Chinese Journal of Electronics
vol.04, no.04, pp.1– 11, 2022
https://doi.org/123.123.123.123

Research Article

Learning to Unlearn for Bayesian Personalized Ranking via Influence Function

Jundong Chen[1,2*], Honglei Zhang[1,3*], Haoxuan Li[4], and Yidong Li[1,3]

1. *Key Laboratory of Big Data & Artificial Intelligence in Transportation (Beijing Jiaotong University), Ministry of Education, Beijing 100044, China*
2. *School of Cyberspace Science and Technology, Beijing Jiaotong University, Beijing 100044, China*
3. *School of Computer Science and Technology, Beijing Jiaotong University, Beijing 100044, China*
4. *Center for Data Science, Peking University, Beijing 100871, China*

Corresponding author: Yidong Li; Email: ydli@bjtu.edu.cn.
Received December 30, 2023; Accepted January 15, 2025; Published xx xx, xxxx.

Abstract — Learning recommender models from vast amounts of behavioral data has become a mainstream paradigm in recent information systems. Conversely, with privacy awareness grown, there has been increasing attention to the removal of sensitive or outlier data from well-trained recommendation models (known as recommendation unlearning). However, current unlearning methods primarily focus on fully/partially retraining the entire model. Despite considerable performance, it inevitably introduces significant efficiency bottlenecks, which is impractical for latency-sensitive streaming services. While recent efforts exploit efficient unlearning in point-wise recommender tasks, these approaches overlook the partial order relationships between items, resulting in suboptimal performance in both recommendation and unlearning capabilities. In light of this, we explore learning to Unlearn for Bayesian Personalized Ranking (UBPR) via influence function, which relies on a pair-wise ranking loss to model user preferences and item characteristics, making unlearning more challenging than in point-wise settings. Specifically, we propose an influence function-guided unlearning framework tailored for pair-wise ranking models to efficiently perform unlearning requests, which involves unlearning partial order relationships while handling negative samples appropriately during the unlearning process. Besides, we prove that our proposed method can theoretically match the performance of retraining counterparts. Finally, we conduct extensive experiments to validate the effectiveness and efficiency of our model.
Keywords — Recommendation unlearning, Influence function, Bayesian personalized ranking, The right to be forgotten.

I. Introduction

Recommender systems play a crucial role in shaping the online experience of users across various digital services, including social media, entertainment, and e-commerce [1–3]. These systems typically learn a recommender model from historical interactions, with model parameters memorizing user behaviors to provide personalized recommendations [4–8]. Recently, the need to remove certain training data, such as sensitive personal data or outlier data points, from well-trained models has gained prominence due to ethical and legal concerns, such as GDPR [9]. Specifically, from the perspective of individual users, there may be requests to delete specific sensitive interactions from the entire training dataset [10]. Besides, from the system's standpoint, eliminating anomalous or poisoned data injected by attackers upon detection is essential for achieving robust recommendations [11–13]. On this basis, recommendation unlearning emerges as a promising research direction to build the next-generation trustworthy recommender systems [14].

Towards this end, current mainstream recommendation unlearning models mainly exploit straightforward and effective unlearning mechanisms by retraining from scratch with the remaining data, thus allowing the physical removal of requested data from the model [15, 16]. According to the scale of utilized data, retraining methods can primarily fall into two categories: full retraining [17, 18] and partial retraining [19, 20]. Full retraining involves retraining the entire model with all remaining data, despite considerable performance, but it is often time-consuming and computationally expensive [17], limiting its practical applicability. Partial retraining refers to the process of partitioning the entire dataset and the complete model into multiple sections, followed by conducting small-scale retraining on specific partitions to accelerate the unlearning process [19]. However, partial retraining is not always effective compared to full retraining, as both model partitioning and data partitioning can negatively

*Co-first authors.

impact recommendation performance [21]. Besides, the underlying assumption for efficient unlearning through partial retraining is that the requested unlearning data is typically distributed in one or few partitions, which is impractical in real-world recommender systems.

Recently, some pioneering works have explored efficient unlearning at different granularities in recommender tasks. Specifically, Christian *et al.* incorporate adversarial training into the classical variational auto-encoder architecture, introducing a novel model called adversarial variational auto-encoder with multinomial likelihood. This model is designed to eliminate implicit information related to protected attributes, such as genders or ages, while maintaining recommendation performance [22]. Besides, Yuan *et al.* propose a novel federated recommendation unlearning model to explore machine unlearning in federated recommender systems, enabling it to efficiently eliminate the influence of specific clients and complete the recovery process. The inspiration comes from the log-based rollback mechanism used in transactions in database management systems [23]. From another perspective, Zhang *et al.* introduce an efficient recommendation unlearning framework to update the model parameters without retraining by estimating the impact of the requested unlearning data on the target model [21]. All of the above methods aim to strike a reasonable balance between preserving recommendation performance and unlearning efficiency.

Although the above methods can efficiently achieve recommendation unlearning, we argue there are still some limitations in the following aspects. Firstly, existing methods primarily focus on unlearning requests on point-wise collaborative filtering tasks (*e.g.,* mean squared error loss and binary cross-entropy loss, etc.), neglecting the inherent partial order relationships among items. This oversight hinders the optimization of both recommendation performance and unlearning efficiency. Secondly, the mentioned approaches require specialized unlearning method designs for different unlearning scenarios. For instance, using adversarial training to unlearn attribute information or employing rollback mechanisms to unlearn client information. This situation, which requires designing specific methods for each scenario, may limit the generality and scalability of the approach.

To address the above challenges, we aim to propose a unified recommendation unlearning framework that can efficiently handle the unlearning requests while capturing the item's partial order relationships. Specifically, we introduce an influence function-guided unlearning approach designed for pair-wise ranking loss, which considers both the partial order relationships between items and the constraints between negative sample pairs during the unlearning process. Besides, from a theoretical perspective, we prove that, under certain assumptions, the proposed approach is almost equivalent to the retraining method in terms of recommendation performance. Finally, we validate the superiority of the proposed method in terms of recommendation performance and unlearning efficiency through extensive experiments. Overall, the main contributions of this work are listed as follows:

- We propose an influence function-guided unlearning framework tailored for pair-wise ranking loss to efficiently perform unlearning requests, which involves unlearning partial order relationships among items while handling negative samples appropriately during the unlearning process.
- From the theoretical perspective, we prove that, under certain assumptions, the proposed approach is almost equivalent to the retraining counterpart in terms of recommendation performance.
- Extensive experiments on two real-world datasets demonstrate the advantages of our model on effectiveness and efficiency over several state-of-the-art models.

II. Related Work

We briefly review two related fields to this work: matrix factorization [1] and recommendation unlearning [15].

1. Matrix Factorization

Matrix factorization (MF), also known as latent factor model, has been the dominant technique in recommender system community for many decades [4]. The objective of MF is to embed users and items into a shared latent subspace, where the similarities between users and items are computed through inner products. Due to its high capability and scalability, MF has garnered significant attention over the years. Several pioneering works have aimed to enhance MF by integrating it with other advanced models [6, 24, 25]. For instance, He *et al.* introduced neural collaborative filtering (NCF) [6], which integrates a multi-layer perceptron into MF, allowing for improved modeling of user-item interactions through non-linear transformations. Apart from fusing advanced models, some recent works have attempted to incorporate the idea of learning to rank on top of MF to model the partial order relationships among items [5, 26, 27]. For instance, Rendle *et al.* proposed the bayesian personalized ranking (BPR) framework to model the ranking relationships between items from a Bayesian perspective [5], indicating that interacted items by a user should be ranked higher than non-interacted items, thereby achieving fine-grained preference modeling. Subsequent works have attempted to incorporate more complex ordinal relationships on the basis of BPR, such as VBPR [26], DVBPR [27]. In summary, various studies have explored the effectiveness of combining complex models or learning-to-rank techniques to improve vanilla MF.

2. Recommendation Unlearning

Recommendation unlearning is a process designed to eliminate the impact of a specified set of training data upon request from a trained recommender model [15]. According to the degree of unlearning, current state-of-the-art methods

Chinese Journal of Electronics

can be broadly categorized into two groups: exact recommendation unlearning [19, 20, 28] and approximate recommendation unlearning [23, 29]. Exact recommendation unlearning aims to entirely eliminate the influence of the requested data from the recommendation model [19], while approximate recommendation unlearning focuses on achieving forgetting guarantees in a statistical sense [29]. Besides, from the perspective of model training mechanisms, current works can be grouped into three types of unlearning algorithms: data reorganization-based [30, 31], model optimization-based [21, 32], and training mechanism-based recommendation unlearning [16, 19]. Among them, data reorganization-based unlearning models, aim to achieve unlearning by manipulating the distribution of data [30, 33]. For instance, IMCorrect [33] achieves efficient recommendation unlearning by correcting the interaction matrix. The model optimization-based unlearning methods primarily focus on three aspects of the model, such as the loss function [28, 34], influence function [21], and gradient updates [32]. They update the model's parameters to match or approximate the parameter distribution of the retrained model. For example, Zhang *et al.* introduced an efficient recommendation unlearning framework aimed at updating the model parameters without retraining. This is achieved by estimating the impact of the requested unlearning data on the target model [21]. Training mechanism-based methods modify the training structure or pipeline to efficiently achieve unlearning, and specific methods include partial retraining [19, 35, 36], model fine-tuning [18], and federated unlearning [23]. For instance, RecEraser divides the training set into multiple shards and trains a sub-model for each shard. When an unlearning request is received, unlearning can be effectively achieved by only retraining the affected sub-model [19]. Ultrare [35] and RRL [36] also adopt such strategy. Different from these methods, we attempt to introduce an efficient unlearning approach with the notion of pair-wise learning, thus achieving the dual advantages of recommendation performance and unlearning efficiency.

III. Methodology

In this section, we first present the preliminaries of the bayesian personalized ranking model, and then we derive an influence function-based Unlearning framework for Bayesian Personalized Ranking (UBPR).

1. Preliminaries

Matrix Factorization is a popular method in recommender systems, which decomposes a sparse rating matrix into two low-rank matrices that represent the latent feature vectors of users and items, respectively. These latent traits can reflect the user's interests and the attributes of the item, allowing for more accurate and flexible recommendations.

Formally, let u denote a user and i denote an item, then the preferences between users and items can be expressed as $\mathbf{p}_u^T\mathbf{q}_i$, where $\mathbf{p}_u \in \mathbb{R}^K$ denotes the embedding vector of user u, $\mathbf{q}_i \in \mathbb{R}^K$ denotes the embedding vector of item i, and K is the length of the embedding vector. Generally, for for the set of all users $\mathcal{U}$ and the set of all items $\mathcal{I}$, the model optimization objective is as follows:

$$\Theta = \arg\min_{\Theta} \sum_{u,i} (\mathbf{r}_{ui} - \mathbf{p}_u^T\mathbf{q}_i)^2 + \lambda\|\Theta\|^2, \tag{1}$$

where $\Theta = \{\mathbf{P}, \mathbf{Q}\} = \{\{\mathbf{p}_u\}_{u\in\mathcal{U}}, \{\mathbf{q}_i\}_{i\in\mathcal{I}}\}$ is the model parameters (i.e., the set of all user and item embedding vectors) and $\mathbf{R} = \{\mathbf{r}_{ui}\}_{u\times i}$ is the original rating matrix.

Even though matrix factorization is designed for the item prediction task of personalized recommendation, it is not directly optimized for ranking. Meanwhile, it can not capture the inherent partial order relationships among items. Rendle *et al.* introduce a Bayesian Personalized Ranking (BPR), which is based on implicit feedback data [5]. It ranks items by the maximum posterior probability obtained from a Bayesian analysis of the problem, which in turn generates recommendations. Considering the inherent partial order relationships among items, it introduces negative sampling based on the assumption that observed interactions should get higher ranking score than the unobserved ones. Formally, let $\hat{y}_{\boldsymbol{ui}}(\Theta)$ denote $\mathbf{p}_{\boldsymbol{u}}^T\mathbf{q}_i$, the loss function of BPR is as follows:

$$L_{\mathrm{BPR}}(\mathcal{D}|\Theta) = \sum_{(u,i,j)\in\mathcal{D}} -\ln\sigma(\hat{y}_{ui}(\Theta) - \hat{y}_{uj}(\Theta)) + \lambda||\Theta||^2, \tag{2}$$

where $\sigma(\cdot)$ is the sigmoid function and λ is model specific regularization. For dataset $\mathcal{D}$, the elements in each triple instance (u, i, j) represent the user u, the observed item $i \in \mathcal{I}_u$, and the randomly sampled unobserved item $j \in \mathcal{I} \setminus \mathcal{I}_u$ (i.e., negative sample), respectively, where $\mathcal{I}$ denotes the whole item set and $\mathcal{I}_u$ denotes the set of all the items which interact with the specific user u. Minimizing the loss function by stochastic gradient descent (SGD) algorithm, it can generate parameters Θ containing a user embedding matrix and an item embedding matrix. Consequently, for each user u, we can calculate the rating $\hat{y}_{ui}(\Theta)$ over all items to obtain a personalized list, and then recommend items for each user.

2. The Proposed UBPR Model

Given a set of implicit feedback data $\mathcal{D}$, through iterative optimization, the recommender model can obtain the optimum parameters Θ. Due to the reason of privacy or security, the subset $\mathcal{D}_f \subseteq \mathcal{D}$ needs to be removed on both the data level and the recommender model level to obtain another parameters Θ^*, and achieve the same effectiveness as model retraining on the retaining dataset $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$, this process is called recommendation unlearning. Let Θ' denote the parameters after model retraining, the unlearning process can be presented as $\Theta \rightarrow \Theta^* \simeq \Theta'$.

For the majority of matrix factorization models, we only need to remove the effect of each $(u, i) \in \mathcal{D}_f$ on parameters updates. Retraining the model from scratch may be a better

Chinese Journal of Electronics

option when the $|\mathcal{D}_r|$ is relatively small to the point where retraining is costly. However, during the process of updating the parameters of the BPR model, we not only considered the implicit feedback user-item pair (u, i) itself, but also sampled negative feedback item j in order to take into account item's partial order relationship. In this case, retraining the model from scratch will be more time-consuming owing to the increase in the amount of dataset due to sampling. Meanwhile, even if some of the existing unlearning algorithms can cope with the time-consuming problem, they are unable to make a good proof that the sample-related inter-item partial order relationship learned during training is removed from the model. So it is vital to design an efficient unlearning algorithm that approximately or fully attains the effectiveness of retraining.

Formally, for the BPR model, we represent the loss function Eq. (2) for short by $\mathcal{L}(\mathcal{D}|\Theta)$, then the original parameters Θ can be obtained by optimizing the following objective:

$$\Theta = \arg\min_{\Theta} \frac{1}{|\mathcal{D}|}\mathcal{L}(\mathcal{D}|\Theta). \tag{3}$$

After removing subset $\mathcal{D}_f$ from $\mathcal{D}$, the parameters of the retrained model can be obtained by a new optimization objective as follows:

$$\begin{aligned}\Theta' &= \arg\min_{\Theta} \frac{1}{|\mathcal{D}_r|}\mathcal{L}(\mathcal{D}_r|\Theta) \\ &= \arg\min_{\Theta}[\frac{1}{|\mathcal{D}_r|}\mathcal{L}(\mathcal{D}|\Theta) - \frac{1}{|\mathcal{D}_r|}\mathcal{L}(\mathcal{D}_f|\Theta)].\end{aligned} \tag{4}$$

Based on the above formulation, we attempt to utilize the difference between the two optimization objectives to erase the effect of the data to be removed by a closed-form update.

Inspired by influence functions [37], we can efficiently approximate the effect of some particular training points on a model's prediction. By quantitatively calculating this effect, we can obtain these points' contribution to the model updating process, which is known as the influence function. Ultimately, by erasing this impact from the original parameters, we can achieve a comparable effectiveness as retraining. Formally, we first upweight the loss of the subset $\mathcal{D}_f$ by some small ϵ on the original optimization objective (Eq. (3)), which generates a new parameters as follows:

$$\Theta_\epsilon = \arg\min_{\Theta}[\frac{1}{|\mathcal{D}|}\mathcal{L}(\mathcal{D}|\Theta) + \epsilon\mathcal{L}(\mathcal{D}_f|\Theta)]. \tag{5}$$

Due to the influence of the upweighted loss over $\mathcal{D}_f$, the changes in parameters $\Delta_{\epsilon,\Theta}$ can be expressed as follows and its derivation is given by Proof 3.2:

$$\Delta_{\epsilon,\Theta} = \Theta_\epsilon - \Theta = -\epsilon H_\Theta^{-1}\nabla_\Theta\mathcal{L}(\mathcal{D}_f|\Theta), \tag{6}$$

where $H_\Theta = \frac{1}{|\mathcal{D}|}\nabla_\Theta^2\mathcal{L}(\mathcal{D}|\Theta)$ is the Hessian matrix and is assumed to be positive definite (PD).

In the Proof 3.2, we also further prove that when $\epsilon = -\frac{1}{|\mathcal{D}|}$, the parameters of the retrained model Θ' can be approximated by $\Theta_{-\frac{1}{|\mathcal{D}|}}$, and thus we obtain the parameters of the unlearned model Θ^*. This process is as follows:

$$\begin{aligned}\Theta' \approx \Theta^* &= \Theta_{-\frac{1}{|\mathcal{D}|}} \\ &= \Theta + \Delta_{-\frac{1}{|\mathcal{D}|},\Theta} = \Theta + \frac{1}{|\mathcal{D}|}H_\Theta^{-1}\nabla_\Theta\mathcal{L}(\mathcal{D}_f|\Theta).\end{aligned} \tag{7}$$

Using the triplet generated by implicit feedback user-item pairs together with sampled negative feedback items as the data in $\mathcal{D}_f$ to conduct the above computational update, we are able to remove the user-item pairs' effect from the model as well as the learned partial order relationships among items.

However, calculating $\Delta_{-\frac{1}{|\mathcal{D}|},\Theta}$ directly is still costly in terms of time and computational resources due to the calculation of the Hessian matrix and its inverse matrix. We employ the combination of Hessian-vector product (HVP) [38] and developed conjugate gradient (CG) [39] to estimate $\Delta_{-\frac{1}{|\mathcal{D}|},\Theta}$ without explicitly calculating the Hessian matrix and its inverse matrix following previous work [40, 41]. That is, let z_Θ denote $\nabla_\Theta\mathcal{L}(\mathcal{D}_f|\Theta)$, we first obtain HVP as follows:

$$H_\Theta z_\Theta = \frac{1}{|\mathcal{D}|}\nabla_\Theta^2\mathcal{L}(\mathcal{D}|\Theta)z_\Theta = \frac{1}{|\mathcal{D}|}\nabla_\Theta(\nabla_\Theta\mathcal{L}(\mathcal{D}|\Theta)^{\mathrm{T}}z_\Theta), \tag{8}$$

where $H_\Theta z_\Theta$ can be used in the developed CG method for solving the following optimization problem and thus we have

$$H_\Theta^{-1}z_\Theta = \arg\min_t[\frac{1}{2}t^{\mathrm{T}}H_\Theta t - z_\Theta^{\mathrm{T}}t], \tag{9}$$

where the gradient of the optimization objective at its optimal point t^* is supposed to be zero, that is, $H_\Theta t^* - z_\Theta = 0$. Thus we obtain $H_\Theta^{-1}z_\Theta = t^*$.

Proof 3.2 Under the objective presented in Eq. (5), we assume it gradually converges to optimum point during the optimization process, so that the gradient at Θ_ϵ is zero. Then, we have

$$\frac{1}{|\mathcal{D}|}\nabla_\Theta\mathcal{L}(\mathcal{D}|\Theta) + \epsilon\nabla_\Theta\mathcal{L}(\mathcal{D}_f|\Theta) = 0. \tag{10}$$

For the following derivation, we make the assumption that $|\mathcal{D}| \gg |\mathcal{D}_f|$ since it is rare to get an almost magnitude of data moved at the same time in practice. Then the slightly weighted term $\nabla_\Theta\mathcal{L}(\mathcal{D}_f|\Theta)$ does not significantly affect the parameters, thus Θ_ϵ is close to the original parameters Θ. Next we can approximate Eq. (5) by applying Taylor expansion at Θ and then we can drive the following formulations:

$$\begin{aligned}0 =& \frac{1}{|\mathcal{D}|}\nabla_\Theta\mathcal{L}(\mathcal{D}|\Theta) + \epsilon\nabla_\Theta\mathcal{L}(\mathcal{D}_f|\Theta) \\ &+[\frac{1}{|\mathcal{D}|}\nabla_\Theta^2\mathcal{L}(\mathcal{D}|\Theta) + \epsilon\nabla_\Theta^2\mathcal{L}(\mathcal{D}_f|\Theta)](\Theta_\epsilon - \Theta) \\ &+o(\Theta_\epsilon - \Theta),\end{aligned} \tag{11}$$

Chinese Journal of Electronics

where $o(\cdot)$ denotes the infinitesimal term, $\frac{1}{|\mathcal{D}|}\nabla_\Theta \mathcal{L}(\mathcal{D}|\Theta) = 0$ as it is the optimum and minimum parameter of Eq. (3). Furthermore, since $\mathcal{L}(\cdot|\Theta)$ is a cumulative loss function, $\mathcal{D}_f \in \mathcal{D}$ and $|\mathcal{D}| \gg |\mathcal{D}_f|$, $\nabla^2_\Theta \mathcal{L}(\mathcal{D}_f|\Theta)$ is ignorable compared to $\nabla^2_\Theta \mathcal{L}(\mathcal{D}|\Theta)$. Meanwhile, ϵ is a small value, thus $\epsilon\nabla^2_\Theta \mathcal{L}(\mathcal{D}_f|\Theta)$ can be dropped. Then, after ignoring $o(\Theta_\epsilon - \Theta)$, we have

$$\epsilon\nabla_\Theta \mathcal{L}(\mathcal{D}_f|\Theta) + \frac{1}{|\mathcal{D}|}\nabla^2_\Theta \mathcal{L}(\mathcal{D}|\Theta)(\Theta_\epsilon - \Theta) \approx 0. \quad (12)$$

Assuming that the Hessian matrix $H_\Theta = \frac{1}{|\mathcal{D}|}\nabla^2_\Theta \mathcal{L}(\mathcal{D}|\Theta)$ is positive definite (PD), thus further we have the following form, i.e., Eq. (6):

$$\Delta_{\epsilon,\Theta} = \Theta_\epsilon - \Theta = -\epsilon H^{-1}_\Theta \nabla_\Theta \mathcal{L}(\mathcal{D}_f|\Theta). \quad (13)$$

Next, we can choose an adequate value to calculate the difference between the original model parameters Θ and the retrained model parameters Θ' according to Eq. (13). We note that Eq. (4) can be adapted as follows:

$$\Theta' = \arg\min_\Theta \frac{|\mathcal{D}|}{|\mathcal{D}_r|}[\frac{1}{|\mathcal{D}|}\mathcal{L}(\mathcal{D}|\Theta) - \frac{1}{|\mathcal{D}|}\mathcal{L}(\mathcal{D}_f|\Theta)], \quad (14)$$

where the constant coefficient $\frac{|\mathcal{D}|}{|\mathcal{D}_r|}$ can be ignored as it does not affect the process of optimization. Comparing Eq. (5) with Eq. (14), we have the statement that Eq. (5) is equivalent to retraining if $\epsilon = -\frac{1}{|\mathcal{D}|}$. Then the parameters of the retrained model Θ' can be approximated by the unlearned parameters Θ^*, and it can be obtained by one-step update as follows:

$$\Theta' \approx \Theta^* = \Theta + \Delta_{-\frac{1}{|\mathcal{D}|},\Theta} = \Theta + \frac{1}{|\mathcal{D}|}H^{-1}_\Theta \nabla_\Theta \mathcal{L}(\mathcal{D}_f|\Theta). \quad (15)$$

IV. Experiment

In this section, we experimentally verify the effectiveness and efficiency of UBPR while analyzing its performance in detail.

1. Experimental Setup

In this subsection, we introduce the benchmark datasets, the evaluation metrics and the compared methods along with the hyper-parameters settings. All experiments are executed on a GPU server with NVIDIA RTX A5000.

1) Datasets

We conduct extensive experiments on two real-world public datasets: MovieLens and Pinterest. Table 1 summarizes the statistics of the datasets.

`MovieLens`. This is an explicit feedback dataset contributed by GroupLens containing user ratings and tags for movies. Different versions of the MovieLens datasets contain different amounts and scales of users, movies, and ratings. Here we use the implicit version converted by [42] which contains one million ratings, where 1 indicates that the user has rated the item and the opposite is true for 0.

`Pinterest`. This is an implicit feedback dataset contributed by [43] containing images collected by users on the Pinterest platform, which is often used to evaluate the content-based image recommendation task. We use the filtered version created by [42] where only users with at least 20 interactions (pins) are retained. Each interaction indicates whether or not the user has pinned the image.

Table 1 Statistics of the experimented datasets.

Dataset	User#	Item#	Interaction#	Sparsity
MovieLens	6,040	3,706	1,000,209	95.53%
Pinterest	55,187	9,916	1,500,809	99.73%

2) Compared Methods

We take ***Retraining***, the most fundamental and raightforward method to realize machine unlearning, as the essential baseline. Furthermore, we compare two other unlearning methods and the details of baselines are as follows,

- Retrain: Training a model from scratch on the remaining dataset to achieve complete unlearning.
- SISA [16]: It randomly splits the data into shards for isolated training, then performs average aggregation on the results of sub-models to yield the final prediction.
- RecEraser [19]: A recommendation unlearning framework, which comprises balanced data partition and attention-based adaptive aggregation.

3) Evaluation Metrics

The experiment primarily aims to evaluate the efficiency and effectiveness of the UBPR. Here we can directly compare the unlearning time with the running time of other methods to certify the efficiency. For effectiveness, that is, whether or not the UBPR has achieved the unlearning effect, here we take the model retrained from scratch as the benchmark for the effect comparison. We expect UBPR to have comparable performance of the personalized ranking list with retraining, which can be evaluated by Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG).

The HR stresses the recommendation accuracy, that is, whether a user's preferred item is included within the top K of the personalized ranking list (short for $list@K$). The NDCG aims to evaluate the position of the user's preferred item in the $list@K$, where higher score represents higher rank. Since it is time-consuming to rank all items for each user, we refer to previous work [44] that randomly samples 99 uninteracted items and calculates the above metrics of the preferred item among these 100 items (99 uninteracted items and the preferred item itself).

In addition, when comparing the performance of the two methods, we focus more on the overall performance of each model, that is, calculating the mean HR and the mean

Chinese Journal of Electronics

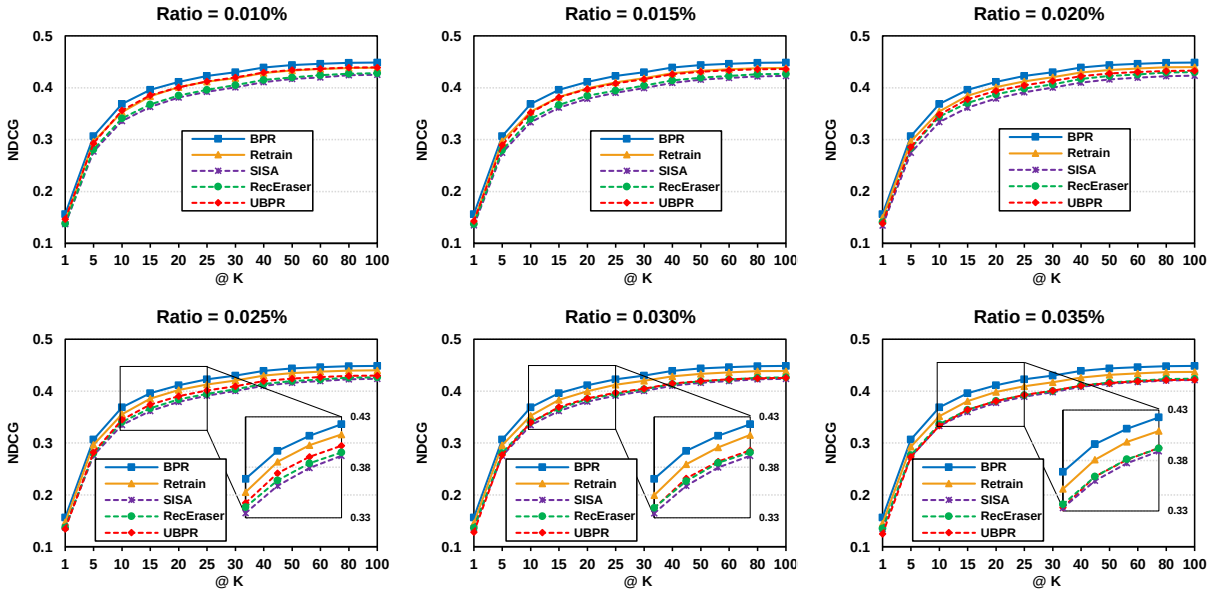

Figure 1 The performance of different models on dataset MovieLens under different unlearning ratios.

NDCG based on all users' $list@K$, which is more statistically meaningful to illustrate the differences between the methods. The value of K in the experiment is taken in the range of $\{1, 5, 10, 20, 25, 30, 40, 50, 60, 80, 100\}$.

4) Hyper-parameters settings

When optimizing the original model, we utilize the Adam optimizer with a batch size of 4096, a learning rate of 0.01, a regularization factor of 0.001, and set the number of epochs to 50, then saving the model under the epoch with the highest mean NDCG on $list@10$ as the original model. We perform all the experiments with the embedding size equal to 8. For Retrain, SISA and RecEraser, the parameters settings are consistent with that of the original model during unlearning. Additionally, we split the data into 5 shards for SISA and RecEraser, where we consider the ideal hypothesis, i.e., the data to be forgotten exists in the same shard. Following the RecEraser settings, we split the data based on user similarity [19]. Next, we unlearn the $\{0.01\%, 0.015\%, 0.02\%, 0.025\%, 0.03\%, 0.035\%\}$ of the total data using different methods, Table 2 shows the size of the two datasets after preprocessing and negative sampling (sampling 4 negative samples for each interaction), as well as the sizes of the data to be forgotten under different ratios.

2. Effectiveness Comparison

In this subsection, we present the experimental results of our proposed unlearning algorithm UBPR and discuss its effectiveness. Figure 1 and Figure 2 illustrate the performance of different models with different unlearning ratios on the two datasets, MovieLens and Pinterest, respectively, with NDCG as the evaluation metric at different K. Here the following noteworthy findings are drawn.

Table 2 Details of the data to be forgotten at different ratios.

Dataset	Size	Ratio	Size
MovieLens	3,976,676	0.01%	396
		0.015%	596
		0.02%	792
		0.025%	992
		0.03%	1,192
		0.035%	1,388
Pinterest	5,782,488	0.01%	620
		0.015%	908
		0.02%	1,216
		0.025%	1,516
		0.03%	1,820
		0.035%	2,120

i) All the unlearning methods can cause some degree of performance damage to the original BPR model, fitting with the instinct that the reduction of the total training samples results in the reduced performance of model.

ii) Compared to SISA and RecEraser, the performance of UBPR model is closeset to that of the retrained model. The differences of the metric NDCG under different K can be within $0.01 \sim 0.03$ on dataset MovieLens, and such differences can be kept within $0.001 \sim 0.003$ on dataset Pinterest, which verifies the effectiveness of our UBPR model.

iii) For our method, the unlearning effect is highly correlated with the size and quality of the dataset. The size of Pinterest is larger than that of MovieLens, and thus the effectiveness is superior with the same unlearning ratio. Additionally, the same observation holds true for SISA and RecEraser, as more data is better for the performance of each sub-model.

iv) As the unlearning ratio increases, our algorithm gradually deviates from the retrained model, this is especially notice-

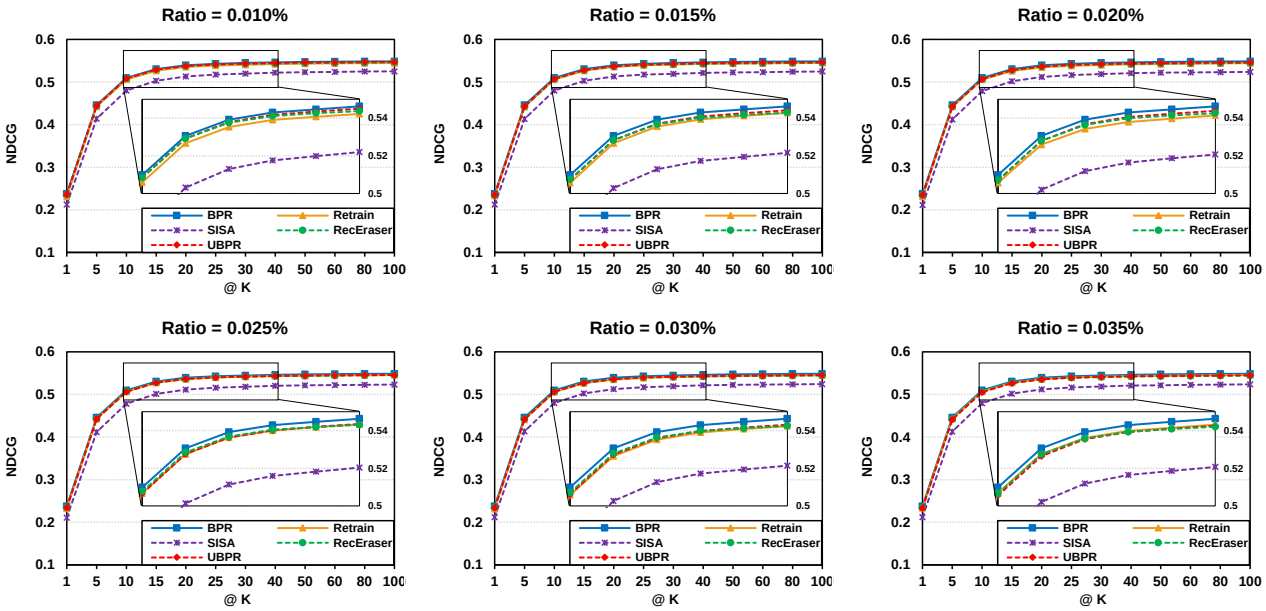

Figure 2 The performance of different models on dataset Pinterest under different unlearning ratios.

able on the dataset MovieLens. This indicates that our model is amount-sensitive and the amount of the data to be forgotten should be as small as possible, which is consistent with practical unlearning settings.

Next, to further analyze the effectiveness of the UBPR, we fix the value of K as 10 and compare different models' performance. Table 3 illustrates the results towards $HR@10$ and $NDCG@10$ of different unlearning algorithms on the two datasets at different unlearning ratios. It presents this two metrics for the original BPR model, the retrained model and the UBPR model, respectively, and denotes the difference between the UBPR model and the retrained model with Δ. From the experimental results, we can find that the difference Δ of $HR@10$ is ignorable (the order of magnitude is 10^{-3}) for both datasets. But for the $NDCG@10$, the Δ on MovieLens tends to enlarge as the unlearning ratio increases (it is acceptable within the order of magnitude equal to 10^{-2}), whereas that on Pinterest can be ignored (the order of magnitude is still 10^{-3}). The results further demonstrate the effectiveness of our approach for recommendation models based on large-scale data.

3. Efficiency Comparison

In this subsection, we compare the running time of different unlearning methods to study the unlearning efficiency. Similarly, we perform this experiment at different unlearning ratios, the results are presented in Figure 3. we can draw the following conclusions:

i) UBPR is significantly faster than all the baselines. For the MovieLens and Pinterest datasets, it speeds up by about $10.0\times$ and $8.0\times$ at all unlearning ratios compared to retraining from scratch, respectively.

ii) SISA and RecEraser can also effectively improve the unlearning speed compared to retraining. However, they both need to store a significant amount of intermediate parameters from training, resulting in a waste of storage resources. Moreover, splitting the model will inevitably harm the overall recommendation performance.

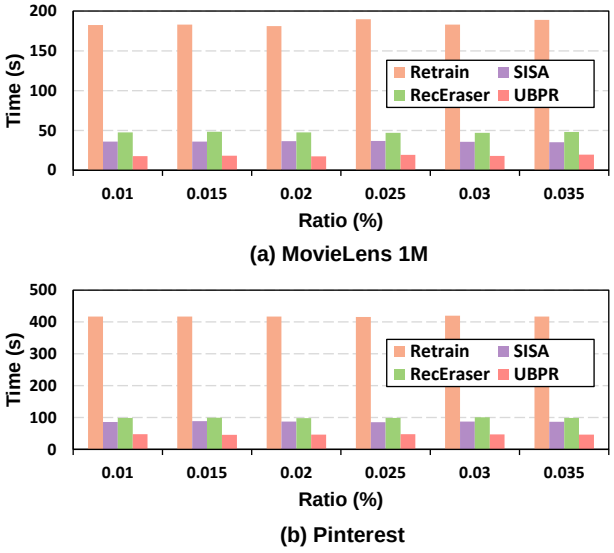

Figure 3 Running time on the two datasets at different ratios.

iii) As the size of the original training dataset increases, all the methods take more time to conduct unlearning as well as ensuring desirable performance. Although the unlearning

Chinese Journal of Electronics

Table 3 Results towards HR@10 and NDCG@10 at different unlearning ratios on the two datasets.

Dataset		MovieLens						Pinterest					
Ratio		0.010%	0.015%	0.020%	0.025%	0.030%	0.035%	0.010%	0.015%	0.020%	0.025%	0.030%	0.035%
HR @10	BPR	0.643	0.643	0.643	0.643	0.643	0.643	0.839	0.839	0.839	0.839	0.839	0.839
	Retrain	0.627	0.623	0.619	0.623	0.616	0.615	0.839	0.836	0.838	0.838	0.838	0.838
	SISA	0.598	0.594	0.595	0.595	0.595	0.594	0.810	0.811	0.809	0.808	0.810	0.809
	RecEraser	0.607	0.606	0.609	0.609	0.607	0.600	0.838	0.839	0.839	0.839	0.838	0.837
	UBPR	0.630	0.625	0.621	0.619	0.616	0.614	0.838	0.837	0.837	0.836	0.836	0.836
	Δ	+0.003	+0.002	+0.002	-0.004	0.000	-0.001	-0.001	+0.001	-0.001	-0.002	-0.002	-0.002
NDCG @10	BPR	0.369	0.369	0.369	0.369	0.369	0.369	0.510	0.510	0.510	0.510	0.510	0.510
	Retrain	0.353	0.354	0.354	0.355	0.352	0.351	0.506	0.506	0.506	0.507	0.506	0.507
	SISA	0.334	0.333	0.334	0.335	0.335	0.333	0.480	0.480	0.479	0.478	0.480	0.479
	RecEraser	0.342	0.340	0.344	0.341	0.340	0.337	0.508	0.507	0.508	0.507	0.507	0.506
	UBPR	0.356	0.352	0.348	0.345	0.340	0.333	0.508	0.507	0.507	0.507	0.506	0.505
	Δ	+0.003	-0.002	-0.006	-0.010	-0.012	-0.018	+0.002	+0.001	+0.001	0.000	0.000	-0.002

time on Pinterest is typically longer than MovieLens, UBPR is still the most efficient method.

iv) The running time of retraining, SISA and RecEraser greatly depends on the size of the remaining dataset and the number of epochs, while the running time of the UBPR depends on the size of the requested subset to be forgotten, the size of the original dataset, and the times of optimization when approximating the influence function with the CG method. The experimental results demonstrate that UBPR strikes a trade-off between accuracy and efficiency in the calculation of influence function.

4. Further Analysis

In this subsection, we further explore the unlearning effect of our proposed method on MovieLens, demonstrating its effectiveness in unlearning the interacted items and partial order relationships among items.

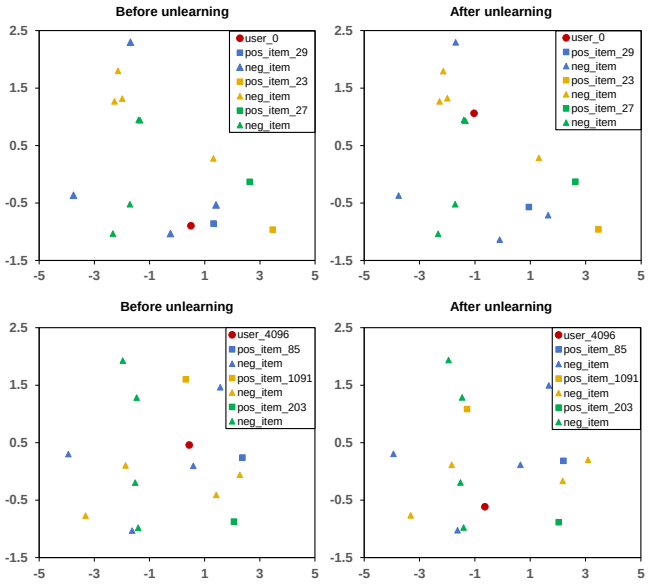

Figure 4 The positional relationship before and after unlearning for user 0 and user 4096.

Visualization of user-item relationships. We first erase a specific user from our original model using UBPR, where we unlearn all the items interacted with this user, together with the sampled negative samples on these items. Since both user and item embedding matrix of the unlearned model are in the same latent space, we perform dimensionality reduction converting them to 2-dimensional level by t-SNE. Subsequently, we present the positional relationship among the unlearned user, user's interacted positive items and the sampled negative ones on low-dimensional space.

Figure 4 shows the results before and after unlearning for users with ID 0 and ID 4096, respectively, where user 0 has 52 interacted items, user 4096 has 29 interacted items and 4 negative samples were sampled for each interacted item. For a clear presentation, here we just randomly show three interacted items with the sampled negative items for each user. In the figure, the red dots denote the user, the little rectangles denote the user's interacted items (positive items), and the little triangles denote the negative items sampled for the positive one, where different colors represent different sets of positive and negative items. The IDs of the users and the interacted items are illustrated in the legends.

Generally, the higher the user-item score, the closer the item embedding vector will be to the user embedding vector in the latent space. BPR is based on the assumption that observed interactions should get higher ranking score than the unobserved ones, that is, the user should be closer to the positive items and as far away from the negative ones as possible in the latent space. As shown in the figure, the user before unlearning is closer to the positive items and farther away from the negative ones. Also there are noticeable distances between the positive and negative items, demonstrating the partial order relationships between items learned during the BPR training process. After unlearning the user using UBPR, the user's position is offset significantly, and there is no obvious preference between the user and either positive or negative items on distance. Conversely, from a statistical perspective, users are observed to be closer to negative samples while being more distant from positive samples. Meanwhile, some positive items also tend to be closer to negative ones. This

Chinese Journal of Electronics

clearly demonstrates that our UBPR method achieves the objective of unlearning by disrupting the existing partial order relationships, thereby pulling negative samples closer while pushing positive samples further away from the target user.

Quantification of user-item relative relationships. To comprehensively evaluate the effectiveness of our method on partial order relationships, we propose a quantitative metric called Positive-Negative Preference Ratio (PNPR) as follows:

$$PNPR = \frac{1}{m}\sum_{j=1}^{m}[\frac{1}{n}\sum_{i=1}^{n}\frac{d_i^{neg}}{d_j^{pos}}], \tag{16}$$

where d^{pos} and d^{neg} denote the distances of the positive and negative item embedding to the user embedding, respectively. Each user interacts with m positive items and samples n negative samples for each positive item. PNPR quantifies how much a given user prefers positive and negative items. Generally, a higher PNPR indicates a clearer differentiation between positive and negative samples. Figure 5 illustrates the PNPR before and after unlearning of the randomly selected users. From the results, it can be seen that users relatively prefer positive items (PNPR>1) before unlearning. Instead, users have no significant preference for positive and negative items after unlearning with either Retrain or UBPR (PNPR≈1). Additionally, our method achieves nearly consistent effects with Retrain, further demonstrating the effectiveness of UBPR for unlearning partial order relationships.

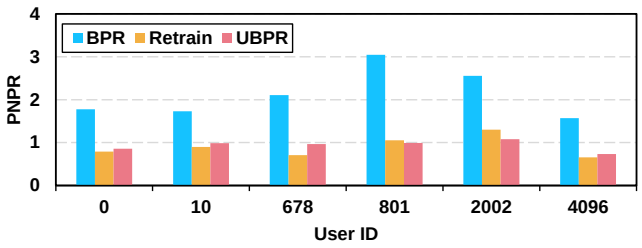

Figure 5 The PNPR before and after unlearning of randomly selected users.

Comparison of unlearning effects on parameters. We adopt previously used Weight Distance [30] to quantitatively measure the unlearning effect of our method. The distance between the parameters of two models helps to further understand the difference in the amount of information between the models. Here, we take the l_2 distance between the parameters of models as Weight Distance. Figure 6 illustrates the distance between the original model before unlearning and the initialized model without any training (O/I), the distance between the retrained model and the original model (R/O), and that between other unlearned models and the original model (S/O for SISA, E/O for RecEraser and U/O for UBPR) under different unlearning ratios on MovieLens. We can see that there is a significant model distance of O/I, since the original model is trained on the initialized model and captures a rich amount of information. Both retraining and unlearning make the model slightly differ from the original model, indicating changes in the amount of information. All unlearning methods achieve comparable results to retraining at parameter level. Notably, the U/O distances tend to be close to the R/O distances, which indicates similar changes in the amount of model information, further demonstrating the unlearning effect of our proposed UBPR.

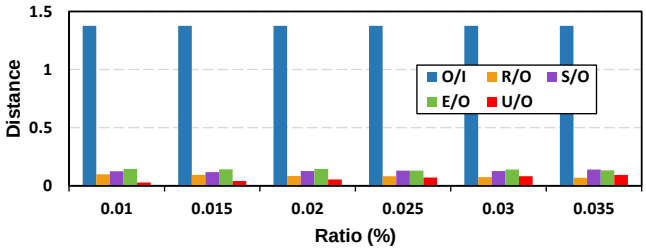

Figure 6 The weight distances between the parameters of models under different unlearning ratios. I,O,R denote the initialized model, original model and retrained model, respectively. S,E,U denote the unlearned model using SISA, RecEraser and UBPR, respectively.

Evaluation under different number of negative samples. The model can capture the partial order relationships between positive and negative items due to the negative sampling mechanism, thus enhancing its performance. To that end, we have further explored the unlearning effectiveness under different number of negative samples, with results shown in Figure 7. For model performance, it is optimal when the number of negative samples is 4. Under-sampling is not conducive to capturing the partial order relationships among items, while over-sampling undermines the overall performance. For unlearning effect, UBPR performs consistently with Retrain under different number of negative samples, evidencing the effectiveness and generality of our method.

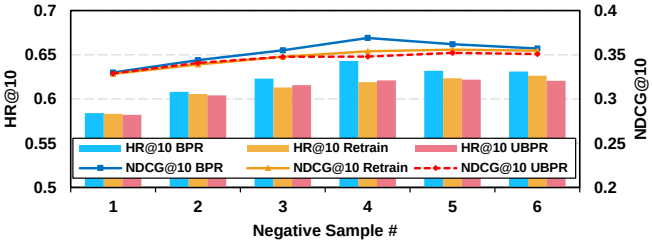

Figure 7 The unlearning effectiveness under different number of negative samples.

V. Conclusion and Future Work

In this work, we propose an efficient and effective unlearning framework tailored for pair-wise ranking models guided by influence function. From a theoretical perspective, we analyze how our proposed method eliminates the effect of interactions to be forgotten along with the sampled negative

Chinese Journal of Electronics

items from the model, and also demonstrate that under certain assumptions, it is effective in line with retraining. We then conduct extensive experiments to verify the effectiveness and efficiency of our proposed algorithm UBPR. The results indicate that the differences of the metric (HR, NDCG) between UBPR and retraining at different unlearning ratios on both datasets are mostly within the order of magnitude equal to 10^{-3} which can be ignorable because of the training stochasticity. Moreover, with such close results, UBPR is about $10.0\times$ and $8.0\times$ faster than retraining on the two datasets, respectively. Additionally, we further analyze how the UBPR eliminates the partial order relationships between items learned during training. By visualizing the unlearning results after the dimensionality reduction, we notice that the preferences between users and items, and the partial order relationships between items are blurred. In conclusion, we theoretically and experimentally demonstrate the effectiveness and efficiency of the UBPR and its advantages over several state-of-the-art models.

There are still a few research directions for future work. First, despite the obvious advantages of the UBPR over retraining, more efficient recommendation unlearning algorithms still need to be explored to cope with the situation where the unlearning requests are frequently submitted. We think that further improving the calculation techniques of the influence function or advanced sampling mechanisms [45] may be a good direction. Besides, recommendation unlearning reflects the problem of robustness of recommender systems, and how to build the trustworthy recommender systems in the future is still a valuable research topic.

Acknowledgements

This work was supported in part by the Fundamental Research Funds for the Central Universities (2023JBZY031), in part by the National Natural Science Foundation of China under Grant (U2268203), in part by Beijing Natural Science Foundation (L221011).

References

[1] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges", *ACM Comput. Surv.*, vol.47, no.1, artilce no.45, 2014.

[2] H. Zhang, X. Zhou, Z. Shen, and Y. Li, "Privfr: Privacy-enhanced federated recommendation with shared hash embedding", *IEEE Transactions on Neural Networks and Learning Systems*, in press, 2024.

[3] H. Zhang, F. Luo, J. Wu, X. He, and Y. Li, "LightFR: Lightweight federated recommendation with privacy-preserving matrix factorization", *ACM Trans. Inf. Syst.*, vol.41, no.4, pp.1–28, 2023.

[4] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems", *Computer*, vol.42, no.8, pp.30–37, 2009.

[5] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback", *arXiv preprint arXiv:1205.2618*, in press, 2012.

[6] X. He, L. Liao, H. Zhang, L. Nie, *et al.*, "Neural collaborative filtering", in *International World Wide Web Conference*, artilce no.173–182, 2017.

[7] H. Zhang, S. Wang, H. Li, C. Zheng, *et al.*, "Uncovering the propensity identification problem in debiased recommendations", in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, IEEE, pp.653–666, 2024.

[8] H. Zhang, H. Li, J. Chen, S. Cui, *et al.*, "Beyond similarity: Personalized federated recommendation with composite aggregation", *arXiv preprint arXiv:2406.03933*, in press, 2024.

[9] P. Voigt and A. Von dem Bussche, "The eu general data protection regulation (gdpr)", *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, vol.10, no.3152676, pp.10–5555, 2017.

[10] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets", in *IEEE Symposium on Security and Privacy*, pp.111–125, 2008.

[11] B. Perozzi, M. Schueppert, J. Saalweachter, and M. Thakur, "When recommendation goes wrong: Anomalous link discovery in recommendation networks", in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.569–578, 2016.

[12] Y. Wang, Y. Liu, Q. Wang, C. Wang, and C. Li, "Poisoning self-supervised learning based sequential recommendations", in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.300–310, 2023.

[13] Q. Zhang, W. Ma, Y. Wang, Y. Zhang, *et al.*, "Backdoor attacks on image classification models in deep neural networks", *Chinese Journal of Electronics*, vol.31, no.2, pp.199–212, 2022.

[14] S. Wang, X. Zhang, Y. Wang, and F. Ricci, "Trustworthy recommender systems", *ACM Transactions on Intelligent Systems and Technology*, in press, 2022.

[15] T. T. Nguyen, T. T. Huynh, P. L. Nguyen, A. W.-C. Liew, *et al.*, "A survey of machine unlearning", *arXiv preprint arXiv:2209.02299*, in press, 2022.

[16] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, *et al.*, "Machine unlearning", in *IEEE Symposium on Security and Privacy*, IEEE, pp.141–159, 2021.

[17] Y. Zhang, F. Feng, C. Wang, X. He, *et al.*, "How to retrain recommender system? a sequential meta-learning method", in *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.1479–1488, 2020.

[18] J. Kim and S. S. Woo, "Efficient two-stage model retraining for machine unlearning", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.4361–4369, 2022.

[19] C. Chen, F. Sun, M. Zhang, and B. Ding, "Recommendation unlearning", in *Proceedings of the ACM Web Conference*, pp.2768–2777, 2022.

[20] Y. Li, C. Chen, X. Zheng, J. Liu, and J. Wang, "Making recommender systems forget: Learning and unlearning for erasable recommendation", *Knowledge-Based Systems*, in press, artilce no.111124, 2023.

[21] Y. Zhang, Z. Hu, Y. Bai, F. Feng, *et al.*, "Recommendation unlearning via influence function", *arXiv preprint arXiv:2307.02147*, in press, 2023.

[22] C. Ganhör, D. Penz, N. Rekabsaz, O. Lesota, and M. Schedl, "Unlearning protected user attributes in recommendations with adversarial training", in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.2142–2147, 2022.

[23] W. Yuan, H. Yin, F. Wu, S. Zhang, *et al.*, "Federated unlearning for on-device recommendation", in *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pp.393–401, 2023.

[24] D. Agarwal and B. C. Chen, "Flda: Matrix factorization through latent dirichlet allocation", in *ACM International Conference on Web Search and Data Mining*, pp.91–100, 2010.

[25] J. Lin, Z. Ye, H. Zhao, and L. Fang, "Deephgnn: A novel deep hypergraph neural network", *Chinese Journal of Electronics*, vol.31, no.5, pp.958–968, 2022.

[26] R. He and J. McAuley, "Vbpr: Visual bayesian personalized ranking from implicit feedback", in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol.30, 2016.

[27] W. Kang, C. Fang, Z. Wang, and J. McAuley, "Visually-aware fashion recommendation and design with generative image models", in *IEEE International Conference on Data Mining*, IEEE, pp.207–216, 2017.

[28] Y. Li, C. Chen, X. Zheng, Y. Zhang, *et al.*, "Making users indistinguishable: Attribute-wise unlearning in recommender systems", in *Proceedings of the 31st ACM International Conference on Multimedia*, pp.984–994, 2023.

[29] Y. Li, C. Chen, X. Zheng, Y. Zhang, *et al.*, "Selective and collaborative influence function for efficient recommendation unlearning", *Expert Systems with Applications*, vol.234, artilce no.121025, 2023.

[30] A. K. Tarun, V. S. Chundawat, M. Mandal, and M. Kankanhalli, "Fast yet effective machine unlearning", *IEEE Transactions on Neural Networks and Learning Systems*, in press, 2023.

[31] Y. Cao and J. Yang, "Towards making systems forget with machine unlearning", in *IEEE Symposium on Security and Privacy*, IEEE, pp.463–480, 2015.

[32] Y. Wu, E. Dobriban, and S. Davidson, "Deltagrad: Rapid retraining of machine learning models", in *International Conference on Machine Learning*, PMLR, pp.10355–10366, 2020.

[33] J. Liu, D. Li, H. Gu, T. Lu, *et al.*, "Recommendation unlearning via matrix correction", *arXiv preprint arXiv:2307.15960*, in press, 2023.

[34] M. Du, Z. Chen, C. Liu, R. Oak, and D. Song, "Lifelong anomaly detection through unlearning", in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pp.1283–1297, 2019.

[35] Y. Li, C. Chen, Y. Zhang, W. Liu, *et al.*, "Ultrare: Enhancing receraser for recommendation unlearning via error decomposition", *Advances in Neural Information Processing Systems*, vol.36, 2024.

[36] X. You, J. Xu, M. Zhang, Z. Gao, and M. Yang, "Rrl: Recommendation reverse learning", in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol.38, pp.9296–9304, 2024.

[37] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions", in *International Conference on Machine Learning*, PMLR, pp.1885–1894, 2017.

[38] B. A. Pearlmutter, "Fast exact multiplication by the hessian", *Neural Computation*, vol.6, no.1, pp.147–160, 1994.

[39] N. Agarwal, B. Bullins, and E. Hazan, "Second-order stochastic optimization for machine learning in linear time", *The Journal of Machine Learning Research*, vol.18, no.1, pp.4148–4187, 2017.

[40] J. Martens *et al.*, "Deep learning via hessian-free optimization.", in *International Conference on Machine Learning*, vol.27, pp.735–742, 2010.

[41] K. Wu, J. Shen, Y. Ning, T. Wang, and W. H. Wang, "Certified edge unlearning for graph neural networks", in *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp.2606–2617, 2023.

[42] X. He, L. Liao, H. Zhang, L. Nie, *et al.*, "Neural collaborative filtering", in *Proceedings of the International Conference on World Wide Web*, pp.173–182, 2017.

[43] X. Geng, H. Zhang, J. Bian, and T.-S. Chua, "Learning image and user features for recommendation in social networks", in *Proceedings of the IEEE International Conference on Computer Vision*, pp.4274–4282, 2015.

[44] A. M. Elkahky, Y. Song, and X. He, "A multi-view deep learning approach for cross domain user modeling in recommendation systems", in *Proceedings of the International Conference on World Wide Web*, pp.278–288, 2015.

[45] M. Guang, C. Yan, G. Liu, J. Wang, and C. Jiang, "A novel neighborhood-weighted sampling method for imbalanced datasets", *Chinese Journal of Electronics*, vol.31, no.5, pp.969–979, 2022.

Jundong Chen is currently a Ph.D. candidate at the School of Cyberspace Science and Technology, Beijing Jiaotong University. His research interest covers machine unlearning and recommender systems. (Email: jundongchen@bjtu.edu.cn)

Honglei Zhang is currently a Ph.D. candidate at the School of Computer Science and Technology, Beijing Jiaotong University. His research interest covers recommender systems and privacy protection. (Email: honglei.zhang@bjtu.edu.cn)

Haoxuan Li is currently a Ph.D. candidate in Center for Data Science, Peking University. His research interests cover causal inference and machine learning theory, recommender system debiasing and fairness, out-of-distribution generalization and data fusion. He has over 20 publications as first author appeared in several top conferences such as ICML, NeurIPS, ICLR, SIGKDD, WWW, AAAI, and IJCAI, journals including TKDE and TORS. Moreover, he has been served as the PC member for several top conferences including ICML, NeurIPS, ICLR, SIGKDD, AAAI, IJCAI, and invited reviewer for prestigious journals such as TOIS, TKDE, TNNLS, TKDD, IPM. (Email: hxli@stu.pku.edu.cn)

Yidong Li is the Vice-Dean and a full Professor in the School of Computer Science and Technology at Beijing Jiaotong University. He received the B.Eng. degree in computer science from Beijing Jiaotong University, and the Master of Information Technology and Ph.D. degrees from the University of Adelaide, in 2006 and 2011, respectively. With main research interests in big data analysis, privacy preserving computing, advanced computing, and intelligent transportation system, he has published more than 200 papers including over 80 papers in international journals such as a variety of IEEE and ACM Transactions. He has also co-authored/coedited 5 books (including proceedings) and contributed several book chapters. He served on the editorial board of numerous journals and chaired several conferences. (Email: ydli@bjtu.edu.cn)