Reinforced Negative Sampling for Recommendation with Exposure Data

Jingtao Ding[1] , **Yuhan Quan**[1] , **Xiangnan He**[2] , **Yong Li**[1] and **Depeng Jin**[1]
[1] Beijing National Research Center for Information Science and Technology (BNRist),
Department of Electronic Engineering, Tsinghua University
[2] School of Information Science and Technology, University of Science and Technology of China
liyong07@tsinghua.edu.cn

Abstract

In implicit feedback-based recommender systems, user exposure data, which record whether or not a recommended item has been interacted by a user, provide an important clue on selecting negative training samples. In this work, we improve the negative sampler by integrating the exposure data. We propose to generate high-quality negative instances by adversarial training to favour the difficult instances, and by optimizing additional objective to favour the real negatives in exposure data. However, this idea is non-trivial to implement since the distribution of exposure data is latent and the item space is discrete. To this end, we design a novel RNS method (short for ***R**einforced **N**egative **S**ampler*) that generates exposure-alike negative instances through feature matching technique instead of directly choosing from exposure data. Optimized under the reinforcement learning framework, RNS is able to integrate user preference signals in exposure data and hard negatives. Extensive experiments on two real-world datasets demonstrate the effectiveness and rationality of our RNS method. Our implementation is available at: https://github.com/dingjingtao/ReinforceNS.

1 Introduction

The prevalence of implicit feedback has boosted the research and development of implicit feedback-based recommender systems [Bayer *et al.*, 2017; Yang *et al.*, 2018]. The key challenge in learning from implicit feedback lies in the natural scarcity of negative signal, known as one-class problem [Pan *et al.*, 2008]. To address this issue, negative sampling has been widely adopted in previous works [Rendle *et al.*, 2009], where the common approach is to uniformly sample negative instances from the missing data (i.e., the unobserved interactions) [Jiang *et al.*, 2018; He *et al.*, 2018; Lin *et al.*, 2019]. This process, with no doubt, plays a critical role in training recommender models from implicit feedback.

Given the importance of designing a quality negative sampler for implicit recommender models, two types of methods have been proposed in previous works. The first is *heuristic-based* methods, including dynamic negative sampling (DNS) that oversamples the hard negative instances during the training process [Zhang *et al.*, 2013; Rendle and Freudenthaler, 2014], and frequency-based sampling that subsamples frequent instances [Caselles-Dupré *et al.*, 2018]. The other type is *auxiliary information-based* methods, which focus on choosing more reliable negative instances by leveraging the auxiliary data such as clicked but non-purchased items in E-commerce websites [Ding *et al.*, 2018a].

In real-world scenarios, platforms can easily collect whether the recommended (i.e., exposed) item has been interacted by a user. These records, also referred to as the *exposure data*, contain rich information about the negative preference of users. However, due to the inaccessibility of such data by the third parties, previous academic research exploits the interaction data only to build the negative sampler. As a result, the reliability of the generated negative instances is questionable. Nevertheless, it is non-trivial to integrate the exposure data into the negative sampler design because of the following challenges:

- **Incompleteness of negative preference in exposure data.** The exposed items are typically selected by a recommendation engine. Besides the exposed but non-interacted items, other non-exposed items can also be the negative preference for a user. If the sampler simply generates negative instances according to the exposure data, it will cause selection bias, making the model under-trained and resulting in suboptimal performance [Lian *et al.*, 2017; Ding *et al.*, 2018a].
- **Difficulty of optimizing the negative sampler.** Due to the discrete sampling process on item IDs, the objective function is non-differentiable. As such, it cannot be optimized with traditional gradient-based techniques that can only deal with continuous functions.

In this work, we design a novel embedding-based *sampler* model named **R**einforced **N**egative **S**ampler (RNS) that learns to generate informative and effective negative samples. Specifically, the *sampler* collaborates with another embedding-based *recommender* model, supplying negative instances for the *recommender* to do pairwise learning. The sampler has two goals — generating **hard** and **real** negative instances. The hard goal is achieved through adversarial training between the *recommender* and the *sampler*. Simultaneously, the *sampler* is also rewarded to generate negative instances that overlap with the non-interacted instances in the exposure data. Corresponding to aforementioned two challenges, here the "reinforced" has two-fold meaning. First, the generation of real negative instances is reinforced by a feature matching technique, which forces the empirical distribution

of the generated and the exposed negative instances to have matched moments in the latent feature space. Second, we consider a reinforcement learning setting in the *sampler* so as to receive gradient information from the *recommender* and optimize the non-differentiable exposure-based objective.

We summarize the contributions of this paper as follows.

1. We are the first to consider generating exposure-alike negative instances for implicit recommendation. The proposed RNS model is general in optimizing any recommender models, with the potential of large impact.
2. We propose two specific designs to generate high-quality negative instances, including the adversarial training for hard negatives and the feature matching for generating exposure-alike negatives that are more reliable.
3. We conduct extensive experiments on two real-world datasets to demonstrate the effectiveness of RNS. More ablation studies verify the efficacy of the two designs and the utility of feature matching in leveraging exposure data.

2 Methodology

We start by introducing some basic notations. For a specific user u, $\mathcal{C}_u$ denotes the set of items that are interacted by u, while $\mathcal{E}_u$ refers to those non-interacted items within u's exposure history. We represent matrices, vectors, and scalars as bold capital letters (e.g., $\mathbf{X}$), bold lower-case letters (e.g., $\mathbf{x}$), and normal lower-case letters (e.g., x), respectively. We use symbols σ and $\odot$ to denote the sigmoid function and element-wise production, respectively.

Our proposed recommender-sampler framework is illustrated in Figure 1. The sampler (S) calculates a probability distribution over a set of candidate negative instances, then samples one of them as the output. Next the recommender (R) is optimized to learn the pairwise ranking relation between a ground truth instance and a generated negative one. After receiving the multiple reward signals, S is encouraged to generate both hard (ω^1_{uj}) and real ($\omega^2_{uj}/\omega^3_{uj}$) negative instances. During training process, R can benefit more from the better quality negative instances and thus perform better on predicating user preference.

2.1 The Recommender Model (R)

To learn recommender models from implicit feedback, Rendle *et al.* [Rendle *et al.*, 2009] proposed the Bayesian Personalized Ranking (BPR) method, which assumes that a positive instance should be predicted with a much higher score over the negative one. Based on BPR, the training objective of R can be formulated as minimizing the following loss function:

$$L_R = \sum_{(u,i)\in\mathcal{C}} -\ln \sigma(\hat{r}_{ui}(\Theta) - \hat{r}_{uj}(\Theta)), \text{ where } j \sim \hat{\Psi}_S(j|u). \quad (1)$$

In (1), for each user u, the predicted preference score on items is denoted as $\hat{r}_{u\bullet}(\Theta)$, where Θ refers to the model parameters. The negative instance j is generated by the sampler (S) according to a conditional distribution $\hat{\Psi}_S(\cdot|u)$, while the positive instance i is randomly chosen from ground truth set $\mathcal{C}_u$. Minimizing L_R is equivalent to maximizing the margin between $\hat{r}_{ui}$ and $\hat{r}_{uj}$, which encourages R to learn the pairwise ranking relation of user preference between i and j.

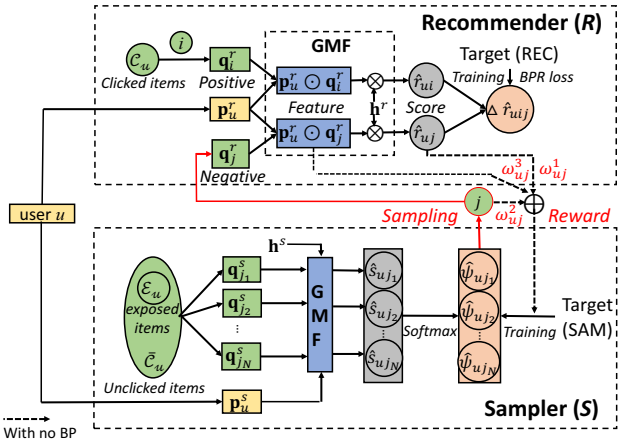


Figure 1: Our proposed recommender-sampler framework.

To calculate $\hat{r}_{u\bullet}(\Theta)$, we adopt the Generalized Matrix Factorization (GMF) [He *et al.*, 2017] model that allows different dimensions of the embedding space to have different weights. Specifically, it first uses an element-wise product to obtain an interacted feature vector, and then project the feature vector to an output score with a weight vector as follows, $\hat{r}_{ui}(\Theta) = \mathbf{h}^{r\,T}\mathbf{f}_{ui} = \mathbf{h}^{r\,T}(\mathbf{p}^r_u \odot \mathbf{q}^r_i)$, where $\mathbf{h}^r \in \mathbb{R}^{K\times 1}$ denotes the learnable weight vector and $\mathbf{f}_{ui}$ denotes the interacted feature vector. The K-dimensional user embedding and item embedding are represented as $\mathbf{p}^r_u$ and $\mathbf{q}^r_i$, respectively. Based on GMF, the model parameters Θ is $\{\mathbf{p}^r_u, \mathbf{q}^r_i, \mathbf{h}^r\}$.

2.2 The Sampler Model (S)

We first introduce two specific design of our proposed reinforced negative sampler (RNS), corresponding to generating hard and real negative samples. Then we train RNS through policy gradient in reinforcement learning (RL) setting.

Adversarial Sampler

Intuitively, S can generate adversarial negative instances that have high scores $\hat{r}_{u\bullet}$, which are hard for R to rank correctly. Therefore, the objective of this adversarial sampler is formulated as maximizing the expectation of j-related part in L_R:

$$L_{AS} = \sum_{(u,i)\in\mathcal{C}} \mathbb{E}_{j\sim\hat{\Psi}_S(j|u)}\big[\underbrace{-\sigma(-\hat{r}_{uj}(\Theta))}_{\text{denoted as } \omega^1_{uj}}\big]. \quad (2)$$

Note that we have left out the logarithm to control its values within $[-1, 0]$. By this means, the generated negative instances are given higher prediction scores by R, being more close to those of positive instances, which provides larger gradients and more information for R.

Exposure-matching Sampler

As both users' interaction and non-interaction are explicitly recorded in the exposure data, it is reasonable to generate real negative samples based on the exposed but not interacted instances. Therefore, our second design of RNS is introducing

an exposure-matching sampler that learns a probability distribution to match negative signal in exposure data. Given u's exposed itemset $\mathcal{E}_u$, a direct design is to maximize the overlap between the set of generated instances and $\mathcal{E}_u$. Specifically, we consider the following objective:

$$L^0_{ES} = \sum_{(u,i)\in\mathcal{C}} \mathbb{E}_{j\sim\hat{\Psi}_S(j|u)}\big[\mathbf{1}_{\mathcal{E}_u}(u,j)\big], \quad \text{where } \mathbf{1}_{\mathcal{E}_u}(u,j) = \begin{cases} 1\ , & j \in \mathcal{E}_u \\ 0\ , & \text{else} \end{cases}, \tag{3}$$

where the binary indicator function $\mathbf{1}_{\mathcal{E}_u}$ serves as a guiding signal, encouraging S to generate a set of negative instances that has a big overlap with exposed itemset $\mathcal{E}_u$.

However, for the set of generated negative instances and the set of exposed instances, L^0_{ES} measures their distance by the size of overlapping set, *i.e.*, a simple scalar, which does not provide information in latent feature level and thus be suboptimal. Consequently, the sampler trained with this objective may tend to choose exactly the same instances from the exposure data, which will harm the performance as not all of the exposed instances are beneficial.

Therefore, we further measure the similarity in the latent feature space. As we adopt GMF to calculate preference scores in R, for a user-item pair denoted as (u,i), the interacted feature vector $\mathbf{f}_{ui}$ is the corresponding representation in the latent feature space. Consider a subset of generated negative instances $\mathcal{G}_s$, for each $(u,j) \in \mathcal{G}_s$, we sample a (u,k) from $\mathcal{E}_u$ and group all these sampled instances into a subset denoted as $\mathcal{E}_s$. Then, to measure the distance between $\mathcal{G}_s$ and $\mathcal{E}_s$, we consider the maximum mean discrepancy (MMD) between the empirical distribution of feature vectors in these two subsets [Li *et al.*, 2015; Zhang *et al.*, 2017]. Concisely, MMD measures the mean squared difference between two sets of samples over a universal reproducing kernel Hilbert space. In our case, the MMD for two empirical distributions of $\mathbf{f}$ in $\mathcal{G}_s$ and $\mathcal{E}_s$ is given by

$$\begin{aligned} J_{MMD^2} = &\frac{1}{L^2}\sum_{(u,j)\in\mathcal{G}_s}\sum_{(u',j')\in\mathcal{G}_s} k(\mathbf{f}^{\mathcal{G}}_{uj}(\Theta), \mathbf{f}^{\mathcal{G}}_{u'j'}(\Theta)) \\ &- \frac{2}{L^2}\sum_{(u,j)\in\mathcal{G}_s}\sum_{(v,k)\in\mathcal{E}_s} k(\mathbf{f}^{\mathcal{G}}_{uj}(\Theta), \mathbf{f}^{\mathcal{E}}_{vk}(\Theta)) \\ &+ \frac{1}{L^2}\sum_{(v,k)\in\mathcal{E}_s}\sum_{(v',k')\in\mathcal{E}_s} k(\mathbf{f}^{\mathcal{E}}_{vk}(\Theta), \mathbf{f}^{\mathcal{E}}_{v'k'}(\Theta)), \end{aligned} \tag{4}$$

where L is the size of two sets and $k(\cdot,\cdot) : \mathbb{R}^K \times \mathbb{R}^K \mapsto \mathbb{R}$ is the kernel function. Here we use a universal Gaussian kernel, *i.e.*,, $k(\mathbf{x},\mathbf{x}') = \exp(-||\mathbf{x}-\mathbf{x}'||^2/2\tau)$ with bandwidth τ, and minimize MMD to match all moments of two distributions.

From the above considerations, we design the exposure-matching sampler with a weighted sum of overlap-based objective and MMD-based objective. Mathematically, it can be formulated as maximizing

$$L_{ES} = \sum_{(u,i)\in\mathcal{C}} \mathbb{E}_{j\sim\hat{\Psi}_S}\big[(1-\beta)\underbrace{\mathbf{1}_{\mathcal{E}_u}(u,j)}_{\text{denoted as } \omega^2_{uj}} + \beta \underbrace{M(u,j)}_{\text{denoted as } \omega^3_{uj}}\big], \tag{5}$$

where β controls the importance of MMD-based objective $M(u,j)$ that aims to match the feature distribution. As the MMD is calculated between two sets, *i.e.*, $\mathcal{G}_s$ and $\mathcal{E}_s$, for each u and u's generated negative instance j, $M(u,j)$ only refers to those corresponding terms in (4), which is given by

$$M(u,j) = \frac{2}{L^2}\big[\sum_{(v,k)\in\mathcal{E}_s} k(\mathbf{f}^{\mathcal{G}}_{uj}, \mathbf{f}^{\mathcal{E}}_{vk}) - \sum_{(u',j')\in\mathcal{G}_s} k(\mathbf{f}^{\mathcal{G}}_{uj}, \mathbf{f}^{\mathcal{G}}_{u'j'})\big]. \tag{6}$$

Training with Reinforcement Learning

Finally, combining the above two specifically designed sampler together, we obtain the following objective for the negative sampler (S):

$$\begin{aligned} L_S &= L_{AS} + \alpha L_{ES} \\ &= \sum_{(u,i)\in\mathcal{C}} \mathbb{E}_{j\sim\hat{\Psi}_S(j|u)}\big[\underbrace{\omega^1_{uj} + \alpha(1-\beta)\omega^2_{uj} + \alpha\beta\omega^3_{uj}}_{\text{denoted as } \omega_{uj}}\big], \end{aligned} \tag{7}$$

where $\{\omega^1_{uj}, \omega^2_{uj}, \omega^3_{uj}\}$ refer to adversarial objective, overlap-based objective and MMD-based objective, respectively (See (2) and (5)). By maximizing L_S, S is encouraged to generate both hard and real negative samples with exposure data.

However, unlike the optimization of R that can be achieved by the stochastic gradient descent (SGD), training S has following two problems. First, it involves a discrete sampling step, which makes simple differentiation infeasible. Second, the exposure-aware indicator function $\mathbf{1}_{\mathcal{E}_u}$ is non-differentiable. Therefore, for S with model parameters Φ, we use the policy gradient based RL [Sutton *et al.*, 2000] to derive its gradient:

$$\begin{aligned} \nabla_\Phi L_S &= \nabla_\Phi \sum_{(u,i)\in\mathcal{C}} \mathbb{E}_{j\sim\hat{\Psi}_S(j|u)}\big[\omega_{uj}\big] \\ &= \sum_{(u,i)\in\mathcal{C}} \mathbb{E}_{j\sim\hat{\Psi}_S(j|u)}\big[\omega_{uj}\nabla_\Phi \log\hat{\Psi}_S(j|u)\big] \\ &\simeq \sum_{(u,i)\in\mathcal{C}} \frac{1}{T}\sum_{j_t\sim\hat{\Psi}_S(j|u), t\le T}\big[\omega_{uj_t}\nabla_\Phi \log\hat{\Psi}_S(j_t|u)\big], \end{aligned} \tag{8}$$

where we approximate the expectation with sampling in the last step. With the RL terminology, the agent, i.e., the sampler S, follows a policy $\hat{\Psi}_S(j|u)$ and takes an action as generating a negative instance j for a certain user u. Then, for each action (u,j), the environment, i.e., the recommender R, will return a reward ω_{uj} to S that guides the direction of optimization. In this way, S is iteratively optimized towards maximizing the returned reward, *i.e.*, generating the negative instances that are both hard and real.

To produce the probability distribution for sampling negative instances, *i.e.*, $\hat{\Psi}_S(j|u)$, S first calculates the scores for a set of negative candidate instances and then obtain their corresponding softmax probability. Similar to R, S also uses GMF to calculate the score with model parameters $\Phi = \{\mathbf{p}^s_u, \mathbf{q}^s_i, \mathbf{h}^s\}$. Mathematically, $\hat{\Psi}_S(j|u)$ is modeled as

$$\hat{\Psi}_S(j|u) = \frac{\exp\hat{s}_{uj}(\Phi)}{\sum_{j'\in\mathcal{N}_u}\exp\hat{s}_{uj'}(\Phi)}, \tag{9}$$

where $\hat{s}_{u\bullet}(\Phi)$ denotes the score and $\mathcal{N}_u$ denotes u's candidate set for the generated negative instances. Intuitively, $\mathcal{N}_u$

should contain all the instances that are not interacted by u, *i.e.*, $\overline{\mathcal{C}}_u$. However, as the hard negative instances are very likely to be those unobserved positive instances, we generate $\mathcal{N}_u$ by uniformly preselecting N_s instance from $\overline{\mathcal{C}}_u$.

The overall training process is summarized in Algorithm 1. Both R and S require pre-training, which is achieved by training the GMF model with BPR objective. The learning process is carried out in mini-batch mode, where R and S alternatively update their parameters.

Algorithm 1: The RNS algorithm.

Data : Interaction data $\mathcal{C} = \{(u, i)\}$, exposure data $\mathcal{E}$;
Input : Pre-trained recommender R_0 with parameters Θ, Pre-trained sampler S_0 with parameters Φ;
Output: The final recommender R for prediction;

while *Stopping criteria is not met* **do**
  Sample a mini-batch of data $\mathcal{C}_s$ from $\mathcal{C}$
  for $(u, i) \in \mathcal{C}_s$ **do**
    Uniformly sample N_s negative instances as $\mathcal{N}_u$;
    Obtain their sampling probability $\{\hat{\Psi}\}$ by (9);
    S generate a negative instance $j \in \mathcal{N}_u$ with $\{\hat{\Psi}\}$;
    $G_R \leftarrow G_R + \nabla_\Theta L_R(u, i, j)$; // R's gradients
    $\mathcal{G}_s.add((u, j))$; // the set of generated instances
    Sample one exposed instance j' from $\mathcal{E}_u$, $\mathcal{E}_s.add((u, j'))$; // used for calculating MMD
  end
  for $(u, j) \in \mathcal{G}_s$ **do**
    Calculate reward ω_{uj} by (2)-(7);
    $G_S \leftarrow G_S + \omega_{uj} \nabla_\Phi \log \hat{\Psi}(j|u)$; // S's gradients
  end
  $\Theta \leftarrow \Theta + \lambda_R G_R$, $\Phi \leftarrow \Phi + \lambda_S G_S$; // update R, S
end

2.3 Discussion

Our proposed recommender-sampler framework follow the general design of Generative Adversarial Networks [Goodfellow *et al.*, 2014], which contains two parts, the generator and the discriminator. In recommendation fields, the previous GAN-based models, such as IRGAN [Wang *et al.*, 2017], CF-GAN [Chae *et al.*, 2018] and AdvIR [Park and Chang, 2019], focuses on training a better generator to deceive the discriminator. In this sense, their generator is optimized to generate the positive instance and thus can predict user preference. However, in our work, the sampler generates the negative instances to train a better recommender as prediction model. The most related work is KBGAN [Cai and Wang, 2018] that generates hard negative instances to train better knowledge graph embeddings. However, our RNS model not only has an adversarial sampler, but also tries to generate real negative instances with exposure data.

3 Experiments

3.1 Experimental Settings

Datasets and Preprocessing. We perform experiments on two real-world datasets with both interactions and exposure:

Beibei[1] is one of the largest Chinese E-commerce websites. We sample a subset of data that contains item clicks and exposure within the time period from 2017/06 to 2017/07.

Zhihu[2] is the largest question-and-answer website in China, where users click articles of interest to read. Here we use a public benchmark released in CCIR-2018 Challenge[3].

In the raw data of Beibei and Zhihu, each user's records are grouped into different sessions, during which the user is recommended with a fixed number of items and only clicks some of them. Therefore, we consider a session-based data preprocessing with three steps. First, we filter out the repetitive clicks after the earliest one, as we aim to recommend novel items. Second, we only retain those unclicked items in the exposure data. Third, we filter out users and items with less than 4 (Beibei) and 6 (Zhihu) sessions to overcome the problem of high sparsity of raw datasets.

Table 1: Statistics of the evaluation datasets.

Dataset	User#	Item#	Train#	Val.#	Test#	Exposure#
Beibei	66,450	59,290	1,617,541	73,906	73,208	29,694,415
Zhihu	16,015	45,782	2,433,969	410,736	440,029	6,711,820

Evaluation Methodology. As the exposure and clicks are grouped into sessions, we adopt an evaluation protocol similar to *leave-one-out* [Rendle *et al.*, 2009], where the click interactions in the latest session of each user are held out for testing. For hyper-parameters tuning we further hold out the latest session from each user's training data as the validation set. Table 1 summarizes the statistics of experiment datasets. For the metrics, we employ *Area Under the Curve (AUC)* and *Normalized Discounted Cumulative Gain* (NDCG) on the ranking of a list of testing items for a user. We fix the list length as L and newly add some unclicked items into a user's list if it is less than L, and it is set as 40 and 160 for Beibei and Zhihu, respectively, which are the same as those in raw data. In this way, both AUC and NDCG equal to 1 when all the clicked items are ranked higher than other unclicked items. Finally we report the average score of all users.

Baselines. We compare our proposed RNS method with three groups of the baselines.

First we consider two common baselines:

- ItemPop. This method ranks items base on their popularity, as judged by the number of click interactions.

- BPR-GMF [Rendle *et al.*, 2009]. BPR optimizes the MF-based model with a pairwise ranking loss to learn from implicit feedback.

For methods related to the adversarial sampler, we choose:

- BPR-DNS [Zhang *et al.*, 2013]. Dynamic Negative Sampling (DNS) selects the item with the highest prediction score among X randomly sampled negatives.

- KBGAN [Cai and Wang, 2018]. With the generator serving as an adversarial sampler, KBGAN can be considered as a soft version of DNS. Here we combine it with BPR objective.

[1] http://www.beibei.com/

[2] https://www.zhihu.com/

[3] https://biendata.com/competition/CCIR2018/

- IRGAN [Wang *et al.*, 2017]. This method is a GAN-based IR model that aims to obtain a better generator through adversarial training, which is used to predict user preference.

Finally, we consider two exposure-enhanced samplers:

- BPR-EN. This method selects negatives only from exposure data. We use it to investigate the impact of selecting negatives from an incomplete candidate set.

- EBPR. Similar to [Ding *et al.*, 2018a], we consider to weight those exposed but unclicked items differently (compared with other unexposed items) when choosing negatives.

Parameter settings. For above baselines, we use GMF as scoring model and explore hyper-parameters similarly as the original paper. The mini-batch size and embedding size for all methods are set as 1024 and 32, respectively. We search L_2 regularizer and learning rate in [10^{-6},10^{-5},10^{-4},10^{-3},10^{-2}] and [0.0001, 0.0005, 0.001, 0.05, 0.1], respectively, and use Adam optimizer for learning. In addition, the size of negative candidate set, *i.e.*, N_s, is set as 100 and 30 in Beibei and Zhihu, respectively, which is optimal among [10, 20, ..., 150].

3.2 Performance Comparison

Table 2 displays the recommendation performance *w.r.t.* AUC and NDCG on two datasets, where we perform paired t-test between RNS and each of baselines over 10-round results.

Table 2: Performance comparison between all the methods, significant test is based on AUC.

Datasets		**Beibei**			**Zhihu**		
Group	**Methods**	**AUC**	**NDCG**	**p-value**	**AUC**	**NDCG**	**p-value**
Com.	ItemPop	0.6694	0.3668	1e-44	0.6500	0.6203	8e-47
	BPR-GMF	0.7065	0.3950	1e-22	0.6903	0.6443	2e-26
AS	BPR-DNS	0.7125	0.4013	3e-24	0.6939	0.6499	7e-35
	KBGAN	0.7109	0.3995	6e-25	0.6934	0.6486	1e-33
	IRGAN	0.7091	0.3963	5e-26	0.6686	0.6272	6e-21
ES	BPR-EN	0.6098	0.3282	9e-38	0.6196	0.5697	3e-40
	EBPR	0.6958	0.3896	1e-28	0.6975	0.6527	2e-19
AS+ES	**RNS**	**0.7168**	**0.4061**	-	**0.7076**	**0.6623**	-

From above results, we have the following observations:

- **RNS significantly improves the recommendation performance by training with the high quality negative instances.** Compared with ItemPop and BPR-GMF, our proposed RNS outperforms the best of them by 1.46% and 2.81% in AUC and NDCG for Beibei and by 2.51% and 2.79% in AUC and NDCG for Zhihu. It demonstrates that generating better quality negative instances is vital for learning user preference among different items.
- **RNS further improves the generation of hard negative instances by integrating exposure information.** We can observe that our proposed RNS achieves the best performance compared to those using the adversarial sampler. For Beibei, it improves the AUC and NDCG by 0.60% and 1.20%, while the improvement is much larger for Zhihu, *i.e.*, 1.97% and 1.91%. Compared with these baselines that simply choose hard negative instances based on the model's own inference, RNS use previous exposure information as the guiding signal to find more reliable negative instances, which explains the above performance improvement. Besides, we observe that BPR-DNS with a rather straightforward design consistently outperforms KBGAN and IRGAN, which may be due to the difficulty of learning an accurate distribution within a large item space ($10^4 \sim 10^5$).
- **RNS better leverages negative preference signal in exposure data.** Corresponding to our aforementioned challenge, we observe the significant performance degradation with a naive exposure-enhanced sampler. For BPR-EN that, as a common practice in most companies, selects negative instances only from the exposure, RNS outperforms it by 17.55% and 23.74% in AUC and NDCG for Beibei and similarly for Zhihu. As for EBPR, the relative improvement of RNS is 3.02% and 4.24% in AUC and NDCG for Beibei and 1.45% and 1.47% in AUC and NDCG for Zhihu. With our designed feature matching scheme, RNS can generate exposure-alike negatives from a much larger space that are not limited to exposure data, avoiding the incompleteness of negative preference in exposure data. This explains the observed outperformance of RNS.

To summarize, these comparisons verify that our proposed RNS model can effectively generate both hard and real negative instances to train a better implicit recommender model.

3.3 Hard Negative v.s. Real Negative

In RNS, we combine the adversarial sampler (AS) and exposure-matching sampler (ES) together so as to improve the quality of generated negative instances. An intuitive question is whether the designed two parts can really help?

Table 3: Impact of AS and ES in RNS.

Datesets	**Beibei**				**Zhihu**			
Methods	α	β	**AUC**	**NDCG**	α	β	**AUC**	**NDCG**
BPR-GMF	-	-	0.7065	0.3950	-	-	0.6903	0.6443
RNS-AS	0.00	-	0.7106	0.3985	0.00	-	0.7002	0.6570
RNS-ES	-	0.30	0.7160	0.4055	-	0.90	0.7066	0.6605
RNS	2.00	0.20	0.7168	0.4061	2.50	0.75	0.7076	0.6623

To answer it, we conduct experiments on two degenerative methods of RNS, in which only the adversarial objective L_{AS} and the exposure-matching objective L_{ES} are considered in sampler, respectively. We adopt the same evaluation method with above experiments, and the performance comparison is shown in Table 3, along with the value of weighting parameters for L_{ES} and MMD-based objective, *i.e.*, α and β. By comparing with BPR-GMF, we observe that both AS part and ES part play an essential role in RNS. Comparatively, generating negative instances that matches the exposure data (RNS-ES) is more helpful than generating hard negative instances through adversarial learning (RNS-AS). As these two samplers capture different signals, our experiments demonstrate that unifying them can achieve further improvement.

3.4 Impact of Feature Matching Scheme

Here we investigate whether the designed feature matching scheme helps generating exposure-alike instances, by

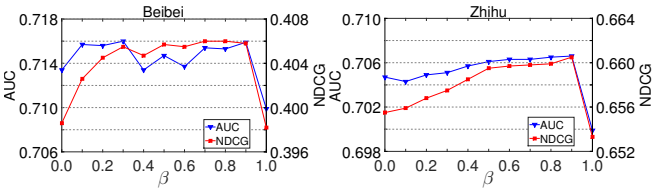
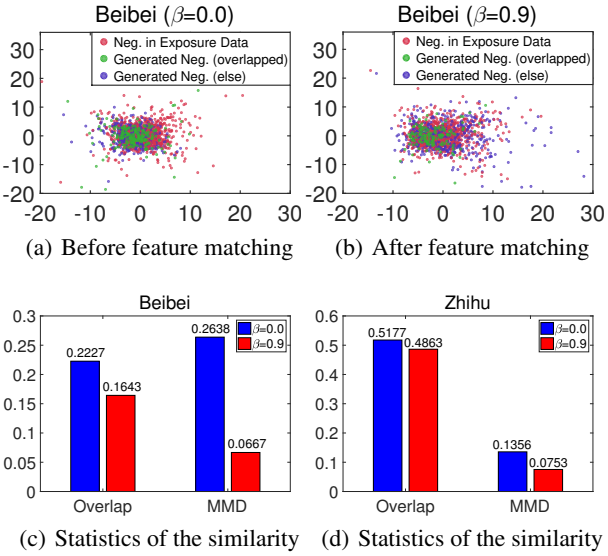


Figure 2: Impact of weight β on RNS-ES's performance.

conducting experiments on RNS-ES that only retain the exposure-matching objective. Figure 2 shows the recommendation performance with respect to different β. We can clearly observe that increasing β from 0.0 to 0.9 can effectively improve the NDCG. More specifically, it increases from 0.3986 to 0.4060 (+1.86%) and 0.6555 to 0.6605 (+0.76%) on Beibei and Zhihu, respectively. As β denotes the weight of MMD-based objective for the feature matching, a peak with large value (0.8∼0.9) highlights the advantage of our proposed feature matching scheme.

To further illustrate its advantage, we randomly select two subsets of generated and exposed instances on Beibei, 1000 each, and visualize their interacted feature vectors before ($\beta = 0.0$) and after ($\beta = 0.9$) feature matching in Figure 3(a) and (b), respectively, via t-SNE. With red points as the exposed instances, blue points as the generated instances and green points as those overlapped ones, we observe a much more similar distribution in Figure 3(b) . On the one hand, the generated negatives (blue) are much closer to exposed ones (red) when $\beta = 0.9$, while they are densely gathered in a small area near $(0, 0)$ when $\beta = 0.0$. On the other hand, the number of overlapped instances (green) decreases significantly with β increasing from 0.0 to 0.9. To measure the similarity between two sets of generated and exposed instances, we calculate two metrics of overlap and MMD and illustrate them in Figure 3(c) and (d), where the smaller value of both overlap and MMD can be observed with $\beta = 0.9$. This finding is interesting and insightful, implying that the proposed feature matching scheme encourages the sampler to focus on generating the negative instances that are similar to exposed instances in the feature space, rather than choosing exposed instances directly. By this means, although the overlap between generated negatives and exposure data is small, they are more similar in the distribution perspective (small MMD).



(a) Before feature matching (b) After feature matching

(c) Statistics of the similarity (d) Statistics of the similarity

Figure 3: 2d t-SNE visualizations of the feature vectors and statistics of the similarity (overlapMMD), before and after feature matching.

4 Related Work

Generating Adversarial Samples for Recommendation. A typical approach is the dynamic negative sampling strategy that generates hard negative instances to construct informative item pairs during the training process [Zhang *et al.*, 2013; Zhang *et al.*, 2019a]. Recently, GAN-based approach has also been adopted in training better recommender models with adversarial instances [Wang *et al.*, 2017; Chae *et al.*, 2018; Park and Chang, 2019]. As we discussed in methodology section, our RNS model differs from them by generating both hard and real negative instances with exposure data.

Recommendation with Exposure Data. There exists no previous work on training a better negative sampler with user exposure. Due to lack of data, most works use probabilistic approach to model user exposure as a latent variable and infer its value from interaction data [Liang *et al.*, 2016] or social relationship [Chen *et al.*, 2019]. Previous works using exposure data in recommendation are not related to user preference learning. Lee *et al.* [2014] propose a re-ranking approach based on items' historical impressions (*i.e.*, exposure). Zhao *et al.* [2018] investigate reasons behind user inaction. In contrast, our RNS improves the performance by leveraging the negative preference information existed in exposure data, which cannot be simply integrated like other auxiliary information in multiple feedback recommendation [Ding *et al.*, 2018b; Gao *et al.*, 2019].

5 Conclusion and Future Work

We study the problem of learning to sample negative instances for recommendation from exposure data, rather than manually designing sampling heuristics. To generate both hard and real negatives, we propose a RNS model that combines adversarial training and feature matching together, which is trained with RL method. With these designs, RNS can not only achieve higher accuracy, but also be applied to any recommender systems with negative sampling. In the future, we plan to learn more general negative samplers for social-aware or context-rich recommender systems [Lin *et al.*, 2019; Zhang *et al.*, 2019b], and other related fields such as network embedding and NLP.

Acknowledgments. This work was supported in part by the National Nature Science Foundation of China under 61861136003, 61621091 and 61673237, Beijing National Research Center for Information Science and Technology under 20031887521, and research fund of Tsinghua University - Tencent Joint Laboratory for Internet Innovation Technology.

References

[Bayer *et al.*, 2017] Immanuel Bayer, Xiangnan He, Bhargav Kanagal, and Steffen Rendle. A generic coordinate descent framework for learning from implicit feedback. In *WWW*, 2017.

[Cai and Wang, 2018] Liwei Cai and William Yang Wang. Kbgan: Adversarial learning for knowledge graph embeddings. In *NAACL*, 2018.

[Caselles-Dupré *et al.*, 2018] H. Caselles-Dupré, F. Lesaint, and J. Royo-Letelier. Word2vec applied to recommendation: Hyperparameters matter. In *RecSys*, 2018.

[Chae *et al.*, 2018] Dong-Kyu Chae, Jin-Soo Kang, Sang-Wook Kim, and Jung-Tae Lee. Cfgan: A generic collaborative filtering framework based on generative adversarial networks. In *CIKM*, 2018.

[Chen *et al.*, 2019] Jiawei Chen, Can Wang, Sheng Zhou, Qihao Shi, Yan Feng, and Chun Chen. Samwalker: Social recommendation with informative sampling strategy. In *WWW*, 2019.

[Ding *et al.*, 2018a] Jingtao Ding, Fuli Feng, Xiangnan He, Guanghui Yu, Yong Li, and Depeng Jin. An improved sampler for bayesian personalized ranking by leveraging view data. In *WWW Companion*, 2018.

[Ding *et al.*, 2018b] Jingtao Ding, Guanghui Yu, Xiangnan He, Yuhan Quan, Yong Li, Tat-Seng Chua, Depeng Jin, and Jiajie Yu. Improving implicit recommender systems with view data. In *IJCAI*, 2018.

[Gao *et al.*, 2019] Chen Gao, Xiangnan He, Dahua Gan, Xiangning Chen, Fuli Feng, Yong Li, Tat-Seng Chua, and Depeng Jin. Neural multi-task recommendation from multi-behavior data. In *ICDE*, 2019.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[He *et al.*, 2017] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *WWW*, 2017.

[He *et al.*, 2018] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. Adversarial personalized ranking for recommendation. In *SIGIR*, 2018.

[Jiang *et al.*, 2018] Zhengshen Jiang, Hongzhi Liu, Bin Fu, Zhonghai Wu, and Tao Zhang. Recommendation in heterogeneous information networks based on generalized random walk model and bayesian personalized ranking. In *WSDM*, 2018.

[Lee *et al.*, 2014] Pei Lee, Laks VS Lakshmanan, Mitul Tiwari, and Sam Shah. Modeling impression discounting in large-scale recommender systems. In *KDD*, 2014.

[Li *et al.*, 2015] Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In *ICML*, 2015.

[Lian *et al.*, 2017] Jianxun Lian, Fuzheng Zhang, Min Hou, Hongwei Wang, Xing Xie, and Guangzhong Sun. Practical lessons for job recommendations in the cold-start scenario. In *Recsys Challenge*, 2017.

[Liang *et al.*, 2016] Dawen Liang, Laurent Charlin, James McInerney, and David M Blei. Modeling user exposure in recommendation. In *WWW*, 2016.

[Lin *et al.*, 2019] Tzu-Heng Lin, Chen Gao, and Yong Li. Cross: Cross-platform recommendation for social e-commerce. In *SIGIR*, 2019.

[Pan *et al.*, 2008] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. One-class collaborative filtering. In *ICDM*, 2008.

[Park and Chang, 2019] Dae Hoon Park and Yi Chang. Adversarial sampling and training for semi-supervised information retrieval. In *WWW*, 2019.

[Rendle and Freudenthaler, 2014] Steffen Rendle and Christoph Freudenthaler. Improving pairwise learning for item recommendation from implicit feedback. In *WSDM*, 2014.

[Rendle *et al.*, 2009] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, 2009.

[Sutton *et al.*, 2000] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, 2000.

[Wang *et al.*, 2017] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *SIGIR*, 2017.

[Yang *et al.*, 2018] Longqi Yang, Yin Cui, Yuan Xuan, Chenyang Wang, Serge Belongie, and Deborah Estrin. Unbiased offline recommender evaluation for missing-not-at-random implicit feedback. In *RecSys*, 2018.

[Zhang *et al.*, 2013] Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *SIGIR*, 2013.

[Zhang *et al.*, 2017] Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. Adversarial feature matching for text generation. In *ICML*, 2017.

[Zhang *et al.*, 2019a] Yongqi Zhang, Quanming Yao, Yingxia Shao, and Lei Chen. Nscaching: Simple and efficient negative sampling for knowledge graph embedding. In *ICDE*, 2019.

[Zhang *et al.*, 2019b] Zhiqian Zhang, Chenliang Li, Zhiyong Wu, Aixin Sun, Dengpan Ye, and Xiangyang Luo. Next: a neural network framework for next poi recommendation. *Frontiers of Computer Science*, 2019.

[Zhao *et al.*, 2018] Qian Zhao, Martijn C Willemsen, Gediminas Adomavicius, F Maxwell Harper, and Joseph A Konstan. Interpreting user inaction in recommender systems. In *RecSys*, 2018.