Variational Session-based Recommendation Using Normalizing Flows

Fan Zhou
University of Electronic Science and Technology of China, Chengdu, China
fan.zhou@uestc.edu.cn

Zijing Wen
University of Electronic Science and Technology of China, Chengdu, China
wenzijing93@gmail.com

Kunpeng Zhang
University of Maryland, College park
kpzhang@umd.edu

Goce Trajcevski
Iowa State University, Ames
gocet25@iastate.edu

Ting Zhong*
University of Electronic Science and Technology of China, Chengdu, China
zhongting@uestc.edu.cn

ABSTRACT

We present a novel generative Session-Based Recommendation (SBR) framework, called VAriational SEssion-based Recommendation (VASER) – a non-linear probabilistic methodology allowing Bayesian inference for flexible parameter estimation of sequential recommendations. Instead of directly applying extended Variational AutoEncoders (VAE) to SBR, the proposed method introduces normalizing flows to estimate the probabilistic posterior, which is more effective than the agnostic presumed prior approximation used in existing deep generative recommendation approaches. VASER explores soft attention mechanism to upweight the important clicks in a session. We empirically demonstrate that the proposed model significantly outperforms several state-of-the-art baselines, including the recently-proposed RNN/VAE-based approaches on real-world datasets.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → *Neural networks.*

KEYWORDS

Session-based recommendation, variational autoencoders, normalizing flows

ACM Reference Format:
Fan Zhou, Zijing Wen, Kunpeng Zhang, Goce Trajcevski, and Ting Zhong. 2019. Variational Session-based Recommendation Using Normalizing Flows. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3308558.3313615

1 INTRODUCTION

Session-based recommendation (SBR) [18, 36] aims at predicting user's next action based on a series of recent actions. It is a kind of sequence learning/recommendation task where longer-term user historical activities are usually unavailable and the recommendations need to be made in accordance with an assumed short-term interests of the (anonymous) user.It has been widely used in various applications, such as news recommendation, e-commerce, video and classified advertisement recommendation [10, 36].

Recent advances in deep learning have spurred the use of recurrent neural networks (RNNs) based methods to model SBR [17, 18, 28, 33], achieving significant improvement on recommendation accuracy over traditional sequence-based models such as factorizing personalized Markov chains (FPMC) [16, 38] and feature-based matrix factorization (MF) [5, 51]. Specifically, GRU4Rec [18] – a first application of augmented gated recurrent units (GRUs) [8] – was developed to address SBR by encoding user's preference and learning it for next-click prediction. Subsequently, a few improvements to GRU4Rec have been proposed – e.g., incorporating attention mechanism [28]; employing hierarchical recurrent networks [28]; augmenting data with additional features associated with items [19]; prioritizing short attention/memory [33]; and introducing more sophisticated ranking algorithms [17].

Complementary to these works, recent efforts on incorporating stochastic latent variables trained by deep generative models (e.g., variational autoencoders (VAE) [25, 40]) have enabled significant progress in many natural language processing tasks (e.g., dialogue generation and machine translation [2, 3, 11, 14, 21]). Similarly, Various generative models including VAEs have demonstrated potential for learning effective non-linear representations of user-item interactions [7, 23, 27, 30, 32] in collaborative filtering settings. For the most part, they either model the generation process of auxiliary information (e.g., content and ratings) [7, 27, 30] or build a probabilistic latent-variable framework that shares statistical strength among users and items [7, 23, 32].

Despite the improvements over conventional item recommendation, the aforementioned Bayesian models cannot be directly generalized to SBR due to the following reasons:
(1) **Data availability**: the lack of users' profile information and long-term interaction data makes these models not work well in SBR settings.
(2) **Bypassing issue**: autoregressive models (e.g., LSTM [20] and GRU [8]) combined with the soft attention mechanisms [1] have capabilities of reconstructing an encoded session on their own. This

*Corresponding author: zhongting@uestc.edu.cn

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313615

may weaken the effects of the incorporated latent factors [3], which can potentially reduce the performance of the VAE-based models. (3) **Biased inference**: VAE based models usually assume a predefined prior for latent factors [24], e.g., multivariate Gaussian which might result in the inferred approximate posterior greatly deviating from the true distribution.

We extend VAEs to model implicit feedbacks of user-item interactions in a session, and present the *VAriational SEssion-based Recommendation (VASER)*. While retaining the *Bayesian inference* of VAEs and enabling exploration of *non-linear* probabilistic latent-variable models, the VASER model: (1) effectively addresses the problem of unimodal and simple parametric problems of existing SBR methods; and (2) largely ameliorates the bias inference problem of existing VAE based recommendation methods. Specifically, we make the following contributions:

- VASER augments the RNNs based SBR models with stochastic latent variables, enabling stable and effective approximate inference of a high-level "objective" of an entire session from the observed clicks.
- We exploit the flows to approximate the real posterior of stochastic latent factors, which can largely alleviate the inference bias in existing VAE based recommendation models and improve the next click prediction accuracy.
- We demonstrate that VASER achieves improvements in SBR performance over the baselines on several real-world datasets.

2 PRELIMINARIES

We now introduce the basic notation used in the rest of the paper and formally define the problem. Recalling that SBR aims at predicting which item an anonymous user would like to click next given his current sequence of interaction data, we also provide an overview of the RNN based approaches.

2.1 Problem Definition

Formally, we have a set of sessions $\mathbf{S}$, and each session $\mathbf{s}_i \in \mathbf{S}$, consists of a sequence of user actions (e.g., click, purchase, etc.). $\mathbf{s}_i = [x_{i,1}, \cdots, x_{i,N}]$ (interchangeably denoted by $x_{i,(1:N)}$), where $x_{i,j} \in \mathbb{R}$ ($1 \leq j \leq N$, N is the length of the session.) is an interaction with item j in the session, assumed to be mapped to the domain $\mathbb{R}$. When no ambiguity arises, we will omit the index of the session – thus, given the prefix $\mathbf{s}' = [x_1, \cdots, x_{N-1}]$ of a session $\mathbf{s}$, the SBR model predicts the label(s) of the next action x_N by learning a classification distribution $\mathbf{y} = [\hat{y}_1, \cdots, \hat{y}_M]$ over M items, where $\hat{y}_j$ refers to a (predicted) probability or a ranking score for the N^{th} interaction with item j. Note that in practice, usually more than one recommendation is made, which is often referred to *top-k* session-based recommendation [28, 52].

2.2 SBR with RNNs

Existing RNN based models, with or without attention, train the sessions in a seq2seq manner. The main differences among them are how to decode the latent factors (or more precisely the last hidden state of the RNN) and how to embed the items. In "vanilla" GRU based models [17–19], decoding reconstructs the session and embedding is a separate layer of training. In attentive RNN-based models [30, 33], however, the encoder acts as an embedding layer

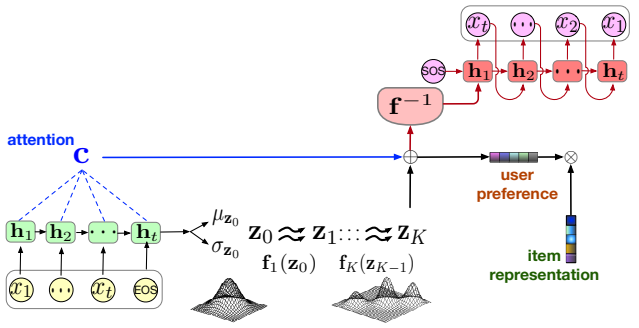


Figure 1: Overview of VASER. Recommendation is made based on the posterior $q(\mathbf{z}_K)$ of the last hidden state and the attention vector. Items are represented by embedding vectors.

– i.e., they train item embedding along with calculating loss of training sessions. Therefore, this type of *supervised* training may indeed "memorize" the sequential information of a given session, which may be "conducted" in the testing phase as the items in testing sessions would look-up the embedding matrix. As observed in the experiments in [30], this dynamic embedding method may significantly improve the performance.

An important observation is that all these works train the model in an *explicit autoregressive* fashion, i.e., they split the sessions (both training and testing) into a set of sub-sessions. Thus, a session $\mathbf{s} = [x_1, \cdots, x_N]$ would be divided into $N-1$ sub-sessions: $\mathbf{s}^1 = [x_1, \cdots, x_{N-1}]$, $\mathbf{s}^2 = [x_1, \cdots, x_{N-2}]$, $\cdots$, $\mathbf{s}^{N-1} = [x_1, x_2]$ and the original session $\mathbf{s}$ – all of which would be fed into the models for training or testing. Although not explicitly specified, this kind of autoregressive training improves the overall performance of the models, since a longer session actually contains (and thus "memorizes") the sub-sessions. We note that this autoregressive training trick has also been explored in recent CNN based SBR models [45, 52].

3 MAIN METHODOLOGIES

We propose the VASER model, as illustrated in Figure 1, consisting of two main components, namely GRU module and attention module. The GRU module captures sequential preferences, and the hidden state can exploit the non-linear preferences. The attention module is used to enhance the GRU network by dynamically selecting and linearly combining different parts of the input sequence. VASER employs a deterministic attention mechanism. VASER incorporates the normalizing flows for flexible posterior approximation. In the sequel, we present the general framework of VASER with theoretical background and training procedure.

3.1 Session Generative Model

We consider a click session generative process as follows. For each session $\mathbf{s} = [x_1, \cdots, x_N]$, the model samples $d-$dimensional latent representation from an appropriate *prior* distribution $p(\mathbf{z})$. The latent factor $\mathbf{z}$ is then transformed via a non-linear function $f_\theta(\mathbf{z})$ – a suitable likelihood function parameterized by θ – to produce a probability distribution $\pi(\mathbf{z})$ (e.g., a *multinomial* distribution) over

M candidate items, from which a session $\mathbf{s}$ is assumed to have been drawn ($\mathbf{z} \sim p(\mathbf{z}); \pi(\mathbf{z}) \propto \exp\{f_\theta(\mathbf{z})\}$):

$$\mathbf{s} \sim f_\theta(\mathbf{z}) = p_\theta(\mathbf{s}|\mathbf{z}) = \prod_{t=2}^{N} p_\theta(x_t|x_{1:t-1}, \mathbf{z}) \quad (1)$$

where $x_{1:t-1}$ indicates the prefix click sequence preceding current click x_t, and $f_\theta(\mathbf{z})$ is a deep neural networks such as a multilayer perceptron (MLP). Thus, the session generation involves making a sequence of discrete decisions, each of which samples an item from a multinomial distribution with a softmax function, to produce a probability vector $\pi(\mathbf{z})$ over the entire item set. The multinomial distribution has been demonstrated to model click data well (cf. [26, 32], although these work were originally designed for CF based recommendation).

This generative process is similar to the sentence generation in [21] and trajectory generation in [53], except that we do not take side-information (e.g., item category, click time, etc.) into account. However, it is straightforward to add additional latent factors to capture various item features, if available, for disentangling the representation.

3.2 Variational Session Inference

In general, the marginal log-likelihood of a session s $\log p_\theta(\mathbf{s}) = \log \int_{\mathbf{z}} p_\theta(\mathbf{s}|\mathbf{z})p(\mathbf{z})\mathrm{d}\mathbf{z}$ is intractable to compute or differentiate directly for flexible generative models, especially for high-dimensional latent variables. Instead, one usually resorts to variational inference by defining a simple parametric distribution over the latent variables (e.g, a factorized Gaussian) $q_\phi(\mathbf{z}|\mathbf{s})$, and maximizing the evidence lower bound (ELBO) on the marginal log-likelihood of each observation:

$$\log p_\theta(\mathbf{s}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{s})} \log \left[\frac{p_\theta(\mathbf{s}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{s})}\right] + \mathbb{KL}\left[q_\phi(\mathbf{z}|\mathbf{s})||p_\theta(\mathbf{z}|\mathbf{s})\right] \quad (2)$$

$$\geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{s})}\left[\log p_\theta(\mathbf{s}, \mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{s})\right] \triangleq \mathcal{L}(\mathbf{s}; \theta, \phi) \quad (3)$$

There are numerous ways to optimize the ELBO, among which VAEs [25] use a parametric inference network and reparameterization of $q_\phi(\mathbf{z}|\mathbf{s})$ to alternatively maximize following reformulation:

$$\mathcal{L}_{\text{VAE}}(\mathbf{s}; \theta, \phi) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{s})}[\log p_\theta(\mathbf{s}) + \log p_\theta(\mathbf{z}|\mathbf{s}) - \log q_\phi(\mathbf{z}|\mathbf{s})]$$

$$= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{s})}[\log p_\theta(\mathbf{s})] - \mathbb{KL}\left[q_\phi(\mathbf{z}|\mathbf{s})||p_\theta(\mathbf{z}|\mathbf{s})\right] \quad (4)$$

Since the first term is a constant, then the objective of maximizing ELOB $\mathcal{L}_{\text{VAE}}(\mathbf{s}; \theta, \phi))$ of $\log p_\theta(\mathbf{s})$ becomes to minimize the Kullback-Leibler (KL) divergence between $q_\phi(\mathbf{z}|\mathbf{s})$ and the true distribution $p_\theta(\mathbf{z}|\mathbf{s})$ (which is always ≥ 0). For brevity, we will sometimes omit the parameters ϕ and θ in subsequent formulae.

3.3 Inference with Normalizing Flows

It is desirable to reduce the (non-negligible) inference gaps, and various improved posterior approximations have been effective in improving variational inference. Although none of the existing methods is able to completely close the gap between approximate posterior and true posterior [6], employing richer posterior/prior distributions can effectively reduce it. The approximation gap, caused by the encoding cost $\mathbb{KL}\left[q_\phi(\mathbf{z}|\mathbf{s})||p_\theta(\mathbf{z}|\mathbf{s})\right]$, is largely due to the improper assumption of the probabilistic distribution [9, 24].

We leverage the flow method [39] to construct more accurate posterior approximation of the session distributions, rather than simple Gaussian assumption in existing works [4, 49]. Normalizing Flows (NF) [39] is a powerful framework for building flexible posterior distributions through an iterative procedure. The main idea is to transform a simple distribution into a complex one through a series of invertible mappings which, in theory, can approximate any complex distribution. Given a variable $\mathbf{z}_0$ with known probability distribution $p_0(\mathbf{z}_0)$ (e.g., Gaussian here) and a chain of invertible transformations $\mathbf{f} = [\mathbf{f}_1, \cdots, \mathbf{f}_K]$, then $\mathbf{z}_k$ can be calculated by composing the transformations from $\mathbf{f}$ as:

$$\mathbf{z}_K = \mathbf{f}_K(\mathbf{z}_{K-1}) = \mathbf{f}_K(\mathbf{f}_{K-1}(\mathbf{z}_{K-2})) = \mathbf{f}_K(\mathbf{f}_{K-1}(\cdots \mathbf{f}_1(\mathbf{z}_0))) \quad (5)$$

Given that each $\mathbf{f}_k \in \mathbf{f}$ is invertible (i.e., $\mathbf{z}_{k-1} = \mathbf{f}_k^{-1}(\mathbf{z}_k)$), and according to the definition of probability $\int p_k(\mathbf{z}_k)d\mathbf{z}_k = \int p_{k-1}(\mathbf{z}_{k-1})d\mathbf{z}_{k-1} = 1$, for a collection of variables $\mathbf{z} = [\mathbf{z}_0, \ldots, \mathbf{z}_K]$, we can obtain the distributions $p_K(\mathbf{z}_K)$ more flexibly:

$$p_K(\mathbf{z}_K) = p_0(\mathbf{z}_0)\left|\det \frac{d\mathbf{z}_K}{d\mathbf{z}_0}\right|^{-1} \quad (6)$$

where $\det \frac{d\mathbf{f}}{d\mathbf{z}}$ is the Jacobian determinant of $\mathbf{f}$.

The path traversed by the random variables $\mathbf{z}_k = \mathbf{f}_k(\mathbf{z}_{k-1})$ with initial distribution $p_0(\mathbf{z}_0)$ is called the *flow*, and the whole path formed by the successive distributions $p_K(\mathbf{z}_K)$ refers to a *normalizing flow*. To ensure Eq.(6) is tractable, it should satisfy that (a) the transformation $\mathbf{f}_k$ must be easy to invert, and (2) the determinant of its Jacobian is easy to compute [39]. The two constraints allow the transformation to be made deeper by composing multiple instances of it, and the result will still be a valid normalizing flow. Now the log-likelihood of approximate posterior $q_K(\mathbf{z}_K|\mathbf{s})$ can be computed iteratively by using the log on both sides of Eq.(6)

$$\log q_K(\mathbf{z}_K|\mathbf{s}) = \log q_0(\mathbf{z}_0|\mathbf{s}) - \sum_{k=1}^{K} \log \det \left|\frac{d\mathbf{z}_k}{d\mathbf{z}_{k-1}}\right| \quad (7)$$

where the base distribution $\mathbf{z}_0 \sim q_\phi(\mathbf{z}_0|\mathbf{s})$ is a Gaussian in our implementation.

One of the flow transformations is the *planar flow* introduced in [39], given by:

$$\mathbf{f}(\mathbf{z}) = \mathbf{z} + \mathbf{u}\sigma(\mathbf{w}^\intercal \mathbf{z} + b) \quad (8)$$

where $\mathbf{u}, \mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ are parameters, and σ is a suitable smooth non-linear activation function (e.g., tanh). According to the *Matrix determinant lemma*, the Jacobian of this transformation is:

$$\left|\det \frac{\partial \mathbf{f}}{\partial \mathbf{z}}\right| = \left|1 + \mathbf{u}^\intercal \sigma'(\mathbf{w}^\intercal \mathbf{z} + b)\mathbf{w}\right|,$$

where σ' is the derivative activation and can be computed in $O(d)$ time – d is the dimension of $\mathbf{z}$.

In this paper, we use the planar flow as the invertible transformation for its simplicity and efficiency. We parameterize the approximation posterior $q_\phi(\mathbf{z}|\mathbf{s})$ with a flow, i.e., $q_\phi(\mathbf{z}|\mathbf{s}) := q(\mathbf{z}_K)$, the ELBO of Eq.(2) can be modified as

$$\mathcal{L}(\mathbf{s}; \theta, \phi) = \mathbb{E}_{q(\mathbf{z}_0)}[\log p_\theta(\mathbf{s}|\mathbf{z}_K)] - \mathbb{E}_{q(\mathbf{z}_0)}[\log q(\mathbf{z}_0)] + \beta\mathbb{E}_{q(\mathbf{z}_0)}[\log p_\theta(\mathbf{z}_K)]$$

$$+ \beta\mathbb{E}_{q(\mathbf{z}_0)}\left[\sum_{k=1}^{K} \log \det \left|\frac{d\mathbf{z}_k}{d\mathbf{z}_{k-1}}\right|^{-1}\right] \quad (9)$$

where the first term is trained to reconstruct the sessions; the second term is a constant; and the last two terms are the flows. The coefficient β is a regularizer of the flows, which is very similar to the annealing factor for regularizing KL-divergence [3].

4 EXPERIMENTAL OBSERVATIONS

We now describe the experimental settings and report the empirical evaluation results.

Table 1: Statistics of the datasets.

Datasets	YOOCHOOSE 1/64	YOOCHOOSE 1/4
#clicks	557,248	832,6407
#train sessions	355,385	621,6184
#test sessions	52,956	56,616
#items	17,626	30,903
avg. session length	6.27	5.83

4.1 Datasets

For fair comparison, we evaluate different methods on a real-world transaction datasets YOOCHOOSE[1], which has been widely used for evaluating SBR approaches. Following previous works [28, 33, 44], we preprocess the primary data as follows: (1) We filter out sessions of length 1 and items that appear less than 5 times for the two datasets; (2) We respectively use the sessions of subsequent day for testing, and then filter out clicks from the test set where the clicked items did not appear in the training set; and (3) We sort the training sequences by time and train all models on more recent fractions (i.e., 1/64 and 1/4) of training sessions. Table 1 shows the statistics of the datasets.

4.2 Baselines

To demonstrate the effectiveness of our model, we conduct extensive comparisons to the following state-of-the-art methods:

- **Item-KNN** [42]: It is an item-to-item model that recommends items that are similar to previously visited items based on cosine similarity.
- **GRU4Rec**[2] [18]: It is an RNN-based deep learning model for session-based recommendation. It employs GRU units to capture sequential patterns and utilizes session-parallel minibatching trick and ranking-based loss functions during the training.
- **NARM**[3] [28]: It is an RNN-based model employing (deterministic) attention mechanism to capture main purpose from the hidden states and combines it with the sequential behavior as the final representation to generate recommendations.
- **STAMP** [33]: It is a priority model which captures users' general interests from the long-term memory of a session context, and current interests from the short-term memory of recent clicks.
- **ReLaVaR** [4]: It is a Bayesian version of GRU4Rec which treats the network recurrent units as stochastic latent variables with some prior distributions and infers the corresponding posteriors for prediction and recommendation generation. This is an item-level variational inference based SBR method which uses independent Gaussian as the prior for items.
- **VRM** [49]: It is a recent proposed method directly applying VAE on session-based recommendation. Unlike ReLaVaR, an item-level variational method, VRM models the stochastic inference on the session-level.

[1]http://2015.recsyschallenge.com/challenge.html
[2]https://github.com/hidasib/GRU4Rec
[3]https://github.com/lijingsdu/sessionRec_NARM

4.3 Metrics

Following previous works [16, 17, 28, 33], the primary evaluation metric is Recall@20 – i.e., the proportion of cases having the desired item falling into the top-20 predicted items in all test cases. Note that the Recall score is equal to the Hit-Precision score used in [33]. The second metric is MRR@20 (Mean Reciprocal Rank) – i.e., the average of reciprocal ranks of the desired items. The reciprocal rank is set to zero if the rank is lower than top-20. MRR takes into account the rank of the item, which is important when the order of recommendations matters. Note that the higher the Recall@20 and MRR@20, the better the performance.

4.4 Settings

For all methods, the embedding size of items is set to 50. The number of hidden units in GRU layer is set to 100. All models are trained with Adam and the mini-batch size is fixed at 512. Following [28, 33], we truncated BPTT using a fixed window of 30 time-steps for the two YOOCHOOSE datasets. Also following [28, 33], 10% of the training data are used as the validation set. For the VASER model, parameters d, K and β are respectively 100, 16 and 0.2, if not specified.

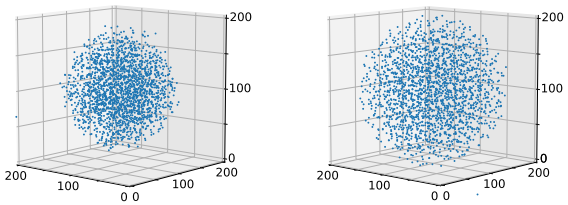
4.5 Overall Performance

Table 2: Performance comparison among all session-based recommendation methods over two datasets.

	YOOCHOOSE1/64		YOOCHOOSE1/4	
	Recall@20(%)	MRR@20(%)	Recall@20(%)	MRR@20(%)
item-KNN	53.12	22.13	52.43	21.75
GRU4REC	62.40	25.36	59.58	22.62
NARM	70.13	29.38	69.75	29.30
STAMP	70.21	29.22	70.45	29.47
ReLaVaR	64.32	25.26	60.53	22.76
CRM	69.32	28.75	68.22	28.35
VASER	71.85	30.05	70.74	29.75

4.5.1 Comparison against SBR baselines (Q1). : Table 2 shows the results of comparison to the existing state-of-the-art SBR methods, from which we can clearly observe that the proposed model perform the best on two metrics throughout two datasets.

Overall, the RNN based methods, including ours, consistently outperform the traditional baselines, which demonstrates that autoregressive models are good at learning sequential user click behaviors.



(a) NARM. (b) VASER.

Figure 2: Visualization of the encoding space obtained via NARM and VASER on YOOCHOOSE 1/64. For better viewing, we randomly select 2,048 test sessions and plot the encoding space using t-SNE.

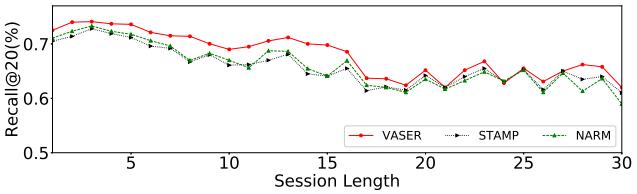


Figure 3: Impact of session length (YOOCHOOSE 1/64).

Nevertheless, RNN models alone cannot deal with complicate user-click sessions which usually have unintended clicks and/or contain one or more browse themes. This problem can be largely overcome by incorporating the attention mechanism in recent methods like NARM and STAMP. The most recent work GRU4Rec++ does not exhibit expected results on the two datasets, regardless that it can improve their original method (GRU4Rec) with the sampling trick. This result also proves one of our motivations that autoregressive models are constrained with their capability of modeling sparse and high-dimensional data.

By modeling session generation in a probabilistic generative latent variable framework, our model outperform the best baseline (either NARM or STAMP) by a significant margin. Note that in our reimplementation, the two baselines (NARM or STAMP) exhibit higher scores than their original reporting on two YOOCHOOSE datasets.

The benefit of VASER can be visualized in Figure 2, where the encoding space of NARM and VASER is plotted with t-SNE. Recall that both methods predict the last item x_N based on the learned representation of prefix session $\mathbf{s}' = [x_1, \cdots, x_{N-1}]$. The main difference is that the encoding space of NARM is the concatenation of deterministic attention $\mathbf{c}$ and the last hidden state $\mathbf{h}_{N-1}$ of GRU, while the encoding space of VASER is the combination of the posterior of hidden state $q(\mathbf{z}_K)$ and the posterior of attention $q(\mathbf{c}_k)$. Apparently, VASER explores more space for encoding the sessions and exhibits more scattered distribution. The benefits of such encoding can be understood intuitively, i.e., the more inseparable the sessions, the more difficult for the models to discriminate the spatial adjacent ones, which, consequently, are more prone to making wrong predictions.

We also investigated the impact of session length on the recommendation performance. Intuitively, the longer the sessions, the worse the prediction performance on average. The results on YOOCHOOSE 1/64 are shown in Figure 3 (results on YOOCHOOSE 1/4 are consistent, but omitted due to the lack of space), whereby we compared to the two best baselines. Our model slightly improve the recommendation performance over the baselines. However, we argue that due to the vanishing gradient problem of autoregressive model, it is hard for RNNs-based methods to further improve the performance on modeling extremely long-term dependencies.

4.5.2 Effect of Components in VASER (Q2). By comparing to the methods directly applying VAE on SBR, both item-level and session-level, we can clearly see the importance of the flow based posterior approximation used in VASER.

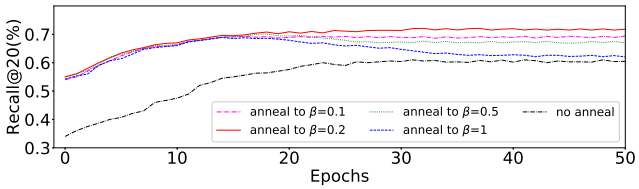
Another important observation is that directly applying VAE on modeling items or sessions is not competitive. ReLaVaR, operating stochastic inference on item-level, is less effective than VRM, which models sessions in a variational seq2seq manner. Note that we omit comparison with CVRM, a variant of VRM, which takes the category information into account, due to the *extremely* sparse category labels on the YOOCHOOSE dataset. In fact, the category information plays a less important role in improving the recommendation performance according to the results in [49]. Although allowing Bayesian inference, the two models may incur larger inference gaps and underfitting problem due to the amortized inference alone used for posterior approximation [9]. This is in accordance with the findings in modeling language with vanilla VAEs [3, 21], i.e., the autoregressive models are powerful enough to decode the entire sequence, resulting in uselessness of stochastic latent factors. More importantly, these methods approximate an improper assumed distribution $q_\phi(\mathbf{z}|\mathbf{s})$, e.g., the choice of diagonal-covariance Gaussian in [3, 21], and thus are subjected to heavy bias inference problem, as explained in Sec. 3.2. In contrast, our VASER model can largely alleviate this problem benefiting from the normalizing flows with flexible posterior approximation.

4.6 Impact of Parameters (Q3)

There are two important factors affecting the performance of the VASER model, i.e., the coefficient β regularizing the flows and the determinant of Jacobian matrix, and K, the number of invertible transformations.

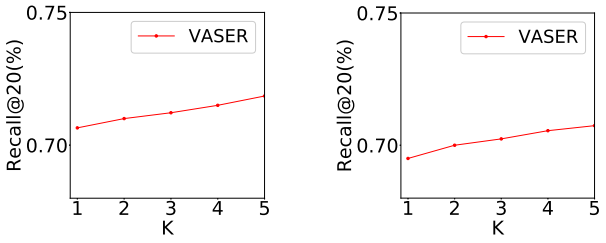
Figure 4 shows the impact of β on VASER, where β is gradually annealed to the value of 0.1, 0.2, 0.5 and 1. We observe in our experiments that the flow terms are usually ordered larger than the reconstruction term. Without annealing or annealing to a larger value, the performance of VASER model are not appealing, and even experience overfitting problem. On the contrary, if the value of β is too small (e.g., below 0.2, the flows does not take effect as the decoder RNN will make the model converge, when the model rely less on the latent factors. As a consequence, there is a significant performance decline. As we explained earlier, this is caused by the overpower performance of RNN decoder. In addition to cost annealing, another possible way of alleviating this problem is to replace RNN with the dilated CNN suggested by [21].

Figure 5 investigates the impact of K on two datasets. The planar flows used in VASER modifies the initial density by applying a series



(a) YOOCHOOSE 1/64.

Figure 4: The impact of β.



(a) YOOCHOOSE 1/64. (b) YOOCHOOSE 1/4.

Figure 5: The impact of K.

of contractions and expansions in the original space. Although, in theory, more transformations could approximate more complicated distribution, a smaller value is enough for the model. Since the RHS of Eq.(8) can be interpreted as a *single-neuron* MLP, it may result in the information going through a single bottleneck. As the volume of the space grows exponentially with the number of dimensions d, it requires many coupling layers to transform a simple base distribution into a complex one [24]. This is demonstrated by the results on YOOCHOOSE 1/64 dataset in Figure 5(a), where the model require more transformations to obtain higher performance.

5 RELATED WORK

5.1 Sequential Recommendation

SBR is essentially a sequence learning problem including typical scenarios such as click/purchase recommendation in e-commerce, music/video recommendation, news items etc. Since only short-term interaction data are available and there is lack of user profile, CF based latent factor models fail to work in these scenarios. Non-parametric methods, such as k-Nearest Neighbor (KNN) and context tree can be used to estimate the user/item similarity for recommending the most similar items to the ones that have been visited/clicked by a user [12, 15, 22, 35, 42]. Naturally, other sophisticated sequence learning approaches can also be adapted to solve the session-based recommendation problem, which incurs MC based models [48] and hybrid CF models like FPMF [16, 38], etc.

RNN models have been successfully applied in many sequence learning tasks, such as machine translation [1], human mobility learning [13], and session-based recommendation [17, 18, 28, 31, 33, 37]. GRU4Rec [18] is a representative RNN based method for SBR, which embeds the clicks into the final hidden state of GRU to represent the current preference. This method has achieved significant improvement against previous sequence learning approaches like FPMC and item similarity based KNN. Several works have been proposed to improve GRU4Rec with various models. For example, NARM and EDRec [28, 34] employ soft attention mechanism [1] to capture the user's main purpose in the current session, which is combined with the last hidden state of GRU to compute the recommendation scores for each candidate item. Hidasi et al. [17] improved their GRU4Rec model by introducing tailored ranking loss functions.

5.2 Deep Generative Recommendation

Although there exist many deep recommendation models as mentioned above, relatively few works in the literature focus on applying generative models in the recommendation systems. Previous autoencoder based models [29, 41, 43, 46, 47, 50] show promising performance but are restricted to learning representation of items, and thus are difficult for Bayesian inference due to lack of Bayesian nature or high computational cost. Several recent works extend the ideas of applying VAEs to CF-based recommendation but mainly focus on combining various auxiliary features [7, 23]. The most related work are ReLaVaR [4] and VRM/CVRM [49], both of which apply VAE on the SBR tasks. ReLaVaR is an item-level stochastic inference method while VRM/CVRM are modeling session in a stochastic seq2seq manner. As sequential VAE models, they can be considered as directly applying VAEs in the SBR scenario.

Compared to existing sequential recommendation approaches, we model the problem within a probabilistic recommendation setting which allows our model for Bayesian inference. In addition, we derive novel flow-based model tailored for SBR task with the flexible posterior approximation, rather than presumed Gaussian distribution in previous work.

6 CONCLUSIONS AND FUTURE WORK

We presented VASER, a flow based generative framework for learning sequential click patterns. A distinct feature of the proposed model implementing VASER is that it enables learning non-linear interactions between user-clicks while allowing Bayesian inference. As demonstrated by the experiments, VASER achieves significant improvements for the session-based recommendation problem in comparison to existing methods. One of the most important implications of the results from this work is that instead of using amortized inference as in existing collaborative/sequential variational recommendation methods, flow based techniques could effectively improve the density approximation and deserves more attention in the recommendation community. In our future work, we are planning to focus on augmenting VASER to consider auxiliary information – e.g., coupling sequential information with other related contexts (category, price and click time), and on tackling the overall efficiency (e.g., by incorporating CNNs).

ACKNOWLEDGEMENTS

This work was supported by National Natural Science Foundation of China (Grants No.61602097 and No.61472064), NSF Grants III 1213038 and CNS 1646107, and an ONR grant N00014-14-10215.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR*.

[2] Hareesh Bahuleyan, Lili Mou, Olga Vechtomova, and Pascal Poupart. 2018. Variational Attention for Sequence-to-Sequence Models. In *COLING*.

[3] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Józefowicz, and Samy Bengio. 2016. Generating Sentences from a Continuous Space. In *CoNLL*.

[4] Sotirios P Chatzis, Panayiotis Christodoulou, and Andreas S Andreou. 2017. Recurrent Latent Variable Networks for Session-Based Recommendation. In *DLRS@RecSys*.

[5] Tianqi Chen, Weinan Zhang, Qiuxia Lu, Kailong Chen, Zhao Zheng, and Yong Yu. 2012. SVDFeature: a toolkit for feature-based collaborative filtering. *JMLR* (2012).

[6] Xi Chen, Diederik P Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2017. Variational Lossy Autoencoder. In *ICLR*.

[7] Yifan Chen and Maarten de Rijke. 2018. A Collective Variational Autoencoder for Top-N Recommendation with Side Information. In *DLRS@RecSys*.

[8] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv:1412.3555* (2014).

[9] Chris Cremer, Xuechen Li, and David K Duvenaud. 2018. Inference Suboptimality in Variational Autoencoders. In *ICML*.

[10] Gabriel de Souza Pereira Moreira, Felipe Ferreira, and Adilson Marques da Cunha. 2018. News Session-Based Recommendations using Deep Neural Networks. In *DLRS@RecSys*.

[11] Yuntian Deng, Yoon Kim, Justin Chiu, Demi Guo, and Alexander M Rush. 2018. Latent Alignment and Variational Attention. In *NIPS*.

[12] Mukund Deshpande and George Karypis. 2004. Item-based top-N recommendation algorithms. *ACM TOIS* (2004).

[13] Qiang Gao, Fan Zhou, Kunpeng Zhang, Goce Trajcevski, Xucheng Luo, and Fengli Zhang. 2017. Identifying Human Mobility via Trajectory Embeddings. *IJCAI* (2017).

[14] Anirudh Goyal, Alessandro Sordoni, Marc-Alexandre Côté, Nan Rosemary Ke, and Yoshua Bengio. 2017. Z-Forcing: Training Stochastic Recurrent Networks. In *NIPS*.

[15] Huifeng Guo, Ruiming Tang, Yunming Ye, Feng Liu, and Yuzhou Zhang. 2018. An Adjustable Heat Conduction based KNN Approach for Session-based Recommendation. *arXiv* (2018).

[16] Ruining He and Julian McAuley. 2016. Fusing Similarity Models with Markov Chains for Sparse Sequential Recommendation. In *ICDM*.

[17] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent Neural Networks with Top-k Gains for Session-based Recommendations. In *CIKM*.

[18] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *ICLR*.

[19] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. 2016. Parallel Recurrent Neural Network Architectures for Feature-rich Session-based Recommendations. In *RecSys*.

[20] Sepp Hochreiter and JÃijrgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* (1997).

[21] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward Controlled Generation of Text. In *ICML*.

[22] Dietmar Jannach and Malte Ludewig. 2017. When Recurrent Neural Networks meet the Neighborhood for Session-Based Recommendation. In *RecSys*.

[23] G Karamanolakis, K R Cherian, AR Narayan Proceedings of the 3rd, and 2018. 2018. Item Recommendation with Variational Autoencoders and Heterogeneous Priors. In *DLRS@RecSys*.

[24] Diederik P Kingma, Tim Salimans, Rafal Józefowicz, Xi Chen, Ilya Sutskever, and Max Welling. 2016. Improving Variational Autoencoders with Inverse Autoregressive Flow. In *NIPS*.

[25] Diederik P Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *ICLR*.

[26] Rahul G Krishnan, Dawen Liang, and Matthew D Hoffman. 2018. On the challenges of learning with inference networks on sparse, high-dimensional data. In *AISTATS*.

[27] Wonsung Lee, Kyungwoo Song, and Il-Chul Moon. 2017. Augmented Variational Autoencoders for Collaborative Filtering with Auxiliary Information. In *CIKM*.

[28] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural Attentive Session-based Recommendation. In *CIKM*.

[29] Sheng Li, Jaya Kawale, and Yun Fu. 2015. Deep Collaborative Filtering via Marginalized Denoising Auto-encoder. In *CIKM*.

[30] Xiaopeng Li and James She. 2017. Collaborative Variational Autoencoder for Recommender Systems. *KDD* (2017).

[31] Zhi Li, Hongke Zhao, Qi Liu, Zhenya Huang, Tao Mei, and Enhong Chen. 2018. Learning from History and Present: Next-item Recommendation via Discriminatively Exploiting User Behaviors. *KDD* (2018).

[32] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. *WWW* (2018).

[33] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: Short-Term Attention/Memory Priority Model for Session-based Recommendation. In *KDD*.

[34] Pablo Loyola, Chen Liu, and Yu Hirate. 2017. Modeling User Session and Intent with an Attention-based Encoder-Decoder Architecture. In *RecSys*.

[35] Fei Mi and Boi Faltings. 2018. Context Tree for Adaptive Session-based Recommendation. *arXiv* (2018).

[36] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-Aware Recommender Systems. *Comput. Surveys* (2018).

[37] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks. In *RecSys*.

[38] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized Markov chains for next-basket recommendation. In *WWW*.

[39] Danilo Jimenez Rezende and Shakir Mohamed. 2015. Variational Inference with Normalizing Flows. *ICML* (2015).

[40] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *ICML*.

[41] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey E Hinton. 2007. Restricted Boltzmann machines for collaborative filtering. In *ICML*.

[42] Badrul Munir Sarwar, George Karypis, Joseph A Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *WWW*.

[43] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. AutoRec: Autoencoders Meet Collaborative Filtering. In *WWW*.

[44] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved Recurrent Neural Networks for Session-based Recommendations. In *DLRS@RecSys*.

[45] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *WSDM*.

[46] Hao Wang, Xingjian Shi, and Dit-Yan Yeung. 2016. Collaborative Recurrent Autoencoder: Recommend while Learning to Fill in the Blanks. In *NIPS*.

[47] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative Deep Learning for Recommender Systems. In *KDD*.

[48] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning Hierarchical Representation Model for NextBasket Recommendation. In *SIGIR*.

[49] Zhitao Wang, Chengyao Chen, Ke Zhang, Yu Lei, and Wenjie Li. 2018. Variational Recurrent Model for Session-based Recommendation. In *CIKM*.

[50] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collaborative Denoising Auto-Encoders for Top-N Recommender Systems. In *WSDM*.

[51] Fajie Yuan, Guibing Guo, Joemon M Jose, Long Chen, Haitao Yu, and Weinan Zhang. 2016. LambdaFM: Learning Optimal Ranking with Factorization Machines Using Lambda Surrogates. In *CIKM*.

[52] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. 2018. A Simple but Hard-to-Beat Baseline for Session-based Recommendations. *arXiv* (2018).

[53] Fan Zhou, Qiang Gao, Goce Trajcevski, Kunpeng Zhang, Ting Zhong, and Fengli Zhang. 2018. Trajectory-User Linking via Variational AutoEncoder. In *IJCAI*.