Neurocomputing 391 (2020) 129–141

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Session-based recommendation via flow-based deep generative networks and Bayesian inference

Ting Zhong[a], Zijing Wen[a], Fan Zhou[a,*], Goce Trajcevski[b], Kunpeng Zhang[c]

[a] School of Information and Software engineering, University of Electronic Science and Technology of China, China
[b] Department of Electrical and Computer Engineering, Iowa State University, Ames IA, USA
[c] Department of Decision Operations & Information Technologies, University of Maryland, College Park MD, USA

ARTICLE INFO

Article history:
Received 20 March 2019
Revised 16 January 2020
Accepted 24 January 2020
Available online 28 January 2020

Communicated by Dr. Jiliang Tang

Keywords:
Session-based recommendation
Variational autoencoders
Normalizing flows
Amortized inference
Attention mechanism

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. We also combine the effectiveness of both stochastic and amortized variational inference to reduce the inference gaps and to alleviate the underfitting problem of variational recommendation. We propose two specific implementations of VASER, both of which explore soft attention mechanism to upweight the important clicks in a session and show that one of them, treating the attention vector as an auxiliary latent factor, can make the variational distribution more expressive, and thus improves the recommendation accuracy over the widely used deterministic attention approaches. Empirically, we show that the proposed models significantly outperform several state-of-the-art baselines, including the recently-proposed RNN/VAE-based approaches, on several real-world datasets.

1. Introduction

Session-based recommendation (SBR) [23,47] aims at predicting user's next action based on recent series of 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 the assumed short-term interests of the (anonymous) user.

Recent advances in deep learning have spurred the use of recurrent neural networks (RNNs) based methods to model SBR [22,23,36,41], achieving significant improvement on recommendation accuracy over traditional sequence-based models such as factorizing personalized Markov chains (FPMC) [21,50] and feature-based matrix factorization (MF) [6,73]. Specifically, GRU4Rec [23] – a first application of augmented gated recurrent units (GRUs) [9] – 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 [36]; employing hierarchical recurrent networks [36]; augmenting data with additional features associated with items [24]; prioritizing short attention/memory [41]; and introducing more sophisticated ranking algorithms [22].

The existing RNNs-like SBR methods often predict the next click in a session based on the hidden state learned so far. They capture the information entropy in the observed session by conditional next-click (output) distributions on previous clicks for every timestep – typically a simple parametric form being chosen, i.e. unimodal or mixtures of unimodal. Such an inherent nature of RNNs [17] may be insufficient for SBR due to highly structured natural sequences in user-click sessions, where different output variables might interplay within a timestep, and complex dependencies exist between variables across timesteps. In addition, estimations at click level only consider immediate "short-term reward", and ignore the global browsing/purchasing consistency, even when combined with powerful attention mechanisms [1]. Furthermore, as the session grows, they could deviate from the original intents, e.g., existing models predict next clicks well for short sequences, but often fail for long sessions [36,41].

Complementary to these works, many augmented RNNs based methods have been developed by exploring multimodal output distributions and uncertainty estimation. For example, recent

* Corresponding author.
E-mail addresses: zhongting@uestc.edu.cn (T. Zhong), fan.zhou@uestc.edu.cn (F. Zhou), gocet25@iastate.edu (G. Trajcevski), kpzhang@umd.edu (K. Zhang).

https://doi.org/10.1016/j.neucom.2020.01.096
0925-2312/

efforts on incorporating stochastic latent variables trained by deep generative models (e.g.,variational autoencoders (VAE) [33,52]) have enabled significant progress in many natural language processing tasks (e.g., dialogue generation and machine translation [2,4,12,17,27]); Various generative models including VAEs have demonstrated potential for learning effective non-linear representations of user-item interactions [8,29,35,38,40] in the collaborative filtering settings. They either model the generation process of auxiliary information (e.g., content and ratings) [8,35,38] or build a probabilistic latent-variable framework that shares statistical strength among users and items [8,29,40].

Despite the improvements over conventional item recommendation, the aforementioned models (e.g., collaborative VAE) 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 [25] and GRU [9]) combined with the soft attention mechanisms [1] have capabilities of reconstructing an encoded session on their own. This (particularly deterministic attention) may weaken the effects of the incorporated latent factors [4], 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 [32], e.g., multivariate Gaussian which, as we will show, (i) is too restrictive for models to learn the true distribution; (ii) 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 trained by both *stochastic* and *amortized* variational inference, enabling stable and effective approximate inference of a high-level "objective" of an entire session from the observed clicks. By modeling and quantifying the stochastic latent variables in sessions, VASER is expected to discover and disentangle causal factors to interpret the user-click data.
- To encode more useful information into the latent variables, we introduce an auxiliary factor that leverages the variational attention on user clicks. Unlike the deterministic attention used in existing works, the proposed novel attention mechanism can accurately model click sessions, without overpowering the latent representation.
- We exploit the normalizing flows [51] 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 on several real-world datasets. We also show that our model, slightly modified, can outperform state-of-the-art collaborative recommendation methods on conventional user-item interaction datasets.

The rest of the paper is arranged as follows. We define the problem and introduce basic background in Section 2. The details of our models are presented in Section 3. Experimental results demonstrating the superiority of our model are discussed in Section 4, followed by reviewing relevant works in Section 5. We conclude this work and point out the future directions in Section 6.

Table 1
Notations.

Symbol	Description
$\mathbf{s}$	a user session.
x_i	an item.
y_j and $\hat{y}_j$	true and predicted score for item x_j.
$\boldsymbol{N}$	the length of a session.
$\boldsymbol{M}$	the number of all items.
$\mathbf{z}$ and $p(\mathbf{z})$	latent factor and its prior.
d	dimension of $\mathbf{z}$.
$\pi(\mathbf{z})$	probability distribution over items.
$p_\theta(\mathbf{s}\|\mathbf{z})$ or $p(\mathbf{s}\|\mathbf{z})$	generative model parameterized by θ.
$q_\phi(\mathbf{z}\|\mathbf{s})$ or $q(\mathbf{z}\|\mathbf{s})$	inference model parameterized by ϕ.
$\mathcal{L}(\mathbf{s};\theta,\phi)$	Evidence lower bound (ELBO).
$\mathbf{h}_t$	hidden state of the t-th step.
$\mathbf{f}_k$	invertible transformation function.
$\boldsymbol{K}$	the number of transformations
$\mathbf{c}$ or $\mathbf{c}_i$	attention vector.

2. Preliminaries

We now formalize the SBR problem and describe limitations of the recent RNN based methods.

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 $x_{i,j} \in \mathbb{R}$ ($1 \le j \le N$, 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.

The notations used in this work are detailed in Table 1, with a note that in practice, usually more than one recommendation is made, which is often referred to *top-k* session-based recommendation [36,74].

SBR with RNNs – why do they work? 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 [22–24], decoding reconstructs the session and embedding is a separate layer of training. In attentive RNN-based models [38,41], however, the encoder acts as an embedding layer – 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 [38], 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 [61,74].

In Table 2, we list the main SBR approaches. In particular, we summarize the methods used in modeling the sessions and their

Table 2
Summary of Session-based recommendation approaches.

Approch	User Profiles	Generative Model	Sequential Information	Attention Mechanism	Autoregressive Training	Ranking Loss	Model
Item-KNN [54]	✓	×	×	×	×	×	KNN
FPMC [50]	✓	×	✓	×	×	×	Markov Chains
BPR [49]	✓	×	×	×	×	✓	MF
GRU4Rec [23]	×	×	✓	×	×	✓	GRU
GRU4Rec+ [60]	×	×	✓	×	✓	✓	GRU
GRU4Rec++[22]	×	×	✓	×	✓	✓	GRU
HRNN [48]	✓	×	✓	×	✓	✓	GRU
NARM [36]	×	×	✓	✓	✓	×	GRU
EDRec [42]	×	×	✓	✓	×	×	GRU
STAMP [41]	×	×	✓	✓	✓	×	LSTM
BINN [39]	✓	×	✓	×	×	×	LSTM
3D-CNN [62]	×	×	✓	×	✓	×	3D-CNN
NextItNet [74]	×	✓	✓	×	✓	×	1D-CNN
SR-GNN [70]	×	×	✓	✓	×	×	GNN
ReLaVaR [5]	×	✓	✓	×	×	×	GRU+VAE
VRM [69]	×	✓	✓	✓	✓	×	GRU+VAE
VASER	×	✓	✓	✓	✓	✓	GRU+Flow

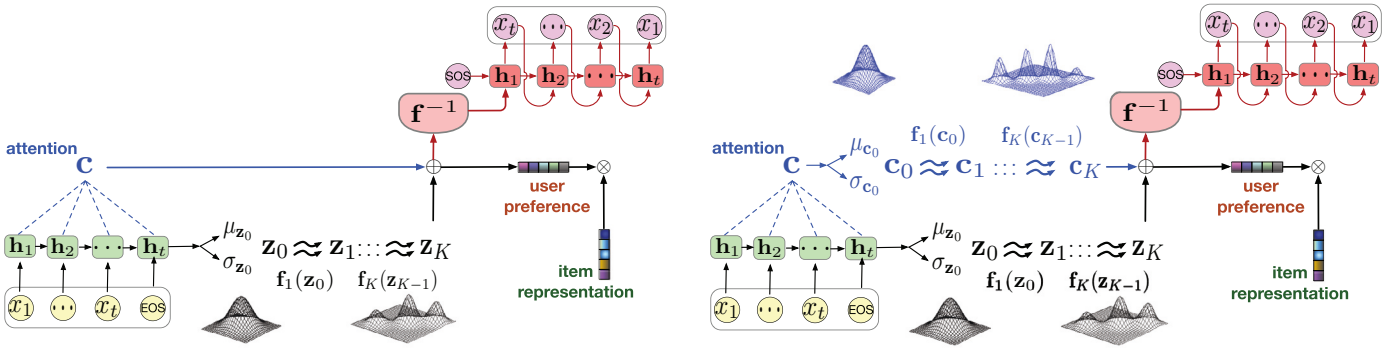


Fig. 1. Overview of VASER-DA and VASER-VA. Recommendation is made based on the posterior $q(\mathbf{z}_K)$ of the last hidden state and the attention vector ($\mathbf{c}$ for deterministic and $q(\mathbf{c}_K)$ for variational). Items are represented by embedding vectors.

main components. The learning mechanisms in these works may have some subtle differences, which will be thoroughly discussed in Section 5.

Limitations: There are at least two kinds of drawbacks that the existing RNN based SBR models may suffer: (1) They are limited to the *shallow* prediction process – i.e., they will have problems of recommending meaningful and diverse user clicks. This is due to the flat sequential generation process followed by RNNs, where each sampled click is only conditioned on the previous ones. Such a process is problematic from a probabilistic perspective, because the model is forced to generate all high-level structures locally on a step-by-step basis in a deterministic way – thereby being constrained with exploring inter-session click dependencies. (2) Although effective in modeling sequential click patterns, they have no stochastic variables at all. The decoding/predicting layer in RNN models the click distribution with autoregressive dependency, i.e., $\prod_i p(x_i|x_{1:i})$. In theory, it allows complete autoregressive factorization and could approximate any probability distributions of clicks. However, limited to the capability of real implementation (i.e., LSTM and GRU), the existing works have to resort to explicit autoregressive splitting of data to remedy the inadequate capabilities of their models which could largely improve the performance (in comparison to relying on the autoregressive nature of RNNs only).

3. Main methodologies

We propose two VASER models: (1) VASER with deterministic attention (VASER-DA); and (2) VASER with variational attention (VASER-VA) – both illustrated in Fig. 1. Each model consists 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-DA employs a deterministic attention mechanism; whereas VASER-VA leverages attention vector as a stochastic latent factor to overcome the bypassing phenomena caused by RNN and deterministic attention mechanism. Both models incorporate the normalizing flows for flexible posterior approximation. In the sequel, we present the general framework of VASER with theoretical background and training procedure, followed by the details of VASER-DA and VASER-VA.

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 ($\mathbf{z} \sim p(\mathbf{z})$; $\pi(\mathbf{z}) \propto \exp\{f_\theta(\mathbf{z})\}$):

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

where $x_{1:t-1}$ indicates the prefix click sequence preceding current click x_t, and $f_\theta(\mathbf{z})$ is a deep neural network 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. [34,40], although these work were originally designed for CF based recommendation).

This generative process is similar to the sentence generation in [27] and trajectory generation in [76], 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:

$$\log p_\theta(\mathbf{s}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{s})} \log \left[\frac{p_\theta(\mathbf{s},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{s})} \right] + \mathbb{KL}\Big[q_\phi(\mathbf{z}|\mathbf{s})||p_\theta(\mathbf{z}|\mathbf{s})\Big]$$
$$\geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{s})}\Big[\log p_\theta(\mathbf{s},\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{s})\Big] \triangleq \mathcal{L}(\mathbf{s};\theta,\phi). \tag{2}$$

There are numerous ways to optimize the ELBO, among which VAEs [33] use a parametric inference network and reparameterization of $q_\phi(\mathbf{z}|\mathbf{s})$ to alternatively maximize following reformulation:

$$\mathcal{L}_{\mathrm{VAE}}(\mathbf{s};\theta,\phi) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{s})}[\log p_\theta(\mathbf{s}) + \log p_\theta(\mathbf{z}|\mathbf{s}) - \log q_\phi(\mathbf{z}|\mathbf{s})]$$
$$= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{s})}[\log p_\theta(\mathbf{s})] - \mathbb{KL}\Big[q_\phi(\mathbf{z}|\mathbf{s})||p_\theta(\mathbf{z}|\mathbf{s})\Big], \tag{3}$$

Since the first term is a constant, then the objective of maximizing ELOB $\mathcal{L}_{\mathrm{VAE}}(\mathbf{s};\theta,\phi))$ 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.

– *Stochastic Variational Inference*: Expectation Maximization (EM) algorithm [11] can be used to optimize the variational inference and learning procedure, where the E-step and M-Step alternate until convergence. This type of mean-field variational inference is restricted to the strong assumption that each latent variable is independent and governed by its own parameters. Another popular choice is to find optimal ϕ^* with iterative gradient ascent by performing *stochastic variational inference* (SVI) [26] in the batched settings. This optimization is computationally expensive since it requires a running iterative inference for each sample. More importantly, inference parameters ϕ are updated independently from generative parameters θ, making it difficult for θ to adapt to optima [30].

– *Amortized variational inference*: Instead of stochastic inference, VAEs [33,52] proposed training generative model and inference model with neural networks (e.g., MLPs) to optimize variational parameters ϕ for a given sample $\mathbf{s}$. To expedite training, VAEs amortize the computational cost of variational inference by an inference network $q_\phi(\mathbf{z}|\mathbf{s})$ (a.k.a *encoder*). This amortized variational inference (AVI) is trained with the reparameterization trick [33] to propagate stochastic gradients from the generative model (a.k.a *decoder*) to the encoder, both of which are learned with the same loss function. Despite that it is much cheaper to compute $q_\phi(\mathbf{z}|\mathbf{s})$ than to obtain an optimal ϕ^* using SVI, there is no guarantee that inference network produces sufficiently good parameters, which in turn may yield a much looser ELBO [34].

– *Inference Gaps & Underfitting*: There are two sources of *inference suboptimality* in above variational parameter estimation with the inference networks. The first one is the *approximation gap* (APG), i.e., minimizing $\mathbb{KL}[q_\phi(\mathbf{z}|\mathbf{s})||p_\theta(\mathbf{z}|\mathbf{s})]$ is done by learning with a *tractable-but-approximate* proposal $q_\phi(\mathbf{z}|\mathbf{s})$ instead of the true posterior $p_\theta(\mathbf{z}|\mathbf{s})$. The second gap is *amortization gap* (AMG) of VAEs, which is caused by updating ϕ in an amortizing manner over the entire training set rather than optimizing for each sample individually as in SVI. The total inference gap $\mathcal{G}$ can be described with:

$$\begin{aligned}\mathcal{G} &= \log p(\mathbf{s}) - \mathcal{L}(\mathbf{s};\theta,\phi)\\ &= (\log p(\mathbf{s}) - \mathcal{L}^*(\mathbf{s};\theta,\phi))_{\mathrm{APG}} + (\mathcal{L}^*(\mathbf{s};\theta,\phi) - \mathcal{L}(\mathbf{s};\theta,\phi))_{\mathrm{AMG}}\\ &= \Big(\mathbb{KL}[q_\phi(\mathbf{z}|\mathbf{s})||p_\theta(\mathbf{z}|\mathbf{s})] - \mathbb{KL}[q^*_\phi(\mathbf{z}|\mathbf{s})||p_\theta(\mathbf{z}|\mathbf{s})]\Big)_{\mathrm{AMG}}\\ &\quad + \Big(\mathbb{KL}[q^*_\phi(\mathbf{z}|\mathbf{s})||p_\theta(\mathbf{z}|\mathbf{s})]\Big)_{\mathrm{APG}},\end{aligned} \tag{4}$$

where $q^*_\phi(\mathbf{z}|\mathbf{s})$ and $\mathcal{L}^*(\mathbf{s};\theta,\phi)$ refer to the ideal approximation and corresponding ELBO, albeit they are not easy to obtain in practice.

The two inference gaps inherent in VAEs would result in model *underfitting* [10,34]. and AMG has been found to be a more prominent cause of inference gap than APG in image datasets (e.g., MNIST and CIFAR) [10]. This would be further exacerbated for high-dimensional and sparse data – which is exactly what is encountered in SBR. Recall that maximizing the ELBO in Eq. (3) requires minimizing the KL term, where the prior $p(\mathbf{z})$ is usually assumed to be independent Gaussian. However, the posterior $p_\theta(\mathbf{z}|\mathbf{s})$ is much more complicated than Gaussian in real case – i.e., in SBR settings it is unreasonable to assume the posterior of a session is a Gaussian, since it may further increase the inference gaps.

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 [7], employing richer posterior/prior distributions can effectively reduce it. The approximation gap, caused by the encoding cost $\mathbb{KL}\Big[q_\phi(\mathbf{z}|\mathbf{s})||p_\theta(\mathbf{z}|\mathbf{s})\Big]$, is largely due to the improper assumption of the probabilistic distribution [10,32].

We leverage the flow method [51] to construct more accurate posterior approximation of the session distributions, rather than simple Gaussian assumption in existing works [5,69]. Normalizing Flows (NF) [51] 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:

$$\begin{aligned}\mathbf{z}_K = \mathbf{f}_K(\mathbf{z}_{K-1}) &= \mathbf{f}_K(\mathbf{f}_{K-1}(\mathbf{z}_{K-2}))\\ &= \mathbf{f}_K(\mathbf{f}_{K-1}(\cdots \mathbf{f}_1(\mathbf{z}_0))).\end{aligned} \tag{5}$$

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:

$$p_K(\mathbf{z}_K) = p_{K-1}(\mathbf{z}_{K-1})\left|\det \frac{d\mathbf{z}_{K-1}}{d\mathbf{z}_K}\right| = p_{K-1}(\mathbf{z}_{K-1})\left|\det \frac{d\mathbf{f}_K^{-1}(\mathbf{z}_K)}{d\mathbf{z}_K}\right| \quad (6)$$

$$= p_{K-1}(\mathbf{z}_{K-1})\left|\det\left(\frac{d\mathbf{f}_k(\mathbf{z}_{K-1})}{d\mathbf{z}_{K-1}}\right)^{-1}\right| \quad (7)$$

$$= p_0(\mathbf{z}_0)\left|\det \frac{d\mathbf{z}_1}{d\mathbf{z}_0}\right|^{-1}\cdots\left|\det \frac{d\mathbf{z}_K}{d\mathbf{z}_{K-1}}\right|^{-1} \quad (8)$$

$$= p_0(\mathbf{z}_0)\left|\det \frac{d\mathbf{z}_K}{d\mathbf{z}_0}\right|^{-1}, \quad (9)$$

where $\det \frac{d\mathbf{f}}{d\mathbf{z}}$ is the Jacobian determinant of $\mathbf{f}$. Eq. (7) is obtained with the inverse function theorem and the application of property $\det(A^{-1}) = (\det(A))^{-1}$ for matrices. Moreover, due to decomposability of determinants ($\det(AB) = \det(A)\det(B)$), we obtain Eq. (8).

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. (9) 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 [51]. 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. (9)

$$\log q_K(\mathbf{z}_K|\mathbf{s}) = \log q_{K-1}(\mathbf{z}_{K-1}|\mathbf{s}) - \log\left|\det \frac{d\mathbf{z}_K}{d\mathbf{z}_{K-1}}\right| = \log q_0(\mathbf{z}_0|\mathbf{s}) - \sum_{k=1}^{K}\log\det\left|\frac{d\mathbf{z}_k}{d\mathbf{z}_{k-1}}\right|, \quad (10)$$

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 [51], given by:

$$\mathbf{f}(\mathbf{z}) = \mathbf{z} + \mathbf{u}\sigma(\mathbf{w}^\intercal\mathbf{z} + b), \quad (11)$$

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:

$$\left|\det\frac{\partial\mathbf{f}}{\partial\mathbf{z}}\right| = \left|\det\left(\mathbf{I} + \mathbf{u}[\sigma'(\mathbf{w}^\intercal\mathbf{z} + b)\mathbf{w}]^\intercal\right)\right| = \left|1 + \mathbf{u}^\intercal\sigma'(\mathbf{w}^\intercal\mathbf{z} + b)\mathbf{w}\right|, \quad (12)$$

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. The number of parameters ($\mathbf{u}$, $\mathbf{u}$ and b) using planar flow with K flow transformations is equal to $(2d+1)IK$, where I is the number of output units of the inference network $q_\phi(\mathbf{z}_K|\mathbf{s})$.

– *Alternative flows*: There exist several alternative choices, such as Autoregressive Flows (IAF [32] and MAF [46]), real NVP [14] and Glow [31]. These flows emphasize different aspects of improving posterior approximation. For example, MAF is more efficient than IAF on density estimation but less efficient on data generation, which, in contrary, can be easily parallelized in IAF. Recent Glow model [31] achieves very high quality of data generation but is very expensive on training – e.g., a week on 40 GPUs for 256 × 256 images – and is not suitable for our case.

Next, we discuss the two specific implementations of VASER.

3.4. VASER with deterministic attention

Attention mechanism, originally used for dynamically aligning the input and output sequences [1], is an effective and robust method, and has been successfully applied in various learning tasks such as machine translation [1], dialogue generation [4] and recommendation [68]. In the case of session-based recommendation, existing works [36,41,42] encode the session intent with a deterministic attention mechanism by computing a probabilistic distribution:

$$\alpha_{it} = \frac{\exp(\mathbf{h}_i^{\text{dec}}\mathbf{W}^\intercal\mathbf{h}_t^{\text{enc}})}{\sum_{t'=1}^{N}\exp(\mathbf{h}_i^{\text{dec}}\mathbf{W}^\intercal\mathbf{h}_{t'}^{\text{enc}})}, \quad (13)$$

where $\mathbf{h}_t^{\text{enc}}$ and $\mathbf{h}_i^{\text{dec}}$ denote the t^{th} and i^{th} hidden state of the encoder and decoder, respectively, and $\mathbf{W}$ is the parameter matrix needed to be learned. The probability distribution α_{it} indeed determines the weight of each item in the input session. Then, the attention vector $\mathbf{c}_i$ is calculated by summing the weighted input:

$$\mathbf{c}_i = \sum_{t=1}^{N}\alpha_{it}\mathbf{h}_t^{\text{enc}} \quad (14)$$

which is fed into the decoder at the i^{th} step and we will denote the attention vector $\mathbf{c}_i$ as $\mathbf{c}$ for simplicity. By incorporating the VAEs into RNN based model, VASER-DA allows Bayesian inference, compared to previous SBR models [22,23,38,41].

In VASER-DA, 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

$$\begin{aligned}\mathcal{L}_{\text{DAF}}(\mathbf{s};\theta,\phi) &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{s})}\left[\log p_\theta(\mathbf{s},\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{s})\right]\\ &= \mathbb{E}_{q(\mathbf{z}_0)}[\log p_\theta(\mathbf{s},\mathbf{z}_K) - \log q(\mathbf{z}_K)]\\ &= \mathbb{E}_{q(\mathbf{z}_0)}[\log p_\theta(\mathbf{s}|\mathbf{z}_K)] - \mathbb{E}_{q(\mathbf{z}_0)}[\log q(\mathbf{z}_0)]\\ &\quad + \beta\mathbb{E}_{q(\mathbf{z}_0)}[\log p_\theta(\mathbf{z}_K)]\\ &\quad + \beta\mathbb{E}_{q(\mathbf{z}_0)}\left[\sum_{k=1}^{K}\log\det\left|\frac{d\mathbf{z}_k}{d\mathbf{z}_{k-1}}\right|^{-1}\right],\end{aligned} \quad (15)$$

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 [4]. We call this implementation deterministic attention flow (DAF).

3.5. VASER with variational attention

If we directly use the above attention vector $\mathbf{c}$, the deterministic attention may be powerful enough to reconstruct the input session and eliminate the influence of the VAE. This phenomenon is also known as the "bypassing" problem in combining VAE and RNN models [4], mainly because of the autoregressive factorization of RNN which, in theory, represents any probability distribution even without dependence on $\mathbf{z}$.

To alleviate this problem, we introduce a variational attention method inspired by recent works on machine translation [2,12] (cf. Fig. 1(b)). More specifically, we treat the attention vector $\mathbf{c}$ as another latent factor in addition to $\mathbf{z}$, both of which are combined to reconstruct the input data by maximizing the following new variational attention flow (VAF) ELBO:

$$\begin{aligned}\mathcal{L}_{\text{VAF}}(\mathbf{s};\theta,\phi) &= \mathbb{E}_{\mathbf{z},\mathbf{c}\sim q_\phi(\mathbf{z},\mathbf{c}|\mathbf{s})}\left[\log p_\theta(\mathbf{s},\mathbf{z},\mathbf{c}) - \log q_\phi(\mathbf{z},\mathbf{c}|\mathbf{s})\right]\\ &= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{s}),\mathbf{c}\sim q_\phi(\mathbf{c}|\mathbf{s})}\left[\log p_\theta(\mathbf{s},\mathbf{z},\mathbf{c}) - \log q_\phi(\mathbf{z}|\mathbf{s}) - \log q_\phi(\mathbf{c}|\mathbf{s})\right]\end{aligned} \quad (16)$$

$$
\begin{aligned}
&= \mathbb{E}_{q(\mathbf{z}_0),q(\mathbf{c}_0)}[\log p_\theta(\mathbf{s},\mathbf{z}_K,\mathbf{c}_K) - \log q(\mathbf{z}_K) - \log q(\mathbf{c}_K)] \\
&= \mathbb{E}_{q(\mathbf{z}_0),q(\mathbf{c}_0)}[\log p_\theta(\mathbf{s}|\mathbf{z}_K,\mathbf{c}_K)] \\
&\quad -\mathbb{E}_{q(\mathbf{z}_0)}[\log q(\mathbf{z}_0)] - \mathbb{E}_{q(\mathbf{c}_0)}[\log q(\mathbf{c}_0)] \\
&\quad + \beta\big(\mathbb{E}_{q(\mathbf{z}_0)}[\log p_\theta(\mathbf{z}_K)] + \mathbb{E}_{q(\mathbf{c}_0)}[\log p_\theta(\mathbf{c}_K)]\big) \\
&\quad + \beta\mathbb{E}_{q(\mathbf{z}_0),q(\mathbf{c}_0)}\left[\sum_{k=1}^{K}\log\det\left|\frac{\partial \mathbf{z}_k\mathbf{c}_k}{\partial \mathbf{z}_{k-1}\mathbf{c}_{k-1}}\right|^{-1}\right],
\end{aligned} \tag{17}
$$

which is derived based on the fact that $\mathbf{c}$ and $\mathbf{z}$ are marginally independent given $\mathbf{s}$ (Eq. (16)), and with two independent flows $q_\phi(\mathbf{z}|\mathbf{s}) := q(\mathbf{z}_K)$ and $q_\phi(\mathbf{c}|\mathbf{s}) := q(\mathbf{c}_K)$ (Eq. (17)). Similar to Eq. (15), β is used to regularize the flows.

By treating attention vector as another stochastic factor, VASER-VA does not employ the power of attention directly and, to an extent, regularizes the learning capability of RNNs. Namely, VASER-VA forces the decoder to condition more on both latent variables $\mathbf{z}$ and $\mathbf{c}$ – otherwise, the RNN decoder may totally overlook them.

3.6. Implementation details

– *Training Objective*: Since recommendation means predicting the next item in a given session, the loss function used is the cross-entropy of the prediction probability:

$$
\mathcal{L}_{\text{rec}} = -\sum_{j=1}^{M}\big[y_j\log\hat{y}_j + (1-y_j)\log(1-\hat{y}_j)\big], \tag{18}
$$

where the binary value y_j refers to the label of item x_j.

Combined with the loss of ELBO, we have the following training objectives of the proposed models:

$$
\mathcal{L} = \mathcal{L}_{\text{rec}} + (-\mathcal{L}(\mathbf{s};\theta,\phi)), \tag{19}
$$

where $\mathcal{L}(\mathbf{s};\theta,\phi)$ is either Eq. (15) or Eq. (17) depending on which VASER model is used.

– *Parameter estimation*: As we explained earlier, there exist inference gaps which may result in bias posterior approximation. A recent popular way of alleviating the inference gaps is the combination of the effectiveness of SVI and AVI. The two inference methodologies for parameter optimization are blended in [34], where the encoder's output initiates the SVI-style optimization. In a similar way, the *iterative amortized inference* [43] learns to perform inference optimization via repeatedly encoding gradients from SVI, while *semi-amortized variational inference* [30] initiates variational parameters with an AVI and subsequently updates them with SVI. A most recent work in [57] proposes to regularize the AVI so as to smooth the inference model, aiming at simultaneously improving inference and generative performance.

In this work, we leverage the effectiveness of both AVI and SVI to train our VASER models towards reducing amortization gap. Algorithm 1 illustrates the process of parameter updating for VASER-VA. We first initialize the local variational parameters ξ with $\xi(\mathbf{s})$, which is predicted by the inference network with parameters ϕ. Then, ξ is iteratively updated to approximate the optimal variational parameters ξ^* of SVI. Subsequently, θ is updated under $\mathcal{L}_{\text{VAF}}(\mathbf{s};\theta_i,\xi_J)$ and the inference network is updated with θ fixed. Note that Algorithm 1 is also applicable for VASER-DA since it only requires one latent factor $\mathbf{z}$.

Discussion: We have proposed two variational session-based models that are very similar in addition to the way of dealing with attention vector. The two VASER models share some preliminary architecture with existing works. We shortly discuss the relations and discriminate our works.

– *Attentive RNN based SBR models*: We share the main architecture with existing works except introducing Bayesian inference for session-based recommendation. For example, we also employ GRU for both encoder and decoder. Following previous work [36], we use a dynamic embedding for item representation. The only difference is that instead of random initialization, we use a unsupervised skip-gram model (e.g., word2vec [45]) to initialize the item representation. The two VASER models may degrade to existing RNN based methods by removing the stochastic inference models. In addition, the presented two latent factor models do not rely on specific underlying architectures, which means we can easily alternate the GRUs with CNN based models [61,62,75,74] that could improve the training efficiency – which is a topic of our future work.

Algorithm 1: Parameter estimation in VASER-VA.

Input: session set: $\mathbf{S}$, encoder: $q_\phi(\mathbf{z},\mathbf{c}|\mathbf{s})$, decoder: $p_\theta(\mathbf{s}|\mathbf{z},\mathbf{c})$, number of iterations: I, learning rates: η_ϕ, η_ξ and η_θ, loss function of ELBO: $-\mathcal{L}_{\text{VAF}}(\mathbf{s};\theta,\xi(\mathbf{s}))$.

Output: Variational and generative parameters ϕ and θ.

for $i = 1,\cdots,I$: **do**
  Sample $\mathbf{s}\sim\mathbf{S}$;
  Set $\xi_0 = \xi(\mathbf{s})$;
  $\mathbf{z}_0\sim(\mathbf{z}_0|\mathbf{s})$, $\mathbf{c}_0\sim(\mathbf{c}_0|\mathbf{s})$;
  $\mathbf{z}_K = \mathbf{f}_k(\mathbf{f}_{k-1}(\cdots\mathbf{f}_1(\mathbf{z}_0)))$; $\mathbf{c}_K = \mathbf{f}_k(\mathbf{f}_{k-1}(\cdots\mathbf{f}_1(\mathbf{c}_0)))$;
  Approximate ξ_j with SVI:
  for $j = 0,\cdots,J-1$: **do**
    $\xi_{j+1} = \xi_j - \eta_\xi\frac{\partial\mathcal{L}_{\text{VAF}}(\mathbf{s};\theta_i,\xi_j)}{\partial\xi_j}$;
  end
  Optimize (θ,ϕ) with AVI:
  Update θ: $\theta_{i+1} = \theta_i - \eta_\theta\triangledown_{\theta_i}\mathcal{L}_{\text{VAF}}(\mathbf{s};\theta_i,\xi_J)$;
  Update ϕ: $\phi_{i+1} = \phi_i - \eta_\phi\triangledown_{\phi_i}\mathcal{L}_{\text{VAF}}(\mathbf{s};\theta_{i+1},\xi(\mathbf{s}))$;
end

– *Variational recommendation models*: The proposed VASER models share the main idea of incorporating deep generative models into recommender systems with existing works [5,8,29,35,38,40,69]. However, these VAE based models are either non-applicable for the SBR scenario or inaccurate on stochastic inference. They usually presume a prior of latent variables and minimize the KL terms in Eq. (3), which is the main cause of approximate inference gap and thus cannot accurately model the complicated ground-truth distribution. On the contrary, we will show that the approaches of applying NF on posterior approximation and variational attention can indeed enhance these collaborative and sequential VAE models.

3.7. Computational complexity

The time complexity of the GRU module is $O(N_s(H^2+HN_o))$, where N_s is the number of items in all sessions, H is the number of hidden units, and N_o is the number of outputs. The attention mechanism in VASER leads to the time complexity of $O(rHN)$, where r is the number of attention vectors, and N denotes the length of the input and output sessions. Compared to previous work [23,36,41], the only overhead of VASER is the invertible transformations in normalizing flows. In particular, it needs to compute the log(det)-Jacobian term which requires in $O(Kd)$ time based on the matrix determinant lemma [19], where d is the dimension of hidden layers and K is the number of transformations. We will investigate the efficiency of the proposed model in next section.

4. Experimental observations

We now describe the experimental settings and report the empirical evaluation results for the following questions:

Table 3
Statistics of the datasets.

Datasets	YOOCHOOSE 1/64	YOOCHOOSE 1/4	DIGINETICA
#clicks	557,248	832,6407	982,961
#train sessions	355,385	621,6184	719,470
#test sessions	52,956	56,616	60,858
#items	17,626	30,903	43,097
avg. session length	6.27	5.83	5.12

- **Q1** How does VASER perform compared with the state-of-the-art session-based recommendation methods?
- **Q2** Is the modeling of variational inference and variational attention helpful for learning more desirable sequential interactions for session-based recommendation?
- **Q3** How do the key hyper-parameters affect VASER's performance?
- **Q4** Can we extend VASER to conventional collaborative recommendation and how does it perform?

4.1. Datasets, baselines, metrics and settings

Datasets. For fair comparison, we evaluate different methods on two real-world transaction datasets, YOOCHOOSE[1] and DIGINETICA[2], which have been widely used for evaluating SBR approaches. Following previous works [36,41,60], 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) For YOOCHOOSE and DIGINETICA, we respectively use the sessions of subsequent day and subsequent week 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 of YOOCHOOSE by time and train all models on more recent fractions (i.e., 1/64 and 1/4) of training sessions. Table 3 shows the statistics of the datasets.

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

(1) Traditional sequence-based recommendation methods:

- **POP**: It simply recommends the items with the largest number of interactions in the training set.
- **Item-KNN** [54]: It is an item-to-item model that recommends items that are similar to previously visited items based on cosine similarity.
- **FPMC** [50]: It is a sequential recommendation approach combining factorized Markov chains with the factorization of the user-item matrix. We adapt it into session-based recommendation scenario by omitting the user latent representations when computing recommendation scores.
- **BPR** [49]: It is one of widely used matrix factorization (MF) methods, which optimizes a pairwise ranking objective function via stochastic gradient descent. Since MF can not be directly applied to session-based recommendation, we compute the similarity scores between a candidate item and the items within the session to make recommendations.

(2) RNN-based SBR methods:

- **GRU4Rec**[3] [23]: 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.
- **GRU4Rec+** [60]: It is an improved version of GRU4Rec that adopts data augmentation and accounts for shifts in the input data distribution to improve the performance of GRU4Rec.
- **NARM**[4] [36]: 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** [41]: 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.
- **GRU4Rec++** [22]: It is a most recent method that extends the GRU4Rec by introducing an improved sampling strategy.

Since our main focus is not on the combination of various features, we omit the comparison with the content-based sequential recommendation models such as [24,39,48,59]. We also note that our contribution is not the model efficiency (CNN based models like [61,62,74] are not considered in our experiments).

(3) VAE-based recommendation methods:

- **ReLaVaR** [5]: 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** [69]: 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.

Metrics. Following previous works [21,22,36,41], 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 [41]. 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.

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 [36,41], we truncated BPTT using a fixed window of 19 time-steps for DIGNETICA and 30 time-steps for the two YOOCHOOSE datasets. Also following [36,41], 10% of the training data are used as the validation set. For the two VASER models, parameters d, K and β are respectively 100, 16 and 0.2, if not specified.

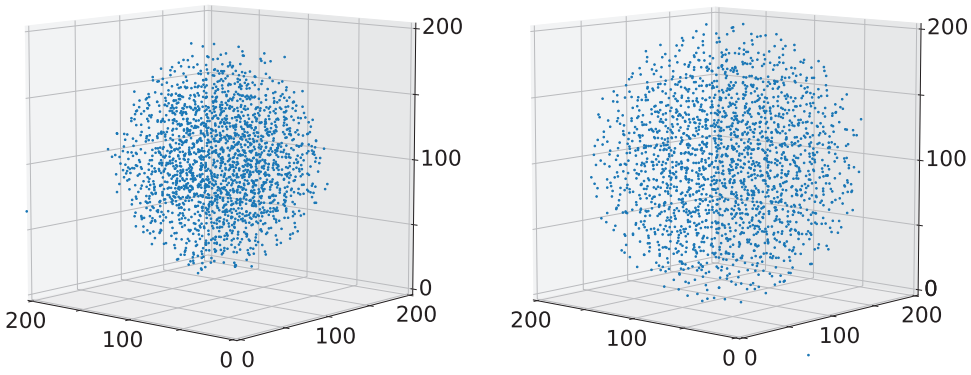
4.2. Overall performance (Q1 & Q2)

4.2.1. Comparison against SBR baselines (Q1)

Table 5 shows the results of comparison to the existing state-of-the-art SBR methods, from which we can clearly observe that the proposed two models perform the best on two metrics throughout three 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. 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

[1] http://2015.recsyschallenge.com/challenge.html
[2] http://cikm2016.cs.iupui.edu/cikm-cup
[3] https://github.com/hidasib/GRU4Rec
[4] https://github.com/lijingsdu/sessionRec_NARM



(a) NARM. (b) VASER-VA.

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

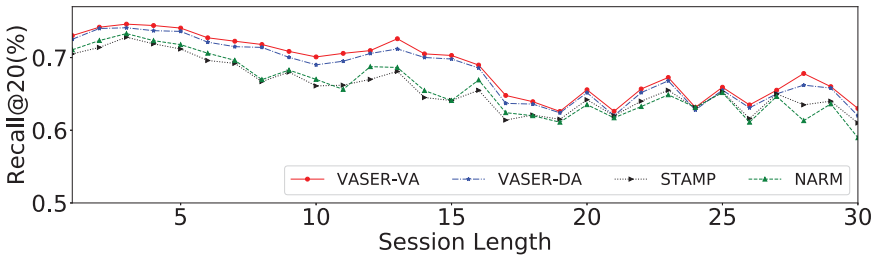


Fig. 3. Impact of session length (YOOCHOOSE 1/64).

work GRU4Rec++ does not exhibit expected results on the three 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 two models outperform the best baseline (either NARM or STAMP) by a significant margin. Take the DIGINETICA dataset for example, VASER-VA achieves 2.2% and 3.6% boost over NARM on Recall@20 and MRR@20, respectively – despite that they are relatively small values, we note that they can be considered as statistically significant (p-*value* < 0.01 of paired t-test) on SBR task over the three datasets that are originally used for CIKM and RecSys challenges. 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 Fig. 2, where the encoding space of NARM and VASER-VA 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-VA is the combination of the posterior of hidden state $q(\mathbf{z}_K)$ and the posterior of variational attention $q(\mathbf{c}_k)$. Apparently, VASER-VA 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 Fig. 3 (results on DIGINETICA and YOOCHOOSE 1/4 are consistent, but omitted due to the lack of space), whereby we compared to the two best baselines. Our models 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.

4.2.2. *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. On the other hand, the only difference between the proposed two models is the way of treating attention vector. In Table 5, we find that VASER-VA always performs better than VASER-DA, which proves the effectiveness of the variational attention mechanism. By treating attention vector as an auxiliary vector, we can successfully weaken the effectiveness of deterministic attention network at the beginning of training, and enforce more useful information to be encoded into the latent space, which is an effective way of alleviating the bypassing problem in modeling sequence data [2,12].

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 no category information associated with items for the DIGNETICA dataset. In fact, the category information plays a less important role in improving the recommendation performance according to the results in [69], largely due to the *extremely* sparse category labels on the YOOCHOOSE dataset. 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 [10]. This is in accordance with the findings in modeling language with

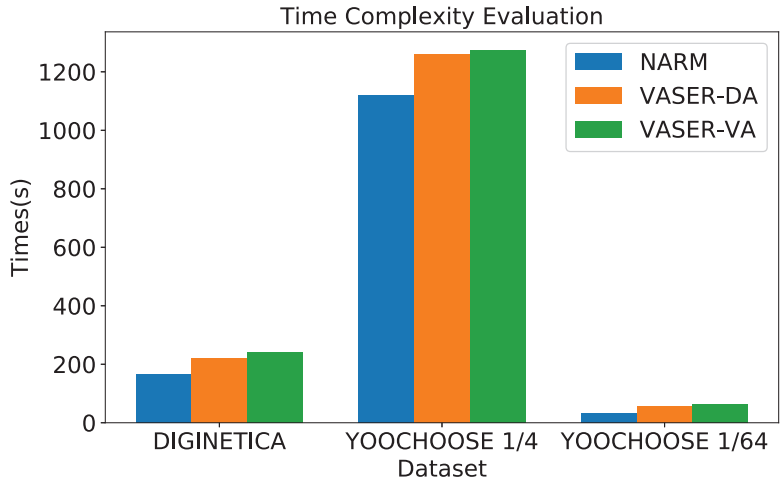
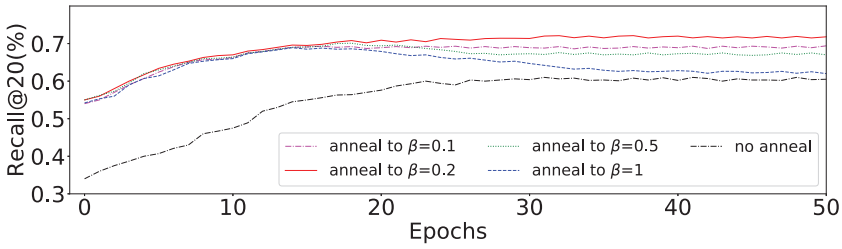
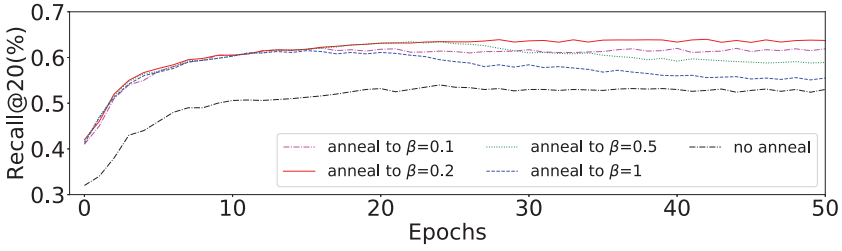


Fig. 4. Average runtime of one training epoch.



(a) YOOCHOOSE 1/64.



(b) DIGINETICA.

Fig. 5. The impact of β.

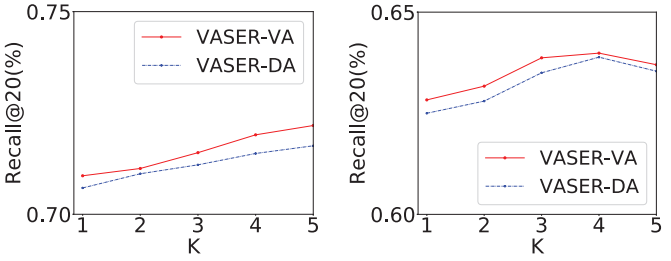
vanilla VAEs [4,27], i.e., the autoregressive models are powerful enough to decode the entire sequence, resulting in uselessness of stochastic latent factors. More importantly, these methods approximate an improper assumed distribution $q_\phi(\mathbf{z}|\mathbf{s})$, e.g., the choice of diagonal-covariance Gaussian in [4,27], and thus are subjected to heavy bias inference problem, as explained in Section 3.2. In contrast, our two VASER models can largely alleviate this problem benefiting from the normalizing flows with flexible posterior approximation.

As mentioned before, the main computational overhead of VASER compared to previous attention-based RNN models [36,41] is from computing the invertible transformations. The comparison of average runtime for one training epoch (as shown in Fig. 4) for our models against NARM – we omitted other methods (e.g., STAMP and GRU4Rec++) becasue their computational complexity is similar to NARM – show that our two models require slightly more time compared to NARM. On the other hand, VASER-VA needs more time than VASER-DA since it consists of an extra flow transformation, which, is negligible especially on larger datasets (e.g., YOOCHOOSE 1/4).

4.3. Impact of parameters (Q3)

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

Fig. 5 shows the impact of β on VASER-VA, 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 (cf. Eq. (17)) are usually ordered larger than the reconstruction term. Without annealing or annealing to a larger value, the performance of VASER models 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 models 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 Hu et al. [27].



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

Fig. 6. The impact of *K*.

Fig. 6 investigates the impact of *K* on two datasets. The planar flows used in VASER modifies the initial density by applying a series 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 two models. Since the RHS of Eq. (11) 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 [32]. This is demonstrated by the results on YOOCHOOSE 1/64 dataset in Fig. 6(a), where the models require more transformations to obtain higher performance. Yet, it is not the case for DIGINETICA in Fig. 6(b), on which too many transformations may result in overfitting problem (we note that in both Fig. 6(b) and (a) the X-axes are in log_2 scales – i.e., the values are 2, 4, 8, 16, and 32). We hypothesize that the reason is that the planar flows need more coupling layer to approximate the distribution of latent factors for smaller/sparser datasets – recall that only 1/64 of sessions are used for training for YOOCHOOSE 1/64 dataset. This result is in accordance the recent discovery of augmenting normalizing flows for density estimation on image data [3].

4.4. Comparison against CF-based VAEs (Q4)

Finally, we would like to investigate whether the proposed models, or more specifically, the normalizing flows, can be used to improve the recommendation performance on collaborative settings compared to CF-based VAE methods.

To adapt VASER to collaborative recommendation scenario, we simply replace the RNNs in VASER with the MLPs, resulting in the method called *Collaborative Normalizing Flows* (CNF). The evaluation is conducted on the citeulike-a dataset used in [38], which contains 5,551 users and 16,980 articles with 204,986 observed user-item pairs. We compare CNF with the following three collaborative VAE models:

- **CVAE** [38] is the first collaborative VAE based item recommendation method, which uses vanilla VAE for item representation.
- **CLVAE** [35] is a conditional ladder VAE [58] based recommendation method which extends the CVAE with hierarchical VAE structure.
- **MVAE** [40] is very similar to CVAE except that it uses multinomial conditional likelihood as the prior.

We follow all the experimental settings in [38] but do not consider the side information such as ratings and comments. In the same spirit, we exclude the comparison with other VAE based methods that mainly leverage side information for recommendation, such as the work in [8,29,72].

Table 4 shows that CNF, by leveraging normalizing flows for posterior approximation, significantly outperforms the baselines.

Table 4
The performance comparison on citeulike-a. P@k, R@k, MAP@k and nDCG@k is the precision, recall, mean average precision, and normalized discounted gain for top k recommendation.

	P@5	P@10	R@5	R@10	MAP@10	nDCG@10
CVAE	0.146	0.122	0.129	0.191	0.328	0.375
CLVAE	0.144	0.119	0.122	0.188	0.321	0.372
MVAE	0.154	0.122	0.137	0.202	0.343	0.362
CNF	**0.167**	**0.143**	**0.152**	**0.215**	**0.381**	**0.435**

This, again, proves our motivation that flexible approximation techniques such as normalizing flows should be considered in deep generative recommender systems. Among the baselines, we observe that: (1) modeling with multinomial conditional likelihood may slightly improve the performance, which conforms to the observation in [40]; and (2) modeling hierarchical VAE on items is not effective on this dataset – we conjecture that CLVAE may suffer overfitting without the side information used in [35].

5. Related work

We now review the relevant literatures from two basic perspectives, and position our work in that context.

5.1. Sequential recommendation

Session-based recommendation [67] is essentially a sequence learning problem [47] 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. Nonparametric 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 [13,28,54,18,44]. Naturally, other sophisticated sequence learning approaches can also be adapted to solve the session-based recommendation problem, which incurs MC based models [66] and hybrid CF models like FPMF [21,50], etc.

RNN models such as LSTM [25] and GRU [9] have been successfully applied in many sequence learning tasks, such as machine translation [1], human mobility learning [15], and session-based recommendation [22,23,36,39,41,48]. GRU4Rec [23] 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 [36,42] 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. STAMP [41] distinguished user interests drift caused by unintended clicks with a priority model to capture users' general interests from the long-term memory of a session context, while taking into account users' current interests from the short-term memory of the last-clicks. Another work [39] combined users' history preference and short-term preference for SBR, which requires the knowledge of long-term user behavior and models the sequential behavior with LSTM. Most recently, Hidasi et al. [22] improved their GRU4Rec model by introducing tailored ranking loss functions. A hierarchical RNN model is used for SBR in [48], which models both inter-session and intra-session patterns with a hierarchical RNN. However, it requires the knowledge about the session users to construct hierarchical RNN models [67]. Although achieving current state-of-the-art performance on SBR task, the sampling

Table 5
Performance comparison among all session-based recommendation methods over three datasets.

	YOOCHOOSE 1/64		YOOCHOOSE 1/4		DIGINETICA	
	Recall@20(%)	MRR@20(%)	Recall@20(%)	MRR@20(%)	Recall@20(%)	MRR@20(%)
POP	8.48	3.52	1.36	0.31	0.91	0.23
item-KNN	53.12	22.13	52.43	21.75	28.35	9.45
FPMC	47.39	19.28	-	-	33.07	8.92
BPR	33.11	13.79	3.43	1.57	15.19	8.68
GRU4Rec	62.40	25.36	59.58	22.62	43.82	15.46
GRU4Rec+	69.35	28.70	69.16	29.23	57.95	24.93
GRU4Rec++	68.43	28.55	68.97	28.41	57.74	26.23
NARM	70.13	29.38	69.75	29.30	62.58	27.35
STAMP	70.21	29.22	70.45	29.47	62.03	27.28
ReLaVaR	64.31	25.26	60.53	22.76	54.95	23.76
CRM	69.32	28.75	68.22	28.35	60.06	26.07
VASER-DA	**71.85**	**30.05**	**70.74**	**29.75**	**63.67**	**28.27**
VASER-VA	**72.12**	**30.33**	**70.96**	**29.90**	**63.99**	**28.34**

tricks and loss functions used are inherently popularity-based and thus may not be easily generalized to all datasets, as argued by the authors in [22].

Existing RNN based methods are limited to the *shallow* generation process, i.e., having problems of generating meaningful and diverse user clicks. This is caused by the flat sequential generation process followed by RNNs, where each token is sampled conditioned only on previous ones. This process is problematic from a probabilistic perspective, because the model is forced to generate all high-level structure locally on a step-by-step basis, which has been investigated in dialogue generation [56] and mobility behavior generation [76].

We note that there are several works [61,62,75,74] emerged recently to capture sequential patterns with Convolutional Neural Networks (CNNs), largely inspired by recent advances in machine translation [16] using CNNs to replace RNNs – the latter has dominated this area in the last decade. The main motivation of these work is to overcome the parallelization problem of RNN models, namely RNNs depend on a hidden state of the entire past input that cannot fully utilize parallel computation within a sequence. Caser [61], for example, embeds the previous t items into $k-$dimensional vectors and forming a $t \times k$ matrix like an image, which can be successfully convoluted with various CNN based models. In addition to computational efficiency, Caser is capable of capturing both “skip” and “union-level” behaviors compared to previous sequential models. Another joint model [75] learns features associated with a session with residual CNN and the sequential behavior with LSTM. A most recent work [74] improves Caser by introducing a more complicated network NextItNet with dilated convolution [74] and residual blocks [20].

Table 2 summarized the main SBR models in the literature.

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 [37,53,55,64,65,71] 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. The first Bayesian deep generative model called collaborative variational autoencoder (CVAE) was proposed in [38], which jointly models the generation of content and the rating information using vanilla VAE [33] in a collaborative filtering (CF) setting. Lee et al. [35] augmented CF with ladder VAE [58] and leveraged generative adversarial nets (GAN) to regularize the proposed collaborative recommendation models. Liang et al. [40] found that VAE model suffers from underfitting when modeling large, sparse, high-dimensional data [34], and presented a multinomial conditional likelihood based VAE framework [40]. Several recent works extend the ideas of applying VAEs to CF-based recommendation but mainly focus on combining various auxiliary features [8,29]. It is worthwhile to mention that a most recent work [63] leverages VAE to learn users' latent interest space and generate plausible appealing new items that *do not exist* in the training set, although its main topic is out of the scope of this work. The most related work are ReLaVaR [5] and VRM/CVRM [69], 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.

Key differences: Our work differs from the above mentioned works in several ways. Compared to existing sequential recommendation approaches, we model the problem within a probabilistic recommendation setting which allows our model for Bayesian inference. In addition to learning sequential behavior of users, our model is capable of capturing non-linear user-item interactions. On the other hand, prior collaborative VAE approaches either model auxiliary information with VAEs or exploit VAEs for richer representations where user profile and long-term preference is available. Thus, these methods are not suitable for SBR task as the latter is restricted to short-term sequence learning: neither allowing long-term preference learning nor scrutinizing user profiles. Furthermore, our proposed models at least have three differences over the existing collaborative and session-based VAE recommendation methods, i.e., (1) we derive novel flow-based ELBOs tailored for SBR task with the flexible posterior approximation, rather than presumed Gaussian distribution in previous work; (2) we introduce a training method combining the effectiveness of both stochastic and amortized variational inference for addressing the inference gap problem in the settings of session-based recommendation; and (3) we explore the way of treating attention vector as latent factor to enhance variational session-based inference and to overcome the KL vanishing problem inherent in combining VAEs with autoregressive models.

6. Conclusions and future work

We presented VASER, a normalizing flow based generative framework for learning sequential click patterns. The proposed two models implementing VASER enable learning non-linear interactions between user-clicks while allowing Bayesian inference, achieving significant improvements for the session-based recommendation problem in comparison to existing methods. There are two important observations in this work: (1) instead of using

amortized inference as in existing collaborative/sequential variational recommendation methods, flow based techniques could effectively improve the density approximation and deserve more attention in the recommendation community; (2) attention mechanism, widely used in existing works, should be carefully treated in the variational recommendation models, for example, considering it as an another latent factor in VASER. 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., using CNNs).

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Ting Zhong: Conceptualization, Methodology, Writing - original draft, Resources. **Zijing Wen:** Software, Validation, Formal analysis, Investigation. **Fan Zhou:** Project administration, Writing - review & editing, Funding acquisition. **Goce Trajcevski:** Writing - review & editing, Funding acquisition. **Kunpeng Zhang:** Writing - review & editing.

Acknowledgements

This work was supported by National Natural Science Foundation of China (Grant No.61602097, 61472064, U19B2028 and 61772117), NSF grants III 1213038 and CNS 1646107, and ONR grant N00014-14-10215.

References

[1] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: International Conference on Learning Representations (ICLR), 2015.
[2] H. Bahuleyan, L. Mou, O. Vechtomova, P. Poupart, Variational attention for sequence-to-sequence models, in: International Conference on Computational Linguistics (COLING), 2018.
[3] R. van den Berg, L. Hasenclever, J.M. Tomczak, M. Welling, Sylvester normalizing flows for variational inference, in: 34th Conference on Uncertainty in Artificial Intelligence (UAI), 2018.
[4] S.R. Bowman, L. Vilnis, O. Vinyals, A.M. Dai, R. Józefowicz, S. Bengio, Generating sentences from a continuous space, in: The SIGNLL Conference on Computational Natural Language Learning (CoNLL), 2016.
[5] S.P. Chatzis, P. Christodoulou, A.S. Andreou, Recurrent latent variable networks for session-based recommendation, Workshop on Deep Learning for Recommender Systems (DLRS@RecSys), 2017.
[6] T. Chen, W. Zhang, Q. Lu, K. Chen, Z. Zheng, Y. Yu, Svdfeature: a toolkit for feature-based collaborative filtering, J. Mach. Learn. Res. (JMLR) (2012).
[7] X. Chen, D.P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, P. Abbeel, Variational lossy autoencoder, in: International Conference on Learning Representations (ICLR), 2017.
[8] Y. Chen, M. de Rijke, A collective variational autoencoder for top-n recommendation with side information, Workshop on Deep Learning for Recommender Systems (DLRS@RecSys), 2018.
[9] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv:1412.3555 (2014).
[10] C. Cremer, X. Li, D.K. Duvenaud, Inference suboptimality in variational autoencoders, in: International Conference on Machine Learning (ICML), 2018.
[11] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the em algorithm, J. Roy. Stat. Soc. (1977).
[12] Y. Deng, Y. Kim, J. Chiu, D. Guo, A.M. Rush, Latent alignment and variational attention, Advances in neural information processing systems (NIPS), 2018.
[13] M. Deshpande, G. Karypis, Item-based top-n recommendation algorithms, ACM Trans. Inform. Syst. (TOIS) (2004).
[14] L. Dinh, J. Sohl-Dickstein, S. Bengio, Density estimation using real nvp, in: International Conference on Learning Representations (ICLR), 2017.
[15] Q. Gao, F. Zhou, K. Zhang, G. Trajcevski, X. Luo, F. Zhang, Identifying human mobility via trajectory embeddings, in: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 2017, pp. 1689–1695.
[16] J. Gehring, M. Auli, D. Grangier, D. Yarats, Y.N. Dauphin, Convolutional sequence to sequence learning, in: International Conference on Machine Learning (ICML), 2017.
[17] A. Goyal, A. Sordoni, M.-A. Côté, N.R. Ke, Y. Bengio, Z-forcing: Training stochastic recurrent networks, Advances in neural information processing systems (NIPS), 2017.
[18] H. Guo, R. Tang, Y. Ye, F. Liu, Y. Zhang, An adjustable heat conduction based knn approach for session-based recommendation, arXiv preprint arXiv:1807.05739 (2018).
[19] D.A. Harville, Matrix Algebra From a Statistician's Perspective, Springer Science & Business Media, 2006.
[20] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[21] R. He, J. McAuley, Fusing similarity models with markov chains for sparse sequential recommendation, in: IEEE International Conference on Data Mining (ICDM), 2016.
[22] B. Hidasi, A. Karatzoglou, Recurrent neural networks with top-k gains for session-based recommendations, in: ACM International Conference on Information and Knowledge Management (CIKM), 2018.
[23] B. Hidasi, A. Karatzoglou, L. Baltrunas, D. Tikk, Session-based recommendations with recurrent neural networks, in: International Conference on Learning Representations (ICLR), 2016.
[24] B. Hidasi, M. Quadrana, A. Karatzoglou, D. Tikk, Parallel recurrent neural network architectures for feature-rich session-based recommendations, in: ACM Conference on Recommender Systems (RecSys), 2016.
[25] S. Hochreiter, J. Schmidhuber, Long short-term memory., Neur. Comput. (1997).
[26] M.D. Hoffman, D.M. Blei, C. Wang, J.W. Paisley, Stochastic variational inference, J. Mach. Learn. Res. (JMLR) (2013).
[27] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, E.P. Xing, Toward controlled generation of text, in: International Conference on Machine Learning (ICML), 2017.
[28] D. Jannach, M. Ludewig, When recurrent neural networks meet the neighborhood for session-based recommendation, in: ACM Conference on Recommender Systems (RecSys), 2017.
[29] G. Karamanolakis, K.R. Cherian, A.N.P.o.t. 3rd, 2018, Item recommendation with variational autoencoders and heterogeneous priors, Workshop on Deep Learning for Recommender Systems (DLRS@RecSys), 2018.
[30] Y. Kim, S. Wiseman, A.C. Miller, D. Sontag, A.M. Rush, Semi-amortized variational autoencoders, in: International Conference on Machine Learning (ICML), 2018.
[31] D.P. Kingma, P. Dhariwal, Glow: Generative flow with invertible 1x1 convolutions, Advances in neural information processing systems (NIPS), 2018.
[32] D.P. Kingma, T. Salimans, R. Józefowicz, X. Chen, I. Sutskever, M. Welling, Improving variational autoencoders with inverse autoregressive flow, Advances in neural information processing systems (NIPS), 2016.
[33] D.P. Kingma, M. Welling, Auto-encoding variational bayes, in: International Conference on Learning Representations (ICLR), 2014.
[34] R.G. Krishnan, D. Liang, M.D. Hoffman, On the challenges of learning with inference networks on sparse, high-dimensional data, in: International Conference on Artificial Intelligence and Statistics (AISTATS), 2018.
[35] W. Lee, K. Song, I.-C. Moon, Augmented variational autoencoders for collaborative filtering with auxiliary information, in: ACM International Conference on Information and Knowledge Management (CIKM), 2017.
[36] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, J. Ma, Neural attentive session-based recommendation, in: ACM International Conference on Information and Knowledge Management (CIKM), 2017.
[37] S. Li, J. Kawale, Y. Fu, Deep collaborative filtering via marginalized denoising auto-encoder, in: ACM International Conference on Information and Knowledge Management (CIKM), 2015.
[38] X. Li, J. She, Collaborative variational autoencoder for recommender systems, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2017.
[39] Z. Li, H. Zhao, Q. Liu, Z. Huang, T. Mei, E. Chen, Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018.
[40] D. Liang, R.G. Krishnan, M.D. Hoffman, T. Jebara, Variational autoencoders for collaborative filtering, in: Proceedings of the International Conference on World Wide Web (WWW), 2018.
[41] Q. Liu, Y. Zeng, R. Mokhosi, H. Zhang, Stamp: Short-term attention/memory priority model for session-based recommendation, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018.
[42] P. Loyola, C. Liu, Y. Hirate, Modeling user session and intent with an attention-based encoder-decoder architecture, in: ACM Conference on Recommender Systems (RecSys), 2017.
[43] J. Marino, Y. Yue, S. Mandt, Iterative amortized inference, in: International Conference on Machine Learning (ICML), 2018.
[44] F. Mi, B. Faltings, Context tree for adaptive session-based recommendation, arXiv preprint arXiv:1806.03733 (2018).
[45] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, Advances in neural information processing systems (NIPS), 2013.
[46] G. Papamakarios, I.M. 0001, T. Pavlakou, Masked autoregressive flow for density estimation, Advances in neural information processing systems (NIPS), 2017.

[47] M. Quadrana, P. Cremonesi, D. Jannach, Sequence-aware recommender systems, ACM Comput. Surv. (2018).
[48] M. Quadrana, A. Karatzoglou, B. Hidasi, P. Cremonesi, Personalizing session-based recommendations with hierarchical recurrent neural networks, in: ACM Conference on Recommender Systems (RecSys), 2017.
[49] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, Bpr: Bayesian personalized ranking from implicit feedback, in: The Conference on Uncertainty in Artificial Intelligence (UAI), 2009.
[50] S. Rendle, C. Freudenthaler, L. Schmidt-Thieme, Factorizing personalized markov chains for next-basket recommendation, in: Proceedings of the International Conference on World Wide Web (WWW), 2010.
[51] D.J. Rezende, S. Mohamed, Variational inference with normalizing flows, in: International Conference on Machine Learning (ICML), 2015.
[52] D.J. Rezende, S. Mohamed, D. Wierstra, Stochastic backpropagation and approximate inference in deep generative models, in: International Conference on Machine Learning (ICML), 2014.
[53] R. Salakhutdinov, A. Mnih, G.E. Hinton, Restricted boltzmann machines for collaborative filtering, in: International Conference on Machine Learning (ICML), 2007.
[54] B.M. Sarwar, G. Karypis, J.A. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: Proceedings of the International Conference on World Wide Web (WWW), 2001.
[55] S. Sedhain, A.K. Menon, S. Sanner, L. Xie, Autorec: Autoencoders meet collaborative filtering, in: Proceedings of the International Conference on World Wide Web (WWW), 2015.
[56] I.V. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A.C. Courville, Y. Bengio, A hierarchical latent variable encoder-decoder model for generating dialogues, in: AAAI Conference on Artificial Intelligence, 2017.
[57] R. Shu, H.H. Bui, S. Zhao, M.J. Kochenderfer, S. Ermon, Amortized inference regularization, Advances in neural information processing systems (NIPS), 2018.
[58] C.K. Sønderby, T. Raiko, L. Maaløe, S.K. Sønderby, O. Winther, Ladder variational autoencoders, Advances in neural information processing systems (NIPS), 2016.
[59] Y. Song, J.-G. Lee, Augmenting recurrent neural networks with high-order user-contextual preference for session-based recommendation, arXiv preprint arXiv:1805.02983 (2018).
[60] Y.K. Tan, X. Xu, Y. Liu, Improved recurrent neural networks for session-based recommendations, Workshop on Deep Learning for Recommender Systems (DLRS@RecSys), 2016.
[61] J. Tang, K. Wang, Personalized top-n sequential recommendation via convolutional sequence embedding, in: International Conference on Web Search and Data Mining (WSDM), 2018.
[62] T.X. Tuan, T.M. Phuong, 3d convolutional networks for session-based recommendation with content features, in: ACM Conference on Recommender Systems (RecSys), 2017.
[63] T.V. Vo, H. Soh, Generation meets recommendation: proposing novel items for groups of users, in: ACM Conference on Recommender Systems (RecSys), 2018.
[64] H. Wang, X. Shi, D.-Y. Yeung, Collaborative recurrent autoencoder: Recommend while learning to fill in the blanks, Advances in neural information processing systems (NIPS), 2016.
[65] H. Wang, N. Wang, D.-Y. Yeung, Collaborative deep learning for recommender systems, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2015.
[66] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, X. Cheng, Learning hierarchical representation model for nextbasket recommendation, in: International Conference on Research and Development in Information Retrieval (SIGIR), 2015.
[67] S. Wang, L. Cao, Y. Wang, A survey on session-based recommender systems, arXiv preprint arXiv:1902.04864 (2019).
[68] X. Wang, L. Yu, K. Ren, G. Tao, W. Zhang, Y. Yu, J. Wang, Dynamic attention deep model for article recommendation by learning human editors' demonstration, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2017.
[69] Z. Wang, C. Chen, K. Zhang, Y. Lei, W. Li, Variational recurrent model for session-based recommendation, in: ACM International Conference on Information and Knowledge Management (CIKM), 2018.
[70] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, T. Tan, Session-based recommendation with graph neural networks, in: AAAI Conference on Artificial Intelligence, 2019.
[71] Y. Wu, C. DuBois, A.X. Zheng, M. Ester, Collaborative denoising auto-encoders for top-n recommender systems, in: International Conference on Web Search and Data Mining (WSDM), 2016.
[72] T. Xiao, S. Liang, H. Shen, Z. Meng, Neural variational hybrid collaborative filtering, arXiv preprint arXiv:1810.05376 (2018).
[73] F. Yuan, G. Guo, J.M. Jose, L. Chen, H. Yu, W. Zhang, Lambdafm: Learning optimal ranking with factorization machines using lambda surrogates, in: ACM International Conference on Information and Knowledge Management (CIKM), 2016.
[74] F. Yuan, A. Karatzoglou, I. Arapakis, J.M. Jose, X. He, A Simple Convolutional Generative Network for Next Item Recommendation, Proceedings of the International Conference on Web Search and Data Mining (WSDM) (2019).
[75] L. Zhang, P. Liu, J.A. Gulla, A deep joint network for session-based news recommendations with contextual augmentation, in: ACM Conference on Hypertext and Social Media (HT), 2018.
[76] F. Zhou, Q. Gao, G. Trajcevski, K. Zhang, T. Zhong, F. Zhang, Trajectory-user linking via variational autoencoder, in: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 2018, pp. 3212–3218.

Ting Zhong received the B.S. degree in computer application and the M.S. degree in computer software and theory from Beijing Normal University, Beijing, China, respectively, in 1999 and 2002, respectively, and the Ph.D. degree in information and communication engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2009, where she was a Lecturer (2003 – 2009) and has been an Associate Professor, since 2010. Her research interests include deep learning, social networks, and cloud computing.

Zijing Wen received the B.S. degree in software engineering from University of Electronic Science and Technology of China in 2016. He is currently pursuing the M.S. degree with the University of Electronic Science and Technology of China. His current research interests include machine learning, spatio-temporal data management, social network knowledge discovery, etc.

Fan Zhou received the B.S. degree in computer science from Sichuan University, China, in 2003, and the M.S. and Ph.D. degrees from the University of Electronic Science and Technology of China, in 2006 and 2012, respectively, where he is currently an Associate Professor with the School of Information and Software Engineering. His research interests include machine learning, spatio-temporal data management, social network knowledge discovery, etc.

Goce Trajcevski received the B.Sc. degree from the University of Sts. Kiril i Metodij, and the M.S. and Ph.D. degrees from the University of Illinois at Chicago. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Iowa State University. His main research interests are in the areas of spatio-temporal data management, uncertainty and reactive behavior management in different application settings, and incorporating multiple contexts. In addition to a book chapter and three encyclopedia chapters, hehas coauthored over 140 publications in refereed conferences and journals. His research has been funded by the NSF, ONR, BEA, and Northrop Grumman Corp. He was the General Co-Chair of the IEEE ICDE 2014, ACM SIGSPATIAL 2019, the PC Co-Chair of the ADBIS 2018 and ACM SIGSPATIAL 2016 and 2017, and has served in various roles in organizing committees in numerous conferences and workshops. He is an Associate Editor of the ACM TSAS and the Geoinformatica Journals.

Kunpeng Zhang received the Ph.D. degree in computer science from Northwestern University. He is a Researcher in the area of large-scale data analysis, with particular focuses on social data mining, image understanding via machine learning, social network analysis, and natural language processing. He is currently an Assistant Professor with the Department of Information Systems, Smith School of Business, University of Maryland, College Park, MA, USA. He has published papers in the area of social media, artificial intelligence, network analysis, and information systems on top conference and journals. He serves as program committees for many conferences and Associate Editors for journals.