Sequential-based Adversarial Optimisation for Personalised Top-N Item Recommendation

Jarana Manotumruksa
University College London
j.manotumruksa@ucl.ac.uk

Emine Yilmaz
University College London, Amazon
emine.yilmaz@ucl.ac.uk

ABSTRACT

Personalised top-N item recommendation systems aim to generate a ranked list of interesting items to users based on their interactions (e.g. click, purchase and rating). Recently, various sequential-based factorised approaches have been proposed to exploit deep neural networks to effectively capture the users' dynamic preferences from their sequences of interactions. These factorised approaches usually rely on a pairwise ranking objective such as the Bayesian Personalised Ranking (BPR) for optimisation. However, previous works have shown that optimising factorised approaches with BPR can hinder the generalisation, which can degrade the quality of item recommendations. To address this challenge, we propose a Sequential-based Adversarial Optimisation (SAO) framework that effectively enhances the generalisation of sequential-based factorised approaches. Comprehensive experiments on six public datasets demonstrate the effectiveness of the SAO framework in enhancing the performance of the state-of-the-art sequential-based factorised approach in terms of NDCG by 3-14%.

ACM Reference Format:
Jarana Manotumruksa and Emine Yilmaz. 2020. Sequential-based Adversarial Optimisation for Personalised Top-N Item Recommendation. In *43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3397271.3401264

1 INTRODUCTION

Personalised top-N item recommendation is an essential feature for many e-commerce, media-sharing and social networking platforms that facilitate users to find interesting items based on their historical interactions (e.g. click, purchase and rating). Matrix Factorisation (MF) is a popular Collaborative Filtering (CF) technique that is widely use to suggest relevant items to users. Previous literature[3, 4, 6, 9] have shown that the sequence of users' interactions play an important role in improving the quality of top-N item recommendation. To capture the users' dynamic preferences from their sequence of interactions, various factorised approaches have been proposed to exploit Deep Neural Networks (DNN) such as Convolutional Neural Network (CNN)[9] and Recurrent Neural Network (RNN)[3, 6]. Recently, Kang and McAuley [4] proposed a Self-Attention Sequential Recommendation (SASRec) approach that exploits attention mechanisms[10] to capture the users' dynamic preferences. They demonstrated that SASRec is more effective than existing CNN and RNN factorised approaches.

Although the use of DNN lead to better performance in many applications, previous work on adversarial machine learning[1] has shown that various DNN approaches are vulnerable to adversarial examples, which are formed by applying small but intentional perturbations to input examples. For instance, Goodfellow *et al.*[1] showed that by adding small adversarial perturbations to an image of panda, a well-trained classier inaccurately predicts the image as a gibbon with a high confidence, although such effect of perturbations on the image is hardly recognised by human. Similarly, recent studies[2, 8] on adversarial machine learning for recommendation have shown that factorised approaches optimised by Bayesian Personalised Ranking (BPR)[7] are vulnerable to the adversarial perturbations. To alleviate this problem, He *et al.*[2] proposed an Adversarial Personalised Ranking (APR) framework that adds adversarial perturbations into MF's parameters during the training process to improve the generalisation and robustness of MF. Although the effects of adversarial training for recommendation have investigated in previous literature[2, 8], none of them has demonstrated such effect on sequential-based factorised approaches.

In this work, we propose a novel Sequential-based Adversarial Optimisation (SAO) framework that aims to train sequential-based factorised approaches with adversarial perturbations. In particular, SAO generates a sequence of adversarial perturbations based on the sequence of users' interactions and add the perturbations into the model's parameters during the training process. We conduct experiments on six public datasets and demonstrate that the SAO framework significantly improve the performance of recently proposed SASRec approach. To the best of our knowledge, this work is the first that studies the effects of adversarial training for sequential-based factorised approaches.

2 METHODOLOGY

In this section, we first formalise the problem statement as well as the notations used in this paper. Next, we briefly describe the state-of-the-art Self-Attention Sequential Recommendation (SASRec) framework. Then, we describe the Sequential-based Adversarial Optimisation (SAO) framework that aims to improve the effectiveness of SASRec for top-N item recommendation.

2.1 Problem Statement and Notations

The task of top-N item recommendation is to generate a ranked list of items that a user might interest given her sequence of historical interactions (e.g. previously purchased items). Let $r_{u,i} \in R$ denote a user $u \in \mathcal{U}$ who has an interaction with item $i \in \mathcal{I}$. Let $\mathcal{I}_u^+$ denote the set of items that the user u has previously interacted and let $\mathcal{S}_u$ = denote the a sequence of user's item interaction (e.g. $S_u = [r_{i_1}, r_{i_2}, ..., r_{i_n}]$), where n is the length of sequence S_u. We use r_{i_t} to denote the t-th item interaction in the sequence.

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401264

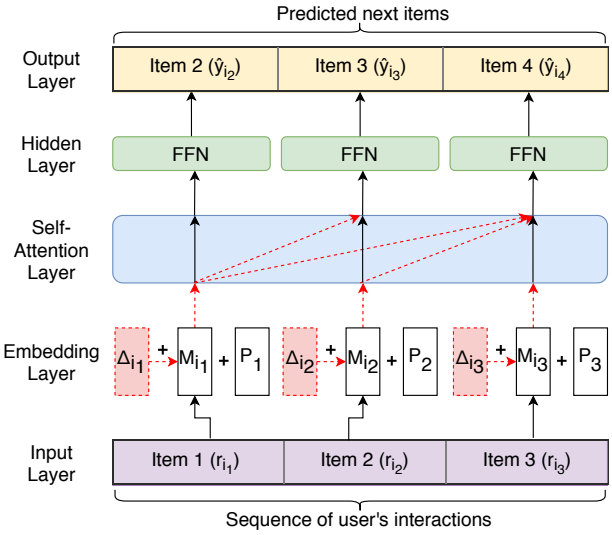

Figure 1: An overview of the SAO framework for SASRec. The sequence of adversarial perturbations Δ are added into each item embedding vector

2.2 Self-Attention Sequential Recommendation

In this section, we describe the Self-Attention Sequential Recommendation (SASRec) approach, proposed by Kang and McAuley [4], that aims to capture the users' dynamic preferences from their sequences of interactions.[1] An overview of SASRec is illustrated in Figure 1. SASRec consists of multiple layers and the output of one layer serves as the input of the layer above. Starting at the bottom of the figure, the input layer consists of a sequence of user's item interactions S_u, which is then fed into the embedding layer. In the embedding layer, there are item embedding matrix $M \in \mathcal{R}^{|\mathcal{I}|\times d}$ and position embedding matrix $P \in \mathcal{R}^{n\times d}$, where d is the dimension of latent factors. M and P matrices are used to project users' item interactions and their position in the sequence into the low dimension latent factors, respectively. In particular, each user's interaction $r_{i,t} \in S_u$ is projected into the latent factor of item $M_i \in \mathcal{R}^d$ and the latent factor of position $P_t \in \mathcal{R}^d$. Then, the output of the embedding layer, $\widehat{E}$, is defined as follows:

$$\widehat{E} = \begin{bmatrix} M_{i_1} + P_1 \\ M_{i_2} + P_2 \\ \dots \\ M_{i_n} + P_n \end{bmatrix} \tag{1}$$

Next, the embedding matrix $\widehat{E}$ is passed to the self-attention layer (see the blue rectangle in Figure 1). In particular, Kang and McAuley exploited the scaled dot-product attention [10] to capture the user's dynamic preferences by aggregating embedding matrix $\widehat{E}$ with adaptive weights as follows:

$$\begin{aligned} v_a &= Attention(\widehat{E}W^Q, \widehat{E}W^K, \widehat{E}W^V) \\ Attention(Q,K,V) &= softmax\left(\frac{QK^T}{\sqrt{d}}V\right) \end{aligned} \tag{2}$$

[1] For comprehensive detail of SASRec, we refer readers to their original paper [4].

where $W^Q, W^K, W^V \in \mathcal{R}^{d\times d}$ are weight matrices. The output of the attention layer v_a is then fed into the hidden layer to capture the correlations between different latent dimension of v_a as follows:

$$F = h_1(ReLU(h_0(v_a))) \tag{3}$$

where ReLU is the Rectified Linear Unit and $h(x) = (W^T x + b)$ is the feedforward neural network (see green box in Figure 1), W and b are the weight matrix and bias vector, respectively. Note that we can stack multiple self-attention and hidden layers into SASRec to learn more complex item transitions. The final output layer of SASRec is the predicted item interaction at time step t, $\hat{y}_{i_t}$, which is defined as follows:

$$\hat{y}_{i_t} = F_t M_{i_t}^T \tag{4}$$

where F_t is the output of the feedforward neural network at position t. The objective function of SASRec is defined as follow:

$$L_{SASRec}(\mathcal{S}|\Theta) = -\sum_{S_u\in S}\sum_{t\in[1,2,\dots,n]}\left[log(\sigma(y_{i_t})) + log(1-\sigma(y_{j_t})\right] \tag{5}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is a logistic function, $\mathcal{S}$ is a set of sequence of users' interactions, Θ is all trainable parameters of SASRec and j is an item the user has never interacted with, sampled from negative pool $\mathcal{I}^- = \mathcal{I} - \mathcal{I}_u^+$.

2.3 Sequential-based Adversarial Optimisation

In this section, we introduce the Sequential-based Adversarial Optimisation (SAO) framework that aims to improve the effectiveness of SASRec for top-N item recommendation. An overview of SAO is illustrated in Figure1. First, inspired by [2], we modify the objective function of SASRec (Equation (2.2)) such that by optimizing it, SASRec is both suitable for top-N item recommendation and robust to the adversarial perturbations as follow:

$$L_{SAO}(S|\Theta) = L_{SASRec}(S|\Theta) + \lambda L_{SASRec}(S|\Theta + \Delta_{adv}) \tag{6}$$

where Δ_{adv} denotes the adversarial perturbations on SASRec's parameters and λ is a hyperparameter that controls the influence of the adversarial term $L_{SASRec}(S|\Theta+\Delta_{adv})$. Next, given a sequence of user's interaction $S_u = [r_{i_1}, r_{i_2}, ..., r_{i_n}]$, the problem of constructing adversarial perturbations Δ_{adv} can be defined as follows:

$$l_{adv}(S_u|\Delta) = -\sum_{t\in[1,2,\dots,n]} \lambda log(\sigma(y_{i_t}(\hat{\Theta}+\Delta))) + log(1-\sigma(y_{j_t}(\hat{\Theta}+\Delta))) \tag{7}$$

where $\hat{\Theta}$ denotes a constant set of the current parameters of SASRec and Δ is the perturbations on SASRec's current parameters. We exploit the fast gradient method, proposed by Goodfellow *et al.*[1], to estimate adversarial perturbations Δ_{adv} (i.e. calculate gradient descent of $l_{adv}(S_u|\Delta)$ with respect to Δ) as follows:

$$\Delta_{adv} = \epsilon\frac{\Gamma}{||\Gamma||} \quad where \quad \Gamma = \frac{\partial l_{adv}(S_u|\Delta)}{\partial\Delta} \tag{8}$$

where $\epsilon \geq 0$ is a hyperparameter that controls the magnitude of the perturbations. Next, to add the adversarial perturbations into SASRec, we modify the embedding and output layers of SASRec

(Equation (1) and Equation (4), respectively) as follows:

$$\widehat{E} = \begin{bmatrix} \Delta_{i_1} + M_{i_1} + P_1 \\ \Delta_{i_2} + M_{i_2} + P_2 \\ \dots \\ \Delta_{i_n} + M_{i_n} + P_n \end{bmatrix} \tag{9}$$

$$\hat{y}_{i_t}(\Theta + \Delta_{adv}) = F_t(M_{i_t} + \Delta_{i_t})^T$$

where $\Delta_{i_t} \in \Delta_{adv}$ is the perturbation vector for item i at position t. Finally, to learn SASRec's parameters Θ, given a sequence of user's interaction $S_u = [r_{i_1}, r_{i_2}, ..., r_{i_n}]$, we aim to optimise the following objective function:

$$l_{SAO}(S_u|\Theta) = -\sum_{t\in[1,2,...,n]} \Big[log(\sigma(y_{i_t}(\Theta)) + log(1 - \sigma(y_{j_t}(\Theta)\Big] - \lambda \Big[log(\sigma(y_{i_t}(\Theta + \Delta_{adv})) + log(1 - \sigma(y_{j_t}(\Theta + \Delta_{adv})\Big] \tag{10}$$

In this problem, we freeze Δ_{adv} to be constant and update Θ by performing stochastic gradient descent.

$$\Theta = \Theta - \eta \frac{\partial l_{SAO}(S_u|\Theta)}{\partial \Theta} \tag{11}$$

where η is a learning rate. The overall learning process for SASRec with SAO is summarised in Algorithm 1. We note that the SAO framework differs from the adversarial optimisation (APR)[2], in several aspects. Regarding the adversarial perturbation construction, SAO takes the sequential properties of users' interactions S_u into account during the construction process (see Equation (2.3)), whereas APR does not. In addition, during the prediction time, SAO can leverage the adversarial perturbations from the previous positions (see red-dashed lines in Figure 1), while APR cannot.

Algorithm 1 Learning process for SASRec with SAO

```
Input: training data S, hyperparameters η, λ, ϵ
Output: SASRec's parameters Θ
Initialise Θ from pre-trained SASRec
repeat
    for S_u ∈ S do
        for t ← 1 to n do
            j ← draw an item from I_u^-
            Compute adversarial perturbations Δ_{i_t} and Δ_{j_t}
            // Equation (8)
            Add adversarial perturbations into Θ
            // Equation (9)
            Updated the SASRec's parameters Θ
            // Equation (11)
        end for
    end for
until convergence
```

3 EVALUATION

In this section, we evaluate the usefulness of the SAO framework in enhancing the performance of SASRec for top-N item recommendation. In particular, we aim to address the following research question, **RQ** Can we leverage the sequence of adversarial perturbations to improve the performance of SASRec? From now on, we term SASRec with SAO as Adversarial Self-Attention Sequential Recommendation (**ASASRec**).

Table 1: Statistics of the six used datasets.

	ML-1M	Beauty	Video	Brightkite	Foursquare	Yelp
#users	6,040	52,204	31,013	14,374	10,766	38,945
#items	3,706	57,288	23,714	5,050	10,695	34,245
#interactions	994,169	342,704	256,094	681,024	1.3M	981,379
avg. user's interactions	164	6	8	52	124	27
avg. item's interactions	268	6	11	134	124	27

3.1 Experimental Setup

We conduct experiments using six publicly available datasets. These datasets represent different item recommendation scenarios for business, product and location check-in, which vary significantly in domains, platforms and sparsity. In particular, to show the generalisation of SAO across multiple platforms and sources of feedback evidence, we use two product datasets from Amazon (Beauty and Video categories), two checkin datasets from Brightkite and Foursquare and two rating dataset from Yelp and MovieLens. We follow the common practice from previous works [2, 5–7] to remove items with less than 10 interactions. Table 1 summarises the statistics of the filtered datasets. We adopt a *leave-one-out* evaluation methodology: for each user, we select their most recent interaction as a ground truth for the testing set, where the remaining interactions are used as the training set. The top-N item recommendation task is thus to rank all items that are not interacted by the user in the training set. Following previous literature [2, 4, 5], we use Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) metrics to measure the quality of the ranked list of items. We conduct significance tests using a paired t-test.

We implement the SAO framework using Tensorflow and release our implementations as open source [2]. The summary of the baselines are described below. **MostPop** is a non-personalised approach that ranks items based on their popularity, computed by the number of interactions for each item in the training dataset. **BPR** is the classical pairwise ranking approach, coupled with Matrix Factorisation, proposed by Rendle *et al.*[7]. **APR** is an Adversarial Personalised Ranking approach, proposed by He *et al.*[2], that applies adversarial perturbations on the BPR approach. **SASREC** is a Self-Attention Sequential Recommendation approach[3] proposed by Kang and McAuley [4] (for more details see Section 2.2)

Note that we ignore existing sequential-based DNN approaches (e.g. GRU4Rec[3] and Caser[9]) since Kang and McAuley [4] have already demonstrated that SASRec significantly outperforms these approaches. The hyperparameters of SASRec and APR. Following [2, 4], for a fair comparison, we set the dimension of the latent factors d of all approaches to be identical: $d = 64$ across six datasets. We initially set learning rate $\eta = 0.05$ and set the batch size to 512. Note that both APR and SAO have two hyperparameters (ϵ and λ), following [2], we set $\epsilon = 0.5$ and $\lambda = 1$ for all datasets. The maximum sequence length n is set to 50.

3.2 Experimental Results

Tables 2 reports the effectiveness of various approaches in term of the HR@10 and NDCG@10 on the six used datasets. Firstly, we note that the relative top-N item recommendation quality of the baselines on the six datasets in terms of the two measures are consistent with the results reported for the various baselines in the corresponding literature [2, 4]. For instance, APR outperforms BPR

[2] https://github.com/feay1234/ASASRec [3] https://github.com/kang205/SASRec

Table 2: Performance in terms of HR and NDCG between various approaches. The best performing result is highlighted in bold; $*$ denote a significant difference compared to the second best performing result, according to the paired t-test for $p < 0.05$.

	MovieLens		Beauty		Video		Foursquare		Brightkite		Yelp	
Model	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG
MostPop	0.2295	0.0562	0.0367	0.0089	0.0879	0.0248	0.1137	0.0847	0.2372	0.1157	0.0775	0.0200
BPR	0.4111	0.1069	0.0841	0.0243	0.2344	0.0646	0.8628	0.5956	0.7064	0.5424	0.2488	0.0677
APR	0.4156	0.1093	0.0914	0.0269	0.2481	0.0693	0.8674	0.5696	0.7071	0.5318	0.2618	0.0700
SASRec	0.6272	0.1957	0.1223	0.0368	0.3105	0.0853	0.8674	0.6833	**0.7395**	0.5651	0.2685	0.0763
ASASRec	**0.6482***	**0.2114***	**0.1374***	**0.042***	**0.3356***	**0.0974***	**0.8760***	**0.7089***	0.7340	**0.5714***	**0.2888***	**0.0841***

on the Yelp and Pinterest datasets[4]. Moreover, SASRec outperforms non sequential-based approaches (BPR and APR) across all datasets. Next, comparing ASASRec with the various baselines, we observe that ASASRec consistently and significantly outperforms BPR, APR and SASRec for HR and NDCG, across all datasets, except for HR on the Brightkite dataset, where SASRec is statistically indistinguishable from ASASRec (difference in HR < 1%). In particular, on the two product datasets from Amazon (Beauty and Video), ASASRec improves NDCG by approximately 14% over SASRec. On the two checkin datasets from Brightkite and Foursquare, ASASRec improves NDCG by approximately 1-3% over SASRec. Next, we note that unlike the Brightkite and Foursquare checkin datasets, the Yelp and MovieLens datasets consists of only user-item ratings, and hence the sequential properties of users' interactions are less likely to be observed. On Yelp and MovieLens datasets, ASASRec significantly improves NDCG and HR by 7-10% and 3-7%, respectively, over SASRec. Overall, these results imply that SAO is useful in enhancing the performance of SASRec for top-N item recommendation.

Next, we investigate the effect of adversarial learning on both BPR and SASRec approaches. Figure 2 shows the performance of BPR, APR, SASRec and ASASRec on the Video and Yelp datasets in terms of HR and NDCG. In particular, we first train BPR and SASRec for 1,000 epochs and then continue training BPR and SASRec with the APR and SAO framework, respectively. From Figure 2, after 1,000 epochs, by further training both BPR and SASRec with APR and SAO, respectively, significant improvements of these two approaches in terms of HR and NDCG can be observed. In contrast, further training both BPR and SASRec without the adversarial perturbations after 1,000 epochs does not lead to any improvements. Therefore, in response to research question **RQ**, we find that the sequence of adversarial perturbations effectively improve the performance of state-of-the-art sequential-based approach (SASRec) across six different datasets in terms of HR and NDCG.

4 CONCLUSION

In this paper, we proposed a novel Sequential-based Adversarial Optimisation (SAO) framework that adds the sequence of adversarial perturbations into the recently proposed Self-Attention Sequential Recommendation (SASRec) to enhance SASRec's performances. Our comprehensive experiments on six large-scale datasets from the MovieLens, Amazon, Brightkite, Foursquare and Yelp demonstrate the effectiveness of SAO. In particular, on the Amazon datasets (Beauty and Video), SAO significantly improves NDCG approximately 14% over SASRec. For future work, we plan to study how to effectively add adversarial perturbations into the deep neural network layers of SASRec (i.e. the self-attention and hidden layers).

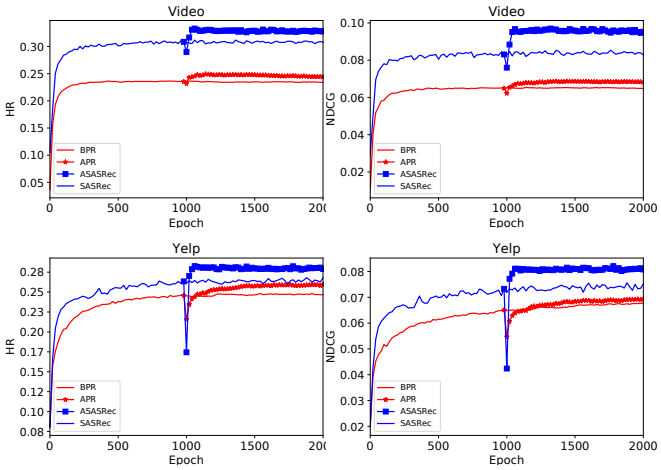

Figure 2: Training curves of BPR, APR, SASRec and ASASRec on the Video and Yelp datasets.

ACKNOWLEDGEMENT

This project was funded by the EPSRC Fellowship titled "Task Based Information Retrieval", grant reference number EP/P024289/1.

REFERENCES

[1] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. In *Proc. of ICLR*.
[2] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. 2018. Adversarial personalized ranking for recommendation. In *Proc. of SIGIR*. 355–364.
[3] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *Proc. of ICLR*.
[4] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *Proc. of ICDM*. IEEE, 197–206.
[5] Jarana Manotumruksa, Craig Macdonald, and Iadh Ounis. 2017. A Deep Recurrent Collaborative Filtering Framework for Venue Recommendation. In *Proc. of CIKM*.
[6] Jarana Manotumruksa, Craig Macdonald, and Iadh Ounis. 2018. A contextual attention recurrent architecture for context-aware venue recommendation. In *Proc. of SIGIR*. 555–564.
[7] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proc. of UAI*.
[8] Jinhui Tang, Xiaoyu Du, Xiangnan He, Fajie Yuan, Qi Tian, and Tat-Seng Chua. 2019. Adversarial training towards robust multimedia recommender system. In *Proc. of TKDE*.
[9] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proc. of WSDM*. 565–573.
[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. of NIPS*. 5998–6008.

[4] These two datasets were used in the experiments of the original paper.