Adversarial Oracular Seq2seq Learning for Sequential Recommendation

Pengyu Zhao* , **Tianxiao Shui*** , **Yuanxing Zhang*** , **Kecheng Xiao** and **Kaigui Bian**
School of EECS, Peking University, Beijing, China
{pengyuzhao, stx_pkucs, longo, kecheng, bkg}@pku.edu.cn

Abstract

Recently, sequential recommendation has become a significant demand for many real-world applications, where the recommended items would be displayed to users one after another and the order of the displays influences the satisfaction of users. An extensive number of models have been developed for sequential recommendation by recommending the next items with the highest scores based on the user histories while few efforts have been made on identifying the transition dependency and behavior continuity in the recommended sequences. In this paper, we introduce the Adversarial Oracular Seq2seq learning for sequential Recommendation (AOS4Rec), which formulates the sequential recommendation as a seq2seq learning problem to portray time-varying interactions in the recommendation, and exploits the oracular learning and adversarial learning to enhance the recommendation quality. We examine the performance of AOS4Rec over RNN-based and Transformer-based recommender systems on two large datasets from real-world applications and make comparisons with state-of-the-art methods. Results indicate the accuracy and efficiency of AOS4Rec, and further analysis verifies that AOS4Rec has both robustness and practicability for real-world scenarios.

1 Introduction

Interaction sequences are commonly used on online services to track user behaviors towards Internet content. Sequential recommendation is one of the most successful recommending strategies in both industry and academia, where the sequence could be taken to characterize users' preferences and avoid exploiting sensitive information. The sequential recommender system has been widely used in online services such as E-commerce platforms and on-demand video services. Recommender systems should identify the pattern of user behaviors and infer the dynamics of user interests to improve users' engagement and potential consumption.

There have been many instances of outbreaks on improving the accuracy of sequential recommendation, where the recommender system can be built upon long/short-term interest [Li *et al.*, 2017a], context features [Tang and Wang, 2018], knowledge graph [Xu *et al.*, 2019], etc. Beyond that, researchers have made efforts on model ensemble from various aspects and injection of prior knowledge [Ren *et al.*, 2019], in expectation of further increment of accuracy and diversity[Wu *et al.*, 2019]. [Zhang *et al.*, 2020] further analyses on the sequence-level representation endow with explanation, making the recommendation much convincing.

We observe that the items which users might be interested in are usually in specific orders, i.e., there exists order effect among items for a specific user. Meanwhile, recommending a sequence of items to be displayed one after another is necessary for many mobile applications such as news feed and short-video feed. However, most previous works focus on predicting the next items based on the historical sequences in sequential recommendation, and select the top-K items based on the similarity between the feature vector and item embedding. Hence, these recommendations neglect the sequence-level *behavior continuity* and the *transition dependency* between the items in the future interactions. Enlightened by recent advances in sequence-to-sequence (seq2seq) learning [Sutskever *et al.*, 2014], we attempt to retrieve the user preference and yield the subsequent sequence of items to be displayed to users with specific order in a generative way.

In light of this, we propose the Adversarial Oracular Seq2seq learning for sequential Recommendation (AOS4Rec), which integrates the sequence-level oracle [Zhang *et al.*, 2019] and adversarial training [Goodfellow *et al.*, 2014] into the seq2seq auto-regressive learning. Specifically, we formulate the sequential recommendation as a seq2seq learning problem, i.e., seq2seq recommendation, and model the sequence-level continuity and the transition dependency via the auto-regression. To solve the inherent exposure bias [Ranzato *et al.*, 2015] in the auto-regressive learning and optimize the recommendation on the sequence level, we introduce the oracular learning to fill the gap between training and recommendation, and then propose a novel seq2seq-recommendation generative adversarial network (SSRGAN) to optimize the entire recommended sequence with the discriminator score. Evaluation over YOOCHOOSE and MovieLens-20M datasets verifies that AOS4Rec performs

*Equal Contribution

well on the accuracy metrics and ranking metrics, outperforming several state-of-the-art recommender systems. Further analysis indicates that adversarial learning and oracular learning can boost the performance of seq2seq recommendation, and the proposed AOS4Rec is available and efficient to be used in real-world applications.

The contribution can be summarized as follows:

- We propose a seq2seq learning strategy for sequential recommendation, which yields a sequence of items consistent with the user preference in sequence level rather than on next-item prediction.
- We present the adversarial oracular learning over the seq2seq recommendation, reducing the exposure bias in the auto-regression while improving the integrality in the recommended sequences.
- Evaluation on large-scale datasets verifies that the proposed method outperforms several state-of-the-art recommender systems with a similar complexity.

2 Related Work

Sequential recommendation. Many approaches have been proposed to recognize the sequential patterns of users and make recommendation, where implicit user behaviors are passively tracked over a sequence of time. Common solutions to sequential recommendation include modelling item-to-item similarity [Xu *et al.*, 2019], identifying sequential context [Hidasi and Karatzoglou, 2018], and fitting knowledge-based behavior pattern [Ren *et al.*, 2019]. Several works attempt to retrieve the main preference within the sequence by attention mechanism [Li *et al.*, 2017a; Kang and McAuley, 2018] or convolutional embedding [Tang and Wang, 2018], and fine-tune the model by pretrained embedding [Sun *et al.*, 2019]. However, most of the sequential recommender systems aim at recommending a list of next items sorted by the predicted scores. They are not appropriate for the emerging feed-based mobile applications where the order of the item sequence is significant. The proposed AOS4Rec considers the *transition dependency* and *behavior continuity* between consecutive recommended items and is capable of yielding appropriate items in sequence.

Adversarial learning for recommendation. Recently, adversarial learning have been introduced to the recommendation task, where a recommender yields the next item via generative manner and a discriminator aims to distinguish the ground truth from the generated items [Wang *et al.*, 2017]. Adversarial learning have been successfully used to improve the performance of recurrent recommendation [Bharadhwaj *et al.*, 2018], enhance the robustness of the pairwise ranking [He *et al.*, 2018], and contribute to the point-of-interest recommendation by exploiting the spatio-temporal information [Zhou *et al.*, 2019]. Besides, the generative design can also introduce diversity to the recommendation [Wu *et al.*, 2019]. AOS4Rec benefits from the adversarial learning, which uses Wasserstein GAN (WGAN) [Arjovsky *et al.*, 2017] and actor-critic algorithm to achieve identical optimization objective as well as the fast and stable training.

Sequence-to-sequence learning. Seq2seq model is verified to be promising on many natural language generation tasks [Sutskever *et al.*, 2014], which encodes the input sequence into a single vector and recurrently generates the outputs to form a sequence. In practice, the sequence generation is executed by searching over the output sequences greedily via beam search. Seq2seq techniques are recently used on recommender systems to investigate the transition dependency [Yu *et al.*, 2019] among the items and present users with items in an order [Sun and Qian, 2019]. In this paper, we take advantage of seq2seq models with adversarial oracular learning to improve the sequential recommendation.

3 Seq2seq Recommendation

3.1 Formulation of Sequential Recommendation

Let $\mathcal{U} = \{u_1, u_2, ..., u_{|\mathcal{U}|}\}$ denote the set of users and $\mathcal{I} = \{i_1, i_2, ..., i_{|\mathcal{I}|}\}$ denote a universe of items. Each user $u \in \mathcal{U}$ has interacted with a sequence of items $\mathcal{S}^u = (i^u_1, i^u_2, ..., i^u_{|\mathcal{S}^u|})$ in order. Given the user interaction history $\mathbf{x} = (x_1, x_2, ..., x_{|\mathbf{x}|}) \in \{\mathcal{S}^{u_1}, \mathcal{S}^{u_2}, ..., \mathcal{S}^{u_{|\mathcal{U}|}}\}$, the goal is to recommend a sequence of items $\mathbf{y} = (y_1, y_2, ..., y_{|\mathbf{y}|})$ that conform to the user interest by considering the sequential information. The length of the recommendation is usually set to a fixed number of T.

3.2 Limitations of Existing Sequential Methods

The existing sequential recommender systems such as [Li *et al.*, 2017a; Hidasi and Karatzoglou, 2018; Tang and Wang, 2018] mainly solve the sequential recommendation in the similar way as the next-item recommendation, i.e., encoding the historical sequence $\mathbf{x}$ into the feature vector, and then selecting the top-K items based on the similarity between the feature vector and item embeddings or the probability distribution upon the candidate item set. However, the results might be sub-optimal because the items are derived individually from the item set, where the mutual effect between the successive items, which is called as *transition dependency*, are neglected. Besides, these works mostly focus on monitoring the sequence-level representation of the previous interest rather than the future interest, while owing to the *behavior continuity*, user behaviors would appear following the short-term interest and interest drifting, providing evidence for recurrently generating a sequence of appropriate items rather than selecting the top-K items. For example, in the scenarios like news feed and short-video feed where the users watch the news/videos in the recommended order, recommender systems are expected to generate a sequence of items with sequential impact, while directly generating a list of items sorted by "score" goes against the requirement of order among items. Hence, the recommender system should take into account the sequence-level behavior continuity and transition dependency, and attempt to simulate the user behaviors to make better-personalized recommendation.

Considering the sequential influence in the recommendation, we conform to the sequence-to-sequence (seq2seq) learning [Sutskever *et al.*, 2014] in neural language processing and formulate the sequential recommendation as a *seq2seq recommendation* problem.

3.3 Seq2seq Learning for Recommendation

The existing seq2seq learning are mainly modelled by auto-regression. In the seq2seq recommendation, the auto-regressive model factorizes the joint probability $P(\mathbf{y}|\mathbf{x})$ into the product of probabilities over the next item in the sequence given the history interaction $\mathbf{x}$ and the context sequence $\mathbf{y}_{1:t-1}$: $P(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{|\mathbf{y}|} P(y_t|\mathbf{y}_{1:t-1}, \mathbf{x})$. Once the model has finished training, the recommended sequence is generated by the beam search of size b. As the recommendation of every step is strictly conditioned on the previous recommended items, the auto-regressive learning describes the time-varying processes in the seq2seq recommendation, i.e., the sequence-level behavior continuity as well as transition dependency between successive items. In the settings of seq2seq recommendation, the sensitive personal information other than the historical interactions is unused and unknown.

The Seq2seq recommendation model consists of both sequence encoder and sequence decoder. The sequence encoder captures the user preference and computes hidden representations from the history interaction $\mathbf{x}$, while the sequence decoder takes the context sequence $\mathbf{y}_{1:t-1}$ as input at each decoding step t to generate the next recommended item y_t. Without loss of generality, we consider two most common seq2seq models in the recommendation task, i.e., RNN-based attentive model and Transformer-based model. The RNN-based model utilizes bidirectional recurrent units as the sequence encoder to obtain the hidden states while an unidirectional recurrent units with attention mechanism is applied to recommend the target sequence of items. In this paper, the gated recurrent unit (GRU) is employed in the RNN-based model, which is common in many recurrent recommender systems. In the Transformer-based model [Vaswani *et al.*, 2017], the sequence encoder and decoder are both composed of identical building blocks stacked by residual connection. In the sequence encoder, the building block consists of a self-attention layer and a feed-forward layer. In the sequence decoder, an additional attention layer is inserted between the self-attention layer and feed-forward layer, which performs the attention over the output of sequence encoder. Besides, the self-attention layer is masked to prevent current item from attending to subsequent items in the decoder to guarantee the auto-regression. The positional embedding is also added on the item embeddings to make use of the order information in the item sequence. The seq2seq recommendation is usually trained by maximum likelihood estimation (MLE).

However, the above auto-regressive learning still suffers from two major limitations: (1) The context sequence of auto-regression is usually set to the ground-truth prefix, denoted as $\mathbf{y}^*_{1:t-1}$, during the training time. Meanwhile, the entire sequence, denoted as $\hat{\mathbf{y}}$, is generated by the sequence decoder when executing recommendation such that the previous generated sequence $\hat{\mathbf{y}}_{1:t-1}$ must be treated as the context. Hence, the context items are drawn from the data distribution at training stage, as opposed to the model distribution drawn at the recommendation stage, leading to the discrepancy of exposure bias [Ranzato *et al.*, 2015] between the two stages. (2) Although the auto-regressive loss can capture the behavior continuity and transition dependency in the sequence generation, the MLE loss might not be a good objective for the seq2seq recommendation because it only tries to imitate the subsequent user behaviors over the historical interactions, which is too simple to measure the sequence-level generation quality from a global perspective in the data distribution.

4 Adversarial Oracular Learning

In this section, we introduce Adversarial Oracular Seq2seq learning for sequential Recommendation (AOS4Rec), which enhances the performance of the seq2seq recommendation via the sequence-level oracle and adversarial learning.

4.1 Sequence-level Oracle for Auto-regression

To directly alleviate the negative impacts from the seq2seq learning, we conform to [Zhang *et al.*, 2019] and introduce sequence-level oracle in the auto-regression.

Oracular learning in seq2seq recommendation. The exposure bias in the auto-regressive learning comes from the inconsistent distribution of context input between the training stage and the recommendation stage. If the context is fed with both ground truth items and the recommended items in the training stage, the exposure bias will be mitigated. The above insight inspires the oracular learning. Specifically, an item sequence of length T, named oracle recommendation, is selected from the model distribution, and then mixed with the ground truth recommendation to form the context sequences. Denote the oracle recommendation as $\mathbf{y}^o = \{y^o_1, ..., y^o_T\}$ and the complete context sequence as $\mathbf{y}^c = \{y^c_1, ..., y^c_T\}$. During each decoding step j, the context item y^c_j is selected from the ground truth recommendation y^*_j with a probability of p or from the oracle recommendation y^c_j with a probability of $1 - p$. Then, the prefixes of $\mathbf{y}^c$ are fed to sequence decoder during the training time to replace the original ground-truth context, and maximize the likelihood of the ground truth sequence as the standard seq2seq learning. At the beginning of training, the ground truth recommendation should be given a higher probability to help the model learn more from the data distribution and avoid being trapped into the local optimum. As the training goes on and the model starts to converge, the oracle recommendation should be chosen more often to handle the situations happened in the recommendation stage and reduce the gap between training and recommendation. Thus, the selection probability should be set to one at the beginning and decrease gradually. In practice, the probability p decays with the training epoch e following the sampling schedule of $p = \frac{\mu}{\mu + \exp(e/\mu)}$, where μ is a tunable hyper-parameter.

Oracle selection by BLEU score. To select an appropriate sequence as the oracle, we adopt the BLEU score [Papineni *et al.*, 2002] in machine translation. The BLEU computes the n-gram overlap between model generation and reference, and counts the number of position-independent matches. It utilizes a modified n-gram precision to avoid generating unreasonable items in successive subsequence. Specifically, it counts the n-gram occurrences in the reference and then clips the count of generated n-grams by the reference count. The clipped n-gram counts are divided by the number of

the origin n-grams counts to compute a precision score p_n. Then, the BLEU score can be simply computed by the exponential weighted sum of the precision scores: $\text{BLEU}_{\text{rec}} = \exp(\sum_{n=1}^{N} \log p_n / N)$. It can be inferred that the consideration of BLEU allows for more flexible recommendation with the sequence-level n-gram matching and thus encourages the model to capture the transition dependency and behavior continuity in the auto-regressive learning. To equip the oracular learning with BLEU score, we perform a beam search of size b, which is the same as the recommendation stage, to select the candidate recommended sequences (each of length T) from model distribution. Then, the sequence with the highest BLEU score (compared with the ground-truth) is selected as the oracle recommendation $\mathbf{y}^o$.

4.2 Adversarial Learning for Recommendation

Though oracular learning could solve the exposure bias in the auto-regressive learning, the MLE is still used as the optimization target, yet the seq2seq model suffers from the drawback (2) described in Sec. 3. Therefore, we apply the adversarial learning [Goodfellow *et al.*, 2014] to optimize the seq2seq recommendation with the sequence-level objective.

Adversarial learning for seq2seq learning. The adversarial learning consists of a generator (G) and a discriminator (D). The generator produces the output to fool the discriminator and the discriminator tries to distinguish the generation from the ground truth. In the seq2seq recommendation, the generator produces the entire recommended sequence given the interaction history while the discriminator learns to maximize the score of ground-truth $D(\mathbf{x}, \mathbf{y}^*)$ and minimize the score of generation $D(\mathbf{x}, \hat{\mathbf{y}})$, depicted by:

$$\begin{aligned}\min_G \max_D \mathbb{E}_{\mathbf{y}^* \sim p_{gt}(\mathbf{x})}[\log D(\mathbf{x}, \mathbf{y}^*)] \\ + \mathbb{E}_{\hat{\mathbf{y}} \sim p_G(\mathbf{x})}[\log(1 - D(\mathbf{x}, \hat{\mathbf{y}}))]\end{aligned} \tag{1}$$

To deal with the back-propagation problem from discrete output of sequential data, previous works [Yu *et al.*, 2017; Li *et al.*, 2017b] formulate the seq2seq learning as a Markov decision process (MDP) and train the generator with the REINFORCE algorithm [Williams, 1992]. In concrete, at each time step t, the agent (generator) possesses the state $\hat{s}_t = (\mathbf{x}, \hat{\mathbf{y}}_{1:t})$ and selects the next action (item) $\hat{y}_{t+1}$ based on the current state. The rewards except for the last time step is set to zero, and the final reward is set to $D(\mathbf{x}, \hat{\mathbf{y}})$. Owing to the sequence-level optimization, the adversarial learning could eliminate the exposure bias and optimize the seq2seq recommendation with the sequence-level measurement, i.e., the discriminator score. However, there are two fatal problems in the existing adversarial learning methods for seq2seq recommendation. First, the discriminator is trained to minimize logarithmic loss in Eqn.(1) while the generator is trained to directly maximize the discriminator score in the reinforcement learning (RL) settings, introducing discrepancy in the joint optimization of generator and discriminator. Second, the use of REINFORCE algorithm results in slow convergence and unstable training, which might not be suitable for the inherently hard-trained adversarial learning. Therefore, we introduce Seq2Seq-Recommendation Generative Adversarial Network (SSRGAN) in this paper.

SSRGAN for seq2seq recommendation. SSRGAN exploits WGAN [Arjovsky *et al.*, 2017] for the identical optimization target and actor-critic algorithm [Konda and Tsitsiklis, 2000] for fast and stable training. The training objective in SSRGAN is depicted as:

$$\min_G \max_D \mathbb{E}_{\mathbf{y}^* \sim p_{gt}(\mathbf{x})}[D(\mathbf{x}, \mathbf{y}^*)] - \mathbb{E}_{\hat{\mathbf{y}} \sim p_G(\mathbf{x})}[D(\mathbf{x}, \hat{\mathbf{y}}))]. \tag{2}$$

It can be deduced that the discrepancy is eliminated in SSRGAN since the generator and discriminator are now trained with the same objective $D(\mathbf{x}, \hat{\mathbf{y}})$ in Eqn.(2). In addition, different from previous adversarial learning approaches, the discriminator score $D(\mathbf{x}, \mathbf{y}_{1:t})$ at step $t < T$ in SSRGAN represents the estimation of state-value function for state $s_t = (\mathbf{x}, \mathbf{y}_{1:t})$, where the state-value function $V_G(s_t)$ is defined by the expected return of the generator from state s_t:

$$V_G(s_t) = \mathbb{E}_{\mathbf{y}_{t+1:T} \sim P_G(s_t)}[D(\mathbf{x}, \mathbf{y})]. \tag{3}$$

With the above definition, the bootstrap algorithms, e.g., TD-learning or Q-learning, can be applied in SSRGAN training.

Generator training in SSRGAN. To accelerate the convergence while stabilizing the seq2seq recommendation, we employ the bootstrapping actor-critic algorithm in the generator training. The gradient of generator is thereby deduced by:

$$\begin{aligned}\nabla G = \mathbb{E}_{\hat{\mathbf{y}} \sim p_G(\mathbf{x})}[\frac{1}{T}\sum_{t=1}^{T}(\gamma D(\mathbf{x}, \hat{\mathbf{y}}_{1:t}) - D(\mathbf{x}, \hat{\mathbf{y}}_{1:t-1})) \\ \nabla \log P_G(\hat{y}_t | \mathbf{x}, \hat{\mathbf{y}}_{1:t-1})],\end{aligned} \tag{4}$$

where the discount rate γ is always set to 1 in SSRGAN. Besides, note that $D(\mathbf{x}, \hat{\mathbf{y}}_{1:t-1})$ is not conditioned on the selected item y_t, so that we have:

$$\begin{aligned}&\mathbb{E}_{\hat{\mathbf{y}} \sim p_G(\mathbf{x})}[D(\mathbf{x}, \hat{\mathbf{y}}_{1:t-1}) \nabla \log P_G(\hat{y}_t | \mathbf{x}, \hat{\mathbf{y}}_{1:t-1})] \\ =&D(\mathbf{x}, \hat{\mathbf{y}}_{1:t-1}) \nabla \sum_{\hat{y}_t} P_G(\hat{y}_t | \mathbf{x}, \hat{\mathbf{y}}_{1:t-1}) = 0.\end{aligned} \tag{5}$$

Accordingly, with an accurate estimation of $V_G(s)$, the G is actually optimized by an unbiased estimation of generator gradient in Eqn.(2):

$$\begin{aligned}\nabla G &= \mathbb{E}_{\hat{\mathbf{y}} \sim p_G(\mathbf{x})}[\frac{1}{T}\sum_{t=1} D(\mathbf{x}, \hat{\mathbf{y}}_{1:t}) \nabla \log P_G(\hat{y}_t | \mathbf{x}, \hat{\mathbf{y}}_{1:t-1})] \\ &= \nabla \mathbb{E}_{\hat{\mathbf{y}} \sim p_G(\mathbf{x})}[D(\mathbf{x}, \hat{\mathbf{y}})],\end{aligned} \tag{6}$$

Discriminator training in SSRGAN. To learn the discriminator score $D(\mathbf{x}, \mathbf{y}_{1:t})$ in the actor-critic algorithm for every step t, the discriminator is now optimized by:

$$\begin{aligned}\nabla D_1 =&\mathbb{E}_{\mathbf{y}^* \sim p_{gt}(\mathbf{x})}[\sum_{t \in \{1,\dots,T\}} \nabla D(\mathbf{x}, \mathbf{y}^*_{1:t})/T] \\ &- \mathbb{E}_{\hat{\mathbf{y}} \sim p_G(\mathbf{x})}[\sum_{t \in \{1,\dots,T\}} \nabla D(\mathbf{x}, \hat{\mathbf{y}}_{1:t})/T].\end{aligned} \tag{7}$$

Obviously, the above formula is equivalent to training discriminator with Eqn.(2) if $D(\mathbf{x}, \mathbf{y}_{1:t}) = V_G(s_t)$ for every $t < T$. Moreover, an additional TD(0) error is introduced

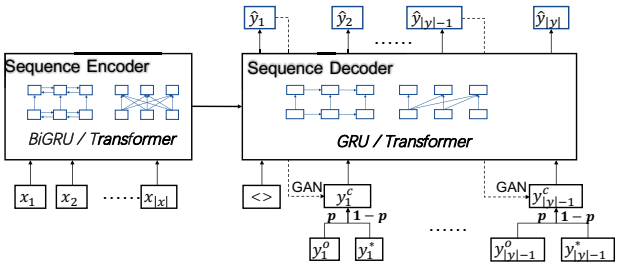

Figure 1: The architecture and training procedures of AOS4Rec. During the oracular learning, the context sequence samples from ground truth item with probability p or from the oracle item with $1-p$ at each time step. In the adversarial training, the generator takes the previous generated item as context input.

to stabilize the estimation of state-value function, where the gradient is calculated by:

$$\nabla D_2 = \frac{1}{T-1}\sum_{t=1}^{T-1}[\mathbb{E}_{y'\sim P_G(\mathbf{x},\hat{\mathbf{y}}_{1:t-1})} D(\mathbf{x},\{\hat{\mathbf{y}}_{1:t-1},y'\}) - D(\mathbf{x},\hat{\mathbf{y}}_{1:t-1})]\nabla D(\mathbf{x},\hat{\mathbf{y}}_{1:t-1}). \quad (8)$$

To compute the expectation in Eqn.(8), we sample c next items from the distribution $P_G(\mathbf{x},\hat{\mathbf{y}}_{1:t-1})$ for every step t.

4.3 Training with Hybrid Loss in AOS4Rec

In summary, we combine the seq2seq, oracular and adversarial learning in AOS4Rec, as shown in Fig. 1. First, we train the seq2seq model with the seq2seq learning and oracular learning. At each training step, the historical sequence $\mathbf{x}$ and ground truth recommendation $\mathbf{y}^*$ are sampled from the dataset. $b=5$ candidate sequences $\mathbf{y}^{can}=\{\mathbf{y}_1^{can},...,\mathbf{y}_b^{can}\}$ are generated by beam search based on the model distribution $P_G(\mathbf{x})$, and then the oracle recommendation $\mathbf{y}^o$ is selected from $\mathbf{y}^{can}$ by BLEU score and mixed with the ground truth $\mathbf{y}^*$ as the context sequence $\mathbf{y}^c$, where μ is set to 12 in the mix probability p. After the model convergence, SSRGAN is initialized by the trained seq2seq model, where the weight of generator is directly copied from the seq2seq model, and then the discriminator is pretrained with data sampled from dataset and the trained generator. Then, generator and discriminator are jointly optimized by alternative learning, where the generator gets progressed via training on g-steps = 1 updates with Eqn.(4), and the discriminator is updated by d-step = 5 with Eqn.(7) for stabilizing the training process. Moreover, discriminator is also trained by Eqn.(8) with the sampled successive item sets of size $c=3$ for an accurate state-value function estimation. Besides, Gumbel noise [Gumbel, 1948] is added in the sequence sampling for increasing both efficiency and robustness of the generation.

5 Evaluation

5.1 Settings

Datasets. We evaluate the proposed AOS4Rec with the baseline methods on two datasets from real-world applications. The datasets vary significantly in domains, behavior patterns, and sparsity: (1) *YOOCHOOSE*. The YOOCHOOSE dataset contains a collection of sessions encapsulating the click events from users. The sessions are short and the items present strong sparsity in the dataset. (2) *MovieLens*. We use MovieLens-20M Dataset, which is a stable benchmark dataset for evaluating performance of recommender systems. We follow the same preprocessing procedure from [Kang and McAuley, 2018; Xu *et al.*, 2019]. We discard users and items with fewer than 4 interactions, and then split the datasets into training sets, validation sets and test sets based on the length of sequences in the datasets, where the second last 20% items of the sequence are used for validation and the last 20% items are used for testing. As the sequence may be very long, we only take at most 6 items at the end of the sequence for validation and test respectively for YOOCHOOSE dataset, and at most 80 items at the end of the sequence for validation and test respectively for MovieLens dataset. All remaining items in the datasets are taken for training. Specific statistics are shown in Tab. 1.

	# User	# Item	Min.	Max.	Avg.
YOOCHOOSE	168202	25804	12	200	18.67
MovieLens	85307	19295	50	9254	214.08

Table 1: Statistics of the two datasets used in our experiments.

Hyper-parameter	Candidate	Selected
learning rate	3e-4, 5e-4, 7e-4, 1e-3, 3e-3	1e-3
batch size	64, 128, 256	128
beam size	1, 3, 5, 7, 10	5
weight-decay	8, 12, 16	12

Table 2: Hyper-parameter settings in AOS4Rec.

Baseline methods. To show the effectiveness of the proposed AOS4Rec, we implement three groups of recommendation baselines. The first group only considers user feedback without considering the sequence order of interactions: (1) *PopRec* is the simplest method which only recommends the most popular items (also used for evaluating the complexity of datasets); (2) *BPR* [Rendle *et al.*, 2009] is a legacy method for matrix factorization from implicit feedback. The second group takes the sequential context into accounts: (1) *GRU4REC*$^+$ [Tang and Wang, 2018] improves GRU4Rec by delicately-designed loss function and sampling strategy; (2) *SASREC* [Kang and McAuley, 2018] uses self-attention to capture the user preference within the sequence; (3) *GCSAN* [Xu *et al.*, 2019] establishes the correlation among items via graph neural network and makes recommendation by Transformer; (4) *Caser* [Tang and Wang, 2018] applies convolutional operations on the embedding matrix for sequential representation; (5) *BERT4REC* [Sun *et al.*, 2019] uses Transformer over the pretrained embeddings by BERT. The third group includes the baselines that utilize adversarial learning to generate the future items: (1) *IRGAN* [Wang *et al.*, 2017] combines generative and discriminative information retrieval in the adversarial training, with the parameters initialized with BPR. (2) *RecGAN* [Bharadhwaj *et al.*, 2018] bypasses the generator differentiation problem by directly performing policy gradient update and yields sequence of items. The above baseline methods are all designed for recommending

		PopRec	BPR	GRU4REC$^+$	SASREC	GCSAN	Caser	BERT4REC	IRGAN	RecGAN	AOS4Rec-R	AOS4Rec-T
YOOCHOOSE	Prec@6	0.109	0.163	0.492	0.531	0.526	0.514	0.527	0.366	0.498	0.539	**0.552**
	NDCG@6	0.126	0.175	0.530	0.546	0.542	0.556	0.541	0.402	0.539	0.552	**0.560**
	BLEU@6	0.002	0.004	0.056	0.063	0.091	0.057	0.060	0.017	0.098	0.128	**0.135**
MovieLens	Prec@10	0.403	0.439	0.645	0.658	0.667	0.660	0.663	0.484	0.620	0.651	**0.682**
	NDCG@10	0.424	0.448	0.621	0.623	0.643	0.634	0.645	0.491	0.619	0.660	**0.680**
	BLEU@10	0.001	0.004	0.038	0.041	0.053	0.037	0.042	0.015	0.065	**0.090**	0.074

Table 3: Performance of the recommender systems on the sequential recommendation task.

	YOOCHOOSE			MovieLens		
	Prec@6	NDCG@6	BLEU@6	Prec@10	NDCG@10	BLEU@10
RNN	0.502	0.509	0.120	0.627	0.642	0.080
+Oracle	0.515	0.512	0.118	0.630	0.653	0.083
+Adver.	0.520	0.534	0.128	0.645	0.658	**0.090**
+both	**0.539**	**0.552**	**0.128**	**0.651**	**0.660**	**0.090**
+both -beam-1	0.504	0.510	0.115	0.632	0.650	0.079
Transformer	0.532	0.537	0.116	0.660	0.631	0.061
+Oracle	0.541	0.549	0.115	0.665	0.653	0.067
+Adver.	0.547	0.558	0.129	0.671	0.667	0.071
+both	**0.552**	**0.560**	**0.135**	**0.682**	**0.680**	**0.074**
+both -beam-1	0.528	0.530	0.118	0.662	0.647	0.060

Table 4: Verification of the design in AOS4Rec.

the next items according to the corresponding output layers.

Implementation details. AOS4Rec is implemented over both RNN-based attentive model (indicated by *AOS4Rec-R*) and Transformer-based model (indicated by *AOS4Rec-T*) delineated in Sec. 3 with adversarial oracular learning and beam search. We employ grid search to find the best settings of hyper-parameters and list the details in Tab. 2. To conduct a fair comparison, the embedding size is set to be 128 for all baseline methods, so that most baselines can perform as well as possible. The training of the baselines follows the recommended procedures in the corresponding papers.

Metrics. To quantify the performance of the compared recommender systems, we adopt Precision@K to evaluate the accuracy, NDCG@K to evaluate the ranking performance and BLEU@K to assess the identification of transition dependency. Here, K is set to 6 for YOOCHOOSE and 10 for MovieLens for the consideration of differences in the sequence lengths of the datasets. To avoid heavy computation on all user-item pairs, we followed the strategy in many works such as [Kang and McAuley, 2018; Xu *et al.*, 2019], where we randomly sample negative items to construct the candidate item sets of size 1000 with ground-truth items for evaluation.

5.2 Performance of AOS4Rec

We first examine the performance of AOS4Rec with the state-of-the-art baselines. Tab. 3 demonstrates the results of the compared methods. It can be observed that AOS4Rec-R outperforms the baseline methods on YOOCHOOSE dataset, while AOS4Rec-T could surpass all the other methods on both datasets. Besides, AOS4Rec presents robustness on the dataset of either long sessions with long-term interests (i.e., MovieLens) or short sessions with mostly short-term interests (i.e., YOOCHOOSE), while some baselines (e.g., GRU4REC$^+$ and Caser) may only perform well on dataset of one attribute. Regarding the sequential metric BLEU, the improvement over the baselines is significant due to the involvement of adversarial oracular learning with the sequence-level optimization. These results verify the availability and effectiveness of AOS4Rec over both RNN-based and Transformer-based recommender systems.

5.3 Model Analysis and Discussion

We further investigate that adversarial learning and oracular learning are both essential parts of the sequential recommendation task. We experiment by adding oracular learning, adversarial learning or both of them on the vanilla seq2seq learning with RNN-based model and Transformer-based model. Besides, we also conduct an experiment by setting the beam size of all sequence generations to one in AOS4Rec, indicating the method that recurrently yields the top-1 items to form the sequence. We collect the metrics on the datasets and record the result in Tab. 4. Compared to the base models (i.e., RNN and Transformer), adversarial learning and oracular learning bring about a remarkable increase in all three metrics respectively. The combination of these two learning methods further improves the performance from the base models by relatively 3.3%$\sim$7.4% on accuracy, 2.8%$\sim$8.4% on NDCG and 6.7%$\sim$21.3% on BLEU. Beyond those, we observe that beam search is significant to the recommendation, especially for the BLEU metric.

The above findings are consistent with the intuition of AOS4Rec. The oracular learning mitigates the discrepancy between training and recommendation, while the adversarial learning ensures the generated sequence to be in a pragmatic sense. The improvements brought by those two learning strategies are orthogonal. In addition, AOS4Rec does not modify the architecture of the base models or increase the complexity of the models, revealing that AOS4Rec is available and efficient to be used in real-world applications.

6 Conclusion

This work presents the adversarial oracular seq2seq learning for optimizing sequential recommendation. The proposed AOS4Rec takes both item-level transition dependency and sequence-level behavior continuity into consideration with seq2seq auto-regressive modelling. To make the model robust and capture the sequential impact in the generated sequences, AOS4Rec utilizes the adversarial learning and oracular learning to advance the seq2seq recommendation. In our evaluation, we perform extensive analysis to verify the highly positive performance of AOS4Rec on both RNN-based and Transformer-based recommender systems.

Acknowledgments

This work is partially supported by National Key Research and Development Program No. 2017YFB0803302, Beijing Academy of Artificial Intelligence (BAAI), and NSFC 61632017.

References

[Arjovsky *et al.*, 2017] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, pages 214–223, 2017.

[Bharadhwaj *et al.*, 2018] Homanga Bharadhwaj, Homin Park, and Brian Y Lim. Recgan: Recurrent generative adversarial networks for recommendation systems. In *RecSys*, pages 372–376. ACM, 2018.

[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*, pages 2672–2680, 2014.

[Gumbel, 1948] Emil Julius Gumbel. *Statistical theory of extreme values and some practical applications: a series of lectures*, volume 33. US Government Printing Office, 1948.

[He *et al.*, 2018] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. Adversarial personalized ranking for recommendation. In *SIGIR*, pages 355–364. ACM, 2018.

[Hidasi and Karatzoglou, 2018] Balázs Hidasi and Alexandros Karatzoglou. Recurrent neural networks with top-k gains for session-based recommendations. In *CIKM*, pages 843–852. ACM, 2018.

[Kang and McAuley, 2018] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 ICDM*, pages 197–206. IEEE, 2018.

[Konda and Tsitsiklis, 2000] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *NIPS*, pages 1008–1014, 2000.

[Li *et al.*, 2017a] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM CIKM*, pages 1419–1428. ACM, 2017.

[Li *et al.*, 2017b] Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. Adversarial learning for neural dialogue generation. In *EMNLP*, 2017.

[Papineni *et al.*, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318. Association for Computational Linguistics, 2002.

[Ranzato *et al.*, 2015] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

[Ren *et al.*, 2019] Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten de Rijke. Repeatnet: A repeat aware neural recommendation machine for session-based recommendation. In *Proceedings of the AAAI*, volume 33, pages 4806–4813, 2019.

[Rendle *et al.*, 2009] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461. AUAI Press, 2009.

[Sun and Qian, 2019] Ke Sun and Tieyun Qian. Seq2seq translation model for sequential recommendation. *arXiv preprint arXiv:1912.07274*, 2019.

[Sun *et al.*, 2019] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *CIKM*, 2019.

[Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112, 2014.

[Tang and Wang, 2018] Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *WSDM*, pages 565–573. ACM, 2018.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[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*, pages 515–524. ACM, 2017.

[Williams, 1992] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[Wu *et al.*, 2019] Qiong Wu, Yong Liu, Chunyan Miao, Binqiang Zhao, Yin Zhao, and Lu Guan. Pd-gan: adversarial learning for personalized diversity-promoting recommendation. In *Proceedings of the 28th IJCAI*, pages 3870–3876. AAAI Press, 2019.

[Xu *et al.*, 2019] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. Graph contextualized self-attention network for session-based recommendation. In *Proc. 28th IJCAI*, pages 3940–3946, 2019.

[Yu *et al.*, 2017] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, 2017.

[Yu *et al.*, 2019] Lu Yu, Chuxu Zhang, Shangsong Liang, and Xiangliang Zhang. Multi-order attentive ranking model for sequential recommendation. In *AAAI*, 2019.

[Zhang *et al.*, 2019] Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. Bridging the gap between training and inference for neural machine translation. In *ACL*, 2019.

[Zhang *et al.*, 2020] Yuanxing Zhang, Pengyu Zhao, Yushuo Guan, Lin Chen, Kaigui Bian, Lingyang Song, Bin Cui, and Xiaoming Li. Preference-aware mask for session-based recommendation with bidirectional transformer. In *Proceedings of ICASSP*, pages 3412–3416, 2020.

[Zhou *et al.*, 2019] Fan Zhou, Ruiyang Yin, Kunpeng Zhang, Goce Trajcevski, Ting Zhong, and Jin Wu. Adversarial point-of-interest recommendation. In *WWW*, pages 3462–34618. ACM, 2019.