BARSA: Budget-Aware Recommendation System Attack

Writen by AI class, XMU
Yihang Zheng 36920221153153[1], Yang Li 36920221153090[1],
Zijia Wang 23020221154176[2], Longfei Mei 23020221154171[2]

[1] Artificial Intelligence Major, School of Informatics, Xiamen University
[2] Computer Science Major, School of Informatics, Xiamen University

Abstract

Selecting the most suitable and cost-effective weblebrities for product promotion and marketing is a major need for every merchant. In this paper, we model the weblebrity selection problem as a budget-aware recommendation system tor attack problem and propose a general framework CASA, which treats weblebrities as attack samples, models the weblebrity recommendation problem as a black-box attack process on an NN-based model, and uses the method based on perturbed sample points related to the interpretability of black-box models to model the influence of attack sample points, so as to filter the most influential and cost-effective attack sample points.

In this work, we (i) explain how we model the problem as a complete recommendation system tor attack problem. (ii) use a perturbed sample point-based approach to obtain the influence of the training samples of a black-boxed NN-based recommendation model. (iii) Integrating user social relationships to aid in improving the recommendation results of the black-box recommendation model to make the recommendation results more relevant to the problem context (iv) Employing suitable approximation algorithms in multiple modules to reduce the model time complexity.

The experimental results show that our model can well filter the most cost-effective weblebrities list and is applicable to all NN-based recommendation models.

1 Introduction

Since 2019, the trend of "weblebrities live with goods recommendation" has swept the Internet e-commerce platform. According to the research report on China's live e-commerce industry in 2021 released by Ariadne Consulting, by the end of 2020, the size of China's live e-commerce users has reached 388 million people, and nearly two-thirds of them will watch and then make purchase.

Recommendation system is an information platform closely related to the effect of weblebrities goods recommendation. The recommendation system is an important part of the major online service platforms and information systems in the Internet era, which effectively completes the connection between users and information by looking for their personalized characteristics and needs, so as to recommend items of interest to users and help them get out of the dilemma of information overload(Covington, Adams, and Sargin 2016).

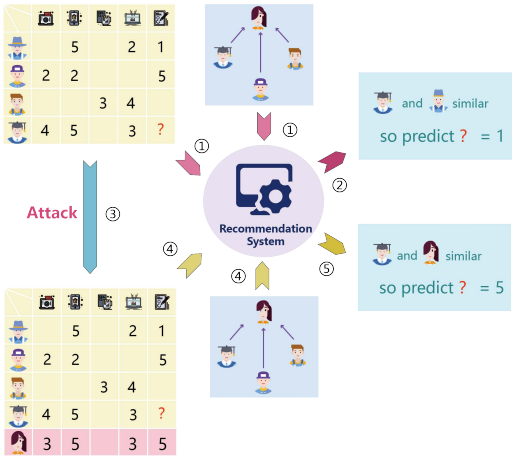

Figure 1: The weblebrity bandwagon is regarded as a recommendation system tor attack Before the attack, the recommendation system considers user 4 (Dr.) to be most similar to user 1 (Gentleman), and thus gives a prediction of 1 point for the target item; after the attack, the recommendation system considers user 4 (Dr.) to be most similar to user 5 (Weblebrity), and the social trust graph shows that user 4 (Dr.) trusts user 5 (Weblebrity), and thus gives a prediction of 5 points for the target item.

From the perspective of recommendation system, the "weblebrities live streaming recommendation" can be regarded as a kind of recommendation system torrent attack. A recommendation system shilling attack is an attack in which an attacker uses some strategies to generate artificially set fake user data and inject it into the recommendation system in order to increase the recommendation rate of his product (i.e., the target of the attack). Figure1 shows an example of a recommendation system suffering from a trust attack, resulting in skewed recommendation results, using Netflix recommendation as a scenario.

The most important issue for merchants in the scenario of "weblebrities live recommendation", is how to select the right weblebrities to maximize the target product recommendation rate under the above constraints, considering that different weblebrities have different social influences and budgets. To solve this kind of problem, this paper proposes a

general framework CASA, which firstly treats weblebrities as fake users that can be set artificially, and the products recommended by weblebrities can be regarded as attack targets; then, this paper models the recommendation system as a black-box NN-based model considering social network graph relations(Li et al. 2021), and models the weblebrities recommendation problem as a black-box attack on the NN-based model. process; then, the perturbation training point-based method(Koh and Liang 2017) proposed by Pang in the field of black box model interpretability models the influence of attack sample points, so as to filter out the most influential cost-effective attack sample points. Specifically, the model consists of four modules: recommendation system module, attack generation module, weblebrities influence evaluation module, and budget-aware weblebrities selection module.The functions implemented in each of these modules are as follows:

(1) In the attack generation module, we adopt the perspective of recommendation system to attack to construct the attack data and use the random attack to generate the shilling attack data and inject it into the recommendation system module.

(2) In the recommendation system module, we integrate the user social network relationship and uses SPEX 3, a social network recommendation system framework that combines the principle of path prediction based on the attention mechanism graph, to model the social influence of weblebrities, so as to assist the traditional recommendation task to give recommendation results and make the recommendation results and the overall framework closer to the realistic context.

(3) In the weblebrity influence evaluation module, we model the recommendation system model as a black box model and uses a method based on perturbed sample points from the black box model interpretability work to quantify the influence of individual weblebrities. This approach makes the framework compatible with arbitrary unknown recommendation systems.

(4) In the budget-aware weblebrities selection module, this paper adopts Box-Cox transformation to Gaussianize the data distribution of each weblebrity measure obtained, so that it has a statistically significant and interpretable linear relationship. And a greedy algorithm is used to give a list of the most suitable weblebrity budgets. In addition, an approximate algorithm framework is evaluated in this paper to solve the problem of high time complexity of the greedy algorithm.

2 Related Work

2.1 Recommendation System

Most of the traditional recommendation system algorithms are based on the idea of collaborative filtering(Goldberg et al. 1992), which can be further divided into two categories, namely content-based collaborative filtering algorithms and model-based collaborative filtering algorithms. Content-based collaborative filtering analyzes the interaction data between users and goods, finds out the users or goods that are most similar to the target users or goods (i.e., neighboring users/neighboring goods) based on the idea of nearest neighbor, and then recommends the target users according to the hobbies of neighboring users/-goods, whose representative algorithms are UserKNN and ItemKNN(Sarwar et al. 2001). Model-based algorithms take the user-goods interaction data as training data and use machine learning methods to build prediction models (Breese, Heckerman, and Kadie 2013), such as the matrix decomposition model(Koren, Bell, and Volinsky 2009) proposed by Koren in 2009, which decomposes and maps the user-goods interaction matrix into two low-dimensional joint hidden factor spaces, corresponding to user preferences and goods, respectively. features, and the BPR model(Rendle et al. 2009) proposed by Rendle et al. in 2012 improves the above matrix decomposition model by partial order loss.

With the rapid development of deep learning, a large number of deep learning techniques began to be introduced into recommendation systems, giving birth to a large number of recommendation system algorithms. Some of these works utilize the powerful expression learning of neural networks to characterize the massive amount of information related to the user and items. For example, NeuMF(He et al. 2017) extends the traditional matrix decomposition model and uses a nonlinear multilayer perceptron to characterize the user-item interaction information; VAE(Sedhain et al. 2015) is based on self-coding techniques and gives recommendation prediction by means of missing value prediction. In recent years, researches in graph recommendation systems (Yu, Zhang, and Qin 2022) (Zhao et al. 2022) has gradually become mainstream. However, graph neural network recommendation systems often face the cold start problem, so solving the cold start problem (Tao et al. 2022) (Neupane et al. 2022) is a major problem for graph neural network recommendation systems.

Another part of the work uses neural networks to process multimodal data such as text, images, audio, video, and graphs to complement the traditional recommendation interaction data to give better recommendations. For example, VPOI(Wang et al. 2017) interacts with user hidden factors and item hidden factors by building image features; another example is SPEX(Li et al. 2021) which is a framework for analyzing social network relationship graphs to assist recommendation systems in making decisions.

2.2 Shilling Attack

The Shilling attack technique is a class of attack that achieves the purpose of attack by forging a set of fake user data to be injected into the user feedback data, the problem was first proposed by Mahony in 2002(Gunes et al. 2014), the attack of Shilling attack is an attack in the training phase.

According to the classification of attack intent, Shilling attacks can be divided into two categories: random attacks and targeted attacks. Random attacks(Lam and Riedl 2004) indiscriminately induce the recommendation system to output inaccurate recommendation results for all users and products, with the goal of reducing the overall performance of the recommendation system. Targeted attacks, on the other hand, are targeted to mislead the recommendation system to produce incorrect prediction results on the target items, and

are mainly divided into two categories, push attacks and nuclear attacks.(Williams, Mobasher, and Burke 2007)

Using ML technology, some studies(Fang et al. 2018) formulate the TO attack as an optimization problem and find the approximate solution by projective gradient descent; others(Christakopoulou and Banerjee 2019) use adversarial generative networks (GANs) to generate fake user data based on the idea of adversarial learning. (Tang, Wen, and Wang 2020) combines these ideas and models the hidden feedback recommendation system as A two-layer optimization model that uses adversarial generation to generate fake user data. In addition to this, there are also some recent studies in the literature on cross-domain (Fan et al. 2021), federal learning (Rong et al. 2022) (Wu et al. 2022) attacks in recommendation system.

2.3 Black-Box Model Interpretability

In the field of black box model interpretability, most of the studies have been analyzed and studied based on perturbing the input observation output. For example, Simonyan and many other scholars have proposed prediction by perturbing the test points and interpreting the model (Adler et al. 2018)(Simonyan, Vedaldi, and Zisserman 2013) by the change in the prediction points. While Ribeiro et al. tried to explain the black-box model by fitting a simpler model locally around the test points(Ribeiro, Singh, and Guestrin 2016). Pang et al. proposed a method based on perturbed training points to find the influence of each training point on the results of the black-box model, and thus explain the black-box model(Koh and Liang 2017). Since the method can trace the source of the model parameters, i.e., the training samples, the method has stronger explanatory power. Therefore, this paper uses Pang's method to explore the influence of training points on the overall recommendation model.

3 Model

3.1 Problem Definition

Given an interaction matrix $\boldsymbol{X} \in \Re^{|\mathcal{U}|,|\mathcal{I}|}$, where $\mathcal{U}$ represents the set of users and $\mathcal{I}$ represents the set of items. The weblebrities set $\hat{\boldsymbol{X}}$ is a subset of $\boldsymbol{X}$. Any element $x_{i,j}$ in $\boldsymbol{X}$ and $\hat{\boldsymbol{X}}$ represents the rating of item j by user i. Then given a trust directed graph $G^{<\mathcal{U},\mathcal{T}>}$, point set $\mathcal{U}$ of this graph consists of the set of users and edge set $\mathcal{T}$ consists of the trust relationships between users. Finally, given a budget set $\boldsymbol{p}$, this budget set has a budget price for any weblebrity.

The goal of this paper is to return an optimal list of $\hat{\mathcal{U}}$ weblebrities users with the above interaction matrix $\boldsymbol{X}$, trust graph $G^{<\mathcal{U},\mathcal{T}>}$ and budget set $\boldsymbol{p}$, which satisfies the condition that the required budget is smaller than the given budget M, when the interaction matrix $\boldsymbol{X}$, trust graph $G^{<\mathcal{U},\mathcal{T}>}$ and budget set $\boldsymbol{p}$ are given. under the condition that its effect of targeted attack (measured using sum of influence) on the overall recommendation system I_{total} is maximum.

3.2 Framework

In this paper, a complete model framework is built, as shown in Figure 3-1. The model consists of five modules: input module, attack generation module, recommendation system module, weblebrity influence evaluation module, and weblebrity screening module, where the other four modules except the input module are independent of each other, i.e., the algorithms used in any module can be replaced independently without affecting the overall framework.

Among them, the input module is responsible for receiving and preprocessing the given interaction matrix $\boldsymbol{X}$, social network trust graph $G^{<\mathcal{U},\mathcal{T}>}$ and budget set $\boldsymbol{p}$ data. The attack generation module accepts the incoming set of netizens from the input module, generates the pre-processed TO attack data and then inputs them into the recommendation system module for recommendation together with other input data from the input module. In the recommendation system module, this paper uses the SPEX improvement framework to process the social trust network data to characterize the social influence of different weblebrities, and uses the framework to assist any black-box recommendation model to give recommendation results. The influence evaluation module is responsible for finding the influence of each training point on the results of the black box model using the trained recommendation system module data and the training data of weblebrities using the perturbation-based training point method proposed by Pang et al. Finally, the obtained influence is passed to the weblebrities filtering module, and any of the algorithmic frameworks is used to give the best $\hat{\mathcal{U}}$ weblebrities set list required in this paper.

3.3 Attack generation module

In a realistic sense, most of the data of weblebrities cannot be modified at will. Therefore, this paper adopts a simple attack strategy in the attack generation module, i.e., the rating value $\boldsymbol{x_{u,k}}$ of weblebrities users u on the target commodity vector $\boldsymbol{k}$ is set to the highest.

3.4 Weblebrities filtering module

In the weblebrities filtering module, this paper provides two filtering strategies, one is based on the greedy algorithm framework and the other is a simple approximate ranking algorithm. The overall algorithmic framework of the full text using the greedy algorithm framework is shown in the algorithm.

In order to make the "Influence Cost Ratio" truly represent the Influence Cost Ratio of the weblebrity to the recommendation system, it is required that the three parameters of Hit, Influence and Budget P of the weblebrity to the recommendation system satisfy a linear relationship. Therefore, we preprocess the Influence obtained from the influence evaluation module of the weblebrity and the price P in the budget set, and map them nonlinearly to Gaussian distribution data. Then, by selecting the optimal solution of Influence to Budget ratio for each weblebrities, and then recalculating the model and Influence after updating the data, we obtain the list of globally optimal solutions for weblebrities.

3.5 Recommendation System Module

The function to be achieved by this module is to train a recommendation model that accepts data from the user product

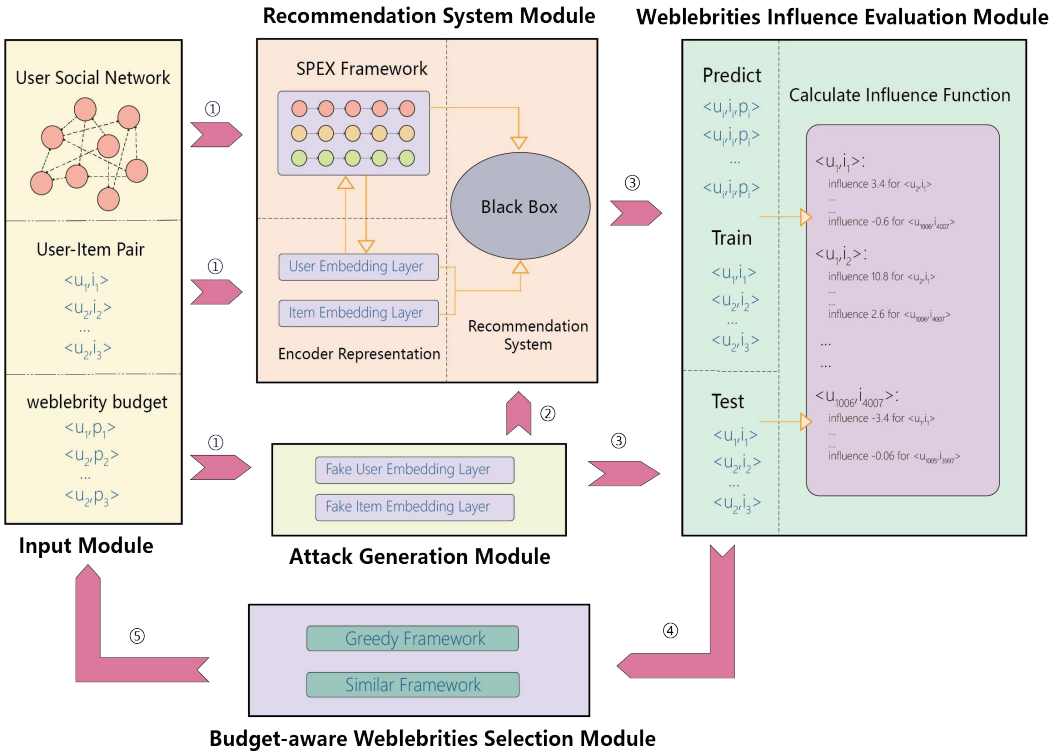

Figure 2: The weblebrity recommendation is regarded as a recommendation system shilling attack Before the attack, the recommendation system considers user 4 (Dr.) to be most similar to user 1 (Gentleman), and thus gives a prediction of 1 point for the target item; after the attack, the recommendation system considers user 4 (Dr.) to be most similar to user 5 (Weblebrity), and the social trust graph shows that user 4 (Dr.) trusts user 5 (Weblebrity), and thus gives a prediction of 5 points for the target item.

interaction matrix $\boldsymbol{X}$ in order to give suitable recommendation results. Since the SPEX framework can assist any black-box model to give recommendation assistance based on social networks, this is in line with the need for a black-box recommendation model in the context of the problem. Therefore, we adopt the SPEX framework (Li et al. 2021) to train the user trust social network graph $G^{<\mathcal{U},\mathcal{T}>}$, and simulate the individual influence heterogeneity of weblebrities and ordinary users in the recommendation system through user social relationships to influence the recommendation results.

Through the prediction and modeling of influence propagation paths, SPEX captures the information contained within the social homogeneity to make social recommendations, and combines with the original recommendation training task and balances the task loss by means of multi-task learning. Figure 3(Li et al. 2021) shows the principle of the SPEX prediction framework.

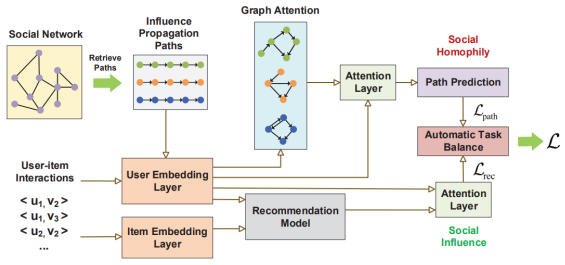

Figure 3: Framework of SPEX

3.6 influence assessment module for weblebrities

The function to be achieved by this module is to receive the recommendation results from the recommendation system module for processing, and then return the influence of the given weblebrity training samples on the model. The following derivation of [1] will give the formula of the influence of the weblebrity training samples on the model.

Let $\mathcal{L}(u,\theta)$ denote the loss function of user u when the parameters of the proxy model are θ, then the total loss can be expressed in terms of empirical risk as

$$\frac{1}{n}\sum_{i=0}^{u}\mathcal{L}(u_i,\theta)=\mathcal{L}(X,\theta). \tag{1}$$

Therefore, the estimation of the model parameters from

[1] the later formula derivation is referenced from Pang et al.'s study (Koh and Liang 2017)

ERM (empirical risk minimization) can be obtained as

$$\hat{\theta} = argmin_{\theta}\mathcal{L}(X, \theta). \quad (2)$$

At this time, a small perturbation is made to the weight of a netroots user u_i, that is, the weight of the netroots user u_i in the training set is increased by ϵ, and the netroots user is less falsifiable in a realistic sense, and only some of the goods can be perturbed, and the model parameters at this time will become

$$\hat{\theta_{\epsilon}} = argmin_{\theta}\mathcal{L}(X, \theta) + I\epsilon\mathcal{L}(u_i, \theta). \quad (3)$$

where I is a vector whose dimension is consistent with the set of items $\mathcal{I}$ and consists of only 0/1 values. $I_j = 0$ means that this user is not perturbed for item j and $I_j = 1$ means that this user is perturbed for item j.

Define the influence function of the training sample on the parameters as the value of the gradient of the parameters to the perturbation when the perturbation $\epsilon \to 0$. From Newton's method, the influence on the parameters is solved as:

$$I_{params}(u_i) = I^T \frac{d\hat{\theta}_{\epsilon,u_i}}{d\epsilon}|_{\epsilon=0} = I^T H_{\hat{\theta}}^{-1} \nabla_{\theta}\mathcal{L}(u_i, \hat{\theta}), \quad (4)$$

where $H_{\hat{\theta}} = \nabla_{\theta}^2\mathcal{L}(X, \hat{\theta})$ is the Hessien matrix of empirical risk.

The influence function of the training sample on the total Loss of the recommendation model yields

$$I_{loss}(u_i) = -(I\nabla_{\theta}\mathcal{L}(X, \hat{\theta}))^T H_{\hat{\theta}}^{-1} \nabla_{\theta}\mathcal{L}(u_i, \hat{\theta}). \quad (5)$$

In this paper, we also use the Stochastic Estimation method proposed by Pang et al.'s study (Koh and Liang 2017) to achieve the efficient calculation of the above Loss influence function.

4 Experiment

This section will briefly describe the work in the experimental part of this paper. This part of the experiment will answer the following questions.

Question 1: Is the influence described above valid as a weblebrity influence?

Question 2: How different are the results given by the approximation framework proposed in this paper, after greatly reducing the time complexity, from the results of the greedy algorithm?

Subsequently, this paper will conduct two parts of experiments based on the above setup. The first part of the experiments is the influence-based attack effectiveness evaluation experiment, which answers question 1 in detail. the second part is the comparison experiment of two netroots screening strategies, which answers question 2 in detail.

4.1 Experiment Setting

Data The data used in this paper is the Epinions dataset [2],which is a large scale, commonly used movie rating dataset in the field of recommendation systems. It contains information about users' ratings of movies and social information among users, where social information is measured using user-to-user trust relationships. The dataset $\boldsymbol{X}$ used in this paper contains 12,645 user ratings of 12,455 products, a total of 365,219 user product interaction items, 699,893 trust relationships between users, and 152,638 influence propagation paths are derived from them.

Due to the lack of real-existing weblebrity candidate list and budget data corresponding to each weblebrity, this paper independently selects the weblebrity candidate list and randomly generates the budget data corresponding to each weblebrity.

Hardware The machine configuration for running the model in this paper is as follows.

- CPU: two Intel(R) Xeon(R) CPUs, model E5-2678 v3 @2.50GHZ.
- RAM: 256GB.
- graphics: four GeForce RTX 2080Ti, where each card has 11GB of video memory.

Metrics In this paper, we use three metrics, Hit@20,Hit@50,CE loss, to measure the effect of the attack, and use the coincidence rate and Kendall rank correlation coefficient to measure the effect of the approximation framework on the simulation of the greedy algorithm. The specific meanings of the metrics are shown below.

- Hit@N: refers to the frequency that the target item k appears in the list of N recommendations returned by the recommendation system to all users. In this paper, we choose two common metrics, N=20 and N=50.
- CE Loss: The cross-entropy loss value of the recommendation prediction value matrix $\hat{\mathrm{X}}$ returned by the recommendation system and the target matrix T (defined as the matrix with the same shape as $\hat{\mathrm{X}}$, only the column value corresponding to the target commodity K is 1 and the rest values are 0).
- Overlap ratio: The number of netmarkets whose list given by the approximation framework is consistent with the list given by the greedy algorithm divided by the average total number of netmarkets.
- Kendall rank correlation coefficient: defined as $\tau = \frac{C-D}{N}$, where N is the total number of netroots, C is the logarithm of elements in the same order, and D is the logarithm of elements in the inverse order. This metric is commonly used to measure the similarity of sequences.

4.2 Influence Calculation Result

This part of the experiment aims to measure whether the weblebrity filtered using influence does produce a better attack effect on the recommender system. The specific experimental steps are as follows.

1. Select the number 0-299 in the user set $\mathcal{U}$ as the candidate list of weblebrities,set the total budget as 300, all weblebrities have the same budget price, and the value is 2.

[2]http://www.trustlet.org/epinions.html

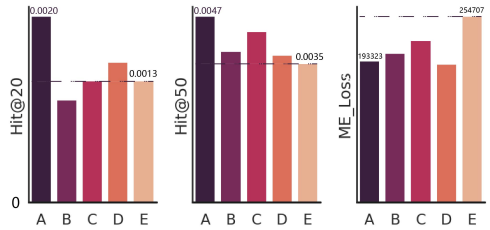

Figure 4: comparative results of indicators in each group The results of the graph are taken from the thirty repetitions of the experiment to take the best performance of the recommendation effect on the test set in one round, ABCDE five groups of experiments were compared with the meaning of the above, it can be seen that the attack performance with the 150 highest influential netizens for the attack (group A) is the best.

2. uses the weblebrity influence evaluation module to filter out the 150 weblebrities with the highest influence on a specific product K among the 0-299 weblebrities.
3. Five sets of experiments are designed to attack the recommender system separately, and the set of weblebrities used in the attack are: **A.** 150 weblebrities with the highest influence among numbers 0-299; **B.** 150 weblebrities with the lowest influence among numbers 0-299; **C.** 150 weblebrities randomly selected among numbers 0-299; **D .** 150 randomly selected weblebrities among all users; **E.** no attack

The experimental results are shown in the following figure 4. Comparing the results of the five groups of experiments, we can see that the weblebrities with high influence actually have a better effect on the attack on the recommendation effect.

4.3 Comparison of Two Screening Frameworks for Netflix

This part of the experiment aims to measure the simulation effect of the approximation framework on the greedy algorithm framework. The specific experimental steps are as follows.

1. Randomly select five users as webmasters, set the total budget to 50, and assign a random price between (5,30) to each user.
2. Use the approximation framework and the greedy algorithm framework to obtain the sequence of netroots $s1$ and $s2$, respectively.
3. Compare the repetition rate of the two sequences, and the Kendall rank correlation coefficient of the repetition part.

The experimental results are shown in the following table 1. The experimental results show that the approximation framework simulates the greedy algorithm framework extremely well.

Order	Repetition rate	Krcc
1	100%	100%
2	100%	100%
3	100%	100%
4	100%	95%
5	100%	100%

Table 1: experiment 4.2 results table

5 Conclusion

In order to propose a reliable, effective and realistic Netflix list screening model with budget, this paper proposes a Netflix list screening model with high generalizability, scalability and applicability to different black box recommendation models from the perspective of recommendation system attack, considering various elements such as social network, recommendation system and black box model influence. And the effectiveness of the model is proved through experiments.

References

Adler, P.; Falk, C.; Friedler, S. A.; Nix, T.; Rybeck, G.; Scheidegger, C.; Smith, B.; and Venkatasubramanian, S. 2018. Auditing black-box models for indirect influence. *Knowledge and Information Systems*, 54(1): 95–122.

Breese, J. S.; Heckerman, D.; and Kadie, C. 2013. Empirical analysis of predictive algorithms for collaborative filtering. *arXiv preprint*.

Christakopoulou, K.; and Banerjee, A. 2019. Adversarial attacks on an oblivious recommender. In *Proceedings of the 13th ACM Conference on Recommender Systems*, 322–330.

Covington, P.; Adams, J.; and Sargin, E. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, 191–198.

Fan, W.; Derr, T.; Zhao, X.; Ma, Y.; Liu, H.; Wang, J.; Tang, J.; and Li, Q. 2021. Attacking black-box recommendations via copying cross-domain user profiles. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, 1583–1594. IEEE.

Fang, M.; Yang, G.; Gong, N. Z.; and Liu, J. 2018. Poisoning attacks to graph-based recommender systems. In *Proceedings of the 34th annual computer security applications conference*, 381–392.

Goldberg, D.; Nichols, D.; Oki, B. M.; and Terry, D. 1992. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12): 61–70.

Gunes, I.; Kaleli, C.; Bilge, A.; and Polat, H. 2014. Shilling attacks against recommender systems: a comprehensive survey. *Artificial Intelligence Review*, 42(4): 767–799.

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, 173–182.

Koh, P. W.; and Liang, P. 2017. Understanding black-box predictions via influence functions. In *International conference on machine learning*, 1885–1894. PMLR.

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30–37.

Lam, S. K.; and Riedl, J. 2004. Shilling recommender systems for fun and profit. In *Proceedings of the 13th international conference on World Wide Web*, 393–402.

Li, H.; Li, L.; Xv, G.; Lin, C.; Li, K.; and Jiang, B. 2021. SPEX: A Generic Framework for Enhancing Neural Social Recommendation. *ACM Transactions on Information Systems (TOIS)*, 40(2): 1–33.

Neupane, K. P.; Zheng, E.; Kong, Y.; and Yu, Q. 2022. A Dynamic Meta-Learning Model for Time-Sensitive Cold-Start Recommendations. *Genre*, 2: 3–0.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009*, 452–461.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135–1144.

Rong, D.; Ye, S.; Zhao, R.; Yuen, H. N.; Chen, J.; and He, Q. 2022. FedRecAttack: Model Poisoning Attack to Federated Recommendation. *arXiv preprint arXiv:2204.01499*.

Sarwar, B.; Karypis, G.; Konstan, J.; and Riedl, J. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, 285–295.

Sedhain, S.; Menon, A. K.; Sanner, S.; and Xie, L. 2015. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th international conference on World Wide Web*, 111–112.

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Tang, J.; Wen, H.; and Wang, K. 2020. Revisiting adversarially learned injection attacks against recommender systems. In *Fourteenth ACM conference on recommender systems*, 318–327.

Tao, W.; Li, Y.; Li, L.; Chen, Z.; Wen, H.; Chen, P.; Liang, T.; and Lu, Q. 2022. SMINet: State-Aware Multi-Aspect Interests Representation Network for Cold-Start Users Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 8476–8484.

Wang, S.; Wang, Y.; Tang, J.; Shu, K.; Ranganath, S.; and Liu, H. 2017. What your images reveal: Exploiting visual contents for point-of-interest recommendation. In *Proceedings of the 26th international conference on world wide web*, 391–400.

Williams, C. A.; Mobasher, B.; and Burke, R. 2007. Defending recommender systems: detection of profile injection attacks. *Service Oriented Computing and Applications*, 1(3): 157–170.

Wu, C.; Wu, F.; Qi, T.; Huang, Y.; and Xie, X. 2022. FedAttack: Effective and Covert Poisoning Attack on Federated Recommendation via Hard Sampling. *arXiv preprint arXiv:2202.04975*.

Yu, W.; Zhang, Z.; and Qin, Z. 2022. Low-pass Graph Convolutional Network for Recommendation.

Zhao, S.; Wei, W.; Zou, D.; and Mao, X. 2022. Multi-view intent disentangle graph networks for bundle recommendation. *arXiv preprint arXiv:2202.11425*.