RRL: Recommendation Reverse Learning

Xiaoyu You, Jianwei Xu, Mi Zhang*, Zechen Gao, Min Yang*

Fudan University
17212010047@fudan.edu.cn, 21210240379@m.fudan.edu.cn, mi_zhang@fudan.edu.cn,
22210240151@m.fudan.edu.cn, m_yang@fudan.edu.cn,

Abstract

As societies become increasingly aware of data privacy, regulations require that private information about users must be removed from both database and ML models, which is more colloquially called 'the right to be forgotten'. Such privacy problems of recommendation systems, which hold large amounts of private data, are drawing increasing attention. Recent research suggests dividing the preference data into multiple shards and training submodels with these shards and forgetting users' personal preference data by retraining the submodels of marked shards. Despite the computational efficiency development compared with retraining from scratch, the overall recommendation performance deteriorates after dividing the shards because the collaborative information contained in the training data is broken. In this paper, we aim to propose a forgetting framework for recommendation models that neither separate the training data nor jeopardizes the recommendation performance, named Recommendation Reverse Learning (RRL). Given the trained recommendation model and marked preference data, we devise *Reverse BPR Objective* (RPR Objective) to fine-tune the recommendation model to force it to forget the marked data. As the recommendation model encodes the complex collaborative information among users, we propose to utilize Fisher Information Matrix (FIM) to estimate the influence of reverse learning on other users' collaborative information and guide the updates of representations. We conduct experiments on two representative recommendation models and three public benchmark datasets to verify the efficiency of RRL. For the forgetting completeness, we use RRL to make the recommendation model poisoned by shilling attacks forget malicious users.

Introduction

Recommender systems play an essential role in a variety of real-world applications such as social media (Song et al. 2021; Wu et al. 2022), e-market (He et al. 2020; Wang et al. 2019; Wu et al. 2021; Chen et al. 2020) and etc. Recommender systems are composed of collected interactions (e.g., clicks, likes, and buys) and recommendation models, where the personalized recommendation services are the prediction results of recommendation models learned from the historical interactions (He et al. 2020; Wang et al. 2019). To provide accurate recommendation services, existing recommendation models rely on collaborative filtering to reconstruct the historical interactions based on the embeddings of users and items. In the face of demands for privacy, data protection regulations[2] require systems gleaning private data like recommendation systems must allow users to eliminate their private data from not only the database but also the recommendation model, which is termed recommendation unlearning (Chen et al. 2022a).

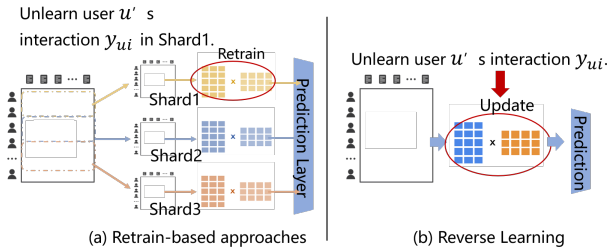

Figure 1: Overview of existing two types of forgetting marked interactions (1) Retrain-based approaches and (2) Reverse Learning.

Recently, recommendation unlearning approaches, which aim to eliminate the influence of marked interactions from recommendation models, become a popular research topic (Chen et al. 2022a; Bourtoule et al. 2021). The naive way to achieve this is to remove the marked interactions from the historical interactions and retrain the recommendation model from scratch, which is computationally expensive Recently, some studies propose to split the historical interactions into several shards and train a submodel on each shard (Chen et al. 2022a; Bourtoule et al. 2021). When the deletion requirement comes, the corresponding submodel will be retrained from scratch. Finally, the predictions of all submodels will be aggregated to provide the recommendation services. We refer to this kind of unlearning approach as retrain-based unlearning. Despite that retrain-based approaches guarantee the enforcement of deletion and unlearning efficiency, the recommendation performance dete-

*Corresponding authors.

[2]CCPA in California, GDPR in Europe, PIPEDA in Canada, LGPD in Brazil, and NDBS in Australia.

riorates considerably [1]. This is because splitting the interaction data into shards will inevitably destroy the collaborative information contained in the interactions, which reveals the similarity between users (items).

In this light, we resort to a recently emerging line of unlearning techniques named reverse learning (Graves, Nagisetty, and Ganesh 2021; Wu et al. 2023). Reverse learning approaches aim to update the parameters of the current model to force the model 'forget' the marked data, as if the marked data did not participate in the training process (Graves, Nagisetty, and Ganesh 2021). To our best knowledge, no effort has been made to tailor the reverse learning for recommendation models, due to the following two main challenges.

- **C1:** *How to determine the reverse objective for forgetting interaction data?* Related literature mainly focuses on traditional machine learning models like image classification models. Given a marked image, the memorization of the model on the marked image can be quantified by the classification loss (Graves, Nagisetty, and Ganesh 2021). Therefore, the reverse objective is to maximize the up-weighted loss. Differently, the marked targets in the settings of recommender systems are users' interactions, about which the information memorized in the recommendation model relies on both the predicted scores and ranks of the marked interactions.
- **C2:** *How to balance the removal completeness and recommendation performance for reversing?* The recommendation model encodes collaborative information latent in the historical interactions to model similarity between users (items) into user and item representations, i.e., collaborative similarity. Accordingly, forgetting marked interactions should not only update the representations of related users and items but also other implicitly related users and items. In addition, such collaboratively updating may influence other collaborative information entangled in the representations. Therefore, how to update the user and item representations according to the reverse objective should be carefully designed.

In this paper, we propose a reverse learning framework for recommendation unlearning, named recommendation reverse learning (RRL). To tackle the first challenge, we devise a reverse objective named *Reversed Personalized Ranking* (RPR) objective, which considers both predicted scores and ranks of the marked interactions. That is, the predicted score of the marked interactions should be similar or even smaller than the items not interacting. As for the second challenge, we introduce the Fisher Information Matrix (FIM) to measure the collaborative similarities among user and item representations. After this, we devise a *Collaborative Similarity Regularizer* (dubbed CS regularizer) and add it to the reverse objective to guide the updates of user and item representations. Finally, we present a theoretical interpretation of the proposed RRL framework, indicating that the impact of RRL on the recommendation model is equivalent to retraining the model from scratch. We summarize the contributions of this paper as follows.

- To our best knowledge, we for the first time tailor reverse learning for forgetting interactions in recommendation tasks.
- We devise the reverse objective with fisher information regularizer to enforce the guarantee of removal completeness and recommendation performance. Besides, we present a theoretical analysis of the proposed framework to facilitate the verification of forgetting.
- We conduct empirical evaluations on three benchmark datasets and implement RRL in two representative recommendation models. The empirical results demonstrate that after forgetting marked interactions the overall performance is still comparable to retrain counterpart while the running time is much shorter. Besides, we design experiments based on shilling attacks to verify the proposed RRL can completely forget the marked interactions.

[1] In our empirical evaluation the recommendation performance will deteriorate $10\% \sim 30\%$

Related Work

Privacy Concern about Recommendation Systems

Recommendation systems face serious data privacy issues since RS engines aim to personalize content according to users' preferences, and handle sensitive information from users to provide personalized recommendations (He et al. 2020; Wang et al. 2019; Chen et al. 2020; Wu et al. 2021). These systems glean large amounts of data for ensuring accurate and engaging recommendations, which are automatically gathered or explicitly provided by users (Portugal, Alencar, and Cowan 2018). Nowadays, users' privacy can be exposed in many ways, e.g., the semi-trust recommendation systems expose users' data to third parties for monetary benefits (Zhang et al. 2021), and malicious attackers steal the users' private information by leveraging the recommendation systems' ability of inferring preferences of users, i.e., membership inference attack (Zhang et al. 2021). Accordingly, lots of efforts have been made to study and develop powerful privacy preservation mechanisms for recommendation systems, which can be roughly grouped into three categories (1) Architecture-based solutions, e.g., federated recommendation learning (Luo, Xiao, and Song 2022; Chen et al. 2023; Cai et al. 2022; Meihan et al. 2022). (2) Algorithm-based solutions, e.g., anonymization techniques (Chang et al. 2010), obfuscation techniques (Zhang et al. 2023), and differential privacy (Shen and Jin 2014; Chen et al. 2022b). (3) Ethical guidelines and regulations for privacy, e.g., right to be forgotten.

Machine Unlearning

Machine unlearning aims to help machine learning models remove the influence of a specific subset of training examples, which is an emergent subfield of pursuing ethical machine learning algorithms (Graves, Nagisetty, and Ganesh 2021; Bourtoule et al. 2021; Liu et al. 2020; Wu, Hashemi, and Srinivasa 2022; Gupta et al. 2021; Sekhari et al. 2021; Lin et al. 2023; Yan et al. 2022; Brophy and Lowd 2021). Machine unlearning approaches are first studied on traditional machine learning tasks like regression

(Tarun et al. 2023) and classification (Brophy and Lowd 2021; Lin et al. 2023), which fall into two groups (1) retrain-based approaches (Bourtoule et al. 2021) and (2) reverse learning approaches (Graves, Nagisetty, and Ganesh 2021; Wu, Hashemi, and Srinivasa 2022). First, retrain-based approaches aim to speed up the naive retrain approach, i.e., retrain the whole machine learning model from scratch (Chen et al. 2022a), by splitting the large model into several lightweight models. To maintain the comparable performance in comparison to the original model, many studies propose ensemble-based strategies to aggregate the prediction results of lightweight models (Bourtoule et al. 2021). This kind of method is efficient in dealing with IID data where the training examples are independent, whereas some research observes large performance degradation in non-IID data like graph structure data (Wu et al. 2023). It is mainly because splitting inevitably breaks the data dependency which is significant for non-IID data learning (Wu et al. 2023). Similarly, for the recommendation model, the utility gap is also large, and thus in this paper, we aim to propose an unlearning mechanism without splitting the recommendation models.

The reverse learning approach's purpose is to revert the optimization process of the model as if the deleted data did not exist (Graves, Nagisetty, and Ganesh 2021; Wu, Hashemi, and Srinivasa 2022). These methods are inspired by the stochastic gradient descent process of machine learning models, and removing the information of the marked data is to revert the process through up-weighted loss (Graves, Nagisetty, and Ganesh 2021) and negative gradients (Wu et al. 2023). To further enhance the reversing time, some studies propose to utilize the influence function to directly approximate the unlearned model (Guo et al. 2019). Nevertheless, to our best knowledge, there is no effort to tailor reverse learning for recommendation models.

Verification of certifying that the model completely removes the information of the marked data is essential. There are two main criteria for the evaluation metrics (1) certify that one cannot easily distinguish between the unlearned models and their retrained counterparts, e.g., the performance degradation of unlearned data (Wu, Hashemi, and Srinivasa 2022) and membership inference attack (Shokri et al. 2017; Conti et al. 2022) (2) inject poisoned data (e.g., backdoor data) for deceiving the machine learning model, and verify the unlearning effectiveness by the attack success rate before/after unlearning (Wu et al. 2023). We mainly use the second criterion to verify the unlearning effectiveness, because (1) the predicted scores of training data are not very significant. (2) there is no efficient membership inference attack against recommendation models.

Backgrounds & Settings

Recommendation system is composed of database D and recommendation model f_θ where θ denotes the model parameters. Specifically, database D contains N users, denoted as $u \in U$, M items, denoted as $i \in I$, and interaction matrix $Y \in \mathbb{R}^{N \times M}$ where $y_{ui} = 1$ represents that user u has interacted with item i and $y_{ui} = 0$ versus vice. Recommendation model f_θ aims to encode the collaborative signals latent in the interaction data that reveals the similarity between users (or items), named collaborative similarity, into users' and items' d-dimentional embeddings, i.e., $e_u = f_\theta(u)$ and $e_i = f_\theta(i)$. After this, the personalized preferences are modeled by $\hat{y}_{ui} = e_u^T e_i$, and then the item having a higher prediction score and never being interacted with by the user will be recommended.

To obtain informative embeddings that can accurately characterize users' preferences, the embeddings will be optimized through a ranking loss to ensure the interacted items have higher prediction scores than those not interacted with. For instance, one of the most popular ranking losses, named Bayesian Personalized Ranking loss (BPR) (Rendle et al. 2012), is represented as

$$\mathcal{L}_{\text{BPR}}(D;\theta) = \sum_{u \in U} \sum_{i \in I_u^+, j \in I_u^-} -\ln\sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda\|\theta\|^2. \quad (1)$$

where $\sigma(\cdot)$ is the sigmoid function, λ is the weight hyperparameter and I_u^+, I_u^- are interacted items set of u and non-interacted items set of u respectively. The optimized parameters of the recommendation model f_θ are denoted as θ_0. In the recommendation model, most of the parameters are embeddings of users and items, and the remaining parameters are additional modules like non-linear layers (He et al. 2017). Therefore, in the following unlearning process, the core is to update the embeddings in the recommendation model according to the unlearning requirements.

The unlearning target of recommendation systems is to remove the private interactions from both the database and recommendation model according to users' requirements. Note that users may require the system to delete either all their interaction data or a subset of their interaction data. For the sake of discussion, we assume that a user u requires the system to delete her/his interactions I_u^+ from the database which are referred to as *marked interactions*[2]. After receiving the requirement, the recommendation system needs to take the following two workflows:

- *Workflow I: Remove data from the database.* Marked interaction data $y_{ui} \in I_u^+$ should be removed from the interaction matrix, e.g., by setting them as 0.
- *Workflow II: Retrain the recommendation model.* The recommendation model f_θ should be retrained according to the training process in Eq 1 on the interaction data updated by Workflow I.

In workflow II, retraining the recommendation model from scratch is computationally inefficient, because the amount of interactions and the size of the recommendation model are both very large. To tackle this, reverse learning mechanism $\mathcal{R}(\cdot)$ takes the current model f_θ, the marked interaction I_u^+ and the database D as inputs, to output a new model f_θ. In contrast to retraining from scratch, the reverse learning mechanism $\mathcal{R}$ updates the current recommendation model for only several steps, which is similar to the inverse of the forward optimization process. Additionally, the new model also satisfies the following criteria **(1)** Unlearning Completeness - to remove the information of marked inter-

[2]We discuss the removal of partial interactions and multiple users in Experiments

actions completely, **(2)** Comparable Recommendation Performance - brings little utility gap in comparison to retraining from scratch.

Recommendation Reverse Learning

In this section, we present a reverse learning framework for recommendations named recommendation reverse learning (RRL), which is mainly composed of reverse personalized ranking objective and collaborative similarity regularizer.

Reverse Personalized Ranking Objective

To achieve the goals of reversion, it is essential to design a suitable reverse objective to guide the reversing process of the recommendation model. As for traditional machine learning tasks like image classification, the classification model tends to perform better on their training data (Graves, Nagisetty, and Ganesh 2021). Accordingly, given a marked image, the reverse objective is to minimize the predicted probability to a sufficiently small value. Similarly, in a recommendation system, the predicted score of interacted items will be larger than the non-interacted items for an arbitrary user. Accordingly, the reverse objective of user u for her/his marked interactions can be: *make the predicted score of marked interactions smaller than items not interacting*, which can be written as follows.

$$\min \mathcal{L}_{\text{RPR}} = \sum_{i \in I_u^+} \sum_{j \in I_u^-} \ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}). \tag{2}$$

The remaining problem is how to update representations.

Collaborative Similarity Analysis via Fisher Information Matrix

Based on the objective of RPR in Eq 2, reverse learning is to update the embeddings of the recommendation model. We consider the following two scenarios: **(1)** Updating the user embedding only: the item embeddings containing the user preference are not updated, resulting in an incomplete removal. **(2)** Updating both the user embeddings and item embeddings: Since preference information is entangled in item embeddings, such updates may cause catastrophic forgetting of other user preferences, resulting in inaccurate recommendations for other users. In a word, updating the embeddings of the recommendation model according to the RPR objective is faced with the trade-off between normal utility and remove completeness. These are mainly because the user and item embeddings are embodied the similarities in the interaction data, i.e., collaborative similarity. To pursue remove completeness and comparable normal utility, we need a method that can measure the collaborative similarities between users (items), and determine which embeddings should be updated.

To this end, we resort to the Fisher Information Matrix (FIM) over the learned embeddings θ_0 to represent the collaborative similarities. FIM is equivalent to the second derivative of the BPR loss. Specifically, given an arbitrary user u, the FIM over embedding e_u measured by current embeddings can be written as $F_u = \mathbb{E}[\frac{\nabla^2 \mathcal{L}_{\text{BPR}}(D;\theta)}{\nabla^2 e_u}]$. This matrix can help measure the correlation between e_u and other embeddings after learning interaction data D. That is, given another arbitrary embedding e_i, the correlation between e_i and e_u can be calculated by $\text{Corr}(e_i, e_u) = e_i^T F_u e_u$.

The larger value of $\text{Corr}(e_i, e_u)$ indicates that the collaborative similarity between e_i and e_u is larger. Accordingly, perturbing e_u leads to the corresponding perturbations in e_i. Therefore, we propose that after perturbations e_u and e_i should satisfy the correlation before perturbations, that is, $\text{Corr}(\delta(e_i), \delta(e_u)) \approx \text{Corr}(e_i, e_u)$.

In this light, we propose to add the collaborative similarities as a regularizer during the optimization of the RPR objective. The revised RPR objective can be written as follows (named Collaborative Similarity Regularizer, dubbed CS regularizer).

$$\mathcal{L}'_{\text{RPR}} = \mathcal{L}_{\text{RPR}} + (\theta - \theta_0)^T F_{\theta_0} (\theta - \theta_0). \tag{3}$$

Note that F_{θ_0} represents the FIM calculated on all embeddings. With the help of the regularizer, when we optimize the RPR objective through SGD, it will collaboratively update other embeddings w.r.t. the embeddings in the RPR objective. As for those embeddings having small collaborative similarities, they will not be updated. Finally, the proposed RRL framework can balance between forgetting completeness and normal utility.

Nevertheless, there still remains a concern that it is computationally expensive to calculate the FIM in the settings of the recommendation model. Therefore, we propose to approximate the FIM by the first-order derivatives, that is, $F_{\theta_0} \approx \mathbb{E}[\nabla_\theta \mathcal{L}_{BPR}(D;\theta) \nabla_\theta \mathcal{L}_{BPR}(D;\theta)^T]$. Such approximation is widely studied and used in the related literatures such as continual learning (Zenke, Poole, and Ganguli 2017) and multi-task learning (Li, Liao, and Carin 2009).

Bayesian Interpretation of RRL

We present the theoretical interpretation of the RRL objective in Eq 3 through the Bayesian theorem. Denote the remaining interactions as $D_{/\Delta D}$. The learning process can be regarded as maximizing the posterior distribution estimated by θ, i.e., $\max P(\theta \mid D)$, with a certain prior distribution of $g(\theta)$. Such posterior distribution $P(\theta \mid D)$ can be decomposed as follows.

$$P(\theta \mid \Delta D, D_{/\Delta D}) = \frac{P(\theta \mid D_{/\Delta D}) P(\Delta D \mid \theta, D_{/\Delta D})}{P(\Delta D, D_{/\Delta D})}. \tag{4}$$

$$\begin{aligned} \log P(\theta \mid D) = \log & P(\theta \mid D_{/\Delta D}) + \\ & \log P(\Delta D \mid \theta) - \log P(\Delta D). \end{aligned} \tag{5}$$

We can derive the log posterior distribution $\log P(\theta \mid D_{/\Delta D})$ as,

$$\log P(\theta \mid D_{/\Delta D}) = \log P(\theta \mid D) - \log P(\Delta D \mid \theta) + \log P(\Delta D).$$

Maximizing the log posterior distribution $\log P(\theta \mid D_{/\Delta D})$ is equivalent to retraining a recommendation model from scratch after removing ΔD from D. According to Eq 6, it is also equivalent to maximizing the posterior distribution on the whole interaction D, and minimizing the likelihood on marked interaction ΔD.

Minimizing the likelihood of marked interaction ΔD is equivalent to minimizing the RPR objective in Eq 2. Then

the optimal parameters θ_0 are learned by maximizing the log-likelihood $P(D \mid \theta)$ which is equivalent to minimizing the BPR loss in Eq 1. We can approximate the posterior $\log P(\theta \mid D_{/\Delta D})$ by leveraging θ_0 and assuming the prior distribution $g(\theta)$ as a normal distribution $\mathcal{N}(\theta, \sigma^2)$ as

$$\mathcal{L}(\theta) \approx \mathcal{L}_{\text{BPR}}(D;\theta_0) + (\theta-\theta_0)^T \frac{\partial \mathcal{L}_{\text{BPR}}(D;\theta)}{\partial \theta_0} + \frac{1}{2}(\theta-\theta_0)^T \frac{\partial^2 \mathcal{L}_{BPR}(D;\theta)}{\partial^2 \theta_0}(\theta-\theta_0). \tag{6}$$

As at the optimal point, we have $\mathcal{L}_{\text{BPR}}(D;\theta_0) \approx 0$ and $\| \frac{\partial \mathcal{L}_{\text{BPR}}(D;\theta)}{\partial \theta_0} \|^2 \approx 0$. Then we can derive the approximation of optimal posterior distribution as

$$\mathcal{L}(\theta) \approx \frac{1}{2}(\theta-\theta_0)^T \frac{\partial^2 \mathcal{L}_{BPR}(D;\theta)}{\partial^2 \theta_0}(\theta-\theta_0). \tag{7}$$

Note that the term $\frac{\partial^2 \mathcal{L}_{BPR}(D;\theta)}{\partial^2 \theta_0}$ is the Fisher Information Matrix. Based on this, we conclude that maximizing the posterior distribution $\log P(\theta \mid D_{/\Delta D})$ is equivalent to the RRL objective in Eq 3 which is to minimize the RPR objective with a CS regularizer.

Comprehensive Framework

In summary, after the arrival of unlearning requirements ΔD, the proposed RRL framework will update current parameters for several steps according to

$$\theta \leftarrow \theta - \alpha \nabla_\theta (\mathcal{L}_{RPR} + (\theta-\theta_0)^T F_{\theta_0}(\theta-\theta_0)). \tag{8}$$

where α is the learning rate. Note that the FIM F_{θ_0} is calculated offline after the model learning on the entire dataset D, which does not influence the complexity of optimizing Eq 8. In addition, we consider the generalization of RRL to the following scenarios,

- *Unlearning more users and more interactions.* There may not be one user to evoke the unlearning, and different users may require to unlearn different numbers of interactions, which only affects the size of marked interactions.
- *Sequential unlearning* $\{\Delta D_1, ..., \Delta D_K\}$. After unlearning a set of marked interaction ΔD_k, the unlearned model is f_{θ_k} and the remaining interactions is $D_{/\{\Delta D_1 \cup ... \cup \Delta D_k\}}$. The FIM will be re-estimated offline according to f_{θ_k} and $D_{/\{\Delta D_1 \cup ... \cup \Delta D_k\}}$, and will be used in the next unlearning ΔD_{k+1}. In our experiments, we verify that the RRL framework can balance the remove completeness and normal utility in the setting of sequential unlearning.
- *Recommendation model with different architectures.* The proposed RRL framework can be directly applied to recommendation models with parameters that are not exactly embeddings, e.g., deep learning-based recommendation models with non-linear layers (He et al. 2017), as the gradients in Eq 8 can be directly computed. The FIM of such parameters indicates the contribution of marked interactions on learning these parameters.

Experiments

In this section, we conduct empirical evaluations on the proposed RRL to study the following research questions. **RQ1:** How does RRL perform in terms of recommendation and running time as compared with the existing unlearning approaches? **RQ2:** Can RRL achieve removal completeness of the marked interactions on the learned recommendation model? **RQ3:** How does the performance of RRL under different unlearning settings?

Dataset	#Users	#Items	#Dense
Gowalla	29,858	40,981	0.084%
Yelp2018	31,668	38,048	0.130%
ML-1M	6,022	3,043	4.888%

Table 1: Statistics of datasets

Experimental Setup

Datasets. We use three real-world recommendation datasets, i.e., Gowalla (Liang et al. 2016), Yelp2018 (Wang et al. 2019), and Movilens-1m (dubbed as ML-1M)[3] which are widely used for benchmarking. Table 1 shows statistics of three datasets. Each dataset is split into training/validation/testing sets by the ratio of $70/10/20\%$. Validation sets are used to tune hyper-parameters.

Baselines & Recommendation models. We compare RRL with three retraining-based unlearning strategies which are Retrain, SISA (Bourtoule et al. 2021) and RecEraser (Chen et al. 2022a). The approach *Retrain* means to delete the marked interactions from the database and train the recommendation model from scratch. SISA is a traditional machine unlearning approach that splits training data into several shards and trains a submodel for each shard. RecEraser enhances SISA by introducing an attention mechanism to aggregate the predictions of each shard, which can help develop the recommendation performance of SISA. In addition, we implement these unlearning strategies and RRL on two representative recommendation models, which are MF-BPR (Rendle et al. 2012) and LightGCN (He et al. 2020). Specifically, MF-BPR optimizes user and item representations according to BPR, and LightGCN leverages neighborhood aggregation to augment the representations.

Evaluation metrics. To evaluate the recommendation performance after unlearning, we adopt two widely-used evaluation metrics (He et al. 2020): Recall@K and NDCG@K, which are measured on the remaining interactions after unlearning. By default, we set $K = 20$. To evaluate the removal completeness, we leverage shilling attacks, e.g., Popularity Attack (Fang et al. 2018; Mobasher et al. 2007) and Bi-level Attack (Tang, Wen, and Wang 2020). First, shilling attacks inject malicious users with poisoned interactions into the database aiming to increase the recommendation of a target item, denoted as $\#Rec$. Second, the unlearning strategies are used to unlearn these malicious users and $\#Rec$ after unlearning is used to measure the unlearning degree. If $\#Rec$ is small enough after unlearning, the removal is considered to be complete. To evaluate the unlearning efficiency, we utilize the training (updating) time dubbed RT.

[3] https://grouplens.org/datasets/movielens/1m/

		MF-BPR			LightGCN		
		Recall	NDCG	RT(s)	Recall	NDCG	RT(s)
Gowalla	Original	0.144	0.114	12575.0	0.182	0.154	23645.0
	Retrain	0.143	0.115	12841.0	0.180	0.153	24120.0
	SISA	0.051	0.046	3782.5	0.080	0.071	26796.9
	RecEraser	0.111	0.094	11165.8	0.148	0.128	17038.6
	Ours	0.144	0.114	3.5	0.181	0.154	255.0
Yelp2018	Original	0.047	0.038	19232.0	0.063	0.052	41687.0
	Retrain	0.047	0.038	19440.0	0.063	0.052	42900.0
	SISA	0.037	0.030	13286.7	0.034	0.028	36507.1
	RecEraser	0.043	0.035	20734.3	0.051	0.042	17462.3
	Ours	0.046	0.037	27.5	0.063	0.052	316.4
ML-1M	Original	0.208	0.194	12171.0	0.246	0.234	20868.0
	Retrain	0.205	0.191	12240.0	0.243	0.230	18900.0
	SISA	0.215	0.211	30016.1	0.192	0.194	5606.8
	RecEraser	0.241	0.227	3745.9	0.226	0.220	8316.3
	Ours	0.207	0.193	116.2	0.250	0.239	682.5

Table 2: Comparison of recommendation performance and running time for different unlearning strategies.

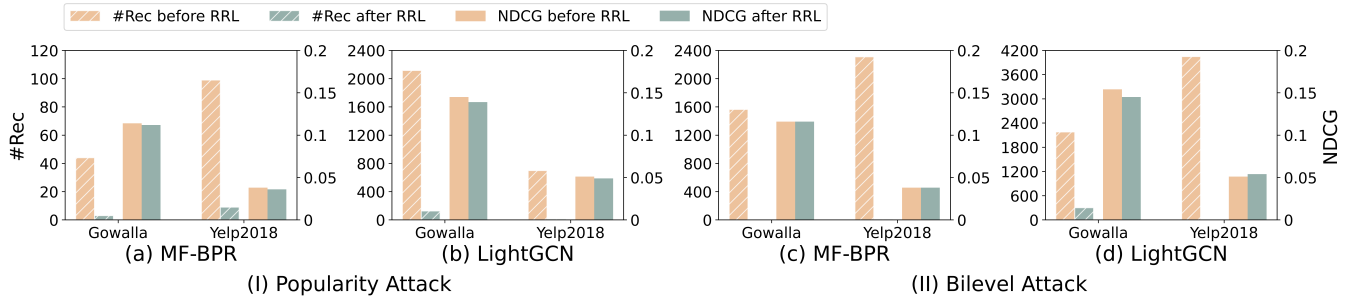

Figure 2: Verification of removal completeness by unlearning malicious users injected by shilling attacks.

Implementation details. We randomly sample 100 users from the database, and 50% of these users require to delete all their interactions. For the remaining 50% of these 100 users, we randomly sample $r\%$ of personalized interactions to unlearn. The remaining users are used to measure the recommendation performance after unlearning. For each baseline model, we mostly follow the suggested experimental settings and set the hyper-parameters as suggested in the original papers, including but not limited to the learning rate, the regularization coefficient, etc. The depth LightGCN is set to 3, and the size of the embeddings of recommendation models is set as 128. Finally, for shilling attacks used to verify removal completeness, we randomly choose an unpopular item as the target item to measure $\#Rec$, and the number of malicious users is set as 10% of all users.

RQ1: Evaluation of Utility and Efficiency

We first present the empirical evaluation of recommendation performance and unlearning efficiency in Table 2. For the overall performance of the unlearned model, the results of the proposed RRL are the most equivalent to the results of Retrain. After unlearning, the recommendation performance of Retrain and RRL will slightly decrease, e.g., 0.001. We infer the main reason is that unlearning 100 users' interactions may increase the sparsity of recommendation data, resulting slight deterioration of recommendation performance. In addition, we have the following observations. **(1)** Both SISA and RecEraser on Gowalla and Yelp2018 cause the deterioration of recommendation performance, e.g., for Gowalla, the decrease of Recall ranges from 0.001 to 0.09, and the decrease of NDCG is in the range of $[0.01, 0.07]$ while for Yelp2018 the decrease of Recall ranges from 0.001 to 0.03, and the decrease of NDCG is in the range of $[0.024, 0.03]$. This is mainly because splitting the interactions into different shards improves the sparsity of these two datasets which are very sparse in the first. **(2)** SISA and RecEraser perform much better than Retrain and RRL on dataset ML-1M. This is because the dataset ML-1M is not very sparse, e.g., its density is about 37 and 58 times larger than Yelp2018 and Gowalla respectively. We have checked the performance of submodels of SISA and RecEraser which is much worse than Retrain and RRL. But the aggregation of results of submodels can enhance the recommendation performance on such datasets.

As for the unlearning efficiency, all the unlearning strategies are much more efficient than Retrain. Especially, RRL is the most effective, e.g., RRL costs only 3.5s on unlearning interactions of Gowalla. This is because when unlearn-

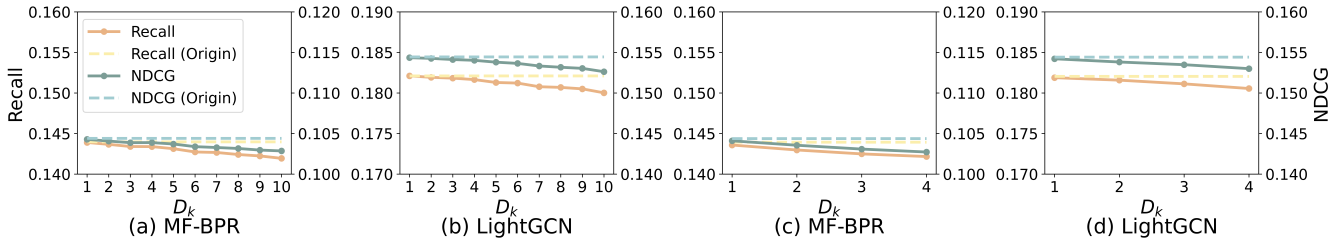

Figure 3: Recommendation performance under the settings of sequential unlearning. The x-axis denotes the index of marked interactions. In (a)-(b), $\mid D_k \mid = 10$, while in (c)-(d), $\mid D_k \mid = 20$.

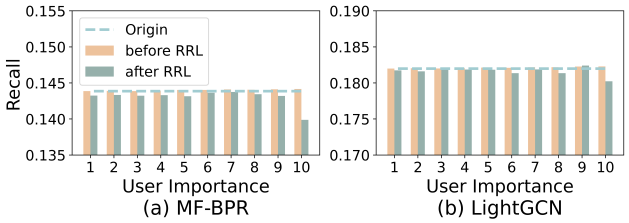

Figure 4: Recommendation performance of unlearning different users.

		MF-BPR			LightGCN		
		Recall	NDCG	#Rec	Recall	NDCG	#Rec
I	Org	0.144	0.114	44	0.174	0.145	2115
	w/	0.144	0.112	3	0.170	0.139	126
	w/o	0.134	0.102	1	0.174	0.144	716
II	Org	0.145	0.116	1564	0.182	0.154	2177
	w/	0.144	0.116	1	0.172	0.145	299
	w/o	0.135	0.104	98	0.170	0.143	786

Table 3: Removal completeness and overall performance w/ or w/o CS. I is Popularity Attack and II is Bilevel Attack.

ing interactions are distributed on all shards SISA and RecEraser have to retrain all models and even additional modules which costs additional training time.

RQ2: Evaluation of Removal Completeness

In order to prove the removal completeness, we propose to utilize shilling attacks to inject malicious users and unlearn these malicious users. The attack performance $\#Rec$ before/after unlearning is shown in Fig 2. As we can see, after unlearning, the $\#Rec$ of the target item will be decreased, e.g., it will be decreased to 0 on MF-BPR model and will be decreased by almost 80 percent on LightGCN. In addition, the proposed RRL is agnostic to the types of shilling attacks. When unlearning malicious users generated by Popularity Attack and Bilevel Attack, the recommendation of the target item can be decreased by at least 80 percent. This indicates that the proposed RRL can enforce the guarantee of removal completeness.

RQ3: Studies of RRL

Sequential unlearning. We also verify the utility in the setting of sequential unlearning. Especially, we split the 100 marked users into 5 or 10 groups. At each time, we unlearn each group of users' interactions. In Fig 3, even though the model is sequentially updated, the recommendation performance will not be impacted. This is because RRL will update the FIM to prepare for the next unlearning. With different sizes of groups, the decreases after unlearning are not very significant. Especially, sequentially unlearning the larger groups, e.g., the size of each group is 20, may influence more about the overall performance because the negative impacts may accumulate.

Unlearning different users. Different users will have different contributions to the learning of recommendation models. For instance, unlearning a user with a large number of historical interactions will decrease the recommendation performance because his/her interactions have contributed more to the recommendation model. The results are shown in Fig 4. The sets of marked users are ranked by the average degree of these users, e.g., from the smallest degree to the largest degree. As we can see, when unlearning the most important users, the overall performance will degrade, but the degradation is not larger than 0.05.

Ablation studies I: w/ or w/o CS regularizer. As shown in Table 3, without a CS regularizer, RRL cannot unlearn the influence of malicious users completely, and the overall performance will also be negatively impacted.

Conclusion

In this work, we present the first reverse learning for recommendation systems to enforce the recommendation model to forget private data. As a solution, we propose a framework called RRL that is composed of a reversed personalized ranking objective and a fisher information regularizer. Extensive experiments validate that RRL achieves the guarantee of removal completeness, impressive forgetting efficiency and comparable normal utility. Future work may consider integrating RRL into more complicated recommendation settings such as sequential recommendation, and multi-model recommendation systems. Moreover, it would be meaningful to deploy and validate the proposed framework on real-world platforms to protect the privacy of recommendation systems.

Acknowledgements

We would like to thank the anonymous reviewers for their insightful comments that helped improve the quality of the paper. This work was supported in part by the National Key Research and Development Program (2021YFB3101200), National Natural Science Foundation of China (61972099,U1736208, U1836210, U1836213, 62172104,62172105, 61902374, 62102093, 62102091). Min Yang is a faculty of Shanghai Institute of Intelligent Electronics Systems, Shanghai Insitute for Advanced Communication and Data Science, and Engineering Research Center of Cyber Security Auditing and Monitoring, Ministry of Education, China. Mi Zhang and Min Yang are the corresponding authors.

References

Bourtoule, L.; Chandrasekaran, V.; Choquette-Choo, C. A.; Jia, H.; Travers, A.; Zhang, B.; Lie, D.; and Papernot, N. 2021. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, 141–159. IEEE.

Brophy, J.; and Lowd, D. 2021. Machine unlearning for random forests. In *International Conference on Machine Learning*, 1092–1104. PMLR.

Cai, J.; Liu, Y.; Liu, X.; Li, J.; and Zhuang, H. 2022. Privacy-Preserving Federated Cross-Domain Social Recommendation. In *International Workshop on Trustworthy Federated Learning*, 144–158. Springer.

Chang, C.-C.; Thompson, B.; Wang, H.; and Yao, D. 2010. Towards publishing recommendation data with predictive anonymization. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, 24–35.

Chen, C.; Sun, F.; Zhang, M.; and Ding, B. 2022a. Recommendation unlearning. In *Proceedings of the ACM Web Conference 2022*, 2768–2777.

Chen, C.; Wu, H.; Su, J.; Lyu, L.; Zheng, X.; and Wang, L. 2022b. Differential private knowledge transfer for privacy-preserving cross-domain recommendation. In *Proceedings of the ACM Web Conference 2022*, 1455–1465.

Chen, G.; Zhang, X.; Su, Y.; Lai, Y.; Xiang, J.; Zhang, J.; and Zheng, Y. 2023. Win-Win: A Privacy-Preserving Federated Framework for Dual-Target Cross-Domain Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 4149–4156.

Chen, L.; Wu, L.; Hong, R.; Zhang, K.; and Wang, M. 2020. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 27–34.

Conti, M.; Li, J.; Picek, S.; and Xu, J. 2022. Label-only membership inference attack against node-level graph neural networks. In *Proceedings of the 15th ACM Workshop on Artificial Intelligence and Security*, 1–12.

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.

Graves, L.; Nagisetty, V.; and Ganesh, V. 2021. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 11516–11524.

Guo, C.; Goldstein, T.; Hannun, A.; and Van Der Maaten, L. 2019. Certified data removal from machine learning models. *arXiv preprint arXiv:1911.03030*.

Gupta, V.; Jung, C.; Neel, S.; Roth, A.; Sharifi-Malvajerdi, S.; and Waites, C. 2021. Adaptive machine unlearning. *Advances in Neural Information Processing Systems*, 34: 16319–16330.

He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; and Wang, M. 2020. 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*, 639–648.

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.

Li, H.; Liao, X.; and Carin, L. 2009. Active learning for semi-supervised multi-task learning. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 1637–1640. IEEE.

Liang, D.; Charlin, L.; McInerney, J.; and Blei, D. M. 2016. Modeling user exposure in recommendation. In *Proceedings of the 25th international conference on World Wide Web*, 951–961.

Lin, H.; Chung, J. W.; Lao, Y.; and Zhao, W. 2023. Machine Unlearning in Gradient Boosting Decision Trees. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1374–1383.

Liu, Y.; Ma, Z.; Liu, X.; Liu, J.; Jiang, Z.; Ma, J.; Yu, P.; and Ren, K. 2020. Learn to forget: Machine unlearning via neuron masking. *arXiv preprint arXiv:2003.10933*.

Luo, S.; Xiao, Y.; and Song, L. 2022. Personalized federated recommendation via joint representation learning, user clustering, and model adaptation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 4289–4293.

Meihan, W.; Li, L.; Tao, C.; Rigall, E.; Xiaodong, W.; and Cheng-Zhong, X. 2022. Fedcdr: federated cross-domain recommendation for privacy-preserving rating prediction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2179–2188.

Mobasher, B.; Burke, R.; Bhaumik, R.; and Williams, C. 2007. Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. *ACM Transactions on Internet Technology (TOIT)*, 7(4): 23–es.

Portugal, I.; Alencar, P.; and Cowan, D. 2018. The use of machine learning algorithms in recommender systems: A systematic review. *Expert Systems with Applications*, 97: 205–227.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*.

Sekhari, A.; Acharya, J.; Kamath, G.; and Suresh, A. T. 2021. Remember what you want to forget: Algorithms for

machine unlearning. *Advances in Neural Information Processing Systems*, 34: 18075–18086.

Shen, Y.; and Jin, H. 2014. Privacy-preserving personalized recommendation: An instance-based approach via differential privacy. In *2014 IEEE International Conference on Data Mining*, 540–549. IEEE.

Shokri, R.; Stronati, M.; Song, C.; and Shmatikov, V. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, 3–18. IEEE.

Song, C.; Wang, B.; Jiang, Q.; Zhang, Y.; He, R.; and Hou, Y. 2021. Social recommendation with implicit social influence. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, 1788–1792.

Tang, J.; Wen, H.; and Wang, K. 2020. Revisiting adversarially learned injection attacks against recommender systems. In *Proceedings of the 14th ACM Conference on Recommender Systems*, 318–327.

Tarun, A. K.; Chundawat, V. S.; Mandal, M.; and Kankanhalli, M. 2023. Deep regression unlearning. In *International Conference on Machine Learning*, 33921–33939. PMLR.

Wang, X.; He, X.; Wang, M.; Feng, F.; and Chua, T.-S. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 165–174.

Wu, G.; Hashemi, M.; and Srinivasa, C. 2022. Puma: Performance unchanged model augmentation for training data removal. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 8675–8682.

Wu, J.; Fan, W.; Chen, J.; Liu, S.; Li, Q.; and Tang, K. 2022. Disentangled contrastive learning for social recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 4570–4574.

Wu, J.; Wang, X.; Feng, F.; He, X.; Chen, L.; Lian, J.; and Xie, X. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, 726–735.

Wu, J.; Yang, Y.; Qian, Y.; Sui, Y.; Wang, X.; and He, X. 2023. GIF: A General Graph Unlearning Strategy via Influence Function. In *Proceedings of the ACM Web Conference 2023*, 651–661.

Yan, H.; Li, X.; Guo, Z.; Li, H.; Li, F.; and Lin, X. 2022. Arcane: An efficient architecture for exact machine unlearning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, 4006–4013.

Zenke, F.; Poole, B.; and Ganguli, S. 2017. Continual learning through synaptic intelligence. In *International conference on machine learning*, 3987–3995. PMLR.

Zhang, J.; Askari, H.; Psounis, K.; and Shafiq, Z. 2023. A Utility-Preserving Obfuscation Approach for YouTube Recommendations. *Proceedings on Privacy Enhancing Technologies*, 4: 522–539.

Zhang, M.; Ren, Z.; Wang, Z.; Ren, P.; Chen, Z.; Hu, P.; and Zhang, Y. 2021. Membership inference attacks against recommender systems. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 864–879.