Who Are the Best Adopters? User Selection Model for Free Trial Item Promotion

ABSTRACT

With the increasingly fierce market competition, offering a free trial has become a potent stimuli strategy to promote products and attract users. By providing users with opportunities to experience goods without charge, a free trial makes adopters know more about products and thus encourages their willingness to buy. However, as the critical point in the promotion process, finding the proper adopters is rarely explored. Empirically winnowing users by their static demographic attributes is feasible but less effective, neglecting their personalized preferences.

To dynamically match the products with the best adopters, in this work, we propose a novel free trial user selection model named SMILE, which is based on reinforcement learning (RL) where an agent actively selects specific adopters aiming to maximize the profit after free trials. Specifically, we design a tree structure to reformulate the action space, which allows us to select adopters from massive user space efficiently.

The experimental analysis on three datasets demonstrates the proposed model's superiority and elucidates why reinforcement learning and tree structure can improve performance. Our study demonstrates technical feasibility for constructing a more robust and intelligent user selection model and guides for investigating more marketing promotion strategies.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

Free Trial, Recommender System, Reinforcement Learning

ACM Reference Format:

. 2018. Who Are the Best Adopters? User Selection Model for Free Trial Item Promotion. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

The development of mobile technology and fierce market competition has promoted the vigorous development of online platforms. Varieties of E-commerce services such as video/music streaming platforms now play a crucial role in our daily lives. However, with the rise of online items and the limitation of platform display pages, significant exposure opportunities only concentrate on a few popular items. Compared with popular items, low-exposure items usually hold a more flexible pricing strategy due to less similar competitive products, thus embracing relatively large marginal profit [62]. Meanwhile, they are more likely to surprise users and thus increase their loyalty and satisfaction to the platform. Its cumulative benefits often exceed expectations [37]. For instance, more than a quarter of Amazon's book sales come from outside its top 100,000 titles [2]. Reasons for the unbalanced exposure opportunities lie in the recommender system behind the page display mechanism. Generally speaking, recommendation algorithms are usually based on collaborative filtering [58], which filters items of user interest based on user/item similarity. Owing to various similar items and rich transactions, popular items are more inclined to be recommended by algorithms. However, these results will undoubtedly reduce the diversity and coverage of items, exacerbate the cold-start problem, and further intensify the popularity bias problem in recommender system [1]. As a consequence, for customers, it may lead to missing high-quality products, thereby affecting their experience and satisfaction. For platforms, it may cause a considerable loss of potential profit and a reduction of competitiveness.

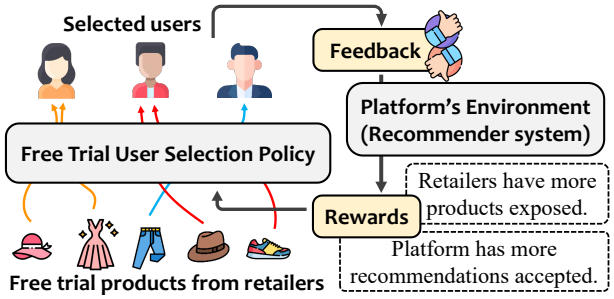

Figure 1: Pipeline of free trial promotion.

To stand out from the fierce market competition, platforms often adopt various promotion methods to quickly grab customers, increase item exposure, and obtain more transactions. Typically, personnel promoting [19, 31], pricing strategy [35, 47], advertising [10, 49], and celebrity endorsement [13, 33] are commonly used to attract customers' attention. However, these approaches have problems like costly, time-consuming, and lacking user feedback. For example, adverting on the Taobao homepage is not only unable to obtain users' immediate comment on the product but also expensive (more than 20,000 US dollars per day).

To fill this gap and win brand reputation, massive companies launch free trial activities, such as Netflix for online movies and TV shows, King's Candy Crush Saga for online games, and Kindle unlimited for eBook. Intuitively, providing the experience opportunities without charge allows customers to obtain direct sensory contact [32], and can further effectively ease users' uncertainty about the utility and quality of the chargeable products. Specially,

Woodstock '18, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

the real feedback from adopters is helpful for improving item quality and also advantageous to pricing decisions [8, 28, 43]. Figure 1 illustrates the pipeline of free trial promotion. Retailers first carefully select a set of users as adopters according to the selection policy. Next, the promoted products are sent to them for free. In return, customers are obliged to give immediate feedback describing their feelings or give amendatory suggestions. Then the platform environment will return rewards signal indicating the increment of exposure which helps adjust the selection policy. In this way, retailers have more products exposed, and the platform has more recommendations accepted.

Although free trial activities have been widely used in the practical marketing promotion scene, far too little attention has been paid to the user selection policy, which is vital in the whole process. Winnowing adopters indiscriminately or simply considering their naive sociological attributes are highly feasible but less effective. Lacking systematically analysis and guidance, these rigid and fixed approaches ignore the dynamic interaction between users and platforms, making it difficult to adapt to the flexible and changeable reality scenario. Additionally, these simple handcrafted rules are insufficient in personalization that is highly required under specific environments. Consequently, promoting items by free trial is not an easy case due to the following challenges: (1) systematically formalizing the loop process: "selecting adopters → receiving feedback → train model → selecting adopters → . . . " and (2) selecting appropriate users who can maximize the exposure of the item under protean interactive environment.

Recently, reinforcement learning (RL) [52] has achieved remarkable success in scenarios requiring dynamic interaction and long-run planning, such as playing games [41, 48], regulating ad bidding [4, 29], and dynamic resource allocation [55, 56]. Considering the dynamic nature of real-world promotion process, we propose **SMILE** (short for "user **S**election **M**odel w**I**th po**L**icy gradi**E**nt") framework under reinforcement learning structure to learn effective selection strategies. It repeatedly selects trial users and improves its own selection strategies through available reward signals until the model converges. Specifically, we model the selection process as an MDP and adopt policy gradient [53], a well-known RL method, to learn how to make decisions for maximizing long-run rewards. For the state representation $\mathbf{s}_t$ at time t, we use the recurrent neural network to embed the historical actions and the corresponding rewards into a low-dimensional hidden vector. For the reward signal, we consider the observable number of Page View (PV) [18, 38] on a pre-defined target item set. Furthermore, to overcome the low convergence problem on a large action space in reinforcement learning, we further reformulate the action space through a balanced hierarchical clustering tree.

In summary, the main contributions of this paper are as follows:

- To the best of our knowledge, this is the first work aiming for increasing promoted item exposure by selecting the best free trial adopters.
- We formulate the problem of user selection and propose an RL-based approach to deal with the dynamic interactions between adopters and the recommender system. For more efficient selection, we design a balanced hierarchical clustering tree to reformulate the action space.
- We conduct extensive experiments on three public datasets to show superior performance and significant efficiency improvement of the proposed SMILE framework and elucidate why RL and the clustering tree structure can improve the performance.

2 RELATED WORK

2.1 Free Trial Marketing Strategy

Product trial was firstly defined as a consumer's first usage experience with a brand or product by Kempf and Smith [32]. It gradually becomes a widely applied marketing strategy for attracting users. According to the survey of marketing week [3], product trials can deepen customers' brand awareness and improve product brand recognition. About 63% of them tend to buy tried products.

Some studies investigated the influence between free trials and users' purchasing intention. Zhu *et al.* [67] explore consumer intention towards free trials of technology-based services and find that perceived usefulness, perceived ease of use, perceived risk, and social influence are essential determinant factors. Sun *et al.* [51] find that most users have little knowledge about new services and thus have low intention to purchase them. Free trial is an effective marketing method to improve users' beliefs about the service. Wang *et al.* [57] prove that users' experience on the mobile newspaper software after the free trial is different from before. Halbheer *et al.* [20] show that free trial strategy influences user's expectations of product quality which is closely linked to the user demand. Foubert and Gijsbrechts [15] find that free trial is more effective in conveying information than advertising because actual usage can quickly reduce user uncertainty. These studies indicate that free trials can improve users' service experience and further influence users' purchase decisions.

Some other studies concentrate on when the firm should adopt the free trial strategy. Cheng *et al.* [9] find that under a strong network effect, the firm is better off offering free trial than segmenting the market by charging a price for a lower quality product. Niu *et al.* [43] find that customers' prior belief plays a key role, and the firm offers free trial only when customers' initial belief is less than a threshold.

These studies indicate that free trials can influence users' purchase decisions and uncover the conditions under which firms should introduce the free trial product. However, there has been little discussion about the process of selecting trial objects which is significant for improving trial quality and better marketing.

2.2 RL-based Recommendation

Reinforcement learning (RL) has been introduced into recommender systems as its advantage of considering users' long-term feedbacks [64, 68]. Zou *et al.* [69] formulate the ranking process as a multi-agent Markov Decision Process, where mutual interactions are incorporated to compute the ranking list. In the 1900s, WebWatcher [30] models the web page recommendation problem as an RL problem and adopts Q-learning to improve its performance. Later, with the development of deep learning, combining deep learning with traditional RL methods is becoming increasingly popular in RS. DQN has been used in clinical applications, such as optimizing

heparin dosage recommendations [42] and optimizing dosage recommendations for sepsis treatment [44]. Recently, many interesting applications have emerged. Google utilizes RL to recommend more suitable video content to its users on YouTube [6]. Fotopoulou *et al.* [14] design an RL-like framework for an activity recommender for students' social-emotional learning. Liu *et al.* [40] use RL to recommend learning activities in a class by monitoring students' learning status.

However, most RL-based models fail to serve for recommender system those need to operate on the large discrete action space. For DQN-based algorithm [64, 65] which needs to find an appropriate action from large action space by value function $Q(s, a)$, to maximize the discounted cumulative reward. The same problem applies to DDPG-based algorithm [25, 63]; it needs to learn a specific ranking function whose complexity of sampling an action grows linearly concerning the size of the action set. Dulac-Arnold *et al.* [12] focus on the large action space problem by modeling the state in the same continuous item embedding space and selecting the items via nearest neighborhood search. Chen *et al.* [5] propose TPGR which aims to represent the item space in the form of a balanced tree and learn a strategy, using policy networks, to select the best child nodes for every non-leaf node. In 2021, Chen *et al.* [7] present a general framework to augment the training of model-free RL agents with modeling user response auxiliary tasks to improve sample efficiency and conduct experiments on industrial recommendation platforms serving billions of users to verify its benefit.

In this paper, based on the structure of TPGR, we propose a balanced tree structure SMILE to select appropriate adopters in large discrete action space to interact with the flexible and changeable scenarios.

3 PROBLEM FORMULATION

In this section, we firstly systematically formalize a new problem called *selecting the best adopters for free trial item promotion*. Then we present our approach based on reinforcement learning to solve this problem.

Given a promoted item set I_p provided for free, we aim to select n adopters that can benefit the retailers and the service provider maximally, i.e., both the profit of the seller and the acceptance of the shopping platform can be improved to the most extent after free trial. We quantify this effect by maximizing the exposure of promoted items without reducing the recommender performance. Our user selection policy will be ceaselessly trained based on the received reward r every round to make better decisions. The complete process is illustrated in Fig. 1.

The whole selection process is formulated as a Markov Decision Process (MDP) in reinforcement learning [52], whose key components are summarized as follows:

- **State**. The model maintains a state $\mathbf{s}_t \in \mathbb{R}^{d_s}$ at time t is regarded as a vector representing information of historical interactions between promoted item $p_i \in I_p$ and system prior to t. In this paper, we obtain the $\mathbf{s}_t$ via a recurrent neural network (RNN).
- **Action**. The model makes an action a_t at time t is to select one adopter for item p_i. Let $\mathbf{e}_{a_t} \in \mathbb{R}^{d_a}$ denote the representation vector of action a_t. In this paper, each action a_t selects only one user u. Hence, we have $\mathbf{e}_{a_t} = \mathbf{e}_u$.
- **Reward**. The recommender system returns a reward score r_t reflecting the exposure of the target items at time t.
- **Transition**. The transition function gives the next state $\mathbf{s}_{t+1}$ after taking action a_t. Due to the state reflecting the historical interactions of the target item, it will be changed given the newly selected user and its corresponding rewards.
- **Policy Network**. The policy network $\pi_\theta = \pi_\theta(a_t|\mathbf{s}_t)$ is the target policy that decides how to make an action a_t conditioned on the state $\mathbf{s}_t$. In this paper, policy network is designed with a fully-connected neural network and a softmax activation function on the output layer. It takes the state $\mathbf{s}_t$ as input and outputs a probability distribution over possible outputs. The probability of a choice a_t is computed as follows:

$$\pi_\theta(a_t|\mathbf{s}_t) = \text{Softmax}(\sigma\left(\mathbf{W_s}^T\mathbf{s_t} + b\right)), \tag{1}$$

where σ is a non-linear activation function, $\mathbf{W_s}$ denotes the weight matrix and b is bias value.

As a consequence, we regard a free trial user selection process as $(\mathbf{s}_1, a_1, r_1, \mathbf{s}_2, \cdots, \mathbf{s}_n, a_n, r_n, \mathbf{s}_{n+1})$, which represents one episode. In detail, the state vector $\mathbf{s}_t$ enters the policy network and outputs an action a_t, i.e., the generated next adopter; then the recommender system returns a reward r_t measuring the quality of this action. The selection process will terminate at a specific state $\mathbf{s}_{n+1}$ when the pre-defined episode length is satisfied. Without loss of generality, we set the length of an episode n to a fixed number [4].

4 PROPOSED METHOD

Figure 2 illustrates the framework of our proposed model SMILE, which contains three modules: A state tracker that provides the state vector $\mathbf{s}_t$ based on previous decisions and rewards, a user selector that outputs the selected trial user a_t, and a reward calculator that returns a reward signal r_t measuring the effect of the free trial on the recommender system. In what follows, we will elaborate on the three modules in detail.

4.1 State Representation

The state is designed to understand item preference in each round of the selection. Figure 3 illustrates the model for generating the state. We adopt a simple recurrent unit (SRU) [36], a RNN model that simplifies the computation and exposes more parallelism, to learn the hidden representations. To integrate historical interactions and feedback information, SRU takes the selected users and the corresponding rewards as input and encodes them into a low-dimensional state vector $\mathbf{s}_i \in \mathbb{R}^{d_s}$. Specially, we set the initial state $\mathbf{s}_1$ as promoted item profiles vector $\mathbf{e}_i$. It can be learned in an end-to-end manner or pre-trained by supervised learning models such as matrix factorization (MF). For the convenience of implementation, we take the pre-trained item matrix to compute $\mathbf{e}_i$ in this part.

Assuming the model is learning the t-th state vector $\mathbf{s}_t$, it takes a sequence of previous selected user embeddings $\{\mathbf{e}_{u_1}, \mathbf{e}_{u_2}, \cdots, \mathbf{e}_{u_{t-1}}\}$

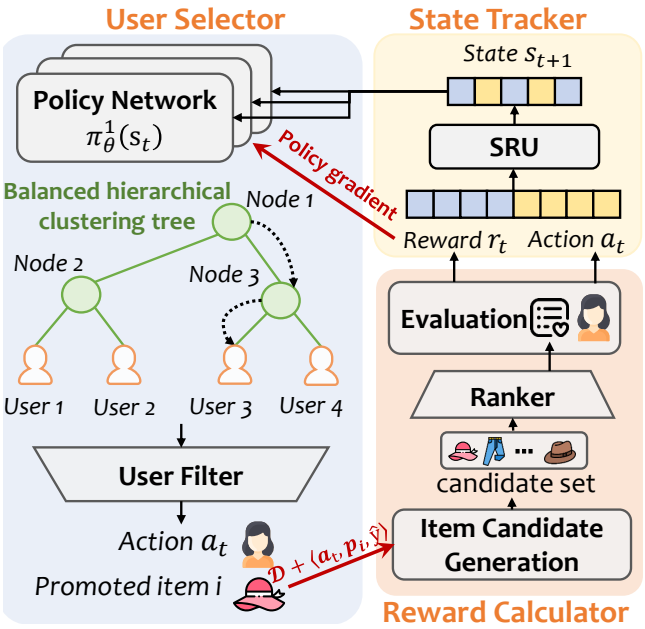

Figure 2: The framework of SMILE.

and their corresponding reward vectors $\{\mathbf{e}_{r_1}, \mathbf{e}_{r_2}, \cdots, \mathbf{e}_{r_{t-1}}\}$ before timestep t as input. Each user is mapped to an embedding vector $\mathbf{e}_{u_i} \in \mathbb{R}^{d_a}$ which is the i-th row of user embedding matrix $\mathbf{U}$. $\mathbf{U} \in \mathbb{R}^{|u| \times d_a}$ is pre-trained by MF [17, 34, 66] and is fixed in the RL process. Equation 2 shows the objective function of MF, where $|u|$ is the number of users and $|i|$ is the number of items. $\mathbf{V} \in \mathbb{R}^{|i| \times d_a}$ represents the item embedding matrix. $\mathbf{Y} \in \mathbb{R}^{|u| \times |i|}$ denotes the user-item interaction matrix, where $y_{ui} = y$ if the user u rated item i as y, otherwise $y_{ui} = 0$.

$$\min_{U \in \mathbb{R}^{|u| \times d_a}, V \in \mathbb{R}^{|i| \times d_a}} \left\| Y - UV^T \right\|_F^2. \tag{2}$$

Simultaneously, the reward value from r_{min} to r_{max} is linearly mapped into a h-dimensional one-hot vector $\mathbf{e}_{r_i} \in \mathbb{R}^{d_h}$ as Equation 3. Assuming that the range of reward values is $(r_{min}, r_{max}]$, we firstly normalize each reward value r to range $(0, h]$ and then utilize the $one\_hot(i, h)$ function to output a h-dimensional vector, where the value of the i-th element is one and the others are set to zero.

$$\begin{aligned} \mathbf{e}_{r_i} &= \text{one\_hot}\left(\hat{r}, h\right), \\ \hat{\mathbf{r}} &= h - \left\lfloor \frac{h \times (r_{max} - r)}{r_{max} - r_{min}} \right\rfloor. \end{aligned} \tag{3}$$

To retain richer semantic information, we concatenate the user embedding vector $\mathbf{e}_{u_t}$ and one-hot reward vector $\mathbf{e}_{r_t}$ into $\mathbf{x}_t = (\mathbf{e}_{u_t}, \mathbf{e}_{r_t})$. Next, we take $\mathbf{x}_t$ as the input of SRU to learn state representation, and the update function of a SRU cell is defined as

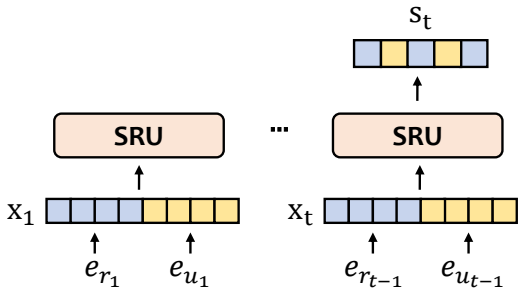

Figure 3: The model for generating the state.

$$\begin{aligned} \tilde{\mathbf{x}}_t &= \mathbf{W}\mathbf{x}_t, \\ \mathbf{f}_t &= \sigma\left(\mathbf{W}_f \mathbf{x}_t + \mathbf{b}_f\right), \\ \mathbf{g}_t &= \sigma\left(\mathbf{W}_g \mathbf{x}_t + \mathbf{b}_g\right), \\ \mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + (1 - \mathbf{f}_t) \odot \tilde{\mathbf{x}}_t, \\ \mathbf{h}_t &= \mathbf{g}_t \odot g\left(\mathbf{c}_t\right) + (1 - \mathbf{g}_t) \odot \mathbf{x}_t, \end{aligned} \tag{4}$$

where $\mathbf{x}_t$ denotes the input vector, $\mathbf{f}_t$ and $\mathbf{g}_t$ denote the forget gate and reset gate respectively, $\mathbf{c}_t$ and $\mathbf{h}_t$ indicate the internal state and output state, and $\odot$ is the elementwise product operator. The final hidden state $\mathbf{h}_t$ is the representation of current state $\mathbf{s}_t$, which is then fed into policy network.

4.2 Architecture of User Selector

To find the best adopters, we propose a two-stage selecting module called *user selector*. It first utilizes the balanced hierarchical clustering tree structure to select the initial trial user and then leverages a user filter to sort out users further. We will separately introduce the two parts in the following.

4.2.1 Balanced Hierarchical Clustering over Users. As mentioned earlier, most RL-based models suffer problems from operating on the large discrete action space which makes the training inefficient and ineffective. In other words, the model has to explore a large discrete action space to find the target adopters to earn positive rewards which makes the time complexity of making a decision linear to the size of the action space. Inspired by the work of Chen *et al.* [5], we reformulate the user action space by building a balanced hierarchical clustering tree $\mathcal{T}$ to achieve high effectiveness. As shown in the *user selector* module in Fig. 2, each leaf node is mapped to a particular user, and each non-leaf node is associated with a policy network. The process of selecting an appropriate user is regarded as a top-down moving from the root to a leaf node.

For the convenience of presentation and implementation, we employ the simple and popular divisive approach to build up the tree $\mathcal{T}$. In this approach, the original data points (i.e., the representation of users) are divided into several clusters, and each cluster is divided into smaller sub-clusters. To make the tree balance, for each node, the difference between heights of its sub-trees is at most one, and the number of child nodes for each non-leaf node is c except for the nodes on the second-to-last layer, whose numbers of child nodes are at most c — considering that the number of users is insufficient to

support constructing a perfect c-ary tree. The value of c is calculated by the whole user set $\mathcal{U}$ and the tree depth d, which is defined as follows:

$$c = \left\lceil |\mathcal{U}|^{\frac{1}{d}} \right\rceil, \tag{5}$$

where $\lceil x \rceil$ returns the smallest integer no less than x.

Next, we employ the PCA-based clustering algorithm to perform the balanced hierarchical clustering over users, which takes a group of user vectors $\{\mathbf{e}_{u_1}, \mathbf{e}_{u_2}, \cdots, \mathbf{e}_{u_m}\}$ and the number of child nodes c as inputs. Specially, the input user vector $\mathbf{e}_{u_i} \in \mathbb{R}^{d_a}$ is the i-th row of the user embedding matrix $\mathbf{U}$. Then, these vectors are divided into c balanced clusters. By repeatedly applying the clustering algorithm until each sub-cluster is associated with only one user, a balanced clustering tree is successfully constructed.

Taking a scenario with four users for illustration, the *user selector* module in Fig. 2 shows the constructed balanced clustering tree with the tree depth d set to two. On the tree $\mathcal{T}$, each leaf node ($user_1 \sim user_4$) represents a user $u \in \mathcal{U}$ and each non-leaf node ($node_1 \sim node_3$) has an independent policy network π_θ. To begin with, a path $\mathcal{P}$ starts at the root node ($node_1$) which takes the aforementioned state $\mathbf{s}_t$ as input and outputs a probability distribution over c child nodes. And the node ($node_3$) with the maximum probability will be extended to the path. Then, the path $\mathcal{P}$ keeps extending until reaching a leaf node then the corresponding user ($user3$) is the selected user.

Accordingly, getting a trial user at timestep t is the process of generating a path $\mathcal{P}_t = \{c_1, c_2, \cdots, c_d\}$ from the root node to a leaf node. It consists of d (i.e., the number of layers in the tree) choices, and each choice is represented as an integer between one and c (i.e., the maximum number of child of each node). Given the state, the probability of action at timestep t is

$$\pi_\theta\left(a_t \mid \mathbf{s}_t\right) = \prod_{i=1}^{d} \pi_{\theta_i}\left(c_i \mid \mathbf{s}_t\right), \tag{6}$$

where $\pi_{\theta_i}\left(c_i \mid \mathbf{s}_t\right)$ is the probability of making each choice in the corresponding policy network from root to the chosen action which is computed in Equation 1. Our goal is to optimize all the policy network set $\pi_\theta = \{\pi_{\theta_1}, \pi_{\theta_2}, \cdots, \pi_{\theta_Q}\}$, where Q denoting the number of non-leaf nodes of the tree which is computed by $Q = \frac{c^d - 1}{c-1}$.

4.2.2 Filtering out Inappropriate Users. By constantly leveraging the tree structure, we can get a set of initial trial users. Unfortunately, not every individual is fond of the free trial items. Indiscriminately providing items to all the primary selected users may hurt customer experiences and reduce their intention of final purchasing. Meanwhile, the platform may receive some low ratings or negative comments, resulting in a negative impact on sales. Consequently, it is crucial to select users who favor the promoted items and are more inclined to give high ratings.

To this end, we design a user filter module as shown in the *user filter* module in Fig. 2 to mimic user preference towards target items based on MF. Without loss of generality, we regard the ratings higher than 3.5 as positive ratings (notice that the highest rating is five) and take this criterion to eliminate inappropriate users. In this way, we can get the filtered users (i.e., action a_t) who are interested in the promoted items and are more likely to give high ratings. And we regard them as the ultimately selected adopters.

4.3 Building Reward Function

With the help of the user selector module, trial adopter a_t is obtained successfully. Then, we adopt the free trial activity to collect immediate user feedback $\langle u, p_i, \hat{y} \rangle$ and append the triplet to the original dataset $\mathcal{D}$, where $\hat{y}$ denotes the predicted rating between the adopter u and the promoted item i computed by Equation 2. In this way, $\mathcal{D}$ embraces the newly created interactions as well as historical interactions concurrently.

Next, we develop a measurable indicator that takes the dataset $\mathcal{D}$ as input and outputs the free trial's effect on the recommender system. Intuitively, real-time sales of promoted items is a favorable indicator reflecting whether the products sell well. But it is impractical to train our model online with real users to capture sales changes because three reasons: (1) For the model, the increment in sales is slow and usually takes weeks to collect sufficient data to make the assessment statistically significant; (2) for users, interacting with a half-baked system can hurt experiences; (3) for the platform, collecting real-time user feedback requires expensive engineering and logistic overhead [16, 26, 27].

In this paper, we introduce the concept of Page View (PV) [18, 38], which is an available evaluation indicator to measure items' exposure within a certain period on the recommender system. We define our reward value as the average exposure of the target promoted item set I_p, which is represented as follows:

$$\mathcal{R}(\mathbf{s}_t, \mathbf{a}_t) = \sum_{u \in \mathcal{U}} \frac{\left|L_u \cap I_p\right|}{\left|I_p\right|}, \tag{7}$$

where $\mathcal{U}$ denotes the whole user set, L_u represents the recommended K items to user u, and I_p is the target item set to be promoted.

As shown in the *reward calculator* module in Fig. 2, the recommended results L_u are generated by the *item candidate generation* and the *ranker* modules [50]. Specifically, the item candidate generation [11] module selects hundreds of items from the entire item corpus to construct a candidate set C_u for each user u. The ranker [22, 24, 59, 61] is responsible for ranking the items in C_u based on Bayesian Personalized Ranking (BPR) [45] algorithm, which can estimate user's preference score on items. Then the K items with the highest scores will be recommended in the final recommendations list L_u. Therefore, the reward value r_t given state $\mathbf{s}_t$ and action a_t is calculated by Equation 7 and is regarded as the signal guiding the whole optimization process.

4.4 Model Optimization with Policy Gradient

As mentioned earlier, we utilize a policy network to learn the strategy of choosing the best subclass at each non-leaf node given the current state. Our main idea is to find the best adopters that can maximize the exposure of promoted items. As illustrated in Fig. 2, for one thing, the output of the reward calculator r_t will enter the state tracker module to learn the next state vector $\mathbf{s}_{t+1}$; for another thing, r_t will be used to train the policy network set $\pi_\theta = \{\pi_{\theta_1}, \pi_{\theta_2}, \cdots, \pi_{\theta_Q}\}$ in the balanced hierarchical clustering tree in user selector module.

Algorithm 1: The procedure of SMILE

Input: Episode length n, Tree depth d, Promoted item set I_p, Original data $\mathcal{D}$, Reward function $\mathcal{R}$, User set $\mathcal{U}$ with representations
Output: Model parameters θ

Calculate the number of child nodes c and non-leaf nodes Q
Construct a balanced clustering tree $\mathcal{T}$ with c child nodes
for $j = 1$ *to* Q **do**
  initialize $\theta_j \leftarrow$ random values
end
repeat
  for $t = 1$ *to* n **do**
    Sample $\mathcal{P}_t = \{c_1, c_2, \cdots, c_d\}$
    Map p_t to a user a_t after passing user filter
    for $i = 1$ *to* $|I_p|$ **do**
      $\mathcal{D} = \mathcal{D} + < a_t, p_i, y >$
    end
    $r_t = \mathcal{R}(\mathbf{s}_t, a_t)$
    if $t < n$ **then**
      Calculate $\mathbf{s}_{t+1}$ by state tracker
    end
  end
  Get $\mathcal{M} = (\mathbf{s}_1, a_1, r_1, \cdots, \mathbf{s}_n, a_n, r_n)$
  for $t = 1$ *to* n **do**
    Update θ according to Equation 9
  end
until *converged*
return θ

We utilize the most commonly used policy gradient methods REINFORCE algorithm [60] to train our model. The objective is to maximize the expected discounted cumulative rewards, i.e.,

$$J(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{n-1} \gamma^t r_t(\mathbf{s}_t, a_t)\right], \tag{8}$$

where γ is the discount factor, η is the learning rate, and r_t denotes target items' exposure within a certain period on recommender system after the free trial process. We can update the parameter θ by Equation 9:

$$\begin{aligned} Q^{\pi_\theta}(\mathbf{s}_t, a_t) &= \sum_{i=t}^{n} \gamma^{i-t} r_i, \\ \Delta\theta &= \nabla_\theta \log \pi_\theta(a_t \mid \mathbf{s}_t) Q^{\pi_\theta}(\mathbf{s}_t, a_t), \\ \theta &= \theta + \eta\Delta\theta, \end{aligned} \tag{9}$$

where $\pi_\theta(a_t \mid \mathbf{s}_t)$ denotes the probability of taking action a_t given state $\mathbf{s}_t$; $Q^{\pi_\theta}(s, a)$ is the cumulative rewards. The overall training process is shown in Algorithm 1.

4.5 Complexity Analysis

In this section, we discuss the complexity of SMILE from time complexity and space complexity. As every policy network is implemented as a full-connected layer, we consider both the time and the space complexity of each policy network as $O(c)$.

Time complexity. Considering the process of selecting one user, the state vector will via d policy networks and each time needs to choose from at most c options. Therefore, the time complexity of selecting one user is $O(d \times c) \simeq O(d \times \left|A^{\frac{1}{d}}\right|)$. As we usually set the value of tree depth d to a small number, our tree structure method can significantly reduce the time complexity compared with other RL-based methods whose time complexity is $O(|A|)$, where A denotes the action space.

Space complexity. The space complexity mainly comes from two parts: the number of policy network (i.e., the number of non-leaf nodes) and its optional range (i.e., the child nodes). We firstly calculate the number of policy network Q. The space complexity of the SMILE is $O(Q \times c) \simeq O(\frac{c^d - 1}{c-1} \times c) \simeq O(|A|)$, which is equal to other RL-based models.

5 EXPERIMENTS AND RESULTS

We conducted extensive experiments on three public datasets to justify our model's superiority and reveal the reasons for its effectiveness.

In this section, we first introduce the statistics of the datasets and present five baselines whose selection strategies are based on user behavior patterns. Besides, we design two evaluation metrics for the reward of the selection models and use two more metrics for the influence of the selection over the recommendation system. We also specify the experiment setup for the evaluation.

Specifically, we will answer the following research questions to unfold the experiments.

RQ1: What is the influence of varying the number of trial users on the exposure effect?

RQ2: Compared with the static free trial user selection policy, how does our model perform?

RQ3: What are the benefits of the tree structure and the impact of tree depths in our model?

RQ4: How does the free trial process influence the effectiveness of the recommender system?

5.1 Datasets

We conducted experiments on three public datasets as follows, and the statistical information of these datasets is shown in Table 1.

- **Movielens100K**[1]: Movielens100K [21] consists of 100,000 movie ratings of 943 users for 1,682 movies.
- **Movielens1M**[2]: Movielens1M [21] contains one million anonymous ratings of 3,900 movies by 6,040 users.
- **Ciao**[3]: Ciao [54] is collected from a real-world social media website. From the originally dataset, we filter out users who rated less than three items and items that received less than

[1] http://grouplens.org/datasets/movielens/100k/
[2] https://grouplens.org/datasets/movielens/1m/
[3] http://www.cse.msu.edu/ tangjili/trust.html

three ratings, which leaves us 6,626 users, 15,048 items, and 161,813 ratings.

5.2 Baselines

Since there is a lack of study investigating the problem of finding the best trial adopters for item promotion, we take five static policies based on user behavior patterns as our baselines:

- **Random**: selecting trial adopters at random.
- **Activity**: ranking users according to their activity (i.e., the number of user transaction volume) and taking the most active users as the trial adopters.
- **Inactivity**: contrary to activity strategy, the inactivity strategy takes the least active users as the trial adopters.
- **HighRating**: ranking users according to their historical ratings and selecting users who prefer to score high ratings as the trial adopters.
- **LowRating**: contrary to highRating strategy, lowRating strategy selects users who prefer to score low ratings as the trial adopters.

Table 1: The statistics of datasets.

DataSet	#Users	#Items	#Ratings	Density
Movielens100K	943	1,682	100,000	6.30%
Movielens1M	6,040	3,900	1,000,209	4.25%
Ciao	7,935	16,200	171,465	0.13%

5.3 Evaluation Metrics

We design two metrics for evaluating the rewards of selection models. As the RL-based methods aim to gain the optimal long-run rewards, we use the average reward (*Avg_reward*) over each selection episode for each promoted item as one evaluation metric. Besides, we adopt the maximum reward (*Max_reward*) value to measure the best performance of selection strategies quickly.

Besides, we employ two widely adopted metrics *Precision@k* and *Recall@k* [23] with $k = 10$ to measure the free trial influence over recommender system. *Precision@k* is the proportion of recommended items in the top-k set that are relevant. *Recall@k* is the fraction of relevant items that have been retrieved in the top-k relevant items.

5.4 Experimental Setup

5.4.1 Simulating User Preference. As mentioned before, not everyone favors the promoted items, so we design a user filter module to simulate adopters' preferences on them and filter those who are not interested. Empirically, the promoted item set I_p always lacks exposure opportunities, i.e., possessing subtle interaction data. To overcome the obstacle of accurately mimicking users' predilections on promoted items, we expressly set I_p as the popular items with considerable interaction data which helps improve prediction accuracy. Next, to imitate the low exposure feature of promoted items, we delete part of their transactions and reconstruct a new dataset. Finally, the user filter module is trained on the original dataset, ensuring the prediction precision between adopters and promoted items. Taking the movielens100k dataset as an example, we first set the promoted items I_p as the top 1% popular items and train our prediction model (i.e., MF) on the original dataset. Then we delete their transactions until the original 5% interaction data is retained. In this way, we can utilize the entire dataset to make predictions and the processed dataset is used in subsequent experiments.

5.4.2 Generating Candidates. To explore the influence of adopters' number on the exposure effect, we select an increasing number of adopters linearly by random metric. Figure 4 reflects that not a more significant number of adopters achieve higher increased exposures. The curve keeps rising at first, and after achieving a peak, it gradually drops. Considering more adopters requires more free items, which is none other than a considerable expense. Therefore, we must control the number of adopters, and it is urgent to set a suitable selection policy to winnow users and achieve high exposure.

Taking the movielens1M dataset as an example, a candidate item set c_u is made up by the promoted set I_p and the 10% other items selected randomly. For the recommendation results, we assume that each user only views K items and defines the items with the highest estimated preference scores in ranker as L_u. Therefore, there will be a high reward if target items frequently appear in users' recommendation results L_u.

5.4.3 Implementation Details. In our experiments, we aim to select 5% adopters among all users, which is the same as the episode length. Once a user u is sampled in each episode, it will be removed from the available users; thus, no repeated users occur in an episode. For the balanced hierarchical clustering tree, we set the tree depth d to two which can achieve the best performance. In the optimization process, we set the discount factor γ to 0.9 and optimize all models with the Adam optimizer.

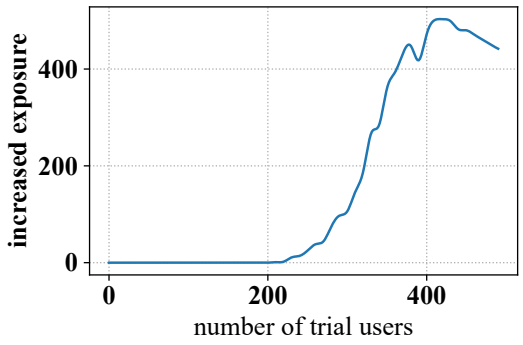

Figure 4: Influence of the increased adopters' number on exposure effect.

5.5 Investigation on The Number of Adopters (RQ1)

In real-world marketing scenarios, the platform will avoid selecting too many trial users considering the limited money and time. To investigate the influence of trial user number on the exposure effect, we conduct two experiments on the Movielens100K dataset.

Figure 4 shows the change of total increased exposure when we incrementally raise the number of adopters by random selection

Table 2: Overall selection performance comparison.

DataSet	Movielens100K		Movielens1M		Ciao	
Metric	Avg_reward	Max_reward	Avg_reward	Max_reward	Avg_reward	Max_reward
Random	4.72	36	10.72	40	32.88	47
Activity	0.14	7	9.78	69	22.42	35
Inactivity	11.57	54	15.46	43	37.5	51
HighRating	8.54	55	13.40	54	34.75	45
LowRating	3.93	24	6.80	27	33.8	40
SMILE	**138.3**	**213**	**55.7**	**89**	**62**	**69**

policy. Surprisingly, they are not always positively correlated, i.e., more adopters do not correspond to more exposure opportunities, which is somewhat counterintuitive. In the beginning, the increased exposure is zero because of the tiny number of adopters. Later, with more adopters selected, the exposure curve begins to rise, indicating that the promotion effect has been achieved. However, the curve gradually decreases after reaching the peak, suggesting that superfluous adopters will reduce the marketing effect. It may be because that the newly added connections violate the authentic user preferences distribution. As a consequence, more adopters require high costs and time but may fail to achieve better performance. It confirms the necessity of exploring a suitable free trial user selection policy that aims to achieve high exposure at the lowest expense.

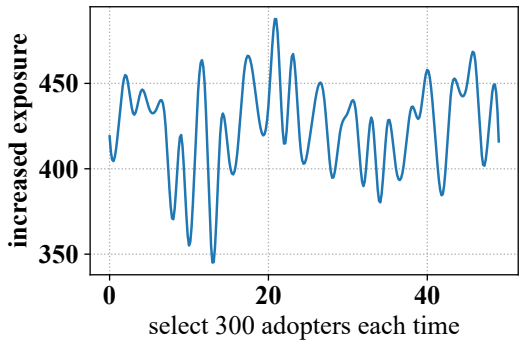

Figure 5: Influence of the same number of adopters on exposure effect.

Figure 5 illustrates the different increased exposure with the same number (three hundred) of adopters every time. Interestingly, the result is not stable as we wish, fluctuating around four hundred. This inconsistency could be attributed to the inaccurate performance of the recommendation algorithm. Therefore, it is rational to record the maximum reward value as an evaluation metric to reflect the potential maximum exposure effect.

5.6 Overall Selection Performance Comparison (RQ2)

To validate the superiority of SMILE, we conduct experiments on three datasets. The summarized results are presented in Table 2. We highlight the best results of all models in boldface. According to the results, we note the following observations:

1) The inactive selection strategy gets a higher average reward than the active one. It might be because the inactive users hold much fewer interactions thus are more sensitive to new interactions and more likely to affect the recommendation system. We can also observe that users who prefer to score high ratings achieve higher reward value users prefer low ratings. The reason is associated with the user filter module, which tends to drop users favoring low ratings. 2) The maximum rewards of the five baseline selection strategies are almost similar, indicting the randomness and instability of selecting users by their attributes. On the whole, it is hard to find a fixed selection strategy from the baselines for the best performance. We argue that these rigid and stationary methods cannot adjust to the flexible reality scene. 3) Among these methods, our proposed SMILE framework achieves the best performance in both metrics. Especially in the Movielens100K dataset, the average reward is far greater than the maximum reward of baselines, reflecting its significant advantages in small datasets. The main reasons are threefold. First, it adopts RL technology for long-run planning and dynamic adaptation, which is absent in other baselines. Second, the hierarchical clustering tends to cluster similar users in the same subtree, which incorporates additional user similarity information into our model. Third, the hierarchical tree-structured can ease the training process to some degree.

5.7 Benefits of Hierarchical Clustering Tree (RQ3)

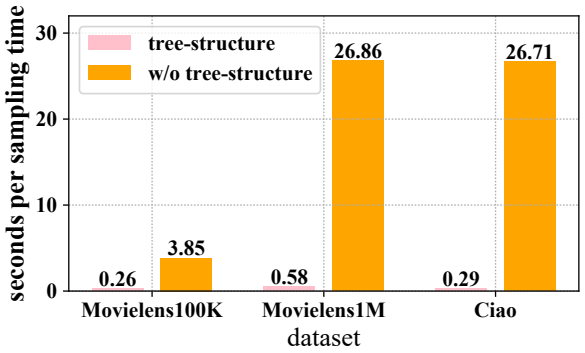

Figure 6: Influence of tree structure.

In the user selector module, we conduct a tree-structured decomposition and adopt a certain number of policy networks with simple architectures. To investigate its feasibility and efficiency, we conduct two experiments. First, we compare the running time in the sampling stage between the model with the hierarchical clustering

Table 3: The free trial influences over recommender systems.

DataSet	Movielens100K		Movielens1M		Ciao	
Metric	Precision@10	Recall@10	Precision@10	Recall@10	Precision@10	Recall@10
Original	0.2194	0.0505	0.1032	0.0347	0.0326	0.0167
SMILE	0.2364	0.0538	0.1066	0.0378	0.0395	0.0198
Improvement	**7.75%**	**6.53%**	**3.29%**	**8.93%**	**21.4%**	**18.7%**

tree structure and the model without a tree structure (i.e., only preserves one policy network, which takes a state as input and gives the policy possibility distribution on all users) on three datasets. To make the comparison fairly, all the experiments are conducted on the same machine with i7-6850K CPU @ 3.60GHz. As shown in Fig. 6, we can easily observe that our hierarchical clustering tree structure takes the shortest running time when sampling an action.

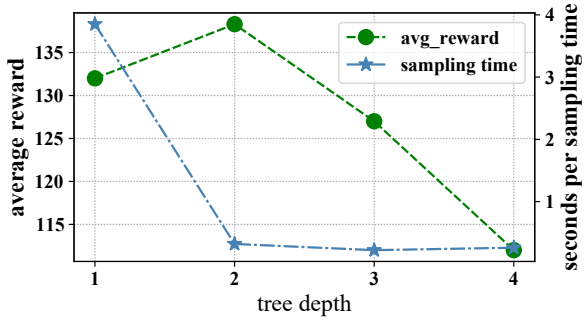

Figure 7: Results under different tree depth.

Next, we set depth from one to four to further explore how different tree depth influences the efficiency and performance of SMILE on the Movielens100K dataset. In particular, the model with tree depth set to one is equivalent to without a tree structure. The green curve in Fig. 7 representing the sampling time shows that the tree structure model significantly improves efficiency. The blue curve presents the performance under different tree depths, from which we can notice that the model with the tree depth set to two reaches the peak performance, while other tree depths cause a slight performance drop. Therefore, setting the depth of the tree to two can significantly reduce the time complexity and provide better performance.

5.8 Free Trial Influences over RS (RQ4)

To simulate the trial process, we establish a triplet $\langle u, p_i, \hat{y} \rangle$ denoting the new interactions between adopter u and promoted target item p_i. However, will the implementation change the actual data distribution and then degrade the performance of the recommendation algorithm? To answer this question, we conduct an experiment to explore the effects of the free trial influences over the BPR recommendation algorithm and explain the applicability of our proposed SMILE model.

As shown in Table 3, instead of reducing its performance, SMILE can improve the accuracy of recommendations. Especially on the Ciao dataset, it achieves an increment by 21.4% and 18.7% in $Precision@10$ and $Recall@10$, respectively. The improvement of the recommender system is equivalent to the increase of users' probability of purchasing products from the recommendation list, which is none other than a fantastic signal indicating the expansion of the global sale on the platform.

This increment may be due to the following two reasons. First, the selected trial users are all interested in promoted items; hence no original data distribution is changed (i.e., the newly added interaction data is consistent with users' historical preferences). Second, more interactions will provide richer information for model learning that alleviates the data-sparse problem and further improve the performance.

6 CONCLUSION AND FUTURE WORK

In this paper, we systematically analyze and formulate the problem of selecting suitable adopters to increase item exposure in the scenario of the recommender system. We propose a novel free trial user selection model named SMILE, which consists of three modules: A state tracker that provides a state vector based on previous decisions and rewards, a user selector that produces selected adopter based on hierarchical clustering over user action space, and a reward calculator that evaluates the selection performance. At last, we utilize policy gradient to update our model. Experiments conducted on three public datasets demonstrate that our proposed SMILE framework can achieve better performance with higher efficiency.

In the future, we seek to tackle the adopter selection problem in the social network environment as the message diffusion in the social graph is a significant factor influencing the promotion effect. We also plan to introduce offline reinforcement learning in our scenario, i.e., learning a debiased user model based on offline data and providing reward value to train our reinforcement learning policy. More powerful reinforcement learning algorithms such as Proximal Policy Optimization [46] and Deep Deterministic Policy Gradient [39] will also be taken into consideration in our future work.

REFERENCES

[1] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. 2019. The Unfairness of Popularity Bias in Recommendation. 2440 (2019). http://ceur-ws.org/Vol-2440/paper4.pdf
[2] Chris Anderson. 2006. *The long tail: Why the future of business is selling less of more.* Hachette Books.
[3] Kapil Bawa and Robert Shoemaker. 2004. The effects of free sample promotions on incremental brand sales. *Marketing Science* 23, 3 (2004), 345–363.
[4] Han Cai, Kan Ren, Weinan Zhang, Kleanthis Malialis, Jun Wang, Yong Yu, and Defeng Guo. 2017. Real-time bidding by reinforcement learning in display advertising. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. 661–670.
[5] Haokun Chen, Xinyi Dai, Han Cai, Weinan Zhang, Xuejian Wang, Ruiming Tang, Yuzhou Zhang, and Yong Yu. 2019. Large-scale interactive recommendation with tree-structured policy gradient. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 3312–3320.

[6] Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H Chi. 2019. Top-k off-policy correction for a REINFORCE recommender system. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 456–464.

[7] Minmin Chen, Bo Chang, Can Xu, and Ed H Chi. 2021. User Response Models to Improve a REINFORCE Recommender System. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 121–129.

[8] Hsing Kenneth Cheng and Yipeng Liu. 2012. Optimal software free trial strategy: The impact of network externalities and consumer uncertainty. *Information Systems Research* 23, 2 (2012), 488–504.

[9] Hsing Kenneth Cheng and Qian Candy Tang. 2010. Free trial or no free trial: Optimal software product design with network effects. *European Journal of Operational Research* 205, 2 (2010), 437–447.

[10] Yu-Jing Chiu, Hsiao-Chi Chen, Gwo-Hshiung Tzeng, and Joseph Z Shyu. 2006. Marketing strategy based on customer behaviour for the LCD-TV. *International journal of management and decision making* 7, 2-3 (2006), 143–165.

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

[12] Gabriel Dulac-Arnold, Richard Evans, Hado van Hasselt, Peter Sunehag, Timothy Lillicrap, Jonathan Hunt, Timothy Mann, Theophane Weber, Thomas Degris, and Ben Coppin. 2015. Deep reinforcement learning in large discrete action spaces. *arXiv preprint arXiv:1512.07679* (2015).

[13] B Zafer Erdogan. 1999. Celebrity endorsement: A literature review. *Journal of marketing management* 15, 4 (1999), 291–314.

[14] Eleni Fotopoulou, Anastasios Zafeiropoulos, Michalis Feidakis, Dimitrios Metafas, and Symeon Papavassiliou. 2020. An interactive recommender system based on reinforcement learning for improving emotional competences in educational groups. In *International Conference on Intelligent Tutoring Systems*. Springer, 248–258.

[15] Bram Foubert and Els Gijsbrechts. 2016. Try it, you'll like it—or will you? The perils of early free-trial promotions for high-tech service adoption. *Marketing Science* 35, 5 (2016), 810–826.

[16] Chongming Gao, Wenqiang Lei, Xiangnan He, Maarten de Rijke, and Tat-Seng Chua. 2021. Advances and Challenges in Conversational Recommender Systems: A Survey. *CoRR* abs/2101.09459 (2021). arXiv:2101.09459 https://arxiv.org/abs/2101.09459

[17] Chongming Gao, Shuai Yuan, Zhong Zhang, Hongzhi Yin, and Junming Shao. 2019. BLOMA: Explain Collaborative Filtering via Boosted Local Rank-One Matrix Approximation. In *Database Systems for Advanced Applications*, Guoliang Li, Jun Yang, Joao Gama, Juggapong Natwichai, and Yongxin Tong (Eds.). Springer International Publishing, Cham, 487–490.

[18] Florent Garcin, Boi Faltings, Olivier Donatsch, Ayar Alazzawi, Christophe Bruttin, and Amr Huber. 2014. Offline and online evaluation of news recommender systems at swissinfo. ch. In *Proceedings of the 8th ACM Conference on Recommender systems*. 169–176.

[19] Myron Glassman and Bruce McAfee. 1992. Integrating the personnel and marketing functions: The challenge of the 1990s. *Business Horizons* 35, 3 (1992), 52–59.

[20] Daniel Halbheer, Florian Stahl, Oded Koenigsberg, and Donald R Lehmann. 2014. Choosing a digital content strategy: How much should be free? *International Journal of Research in Marketing* 31, 2 (2014), 192–206.

[21] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (TIIS)* 5, 4 (2015), 1–19.

[22] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[23] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 5–53.

[24] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[25] Yujing Hu, Qing Da, Anxiang Zeng, Yang Yu, and Yinghui Xu. 2018. Reinforcement learning to rank in e-commerce search engine: Formalization, analysis, and application. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 368–377.

[26] Rolf Jagerman, Krisztian Balog, and Maarten De Rijke. 2018. Opensearch: lessons learned from an online evaluation campaign. *Journal of Data and Information Quality (JDIQ)* 10, 3 (2018), 1–15.

[27] Rolf Jagerman, Ilya Markov, and Maarten de Rijke. 2019. When people change their mind: Off-policy evaluation in non-stationary recommendation environments. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 447–455.

[28] Weiling Jiao, Hao Chen, and Yufei Yuan. 2020. Understanding users' dynamic behavior in a free trial of IT services: A three-stage model. *Information & Management* 57, 6 (2020), 103238.

[29] Junqi Jin, Chengru Song, Han Li, Kun Gai, Jun Wang, and Weinan Zhang. 2018. Real-time bidding with multi-agent reinforcement learning in display advertising. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 2193–2201.

[30] Thorsten Joachims, Dayne Freitag, Tom Mitchell, et al. 1997. Webwatcher: A tour guide for the world wide web. In *IJCAI (1)*. Citeseer, 770–777.

[31] Craig C Julian and Bashar Ramaseshan. 1994. The role of customer-contact personnel in the marketing of a retail bank' s services. *International Journal of Retail & Distribution Management* (1994).

[32] Deanna S Kempf and Robert E Smith. 1998. Consumer processing of product trial and the influence of prior advertising: A structural modeling approach. *Journal of Marketing Research* 35, 3 (1998), 325–338.

[33] Puja Khatri. 2006. Celebrity endorsement: A strategic promotion perspective. *Indian media studies journal* 1, 1 (2006), 25–37.

[34] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[35] Eunkyu Lee and Richard Staelin. 1997. Vertical strategic interaction: Implications for channel pricing strategy. *Marketing science* 16, 3 (1997), 185–207.

[36] Tao Lei, Yu Zhang, and Yoav Artzi. 2017. Training RNNs as Fast as CNNs. *CoRR* abs/1709.02755 (2017). arXiv:1709.02755 http://arxiv.org/abs/1709.02755

[37] Jingjing Li, Ke Lu, Zi Huang, and Heng Tao Shen. 2019. On both cold-start and long-tail recommendation with social data. *IEEE Transactions on Knowledge and Data Engineering* 33, 1 (2019), 194–208.

[38] Zhao Li, Junshuai Song, Shichang Hu, Shasha Ruan, Long Zhang, Zehong Hu, and Jun Gao. 2019. Fair: Fraud aware impression regulation system in large-scale real-time e-commerce search platform. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 1898–1903.

[39] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).

[40] Su Liu, Ye Chen, Hui Huang, Liang Xiao, and Xiaojun Hei. 2018. Towards smart educational recommendations with reinforcement learning in classroom. In *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*. IEEE, 1079–1084.

[41] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.

[42] Shamim Nemati, Mohammad M Ghassemi, and Gari D Clifford. 2016. Optimal medication dosing from suboptimal clinical examples: A deep reinforcement learning approach. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2978–2981.

[43] Baozhuang Niu, Haoran Yue, Huajiang Luo, and Weixin Shang. 2019. Pricing for newly-launched experience products: Free trial or not? *Transportation Research Part E: Logistics and Transportation Review* 126 (2019), 149–176.

[44] Aniruddh Raghu, Matthieu Komorowski, Imran Ahmed, Leo Celi, Peter Szolovits, and Marzyeh Ghassemi. 2017. Deep reinforcement learning for sepsis treatment. *arXiv preprint arXiv:1711.09602* (2017).

[45] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian Personalized Ranking from Implicit Feedback. *CoRR* abs/1205.2618 (2012). arXiv:1205.2618 http://arxiv.org/abs/1205.2618

[46] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[47] Venkatesh Shankar and Ruth N Bolton. 2004. An empirical analysis of determinants of retailer pricing strategy. *Marketing Science* 23, 1 (2004), 28–49.

[48] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature* 529, 7587 (2016), 484–489.

[49] Manohar Singh, Sheri Faircloth, and Ali Nejadmalayeri. 2005. Capital market impact of product marketing strategy: Evidence from the relationship between advertising expenses and cost of capital. *Journal of the Academy of Marketing Science* 33, 4 (2005), 432–444.

[50] Junshuai Song, Zhao Li, Zehong Hu, Yucheng Wu, Zhenpeng Li, Jian Li, and Jun Gao. 2020. Poisonrec: an adaptive data poisoning framework for attacking black-box recommender systems. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 157–168.

[51] Kai Sun, Meiyun Zuo, and Dong Kong. 2017. What can product trial offer?: The influence of product trial on Chinese consumers' attitude towards IT product. *International Journal of Asian Business and Information Management (IJABIM)* 8, 1 (2017), 24–37.

[52] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[53] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*. 1057–1063.

[54] Jiliang Tang, Huiji Gao, and Huan Liu. 2012. mTrust: Discerning multi-faceted trust in a connected world. In *Proceedings of the fifth ACM international conference on Web search and data mining*. 93–102.

[55] Zhiqing Tang, Weijia Jia, Xiaojie Zhou, Wenmian Yang, and Yongjian You. 2020. Representation and Reinforcement Learning for Task Scheduling in Edge Computing. *IEEE Transactions on Big Data* (2020), 1–1. https://doi.org/10.1109/TBDATA.2020.2990558

[56] Jia Wang, Jiannong Cao, Senzhang Wang, Zhongyu Yao, and Wengen Li. 2020. IRDA: Incremental Reinforcement Learning for Dynamic Resource Allocation. *IEEE Transactions on Big Data* (2020), 1–1. https://doi.org/10.1109/TBDATA.2020.2988273

[57] Ting Wang, Lih-Bin Oh, Kanliang Wang, and Yufei Yuan. 2013. User adoption and purchasing intention after free trial: an empirical study of mobile newspapers. *Information Systems and e-Business Management* 11, 2 (2013), 189–210.

[58] Wei Wang, Tao Tang, Feng Xia, Zhiguo Gong, Zhikui Chen, and Huan Liu. 2020. Collaborative Filtering with Network Representation Learning for Citation Recommendation. *IEEE Transactions on Big Data* (2020), 1–1. https://doi.org/10.1109/TBDATA.2020.3034976

[59] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.

[60] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8, 3 (1992), 229–256.

[61] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 346–353.

[62] Hongzhi Yin, Bin Cui, Jing Li, Junjie Yao, and Chen Chen. 2012. Challenging the Long Tail Recommendation. *Proc. VLDB Endow.* 5, 9 (2012), 896–907. https://doi.org/10.14778/2311906.2311916

[63] Xiangyu Zhao, Long Xia, Liang Zhang, Zhuoye Ding, Dawei Yin, and Jiliang Tang. 2018. Deep reinforcement learning for page-wise recommendations. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 95–103.

[64] Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Long Xia, Jiliang Tang, and Dawei Yin. 2018. Recommendations with Negative Feedback via Pairwise Deep Reinforcement Learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, Yike Guo and Faisal Farooq (Eds.). ACM, 1040–1048. https://doi.org/10.1145/3219819.3219886

[65] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. 2018. DRN: A Deep Reinforcement Learning Framework for News Recommendation. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, Pierre-Antoine Champin, Fabien Gandon, Mounia Lalmas, and Panagiotis G. Ipeirotis (Eds.). ACM, 167–176. https://doi.org/10.1145/3178876.3185994

[66] Hao Zhou, Geng Yang, Yang Xiang, Yunlu Bai, and Weiya Wang. 2021. A Lightweight Matrix Factorization for Recommendation with Local Differential Privacy in Big Data. *IEEE Transactions on Big Data* (2021), 1–1. https://doi.org/10.1109/TBDATA.2021.3139125

[67] Dong Hong Zhu and Ya Ping Chang. 2014. Investigating consumer attitude and intention toward free trials of technology-based services. *Computers in Human Behavior* 30 (2014), 328–334.

[68] Lixin Zou, Long Xia, Pan Du, Zhuo Zhang, Ting Bai, Weidong Liu, Jian-Yun Nie, and Dawei Yin. 2020. Pseudo Dyna-Q: A reinforcement learning framework for interactive recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 816–824.

[69] Shihao Zou, Zhonghua Li, Mohammad Akbari, Jun Wang, and Peng Zhang. 2019. Marlrank: Multi-agent reinforced learning to rank. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2073–2076.