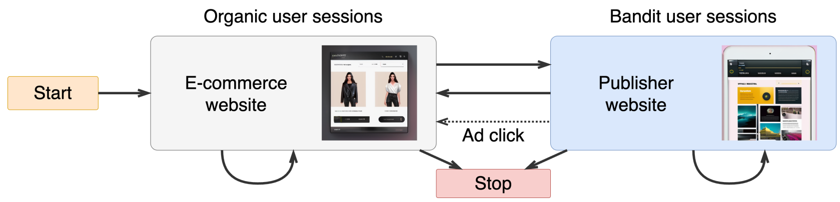
All components of RecSim are fully configurable, and it supports a wide range of possibilities for all four above-mentioned models. RecSim itself does not contain any learnable models that can generate recommendations. Its main goal was to facilitate the development of recommender algorithms that support user states and "non-static" users [44, 46, 125], especially the increasingly popular field of recommender systems based on reinforcement learning [7, 19, 21, 37, 51, 68].

The ideas of RecSim were further developed in RecSim NG, a new version of the platform [82] with an emphasis on collaborative interactive recommendations where a recommender system engages in multi-turn (non-myopic) cooperative exploration with the user. RecSim NG extends the ideas of RecSim by allowing the above-mentioned models to be developed in a probabilistic programming environment, specifically Edward2 [115]. This allows the developer to specify, say, a user model as a dynamic Bayesian network [88], complete with latent variables and complex priors that can be incorporated with MCMC-based and variational approximate Bayesian inference methods.

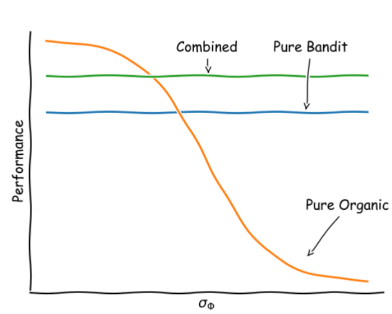
In RecSim NG, the user response function (the user choice model in the terminology above) consists of two parts: affinity model and the choice distribution itself. The affinity model is an arbitrary externally defined function that takes user and item states as input and produces user-item relevance scores. The authors suggest a simple negative Euclidean distance as a basic affinity model and several more complex variations for its further development. The choice model can be greedy (picking the most relevant item) or stochastic; the authors suggest three options for the choice distribution: multinomial logistic (softmax over affinities), cascade, and Plackett-Luce.

4.2. RecoGym. RecoGym [104] is a simulation environment designed specifically with reinforcement learning in mind, as reflected in its name reminiscent of OpenAI Gym [14]; actually, RecoGym itself is also released as an OpenAI Gym environment. Its model of user activity is shown in Fig. 15a: a user alternates between organic sessions and publisher sessions until the session ends; advertising recommendations can be shown during publisher sessions, and the goal of recommendations is to show personalized ads that would incentivize the user to transition back to the e-commerce website.

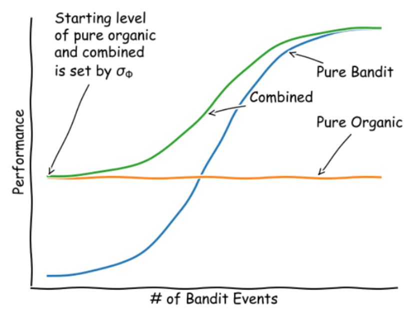
Publisher sessions are modeled as multiarmed bandits, and organic behaviour is modeled with a categorical distribution that could result from a recommender system. The interaction between them is modeled as $\Phi_{u,a,t} = f(\Lambda_{u,p,t} + \epsilon_{u,a,t})$, where $\Phi_{u,a,t}$ is the click-through rate for recommendation a to user u at time t, $\Lambda_{u,p,t}$ is the organic score such that the probability of organically viewing product p at time t for user u is $\sigma(\Lambda_{u,p,t})$, and $\epsilon_{u,a,t}$ is the noise with mean 0 and variance σ_Φ. The system's behaviour depends significantly on the noise variance σ_Φ: Figure 15b shows that as σ_Φ increases the organic performance decreases accordingly,



(a)



(b)



(c)

Figure 15. RecoGym [104]: (a) rough outline of the system; (b) performance as a function of σ_Φ; (c) performance as a function of the number of bandit events (performance illustrations from [104]).

while Figure 15c shows that as the number of bandit events increases the overall performance tends to the pure bandit performance, and organic performance becomes irrelevant. We also note prior work on learning from bandit feedback in recommender systems [58, 59].

RecoGym has served as the basis for the RecoGym Challenge[2] announced at REVEAL 2019; we note the report written by the winning team that outlines both RecoGym operation (in more detail than the original paper) and their winning solution [57].

4.3. Simulated Users for Measuring Recommender System Effects. This interesting work by *Google* researchers Yao et al. [128] presents a user simulation framework with an eye towards measuring the effects of a

[2] `https://sites.google.com/view/recogymchallenge/`

recommender system on user behaviour. E.g., a food recommender system might be promoting unhealthy habits, while the filter bubble effect might lead to narrowing of the user's set of preferred items over time; this leads to a field known as *responsible recommendation* [13, 24, 32, 89].

The proposed simulation framework divides a user's single interaction with the system into two parts:

- the *selection model* defines how a user chooses an item from a slate (presented selection of items, i.e., the current set of impressions), and
- the *feedback model* defines how the user rates the chosen item after interacting with it.

The authors investigate several simple models of user behaviour and study how standard recommender systems (e.g., matrix factorization and large-scale neural recommenders) shape user trajectories under different assumptions regarding user behaviour. They show that recommender systems exhibit nontrivial temporal dynamics even under completely random user behaviour, and the results suggest further work in disentanglement between the effect of user preferences and recommender system design is needed. A similar study has also been undertaken in [43], and we note that prior analysis of recommender systems biases and their effects on user trajectories, as well as ML fairness in general, has also been often done with simulated synthetic users [23, 55, 112].

4.4. SOFA. SOFA [49], which stands for the Simulator for OFfline leArning and evaluation, is designed for reinforcement learning for recommendations (Figs. 16a and 16b). Its basic idea stems from an important limitation that the authors identify in previously developed simulators: ignoring interaction biases present in training user data leads to these biases affecting the simulation and thus negatively affecting recommender models learned via this simulation. Thus, SOFA is explicitly designed to introduce a debiasing step before adding logged data into the rating matrix, as illustrated in Fig. 16c. This debiasing transformation, called the Intermediate Bias Mitigation Step (IBMS), can have different forms, but the authors propose to use the inverse propensity scoring (IPS) approach known in causal inference literature [54, 69]. The authors conclude that IBMS can indeed mitigate interaction bias in logged data to a significant extent, and debiasing appears to be an important step in developing simulators for training RL-based recommender systems.

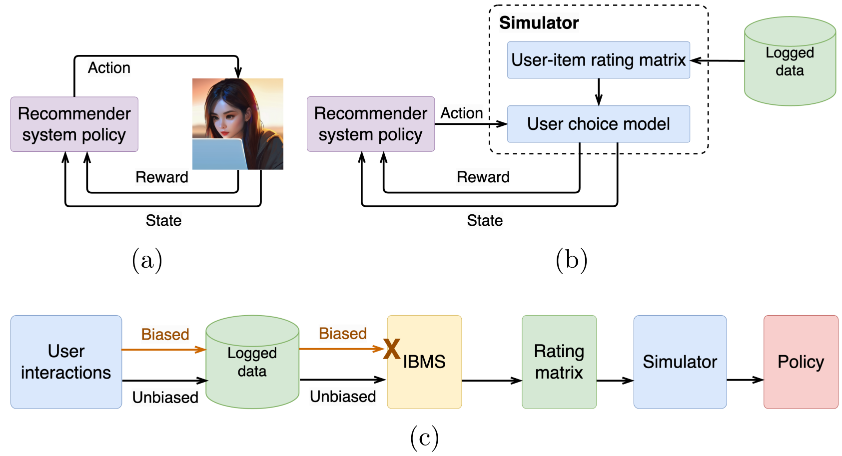


Figure 16. The SOFA simulator [49]: (a) general structure of RL-based recommendations; (b) using a simulator in RL-based recommendations; (c) the IBMS step debiases logged data before it reaches the rating matrix.

4.5. Virtual TaoBao. Virtual Taobao [107] is a simulator designed to imitate product search on Taobao, in particular to serve as an environment for reinforcement learning. The high-level architecture is shown in Fig. 17a; Virtual Taobao consists of:

- GAN-SD (GAN for simulating distributions) that generates synthetic customers together with their initial requests that represent the first user input (see Fig. 17b for an overview of the engine's and customer's view of the interaction); GAN-SD is a regular GAN whose generator loss function is modified with distribution-based constraints;
- MAIL (multi-agent adversarial imitation learning), an approach extending GAIL [47] to the multi-agent setting, that generates interactions between customers and the platform by learning their policies via imitation learning.

The authors report that Virtual Taobao does capture the properties of the real environment faithfully and improves reinforcement learning results. Fig. 17b shows the customer and engine views of the interaction: in the engine view, the state consists of customer features and their request and

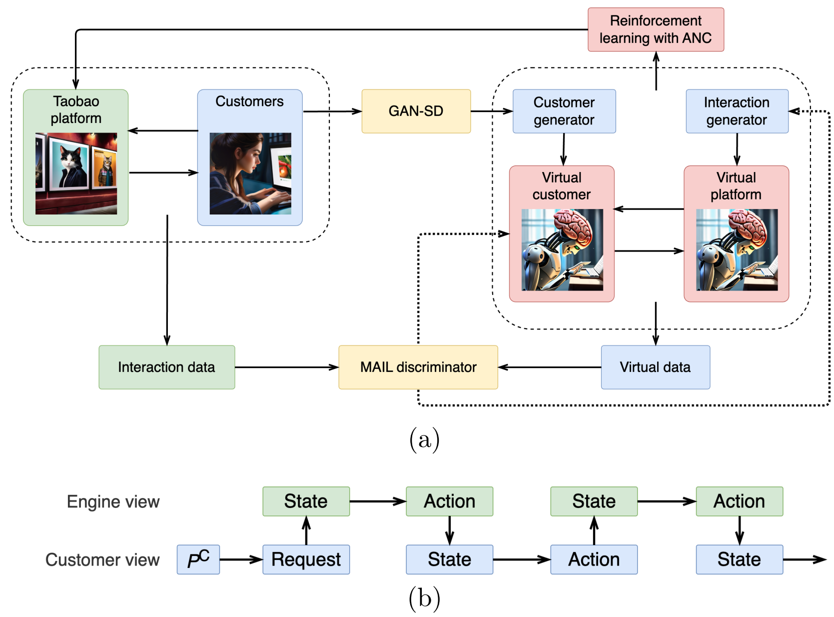


Figure 17. Virtual Taobao [107]: (a) reinforcement learning and data generation; (b) customer and engine views of the interaction.

the action is a recommendation parameterized as a vector in $\mathbb{R}^d$, while in the customer view, the state consists of customer features and engine action, while possible actions include making a purchase, turning to another page, and leaving the system.

4.6. SARDINE. A very recent work by Deffayet et al. [26] presents SARDINE (**S**imulator for **A**utomated **R**ecommendation in **D**ynamic and **IN**teractive **E**nvironments). SARDINE aims to provide a simulator able to answer the following questions:

- how to enable multi-step reasoning and control user-related metrics in the long run, a question especially important for reinforcement learning in recommender systems [20, 27, 33, 126];

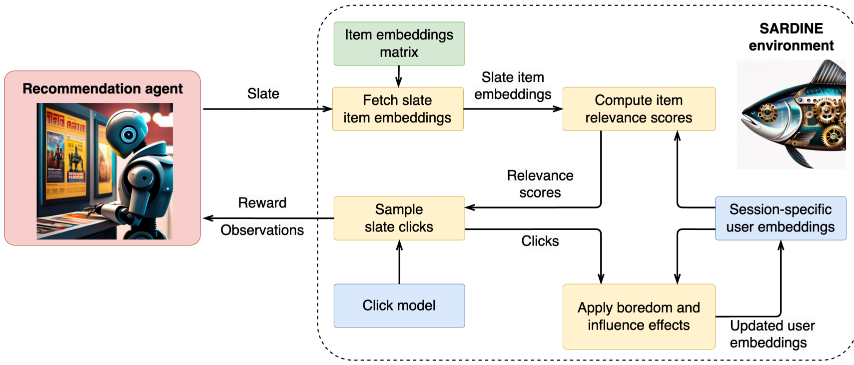


Figure 18. Structure of the SARDINE simulator [26].

- how to learn meaningful and reliable information from biased data, a problem well-known in real-life recommender systems and studied for offline reinforcement learning [27, 126] and other methods [25, 41, 62];
- how to make sure that interactive recommender systems are robust to uncertainties of the real world, specifically external factors influencing user behaviour, item values, and user preferences [63, 91];
- how to effectively and efficiently recommend slates of items to users in a dynamic and interactive environment, a question especially relevant to our problem setting and studied in literature developing slate recommendation policies [52, 90, 106, 111].

The structure of the SARDINE simulator is presented in Fig. 18. It is designed for slate recommendations in a dynamic environment, where a user interacts with a recommender system over L steps. On every step, the recommender system presents a slate of S items, the user may click some of them, and these clicks are returned to the recommender system. The simulator incorporates (see Fig. 18):

- item and user embeddings, defined in this case as randomly generated sparse embeddings over a certain set of topics $\mathcal{T}$;
- initial recommendation done in the simulator independently of the recommender agent;
- relevance computation based on the dot product between the item and user embedding;

- a position-based click model, where the probability of a click is defined as the product of the item's attractiveness (computed from the relevance score) and its examination probability computed from the item's rank in the slate;
- two long-term mechanisms to model the user's evolving preferences: a boredom mechanic, which comes into play if the user is shown too many items with the same topic, and clicked item influence.

Deffayet et al. report experiments on a number of baselines, including reinforcement-based approaches and reranking approaches, and emphasize that the proposed simulator is better able to investigate the research questions listed above [26].

§5. Other approaches

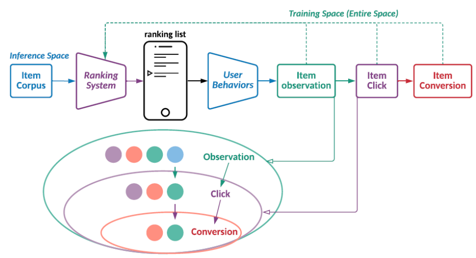
5.1. Multi-Scale User Interactions. An important extension of user feedback generation deals with generating *multi-scale* user behaviour, for instance, sequences such as "impression → click → conversion" which constitute the essence of the sales funnel. Here we note the recent model called HEROES (Hierarchical rEcurrent Ranking On the Entire Space) by Jin et al. [60] who present a simulator for impression-click-conversion pipelines.

Figure 19a shows the pipeline of the system HEROES is modeling: a user sees a ranked list of recommendations, may click some of the items, and then may purchase some of the clicked items (the conversion event). This leads to user behaviour being defined by a combination of several different effects, as exemplified in Fig. 19b. To model user behaviour, HEROES uses two layers: the CTR layer (click-through rate) models the impression to click behaviour, and the CVR layer (conversion rate) models the click to conversion behaviour. Whenever a click occurs in the data, a special gating mechanism allows information to pass to the CVR layer and continue modeling there, as shown in Fig. 19c. The overall architecture of every layer is recurrent, as shown in detail in Fig. 19d.

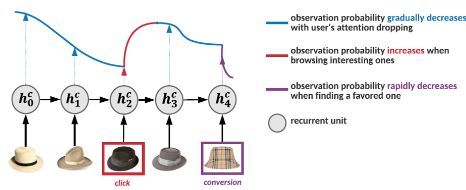
The authors report excellent results in predicting the click and conversion events on three large-scale industrial datasets: Criteo[3] [102], Taobao E-Commerce[4], and Diantao Live Broadcast (proprietary).

[3] https://ailab.criteo.com/ressources/

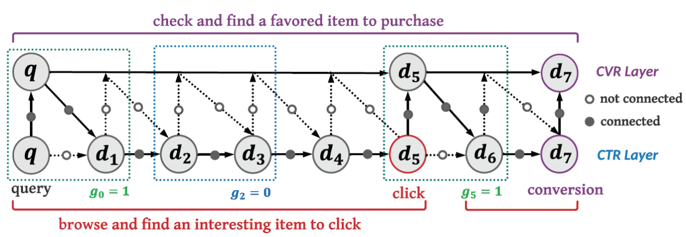
[4] https://tianchi.aliyun.com/datalab/dataSet.html?dataId=408



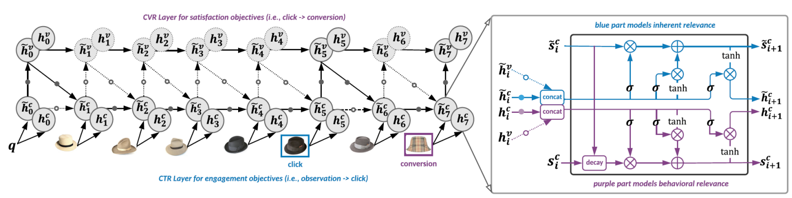
(a)



(b)



(c)



(d)

Figure 19. Multi-scale modeling with HEROES (illustrations from [60]): (a) the system pipeline; (b) sample user analysis; (c) the gating mechanism; (d) layer architecture.

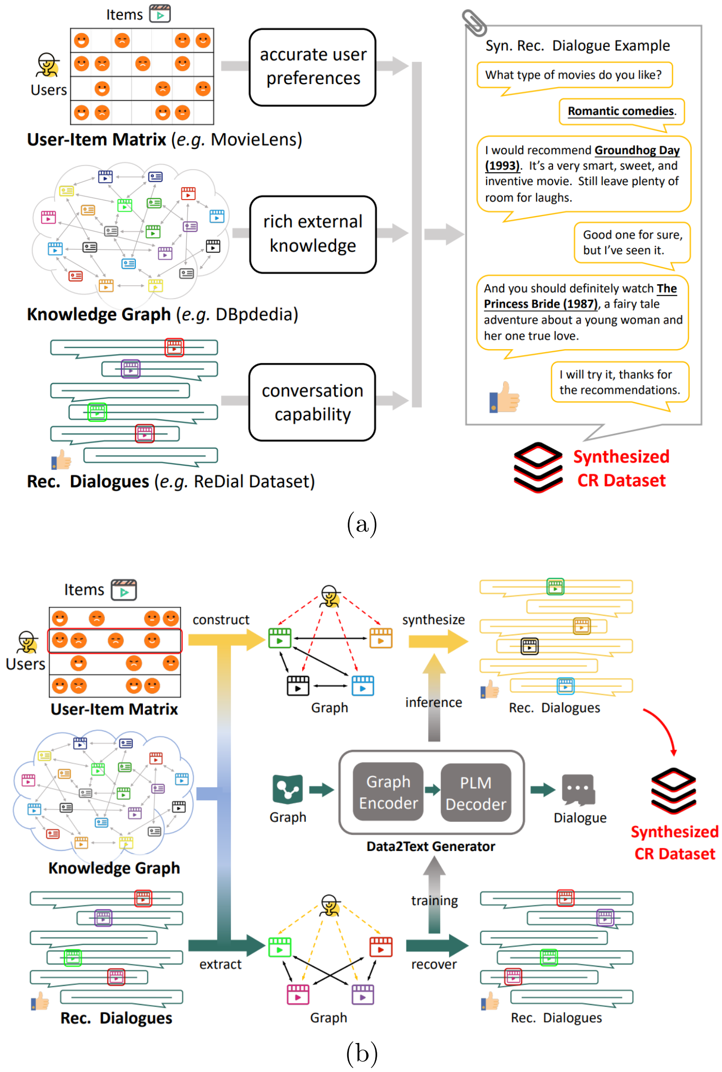
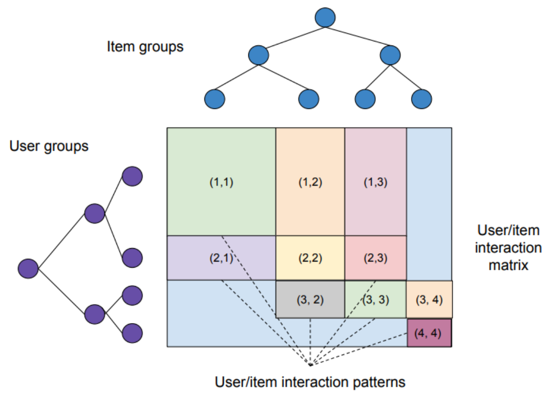


Figure 20. AUGUST (illustrations and examples from [73]): (a) basic overview; (b) detailed generation process.

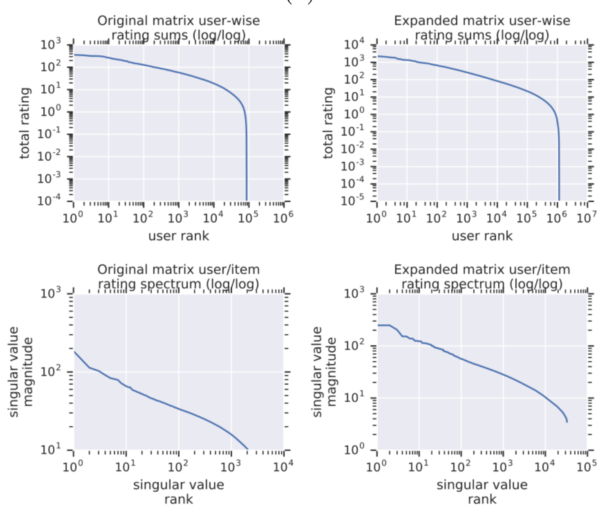
5.2. Generating Additional Information. Some synthetic data generators for recommender systems create entire datasets with additional information such as, e.g., the content of items. For instance, AUGUST [73] (Automatic Generation Understudy for Synthesizing Conversational Recommendation Datasets) aims to generate data for conversational recommendations, i.e., natural language dialogues between (simulated) users and the recommender system, as shown in Fig. 20. In this case, however, the main novelties of this work were in conditional text generation, in particular introducing predefined knowledge into natural language conversations, rather than generating user responses in the collaborative filtering sense of the word.

5.3. Synthetic Recommender Data Generation with Graph Theory. An interesting approach for generating synthetic data was proposed by Belletti et al. [9]. They aim to bridge the gap between relatively small academic datasets and huge real-world recommender systems. To do so, they propose to expand existing user-item interaction matrices with fractal Kronecker expansions. They note that user-item interactions have a hierarchical nature: one can combine users and items into groups and form coarse interaction matrices for these groups, as numerous clustering-based recommender algorithms had done (Figure 21a). Then, they propose to use the Kronecker expansion [64,65] to take this hierarchical structure one step further down, treating original users and items as groups of users and groups of items and expanding the matrix to represent a finer structure while preserving the main statistical properties of the original graph. As a result, they are able to obtain an expanded version of *MovieLens* with 10 billion ratings for 864K items and 2M users (compared to 20M ratings for 27K items and 138K users in the original) while preserving all the main properties such as the distribution of user and item ratings and the spectrum of the user-item interaction matrix (Figure 21b).

5.4. Other works. Among other approaches, we note synthetic data generation for conversational recommender systems [108], synthetic generation of simulated user profiles based on real social network users (Fig. 22) [84], an experimental study of several multi-attribute utility collaborative filtering algorithms based on synthetic data [77], a web-based testing tool ColLaFiS for multi-criterial evaluations of collaborative filtering systems with generated data [75] and further analyses based on it [76], and tools designed to decouple evaluation from development of recommender systems



(a)



(b)

Figure 21. Fractal expansions for user-item interaction matrices (illustrations and plots from [9]): (a) the hierarchical nature of user-item interactions; (b) statistics of individual variables.

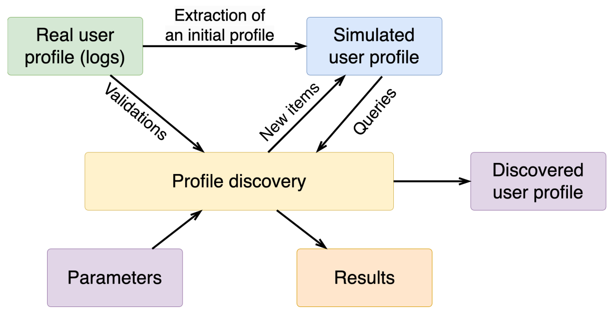


Figure 22. Social user profiles generation [84].

and allow for fair experimental comparisons, not necessarily but perhaps also on synthetic data [10, 85, 86, 105].

§6. Datasets for user response evaluation

Although there are plenty of recommender system datasets with widely varying size and properties, our task imposes additional restrictions on the datasets' composition and available data dimensions. The most important point is that datasets for user response modeling must contain not only the history of user-item interaction, but also the history of recommendations *showed* to a specific user (*impressions*). Unfortunately, standard recommender datasets very seldom contain this information, which makes the search for datasets both harder and more important. It is telling that the authors of *ContentWise Impressions* (Section 6.1.1) in their 2020 paper say: "Impression datasets... can be classified into two categories: private datasets, collected by the authors of the article but... not made accessible to the community, and non-redistributable datasets, made accessible only to the participants of a challenge under a non-redistribute clause... To the best of our knowledge, no open-source dataset with impressions exists." Still, some datasets with impressions have begun to appear, and in Section 6.1 we review what is currently available in the field. Section 6.2 describes e-commerce datasets that can be adapted for user response modeling.

6.1. Datasets with Logged Impressions.

6.1.1. *ContentWise Impressions.* The *ContentWise Impressions* dataset, presented in [79], contains logged recommendations and user interactions with them. It was collected from a video streaming service with four mutually exclusive categories of items: movies, movies and clips in series, TV movies or shows, and episodes of TV series. Therefore, most items in *ContentWise Impressions* are grouped into *series*, and each item is associated with the series ID and position in the series. Figure 23 illustrates the user screen layout in the streaming service: rows contain generated recommendations (impressions), and more relevant items are positioned in top rows and closer to the left.

The dataset consists of three files (tables):

(1) *interactions.csv* — user-recommendation interaction data: timestamp, user, item, series and recommendation ids, episode number (item position in the series), interaction type, vision factor and explicit rating;
(2) *impressions-direct-link.csv* — recommendations data: recommendation id, row position (of the entire recommendation), recommendation list length, recommended series list;
(3) *impressions-non-direct-link.csv* — user-item interaction data that does not log the respective set of recommended items; this part of the data is not suitable for user response modeling, although it can be used in other parts of the simulator pipeline, e.g. for user profile modeling.

Joining the first two tables via "*recommendation id*" yields a dataset with user response known for each pool of recommended items, exactly as needed for user response modeling.

6.1.2. *RL4RS.* The RL4RS dataset (Reinforcement Learning for Recommender Systems) [121] is another relatively new data source with logged recommended items, released by *NetEase Games*. The data comes from a mobile game which means that impressions are shown with special unlock rules illustrated in Fig. 24: in the RL4RS-Slate dataset (Fig. 24a), new bundles of recommended items on a single page are shown after an unlock condition is fulfilled (the current list is sold out), while the RL4RS-SeqSlate dataset (Fig. 24b) adds the relationship between different pages and asks to maximize the total reward of a session that spans multiple pages of recommendations.

Compared to *ContentWise*, this dataset contains very rich user and item information: an anonymized user portrait (profile), user click history embeddings, item features that include price, category, and more, and

Figure 23. Structure of a *ContentWise Impressions* slate [79].

other features. An important domain-specific feature of this dataset is that it contains information about only 283 items along with 140K users and a large number of item features: 40 values in an item embedding along with values directly associated with the price of every item, its location, and an indicator whether the item is “special”.

6.1.3. *TenRec.* *Tenrec* [130] is a dataset suite developed for multiple recommendation tasks, collected from *Tencent*’s two different recommendation platforms for feeds (anonymized in the paper and called below QB and QK). The dataset consists of two parts (articles and video), both containing user and item ids with various types of user feedback, which makes it an excellent candidate for studying user responses.

Tenrec is a large dataset with four different recommendation scenarios: QB-article, QB-video, QK-article, and QK-video; some users and items

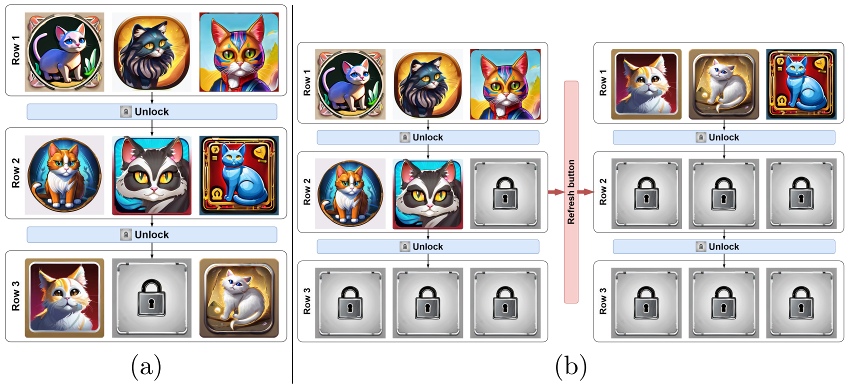


Figure 24. RL4RS [121]: (a) RL4RS-Slate; (b) RS-SeqSlate.

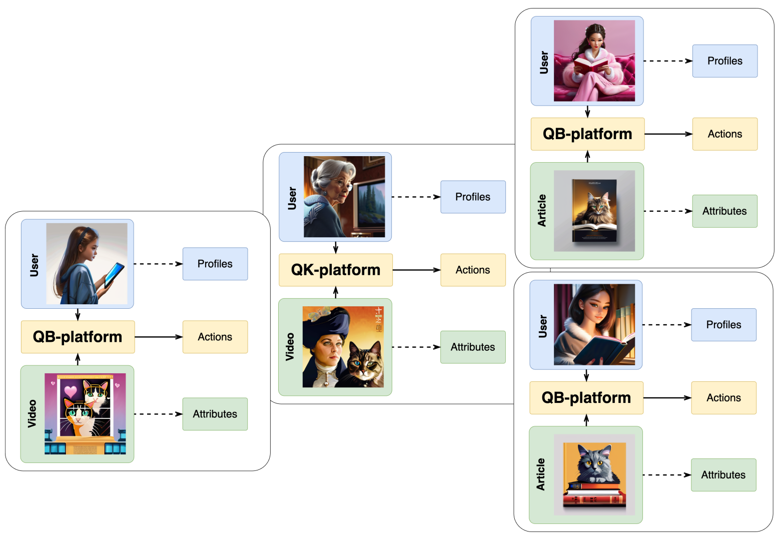


Figure 25. Overlaps between users and items for the four settings presented in *Tenrec* [130].

overlap across the two platforms, as illustrated in Figure 25. The QB part is relatively small—34K users and 130K items in QB-video, 25K users and 7.3K items in QB-article—but the QK parts are quite large: there are over

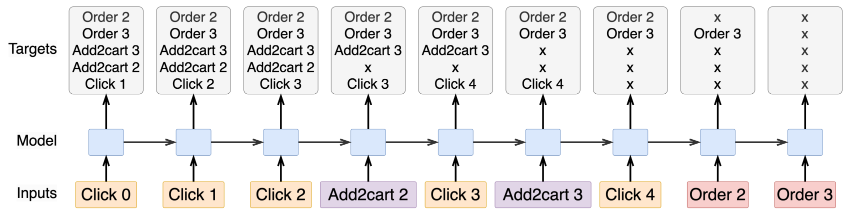


Figure 26. A sample user session in OTTO [98].

5M users and 3.7M items in QK-video with over 142M clicks and 1.3M users and 220K items with 46M clicks in QK-article. Moreover, it contains quite lengthy user interaction sessions, with over 2300 sessions of length over 300 [130].

This dataset is especially well-suited for our needs. First, unlike most other collaborative filtering datasets, *Tenrec* does track impressions for videos (called "exposures" in [130]), with just under 500M impressions in the QK-video part alone.

Moreover, similar to the situation with the sales funnel in e-commerce, *Tenrec* contains additional types of interactions that can be viewed as a user response, namely likes, shares, followings, favorites, and reads in case of articles. In these parts of the dataset, we can use clicked items as a "recommendation pool" and these additional interactions (which signify special interest) in place of user responses.

6.2. E-Commerce Datasets. Since datasets with logged impressions are few and far between, we suggest using e-commerce datasets that usually contain a sales funnel: a user views some pool of items, then clicks on some subset of them, and finally purchases some subset of clicked items. Formally speaking, such datasets usually contain events of several different types. While impressions *per se* are not usually recorded in these datasets, the presence of several levels of the sales funnel allows to model user responses on the next steps: for instance, clicks can be viewed as "impressions", and adding items to the cart or purchasing decisions can be viewed as the user response. Unfortunately, most such datasets lack user and item features (usually due to privacy concerns) and are often appropriate only for sequence-based approaches.

6.2.1. *OTTO.* OTTO [98] is a new dataset released in 2022 by the OTTO online store and app. The dataset contains sessions that reflect user interaction with the online store and include information on their clicks, adding items to the shopping cart, and orders. Figure 26 illustrates the structure of an OTTO user session: for every moment in time (timestamp), the target for the “Click” action is only the next item clicked in this session, while the targets for “Add to cart” and “Order” actions always includes all such actions still remaining in the current session.

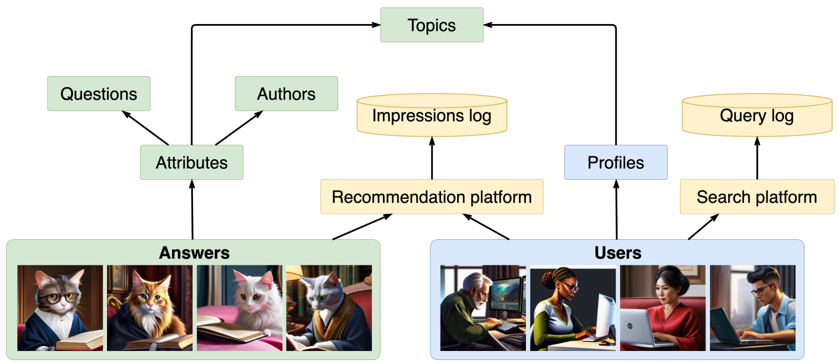
However, this dataset does not contain user features or even user ids, so one cannot cross-reference different sessions of the same user, and the dataset can be utilized only in the setting of sequential recommender systems.

6.2.2. *RetailRocket.* The *RetailRocket* dataset [5] also contains information about user behaviour in a real-world e-commerce system with several types of events—clicks, add to carts, and transactions—collected over 4.5 months. Unfortunately, similar to OTTO, *RetailRocket* does not contain either any user features or persistent user ids. However, it does contain a rich set of item features:

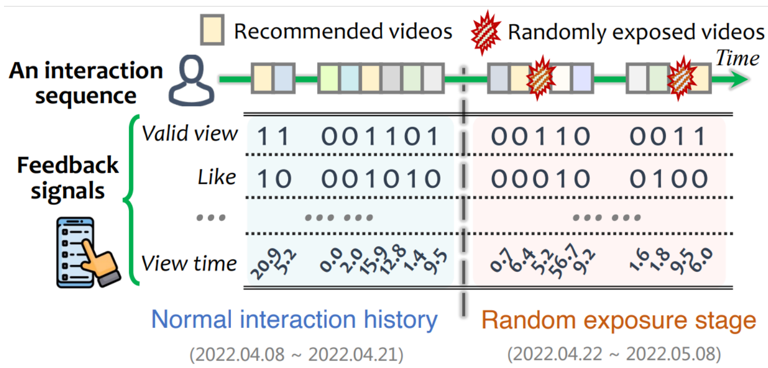
- item categories are represented with a detailed category tree;
- other features such as the price vary with time, which opens up other possibilities for recommendation analysis.

6.2.3. *REES OpenCDP.* The OpenCDP REES collection of datasets [1–4] contains user behavior data for four different online stores: multi-category, electronics, cosmetics, and jewelry. The jewelry store data does not contain event types and hence is irrelevant. The other three datasets have the same structure but different sizes. These datasets contain no user features and a relatively poor set of item features (category, brand, and price), but they do contain persistent user ids, which makes it possible to cross-reference users across sessions.

6.3. Non-Recommender Datasets with Similar Structure. Other fields of machine learning, including information retrieval systems and question answering systems, also feature datasets with similar structure. Compared with e-commerce, they are less well suited for recommender systems, but can be suitable in some cases.



(a)



(b)

Figure 27. Datasets from large online article/video recommendation systems: (a) *ZhihuRec* [42]; (b) *KuaiRand* (illustration from [34]).

6.3.1. *ZhihuRec*. *ZhihuRec* [42] is a relatively large dataset collected on the Q&A platform *Zhihu*; it is composed of about 100M interactions of nearly 800K users, 165K questions, and 554K answers on 70K topics from 240K authors, with over 500K user query keywords recorded.

The dataset consists of two parts: user search query log and impressions from the platform's recommendation system (see Fig. 27a for an illustration). The search query log dataset seems to be suitable only for information retrieval systems, but the impressions dataset may be applicable in recommender system research, including user response modeling. It contains a user's views of answers to various questions on *Zhihu* and responses for some of them. The dataset also contains user and answer features along with persistent user ids. Note that answers are not shown to the users by some recommendation policy but rather as a result of user search queries that are logged as well.

6.3.2. *Yandex Personalized Web Search Challenge.* This dataset, published by the *Yandex* search engine [6], contains information of user search queries with top 10 found URLs and user clicks on them. No user or item (URL) features are provided. Moreover, terms in the queries are anonymized as well, which makes it impossible to obtain informative features through, e.g., embeddings of the queries.

6.3.3. *JDSearch.* The *JDSearch* dataset [70] is at the junction of the fields of information retrieval and e-commerce. It contains search queries with found items and user interactions with such items along with some anonymized item features. It can be used in the same settings as previous ones.

6.4. Less relevant datasets. The following datasets may be in some contexts considered as data sources for user response modeling, but in our opinion are less suited for this task.

6.4.1. *MTS Kion. MTS Kion* [97] is a streaming service dataset with implicit feedback (percentage of watched time) that could be potentially used as user response. While having a small average user history (5), it contains user features such as age and income that are even not obfuscated (very rare for real world recommender datasets!). Item features are also provided, including common metadata for movies such as title, starring actors, age rating, and so on. The dataset is relatively large: it contains 5.4M interactions between 962K users and 15K items. However, is does not contain anything that could be treated as a "recommendation pool", so the only way to use it would be to use watched videos as an "impression" and watching a video to a high percentage as positive feedback.

	Users	Items	Interactions	Impressions	Slates
ContentWise [79]	42.1K	145K	10.5M	23.3M	✓
RL4RS [121]	150K	283	9M	16M	✓
Tenrec QB-Video [130]	34K	130K	1.7M	2.4M	—
Tenrec QB-Article [130]	25K	7.3K	348K	—	—
Tenrec QK-Video [130]	5M	3.7M	142M	493.5M	—
Tenrec QK-Article [130]	1.3M	220K	46M	—	—
OTTO [98]*	12M	1.8M	220M	—	—
RetailRocket [5]	3.4K	8.9K	14K	92.5K	—

Table 1. Dataset summary statistics; * OTTO has no user ids, so we show the number of sessions.

6.4.2. *Amazon M2.* *Amazon M2* [61] is a dataset for next-item recommendation systems with textual features of products. It contains 4M *sessions*, that is, sequences of items a user has interacted with. Unfortunately, it does not contain anything resembling the sales funnel, but may be useful for the COLES approach.

6.4.3. *Kuaishou.* The *Kuaishou* app has given rize to three different published datasets [34,110]. Two of them contain rich user and item features together with a wide variety of interaction types. Although the datasets contain different types of feedback, so the sales funnel is present, we cannot use this datasets directly similar to other e-commerce datasets since its data is not organized into sessions. In particular, the *KuaiRand* dataset contains user-item interactions with both recommended and randomly shown videos, which may be beneficial for user response modeling (Fig. 27b).

Unlike the first two *Kuaishou* datasets, the third one does not contain either different types of interactions or sequential interactions, and is in essence a simple collaborative filtering dataset. The feature that distinguishes this dataset from the others is its density, which is over 0.99, which allows to explicitly evaluate recommender systems in an offline setup. User and item features are also provided.

6.5. Summary. Table 1 shows a summary table of available datasets. We believe that the two most convenient datasets that provide slates and impressions and might also be suitable for subsequent experiments with offline reinforcement learning are *ContentWise* and RL4RS.

§7. Conclusion

In this survey, we have considered a wide variety of user response modeling models, simulators, and datasets. In summary, we note several basic approaches that modern user response modeling mainly falls into:

- probabilistic models (Section 2) try to capture the interactions of latent variables characterizing users, items, and recommendation contexts in the form of probability distributions, usually as a graphical probabilistic model;
- adversarial approaches (Section 3) generate synthetic user feedback with adversarial models that comprise a generator and a discriminator trying to distinguish synthetic feedback produced by the generator from real data;
- simulators (Section 4) can use probabilistic, neural, and/or adversarial methods, but their main characteristic feature is that they usually provide environments designed to train recommender systems with reinforcement learning, a growing and important subfield of recommender systems today.

We have also noted the most important datasets available to be used for user response modeling (Section 6, see in particular the summary table in Section 6.5), highlighting the most suitable ones for further model development.

References

1. *eCommerce behavior data from multi category store*, `https://www.kaggle.com/datasets/mkechinov/ecommerce-behavior-data-from-multi-category-store`.
2. *eCommerce events history in cosmetics shop*, `https://www.kaggle.com/datasets/mkechinov/ecommerce-events-history-in-cosmetics-shop`.
3. *eCommerce events history in electronics store*, `https://www.kaggle.com/datasets/mkechinov/ecommerce-events-history-in-electronics-store`.
4. *eCommerce purchase history from jewelry store*, `https://www.kaggle.com/datasets/mkechinov/ecommerce-purchase-history-from-jewelry-store`.
5. *RetailRocket recommender system dataset*, `https://www.kaggle.com/datasets/retailrocket/ecommerce-dataset`.
6. *Personalized web search challenge*, `https://kaggle.com/competitions/yandex-personalized-web-search-challenge`, 2013.
7. M. M. Afsar, T. Crump, and B. H. Far, *Reinforcement learning based recommender systems: A survey*, CoRR **abs/2101.06286** (2021).
8. R. Agrawal, R. Srikant, et al., *Fast algorithms for mining association rules*, Proc. 20th int. conf. very large data bases, VLDB, vol. 1215, Santiago, Chile, 1994, pp. 487–499.

9. F. Belletti, K. Lakshmanan, W. Krichene, Y. Chen, and J. R. Anderson, *Scalable realistic recommendation datasets through fractal expansions*, CoRR **abs/1901.08910** (2019).
10. A. Bellog'in and A. Said, *Improving accountability in recommender systems research through reproducibility*, User Modeling and User-Adapted Interaction **31** (2021), 941 – 977.
11. H. Bharadhwaj, H. Park, and B. Y. Lim, *Recgan: Recurrent generative adversarial networks for recommendation systems*, Proceedings of the 12th ACM Conference on Recommender Systems (New York, NY, USA), ACM, 2018, pp. 372–376.
12. J. Bobadilla, A. Gutiérrez, R. Yera, and L. Martínez, *Creating synthetic datasets for collaborative filtering recommender systems using generative adversarial networks*, arXiv preprint arXiv:2303.01297 (2023).
13. D. Bountouridis, J. Harambam, M. Makhortykh, M. Marrero, N. Tintarev, and C. Hauff, *SIREN: A simulation framework for understanding the effects of recommender systems in online news environments*, Proceedings of the Conference on Fairness, Accountability, and Transparency (New York, NY, USA), Association for Computing Machinery, 2019, pp. 150–159.
14. G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, *OpenAI Gym*, arXiv preprint arXiv:1606.01540 (2016).
15. X. Cai, J. Han, and L. Yang, *Generative adversarial network based heterogeneous bibliographic network representation for personalized citation recommendation*, Proceedings of the AAAI Conference on Artificial Intelligence **32** (2018), no. 1.
16. D.-K. Chae, J.-S. Kang, S.-W. Kim, and J. Choi, *Rating augmentation with generative adversarial networks towards accurate collaborative filtering*, The World Wide Web Conference (New York, NY, USA), Association for Computing Machinery, 2019, pp. 2616–2622.
17. D.-K. Chae, J.-S. Kang, S.-W. Kim, and J.-T. Lee, *CFGAN: A generic collaborative filtering framework based on generative adversarial networks*, Proceedings of the 27th ACM International Conference on Information and Knowledge Management (New York, NY, USA), Association for Computing Machinery, 2018, pp. 137–146.
18. H. Chen and J. Li, *Adversarial tensor factorization for context-aware recommendation*, Proceedings of the 13th ACM Conference on Recommender Systems (New York, NY, USA), Association for Computing Machinery, 2019, pp. 363–367.
19. M. Chen, A. Beutel, P. Covington, S. Jain, F. Belletti, and E. H. Chi, *Top-k off-policy correction for a REINFORCE recommender system*, CoRR **abs/1812.02353** (2018).
20. M. Chen, A. Beutel, P. Covington, S. Jain, F. Belletti, and E. H. Chi, *Top-k off-policy correction for a reinforce recommender system*, Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (New York, NY, USA), Association for Computing Machinery, 2019, pp. 456–464.
21. X. Chen, L. Yao, J. McAuley, G. Zhou, and X. Wang, *Deep reinforcement learning in recommender systems: A survey and new perspectives*, Knowledge-Based Systems **264** (2023), 110335.
22. E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart, and J. Sun, *Generating multi-label discrete patient records using generative adversarial networks*, Proceedings of

the 2nd Machine Learning for Healthcare Conference (F. Doshi-Velez, J. Fackler, D. Kale, R. Ranganath, B. Wallace, and J. Wiens, eds.), Proceedings of Machine Learning Research, vol. 68, PMLR, 18–19 Aug 2017, pp. 286–305.

23. A. D'Amour, H. Srinivasan, J. Atwood, P. Baljekar, D. Sculley, and Y. Halpern, *Fairness is not static: Deeper understanding of long term fairness via simulation studies*, Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (New York, NY, USA), Association for Computing Machinery, 2020, pp. 525–534.
24. S. Dean, S. Rich, and B. Recht, *Recommendations and user agency: The reachability of collaboratively-filtered information*, Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (New York, NY, USA), Association for Computing Machinery, 2020, pp. 436–445.
25. R. Deffayet, P. Hager, J.-M. Renders, and M. de Rijke, *An offline metric for the debiasedness of click models*, Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (New York, NY, USA), Association for Computing Machinery, 2023, pp. 558–568.
26. R. Deffayet, T. Thonet, D. Hwang, V. Lehoux, J.-M. Renders, and M. de Rijke, *SARDINE: A simulator for automated recommendation in dynamic and interactive environments*, 2023.
27. R. Deffayet, T. Thonet, J.-M. Renders, and M. de Rijke, *Generative slate recommendation with reinforcement learning*, Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (New York, NY, USA), Association for Computing Machinery, 2023, pp. 580–588.
28. M. del Carmen Rodríguez-Hernández, S. Ilarri, R. Hermoso, and R. T. Lado, *Towards trajectory-based recommendations in museums: Evaluation of strategies using mixed synthetic and real data*, EUSPN/ICTH, 2017.
29. M. del Carmen Rodríguez-Hernández, S. Ilarri, R. Trillo, and R. Hermoso, *Context-aware recommendations using mobile p2p*, Proceedings of the 15th International Conference on Advances in Mobile Computing & Multimedia (New York, NY, USA), Association for Computing Machinery, 2017, pp. 82–91.
30. M. Deshpande and G. Karypis, *Item-based top-n recommendation algorithms*, ACM Transactions on Information Systems (TOIS) **22** (2004), no. 1, 143–177.
31. W. Fan, T. Derr, Y. Ma, J. Wang, J. Tang, and Q. Li, *Deep adversarial social recommendation*, Proceedings of the 28th International Joint Conference on Artificial Intelligence, AAAI Press, 2019, pp. 1351–1357.
32. D. Fleder and K. Hosanagar, *Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity*, Management Science **55** (2007).
33. C. Gao, K. Huang, J. Chen, Y. Zhang, B. Li, P. Jiang, S. Wang, Z. Zhang, and X. He, *Alleviating Matthew effect of offline reinforcement learning in interactive recommendation*, Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (New York, NY, USA), Association for Computing Machinery, 2023, pp. 238–248.
34. C. Gao, S. Li, Y. Zhang, J. Chen, B. Li, W. Lei, P. Jiang, and X. He, *KuaiRand: An unbiased sequential recommendation dataset with randomly exposed videos*, Proceedings of the 31st ACM International Conference on Information & Knowledge Management, ACM, oct 2022.

35. M. Gao, J. Zhang, J. Yu, J. Li, J. Wen, and Q. Xiong, *Recommender systems based on generative adversarial networks: A problem-driven perspective*, Information Sciences **546** (2021), 1166–1185.
36. N. Gao, H. Xue, W. Shao, S. Zhao, K. K. Qin, A. Prabowo, M. S. Rahaman, and F. D. Salim, *Generative adversarial networks for spatio-temporal data: A survey*, ACM Trans. Intell. Syst. Technol. **13** (2022), no. 2.
37. J. Gauci, E. Conti, Y. Liang, K. Virochsiri, Z. Chen, Y. He, Z. Kaden, V. Narayanan, and X. Ye, *Horizon: Facebook's open source applied reinforcement learning platform*, arXiv preprint arXiv:1811.00260 (2018).
38. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative Adversarial Networks*, ArXiv e-prints (2014).
39. I. J. Goodfellow, *NIPS 2016 tutorial: Generative adversarial networks*, CoRR **abs/1701.00160** (2017).
40. G. Guo, H. Zhou, B. Chen, Z. Liu, X. Xu, X. Chen, Z. Dong, and X. He, *IPGAN: Generating informative item pairs by adversarial sampling*, IEEE Transactions on Neural Networks and Learning Systems **33** (2022), no. 2, 694–706.
41. S. Gupta, P. Hager, J. Huang, A. Vardasbi, and H. Oosterhuis, *Recent advances in the foundations and applications of unbiased learning to rank*, Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (New York, NY, USA), Association for Computing Machinery, 2023, pp. 3440–3443.
42. B. Hao, M. Zhang, W. Ma, S. Shi, X. Yu, H. Shan, Y. Liu, and S. Ma, *A large-scale rich context query and recommendation dataset in online knowledge-sharing*, 2021.
43. N. Hazrati and F. Ricci, *Simulating users' interactions with recommender systems*, Adjunct Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization (New York, NY, USA), Association for Computing Machinery, 2022, pp. 95–98.
44. R. He and J. McAuley, *Fusing similarity models with Markov chains for sparse sequential recommendation*, 2016 IEEE 16th International Conference on Data Mining (ICDM) (2016), 191–200.
45. X. He, Z. He, X. Du, and T.-S. Chua, *Adversarial personalized ranking for recommendation*, The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (New York, NY, USA), Association for Computing Machinery, 2018, pp. 355–364.
46. B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, *Session-based recommendations with recurrent neural networks*, 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings (Y. Bengio and Y. LeCun, eds.), 2016.
47. J. Ho and S. Ermon, *Generative adversarial imitation learning*, Advances in Neural Information Processing Systems (D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, eds.), vol. 29, Curran Associates, Inc., 2016.
48. K. Houkjær, K. Torp, and R. Wind, *Simple and realistic data generation*, Very Large Data Bases Conference, 2006.

49. J. Huang, H. Oosterhuis, M. De Rijke, and H. Van Hoof, *Keeping dataset biases out of the simulation: A debiased simulator for reinforcement learning based recommender systems*, Proceedings of the 14th ACM Conference on Recommender Systems, 2020, pp. 190–199.
50. E. Ie, C. Hsu, M. Mladenov, V. Jain, S. Narvekar, J. Wang, R. Wu, and C. Boutilier, *RecSim: A configurable simulation platform for recommender systems*, CoRR **abs/1909.04847** (2019).
51. E. Ie, V. Jain, J. Wang, S. Narvekar, R. Agarwal, R. Wu, H.-T. Cheng, T. Chandra, and C. Boutilier, *SlateQ: A tractable decomposition for reinforcement learning with recommendation sets*, Proceedings of the 28th International Joint Conference on Artificial Intelligence, AAAI Press, 2019, pp. 2592–2599.
52. ———, *Slateq: A tractable decomposition for reinforcement learning with recommendation sets*, Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, International Joint Conferences on Artificial Intelligence Organization, 7 2019, pp. 2592–2599.
53. S. Ilarri, R. Hermoso, R. Trillo-Lado, and M. del Carmen Rodríguez-Hernández, *A review of the role of sensors in mobile context-aware recommendation systems*, International Journal of Distributed Sensor Networks **11** (2015), no. 11, 489264.
54. G. W. Imbens and D. B. Rubin, *Causal inference in statistics, social, and biomedical sciences*, Cambridge University Press, 2015.
55. D. Jannach, L. Lerche, I. Kamehkhosh, and M. Jugovac, *What recommenders recommend: An analysis of recommendation biases and possible countermeasures*, User Modeling and User-Adapted Interaction **25** (2015), no. 5, 427–491.
56. D. Jeske, P. Lin, C. Rendon, R. Xiao, and B. Samadi, *Synthetic data generation capabilties for testing data mining tools*, 10 2006, pp. 1–6.
57. O. Jeunen and B. Goethals, *Lessons learned from winning the RecoGym challenge*, `https://olivierjeunen.github.io/recogym/`, 2021.
58. O. Jeunen, D. Mykhaylov, D. Rohde, F. Vasile, A. Gilotte, and M. Bompaire, *Learning from bandit feedback: An overview of the state-of-the-art*, CoRR **abs/1909.08471** (2019).
59. O. Jeunen, D. Rohde, and F. Vasile, *On the value of bandit feedback for offline recommender system evaluation*, CoRR **abs/1907.12384** (2019).
60. J. Jin, X. Chen, W. Zhang, Y. Chen, Z. Jiang, Z. Zhu, Z. Su, and Y. Yu, *Multi-scale user behavior network for entire space multi-task learning*, Proceedings of the 31st ACM International Conference on Information & Knowledge Management (New York, NY, USA), Association for Computing Machinery, 2022, pp. 874–883.
61. W. Jin, H. Mao, Z. Li, H. Jiang, C. Luo, H. Wen, H. Han, H. Lu, Z. Wang, R. Li, Z. Li, M. X. Cheng, R. Goutam, H. Zhang, K. Subbian, S. Wang, Y. Sun, J. Tang, B. Yin, and X. Tang, *Amazon-M2: A multilingual multi-locale shopping session dataset for recommendation and text generation*, 2023.
62. T. Joachims, A. Swaminathan, and T. Schnabel, *Unbiased learning-to-rank with biased feedback*, Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (New York, NY, USA), Association for Computing Machinery, 2017, pp. 781–789.

63. N. Knyazev and H. Oosterhuis, *A lightweight method for modeling confidence in recommendations with learned beta distributions*, Proceedings of the 17th ACM Conference on Recommender Systems, ACM, 2023.
64. J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, *Kronecker graphs: An approach to modeling networks*, Journal of Machine Learning Research **11** (2010), no. 33, 985–1042.
65. J. Leskovec, D. Chakrabarti, J. Kleinberg, and C. Faloutsos, *Realistic, mathematically tractable graph generation and evolution, using kronecker multiplication*, Knowledge Discovery in Databases: PKDD 2005 (Berlin, Heidelberg) (A. M. Jorge, L. Torgo, P. Brazdil, R. Camacho, and J. Gama, eds.), Springer Berlin Heidelberg, 2005, pp. 133–145.
66. R. Li, X. Wu, and W. Wang, *Adversarial learning to compare: Self-attentive prospective customer recommendation in location based social networks*, Proceedings of the 13th International Conference on Web Search and Data Mining (New York, NY, USA), Association for Computing Machinery, 2020, pp. 349–357.
67. P. Lin, B. Samadi, A. Cipolone, D. Jeske, S. Cox, C. Rendon, D. Holt, and R. Xiao, *Development of a synthetic data set generator for building and testing information discovery systems*, Third International Conference on Information Technology: New Generations (ITNG'06), 2006, pp. 707–712.
68. Y. Lin, Y. Liu, F. Lin, P. Wu, W. Zeng, and C. Miao, *A survey on reinforcement learning for recommender systems*, CoRR **abs/2109.10665** (2021).
69. R. J. A. Little and D. B. Rubin, *Statistical analysis with missing data*, John Wiley & Sons, 2019.
70. J. Liu, Z. Dou, G. Tang, and S. Xu, *JDSearch: A personalized product search dataset with real queries and full interactions*, 2023.
71. J. Liu, W. Pan, and Z. Ming, *CofiGAN: Collaborative filtering by generative and discriminative training for one-class recommendation*, Knowledge-Based Systems **191** (2020), 105255.
72. W. Liu, Z.-J. Wang, B. Yao, and J. Yin, *Geo-ALM: PoI recommendation by fusing geographical information and adversarial learning mechanism*, 08 2019, pp. 1807–1813.
73. Y. Lu, J. Bao, Z. Ma, X. Han, Y. Wu, S. Cui, and X. He, *AUGUST: an automatic generation understudy for synthesizing conversational recommendation datasets*, Findings of the Association for Computational Linguistics: ACL 2023 (Toronto, Canada), Association for Computational Linguistics, July 2023, pp. 10538–10549.
74. Y. Lyu, S. Dai, P. Wu, Q. Dai, Y. Deng, W. Hu, Z. Dong, J. Xu, S. Zhu, and X.-H. Zhou, *A semi-synthetic dataset generation framework for causal inference in recommender systems*, arXiv preprint arXiv:2202.11351 (2022).
75. N. Manouselis and C. Costopoulou, *Designing a web-based testing tool for multicriteria recommender systems*, Engineering Letters **13** (2006), no. 3, 417–427.
76. ———, *Experimental analysis of design choices in multiattribute utility collaborative filtering*, International Journal of Pattern Recognition and Artificial Intelligence **21** (2007), no. 02, 311–331.
77. ———, *Preliminary study of the expected performance of maut collaborative filtering algorithms*, The Open Knowlege Society. A Computer Science and Information Systems Manifesto (Berlin, Heidelberg) (M. D. Lytras, J. M. Carroll, E. Damiani,

R. D. Tennyson, D. Avison, G. Vossen, and P. Ordonez De Pablos, eds.), Springer Berlin Heidelberg, 2008, pp. 527–536.
78. B. M. Marlin, S. T. Roweis, and R. S. Zemel, *Unsupervised learning with non-ignorable missing data*, International Workshop on Artificial Intelligence and Statistics, PMLR, 2005, pp. 222–229.
79. F. B. P. Maurera, M. F. Dacrema, L. Saule, M. Scriminaci, and P. Cremonesi, *ContentWise impressions*, Proceedings of the 29th ACM International Conference on Information & Knowledge Management, ACM, oct 2020.
80. J. McInerney, E. Elahi, J. Basilico, Y. Raimond, and T. Jebara, *Accordion: a trainable simulator for long-term interactive systems*, Proceedings of the 15th ACM Conference on Recommender Systems, 2021, pp. 102–113.
81. M. Mirza and S. Osindero, *Conditional generative adversarial nets*, CoRR **abs/1411.1784** (2014).
82. M. Mladenov, C. Hsu, V. Jain, E. Ie, C. Colby, N. Mayoraz, H. Pham, D. Tran, I. Vendrov, and C. Boutilier, *RecSim NG: toward principled uncertainty modeling for recommender ecosystems*, CoRR **abs/2103.08057** (2021).
83. E. Montahaei, D. Alihosseini, and M. S. Baghshah, *DGSAN: discrete generative self-adversarial network*, CoRR **abs/1908.09127** (2019).
84. M. Montaner, B. López, and J. L. de la Rosa i Esteva, *Evaluation of recommender systems through simulated users*, International Conference on Enterprise Information Systems, 2004.
85. D. Monti, E. Palumbo, and G. Rizzo, *Sequeval: An offline evaluation framework for sequence-based recommender systems*, Information **10** (2019), 174.
86. D. Monti, G. Rizzo, and M. Morisio, *A distributed and accountable approach to offline recommender systems evaluation*, CoRR **abs/1810.04957** (2018).
87. D. Monti, G. Rizzo, and M. Morisio, *All you need is ratings: A clustering approach to synthetic rating datasets generation*, arXiv preprint arXiv:1909.00687 (2019).
88. K. Murphy, *Dynamic Bayesian networks: Representation, inference and learning*, UC Berkeley, Computer Science Division, 2002.
89. T. T. Nguyen, P.-M. Hui, F. M. Harper, L. Terveen, and J. A. Konstan, *Exploring the filter bubble: The effect of using recommender systems on content diversity*, Proceedings of the 23rd International Conference on World Wide Web (New York, NY, USA), Association for Computing Machinery, 2014, pp. 677–686.
90. H. Oosterhuis, *Computationally efficient optimization of plackett-luce ranking models for relevance and fairness*, Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (New York, NY, USA), Association for Computing Machinery, 2021, pp. 1023–1032.
91. H. Oosterhuis and M. d. de Rijke, *Robust generalization and safe query-specialization in counterfactual learning to rank*, Proceedings of the Web Conference 2021 (New York, NY, USA), Association for Computing Machinery, 2021, pp. 158–170.
92. D. H. Park and Y. Chang, *Adversarial sampling and training for semi-supervised information retrieval*, The World Wide Web Conference (New York, NY, USA), Association for Computing Machinery, 2019, pp. 1443–1453.

93. N. Park, M. Mohammadi, K. Gorde, S. Jajodia, H. Park, and Y. Kim, *Data synthesis based on generative adversarial networks*, Proc. VLDB Endow. **11** (2018), no. 10, 1071–1083.
94. M. Pasinato, C. E. Mello, M.-A. Aufaure, and G. Zimbrao, *Generating synthetic data for context-aware recommender systems*, 2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence, 2013, pp. 563–567.
95. J. Pearl, *Causal inference in statistics: An overview*, Statistics Surveys **3** (2009), no. none, 96–146.
96. J. Pearl, *Causality: Models, reasoning and inference*, Cambridge University Press, 2009.
97. A. Petrov, I. Safilo, D. Tikhonovich, and D. Ignatov, *MTS KION implicit contextualised sequential dataset for movie recommendation*, 2022.
98. T. W. Philipp Normann, Sophie Baumeister, *OTTO recommender systems dataset: A real-world e-commerce dataset for session-based recommender systems research.*
99. A. Popescul, D. M. Pennock, and S. Lawrence, *Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments*, Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (San Francisco, CA, USA), Morgan Kaufmann Publishers Inc., 2001, pp. 437–444.
100. A. Radford, L. Metz, and S. Chintala, *Unsupervised representation learning with deep convolutional generative adversarial networks*, CoRR **abs/1511.06434** (2015).
101. D. Rafailidis and F. Crestani, *Adversarial training for review-based recommendations*, Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (New York, NY, USA), Association for Computing Machinery, 2019, pp. 1057–1060.
102. K. Ren, Y. Fang, W. Zhang, S. Liu, J. Li, Y. Zhang, Y. Yu, and J. Wang, *Learning multi-touch conversion attribution with dual-attention mechanisms for online advertising*, Proceedings of the 27th ACM International Conference on Information and Knowledge Management (New York, NY, USA), Association for Computing Machinery, 2018, pp. 1433–1442.
103. M. d. C. Rodríguez Hernández, S. Ilarri, R. Hermoso, and R. Trillo, *DatagenCARS: A generator of synthetic data for the evaluation of context-aware recommendation systems*, Pervasive and Mobile Computing **38** (2016).
104. D. Rohde, S. Bonner, T. Dunlop, F. Vasile, and A. Karatzoglou, *RecoGym: A reinforcement learning environment for the problem of product recommendation in online advertising*, CoRR **abs/1808.00720** (2018).
105. A. Said and A. Bellogín, *Comparative recommender system evaluation: Benchmarking recommendation frameworks*, Proceedings of the 8th ACM Conference on Recommender Systems (New York, NY, USA), Association for Computing Machinery, 2014, pp. 129–136.
106. O. Sakhi, D. Rohde, and N. Chopin, *Fast slate policy optimization: Going beyond Plackett-Luce*, 2023.

107. J.-C. Shi, Y. Yu, Q. Da, S.-Y. Chen, and A.-X. Zeng, *Virtual-Taobao: Virtualizing real-world online retail environment for reinforcement learning*, Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 4902–4909.
108. A. Suglia, C. Greco, P. Basile, G. Semeraro, and A. Caputo, *An automatic procedure for generating datasets for conversational recommender systems*, Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum, Dublin, Ireland, September 11-14, 2017 (L. Cappellato, N. Ferro, L. Goeuriot, and T. Mandl, eds.), CEUR Workshop Proceedings, vol. 1866, CEUR-WS.org, 2017.
109. Z. Sun, B. Wu, Y. Wu, and Y. Ye, *APL: Adversarial pairwise learning for recommender systems*, Expert Syst. Appl. **118** (2019), no. C, 573–584.
110. Z. Sun, Z. Si, X. Zang, D. Leng, Y. Niu, Y. Song, X. Zhang, and J. Xu, *KuaiSar: A unified search and recommendation dataset*, 2023.
111. A. Swaminathan, A. Krishnamurthy, A. Agarwal, M. Dudik, J. Langford, D. Jose, and I. Zitouni, *Off-policy evaluation for slate recommendation*, Advances in Neural Information Processing Systems (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.
112. Z. Szlávik, W. Kowalczyk, and M. Schut, *Diversity measurement of recommender systems under different user choice models*, Proceedings of the International AAAI Conference on Web and Social Media **5** (2021), no. 1, 369–376.
113. J. Tang, X. Du, X. He, F. Yuan, Q. Tian, and T.-S. Chua, *Adversarial training towards robust multimedia recommender system*, IEEE Transactions on Knowledge and Data Engineering **32** (2020), no. 5, 855–867.
114. Y. Tong, Y. Luo, Z. Zhang, S. Sadiq, and P. Cui, *Collaborative generative adversarial network for recommendation systems*, 2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW), 2019, pp. 161–168.
115. D. Tran, M. W. Hoffman, D. Moore, C. Suter, S. Vasudevan, and A. Radul, *Simple, distributed, and accelerated probabilistic programming*, Advances in Neural Information Processing Systems (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), vol. 31, Curran Associates, Inc., 2018.
116. T. Tran, R. Sweeney, and K. Lee, *Adversarial Mahalanobis distance-based attentive song recommender for automatic playlist continuation*, Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (New York, NY, USA), Association for Computing Machinery, 2019, pp. 245–254.
117. J. Traupman and R. Wilensky, *Collaborative quality filtering: Establishing consensus or recovering ground truth?*, International Workshop on Knowledge Discovery on the Web, Springer, 2004, pp. 73–86.
118. K. Tso and L. Schmidt-Thieme, *Empirical analysis of attribute-aware recommender system algorithms using synthetic data*, Journal of Computers **1** (2006).
119. K. H. L. Tso and L. Schmidt-Thieme, *Empirical analysis of attribute-aware recommendation algorithms with variable synthetic data*, Data Science and Classification (Berlin, Heidelberg) (V. Batagelj, H.-H. Bock, A. Ferligoj, and A. Žiberna, eds.), Springer Berlin Heidelberg, 2006, pp. 271–278.
120. J. Wang, L. Yu, W. Zhang, Y. Gong, Y. Xu, B. Wang, P. Zhang, and D. Zhang, *IRGAN: A minimax game for unifying generative and discriminative information*

retrieval models, Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (New York, NY, USA), Association for Computing Machinery, 2017, pp. 515–524.
121. K. Wang, Z. Zou, M. Zhao, Q. Deng, Y. Shang, Y. Liang, R. Wu, X. Shen, T. Lyu, and C. Fan, *RL4RS: A real-world dataset for reinforcement learning based recommender system*, 2023.
122. Q. Wang, H. Yin, Z. Hu, D. Lian, H. Wang, and Z. Huang, *Neural memory streaming recommender networks with adversarial training*, Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (New York, NY, USA), Association for Computing Machinery, 2018, pp. 2467–2475.
123. Q. Wang, H. Yin, H. Wang, Q. V. H. Nguyen, Z. Huang, and L. Cui, *Enhancing collaborative filtering with generative augmentation*, Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (New York, NY, USA), Association for Computing Machinery, 2019, pp. 548–556.
124. Z. Wang, M. Gao, X. Wang, J. Yu, J. Wen, and Q. Xiong, *A minimax game for generative and discriminative sample models for recommendation*, Advances in Knowledge Discovery and Data Mining: 23rd Pacific-Asia Conference, PAKDD 2019, Macau, China, April 14-17, 2019, Proceedings, Part II (Berlin, Heidelberg), Springer-Verlag, 2019, pp. 420–431.
125. C.-Y. Wu, A. Ahmed, A. Beutel, A. J. Smola, and H. Jing, *Recurrent recommender networks*, Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (New York, NY, USA), Association for Computing Machinery, 2017, pp. 495–503.
126. X. Xin, T. Pimentel, A. Karatzoglou, P. Ren, K. Christakopoulou, and Z. Ren, *Rethinking reinforcement learning for recommendation: A prompt perspective*, Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (New York, NY, USA), Association for Computing Machinery, 2022, pp. 1347–1357.
127. L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, Modeling Tabular Data Using Conditional GAN, Curran Associates Inc., Red Hook, NY, USA, 2019.
128. S. Yao, Y. Halpern, N. Thain, X. Wang, K. Lee, F. Prost, E. H. Chi, J. Chen, and A. Beutel, *Measuring recommender system effects with simulated users*, CoRR **abs/2101.04526** (2021).
129. F. Yuan, L. Yao, and B. Benatallah, *Adversarial collaborative auto-encoder for top-n recommendation*, 2019 International Joint Conference on Neural Networks (IJCNN), 2019, pp. 1–8.
130. G. Yuan, F. Yuan, Y. Li, B. Kong, S. Li, L. Chen, M. Yang, C. Yu, B. Hu, Z. Li, Y. Xu, and X. Qie, *Tenrec: A large-scale multipurpose benchmark dataset for recommender systems*, 2023.
131. Y. Zhang, F. Feng, X. He, T. Wei, C. Song, G. Ling, and Y. Zhang, *Causal intervention for leveraging popularity bias in recommendation*, Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (New York, NY, USA), Association for Computing Machinery, 2021, pp. 11–20.

132. W. Zhao, B. Wang, J. Ye, Y. Gao, M. Yang, and X. Chen, *Plastic: Prioritize long and short-term information in top-n recommendation using adversarial training*, 07 2018, pp. 3676–3682.
133. F. Zhou, R. Yin, K. Zhang, G. Trajcevski, T. Zhong, and J. Wu, *Adversarial point-of-interest recommendation*, The World Wide Web Conference (New York, NY, USA), Association for Computing Machinery, 2019, pp. 3462–34618.

Saint Petersburg State University,
Russia
E-mail: st068200@student.spbu.ru

Steklov Institute
of Mathematics St. Petersburg,
Russia
E-mail: ilya.shenbin@gmail.com

E-mail: anton.m.alexeyev@gmail.com

Sber AI Lab,
Moscow, Russia
E-mail: AAVolodkevich@sberbank.ru

E-mail: AVaVasilyev@sber.ru

Steklov Institute
of Mathematics at St. Petersburg,
Russia
E-mail: sergey@logic.pdmi.ras.ru

Поступило 6 сентября 2023 г.