GD-REC: Data-Free Learning of Knowledge Distillation for Recommender System

Li-e Wang[1,2] **Yutian Zheng**[2] **Xianxian Li** *[1,2] **Yange Guo** [2] **Yan Bai**[3] **Yuan Liang**[2]

[1] Guangxi Key Lab of Multi-source Information Mining and Security,Guangxi Normal University
[2] School of Computer Science and Engineering,Guangxi Normal University, Guilin 541004, China
[3] School of Engineering and Technology,University of Washington Tacoma, Tacoma, WA, USA
Wanglie@gxnu.edu.cn,zytiansoft@126.com,lixx@gxnu.edu.cn

Abstract

Recommender systems have begun to use knowledge distillation for balancing privacy guarantee and lightweight requirements with recommended precision. They can extract knowledge from a teacher model with high parameters and high latency to train a student model with low parameters and low inference time while maintaining considerable performance. However, the current technology still faces the two challenges. First, most methods only focus on how student models can better imitate the results of the teacher model, but in reality, teacher models' performance is often not as good as expected due to the problem of data sparseness. Second, knowledge distillation is still challenging due to user privacy protection requirements. Aiming at the above problems, we propose a new **REC**ommendation framework with **G**enerated Adversarial Network and Knowledge **D**istillation called GD-REC. GD-REC use the generative adversarial network to generate a recommendation list and reconstruct the knowledge distillation loss function to process data. In this process, our framework generates a recommendation list to alleviate the problem of data sparseness, and more efficiently transfer knowledge to a student model for training. Specifically, the student model does not use the original users' dataset for training. As a result, we strengthen user privacy protection. Extensive experimental results on public datasets show that GD-REC outperforms that existing methods in terms of accuracy and efficiency.

Introduction

While the amount of data on the Internet is increasing, users are overwhelmed by excess information (Järvelin and Kekäläinen 2002). This phenomenon is called information overloading. Recommender system, being an effective tool to solve information overloading, has been widely used in many areas (He et al. 2017; Batmaz et al. 2019).

Recommender systems based on neural network models often use significant computing resources and rely on large-scale trainings. Effects they bring are naturally apparent, but the cost is very high (Zhang et al. 2019a). Many existing recommender systems are associated with high-level parameters (Kweon, Kang, and Yu 2021; Li et al. 2016). The recommended model parameters are typically one or more orders of magnitude larger than the traditional models. The computation cost is also much higher. Therefore, how to deploy a practical recommendation model that is suitable for a wild application, such as on a real-time website or in a resource-constrained environment, is very challenging.

To solve this problem, many existing works have begun to apply knowledge distillation (Tang and Wang 2018; Lee et al. 2019; Kweon, Kang, and Yu 2021). It is a model-independent technology that can use a high-performance and high-complexity teacher model to train a simpler and more efficient student model. It, basically, is divided into two steps: first, a teacher model needs to be trained with the interaction between users and items. This dataset has a binary label: 1 means that the user purchases or clicks on an item; otherwise, it means there is no relationship. Next, the teacher model's predictions ("soft labels") and the original data are used to train a student model. Student models after knowledge distillation training can not only achieve the same performance as the teacher model but also has lower parameters and recommending delay (Hinton, Vinyals, and Dean 2015).

The core idea is to use a teacher model to transfer the knowledge to a student model, thereby accelerating and improving the training in the student model. The knowledge includes implicit preferences for interaction between users and items, and more importantly, it conveys potential feedback between users and items. The feedback is often not learned by simple student models. Through the additional supervision of the teacher model, several current methods achieve a performance comparable to the teacher model with a shorter reasoning time.

However, the existing methods still have limitations. On one hand, a fundamental prerequisite in knowledge distillation is that a teacher model with excellent performance is needed. Through the guidance of an excellent teacher model, the student model after knowledge distillation can have considerably good performance. The problem of data sparseness in recommender system often leads us to fail training an excellent teacher model (Li et al. 2016). Due to the lack of interaction between users and items makes the knowledge transmitted by a teacher model incomplete. It is unable to identify the user's potential preferences (Mishra and Marr 2017), resulting in a decrease of accuracy. It is essential to alleviate the problem of data sparsity and strengthen the train-

*Corresponding author

ing of teacher models.

On the other hand, for an effective recommender systems, another challenge is privacy protection (Chen et al. 2018; Li et al. 2019). The new bill of General Data Protection Regulation introduced by the European Union indicates that supporting user data security management will be a global trend, especially for recommender systems. Users do not want the goods they have purchased to be leaked (Goodfellow et al. 2014). To meet this challenge, we propose a knowledge distillation scheme without being in touch with users' original data, instead, our student recommendation model uses generated data and knowledge distillation for training.

We propose a new data-free **REC**ommendation framework called GD-REC. The GD-REC combines **G**enerative Adversarial Networks and Knowledge **D**istillation to alleviate the problem of data sparsity, thereby reducing recommended model parameters and achieving large-scale deployment. Unlike most existing recommender system methods simply based on knowledge distillation technology, we incorporate generative adversarial networks into the knowledge distillation training process. Specifically, using a generative network to generate data similar to the user distribution can effectively alleviate data sparsity and make the teacher model more accurate. Moreover, the generated data list is used for knowledge distillation training so as to protect the privacy of users data. The results show that the student model trained with GD-REC can achieve better performance as compared with the other existing methods.

The proposed GD-REC framework can be applied to many open-world scenarios because it alleviates the problem of data sparseness and trains teacher models with higher performance, and enhance user privacy protection. The purpose of framework is to keep the balance of accuracy, efficiency, and privacy in the recommendation system. The main contributions of our work are summarized as follows:

- We developed a novel knowledge distillation recommendation framework, GD-REC. The GD-REC is suitable for lightweight recommended scenarios. It reduces model parameters and model recommending time, and thus, facilitates a large-scale deployment of recommendation models.
- Different from traditional knowledge distillation techniques, GD-REC uses generative adversarial networks for training. It alleviates the problem of data sparsity and better trains excellent teacher models.
- Since student models only use the generative adversarial networks to generate recommendation list data and realize knowledge distillation recommendation for training without being in touch with users' original dataset, user privacy is protected.
- Through extensive experiments on real datasets, the advantages of the GD-REC framework is verified: our experimental results show that GD-REC is more accurate than the current state-of-the-art knowledge distillation methods measured by HR and NDCG.

Related Work

Knowledge distillation can be applied to the recommendation as proposed by Hinton (Hinton, Vinyals, and Dean 2015). The key idea is to transfer teacher models' knowledge learning to student models, and compress the model. The Ranking Distillation(RD) proposed by Tang et al (Tang and Wang 2018) is a novel method of knowledge distillation recommendation, which can make student model pay more attention to the top-K list output by a teacher model, allowing the student model to learn not only from labelled data but also from unlabeled data. Collaborative Distillation(CD) (Lee et al. 2019) improves the RD to disregard the soft target information in a distillation process, allowing students to imitate the teacher model's logit prediction score and the item with higher ranking recommendation. Kweon et al (Kweon, Kang, and Yu 2021) proposed Bidirectional Distillation(BD), which recognizes that students can also perform better than teachers. Trained in the bidirectional way, both teacher and student are improved compared to when being trained separately. The current knowledge distillation recommendation methods still have key limitations. Student models pay more attention to the top-K list of teacher models. However, due to the sparse data of a recommender system, the teacher recommendation list may not describe user preferences. It is critical to perfecting user portraits and building a better and powerful teacher model in order to distil better. In reality, training data sets often have security restrictions or may not be fully available. Therefore, it is necessary to study the recommended model compression method in the absence of data.

In this paper, we focus on the problems of data sparsity and privacy in lightweight recommended scenarios. We propose a new knowledge distillation recommendation framework named GD-REC. The GD-REC recommendation framework allows the final recommendation model to achieve distillation training without being in touch with the users' original data. The GD-REC adopts the idea of a generative adversarial network to alleviate data sparseness. It trains a high-performance student model from a teacher. Furthermore, it reduces the parameters of the recommended model and achieves a large-scale model deployment.

Our GD-REC Framework

We propose the GD-REC framework that student model does not use original user data, instead, learns from teacher predictions and generated data. The GD-REC method has two critical components: 1) generating user data by the adversarial networks, and 2) transferring the knowledge distillation technique between the teacher's prediction and the student's prediction.

Generating Recommendation List

Our generative adversarial network framework is constructed based on CGAN (Mirza and Osindero 2014), in which generator(G) and discriminator(D) are inputs based on user conditions. It can meet the needs of users' personalized recommendations. The two inputs to the generator are the vector C representing items purchased by a user and random noise Z. G will generate an n-dimensional fake purchase vector GF.

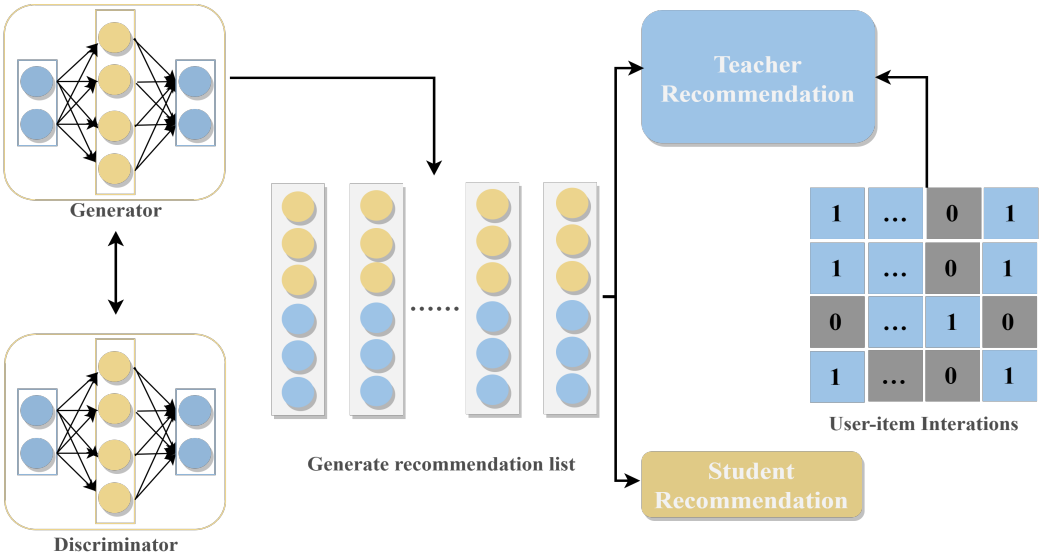


Figure 1: Illustration of GD-REC framework. On the left, generating a recommendation list related to user preferences and interests. On the right, knowledge distillation based on the generated recommendation list, and transferred knowledge extracted from the teacher's prediction.

At the same time, the discriminator gives its binary discriminative value of true or false according to the input *GF*. Both G and D use multilayer neural networks, which are parameterized by ϕ and θ respectively. Each value in the vector *GF* is the probability of the user interacting with the item. D outputs a binary judgment value $\{1, 0\}$. We use stochastic gradient descent and back propagation to train G and D, and alternately update their parameters by ϕ and θ during the training process. When one of them is updated, the other will be fixed. Users vector is used for adversarial training, and recommendation list is output by iterative training.

In the overall process, our generative adversarial networks are consistent with traditional generation networks. However, the traditional generation networks cannot be directly used in the recommender systems scenario because its fundamental goal is to generate real data. The Red Green Blue-based image representation is different from the binary interaction vector in a recommendation system. When generator(G) tries to generate a fake purchase vector that can deceive discriminator(D), the most straightforward and practical method is to generate a binary interaction vector with output predictions. This result is contrary to reality. It cannot correctly capture the user's personalized preferences and generate a suitable top-*K* recommendation list. To solve this problem, a mask mechanism is added.

The main idea of the mask mechanism consists of two parts: first, a new sampling method is adopted. We aim to let the generator perform negative sampling. Each individual user's data in the interaction matrix that has not interacted between the user and the item is randomly selected as a negative sample. In our experiment, the random selection ratio is controlled at 30%. We train the generator to make its prediction score for negative samples to approach zero. Second, in each training iteration, the *GF* output by the generator will perform a dot product operation with the vector E_u.

$$i = GF_u \odot E_u \tag{1}$$

Where the value of the vector E_u is a binary vector. If there is an interaction between the user and the item, the value is 1. Otherwise, it is 0. Our goal is to better guide the vector generated by the generator to make the purchased item score be closer to 1, and the unpurchased vector to approach 0. Therefore, it can better simulate the real user data and alleviate data sparsity. After doing so, only the output of G is more in line with user preferences to help the learning of G and D.

$$\begin{aligned} J^D = -\sum u \left(\log D\left(GF_u | C_u\right) + \right. \\ \left. \log\left(1 - D\left(GF_u \odot \left(E_u + K_u\right) | C_u\right)\right)\right. \end{aligned} \tag{2}$$

$$J^G = \sum u \left(\log\left(1 - D\left(GF_u \odot \left(E_u + K_u\right) | C_u\right)\right)\right) \tag{3}$$

Where C_u represents the input user's actual purchase vector, K_u represents an *n*-dimensional indication vector. If the user interacts with the item, K_u represents 1; otherwise it is 0.

Data-Free in Knowledge Distillation

We design a knowledge distillation method without original data to transfer knowledge. Similar to the traditional knowledge distillation method, the training of model requires two steps. First of all, a teacher uses the generated recommendation list and the binary training set for simultaneous training, and obtains a predicted preference score ("soft target") of the teacher model for the items in the generated list. Second, the student network uses the recommendation list and the teacher's soft target to train a smaller student model. In this process, the teacher model can make reasonable use of

Algorithm 1: Training Student Model

Input: *GF*, Original datasets *R*, θ_T and θ_S
Output: θ_S

```
Initialization θ_T and θ_S
Train θ_T with original datasets
Train θ_T and θ_S with generate recommendation list
while no converged do
    for u, i ∈ GF do
        Get the prediciton of θ_T
        Compute L_GF(θ_S) and L_GD(θ_S; θ_T)
        Update θ_S using Equation (5)
    end for
end while
return θ_S
```

the generated list to alleviate the challenge caused by data sparsity. Consequently, it accelerates student learning. The results show that the teachers trained with GD is better than the traditional KD. Based on this idea, we reformulate the teacher model and student model loss as follows:

$$\mathcal{L}_T(\theta_T) = \mathcal{L}_{CF}(\theta_T) + \mathcal{L}_{GF}(\theta_T) \tag{4}$$

$$\mathcal{L}_S(\theta_S) = \mathcal{L}_{GF}(\theta_S) + \lambda_{T\to S}\mathcal{L}_{GD}(\theta_S;\theta_T) \tag{5}$$

We design an improved collaborative filtering loss function to better use the recommendation list generated by the generative adversarial network. Current knowledge distillation methods recommend using raw data to calculate the cross-entropy loss between the predicted label and the hard label. This method may not be optimistic in real world scenarios. Additionally, traditional knowledge distillation does not consider user privacy. We believe that, for now, the use of generated data to make model predictions can help protect user privacy, and speed up student model training, allowing student models to focus on accurate results.

In most cases, the overall performance of a teacher is better than that of a student, so the teacher's prediction is more reliable than the student. Therefore, students still have to follow the teacher's prediction in the generated recommendation list. We formalize the distillation loss of the teacher's knowledge transfer to the student. Based on the original loss, we put more emphasis on the prediction of the generated recommendation list. Therefore, the following distillation loss is designed for each user:

$$\begin{aligned}\mathcal{L}_{GD}(\theta s;\theta_T) = -\sum_{i\in GF(I)} &(q\log(P(u,i)) \\ &+ (1-q)\log(1-P(u,i))\end{aligned} \tag{6}$$

Where $P(u,i)$ the prediction of a recommender and the $GF(I)$ represents a set of items in the recommended list. $P(u,i)$ is calculated by $\sigma(z_{ui}/t)$,$\sigma()$is the sigmoid function, *t* is the temperature, and z_ui is the logit value output by the model.

Dataset	User	Item	Ratings	Sqarsity
CiteUlike	5219	25975	130799	99.91%
Yelp	25677	25815	730623	99.89%
Foursquare	2293	61858	537167	99.62%

Table 1: The characteristics of three data sets

Overview of the GD-REC Framework

Figure 1 shows an overview of the GD-REC framework. The overall model is divided into two parts: the left part is the generative adversarial networks, and the right part is the knowledge distillation module. The model is divided into two training phases. The first phase is trained offline. The Generator and Discriminator jointly conduct adversarial training to generate a recommendation list and train a powerful teacher model. The well-trained teacher model is then used to make predictions on the generated recommendation list. The additional knowledge is used in the second stage to train a smaller student model, producing the final recommendation list. Since the teacher model has the characteristics of high parameters, high performance, and high computational cost, it can provide additional knowledge for student model training. After training, the student model inherits the high performance of the teacher model. Therefore, users only need to download the lightweight student model to complete efficient recommendation applications.Algorithm 1 shows a complete GD training process.

Experiment

In this section, we set up 18 experimental environments (3 real data sets * 2 experimental models * 3 model parameters of different sizes) to verify the effectiveness of our GD-REC framework.

Experimental Setup

Datasets. We use three public real-world data sets for extensive experiments: CiteULike (Zhang, Lian, and Yang 2017), Foursquare (Yang et al. 2014), Yelp (He et al. 2016). In a preprocessing, we filtered users and items with less than 5 sets of data in CiteULike, Foursquare, and less than 10 users in Yelp (Kang et al. 2019). Table 1 summarizes the characteristics of three data sets.

Base Models. Since the GD-REC framework is suitable for any top-*K* recommendation model, we choose two models with different architectures and optimization strategies to verify the effectiveness of GD-REC. They are a deep learning model, NeuMF, and a latent factor model, BPR. The two models are widely used in top-*K* recommendation scenarios. As a model for teachers and students, it can better verify the impact of generative adversarial network and knowledge distillation technology on recommendation accuracy.

NeuMF (He et al. 2017)is a deep learning recommendation algorithm, based on matrix factorization and multi-layer perceptrons to learn the interaction between users and items.

Model	Method	CiteULike		Yelp		Foursquare	
		HR@50	NDCG@50	HR@50	NDCG@50	HR@50	NDCG@50
NeuMF	Teacher	0.125	0.033	0.085	0.034	0.186	0.064
	Student	0.069	0.179	0.058	0.016	0.127	0.049
	CD	0.082	0.022	0.071	0.018	0.154	0.059
	RD	0.075	0.019	0.066	0.016	0.144	0.054
	BD	0.092	0.024	**0.077**	0.020	0.165	**0.063**
	GD	**0.105**	**0.027**	0.073	**0.022**	**0.169**	0.059
BPR	Teacher	0.139	0.037	0.107	0.030	0.191	0.064
	Student	0.072	0.015	0.045	0.015	0.102	0.037
	CD	0.080	0.018	0.066	0.014	0.151	0.056
	RD	0.078	0.017	0.066	0.015	0.141	0.051
	BD	0.085	0.021	0.073	**0.024**	0.161	**0.061**
	GD	**0.097**	**0.023**	**0.077**	0.018	**0.163**	0.058

Table 2: Performance comparison of each model on three datasets

BPR (Rendle et al. 2012)is a sorting recommendation algorithm based on matrix factorization, focusing on user privacy feedback. It uses the pairwise loss function for optimization.

Evaluation Protocol. We follow the leave-one-out evaluation protocol (Kang et al. 2019). For each user, the last timestamp of the interaction record between user and item as test data, and the rest data is used to train the model. We select all unrated items as candidate items. Although it is time-consuming, it enables a more thorough evaluation compared to using random.

Evaluation. We focus on top-*K* recommender systems, and use a wide range of metrics to measure the accuracy of the top-*K* recommendations including hit rate (HR) and normalized cumulative depreciation return (NDCG(Batmaz et al. 2019)). The size of the ranking list *K* HR@*K* and NDCG@*K* are 50. HR@*K* tests whether the item appears in the top-*K* recommendation list or not, and NDCG@*K* recommends higher-ranked items to the top-*K* to calculate higher scores. The higher the value, the better the recommendation. We calculate the two indicators, HR@*K* and NDCG@*K* for each user and then calculate the average score. The final score reported is the value of the score of five independent runs.

Methods Compared. The proposed GD-REC framework is compared with the most advanced methods below:

Ranking Distillation (RD) (Tang and Wang 2018) is a pioneering work that combines knowledge distillation with a top-*K* recommender system. In RD, the teacher model teaches the student models to predict the top items for training.

Collaborative Distillation (CD) (Lee et al. 2019) uses the soft target of the teacher model to train the student model to solve the problem of RD, that is, ignoring the output of the teacher model.

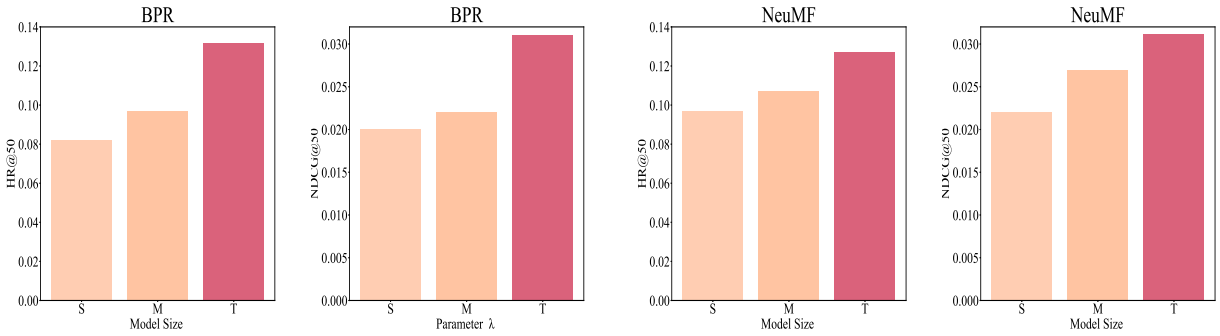
Bidirectional Distillation (BD) (Kweon, Kang, and Yu 2021) is the most advanced method of applying KD to recommender systems. The teacher and the student are collaboratively improved with each other during the training.

Implementation Details. We use PyTorch (Paszke et al. 2019) to implement and the Adam optimizer with L2 regularization to train all Base Models. For each data set, the hyperparameters are adjusted by using grid search on the validation set. Learning rate is selected from $\{0.1, 0.2, 0.01, 0.02, 0.001, 0.0001\}$, and the model regularizer is chosen from $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$. The batch we set is 1000, and the early stop strategy is adopted. For the two basic models (BPR, NeuMF), the number of negative samples is set to 1, and for NeuMF, a two-layer MLP is used as the network.

For knowledge distillation methods (RD,CD,BD,GD), in each model and each data set, we increase the number of model parameters until the recommended performance does not increase anymore, and then use the highest performance model as the teacher model. The well-trained teacher models are represented by BPR-T and NeuMF-T. For the student model, we construct it by building the teacher model and use 50% and 30% of the learning parameters. Among them, BPR-S and NeuMF-S represent student models that only receive top-*K* recommendation training. Note that the student model did not make any changes to the model structure. The weight for KD loss is chosen from $\{1, 10^{-1}, 10^{-2}, 10^{-3}\}$ and the temperature *t* for logits is chosen from $\{1, 2, 5, 10\}$ λ are set to 0.7. For other hyperparameters, we use the values recommended from the public implementation and the original papers.

Performance Comparison

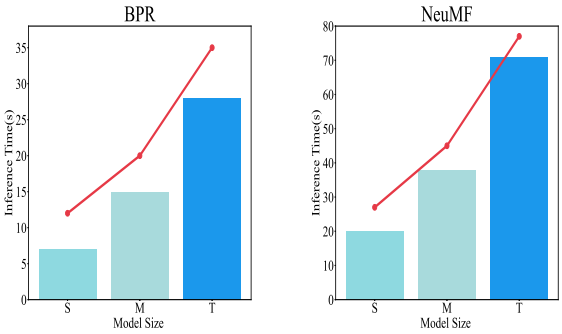
Table 2 shows the top-*K* recommendation performance of each method on three real data sets and two different Base models. In Table 2, Teacher and Student represent the basic models trained separately without the aid of any technology. CD,RD and BD are the results of training. GD corresponds to the result of using KD in this article. The GD method is superior to the existing state-of-the-art KD methods on two base models with different architectures and optimization strategies. In the following, we analyze the results from



(a) Three BPR models with different parameters

(b) Three NeuMF models with different parameters

Figure 2: Three different parameter model performance



(a) Three BPR models with different parameters

(b) Three NeuMF models with different parameters

Figure 3: Comparison of three different parameters inference time

different perspectives.

First of all, we have observed in the experiments that the proposed GD framework can effectively improve the performance of the student model, especially on the CiteULike dataset the HR@50 of NeuMF goes up from 0.075 to 0.105. The student model of the GD method can learn from the teacher's prediction and the generation of the recommendation list to learn the user's potential preferrence, including the relationship between users and items. Compared with other methods, our models have limited ability to reduce the adverse effects of large performance gaps, and pay more attention to accurate information to achieve more effective training.

Second, we found that GD achieves higher performance gains, especially in the BPR model performance (the NDCG@50 in CiteULike goes up from 0.017 to 0.023). The advantages of GD come from two reasons. It focuses more on adversarial network generation lists because it can score close to 1 in interaction items and vice versa. It can better capture the ranking order of interaction or non-interaction items. Similarly, in the process of GD's internal knowledge

	method	K=5	K=10	K=20
Hit@K	Blurm	0.184	0.263	0.364
	DPAE	0.185	0.285	0.397
	DPMF	0.195	0.280	0.394
	GERAI	0.333	0.495	0.670
	GD	0.321	0.504	0.634

Table 3: Recommendation effectiveness results

distillation, improved teacher models can effectively alleviate the challenges brought by sparsity, thereby accelerating the progress. The knowledge conveyed by the improvement of teacher performance will be more accurate. To verify the effectiveness of this contribution, we will conduct an in-depth analysis in Design Choice Analysis experiment.

Finally, we observe that GD is for the most part superior to the comparison method. However, the improvements are not quite strong in Yelp's HR@50. These are the results we predicted because our student model uses the generated recommendation list and the output of the teacher network for training. Our student model is not exposed to the original data training, and does not to use all the data set training like other methods. Although the performance cannot be further largely improved, the privacy protection effect that GD brings in is more in line with applications in a real world.

Accuracy and Privacy

To strengthen the validity and privacy of our model in a personalized recommendation, we conducted experiments in the ML-100K datasets. It contains 100,000 ratings from 943 Users on 1,682 movies collected from the MovieLens website. We summarize recommendation Model Performance on recommended system privacy protection with Table 3.

We make a comparison with the recent work related to the privacy protection of recommendation systems, especially differential privacy. It can be seen that the recommendation accuracy of GD in the case of K= $\{5, 10, 20\}$ keeps relevant competitiveness with the recommendation system combining differential privacy and graph neural network. It is important

MODEL		CiteUlike	Yelp	Foursquare
NeuMF-T	KD	0.117	0.079	0.162
	GD	**0.125**	**0.085**	**0.180**
BPR-T	KD	0.122	0.087	0.172
	GD	**0.139**	**0.107**	**0.191**

Table 4: Comparison of teacher performance on knowledge distillation of each model on three datasets

to note that, unlike other literature, our model training does not use original data. The use of differential privacy and other related technologies to protect recommended user privacy still has the risk of disclosing original data. Our method benefits from generative adversarial network manufacturing fake recommendation lists and has stronger privacy protection capability.

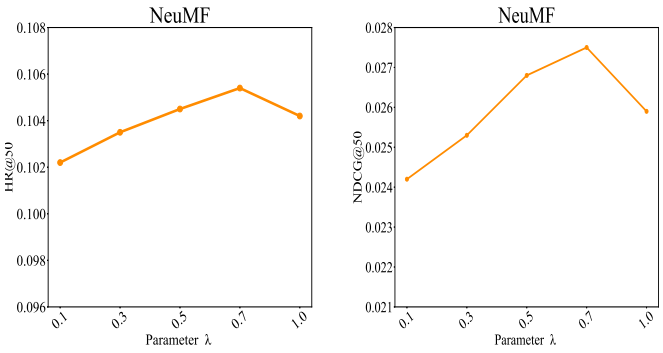
Model Time Comparison

Figures 2 and 3 show the experimental results of the online reasoning efficiency of different base model sizes. To make inferences, we use Pytorch with CUDA on Intel(R) Xeon(R) Silver 4215 CPU and Tesla V100 GPU. We report the performance of the two base models on the CiteULike dataset at different model sizes and their inference time. Figures 2(a) and 2(b) respectively plot the performance of BPR and NeuMF models with different parameter sizes in terms of under the same index. Figure 3 compares their inference time. In these figures, T represents the teacher model, M represents the medium model size (i.e., 50% teacher), and S represents the small model size (i.e., 30% teacher).

We observe that the model size is directly proportional to the model accuracy, and the trend is always the same in the two different optimization strategies, BPR and NeuMF. When compared with the teacher model, the student model trained with GD achieves similar performance with only about 50% of the learning parameters. Smaller models require less computational cost and memory cost so that less inference can be obtained. Our method achieve a better trade-off between effectiveness and efficiency. Especially in the deep recommendation model with many learning parameters, we can obtain higher accuracy from a smaller model size. GD-REC can be applied to recommender systems applications, which will significantly improve the efficiency of online recommending.

Design Choice Analysis

We have conducted a quantitative and qualitative analysis of the proposed method, which has verified the superiority of our design choices. In terms of teacher model performance, the comparison is summarized in Table 3. For the two basic models, we study the performance of the teacher model on the three data sets of HR@50, where KD represents normal knowledge distillation training to obtain the teacher model results, and GD represents the teacher model that uses the generated adversarial networks to generate data for additional training.



(a)

Figure 4: Influence of different λ on distillation

As shown in Table 3, trained by the GD method using the same structure of the teacher model are better than the ordinary knowledge distillation method (e.g., HR@50 in Yelp goes up from 0.087 to 0.107). This result demonstrates the advantage of using generative adversarial networks for additional supervision. By generating the additional training of the recommendation list, a high-performance teacher model can be effectively improved. This improvement is due to the fact that our generative adversarial network approach is trained by vector adversarial training. With the competition of G and D, we obtain more reliable prediction scores and generate recommendation list data that meet user preferences.

Hyperparameter Analysis

In this section, we focus on the influence of hyperparameters on the GD process. We report the HR@50 and NDCG@50 results of NeuMF on the CiteUlike dataset. The similar trends are also observed on other data sets and other models.

Figure 4 shows the effect of the hyperparameter of knowledge distillation λ on the overall GD distillation loss. As shown in Figure 4, in terms of HR@50 and NDCG@50, the overall performance is the best when λ is 0.7. The reason behind this is that generating recommendation lists do not completely train a suitable student model, and adding strong teacher supervision could be more effective. However, it is not foolproof to increase λ blindly. When λ is 1, its performance is not the best, which shows that too much distillation could lead to poor performance.

Conclusion

In this article, we proposed a new knowledge distillation model framework called GD-REC. The GD-REC provides users with lightweight recommendation services in open-world scenarios. It addresses the problems of data sparsity, recommend efficiency, and privacy protection in the top-K recommendations. We are committed to reducing inference time and strengthening privacy protection while ensuring accuracy. In the future, we will expand this framework in cross-domain recommendation applications.

Acknowledgements

The authors wish to thank the anonymous reviewers for their helpful comments and suggestions for this paper. This is in supported by the National Natural Science Foundation of China under Grants U21A20474, by the Guangxi "Bagui Scholar" Teams for Innovation and Research Project, by the Guangxi Collaborative Innovation Center of Multi-source Information Integration and Intelligent Processing, by the Guangxi Natural Science Foundation (2020GXNSFAA297075,2020GXNSFBA297108), by the Guangxi Key Laboratory of Trusted Software(No. KX202037), by the Guangxi Science and Technology(No. GuiKeAD 20297054), by the Research Fund of Guangxi Key Lab of Multi-source Information Mining Security (No. 19-A-02-02), and by the National Science Foundation (NSF) Grant 1921576.

References

Arjovsky, M.; and Bottou, L. 2017. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*, 214–223. PMLR.

Bachrach, Y.; Finkelstein, Y.; Gilad-Bachrach, R.; Katzir, L.; Koenigstein, N.; Nice, N.; and Paquet, U. 2014. Speeding up the xbox recommender system using a euclidean transformation for inner-product spaces. In *Proceedings of the 8th ACM Conference on Recommender systems*, 257–264.

Batmaz, Z.; Yurekli, A.; Bilge, A.; and Kaleli, C. 2019. A review on deep learning for recommender systems: challenges and remedies. *Artificial Intelligence Review*, 52(1): 1–37.

Chae, D.-K.; Kang, J.-S.; Kim, S.-W.; and Lee, J.-T. 2018. Cfgan: A generic collaborative filtering framework based on generative adversarial networks. In *Proceedings of the 27th ACM international conference on information and knowledge management*, 137–146.

Chen, C.; Liu, Z.; Zhao, P.; Zhou, J.; and Li, X. 2018. Privacy preserving point-of-interest recommendation using decentralized matrix factorization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Chen, H.; Wang, Y.; Xu, C.; Yang, Z.; Liu, C.; Shi, B.; Xu, C.; Xu, C.; and Tian, Q. 2019. Data-free learning of student networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3514–3522.

Deng, Z.; Huang, L.; Wang, C.; Lai, J.; and Yu, P. S. 2019. DeepCF: A Unified Framework of Representation Learning and Matching Function Learning in Recommender System. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 61–68. AAAI Press.

Gao, M.; Zhang, J.; Yu, J.; Li, J.; Wen, J.; and Xiong, Q. 2021. Recommender systems based on generative adversarial networks: A problem-driven perspective. *Information Sciences*, 546: 1166–1185.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

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.

He, X.; Zhang, H.; Kan, M.-Y.; and Chua, T.-S. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 549–558.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Järvelin, K.; and Kekäläinen, J. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4): 422–446.

Kang, S.; Hwang, J.; Kweon, W.; and Yu, H. 2020. DE-RRD: A Knowledge Distillation Framework for Recommender System. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 605–614.

Kang, S.; Hwang, J.; Lee, D.; and Yu, H. 2019. Semi-supervised learning for cross-domain recommendation to cold-start users. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 1563–1572.

Kweon, W.; Kang, S.; and Yu, H. 2021. Bidirectional Distillation for Top-K Recommender System. In *Proceedings of the Web Conference 2021*, 3861–3871.

Lee, J.; Choi, M.; Lee, J.; and Shim, H. 2019. Collaborative Distillation for Top-N Recommendation. In Wang, J.; Shim, K.; and Wu, X., eds., *2019 IEEE International Conference on Data Mining, ICDM 2019, Beijing, China, November 8-11, 2019*, 369–378. IEEE.

Li, H.; Chan, T. N.; Yiu, M. L.; and Mamoulis, N. 2017. FEXIPRO: fast and exact inner product retrieval in recommender systems. In *Proceedings of the 2017 ACM International Conference on Management of Data*, 835–850.

Li, H.; Hong, R.; Lian, D.; Wu, Z.; Wang, M.; and Ge, Y. 2016. A Relaxed Ranking-Based Factor Model for Recommender System from Implicit Feedback. In *IJCAI*, 1683–1689.

Li, Y.; Wang, S.; Pan, Q.; Peng, H.; Yang, T.; and Cambria, E. 2019. Learning binary codes with neural collaborative filtering for efficient recommendation systems. *Knowledge-Based Systems*, 172: 64–75.

Liu, R.; Liang, J.; Gao, W.; and Yu, R. 2018. Privacy-based recommendation mechanism in mobile participatory sensing systems using crowdsourced users' preferences. *Future Generation Computer Systems*, 80: 76–88.

Liu, X.; Li, Q.; Ni, Z.; and Hou, J. 2019. Differentially Private Recommender System with Autoencoders. In *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom)*

and IEEE Smart Data (SmartData), iThings/GreenCom/CPSCom/SmartData 2019, Atlanta, GA, USA, July 14-17, 2019, 450–457. IEEE.

Mirza, M.; and Osindero, S. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Mishra, A.; and Marr, D. 2017. Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy. *arXiv preprint arXiv:1711.05852*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32: 8026–8037.

Radford, A.; Metz, L.; and Chintala, S. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

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

Tang, J.; and Wang, K. 2018. Ranking distillation: Learning compact ranking models with high performance for recommender system. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2289–2298.

Wang, J.; Yu, L.; Zhang, W.; Gong, Y.; Xu, Y.; Wang, B.; Zhang, P.; and Zhang, D. 2017. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, 515–524.

Wang, Q.; Yin, H.; Wang, H.; Nguyen, Q. V. H.; Huang, Z.; and Cui, L. 2019a. Enhancing collaborative filtering with generative augmentation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 548–556.

Wang, Z.; Gao, M.; Wang, X.; Yu, J.; Wen, J.; and Xiong, Q. 2019b. A minimax game for generative and discriminative sample models for recommendation. In *Pacific-Asia conference on knowledge discovery and data mining*, 420–431. Springer.

Wu, C.; Wu, F.; Wang, X.; Huang, Y.; and Xie, X. 2021. Fairness-aware News Recommendation with Decomposed Adversarial Learning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, 4462–4469. AAAI Press.

Yang, D.; Zhang, D.; Zheng, V. W.; and Yu, Z. 2014. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(1): 129–142.

Zhang, S.; Yao, L.; Sun, A.; and Tay, Y. 2019a. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1): 1–38.

Zhang, S.; Yin, H.; Chen, T.; Huang, Z.; Cui, L.; and Zhang, X. 2021. Graph Embedding for Recommendation against Attribute Inference Attacks. In Leskovec, J.; Grobelnik, M.; Najork, M.; Tang, J.; and Zia, L., eds., *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, 3002–3014. ACM / IW3C2.

Zhang, S.; Yin, H.; Wang, Q.; Chen, T.; Chen, H.; and Nguyen, Q. V. H. 2019b. Inferring Substitutable Products with Deep Network Embedding. In Kraus, S., ed., *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, 4306–4312. ijcai.org.

Zhang, Y.; Lian, D.; and Yang, G. 2017. Discrete personalized ranking for fast collaborative filtering from implicit feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.