


# Dynamic Target User Selection Model For Market Promotion with Multiple Stakeholders

Linxin Guo<sup>1</sup>, Shiqi Wang<sup>1</sup>, Min Gao <sup>1</sup>, and Chongming Gao<sup>2</sup>

<sup>1</sup> Chongqing University, Chongqing, China  
gaomin@ccqu.edu.cn

<sup>2</sup> University of Science and Technology of China

**Abstract.** While recommendation platforms present merchants with a vast and transparent sales avenue, they have inadvertently favored dominant merchants, often sidelining small-sized businesses. Addressing this challenge, platforms are deploying multifaceted market promotion strategies both to help merchants identify potential users and to spotlight emerging items for users. A crucial aspect of these strategies is the efficient selection of target users. By channeling resources towards the most responsive users, there's potential for a heightened return on marketing investments. In light of limited research in this domain, we put forth a tri-stakeholder considered user selection model with social networks (TriSUMS). This model recognizes the intertwined interests of three core stakeholders: merchants (items), platforms, and users. It harmonizes the objectives of these stakeholders through an integrated reward function and incorporates social networks to identify the prime target users for items of merchants adeptly. We validate TriSUMS using an exhaustive exposure user-item interaction dataset, assessed within a solid offline reinforcement learning framework.

**Keywords:** Market Promotion · Recommender System · Reinforcement Learning

## 1 Introduction

With the rapid development of information technology and the widespread application of big data technology, the Internet has penetrated into all aspects of human life. However, while technology enriches human life, it also brings out the problem of information overload. To solve the above problems, recommendation systems [2] have emerged. The recommendation systems use historical behavioral data to extract users' preferences and provide precise recommendations. It not only improves the accuracy of information propagation but also optimizes the user experience. Current mainstream personalized recommendation algorithms include content-based recommendations [14, 8], collaborative filtering-based recommendations [17, 16], and social network-based recommendations [21, 3].

In the current competitive market environment, users are the core of marketing activities. Merchants pay special attention to target users to increase item

sales, expand market share and enhance brand awareness. **Selecting appropriate target users for market promotion is the key link in the marketing process.** Nathan Fong et al. [4] found that targeted promotional activities based on personal purchase history can increase sales. Liu et al. [13] mentioned that the selection of target users in the advertising process usually takes into account the past behavior, identity, geographical location, and other attributes of consumers. Margaret et al. [1] found that brand familiarity will affect the effect of advertising repetition, so for brands familiar to users, the number of advertising repetitions can be higher.

The above research has shown that selecting appropriate target users for market promotion can benefit multiple stakeholders, i.e., merchants, platforms, and users. [19]. However, how to select appropriate users for market promotion is still under exploration. Most of these selection methods are based on heuristic rules and do not consider all stakeholders. Moreover, incompletely considering the interest of three stakeholders can lead to the collapse of the platform’s entire business ecosystem. As the very core of market promotion, the interest of merchants is the exposure rate of their products. However, increasing the exposure of promoted products may harm the interest of users who require an accurate recommendation list to overcome the information overload problem. The platform needs to balance the demand of both merchants and users. While for the platform itself, improving the diversity of recommendation lists can also help discover potential new merchants and attract corresponding new users to help the further development of the platform.

To this end, we propose a dynamic target user selection model TriSUMS (Tri-Stakeholder User selection Model with Social networks), for all these stakeholders. For merchants, TriSUMS prioritizes the exposure of their products to ensure they receive the exposure increase they require. For users, the model emphasizes tailoring recommendations according to their interests, ensuring that they receive content relevant to their historical preferences, thus assisting them in navigating through the vast sea of information. As for the platform’s longevity and growth, it places a strong emphasis on the diversity of the products showcased in the recommendation lists. Specifically, TriSUMS quantifies this balance through metrics like the frequency of an item’s appearance in recommendation lists, the alignment of user recommendation lists with their historical interests, and the overall diversity of the recommendation list. In TriSUMS, three reward functions are designed for every kind of stakeholder. And these reward functions are combined into an integrated reward function to guild the training process. TriSUMS learns user-selecting policies in a dynamic environment, which can select optimal target users for market promotion to maximize the integrated reward function of multiple stakeholders. The contributions of this paper can be summarized as follow:

- We propose a target user selection model TriSUMS that considers multiple stakeholders. By comprehensively considering the interests of merchants, platforms, and users in marketing scenarios with users’ social relationships, TriSUMS can increase the reward of the above-mentioned stakeholders.

- We construct a reliable simulation environment using a full exposure dataset and establish a robust offline reinforcement learning evaluation framework to assess user satisfaction when the user-selecting policy of TriSUMS is applied.
- We conduct extensive experiments to verify the effectiveness of the proposed model to validate the effectiveness of the model in improving the rewards of multiple stakeholders, demonstrating the model’s superior performance in a reliable evaluation framework.

## 2 Preliminaries

To learn user-selecting policies in a dynamic environment, reinforcement learning technology [20] is a good way to achieve this. In this section, we introduce reinforcement learning and its variants, offline reinforcement learning. We also give the problem formulation of our work in this section.

### 2.1 Reinforcement Learning

The problem of reinforcement learning is how agents make decisions in complex and uncertain environments to maximize cumulative rewards. Different from supervised learning, agents explore the environment through trial and error and constantly seek better strategies to obtain the maximum cumulative rewards. The interaction process between intelligent agents and the environment can be formalized as a five-tuple  $\langle A, S, P, R, \mu \rangle$ , including action space  $A$ , state space  $S$ , and state transition probability  $P : S \times S \times A \rightarrow [0, 1]$ , Reward Value  $R : S \times A \rightarrow \mathcal{R}$  and the discounted factor  $\mu \in [0, 1]$ .

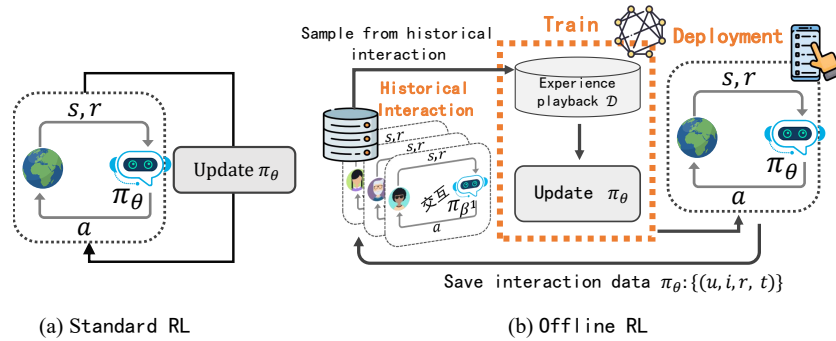
At the time  $t$ , the agent observes the environment state  $s_t \in S$  and takes action  $a_t \in A$  through the policy  $\pi$ . At the next time  $t + 1$ , the environment feeds back a reward  $r_t \in R$  and transports itself to a new state  $s_{t+1}$  through the state transition probability  $P$ . The agent constantly adjusts its policy  $\pi$  through the reward  $R$ , and steps into the next decision process. By repeating this process, the agent can get a trajectory  $(s_0, a_0, r_0, s_1, a_1, r_1, \dots, s_n, a_n, r_n)$ . The target of reinforcement learning is to find out a policy  $\pi$  that can maximize the cumulative reward  $G_t$ :

$$G_t = r_0 + \mu r_1 + \mu^2 r_2 + \dots + \mu^n r_n = \sum_{k=0}^{k=n} \mu^k r_k, \quad (1)$$

where  $\mu$  is the discount factor that is used to weaken the future reward. Especially, if  $\mu$  is close to 0, the agent focuses more on short-term reward, and if  $\mu$  is close to 1, the agent tends to increase the long-term cumulative reward.

### 2.2 Offline Reinforcement Learning

Conducting online reinforcement learning in real-life scenarios is significantly difficult, often facing high costs and risks. Plenty of application fields [6, 9] have demonstrated its risk. In recommendation, users need to constantly interact with



**Fig. 1.** The standard reinforcement learning and offline reinforcement learning

the agent. This process is unrealistic, as users do not have the patience to interact with an immature system.

To solve the above problems, a variant of reinforcement learning, i.e., offline reinforcement learning [10, 11], came into being. It requires agents to learn from fixed batches of offline history data without any real-time interaction with the environment. The problem that offline reinforcement learning focuses on is how to effectively use the massive offline data to obtain a strategy that maximizes the cumulative reward. Offline reinforcement learning samples from the experience playback pool  $\mathcal{D}$  and updates the strategy  $\pi_\theta$ . After offline training, the model is deployed to the online environment to verify its effect. Compare to standard reinforcement learning, offline reinforcement learning is safer due to the removal of high-frequency real-time interaction with the environment.

### 2.3 Problem Formulation

Let  $U$  be the set of users and  $I$  be the set of items.  $R \in \mathcal{R}^{|U| \times |I|}$  is the interaction where  $R_{ui} = 1$  indicates user  $u$  has interacted with item  $i$  and  $R_{ui} = 0$  indicates there is no interaction between  $u$  and  $i$ .  $S \in \mathcal{R}^{|U| \times |U|}$  is the social relationship where  $S_{uv} = 1$  indicates user  $u$  and  $v$  are friends and  $S_{uv} = 0$  indicates user  $u$  and  $v$  do not know each other.

Our goal is to learn a user-selecting policy  $\pi$  that can maximize the reward of merchants, users, and the platform. The policy  $\pi$  selects optimal users to interact with merchants' promotional items to simulate the market promotion process. After the establishment of these interactions, the recommendation lists for users are changed. To ensure the overall reward of the three stakeholders at the same time, the designed integrated reward function  $R_s$  is maximized during the training process.

## 3 Methodology

In this section, we consider the interests of three stakeholders and propose a dynamic target user selection model TriSUMS (Tri-Stackholder User selection

Model with Social networks). TriSUMS considers not only the reward of merchants but also the reward of the platform and users. The framework introduces an offline reinforcement learning algorithm to train the interactive recommendation model and builds a reliable simulation environment based on the latest KuaiRec [5] dataset and the classic LastFM dataset to evaluate the effectiveness of the model.

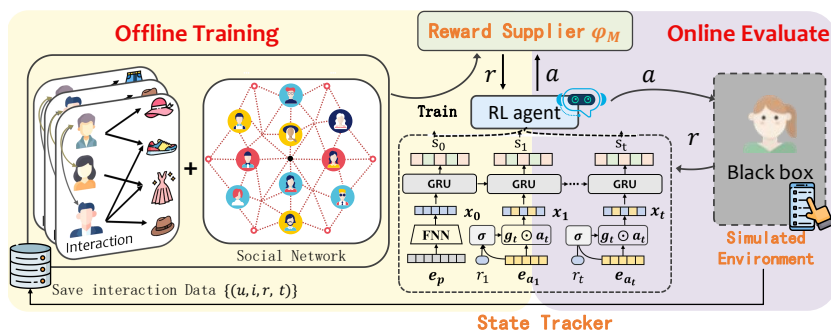


Fig. 2. The framework of TriSUMS

### 3.1 Overall Framework

TriSUMS mainly includes four key modules: a reward supplier, a reinforcement learning agent (RL Agent), a state tracker, and a simulated environment. As shown in Fig. 2, TriSUMS first utilizes offline interactive data  $\{(u, i, r, t)\}$ , a set of quadruples that contain user  $u$  interacted with item  $i$  at time  $t$  and user's social relationship  $r$ , to train the strategy. Then TriSUMS tests the impact of the model in a simulated environment. The details of the four modules are as follows:

The reward supplier is a recommendation model. Its recommendation performance can reflect the effect of market promotion. We use the interaction  $R$  and social relationship  $S$  to build the adjacent matrix, which is shown as follows:

$$A = \begin{pmatrix} S & R \\ R^T & 0 \end{pmatrix}, \quad (2)$$

and the LightGCN [7] with the above adjacent matrix is used as the reward supplier. The reward supplier provides reward signals in the dynamic interactive marketing process and evaluates the impact of user selection.

The state tracker is based on a GRU model, which can automatically extract the most relevant information for current market promotion from the vectors representing item attributes  $e_i$  and historical target user vectors  $\{e_{a_1}, \dots, e_{a_t}\}$ .

The RL agent interacts with the reward supplier. During this interaction, the reward supplier is responsible for providing timely and accurate reward signals to

the RL Agent. The RL Agent here can be any reinforcement learning algorithm, such as PPO [15], DDPG [12].

The simulated environment is used to simulate a real business environment, which is a black box that can return user feedback for model evaluation when the algorithm selects the target user.

### 3.2 Construction of Multi-Stakeholder Reward Function

Market promotion involves multiple stakeholders, including merchants, users, and platforms. It is necessary to balance the interests of them. In this section, we design corresponding reward functions for each stakeholder and then synthesized them to form a reward function:

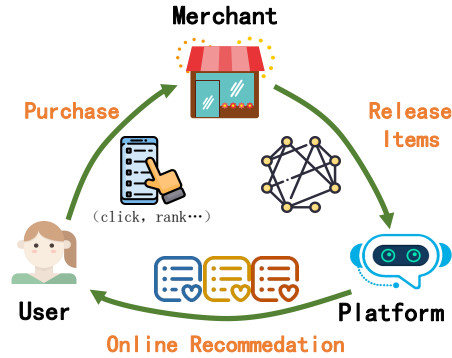


Fig. 3. The three stakeholders in market promotion

**Reward Function for Merchants:** Merchants are the very core of market promotion since the promotion is always launched by them. Merchants focus on the exposure of goods and expect to increase the exposure of goods through market promotion, thus increasing sales revenue. The direct way to measure the effect of market promotion is how many items of the merchants are recommended by the recommender system. Therefore, we set the reward function for the merchants as the change in the number of items displayed on the recommendation page:

$$\mathcal{R}_m(s_t, a_t) = \frac{Exp(I_p^t) - Exp(I_p^{t-1})}{Exp(I_p^{t-1})}, \quad (3)$$

where  $Exp(I_p^t)$  indicates the number of promotional items displayed on the recommendation page at time  $t$ .

**Reward Function for Users:** Users are the receivers of market promotion. Moreover, it is evident that promoting appropriate items are acceptable for users, and users may feel unhappy when promoted improper items to them. Since recommendation metrics can effectively predict users' interests. Considering the

change in recommendation loss can measure how users feel when the market promotion is adopted, we use the loss as the user reward function:

$$\mathcal{R}_u(s_t, a_t) = \frac{L_t - L_{t-1}}{L_{t-1}}, \quad (4)$$

where  $L_t$  is the loss of the LightGCN integrated with social networks when selecting the target users for the promotion.

**Reward Function for Platform:** The recommendation platform is the basis of online transactions, which connects users and merchants. The platform offers merchants a place where merchants can display their products to numerous users, and the platform also uses recommendation algorithms to filter appropriate products for users and reduce the information overload problem. Firstly, the platform needs to balance the interest of users and merchants, providing them with a good experience, this part is included in the reward function for merchants and users. However, traditional recommendation algorithms often focus on popular products, while long-tail items with relatively low sales but of a wide variety are ignored, leading to a monotonous recommended list, and finally result in damaging the overall ecology. Also, the recommendation platform has the following advantages in recommending long tail items: firstly, long tail items can meet users' diverse needs for products, thereby improving user stickiness and satisfaction. Secondly, although the sales of individual products are relatively low, long-tail items can bring more business opportunities. Finally, recommending long-tail items can also help the platform optimize product inventory and reduce warehousing costs. The long-tail item coverage is defined as the proportion of the long-tail items recommended to all items. Therefore, the change in long-tail item coverage is used as the reward function for the platform:

$$Diversity = \frac{\cup_{u \in U}(L_u)}{|V|}, \quad (5)$$

$$\mathcal{R}_p(s_t, a_t) = \frac{Diversity_t - Diversity_{t-1}}{Diversity_{t-1}}. \quad (6)$$

**Integrated Reward Function for Multiple Stakeholders:** In practical scenarios, there are often contradictions in the interests of these stakeholders. Merchants hope to maximize the exposure of their own products and attract more users to purchase, but this may lead to a waste of platform resources and user dissatisfaction; The platform hopes to enhance its uniqueness and diversity by increasing the exposure rate of long-tail products, but this may affect the exposure rate of mainstream products and the profits of corresponding merchants; Users hope to purchase their favorite products and enjoy discounts, but this may lead to waste of platform resources and reduced profits for merchants. To solve these contradictions, we integrate three rewards with a weighted summation, achieving balance and measurement of multi-party interests:

$$R_s(s_t, a_t) = \alpha R_m(s_t, a_t) + \beta R_u(s_t, a_t) + \gamma R_t(s_t, a_t). \quad (7)$$

The integrated reward function can balance the interests of merchants, platforms, and users during the optimizing process, improving the exposure of promoted items, coverage of long-tail items, user experience, and other indicators, thus achieving a win-win situation for multiple stakeholders. In addition, by adjusting the weight parameters  $\alpha$ ,  $\beta$ , and  $\gamma$ , the integrated reward function can also adjust to different market promotion scenarios with different benefit allocations, further improving promotion effectiveness.

### 3.3 Offline Reinforcement Learning-Driven Framework

In this section, we propose a dynamic target user selection model TriSUMS that takes multiple stakeholders into account. Fig. 2 shows the process of selecting target users at different times in the model. The key variables involved in the model are as follows:

**Action:**  $a_t$  represents the action taken by the interactive strategy at the moment  $t$ . In this section, the action selects a user  $u$ , so the representation vector  $e_a$  of an action  $a$  and the standard vector of the user  $e_u$  selected by the action are equivalent, i.e.,  $e_a = e_u$ .

**Status:**  $s_t \in \mathbf{R}^{d_s}$  indicates the interaction state at  $t$ , which provide overall historical information for agent.  $s_t$  includes the representation vector  $e_i$  of the interactive information of the item and the user information that has been selected for the item in the whole interactive trajectory process  $\{e_{a_1}, \dots, e_{a_t}\}$ .  $s_t$ .

**Reward Signal:**  $r_t$  represents the feedback signal provided by the reward provider  $\varphi_M$  after the policy selection action  $a_t$  at time  $t$ , which is calculated through the reward function of Eq. 7.

**Policy network:**  $\pi_\theta = \pi_\theta(a_t|s_t)$  selects actions  $a_t$  based on the current state  $s_t$ . It takes state  $s_t$  as input and outputs a probability distribution. The probability of action  $a_t$  being selected is as follows:

$$\pi_\theta(a_t|s_t) = \text{ReLu}(\sigma(W_s^T s_t + b_t)), \quad (8)$$

where  $\sigma$  represents the nonlinear activation function,  $W_s^T \in \mathcal{R}^{d_s \times d_a}$  and  $b_t \in \mathcal{R}^{d_a}$  represent the weight matrix and bias, which are learned through the training process.

### 3.4 Proximal Policy Optimization

We use a variant of the PG algorithm - Proximal Policy Optimization (PPO) [15] to train the model. The PPO algorithm constrains the update amplitude by limiting the distance of new and old policies between each update step, solving the problem of the PG algorithm that may cause significant changes and unstable training in one step. The objective function of the PPO algorithm contains two parts, one is to improve the performance of the policy, other is to ensure the stability of the training process. The objective function of the PPO algorithm is defined as:

$$\mathbf{E}_t[\min(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \hat{A}_t, \text{clip}(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}, 1 - \epsilon, 1 + \epsilon) \hat{A}_t)], \quad (9)$$



where the first term in the objective function is to improve policy performance. The second term uses the pruning function  $clip(\cdot)$  to limit the amplitude of policy updates.  $\epsilon$  is a hyperparameter that limits the maximum update amplitude of the policy parameter  $\theta$  in a single step. To achieve this, the  $clip(x, a, b)$  function limits the value of  $x$  to the interval  $[a, b]$ . If  $x$  is smaller than  $a$ , then the output of  $clip(x, a, b)$  is  $a$ , if  $x$  is bigger than  $b$  the output is  $b$ , and if  $x$  is between  $a$  and  $b$  the output is  $x$ . The  $\theta_{old}$  represents the old version of the policy parameter  $\theta$ . Thus, the update amplitude of  $\theta$  is limited in  $\epsilon$ .  $\hat{A}_t$  is the function of cumulative reward:

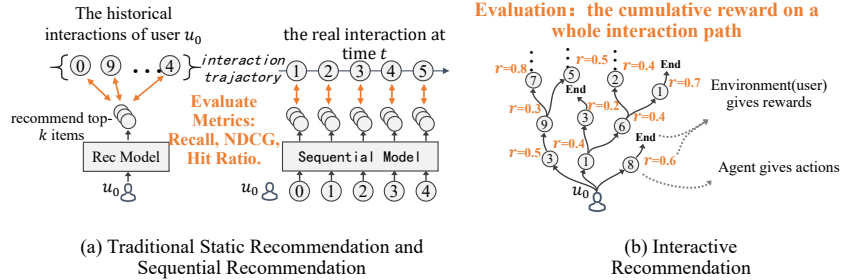
$$\hat{A}_t = \hat{A}_t^{GAE(\mu, \lambda)} = \sum_{l=0}^{\infty} (\mu\lambda)^l \delta_{t+l}^V, \quad (10)$$

where  $\lambda$  is a hyperparameter that balances bias and variance.  $\delta_t^V = r_t + \mu V(s_t + 1) - V(s_t)$  represents the residual of the value function  $V$ ,  $\mu$  is the discount factor. The value function  $V$  is as follows:

$$V(s_t) = V^{\pi_{\theta}, \mu}(s_t) = \mathbf{E}_{s_{t+1}:\infty, a_t:\infty} \left[ \sum_{l=0}^{\infty} \mu^l r_{t+l} \right]. \quad (11)$$

### 3.5 Evaluation Framework

After the training process, we need to simulate the online evaluation. This process aims to analyze the impact on users in real scenarios. Therefore, it is necessary to develop a reliable and robust evaluation framework to accurately evaluate model performance. Fig. 4 shows the evaluation methods of traditional static recommendation, sequential recommendation, and interactive recommendation.



**Fig. 4.** Evaluation methods of traditional and interactive recommendation

Fig. 4(a) shows the eval methods of traditional static recommendation and sequential recommendation. Among them, traditional static recommendation algorithms recommend a list of products that users may be interested in based on their interaction history. To evaluate the accuracy of recommendations, this product is generally compared with the real user interaction set in the test set,

and evaluation indicators such as Recall, normalized discount cumulative gain (NDCG), and Hit Ratio are used to quantitatively analyze the recommendation effect. However, these evaluation methods that consider the products in the test set as standard answers do not conform to real recommendation scenarios. Because the interaction between users and products in the test set does not accurately reflect users’ true preferences, it may only stem from curiosity or herd mentality. Meanwhile, the fact that one user has not interacted with a certain product does not mean that the user is uninterested in the product, it may be because the user has not yet discovered such a product. Also, in sequential recommendation methods, users’ historical interactive products are typically modeled as sequences or trajectories with temporal characteristics. The goal is to predict the products that users may interact with at any given time. This method of using historical data as evaluation criteria is also not in line with actual recommendation scenarios, since fixed sequences or trajectories ignore the probability that users may interact with other items.

In actual recommendation scenarios, when users browse products on the recommendation platform, they may have no idea what they want. Meanwhile, they will provide feedback based on the platform’s recommendation content and find the products they truly want to purchase through continuous interaction. If a good experience is obtained during this interaction process, users will continue to use the recommendation platform. Compared to static metrics such as accuracy and recall, recommendation platforms pay more attention to the long-term improvement of user experience satisfaction. These long-term metrics are often difficult to be covered and captured by traditional static and sequential recommendations modeling.

As shown in Fig. 4(b), in interactive recommendation scenarios, the interactions between users and agents are real-time rather than special history trajectories, presenting a divergent trend. Evaluation in interactive scenarios requires recording the cumulative reward of all these paths. This section uses the KuaiRec dataset[5] released by Kwai and the team of China University of Science and Technology to build a reliable evaluation framework and evaluate the impact of TriSUMS model on user satisfaction in real online recommendation scenarios. Compared with traditional highly sparse recommendation datasets, the KuaiRec dataset observation data contains a user-product interaction matrix with a density up to 99.6%, which can provide feedback for each action taken by the agent to calculate the cumulative satisfaction of users. The full exposure dataset as a simulation environment can provide strong support for the evaluation.

## 4 Experiments

This section introduces experimental design and analysis of experimental results to verify the effectiveness of the methods proposed in market promotion scenarios, as well as the effectiveness of social networks in improving recommendation performance. Specifically, we conduct experiments on two public datasets to analyze the following research questions (RQs):

- **RQ1:** How does the TriSUMS model improve the multi-stakeholder rewards compared with the static and collaborative filtering-based user selection strategy?
- **RQ2:** How does the TriSUMS model perform in the evaluation of the simulation environment compared with the static target user selection strategy and the user selection strategy based on collaborative filtering?
- **RQ3:** How do the user social relationships impact Precision, Recall, and NDCG in market promotion scenarios?

#### 4.1 Datasets

We use two public datasets containing user social relationships, LastFM [18], and KuaiRec [5], for experiments. As shown in Table 1, the LastFM dataset is a commonly used dataset for music recommendation, containing interaction records of 1,892 users and 17,632 items. Since LastFM is highly sparse and cannot provide data support for the simulation environment in the model evaluation phase, matrix factorization is used to fill in the missing values.

**Table 1.** The statistics of the datasets

dataset	train/test	user	item	interaction	density	social relation
LastFM	train	1,892	4,489	42,135	0.62%	25,434
	test	1,858	3,285	78,286,830	100%	
KuaiRec	train	7,176	10,728	12,530,806	16.28%	670
	test	1,411	3,327	4,676,570	99.6%	

As shown in Fig. 5, the KuaiRec dataset consists of a sparse large matrix and a dense small matrix. The small matrix with red dashed lines contains almost no missing values for user video interactions, with a density of 99.6%. The missing 0.4% interactions are due to some users having blocklisted some video makers, and the platform cannot expose such videos to these users. We can treat these missing interactions as uninterest. This full exposure matrix can provide accurate and comprehensive feedback for the model evaluation stage. The blue dashed part is a large matrix with an interaction density of 16.3%, used for offline training of the model.

#### 4.2 Baselines

The existing user-selecting policies are mainly based on historical behavior such as purchase [4] and brand familiarity [1], and there is no user selection algorithm for market promotion. To ensure the effectiveness of the experiment. Five static selection strategies are used, and two machine learning-based comparison methods are designed. Seven baselines include Random selection, Active first, Inactive first, High Rating first, Low Rating first, Item CF, and User CF. The details of the seven methods are shown as follows:

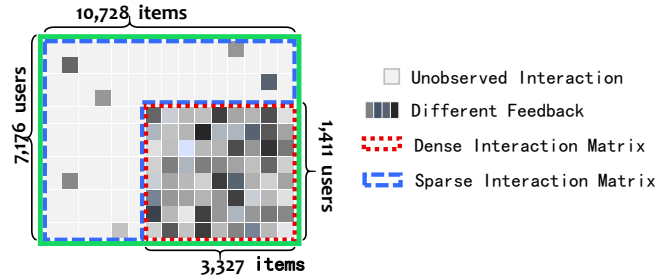


Fig. 5. The fully-observed dataset KuaiRec

**Random Selection:** Randomly select a target user and establish a connection with the promotional item set. The advantage of this method is that it is simple and easy to implement, but it may not be optimized for specific user groups, resulting in unstable promotion results.

**Active First:** This method sorts the user interaction volume (i.e. historical purchase data) and randomly selects target users from the top 30% of active users. Active users are more likely to notice promotional items, which may increase their exposure rate. However, this approach may overly focus on active users, leading to neglecting the needs of other user groups.

**Inactive First:** Contrary to the high activity priority selection method, this method randomly selects target users from the bottom 30% of non-active users. The purpose of this method is to avoid user churn and expand the audience for promoting the item. However, this method may result in less effective promotion, as inactive users may not be interested in new items.

**High Rating First:** This method calculates the average rating of users on all interactive items in the recommendation dataset and randomly selects target users from the top 30% of high-scoring users. High-scoring users may be more attracted to promotional items, increasing their exposure rate. However, this method may overlook the needs of low-scoring users and limit the scope of promotion effectiveness.

**Low Rating First:** Contrary to the high-scoring priority selection method, this method randomly selects target users from low-scoring users who rank in the bottom 30% of the score. This method attempts to expand the audience range of promotional items but may face the problem of low-rated users lacking interest in promoting the items.

**Item CF:** This method targets users who have purchased similar promotional items by analyzing and evaluating the similarity between items. This method helps find users interested in promoting the item, thereby increasing exposure. But this method may fail to identify potential new user groups.

**User CF:** By analyzing and evaluating the similarity between users, this method selects users similar to those who have already purchased promotional items as the target users. This method attempts to identify potentially interested users through user similarity, thereby increasing the exposure of promotional

items. However, this method may be limited by the accuracy of calculating the similarity between users and may overlook potential user groups that have not yet been discovered.

### 4.3 Evaluation Metrics

Considering that the goal of TriSUMS is to balance the interests and needs of merchants, users, and platforms, we selected three evaluation metrics: **product exposure**, **recommendation accuracy**, and **recommendation coverage**. Product exposure reflects merchants’ demand for product promotion, recommendation accuracy reflects users’ demand for personalized recommendations, and recommendation coverage reflects the platform’s demand for expanding recommendation scope.

In addition, we also use three common evaluation metrics for recommendation systems: *Precision@k*, *Recall@k*, and *NDCG@k* to measure the impact of social networks on recommendation performance. *Precision@k* is the ratio of the number of correctly predicted items in the recommendation results to the length of the recommendation list. It measures how many items on the recommendation list are truly of interest to users. *Recall@k* refers to the ratio of the correct number of recommended items to the number of all items that should be recommended. It measures how many items that users are interested in are recommended. *NDCG@k* considers the ranking of items and evaluates the accuracy of recommended item ranking.

### 4.4 Parameters Settings

The weights  $\alpha$ ,  $\beta$ , and  $\gamma$  of the reward function in Eq. 7 are set to 0.8, 0.1, and 0.1, respectively. Specifically, merchants are the direct beneficiaries and main supporters of market promotions, with the aim of increasing product exposure. Therefore, the interests of merchants should receive the greatest attention in the reward function, with a weight set at 0.8. As the strategy implementer of market promotions, the platform needs to ensure that the strategy implementation process does not affect the platform’s own benefits. Therefore, the platform interests should be included in the weight setting, with a weight of 0.1. Users are also an essential part of market promotions, as they bring sales and profits by purchasing products. Therefore, during market promotions, it is necessary to ensure that users can obtain a diverse recommendation list with a weight of 0.1.

In the experiment, the model selects 100 target users (i.e. round length  $n$ ) and establishes interaction with 1% of promotional items (i.e. the promotional item  $|I_p|$ ). The length of the recommended list  $k$  is 10, and the discount factor  $\mu$  is 0.9. The optimizer is Adam, and the initial learning rate is 0.005.

### 4.5 Overall Performance (RQ1)

Fig. 6 shows the experimental results of eight methods on three evaluation metrics: product exposure, recommendation accuracy, and recommendation coverage. Observations can lead to the following conclusion:

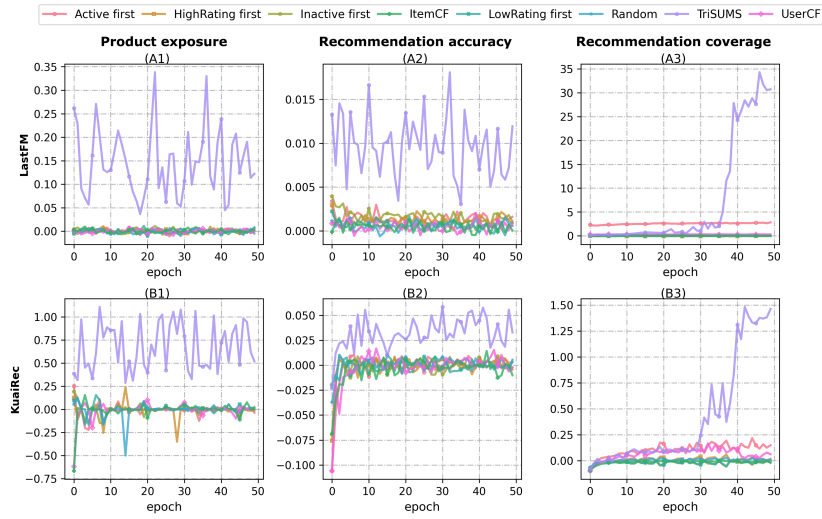


Fig. 6. Overall Performance

- TriSUMS outperformed the baseline model in all three evaluation metrics on two datasets, indicating that the TriSUMS has better overall performance in meeting user needs, improving platform revenue, and promoting the overall development of recommendation platforms.
- In terms of product exposure, the high activity priority selection method and the high score priority selection method perform relatively well. The performance of low activity priority selection and low rating priority selection methods is poor, mainly due to users with lower participation and low rating tendency, whose interest preferences are often vague and, therefore, not suitable as the target user group for promotional activities. The random selection method performs the worst because it does not utilize any information to optimize the selection strategy.
- In terms of recommendation accuracy, UserCF and ItemCF perform relatively well, due to the algorithm based on collaborative filtering fully mining the similarity information between users and products. At the same time, static strategies such as Active First, Inactive First, High Rating First, Low Rating First, and Random perform relatively poorly.
- In terms of recommendation coverage, the high activity priority selection method performs well, mainly due to frequent interaction between active users and recommendation platforms, as well as rich behavioral data. The interests and preferences of active users are more accurately captured, making them suitable target user groups for promotional activities.

In addition, it can be observed that the TriSUMS shows significant fluctuations in the result curves on all three metrics. There are two main reasons for this phenomenon: 1) Reinforcement learning needs to balance the exploration

of unknown states and behaviors with the use of known information. 2) Reinforcement learning usually relies on delayed rewards. However, the TriSUMS algorithm achieved better performance in all three metrics in the later stage.



Fig. 7. Comparison of evaluation results in simulation environment

#### 4.6 Online Reward Evaluation (RQ2)

Figure 7 shows the experimental results of the target user selection model TriSUMS and seven baseline models proposed in this section on the LastFM and KuaiRec datasets. The online reward (i.e., the values in the dense matrix) can reflect how satisfied users are. The horizontal axis epoch represents the number of test rounds, and the vertical axis represents online reward. We can find TriSUMS performs significantly better than the baseline model on both datasets, which means that TriSUMS can meet users’ requirements in a dynamic environment.

In the LastFM dataset, the performance of the seven baseline models is relatively close. The online reward fluctuates between 49.98 and 50.01 because the baseline models cannot fully capture the dynamic interaction relationship between users and items. The online reward of our proposed TriSUMS fluctuates between 50.03 and 50.46.

In the KuaiRec dataset, Item CF performs well for its ability to effectively mine the similarity information between users and items. Meanwhile, the effect of the high rating first method (online reward fluctuates around 100) is significantly better than the low rating first method (online reward below 80) because users who tend to give high ratings to products are more likely to generate positive feedback. The performance of active first, inactive first, random, and user CF is similar. Their online reward value fluctuates between 81 and 90. It is worth noting that after training for a period of time, our TriSUMS model has an online reward above 140. This can be attributed to the advantages of rein-

forcement learning-based methods in capturing dynamic environmental changes more effectively and focusing on long-term benefits.

#### 4.7 Ablation Study (RQ3)

To verify the role of user social relationships in improving the effectiveness of the model, this study designed an ablation experiment. The experiment compared the performance of the TriSUMS model and the model without social relationships,  $\text{TriSUMS}^{w/oS}$ , on the metrics of  $\text{Precision@10}$ ,  $\text{Recall@10}$ , and  $\text{NDCG@10}$ .

**Table 2.** The comparison of two variants of TriSUMS

Dataset Metric	KuaiRec			LastFM		
	Precision@10	Recall@10	NDCG@10	Precision@10	Recall@10	NDCG@10
$\text{TriSUMS}^{w/oS}$	0.2528	0.0132	0.2315	0.0752	0.2679	0.2096
TriSuMS	0.2571	0.0136	0.2378	0.0776	0.2768	0.2141
improve	1.70%	3.03%	2.72%	3.25%	3.28%	2.15%

As shown in Table 2, the TriSUMS model with extra social relationships achieves 1.70%, 3.03%, and 2.72% improvements in the KuaiRec dataset, as well as 3.25%, 3.28%, and 2.15% improvements in LastFM dataset, compares to the  $\text{TriSUMS}^{w/oS}$  model. This indicates that after adding user social relationships, the TriSUMS model can better capture user interests.

## 5 Conclusion

In this work, we introduce the dynamic selection model of TriSUMS. It considers the social relations of users and three major stakeholders in the market promotion process - merchants, platforms, and users, respectively. While improving the exposure of items, TriSUMS takes into account the accuracy and diversity of recommendations to meet the needs of different stakeholders. We utilize a full exposure dataset to construct a reliable simulation environment for evaluating the impact of the model on user satisfaction. The experimental results show that the TriSUMS performs better in improving user experience and other metrics compared to other models. This is mainly due to the following reasons: (1) Reinforcement learning usually focuses more on long-term rewards throughout the decision-making process. This section designs reward functions for multiple stakeholders to guide strategy updates to maximize cumulative benefits. (2) Reinforcement learning methods continuously explore the location environment during the learning process, which is more adaptable to changing new scenarios and adjust strategies adaptively compared to fixed selection strategies.



## References

1. Campbell, M.C., Keller, K.L.: Brand familiarity and advertising repetition effects. *Journal of consumer research* **30**(2), 292–304 (2003)
2. Covington, P., Adams, J., Sargin, E.: Deep neural networks for youtube recommendations. In: *Proceedings of the 10th ACM conference on recommender systems*. pp. 191–198 (2016)
3. Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J., Yin, D.: Graph neural networks for social recommendation. In: *The world wide web conference*. pp. 417–426 (2019)
4. Fong, N., Zhang, Y., Luo, X., Wang, X.: Targeted promotions on an e-book platform: Crowding out, heterogeneity, and opportunity costs. *Journal of Marketing Research* **56**(2), 310–323 (2019)
5. Gao, C., Li, S., Lei, W., Li, B., Jiang, P., Chen, J., He, X., Mao, J., Chua, T.S.: KuaiRec: A fully-observed dataset for recommender systems. *arXiv preprint arXiv:2202.10842* (2022)
6. Gu, S., Holly, E., Lillicrap, T., Levine, S.: Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In: *2017 IEEE international conference on robotics and automation (ICRA)*. pp. 3389–3396. IEEE (2017)
7. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: LightGCN: Simplifying and powering graph convolution network for recommendation. In: *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. pp. 639–648 (2020)
8. Javed, U., Shaukat, K., Hameed, I.A., Iqbal, F., Alam, T.M., Luo, S.: A review of content-based and context-based recommendation systems. *International Journal of Emerging Technologies in Learning (iJET)* **16**(3), 274–306 (2021)
9. Kiran, B.R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A.A., Yogamani, S., Pérez, P.: Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems* **23**(6), 4909–4926 (2021)
10. Lange, S., Gabel, T., Riedmiller, M.: Batch reinforcement learning. *Reinforcement learning: State-of-the-art* pp. 45–73 (2012)
11. Levine, S., Kumar, A., Tucker, G., Fu, J.: Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643* (2020)
12. Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D.: Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015)
13. Liu-Thompkins, Y.: A decade of online advertising research: What we learned and what we need to know. *Journal of advertising* **48**(1), 1–13 (2019)
14. Lops, P., De Gemmis, M., Semeraro, G.: Content-based recommender systems: State of the art and trends. *Recommender systems handbook* pp. 73–105 (2011)
15. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017)
16. Shen, J., Zhou, T., Chen, L.: Collaborative filtering-based recommendation system for big data. *International Journal of Computational Science and Engineering* **21**(2), 219–225 (2020)
17. Suganeshwari, G., Syed Ibrahim, S.: A survey on collaborative filtering based recommendation system. In: *Proceedings of the 3rd international symposium on big data and cloud computing challenges (ISBCC-16’)*. pp. 503–518. Springer (2016)
18. Tang, J., Gao, H., Liu, H.: mTrust: Discerning multi-faceted trust in a connected world. In: *Proceedings of the fifth ACM international conference on Web search and data mining*. pp. 93–102 (2012)

19. Wang, S., Gao, C., Gao, M., Yu, J., Wang, Z., Yin, H.: Who are the best adopters? user selection model for free trial item promotion. *IEEE Transactions on Big Data* **9**(2), 746–757 (2023). <https://doi.org/10.1109/TBDATA.2022.3205334>
20. Wiering, M.A., Van Otterlo, M.: Reinforcement learning. *Adaptation, learning, and optimization* **12**(3), 729 (2012)
21. Zhao, T., McAuley, J., King, I.: Leveraging social connections to improve personalized ranking for collaborative filtering. In: *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*. pp. 261–270 (2014)